



Σχολή Θετικών Επιστημών και Τεχνολογίας

Πληροφορική

Πτυχιακή Εργασία

Σχεδίαση και Υλοποίηση διαδικτυακής πλατφόρμας για  
εξυπηρέτηση παιχνιδιών καρτών απομακρυσμένων παικτών

Νικόλας Ψύλλου

Επιβλέπων καθηγητής: Μηνάς Δασυγένης

Πάτρα, Ιούλιος 2023

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή («συγγραφέας/δημιουργός») που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο ΕΑΠ, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.



Σχεδίαση και Υλοποίηση διαδικτυακής πλατφόρμας για  
εξυπηρέτηση παιχνιδιών καρτών απομακρυσμένων παικτών

Νικόλας Ψύλλου

Επιτροπή Επίβλεψης Πτυχιακής Εργασίας

Επιβλέπων Καθηγητής:

ΔΑΣΥΓΕΝΗΣ ΜΗΝΑΣ

Επίκουρος Καθηγητής, Παν. Δυτικής  
Μακεδονίας

Συν-Επιβλέπων Καθηγητής:

ΑΘΑΝΑΙΟΣ ΚΑΚΑΡΟΥΝΤΑΣ

Αναπληρωτής Καθηγητής, Πανεπιστήμιο  
Θεσσαλίας

Συν-Επιβλέπων Καθηγητής:

ΑΛΕΠΗΣ ΕΥΘΥΜΙΟΣ

Αναπληρωτής Καθηγητής, Πανεπιστήμιο  
Πειραιώς

Πάτρα, Ιούλιος 2023

## Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω την οικογένειά μου για την υποστήριξη της καθ' όλη την διάρκεια των σπουδών μου στο Ελληνικό Ανοικτό Πανεπιστήμιο και τους συμφοιτητές μου, με τους οποίους μοιραστήκαμε πολλές χαρούμενες στιγμές. Ιδιαίτερα, θα ήθελα να ευχαριστήσω την κοπέλα μου που συνέβαλε στην ολοκλήρωση της διπλωματικής μου εργασίας.

Επίσης, ευχαριστώ θερμά όλο το εκπαιδευτικό προσωπικό που μου προσέφερε όλες τις γνώσεις και τα απαραίτητα εφόδια για τη μελλοντική μου επαγγελματική πορεία μου ως προγραμματιστής και μηχανικός υπολογιστών.

Τέλος, θα ήθελα να ευχαριστήσω τον καθηγητή και επιβλέποντα της πτυχιακής μου εργασίας, Δρ. Μηνά Δασυγένη, για τη συνεργασία μας και για την άμεση και πολύτιμη βοήθεια που μου παρείχε κατά τη διάρκεια των σπουδών μου και στην ολοκλήρωση της πτυχιακής μου εργασίας.

## Περίληψη

Η συγκεκριμένη πτυχιακή εργασία αφορά την κατασκευή μιας ηλεκτρονικής πλατφόρμας που θα φιλοξενεί χρήστες σε πραγματικό χρόνο. Οι χρήστες θα έχουν την δυνατότητα να παίζουν παιχνίδια με κάρτες, είτε μιας κανονικής τράπουλας, είτε μία ειδικής τράπουλας. Σε κάθε τραπέζι θα υπάρχει ο υπεύθυνος παιχνιδιού που θα έχει τον κύριο έλεγχο του τραπεζιού και θα θέτει τους κανόνες του παιχνιδιού.

Αυτή η πλατφόρμα έχει ως κύριο στόχο να δώσει την ευκαιρία σε χρήστες ώστε να μπορούν να παίζουν τα αγαπημένα τους παιχνίδια με κάρτες, στο διαδίκτυο παρέα με φίλους. Στα ηλεκτρονικά τραπέζια θα μπορούν να παίζουν διάφορα παιχνίδια και να συνομιλήσουν με τους άλλους χρήστες, όπως και θα γινόταν στην πραγματικότητα.

Αρχικά, θα αναλυθούν κάποια παιχνίδια με κάρτες, θα καταγραφούν οι λειτουργικότητες τους ώστε να καταγραφούν οι ενέργειες τους που γίνονται σε παιχνίδια με κάρτες. Έτσι μέσω αυτής της ανάλυσης και καταγραφής, η πλατφόρμα θα έχει την δυνατότητα να εξυπηρετήσει ένα μεγάλο αριθμό παιχνιδιών με κάρτες.

Η πλατφόρμα θα προσφέρει την ευκαιρία σε ανθρώπους που θέλουν να παίζουν παιχνίδια με κάρτες και δεν έχουν την δυνατότητα να το πράξουν σε πραγματικό χρόνο πιθανώς λόγω κάποιας απόστασης. Επίσης θα μπορούν να παίζουν παιχνίδια που δεν υπάρχουν στο διαδίκτυο ή και παιχνίδια που υπάρχουν αλλά με τροποποιήσεις καθώς οι κανόνες των παιχνιδιών σε αυτή την πλατφόρμα θα ορίζονται από τον υπεύθυνο του παιχνιδιού.

Η πλατφόρμα θα αναπτυχθεί σε μορφή ιστοσελίδας, ώστε οι χρήστες να μπορούν να την χρησιμοποιήσουν από όποια συσκευή επιθυμούν. Η βασική γλώσσα ανάπτυξης θα είναι η JavaScript με σύγχρονες βιβλιοθήκες όπου θα χρησιμοποιηθεί για το backend η Nest.js και για frontend η Vue.js.

### Λέξεις – Κλειδιά

Απομακρυσμένοι παίχτες, Παιχνίδια με κάρτες, Επικοινωνία σε πραγματικό χρόνο, Ηλεκτρονικό τραπέζι, Επιτραπέζια παιχνίδια

## **Abstract**

The present dissertation is about the development of a digital platform that hosts users in real time. Its users will play card games and they can choose between a typical or a specialising card pack. On each game, a user can be the supervisor of the game who sets the rules of the game and oversees the game.

The aim of the digital platform is to offer users the opportunity to play their favorite card games online with friends while they are chatting to each other, as if they are meeting in an actual place.

Firstly, card games will be reviewed to identify their functions, rules and outcomes. This review will ensure that the digital platform supports a wide array of card games.

The platform enables people to play card games with friends by not having to all commute to the same place. It also let the users play card games that are unavailable or they can modify existing card games as the supervisor user can set rules.

The platform will be developed in a webpage so that the users can use it from any device. The platform will be written in JavaScript using the libraries Nest.js and Vue.js for backend and frontend respectively.

## **Keywords**

Online Players, Card Games, Real time communication, Online table, Board Games

## Περιεχόμενα

Ευχαριστίες .....	iv
Περίληψη.....	v
Abstract .....	vi
Περιεχόμενα .....	vii
Κατάλογος Εικόνων / Σχημάτων .....	x
Κατάλογος Πινάκων .....	xii
Συντομογραφίες & Ακρωνύμια.....	xiii
1. Εισαγωγή.....	1
1.1 Περιγραφή Σημερινής Κατάστασης .....	1
1.1.1 Η Βιομηχανία της τεχνολογίας .....	2
1.1.2 Παιχνίδια .....	2
1.2 Ιδέα και σκοπός υλοποίησης .....	3
1.3 Παρόμοια Συστήματα.....	4
1.3.1 dod.gr .....	4
1.3.2 Game Structor .....	5
1.4 Σύνοψη διπλωματικής εργασίας.....	7
2. Θεωρητικό υπόβαθρο.....	8
2.1 Διαδίκτυο .....	8
2.2 Ιστοσελίδες.....	8
2.3 Γλώσσες προγραμματισμού front-end .....	8
2.3.1 HTML .....	9
2.3.2 CSS.....	10
2.3.3 Tailwind CSS .....	10
2.3.4 JavaScript .....	10
2.3.5 TypeScript .....	11
2.3.6 Vue .....	12
2.4 Γλώσσες προγραμματισμού back-end .....	12
2.4.1 Node.js .....	13
2.4.2 Nest.js.....	13
2.5 Βάση Δεδομένων.....	14
2.5.1 MySQL.....	14
2.5.2 Type ORM .....	15
2.6 WebSocket .....	15
2.7 Λογισμικά εργαλεία .....	16
2.7.1 Visual Studio Code .....	16
2.7.2 Postman .....	16
2.7.3 Figma.....	16
2.7.4 Send Grid .....	17
2.7.5 Git.....	17
2.8 Σύνοψη κεφαλαίου .....	17
3. Σχεδιασμός διαδικτυακής πλατφόρμας.....	18
3.1 Κατηγορίες χρηστών .....	18
3.2 Ανάλυση πλατφόρμας .....	18

3.3 Ανάλυση παιχνιδιών με κάρτες.....	19
3.3.1 Blackjack.....	19
3.3.2 Μπιρίμπα.....	20
3.3.3 Πόκερ.....	23
3.3.4 Υπο.....	24
3.3.5 Saboteur.....	26
3.4 Λειτουργικές απαιτήσεις πλατφόρμας.....	28
3.4.1 Λειτουργικές απαιτήσεις ηλεκτρονικού τραπέζιου.....	29
3.4.2 Λειτουργικές απαιτήσεις Game Master.....	30
3.4.3 Λειτουργικές απαιτήσεις Game Designer.....	30
3.5 Προδιαγραφές πλατφόρμας.....	31
3.6 Περιπτώσεις χρήσης.....	32
3.6.1 Εγγραφή του χρήστη στην διαδικτυακή πλατφόρμα.....	32
3.6.2 Είσοδος του χρήστη στην διαδικτυακή πλατφόρμα.....	33
3.6.3 Αποσύνδεση του χρήστη από την διαδικτυακή πλατφόρμα.....	34
3.6.4 Επαναφορά κωδικού πρόσβασης.....	35
3.6.5 Επεξεργασία προσωπικών δεδομένων του χρήστη.....	36
3.6.6 Δημιουργία κάρτας.....	36
3.6.7 Δημιουργία τράπουλας.....	37
3.6.8 Δημιουργία παιχνιδιού.....	37
3.6.9 Δημιουργία ηλεκτρονικού τραπέζιου.....	39
3.6.10 Είσοδος στο ηλεκτρονικό τραπέζι.....	39
3.6.11 Προβολή και επεξεργασία όλων των χρηστών.....	40
3.7 Σχεδιασμός και ανάλυση βάσης δεδομένων.....	40
3.7.1 Πίνακας users.....	41
3.7.2 Πίνακας cards.....	45
3.7.3 Πίνακας decks.....	47
3.7.4 Πίνακας cards_deck.....	50
3.7.5 Πίνακας games.....	50
3.7.6 Πίνακας roles.....	54
3.7.7 Πίνακας status.....	56
3.7.8 Πίνακας teams.....	58
3.7.9 Πίνακας hand_start_cards.....	59
3.7.10 Πίνακας decks_games.....	62
3.7.11 Πίνακας tables.....	63
3.7.12 Πίνακας table_users.....	66
3.7.13 Πίνακας table_decks.....	70
3.7.14 Πίνακας table_cards.....	72
3.7.15 Πίνακας ranks.....	75
3.8 Ασφάλεια συστήματος.....	77
3.9 Σύνοψη κεφαλαίου.....	78
4. Λειτουργίες και διεπαφή χρήστη.....	79
4.1 Αρχική Σελίδα.....	79
4.2 Εγγραφή χρήστη.....	80
4.3 Είσοδος χρήστη.....	81
4.4 Επανεκδοση κωδικού πρόσβασης.....	86



4.5 Δημιουργία κάρτας.....	87
4.6 Δημιουργία τράπουλας.....	89
4.7 Δημιουργία παιγνιδιού .....	90
4.8 Δημιουργία ηλεκτρονικού τραπέζιού.....	93
4.9 Προθάλαμος χρήστη .....	94
4.10 Ηλεκτρονικό τραπέζι.....	96
4.10.1 Μενού Game Master .....	98
4.10.2 Ηλεκτρονικό τραπέζι σε κατάσταση εξέλιξης .....	100
4.10.3 Μενού παίχτη .....	102
4.11 Ταμπλό χρήστη .....	104
4.12 Ταμπλό διαχειριστή.....	107
4.13 Web Sockets Events .....	108
5. Επίλογος.....	113
5.1 Ανακεφαλαίωση.....	113
5.2 Μετρικά του συστήματος.....	114
5.2.1 Μετρικά κώδικα front-end .....	114
5.2.2 Μετρικά κώδικα back-end .....	115
5.3 Ανάλυση SWOT.....	115
5.3.1 Δυνατά σημεία .....	116
5.3.2 Αδυναμίες.....	116
5.3.3 Ευκαιρίες.....	116
5.3.4 Απειλές.....	117
5.4 Πλάνο ελέγχου ορθής λειτουργίας.....	117
5.4.1 Περιβάλλον παραγωγής .....	117
5.4.2 Δοκιμή.....	118
5.5 Μελλοντικές επεκτάσεις .....	119
5.6 Συμπεράσματα .....	120
Παράρτημα.....	121
Βιβλιογραφία.....	123

## Κατάλογος Εικόνων / Σχημάτων

Εικόνα 1 - Το ηλεκτρονικό τραπέζι της Μπριμίπας από το dod.gr.....	5
Εικόνα 2 - Δημιουργία παιχνιδιού στο Game Structor .....	6
Εικόνα 3 - Στατιστική μελέτη ως προς την χρήση των γλωσσών προγραμματισμού από το Stack Overflow [16] .....	11
Εικόνα 4 - Στατιστική μελέτη ως προς την χρήση των βάσεων δεδομένων .....	14
Εικόνα 5 - Πίνακας MySQL .....	15
Εικόνα 6 - Παιχνίδι Blackjack .....	20
Εικόνα 7 - Τα χαρτιά που έχει η τράπουλα της μπριμίπας .....	21
Εικόνα 8 - Παιχνίδι Πόκερ .....	23
Εικόνα 9 - Κάρτες από την τράπουλα του Uno .....	24
Εικόνα 10 - Κάρτες από την τράπουλα του Saboteur .....	26
Εικόνα 11 - Διάγραμμα σχέσεων οντοτήτων (ER Diagram) βάσης δεδομένων .....	41
Εικόνα 12 - Πίνακας users .....	41
Εικόνα 13 - Πίνακας cards .....	45
Εικόνα 14 - Πίνακας decks .....	47
Εικόνα 15 - Πίνακας cards_deck .....	50
Εικόνα 16 - Πίνακας games .....	51
Εικόνα 17 - Πίνακας roles .....	54
Εικόνα 18 - Πίνακας status .....	56
Εικόνα 19 - Πίνακας teams .....	58
Εικόνα 20 - Πίνακας hand_start_cards .....	60
Εικόνα 21 - Πίνακας decks_game .....	62
Εικόνα 22 - Πίνακας tables .....	63
Εικόνα 23 - Πίνακας table_users .....	66
Εικόνα 24 - Πίνακας table_decks .....	70
Εικόνα 25 - Πίνακας table_cards .....	72
Εικόνα 26 - Πίνακας ranks .....	75
Εικόνα 27 – Στιγμιότυπο από την σελίδα Home page .....	79
Εικόνα 28 - Στιγμιότυπο από την σελίδα Register .....	80
Εικόνα 29 - Στιγμιότυπο από την σελίδα Register με μηνύματα αποτυχίας .....	80
Εικόνα 30 - Στιγμιότυπο από το email επιβεβαίωσης ηλεκτρονικού ταχυδρομείου .....	81
Εικόνα 31 - Στιγμιότυπο από την σελίδα Login .....	82
Εικόνα 32 - Στιγμιότυπο από την σελίδα Forgot Password .....	86
Εικόνα 33 - Στιγμιότυπο από το email επιβεβαίωσης αλλαγή κωδικού πρόσβασης .....	86
Εικόνα 34 - Στιγμιότυπο από την σελίδα Reset Password .....	87
Εικόνα 35 - Στιγμιότυπο από το ανέβασμα μίας κάρτας στην σελίδα Create Card .....	88
Εικόνα 36 - Στιγμιότυπο από το ανέβασμα πολλών καρτών στην σελίδα Create Card .....	88
Εικόνα 37 - Στιγμιότυπο από την σελίδα Create Deck .....	89
Εικόνα 38 - Φόρμα δημιουργίας παιχνιδιού - Βήμα 1 .....	91
Εικόνα 39 - Φόρμα δημιουργίας παιχνιδιού - Βήμα 2 .....	92
Εικόνα 40 - Φόρμα δημιουργίας παιχνιδιού - Βήμα 3 .....	93
Εικόνα 41 - Φόρμα δημιουργίας ηλεκτρονικού τραπεζιού .....	94
Εικόνα 42 - Φόρμα εισαγωγής ψευδώνυμου για τον ανώνυμο χρήστη .....	95

Εικόνα 43 - Προθάλαμος χρήστη .....	96
Εικόνα 44 - Ηλεκτρονικό τραπέζι σε κατάσταση αναμονής .....	96
Εικόνα 45 - Γραπτή συνομιλία του ηλεκτρονικού τραπεζιού .....	97
Εικόνα 46 - Αναδύομενο παράθυρο των πληροφοριών του ηλεκτρονικού τραπεζιού .....	97
Εικόνα 47 - Μενού εργαλείων του Game Master .....	98
Εικόνα 48 - Ρυθμίσεις χρηστών του ηλεκτρονικού τραπεζιού .....	100
Εικόνα 49 - Ρυθμίσεις χρηστών του πίνακα βαθμολογίας του ηλεκτρονικού τραπεζιού .....	100
Εικόνα 50 - Blackjack παιχνίδι σε ηλεκτρονικό τραπέζι .....	101
Εικόνα 51 - Saboteur παιχνίδι σε ηλεκτρονικό τραπέζι .....	102
Εικόνα 52 - Μενού εργαλείων παίχτη .....	102
Εικόνα 53 - Μενού εργαλείων παίχτη όταν παίζει .....	103
Εικόνα 54 - Στιγμιότυπο από την σελίδα Dashboard ενός χρήστη .....	104
Εικόνα 55 - Στιγμιότυπο από την σελίδα Tables ενός χρήστη .....	104
Εικόνα 56 - Στιγμιότυπο από την σελίδα προβολής παιχνιδιών του χρήστη .....	105
Εικόνα 57 - Στιγμιότυπο από την επεξεργασία παιχνιδιού από το ταμπλό του χρήστη .....	105
Εικόνα 58 - Στιγμιότυπο από την σελίδα προβολής τραπουλών του χρήστη .....	106
Εικόνα 59 - Στιγμιότυπο από την σελίδα προβολής καρτών του χρήστη .....	106
Εικόνα 60 - Στιγμιότυπο από την σελίδα Settings .....	107
Εικόνα 61 - Στιγμιότυπο από την αλλαγή του κωδικού πρόσβασης του χρήστη .....	107
Εικόνα 62 - Στιγμιότυπο από την σελίδα προβολή χρηστών του διαχειριστή .....	108
Εικόνα 63 - Στιγμιότυπο από την επεξεργασία ενός χρήστη από την σελίδα του διαχειριστή .....	108
Εικόνα 64 - Στιγμιότυπο πριν από την επανασύνδεση του παίχτη στο παιχνίδι στην σελίδα "Προθάλαμος χρήστη" .....	111
Εικόνα 65 - Στιγμιότυπο από την κονσόλα του διακομιστή για τα Web Socket Events .....	112
Εικόνα 66 - Στιγμιότυπο από την κονσόλα του backend λογισμικού με μήνυμα επιτυχής εισαγωγής ψεύτικων δεδομένων στην βάση δεδομένων .....	122

## **Κατάλογος Πινάκων**

Πίνακας 1 - Web Socket Events .....	110
Πίνακας 2 - Μετρικά κώδικα frontend εφαρμογής.....	114
Πίνακας 3 - Μετρικά μεταγλωττισμένου κώδικα frontend εφαρμογής .....	115
Πίνακας 4 - Μετρικά κώδικα backend εφαρμογής .....	115
Πίνακας 5 - Μετρικά μεταγλωττισμένου κώδικα backend εφαρμογής.....	115

## Συνομογραφίες & Ακρωνύμια

Ακολουθούν κάποια παραδείγματα:

ΕΑΠ	Ελληνικό Ανοικτό Πανεπιστήμιο
ΠΕ	Πτυχιακή Εργασία
GM	Game Master
ΠΜΚ	Παιχνίδια με κάρτες
ΗΤ	Ηλεκτρονικό τραπέζι
GD	Game Designer

## 1. Εισαγωγή

Η τεχνολογία κατά τον 21<sup>ο</sup> αιώνα έχει σημειώσει ραγδαίες εξελίξεις με καθαριστικό σημείο την περίοδο της πανδημίας του κορονοϊού [1]. Πριν από την πανδημία, η κοινωνία παρουσίαζε τάσεις ώστε να ψηφιοποιήσει [2] τον κόσμο κάτι που έμοιαζε αδύνατο τότε, όχι προς το ότι η τεχνολογία δεν ήταν έτοιμη, αλλά το ότι η ανθρωπότητα δεν ήταν έτοιμη για μια τόσο μεγάλη αλλαγή. Μια τέτοια αλλαγή χρειάζεται χρόνο για να αφομοιωθεί στον κόσμο αφού δεν είναι όλες οι γενιές άνετες με την χρήση της τεχνολογίας. Η πανδημία ανάγκασε τις κοινωνίες να προσαρμοστούν άμεσα στο νέο μοντέλο του ψηφιοποιημένου κόσμου, όπου ο άνθρωπος θα μπορούσε κάνει τις περισσότερες ενέργειες από το σπίτι χωρίς να είναι εκτεθειμένος στους κινδύνους της πανδημίας [3]. Ενέργειες όπως τα ηλεκτρονικά μαθήματα, οι ηλεκτρονικές αγορές, οι πληρωμές λογαριασμών, η κοινωνικοποίηση και η ψυχαγωγία. Με αφορμή λοιπόν την πανδημία και ότι ο κόσμος δεν είχε πολλές επιλογές τότε για να εκπληρώσει κάποιες βασικές ανάγκες όπως αυτές που αναφέρονται πιο πάνω, η παρούσα πτυχιακή εργασία (ΠΕ) θα έχει ως στόχο την δημιουργία μιας ψηφιακής ψυχαγωγικής πλατφόρμας που θα υποδέχεται απομακρυσμένους χρήστες σε πραγματικό χρόνο και θα έχουν την δυνατότητα να παίζουν παιχνίδια με κάρτες (ΠΜΚ) χωρίς να επηρεάζονται από την απόσταση. Η παρούσα εργασία έχει ως στόχο να αναπαραστήσει ένα ηλεκτρονικό τραπέζι (HT) να μοιάζει με πραγματικό τραπέζι για κάρτες, όπου οι χρήστες θα μπορούν να παίξουν διάφορα παιχνίδια. Οι κανόνες του κάθε παιχνιδιού θα ορίζονται από το υπεύθυνο του τραπέζιου και όχι από την πλατφόρμα. Στόχος είναι η αναπαραστάση μιας κλασσικής ψυχαγωγικής βραδιάς που μαζεύεται μία παρέα σε ένα σπίτι για να παίξουν παιχνίδια με κάρτες.

### 1.1 Περιγραφή Σημερινής Κατάστασης

Την σήμερα ημέρα η τεχνολογία είναι απαραίτητη στην ζωή μας τόσο που αρκετοί άνθρωποι δεν μπορούν να ζήσουν χωρίς αυτή. Μια τόσο μεγάλη αλλαγή στην ζωή μας έχει φέρει βέβαια διάφορες αντιδράσεις στον κόσμο, καθώς σε αρκετούς τομείς η τεχνολογία αντικατέστησε τον άνθρωπό με μηχανές [4] και έτσι μειώθηκαν οι δουλειές σε κάποιες βιομηχανίες. Υπάρχει όμως και η αντίθετη όψη του νομίσματος που έχει την άποψη ότι η τεχνολογία έχει εξελίξει τον άνθρωπο, και πως με την κατάλληλη χρήση της τεχνολογίας αναπτυχθήκανε αρκετές εταιρίες με αποτέλεσμα να έχουν δώσει θέσεις εργασίας [5] σε

πολλούς ανθρώπους. Είναι φανερό πως η τεχνολογία έχει πράγματι εξελίξει τον άνθρωπο κι' αυτό έχει ως αποτέλεσμα να έχουν μεγαλώσει και οι απαιτήσεις της ζωής του. Πλέον πολλές υποχρεώσεις του ανθρώπου έχουν αντικατασταθεί από μηχανές και αφού αυτό κάνει την ζωή του ευκολότερη, νιώθει συνεχώς την ανάγκη να εξελίσσεται τεχνολογικά .

### **1.1.1 Η Βιομηχανία της τεχνολογίας**

Οι τεχνολογικές εταιρίες γνωρίζοντας πως η κοινωνία έχει συνεχώς την ανάγκη για εξέλιξη, έχουν ως κύριο μέλημα τους να απλοποιήσουν την ζωή ενός καθημερινού ανθρώπου και να του προσφέρουν μεγαλύτερη ψυχαγωγία κατά την διάρκεια της μέρας του. Συνεχώς δημιουργούν εφαρμογές σε διαφορετικές εργασιακές βιομηχανίες ώστε να κάνουν και πιο εύκολη την δουλειά του. Επίσης δημιουργούν εφαρμογές που έχουν να κάνουν με κοινωνικές σχέσεις ώστε οι άνθρωποι να έχουν την δυνατότητα επικοινωνίας με γνωστούς, να γνωρίζουν νέο κόσμο και άλλες πολλές εφαρμογές που έχει ανάγκη ο μέσος άνθρωπος σήμερα.

Μία βιομηχανία που παρατηρούμε να έχει μεγάλη ανάπτυξη τα τελευταία χρόνια είναι και αυτή των παιχνιδιών. Αρκετοί άνθρωποι ανεξαρτήτου ηλικίας αφιερώνουν μεγάλο χρονικό διάστημα της καθημερινότητας τους παίζοντας παιχνίδια [6].

### **1.1.2 Παιχνίδια**

Πάντα τα παιχνίδια πρόσφεραν μια ευχάριστη νότα στην ζωή του ανθρώπου. Τα παιχνίδια πέραν από την χαλάρωση που προσφέρουν σε μία πιεστική καθημερινότητα, προσφέρουν και κοινωνικοποίηση με άλλους ανθρώπους, γνώσεις και καινούργιες δεξιότητες. Είναι συχνό φαινόμενο στις παρέες να οργανώνονται βραδιές με παιχνίδια, είτε ηλεκτρονικά παιχνίδια, είτε επιτραπέζια παιχνίδια, ακόμη ΠΜΚ. Οι τεχνολογικές εταιρίες δίνουν μεγάλο βάρος στην βιομηχανία των παιχνιδιών και τώρα υπάρχει ένας μεγάλος αριθμός εταιριών που ασχολούνται αποκλειστικά με αυτά.

Υπάρχουν αρκετά επιτραπέζια παιχνίδια και ΠΜΚ που είναι διαχρονικά, όπως η Μονοπολύ, το πόκερ, το Blackjack, η μπιρίμπα και άλλα πολλά. Αρκετές εταιρίες έχουν δημιουργήσει την ηλεκτρονική μορφή αυτών των γνωστών παιχνιδιών με αποτέλεσμα να έχουν μεγάλη απήχηση στο κοινό, καθώς τα ΠΜΚ είναι συνήθως γρήγορα στο να τα μάθεις, ελάχιστος χρόνος προς την διάρκεια του παιχνιδιού και επίσης δεν χρειάζεται μόνο

δεξιότητες για να κριθεί νικητής κάποιος αλλά και τύχη. Από πάντα, τα καζίνο που παίζονται ΠΜΚ σε τυχερά παιχνίδια είχανε υψηλή απήχηση στο κοινό, και αρκετές βιομηχανίες το εκμεταλλεύτηκαν αυτό και κατασκεύασαν ανάλογες πλατφόρμες [7].

## 1.2 Ιδέα και σκοπός υλοποίησης

Έχουν δημιουργηθεί αρκετές εφαρμογές στο παρελθόν με πλατφόρμες που είναι αποκλειστικά για ένα παιχνίδι και με κανόνες που ορίζει ο δημιουργός του παιχνιδιού. Υπάρχουν επίσης πλατφόρμες όπου φιλοξενούν αρκετά παιχνίδια σε μία πλατφόρμα αλλά και πάλι οι κανόνες των παιχνιδιών είναι αυτοί που ορίζουν οι δημιουργοί τους. Αυτή η πλατφόρμα θα αφορά ΠΜΚ και θα ξεχωρίζει για το ότι οι κανόνες δεν θα είναι ορισμένοι από το πρόγραμμα – δημιουργό της πλατφόρμας αλλά από τον υπεύθυνο του τραπεζιού όπου θα τον ορίσουμε ως Game Master (GM). Όπως και στην πραγματική ζωή, υπάρχει κάποιος GM στην παρέα ο οποίος ενημερώνει τους κανόνες του παιχνιδιού και στην συνέχεια οι παίκτες αρχίζουν να παίζουν. Αν κάποιος παίχτης κάνει κάτι λάθος που είναι εκτός των κανόνων του παιχνιδιού τότε ο GM τον ενημερώνει για το λάθος του και τον διορθώνει.

Έτσι και στο ΗΤ την πλατφόρμας, ο GM θα έχει τον απόλυτο έλεγχο του παιχνιδιού, θα ενημερώνει για του κανόνες του παιχνιδιού και θα αρχίζει το παιχνίδι. Σε περίπτωση λάθους θα διακόπτει το παιχνίδι, θα ενημερώνει το παίχτη για το λάθος του και στην συνέχεια θα συνεχίζει το παιχνίδι από το σημείο που διακόπηκε.

Θα δίνει την δυνατότητα στους χρήστες να παίζουν οποιοδήποτε παιχνίδι επιθυμούν με κάρτες, παραλλαγές των αγαπημένων τους παιχνιδιών ή και ακόμη να δημιουργήσουν δικό τους παιχνίδι.

Ο GM θα προσαρμόζει το ΗΤ μέσα από ρυθμίσεις που θα επιλέγει στον προθάλαμο του παιχνιδιού.



## 1.3 Παρόμοια Συστήματα

Μετά από έρευνα στο διαδίκτυο, παρατηρήθηκε ότι υπάρχουν αρκετές πλατφόρμες που προσφέρουν ΠΜΚ αλλά σε πολύ λίγες μπορούν οι χρήστες να δημιουργήσουν το δικό τους παιχνίδι. Σε αυτή την ΠΕ θα εξεταστεί μία πλατφόρμα που προσφέρει ΠΜΚ μόνο για να παίζουν χρήστες και άλλη μία που δίνει την ευκαιρία στο χρήστη να δημιουργήσει το δικό του παιχνίδι και να το παίξει. Έτσι θα προκύψουν συμπεράσματα για το πως ο χρήστης δημιουργεί ένα παιχνίδι και τις δυνατότητες που θα πρέπει να έχει το ΗΛ.

### 1.3.1 dod.gr

Η πλατφόρμα dod.gr [8] παρέχει μια μεγάλη συλλογή από παιχνίδια επιτραπέζια ή με κάρτες όπου στην αρχική οθόνη ο χρήστης καταχωρεί τα στοιχεία του λογαριασμού του για να εισέλθει ή να εγγραφεί στην πλατφόρμα. Αν επιθυμεί μπορεί να εισέλθει ως επισκέπτης αλλά θα έχει την δυνατότητα να συμμετέχει σε παιχνίδια μόνο με ανώνυμους παίκτες.

Όταν συνδεθεί στην κεντρική οθόνη, έχει πρόσβαση σε όλα στα διαθέσιμα παιχνίδια που παρέχει η πλατφόρμα. Με την επιλογή ενός παιχνιδιού, η πλατφόρμα βάζει τον χρήστη στο παιχνίδι και του δίνει την δυνατότητα να δημιουργήσει ένα τραπέζι ή να καθίσει σε ένα τραπέζι που έχει δημιουργηθεί από κάποιο άλλο.

Αν θα δημιουργήσει τραπέζι, τότε του εμφανίζεται μία οθόνη για την επιλογή των ρυθμίσεων του παιχνιδιού και όταν τις ολοκληρώσει, το τραπέζι δημοσιεύεται στα διαθέσιμα τραπέζια στους υπόλοιπους χρήστες της πλατφόρμας. Στις ρυθμίσεις μπορεί να επιλέξει αν θα παίξει μόνο με φίλους, αν θα έχει κωδικό το τραπέζι και τον χρόνο αναμονής για κάθε γύρο του παίχτη.

Αφού εισέλθει στο τραπέζι ο χρήστης περιμένει να καθίσουν όλοι οι παίκτες που απαιτούνται για να ξεκινήσει το παιχνίδι. Στο τραπέζι υπάρχει και η δυνατότητα οι παίκτες να μιλήσουν μεταξύ τους μέσω γραπτών μηνυμάτων που είναι ορατά σε όλο το τραπέζι.



Εικόνα 1 - Το ηλεκτρονικό τραπέζι της Μπιρίμπας από το dod.gr

Με την ολοκλήρωση του παιχνιδιού καταγράφεται μια βαθμολογία για τον παίχτη. Ο παίχτης έχει δύο βαθμολογίες: μία για το πόσο καλός είναι στα παιχνίδια με τις νίκες που έχει κάνει και μία άλλη για το αν είναι έμπιστος παίχτης στην πλατφόρμα.

### 1.3.2 Game Structor

Η πλατφόρμα Game Structor προσφέρει στο χρήστη την δυνατότητα να δημιουργήσει το δικό του ΠΜΚ και στην συνέχεια να παίξει το παιχνίδι με άλλους χρήστες.

Αρχικά ο χρήστης πρέπει να δημιουργήσει ένα λογαριασμό για να έχει την δυνατότητα να δημιουργήσει ένα παιχνίδι. Αφού εισέλθει με τα στοιχεία του εμφανίζει μια λίστα με επιλογές ώστε να δημιουργήσει τις κάρτες του, τις τράπουλες, το τραπέζι του, τα ζάρια και διάφορες άλλες επιλογές που φαίνονται στην παρακάτω εικόνα.

The screenshot shows the Game Structor website interface. At the top, there is a navigation bar with links for 'Play Game', 'Search', 'My Stuff', and 'Account'. Below this, the user profile for 'KUDOS' is displayed with a reputation of 0. The main content area is divided into two sections: 'Games' and 'Components'. The 'Games' section has a table with columns for 'Created', 'Times Played', 'Average Rating', and 'Total Votes', all showing 0. The 'Components' section has a table with columns for 'Created', 'Page Views', and 'Community Copies', all showing 0. Each row in the 'Components' table has a 'Create' button.

Games				
Created	Times Played	Average Rating	Total Votes	
0	0	0	0	Create

Components				
	Created	Page Views	Community Copies	
Boards	0	0	0	Create
Tiles	0	0	0	Create
Tokens	0	0	0	Create
TileBags	0	0	0	Create
Decks	0	0	0	Create
Cards	0	0	0	Create
Dice	0	0	0	Create
Dice Sides	0	0	0	Create
Rules	0	0	0	Create

Εικόνα 2 - Δημιουργία παιχνιδιού στο Game Structor

Ο χρήστης καλείται να τα δημιουργήσει όλα από την αρχή μόνος του, για παράδειγμα θα πρέπει να ανεβάσει μια-μια τις εικόνες με κάρτες και να τους δώσει ονόματα επιπλέον θα πρέπει να επιλέξει τις κάρτες που έφτιαξε για την δημιουργία μιας τράπουλας.

Εφόσον ολοκληρώσει όλες τις ρυθμίσεις του παιχνιδιού, αποθηκεύονται στα παιχνίδια του και το παιχνίδι που δημιούργησε είναι διαθέσιμο να φορτωθεί σε ένα τραπέζι. Επομένως ο χρήστης δημιουργεί ένα τραπέζι και προσκαλεί χρήστες για να αρχίσει το παιχνίδι.

Στο τραπέζι, υπάρχουν σε ένα μέρος όλες οι τράπουλες, μαζί με τις στοίβες με τα σκάρτα για την κάθε μία και οι χρήστες χωρίς κάποια σειρά μπορούν τα τραβάνε χαρτιά. Το χαρτί που τράβηξε ο παίχτης μπορεί να το βάλει κατευθείαν στο τραπέζι ή στο χέρι του. Τα χαρτιά που είναι στο χέρι του μπορεί να τα κατεβάσει στο τραπέζι είτε ανοιχτά είτε κλειστά ή και να τα τοποθετήσει σε μία τράπουλα. Επίσης, τα χαρτιά που βρίσκονται στο τραπέζι μπορεί ο χρήστης να τα τοποθετήσει οπουδήποτε στο τραπέζι με όποια φορά θέλει και αν το επιθυμεί να τα μαζέψει στην στοίβα που είναι στο χέρι του. Στο τραπέζι υπάρχει

η λίστα με τους παίχτες που είναι συνδεδεμένοι και μπορούν να συνομιλήσουν μεταξύ τους μέσω γραπτών μηνυμάτων.

Εφόσον ολοκληρωθεί το παιχνίδι και το τραπέζι τερματιστεί η πλατφόρμα επιστρέφει τον χρήστη αυτόματα στην αίθουσα εύρεσης παιχνιδιού.

## 1.4 Σύνοψη διπλωματικής εργασίας

Η ΠΕ έχει την εξής δομή:

- **Κεφάλαιο 1:** Γίνεται μια εισαγωγή για την ανάπτυξη της τεχνολογίας τον 21ο αιώνα μετά την πανδημία του κορονοϊού καθώς και για την γενική ιδέα της πλατφόρμας. Αναλύεται η γενική ιδέα της πλατφόρμας όπως και ο σκοπός υλοποίησης της και παρουσιάζεται η διαφορετικότητα της ως προς τις άλλες παρόμοιες πλατφόρμες. Γίνεται παρουσίαση και ανάλυση των άλλων εφαρμογών με στόχο να χρησιμοποιηθούν στην συνέχεια για να επιτευχθεί καλύτερη πλατφόρμα σε σχέση με αυτές.
- **Κεφάλαιο 2:** Καταγράφονται οι τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν στην ΠΕ και αναλύεται το θεωρητικό τους υπόβαθρο.
- **Κεφάλαιο 3:** Γίνεται πλήρης ανάλυση σχετικά με τις λειτουργικές απαιτήσεις του συστήματος και αναλύονται κάποια ΠΜΚ που θα υποστηρίξει η πλατφόρμα. Αυτό γίνεται για να επιλεγεί η κατάλληλη σχεδίαση για την πλατφόρμα και με αυτό θα διευκολυνθεί η ομαλή ανάπτυξη κώδικα της πλατφόρμας ώστε να μπορεί και η πλατφόρμα να εξυπηρετήσει όσα περισσότερα ΠΜΚ μπορεί. Στην συνέχεια αναλύονται και οι περιπτώσεις χρηστών ώστε να προκύψει ο τρόπος υλοποίησης της πλατφόρμας και της βάσης δεδομένων της εφαρμογής.
- **Κεφάλαιο 4:** Θα γίνει ανάλυση του κώδικα που χρησιμοποιήθηκε ως προς την υλοποίηση στο backend και στο frontend. Θα παρουσιαστούν και οι εικόνες από τις διεπαφές της πλατφόρμας.
- **Κεφάλαιο 5:** Στο τελευταίο κεφάλαιο θα γίνει μία μικρή περιγραφή των όσων έγιναν κατά την διάρκεια υλοποίησης της ΠΕ, τα προβλήματα που προέκυψαν καθώς μελλοντικές εκδόσεις που μπορούν προκύψουν σε αυτή την πλατφόρμα. Επιπλέον, θα παρουσιαστεί μια συνοπτική περιγραφή των πλεονεκτημάτων και μειονεκτημάτων της πλατφόρμας.

## 2. Θεωρητικό υπόβαθρο

Σε αυτό το κεφάλαιο θα αναλυθούν οι τεχνολογίες και τα εργαλεία που θα χρησιμοποιηθούν για την κατασκευή της συγκεκριμένης πλατφόρμας ως προς το λόγο και τρόπο της χρήσης τους.

### 2.1 Διαδίκτυο

Το διαδίκτυο [9] είναι ένα παγκόσμιο δίκτυο δισεκατομμυρίων υπολογιστών και άλλων συσκευών που οι άνθρωποι μπορούν να μοιράζονται πληροφορίες και να επικοινωνούν μεταξύ τους χωρίς η απόσταση να τους είναι εμπόδιο. Η σύνδεση στο διαδίκτυο πραγματοποιείται μέσω των τηλεφωνικών γραμμών που παρέχεται από παρόχους υπηρεσιών διαδικτύου. Η μεταφορά της πληροφορίας γίνεται κυρίως με την χρήση του TCP/IP (Transmission Control Protocol/Internet Protocol) πρωτόκολλου.

### 2.2 Ιστοσελίδες

Στον διαδίκτυο υπάρχουν αμέτρητες ιστοσελίδες [10] που οι περισσότερες είναι διαθέσιμες για όλους τους χρήστες, κάποιες είναι ιδιωτικές. Οι ιστοσελίδες περιέχουν κείμενα, φωτογραφίες, βίντεο και αρχεία και συνήθως ανήκουν σε ένα άτομο ή σε μία επιχείρηση.

Η κάθε ιστοσελίδα αποτελείται από μια μοναδική διεύθυνση που ονομάζεται URL (Uniform Resource Locator) και με την χρήση αυτής της διεύθυνσης ο χρήστης μπορεί να την επισκεφτεί μέσω ενός προγράμματος περιήγησης.

Για να κατασκευαστεί μία ιστοσελίδα γίνεται χρήση αρκετών προγραμματιστικών γλωσσών, είτε αφορούν την εμφάνιση της ιστοσελίδας, είτε την επικοινωνίας της με ένα διακομιστή. Μία ιστοσελίδα όμως μπορεί να κατασκευαστεί και μόνο με μία γλώσσα σήμανσης υπερκειμένου όπως είναι η HTML. Μία ιστοσελίδα ωστόσο μπορεί να αποτελείται είτε από μία σελίδα είτε από πάρα πολλές σελίδες.

Υπάρχουν δύο ειδών ιστοσελίδες, οι στατικές και οι δυναμικές. Σε μία στατική ιστοσελίδα ο χρήστης έχει την δυνατότητα μόνο να διαβάσει το κείμενο, τις φωτογραφίες και ότι άλλο έχει, χωρίς να αλλάξει κάτι στην σελίδα. Αντιθέτως, στις δυναμικές μπορούν να αλλάξουν οι πληροφορίες ανάλογα με τις ενέργειες που θα έχει ο χρήστης στην σελίδα.

### 2.3 Γλώσσες προγραμματισμού front-end

Η χρήση των γλωσσών προγραμματισμού front-end αποσκοπούν στην εμφάνιση της πληροφορίας στην ιστοσελίδα. Οι βασική τριάδα γλωσσών που χρησιμοποιούνται είναι η HTML, CSS και JavaScript. Η πρώτη προσφέρει την πληροφορία στον χρήστη, η δεύτερη στο πως θα εμφανιστεί σε διάταξη και η τρίτη αφορά την λειτουργικότητα της ιστοσελίδας. Η CSS σε συνδυασμό με την JavaScript μπορούν να δημιουργήσουν όμορφα γραφικά, κινούμενα σχέδια και να δώσουνε μία ξεχωριστή εμπειρία στο χρήστη.

Με την ανάπτυξη της τεχνολογίας, έχουν βγει αρκετές ολοκληρωμένες βιβλιοθήκες που χρησιμοποιούνται ώστε να γίνεται γενικά πιο εύκολη η δημιουργία μιας ιστοσελίδας αλλά και οι ιστοσελίδες να είναι πιο φιλικές προς τον χρήστη σε διαφορετικές συσκευές πέρα του υπολογιστή. Η JavaScript έχει αρκετές βιβλιοθήκες όπως την Angular, React και Vue, που προσφέρουν εντυπωσιακές λειτουργίες και ταχύτητα στην ιστοσελίδες. Σε αυτή

την πτυχιακή έγινε η επιλογή της βιβλιοθήκης Vue και θα αναλυθούν περισσότερο σε επόμενο κεφάλαιο οι δυνατότητες της.

### 2.3.1 HTML

Η HTML [11] είναι μια γλώσσα σήμανσης υπερκειμένου που χρησιμοποιείται για την κατασκευή ιστοσελίδων. Συγγράφονται κείμενα ανάμεσα σε ετικέτες ώστε το πρόγραμμα περιήγησης να κατανοήσει και να χειριστεί το κείμενο ανάλογα με τις ετικέτες. Υπάρχουν ετικέτες για διαφορετικού τύπου κείμενου. Για παράδειγμα η ετικέτα `<p></p>` είναι για παραγράφους, η `<h1></h1>` για τίτλους, η `<table></table>` για πίνακες, οι `<ul></ul>` `<ol>` `</ol>` για λίστες και άλλα πολλά. Η σύνταξη της ιστοσελίδας έχει πάντα την μορφή:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Document</title>
  </head>
  <body>
    <h1 id="hero_title">Hello World</h1>
  </body>
</html>
```

Το κυρίως κείμενο συγγράφεται ανάμεσα στην ετικέτα του body και ο εξής κώδικας θα εκτυπώσει στην ιστοσελίδα τον τίτλο Hello World.

### 2.3.2 CSS

Η CSS [12] είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται για την εφαρμογή στυλ στα στοιχεία της HTML σελίδας. Σε μια ετικέτα της HTML μπορεί να προστεθεί ένας επιλογέας μοναδικού στοιχείου ή ομαδικού στοιχείου. Στην CSS γράφεται κώδικας για αυτούς του επιλογείς και προσθέτει το στυλ στην ιστοσελίδα. Για παράδειγμα, όταν στην ετικέτα <head> του προηγούμενου παραδείγματος προσθέτει ο εξής κώδικας

```
<style>
  #hero_title {
    color: blue;
  }
</style>
```

ο τίτλος «Hello World» θα έχει χρώμα μπλε. Καλύτερη προγραμματιστική λύση θεωρείται ο κώδικας της CSS να είναι σε διαφορετικά αρχεία για καλύτερη οργάνωση των αρχείων και να γίνεται η κλήση τους από την ετικέτα <head> της HTML για να φορτωθούν έτσι πιο γρήγορα τα αρχεία της CSS.

### 2.3.3 Tailwind CSS

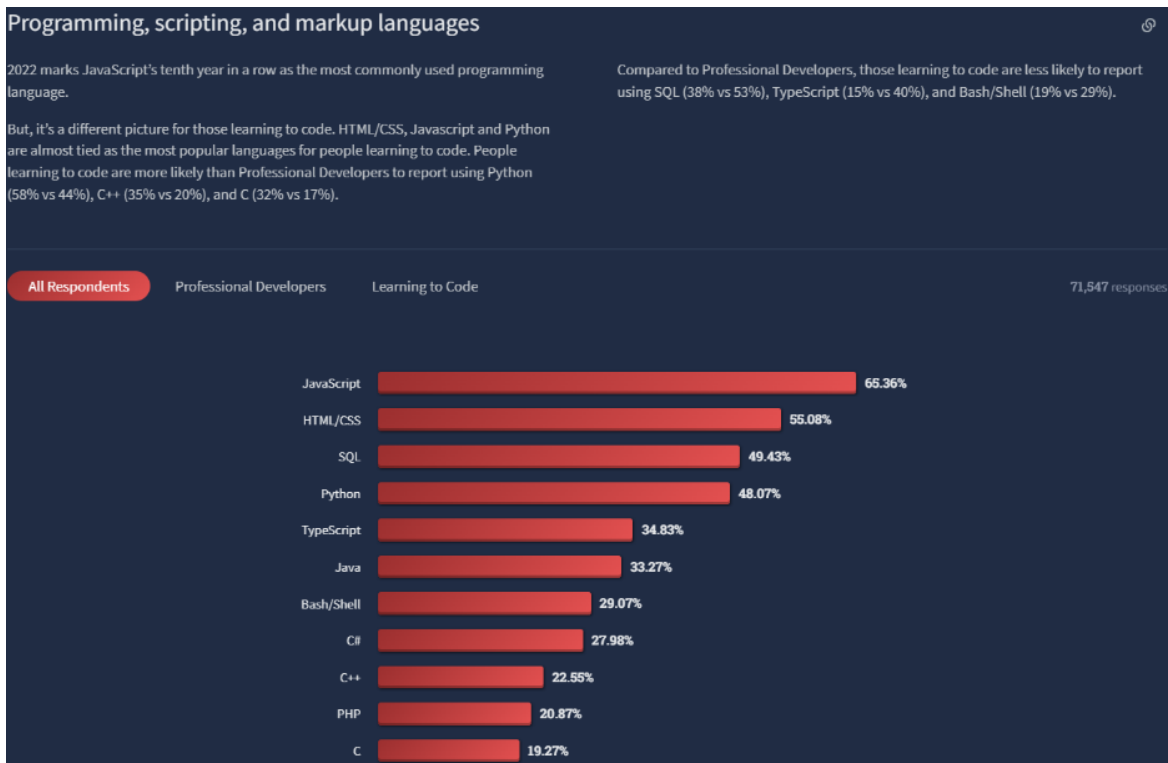
Υπάρχουν αρκετές βιβλιοθήκες στην CSS που λύνουν τα χέρια του προγραμματιστή στο να μην γράφει τεράστια αρχεία κώδικα CSS. Σε αυτή την ΠΕ επιλέχθηκε η Tailwind CSS [13] που έχει υλοποιημένους σχεδόν όλους τους επιλογείς για κανόνες της CSS και μπορούν να χρησιμοποιηθούν κατευθείαν στο HTML αρχείο. Δίνεται η ευελιξία στον προγραμματιστή να χρησιμοποιήσει τους έτοιμους επιλογείς κατευθείαν με την συγγραφή του κειμένου χωρίς να τους υλοποιήσει στα css αρχεία. Στην εγκατάσταση της Tailwind CSS δημιουργείται ένα αρχείο tailwind.config.js, με το οποίο δίνεται η δυνατότητα στον προγραμματιστή να προσθέσει τις γενικές ρυθμίσεις που θα έχει η Tailwind CSS ως προς τα χρώματα της παλέτας του, το μέγεθος των γραμμάτων, το μέγεθος του container, το ύψος των περιθωρίων ανάμεσα στις ετικέτες και άλλα πολλά. Για να προκύψει το “Hello World” σε χρώμα μπλε, με φόντο πράσινο, χρειάζεται ο εξής κώδικας:

```
<h1 class="bg-green-900 text-blue-900">Hello World</h1>
```

### 2.3.4 JavaScript

Υπήρχε η ανάγκη να κατασκευαστεί μια πιο προγραμματιστική γλώσσα για να ενισχύσει το front-end development και να πλαισιώσει τις γλώσσες HTML και CSS. Η JavaScript [14] δεν έχει κάποια σχέση με την Java παρά μόνο ως προς την σύνταξη και το κοινό πρόθεμα. Είναι γλώσσα που είναι ανεκτική σε προγραμματιστικά λάθη για αυτό αν υπάρξει κάποιο πρόβλημα στο κώδικα, θα προβληθεί η σελίδα, έστω και άστοχα.

Η JavaScript είναι η πιο δημοφιλής γλώσσα για το 2022 με βάση τις στατιστικές που δημοσιεύτηκαν από την Stack Overflow.



Εικόνα 3 - Στατιστική μελέτη ως προς την χρήση των γλωσσών προγραμματισμού από το Stack Overflow [15]

Έχει πρόσβαση σε όλο το HTML DOM Document Object και μπορεί να αλληλοεπιδράσει με όλες τα στοιχεία της HTML, ώστε να προσθέσει, να διαγράψει ή να αλλάξει κείμενο, στυλ. Ακόμη μπορεί να κάνει και κάποιες συγκριμένες ενέργειες ανάλογα με τις ανάγκες της ιστοσελίδας, για παράδειγμα με το πάτημα ενός κουμπιού να αρχίζει ένας μετρητής να μετράει και να ενημερώνει την σελίδα για τις αλλαγές του μετρητή που τρέχουν στην JavaScript.

### 2.3.5 TypeScript

Στην JavaScript υπάρχει η δυνατότητα σε μία μεταβλητή που ορίστηκε να είναι σε τύπο αριθμού, στην συνέχεια να τροποποιηθεί και να γίνει σε τύπο γραμματοσειράς. Καθώς μεγαλώνει ένα πρόγραμμα σε όγκο και κώδικα αυτό παύει να είναι πλεονέκτημα γιατί δεν είναι ξεκάθαρος ο τύπος της μεταβλητής και δημιουργεί μπερδεμα στον προγραμματιστή. Επίσης μπορεί να επιφέρει και προβλήματα στο πρόγραμμα. Με βάση αυτή την ιδέα κατασκευάστηκε η TypeScript [16] που είναι συντακτικό υπερσύνολο της JavaScript. Έτσι ο επεξεργαστής κειμένου εντοπίζει τους λάθους τύπους και δεν μεταγλωττίζει τον κώδικα αν δεν διορθωθεί το λάθος προς τον τύπο της μεταβλητής. Ο ορισμός των τύπων γίνεται με τον εξής τρόπο:



```
const car: { type: string, model: string, year: number } = {  
  type: "Toyota",  
  model: "Corolla",  
  year: 2009  
};
```

### 2.3.6 Vue

Η Vue [17] είναι μια βιβλιοθήκη της JavaScript ανοικτού κώδικα που κυκλοφόρησε το 2014 από τον Evan You. Εκμεταλλευόμενος την εργασιακή του εμπειρία στην Google ως προγραμματιστής σε Angular, χρησιμοποίησε τα κομμάτια που του άρεσαν από την Angular και κατασκεύασε μια εντελώς νέα και ελαφριά βιβλιοθήκη. Η Vue στηρίζεται στην λογική ότι η πολυσέλιδη ιστοσελίδα είναι μία σελίδα. Δηλαδή σε κάθε αλλαγή σελίδας, δεν φορτώνει μία νέα σελίδα αλλά ένα νέο component. Επομένως δεν αλληλοεπιδρά άμεσα με το DOM αλλά με ένα εικονικό DOM που τις δίνει ταχύτερη φόρτωση που ήταν στα ζητούμενα του Evan You. Αυτό επιτυγχάνεται με την χρήση της ετικέτας <div> στο βασικό HTML αρχείο και έχει κωδικό όνομα, app. Όλη η Vue τρέχει σε αυτή την ετικέτα.

Το κάθε component της Vue έχει κύκλο ζωής (Lifecycle Hooks [18]) και δίνει τεράστιες δυνατότητες στα χέρια του προγραμματιστή ως προς το χρονικό σημείο που θα χρησιμοποιήσει τις μεταβλητές ή τις κλήσεις των συναρτήσεων του. Για παράδειγμα στην συνάρτηση create ο προγραμματιστής εκτελεί εντολές πριν την δημιουργία του component, η συνάρτηση mounted εκτελεί εντολές αφού δημιουργηθεί το component, και η συνάρτηση unmounted καλείται όταν παύει να ισχύει το component.

Το component χωρίζεται σε 3 μέρη (ετικέτες): <template>, <script> και <style>. Η πρώτη ετικέτα χρησιμοποιείται για την κύρια συγγραφή της HTML. Μπορούν επίσης να χρησιμοποιηθούν όμως και JavaScript μεταβλητές, συνθήκες και επαναλήψεις. Στην δεύτερη ετικέτα είναι αποκλειστικά κώδικας JavaScript με σύνταξη της Vue με την δυνατότητα να γίνει η χρήση της TypeScript. Η τρίτη και τελευταία ετικέτα αφορά την γλώσσα CSS, στην οποία μπορούν να χρησιμοποιηθούν και μεταβλητές από το component. Τα component είναι δομημένα με τέτοιο τρόπο ώστε να μπορούν να λειτουργούν και αυτόνομα και να είναι επαναχρησιμοποιήσιμα μέσα στο πρόγραμμα.

## 2.4 Γλώσσες προγραμματισμού back-end

Οι back-end γλώσσες προγραμματισμού δεν έχουν να κάνουν καθόλου με την εμφάνιση της ιστοσελίδας αντιθέτως είναι ένα μέρος της ιστοσελίδας που δεν φαίνεται καν στο χρήστη. Οι γλώσσες αυτές είναι υπεύθυνες για την οργάνωση και διαχείριση των δεδομένων. Αυτές οι γλώσσες τρέχουν στο διακομιστή, παίρνουν τα δεδομένα από την βάση δεδομένων και τα προωθούν στο front-end ώστε οι πληροφορίες αυτές να εμφανιστούν στον χρήστη. Μπορεί ωστόσο ο χρήστης να στείλει τα δεδομένα πίσω. Με αυτό τον τρόπο οι ιστοσελίδες γίνονται δυναμικές και έτσι η ιστοσελίδα έχει διαφορετικά δεδομένα για τον κάθε χρήστη.

## 2.4.1 Node.js

Η Node.js [19] είναι βιβλιοθήκη ανοιχτού κώδικα που τρέχει στο περιβάλλον της JavaScript. Η Node.js δεν τρέχει στο πρόγραμμα περιήγησης όπως κάνει η απλή JavaScript αλλά τρέχει σε διακομιστή στο διαδίκτυο. Είναι τέλεια για την χρήση σε προγράμματα για ανταλλαγή δεδομένων με την ιστοσελίδα, καθώς καλεί συναρτήσεις ασύγχρονα και η ιστοσελίδα γίνεται δυναμική.

Είναι πολύ γνωστή βιβλιοθήκη και υπάρχουν πάρα πολλά διαθέσιμα πακέτα που μπορούν να εγκατασταθούν σε αυτή ώστε να μειώσει το χρόνο υλοποίησης μίας εφαρμογής. Το συντακτικό της είναι το ίδιο με την JavaScript που είναι πλεονέκτημα για του προγραμματιστές που την γνωρίζουν. Στα βασικά της πλεονεκτήματα προστίθεται η ταχύς μεταφορά δεδομένων γιατί τα αρχεία της φορτώνονται κατευθείαν από την βιβλιοθήκη. Επιπλέον, λόγω της ασύγχρονης αρχιτεκτονικής που κατασκευάστηκε, οι κλήσεις μπορούν να γίνουν ταυτόχρονα χωρίς να περιμένουν να ολοκληρωθεί η μία κλήση για να μπορεί να εκτελέσει η επόμενη κλήση. Αυτό επιτυγχάνεται με την εφαρμογή «Asynchronous in Nature and Event driven».

## 2.4.2 Nest.js

Για να μπορεί να τρέξει η Node.js στην μεριά του διακομιστή χρειάζεται την βιβλιοθήκη της Express.js. Η Nest.js [20] είναι βιβλιοθήκη που είναι υλοποιημένη πάνω στο περιβάλλον της Node.js σε συνδυασμό με TypeScript. Αυτός ο συνδυασμός για δημιουργία της Nest.js δίνει την δυνατότητα στους προγραμματιστές να φτιάξουν μεγάλης κλίμακας εφαρμογές. Επίσης, δίνει την δυνατότητα στους προγραμματιστές, μέσα από εντολές του command line, να δημιουργούν έτοιμα αρχεία κώδικα.

Η αρχειοθέτηση των αρχείων χωρίζεται σε 3 βασικά αρχεία που προσφέρουν καλύτερη οργάνωση στον κώδικα. Αρχικά, οι APIs κλήσεις χωρίζονται σε ενότητες. Σε κάθε ενότητα υπάρχουν πάντα τα αρχεία module, controllers και service. Το αρχείο module αφορά τις βασικές ρυθμίσεις της ενότητας ως προς τα αρχεία που θα χρησιμοποιεί ο προγραμματιστής και με ποιες άλλες ενότητες θα έχει αλληλοεπίδραση. Το αρχείο controller αφορά τις κλήσεις των APIs ως προς την διεύθυνση του κάθε API και τον τύπο της κλήσης του. Η χρήση του middleware γίνεται στο αρχείο του controller με decorators που προσφέρονται από την βιβλιοθήκη, για να ελέγχει την πρόσβαση του χρήστη ανάλογα με τον ρόλο που θα έχει σε κάθε API. Τέλος, στα αρχεία service υπάρχουν όλες οι ενέργειες της συνάρτησης του κάθε API όπου αυτά τα αρχεία καλούνται από τους controller. Πέρα από τα βασικά αρχεία που χωρίζονται σε φακέλους, μπορούν να χρησιμοποιηθούν και άλλα που προσφέρονται από την βιβλιοθήκη. Κάποια από αυτά είναι τα models που περιέχουν αρχεία που έχουν συσχέτιση ως προς το αντικείμενο ή προς το σχήμα ενός πίνακα, άλλα είναι τα dto (Data transfer objects) που χρησιμοποιούνται για να προσδιοριστεί το αντικείμενο που θα λάβει το API, τα guards που πραγματοποιούν ελέγχους πριν να υλοποιηθεί η συνάρτηση του API που κλήθηκε και άλλα πολλά.

## 2.5 Βάση Δεδομένων

Η βάση δεδομένων είναι απαραίτητο εργαλείο για το backend και έχει ρυθμιστεί για εύκολη πρόσβαση, διαχείριση, ενημέρωση και ανάλυση δεδομένων. Στην βάση δεδομένων αποθηκεύονται όλες οι πληροφορίες μίας ιστοσελίδας. Ένα μεγάλο μέρος αυτών των δεδομένων αφορά προσωπικά στοιχεία. Σε μία βάση μπορούν να αποθηκευτούν οποιεσδήποτε πληροφορίες μιας ιστοσελίδας αλλά συνήθως αυτές που αποθηκεύονται αφορούν δεδομένα όπως είναι τα στοιχεία του χρήστη, οι συναλλαγές του, πληροφορίες προϊόντων, ενέργειες που κάνει ο χρήστης και άλλα πολλά.

Υπάρχουν διάφορων τύπου βάσεις δεδομένων και ο κάθε τύπος προσφέρει διαφορετικές δυνατότητες καθώς έχουν διαφορετικές αρχιτεκτονικές. Η επιλογή γίνεται μέσω διάφορων κριτηρίων όπως για παράδειγμα, το εύρος των πληροφοριών, το κόστος, τον τρόπο της εμφάνισης των δεδομένων ακόμη και από πόσους χρήστες θα έχει η ιστοσελίδα.

### 2.5.1 MySQL

Η MySQL [21] είναι η πιο δημοφιλής βάση δεδομένων που υποστηρίζεται από την Oracle. Χρησιμοποιείται από πάρα πολλούς προγραμματιστές καθώς είναι ανοιχτού κώδικα και μπορεί να την χρησιμοποιήσει ο καθένας χωρίς κάποιος κόστος.



Εικόνα 4 - Στατιστική μελέτη ως προς την χρήση των βάσεων δεδομένων

Είναι μια σχεσιακή βάση δεδομένων όπου τα δεδομένα είναι χωρισμένα σε αρκετούς πίνακες συνδεδεμένους μεταξύ τους και όχι όλα σε μια στοιβή. Η σχεσιακή βάση μπορεί να επιστρέψει με μια εντολή SQL αρκετά στοιχεία από αρκετούς πίνακες στον χρήστη και έτσι να επιτευχθεί καλύτερη οργάνωση των δεδομένων σ' αυτήν.

Οι πίνακες αποτελούνται από δισδιάστατους πίνακες που οι στήλες τους θέτουν τα δεδομένα ενώ οι γραμμές αποτελούν την πληροφορία. Ένα πίνακας της MySQL έχει αυτήν την εμφάνιση.

* id_#	email	password	role	isEmailConfirmed	username	created_at	updated_at
1	13 psillovits@gmail.com	\$2b\$12\$XhLp1OoQ5EmEUaIMf2jX4uIBFL07MnJf1pT.QHTi5aY7Iw9TYE2	user		1 Psillovits	2022-10-12 00:04:56	2022-10-12 00:04:56
2	14 npsillou@gmail.com	\$2b\$12\$3UPqgJm/0ACWjQDVTARJ3.oypBe6NmWm20DgcQANxKp6Zsgwv...	user		1 npsillou	2022-10-12 00:04:56	2022-10-15 14:10:57
3	16 npsillou1@gmail.com	\$2b\$12\$6NuStkiRuUqprW7Cui9C...1kjU3q7BEQ3sOMzcQOXRREvdfCRLZk	user		0 npsillou	2022-10-12 00:04:56	2022-10-12 00:04:56
4	17 samhita.anna@eleden.org	\$2b\$12\$ifjAGm0Up0MSVEaEfuUUKum1Y5dihY8qLrHjoGVNZRU7klvmRB...	user		0 nikolas2610	2022-10-12 00:04:56	2022-10-12 00:04:56
5	18 rylynn.callahan@eleden.org	\$2b\$12\$5/rqRi5MxuzhRfEGbdJCMpKelyFD6no/AgwX6im3KmPCDmNzHi...	user		0 rylynn	2022-10-12 00:04:56	2022-10-12 00:04:56
6	19 caralena.fayrouz@eleden.org	\$2b\$12\$u24knean7c/GV3wFzjcp.wFunWLFd1jArPPgbtBWi9y7B1FIMa	user		0 caralena	2022-10-12 00:04:56	2022-10-12 00:04:56
7	20 caralena.fayrouz1@eleden.org	\$2b\$12\$E5.p3WYR0ZVBBXpNOckM8.BkgC68denKcl8eq1Pa9gixJ4vZom...	user		0 caralena	2022-10-12 00:04:56	2022-10-12 00:04:56
8	25 caralena.fayrouz1@eleden.org	\$2b\$12\$5Shwa6P9EwmGvGbjK9B9HKe9GyB/k09z5Fmej8RY4nhHF6RIK7z0f6	user		0 nikolas2610	2022-10-12 00:04:56	2022-10-12 00:04:56
9	27 hudaifa.khameron@eleden.org	\$2b\$12\$B4U09WuWhe0jyx2zQ3ls1.KWbFrlWrdW.P5dHgw0e0o1dnKxeasi	user		0 hudaifa	2022-10-12 00:06:19	2022-10-12 00:06:19
10	28 betina.launa@eleden.org	\$2b\$12\$WGsVz9g94M4nXl0yHze.ε3bxiPBd/5t/GgilRInobFqJ5eAovLxq	user		0 betina	2022-10-12 01:07:34	2022-10-12 01:07:34
11	30 betina.launa1@eleden.org	\$2b\$12\$J9lp0UlwUFjfmV4ikloHu78i9yKWnJpIRU0iFKCwif/rwbesm8y	user		0 betina	2022-10-12 01:10:01	2022-10-12 01:10:01
12	31 sare.caylee@minutestep.com	\$2b\$12\$54GMZKvQf/EBMv02mlCpdecZ21P5A54v4zJpspFKR8me8mYkq...	user		0 sare	2022-10-12 20:49:52	2022-10-12 20:49:52
13	33 baker.jeremiah@minutestep.com	\$2b\$12\$2wjL2tXu1ZW8Wtyb.bFQ0et3W.lImwdyJ92TDePq91m9ixC/0zSk	user		1 jeremiah	2022-10-12 20:51:36	2022-10-12 21:00:00
14	34 baker.jeremiah1@minutestep.com	\$2b\$12\$5/X7yVnq9cHG8S6vIG2wJeAwVHLCH5Q5GYNW9xfus1F4tCzsnwa	user		0 jeremiah	2022-10-12 20:52:44	2022-10-12 20:52:44
15	35 baker.jeremiah2@minutestep.com	\$2b\$12\$F6dzBptyKUSXOZdBf4cOuXHQ.v9AGT2vcXPDzklVndDdYiZZ0u	user		0 nikolas2610	2022-10-12 20:54:42	2022-10-12 20:54:42
16	36 baker.jeremiah3@minutestep.com	\$2b\$12\$2TXmxIszajnOXovXz2NesDvVnki3yJAgfNk0mVmQrdi8T1C.m	user		0 nikolas2610	2022-10-12 20:55:29	2022-10-12 20:55:29
17	37 psillovits5@gmail.com	\$2b\$12\$Xv6GUiU72/SbH9UGYeiFOiCHmaC9qAIQPK4wYBbv5jWXFm...	user		0 Psillovits	2022-10-14 21:05:07	2022-10-14 21:05:07
18	38 jordyan.laith@minutestep.com	\$2b\$12\$uZq2h8imZv4kU.4UJefid.XzHf4zBLnTnBo8j9cWxfX104b1ve1u	user		0 Psillovits	2022-10-14 21:05:50	2022-10-14 21:05:50
19	39 cole.yavontae@minutestep.com	\$2b\$12\$91DalyMCj9abiTslnhKteM3HIO1TmqeGW0nbOYL4wsczb480.	user		0 cole	2022-10-14 22:26:22	2022-10-14 22:26:22
20	40 samayah.randy@minutestep.com	\$2b\$12\$7NCTe8pBVLI5Hrkl.Swpd.LwiotdgnarHo6sa7HR8YtDkDKMccsc7G	user		1 samayah	2022-10-14 22:28:22	2022-10-15 00:23:46
21	41 kam.keynan@minutestep.com	\$2b\$12\$SA0hp0q5D/pYjkiWSPcif.eirCZeT0z2iod2BFiKA7.ueliGireC	user		1 keynan	2022-10-14 23:21:58	2022-10-14 23:22:21
22	42 clodagh.elliemae@minutestep.com	\$2b\$12\$snQ6f0TFkaO3yfstz8QX.utbVseV7GqHbSvYtu6W9RC3wI2iop0W	user		0 clodagh	2022-10-14 23:23:40	2022-10-14 23:23:40
23	47 lewyn.sven@minutestep.com	\$2b\$12\$7J3T4qyeH2LoeHDbSvUuZWCILbwOlughGbfLBGP2e/appKqE...	user		1 lewyn	2022-10-16 13:16:31	2022-10-16 13:17:11
24	48 taniyah.evalene@minutestep.com	\$2b\$12\$B5L81j1pWpJ0Krcpf7Q30E9DkrCgq0o130Gv.jjWUqdn/32PLPG	user		1 taniyah	2022-10-28 17:24:18	2022-10-28 17:26:25

Εικόνα 5 - Πίνακας MySQL

## 2.5.2 Type ORM

Το Type ORM [22] είναι μία βιβλιοθήκη ανοικτού κώδικα που μπορεί να χρησιμοποιηθεί στην JavaScript και στην TypeScript ώστε να πραγματοποιηθεί οποιαδήποτε σύνδεση με μια οποιαδήποτε βάση δεδομένων. Το ORM είναι γνωστό ως Object Relational Mapper (Αντικειμενικός Σχεσιακός Χαρτογράφος). Με την χρήση της TypeScript, αναπτύσσονται τα αντικείμενα με τους τύπους τους στον κώδικα και η Type ORM τα μετατρέπει σε πίνακες της βάσης δεδομένων.

Η εξής βιβλιοθήκη δίνει την δυνατότητα στο προγραμματιστή να εξοικονομήσει χρόνο καθώς δεν συγγράφει ερωτήματα SQL, αλλά πιο απλές εντολές σε JavaScript. Επίσης, το σχήμα της βάσης κατασκευάζεται με το τρέξιμο του προγράμματος και μπορεί να γίνουν εύκολα τροποποιήσεις κατά την διάρκεια της ανάπτυξης της εφαρμογής.

## 2.6 WebSocket

Το WebSocket [23] είναι πρωτόκολλο επικοινωνίας στο διαδίκτυο μεταξύ του εξυπηρετητή και του πελάτη. Η κύρια διαφορά του με άλλα πρωτόκολλα όπως είναι το HTTP είναι ότι η σύνδεση παραμένει ζωντανή μέχρι να διακοπεί από κάποια πλευρά. Δίνει την δυνατότητα στους χρήστες να επικοινωνούν σε πραγματικό χρόνο χωρίς διακοπές. Αυτό το πρωτόκολλο χρησιμοποιείται κυρίως σε παιχνίδια και σε εφαρμογές συνομιλίας όπου στέλνονται συνεχώς δεδομένα χωρίς να χρειάζεται η ανανέωση της σελίδας.

Το WebSocket θα χρησιμοποιηθεί αρκετά στην ΠΕ και συγκεκριμένα στο ΗΤ καθώς η πλατφόρμα αφορά παιχνίδι με πολλούς παίκτες σε πραγματικό χρόνο που θα επικοινωνούν μεταξύ τους.

## 2.7 Λογισμικά εργαλεία

Πέραν από τις τεχνολογίες που αναφέρθηκαν, χρειάζονται επιπρόσθετα εργαλεία που θα βοηθήσουν στην ανάπτυξη της πλατφόρμας. Τα εργαλεία θα χρησιμεύσουν στην συγγραφή, έλεγχο και σώσιμο του κώδικα, στον έλεγχο της πλατφόρμας, στην εμφάνιση της και σε εξωτερικές υπηρεσίες.

### 2.7.1 Visual Studio Code

Ένα πηγαίος κώδικας μπορεί να γραφτεί και σε ένα απλό επεξεργαστή κειμένου. Όμως ο επεξεργαστής κειμένου δεν είναι τόσο φιλικός με τον προγραμματιστή καθώς όσο μεγαλώνει η εφαρμογή παράλληλα και ο κώδικας, γίνεται δυσκολότερο για τον ίδιο. Για αυτό τον λόγο δημιουργήθηκαν επεξεργαστές πηγαίου κώδικα για τους προγραμματιστές, ώστε να αυξηθεί η αποτελεσματικότητα. Σε ένα τέτοιο επεξεργαστή, υπάρχουν διαφορετικά χρώματα, ειδικές συντομεύσεις, πρόσθετα εργαλεία αλλά το πιο σημαντικό, μπορεί να εντοπίσει τα συνταχτικά λάθη του προγραμματιστή ώστε να τα διορθώσει.

Σε αυτή την ΠΕ θα χρησιμοποιηθεί το VS Code [24] (Visual Studio Code) που είναι δωρεάν καθώς είναι εφαρμογή ανοιχτού κώδικα. Το VS Code έχει μεγάλη κοινότητα και παρέχει απεριόριστα επιπρόσθετα εργαλεία δωρεάν ώστε να κάνει την ζωή του προγραμματιστή πιο εύκολη.

### 2.7.2 Postman

Το Postman [25] είναι ένα εργαλείο που χρησιμοποιούν οι προγραμματιστές ώστε να δοκιμάσουν τα APIs (Application programming interface) τους. Μπορούν να δημιουργήσουν APIs για όλα τα είδη των κλήσεων που παρέχει το πρωτόκολλο HTTP όπως τα GET, POST, PUT and PATCH. Έχουν την δυνατότητα να αποθηκεύσουν όλες τις κλήσεις τους για τα διαφορετικά περιβάλλον που δημιουργούν και να τα μοιραστούν με άλλους προγραμματιστές είτε στην μορφή που τα έχει το Postman ή σε μορφή εγγράφου.

### 2.7.3 Figma

Το Figma [26] είναι ένα εργαλείο που κυρίως χρησιμοποιείται από τους Web Designer, ώστε να σχεδιάσουν την εμφάνιση της σελίδας πριν φτάσει στα χέρια του προγραμματιστή. Συχνά το ρόλο του Web Designer τον αναλαμβάνει ένας προγραμματιστής ιστοσελίδων καθώς μέσα από την εμπειρία του μπορεί να δημιουργήσει μία ιστοσελίδα που θα είναι όμορφη και φιλική προς τον χρήστη (UI & UX Experience). Το Figma δίνει την δυνατότητα στο χρήστη να σχεδιάσει την ιστοσελίδα με απλή σχεδίαση και χωρίς κώδικα. Επιλέγει την κατάλληλη γραμματοσειρά, την παλέτα με τα χρώματα του και αρχίζει την γραφική υλοποίηση για την ιστοσελίδα. Έτσι σχηματίζει τη μορφή που θα έχει η σελίδα χωρίς να ασχολείται με την υλοποίηση του κώδικα και μπορεί να φτιάξει και τις εμφανίσεις της σελίδας σε άλλες συσκευές.

## 2.7.4 Send Grid

Το SendGrid [27] είναι ένας πάροχος του SMTP πρωτόκολλου που εξυπηρετεί την αποστολή των ηλεκτρονικών μηνυμάτων(emails). Με την δημιουργία λογαριασμού στο SendGrid, δίνει την δυνατότητα στο χρήστη να δημιουργήσει το δικό του Web API, ώστε να το χρησιμοποιήσει στην εφαρμογή του και να στέλνει ασφαλές τα email. Ένα συχνό πρόβλημα που αντιμετωπίζουν οι προγραμματιστές και το SendGrid το λύνει, είναι ότι στον περιβάλλον υλοποίησης της πλατφόρμας του προγραμματιστή, δεν παρέχεται SMTP διακομιστής και έτσι τα email αναγνωρίζονται ως κακόβουλα.

## 2.7.5 Git

Το Git [28] είναι ένα από τα βασικότερα εργαλεία του προγραμματιστή καθώς διαφυλάσσεται έτσι ο κώδικας του στο διαδίκτυο, όχι απαραίτητα ως κοινόχρηστος, και κρατάει όλες τις εκδόσεις του κώδικα του. Αυτό το εργαλείο δίνει την δυνατότητα στις ομάδες που αποτελούνται από προγραμματιστές να δουλεύουν παράλληλα στον ίδιο κώδικα και να συγχωνεύσου εύκολα των κώδικα τους. Επίσης παρέχει ακόμη αρκετές δυνατότητες, όπως την αναίρεση την υλοποίησης τους, την επιστροφή σε προηγούμενη έκδοση, έλεγχος του κώδικα πριν την συγχώνευση και αρκετά άλλα. Τέλος, το Git είναι ανοικτού κώδικα και παρέχεται δωρεάν.

## 2.8 Σύνοψη κεφαλαίου

Σε αυτό το κεφάλαιο έγινε μία γενική αναφορά ως προς τι έννοιες που θα πρέπει να γνωρίζει ένας προγραμματιστής ιστοσελίδων. Έγινε ανάλυση ως προς τις τεχνολογίες, τις βιβλιοθήκες και τα εργαλεία που θα χρησιμοποιηθούν για την υλοποίηση της πλατφόρμας.

### 3. Σχεδιασμός διαδικτυακής πλατφόρμας

Σε αυτό το κεφάλαιο θα γίνει η ανάλυση της πλατφόρμας ως προς τους ρόλους και τις λειτουργίες που θα έχουν οι χρήστες εντός αυτής αλλά και εντός του ΗΤ. Θα γίνει ανάλυση για τα ΠΜΚ ώστε το ΗΤ να προσφέρει τις απαραίτητες λειτουργίες στους χρήστες και να είναι μια αναπαράσταση ενός αληθινού τραπέζιου.

#### 3.1 Κατηγορίες χρηστών

Η πλατφόρμα θα εξυπηρετεί διαφόρων ειδών χρήστες και γι' αυτό απαιτείται η ανάλυση και ο προσδιορισμός των λειτουργιών κάθε χρήστη. Οι ομάδες χρηστών που θα εξυπηρετεί η πλατφόρμα θα είναι τρεις: ο ανώνυμος χρήστης, ο εγγεγραμμένος χρήστης και ο διαχειριστής της πλατφόρμας.

Ο ανώνυμος χρήστης θα έχει περιορισμένες λειτουργίες στην πλατφόρμα σε σχέση με τους υπόλοιπους χρήστες λόγω της ανώνυμης ταυτότητας του. Θα έχει πρόσβαση στην αίθουσα αναμονής και εύρεσης παιχνιδιού, όπου θα υπάρχουν τα διαθέσιμα παιχνίδια και θα μπορεί να εισέλθει μόνο σε κάποιο ΗΤ για να λάβει μέρος σε ένα παιχνίδι. Προϋπόθεση για να εισέλθει στην βασική σελίδα είναι να ορίσει ένα ψευδώνυμο.

Ο εγγεγραμμένος χρήστης θα μπορεί να χρησιμοποιήσει σχεδόν τις περισσότερες λειτουργίες της πλατφόρμας. Θα μπορεί να επεξεργαστεί το προφίλ του, να εισέλθει σε ηλεκτρονικά τραπέζια για να λάβει μέρος σε κάποιο παιχνίδι και το πιο σημαντικό, θα έχει την δυνατότητα να δημιουργήσει το δικό του παιχνίδι ή τραπέζι.

Ο διαχειριστής, λόγω και της ιδιότητας του, θα έχει όλες τις λειτουργίες των υπόλοιπων χρηστών κα επιπλέον κάποιες άλλες που θα σχετίζονται με την ομαλή διαχείριση της πλατφόρμας. Θα έχει πρόσβαση στην λίστα με τους διαθέσιμους χρήστες της πλατφόρμας, θα μπορεί να την επεξεργαστεί, μέχρι και να διαγράψει χρήστες από αυτή. Θα λαμβάνει επίσης και μηνύματα επικοινωνίας από τους χρήστες για τυχόν διευκρινήσεις σχετικά με την πλατφόρμα ή σφάλματα της πλατφόρμας.

#### 3.2 Ανάλυση πλατφόρμας

Πέραν από τους κύριους ρόλους που αναλύθηκαν στο προηγούμενο κεφάλαιο, οι εγγεγραμμένοι χρήστες θα έχουν τρεις επιπλέον ρόλους. Ένας χρήστης μπορεί να είναι είτε ο δημιουργός ενός παιχνιδιού (Game Designer), είτε ο υπεύθυνος του τραπέζιου (Game Master) είτε ένα απλός παίχτης.

Ο Game Designer (GD), θα μπορεί να δημιουργήσει ή να τροποποιήσει ένα παιχνίδι. Θα έχει τις εξής επιλογές στο παιχνίδι που θα δημιουργήσει, να μπορεί να το κρατήσει για τον εαυτό του και να το φορτώσει σε ένα ΗΤ ή να το ορίσει ως κοινόχρηστο για να είναι διαθέσιμο και στους υπόλοιπους χρήστες.

Τον ρόλο του GM θα τον έχει αυτόματα ο χρήστης που θα δημιουργήσει το ΗΤ. Για να δημιουργήσει το τραπέζι πρέπει να διαλέξει ένα παιχνίδι είτε από τα δικά του είτε από τα κοινόχρηστα. Επιπλέον πρέπει να προσθέσει κάποιες ρυθμίσεις. Με την ολοκλήρωση των ρυθμίσεων, το τραπέζι θα είναι διαθέσιμο για να εισέλθουν οι υπόλοιποι χρήστες, να πάρουν θέση στο ΗΤ και να παίξουν το επιλεγμένο παιχνίδι. Οι χρήστες που απλά θα λάβουν μέρος στο ΗΤ είναι απλοί παίχτες.

### 3.3 Ανάλυση παιχνιδιών με κάρτες

Τα ΠΜΚ που επιλέχθηκαν για ανάλυση είναι τα δημοφιλή Blackjack, Μπιρίμπα, Poker, Uno και Saboteur. Τα κριτήρια επιλογής αυτών των ΠΜΚ, έχουν ως εξής:

- Να χρησιμοποιούν την κλασική τράπουλα χαρτιών ή ειδική τράπουλα χαρτιών
- Να είναι ατομικά ή ομαδικά
- Το παιχνίδι να χρησιμοποιεί περισσότερες από μία τράπουλες κατά την διάρκεια του παιχνιδιού
- Οι παίχτες να έχουν διαφορετικούς ρόλους

#### 3.3.1 Blackjack

##### Γενικοί κανόνες:

- Παίζεται με μία κανονική τράπουλα χαρτιών (52 φύλλα χωρίς μπαλαντέρ)
- Παίζεται από 2 έως 7 παίχτες
- Ένας παίχτης είναι ο ντίλερ και οι άλλοι παίχτες παίζουν ατομικά εναντίων του ντίλερ

##### Αξία των χαρτιών:

- Από το νούμερο 2 έως 10 οι πόντοι είναι όσοι αυτοί που αναγράφονται στο φύλλο
- Ο βαλές, η ντάμα και ο ρήγας έχουν αξία 10
- Ο άσσος έχει αξία 1 ή 11 ανάλογα με την επιλογή του παίχτη που το έχει στην κατοχή του

##### Πως παίζεται:

Όλοι οι παίχτες είναι εναντίων του ντίλερ, ο ντίλερ είναι ενεργός στο παιχνίδι παίζοντας σχεδόν με τον ίδιο τρόπο που παίζουν και οι υπόλοιποι παίχτες.

Ο ντίλερ αρχικά μοιράζει από ένα χαρτί σε όλους τους παίχτες (και στον εαυτό του). Όλα τα χαρτιά είναι φανερά σε όλους. Το δεύτερο χαρτί που δίνεται στους παίχτες είναι και πάλι φανερό, αυτή την φορά όχι όμως για τον ντίλερ. Το δεύτερο χαρτί του ντίλερ παραμένει κρυφό στους υπολοίπους παίχτες. Το παιχνίδι στην Ελλάδα ονομάζεται και “21” για τον λόγο ότι το συνολικό άθροισμα των χαρτιών του κάθε παίχτη δεν πρέπει να ξεπερνάει το 21. Όταν έρθει η σειρά του παίχτη έχει το δικαίωμα να ζητήσει χαρτί από τον ντίλερ ή να πάει πάσο. Αν ξεπεράσει το 21 καίγεται και βγαίνει εκτός γύρου. Αυτό ισχύει για όλους τους παίχτες και για τον ντίλερ.

##### Νικητής:

Στο τέλος του γύρου εμφανίζει τα χαρτιά του ο ντίλερ, αν είναι πάνω από 21 τότε όλοι οι παίχτες που δεν κήκαν κατά την διάρκεια του γύρου, νικάνε. Αν δεν καεί ο ντίλερ τότε γίνεται σύγκριση μεταξύ στο άθροισμα χαρτιών που έχει κάθε παίχτης με αυτό που έχει ο ντίλερ. Νικάει κάποιος αν έχει κάνει ακριβώς 21 πόντους αλλιώς αυτός που έχει το μεγαλύτερο άθροισμα (με μεγαλύτερο το 21). Σε περίπτωση που έχουν το ίδιο άθροισμα παίχτης και ντίλερ, θεωρείται ισοπαλία. Όταν ελεγχτούν όλοι οι παίχτες ξεχωριστά τότε ολοκληρώνεται και ο γύρος. Το παιχνίδι παίζεται όσες φορές ακόμη επιθυμούν οι παίχτες χωρίς να είναι απαραίτητη η καταγραφή των νικών γιατί συνήθως το παιχνίδι παίζεται με μάρκες.



Συμπεράσματα:

- Υπάρχουν δύο ρόλοι στο παιχνίδι, ο υπεύθυνος του παιχνιδιού που θα είναι ο ντίλερ και οι παίχτες
- Μοιράζονται κάρτες στους παίχτες με την έναρξη του παιχνιδιού
- Ο ντίλερ έχει στην κατοχή του την τράπουλα και αυτός μοιράζει τις κάρτες στους παίχτες όταν του ζητηθεί
- Τα χαρτιά του κάθε παίχτη είναι φανερά στο τραπέζι
- Οι παίχτες πρέπει να γνωρίζουν ποιος είναι ο ντίλερ
- Θα πρέπει να υπάρχει πεδίο επιλογής με το οποίο οι παίχτες θα ζητάνε χαρτί ή θα πηγαίνουν πάσο
- Τα χαρτιά από κρυφά μπορούν να γίνουν φανερά
- Ο ντίλερ μαζεύει τα φύλλα, τα ανακατεύει και τα ξεκινάει από την αρχή το παιχνίδι



Εικόνα 6 - Παιχνίδι Blackjack

### 3.3.2 Μπιρίμπα

Γενικοί κανόνες:

- Το παιχνίδι παίζεται με 4 παίχτες
- Το παιχνίδι παίζεται σε 2 ομάδες όπου η κάθε ομάδα αποτελείται από 2 παίχτες
- Οι παίκτες που ανήκουν στην ίδια ομάδα κάθονται διαγωνίως απέναντι
- Όλοι οι παίχτες έχουν τον ίδιο ρόλο στο παιχνίδι
- Παίζεται με 2 κανονικές τράπουλες χαρτιών και ακόμη 2 μπαλαντέρ



Εικόνα 7 - Τα χαρτιά που έχει η τράπουλα της μπιρίμπας

#### Μοίρασμα καρτών:

- Μοιράζονται από 11 κλειστά χαρτιά στον κάθε παίχτη
- Μοιράζονται επίσης από 11 χαρτιά σε κάθε ομάδα τα οποία είναι κλειστά και είναι διαθέσιμα υπό κάποιες συνθήκες του παιχνιδιού και μόνο. Ονομάζονται μπιριμπάκια
- Γυρίζεται ένα χαρτί ανοιχτά στο κέντρο το οποίο και ορίζει το ατού του παιχνιδιού
- Η υπόλοιπη τράπουλα είναι διαθέσιμη στο τραπέζι για το τράβηγμα χαρτιών από τους παίχτες

#### Ιδιότητες των χαρτιών:

- Όλα είναι κανονικά φύλλα εκτός από τους μπαλαντέρ οι οποίοι μπορούν να πάρουν την θέση οποιουδήποτε χαρτιού
- Τα φύλλα με τον αριθμό 2 είναι και αυτοί μπαλαντέρ
- Η κανονική τράπουλα έχει 4 φυλές. Μία από αυτές τις φυλές είναι το ατού. Αυτή η φυλή ορίζεται από το πρώτο φύλο που γυρνάει στο κέντρο αφού μοιραστούν τα χαρτιά στους παίχτες. Οι συνδυασμοί των χαρτιών με αυτή την φυλή έχουν περισσότερη αξία
- Η συνδυασμοί των καρτών που βρίσκονται στο τραπέζι έχουν διαφορετικούς πόντους ανάλογα με το αν μια μπιρίμπα έχει μπαλαντέρ ή όχι ή αν μια μπιρίμπα ανήκει στην φυλή των ατού

### Πως παίζεται:

Σε κάθε γύρο παίζει πρώτα ο αμέσως επόμενος παίχτης από αυτόν που έπαιξε στον προηγούμενο γύρο πρώτος. Ο παίχτης που παίζει έχει το δικαίωμα να τραβήξει ένα χαρτί από την τράπουλα ή να μαζέψει όλα τα χαρτιά από την στοίβα με τα σκάρτα χαρτιά με την προϋπόθεση να δικαιολογήσει έστω ένα από τα χαρτιά που ήταν στην στοίβα. Η δικαιολόγηση γίνεται με το να προσθέσει ένα από τα σκάρτα χαρτιά, στα χαρτιά που έχει ήδη κατεβάσει ή με την βοήθεια ενός σκάρτου να δημιουργήσει ένα συνδυασμό με ελάχιστο πλήθος 3 χαρτιά της ίδιας φυλής. Έπειτα τα κατεβάζει κάτω στο τραπέζι για να είναι ορατά σε όλους του παίχτες. Μπορεί να δημιουργήσει και άλλους συνδυασμούς από τα χαρτιά που έχει στο χέρι εφόσον το επιθυμεί. Για να ολοκληρώσει το γύρο του οφείλει να ρίξει ένα σκάρτο στην στοίβα με τα σκάρτα. Αυτό γίνεται μέχρι να τελειώσουν όλα τα φύλλα της τράπουλας. Η κάθε ομάδα έχει συγκεκριμένο χώρο στο τραπέζι όπου μαζεύει τις μπιρίμπες της. Στόχος της κάθε ομάδας είναι να συνδυάσει όσα περισσότερα φύλλα μπορεί, να κάνει μπιρίμπες και να τις κατεβάσει στο τραπέζι. Οι μπιρίμπες πρέπει να έχουν ελάχιστο πλήθος 7 χαρτιά. Αν σε κάποιο παίχτη τελειώσουν τα φύλλα παίρνει μπιριμπάκια με την προϋπόθεση ότι η ομάδα του έχει τουλάχιστο μία μπιρίμπα. Η κάθε ομάδα έχει στην διάθεση της ένα μπιριμπάκι. Αν τελειώσουν τα χαρτιά ενός παίχτη αφού έχει πάρει και μπιριμπάκια τότε ολοκληρώνεται το παιχνίδι με την προϋπόθεση ότι έχει μια καθαρή μπιρίμπα. Όταν ολοκληρωθεί το παιχνίδι τότε γίνεται και η καταμέτρηση των πόντων της κάθε ομάδας.

### Νικητής:

Στο τέλος του κάθε παιχνιδιού γίνεται η καταμέτρηση της βαθμολογίας. Η ομάδα που θα φτάσει πρώτη στους 3000 πόντους ή σε αριθμό που έθεσαν οι παίχτες στην αρχή του παιχνιδιού είναι η νικήτρια.

### Συμπέρασμα:

- Όλοι οι παίχτες έχουν τον ίδιο ρόλο
- Παίζουν πάντα με την ίδια σειρά
- Πρέπει να αναγράφεται στο παίχτη σε ποια ομάδα ανήκει καθώς οι παίχτες που ανήκουν στην ίδια ομάδα κάθονται διαγωνίως
- Το πλέγμα του τραπέζιου αποτελείται από 2 μέρη(όσες και οι ομάδες) όπου το κάθε ένα είναι ο χώρος όπου προσθέτουμε τους συνδυασμούς των καρτών τους οι παίχτες
- Έχει σημασία ο αριθμός των υπόλοιπων καρτών της τράπουλας
- Ο παίχτης μαζεύει όλα τα φύλλα από την στοίβα με τα σκάρτα
- Δημιουργούνται δύο επιπλέον τράπουλες μετά το μοίρασμα των καρτών στους παίχτες
- Τα φύλλα που κρατάνε οι παίχτες είναι πάντα κρυφά
- Τα φύλλα που είναι στο τραπέζι είναι πάντα ανοιχτά
- Στο τέλος πρέπει να εμφανιστούν τα φύλλα στα χέρια των παιχτών και από το τραπέζι τους για καταμέτρηση της βαθμολογίας

### 3.3.3 Πόκερ

#### Γενικοί κανόνες:

- Παίζεται με μία κανονική τράπουλα χαρτιών (χωρίς μπαλαντέρ)
- Παίζεται συνήθως από 2 παίκτες και συνήθως έως 7 παίκτες
- Υπάρχουν 2 ρόλοι: οι παίκτες και ο ντίλερ



Εικόνα 8 - Παιχνίδι Πόκερ

#### Πως παίζεται:

Στο τραπέζι υπάρχουν οι παίκτες και ο ντίλερ. Ο κάθε παίκτης παίζει μόνος του εναντίων όλων των άλλων. Το παιχνίδι ως επί τον πλείστον παίζεται με μάρκες και ο ρόλος του ντίλερ είναι να μοιράζει τα χαρτιά και τις μάρκες. Αρχικά ο ντίλερ μοιράζει από 2 κλειστά φύλλα στον κάθε παίκτη και επιλέγουν αν θα ποντάρουν ή όχι. Στην συνέχεια απομένουν 3 γύροι όπου οι παίκτες έχουν την επιλογή να ποντάρουν για να συνεχίσουν ή να πάνε πάσο. Στο πρώτο γύρο ο ντίλερ γυρίζει 3 χαρτιά στο τραπέζι που είναι ανοικτά και κοινά προς όλους ενώ στους άλλους δύο γύρους από ένα χαρτί αντίστοιχα.

#### Νικητής:

Οι παίκτες πρέπει να συνδυάσουν τα φύλλα που κρατάνε και αυτά που είναι κάτω στο τραπέζι ώστε να κάνουν το καλύτερο συνδυασμό, για να κερδίσουν την παρτίδα. Όταν ολοκληρωθεί το παιχνίδι επαναλαμβάνεται για όσους γύρους επιθυμούν οι παίκτες

Συμπέρασμα:

- Μέγιστοι παίκτες 7
- Χρήση κανονικής τράπουλας
- Τα χαρτιά που γυρνάει ο ντίλερ βρίσκονται στην μέση του τραπέζιου που είναι ορατά σε όλους
- Τα φύλλα στα χέρια των παιχτών είναι κρυφά
- Ατομικό πλέγμα για τα χαρτιά στο χέρι του κάθε παίχτη με μέγιστο 2 χαρτιά
- Ο ντίλερ είναι ο GM
- Παίζεται με μάρκες
- Στο τέλος του γύρου εμφανίζονται λίστα με τα χαρτιά του τραπέζιου και τα χαρτιά που έχουν οι παίκτες στα χέρια τους για να καθοριστεί ο νικητής
- Ο παίχτης έχει την δυνατότητα να μην λάβει μέρος στον γύρο (πιθανώς λόγω κακού συνδυασμού χαρτιών στο χέρι) και πρέπει να διακρίνεται μέσω της κατάστασης του παίχτη, ώστε να συμπεριληφθεί στους νικητές

### 3.3.4 Uno

Γενικοί κανόνες:

- Παίζεται από 2 έως 10 παίκτες
- Υπάρχει μόνο ένας ρόλος στο παιχνίδι και κάθε παίχτης παίζει μόνος του
- Παίζεται από ειδική τράπουλα με χαρτιά. Η τράπουλα είναι διαμορφωμένη ως εξής: 76 Αριθμητικές κάρτες 0-9 σε 4 χρώματα (πράσινο, κόκκινο, κίτρινο, μπλε) και 32 κάρτες ενέργειας



Εικόνα 9 - Κάρτες από την τράπουλα του Uno

Οι ειδικές κάρτες έχουν ιδιότητες οι οποίες είναι:

- “Παράλειψη” ακυρώνει την σειρά του επόμενου παίχτη
- “Αντίστροφη” αλλάζει την κατεύθυνση της σειράς που παίζουν οι παίχτες
- “Ίσοπαλία +2”, ο επόμενος παίχτης τραβάει ακόμα 2 χαρτιά από την τράπουλα
- “Άγρια” όπου αλλάζει το χρώμα του παιχνιδιού
- “Άγρια +4” όπου αλλάζει χρώμα και ο επόμενος παίχτης τραβάει και 4 φύλλα από την τράπουλα

Πως παίζεται:

Μοιράζονται 7 κρυφά φύλλα σε κάθε παίχτη και ένα φύλλο ανοικτό στην στοίβα με τα σκάρτα στο κέντρο. Οι παίχτες παίζουν με την σειρά που κάθονται στο τραπέζι εκτός αν χρησιμοποιηθεί κάρτα ενέργειας που αλλάζει την φορά. Σε κάθε γύρο αν ο παίχτης έχει χαρτί που ταιριάζει στο χρώμα ή αριθμό ή ειδική κάρτα, τότε ρίχνει χαρτί στην στοίβα με τα σκάρτα αλλιώς τραβάει φύλλο από την τράπουλα. Αν το χαρτί που τράβηξε μπορεί να το ρίξει μέσα στην στοίβα τότε το ρίχνει, αλλιώς παίζει ο επόμενος παίχτης.

Όταν σε ένα παίχτη απομείνει ένα χαρτί οφείλει να πει “UNO” για να δηλώσει έτσι στους υπόλοιπους παίχτες πως του έμεινε ένα χαρτί και στην συνέχεια να χρησιμοποιήσει το τελευταίο χαρτί του για να ολοκληρώσει το παιχνίδι. Τότε κρίνεται ο νικητής του παιχνιδιού και οι υπόλοιποι παίχτες γυρνάνε τα χαρτιά τους για να καταμετρηθούν οι πόντοι. Αν δεν μπορεί να χρησιμοποιήσει το τελευταίο του χαρτί τραβάει χαρτί και επαναλαμβάνεται η διαδικασία με την δήλωση τους ενός χαρτιού μέχρι το τέλος του παιχνιδιού. Εάν σε κάποιο παίχτη απομείνει ένα χαρτί και δεν πει UNO, πρέπει να πάρει 2 νέα χαρτιά.

Νικητής:

Το παιχνίδι ολοκληρώνεται όταν κάποιος παίχτης φτάσει στους 500 πόντους πρώτος.

Συμπέρασμα:

- Μέγιστα άτομα 8
- Δεν μπορεί να γίνεται η επιλογή του επόμενου παίχτη γιατί υπάρχουν περιπτώσεις όπου επόμενος παίχτης είναι 2 θέσεις στα δεξιά του ενεργού παίχτη ή στην επόμενη αριστερή θέση
- Διαφορετική τράπουλα από την κλασσική τράπουλα χαρτιών (ειδική)
- Ατομικό πλέγμα για τα χαρτιά στο χέρι του κάθε παίχτη χωρίς μέγιστο όριο στα χαρτιά
- Υπάρχουν κανονικές κάρτες και κάρτες ενέργειας, αλλά όλες ανήκουν στην ίδια τράπουλα. (Και στην μοιρασιά αλλά και στην τράπουλα από όπου τραβάει ο παίχτης)
- Ειδικό πεδίο επιλογής με το οποίο θα ενημερώνει ο παίχτης ότι λέει UNO
- Τα χαρτιά που κρατάνε οι παίχτες είναι κρυφά
- Όλοι οι παίχτες έχουν το ίδιο ρόλο
- Στο τέλος πρέπει να εμφανιστούν τα φύλλα των χεριών των παιχτών για καταμέτρηση της βαθμολογίας
- Υπάρχει πίνακας βαθμολογίας που χειρίζεται ο GM

### 3.3.5 Saboteur

#### Γενικοί κανόνες:

- Παίζεται από 3 έως 10 παίκτες
- Υπάρχουν 2 διαφορετικοί ρόλοι στο παιχνίδι: οι σαμποτέρ και οι χρυσοθήρες
- Παίζεται με ειδική τράπουλα καρτών που αποτελείται από 44 κάρτες μονοπατιών, 27 κάρτες ενεργειών, 28 κάρτες χρυσού, 7 χρυσοθήρες και 4 σαμποτέρ

#### Οι ειδικές κάρτες έχουν ιδιότητες οι οποίες είναι:

- Οι 44 κάρτες μονοπατιών χρησιμοποιούνται για την δημιουργία μονοπατιού ώστε οι χρυσοθήρες να φτάσουν στο χρυσό. Ανάμεσα σε αυτές τις κάρτες υπάρχει μία κάρτα έναρξης (απεικονίζει μία σκάλα) και τρεις κάρτες τερματισμού εκ των οποίων η μία είναι ο χρυσός που αναζητούν οι παίκτες
- Οι 27 κάρτες ενεργειών δίνουν την δυνατότητα στους παίκτες να χαλάσουν ή να επισκευάσουν μέρη από το καρότσι κάποιου άλλου παίχτη ώστε να του είναι αδύνατο χτίσει κάποιο μονοπάτι. Επίσης υπάρχουν κάρτες ενέργειας που καταστρέφουν κάποιο μονοπάτι ή που μπορούν να δουν αν βρίσκεται όντως ο χρυσός σε μία από τις 3 κάρτες τερματισμού
- Οι 28 κάρτες χρυσού είναι για τους πόντους του κάθε παίχτη που καταγράφονται στο τέλος
- Οι 11 κάρτες με τους ρόλους των παιχτών αποτελούνται από 7 που είναι οι χρυσοθήρες και άλλους 4, τους σαμποτέρ



Εικόνα 10 - Κάρτες από την τράπουλα του Saboteur

### Πως παίζεται:

Με βάση το πόσοι παίκτες κάθονται στο τραπέζι δημιουργείται η τράπουλα με τους ρόλους και μοιράζονται στους παίκτες, είναι κρυφοί προς του άλλους παίκτες. Η μοιρασιά γίνεται με τον εξής τρόπο:

- 3 παίκτες: 1 σαμποτέρ και 3 χρυσοθήρες
- 4 παίκτες: 1 σαμποτέρ και 4 χρυσοθήρες
- 5 παίκτες: 2 σαμποτέρ 4 χρυσοθήρες
- 6 παίκτες: 2 σαμποτέρ και 5 χρυσοθήρες
- 7 παίκτες: 3 σαμποτέρ και 5 χρυσοθήρες
- 8 παίκτες: 3 σαμποτέρ και 6 χρυσοθήρες
- 9 παίκτες: 3 σαμποτέρ και 7 χρυσοθήρες
- 10 παίκτες: όλες οι κάρτες των νάνων

Στην συνέχεια από τις κάρτες μονοπατιού τοποθετείτε η κάρτα έναρξης στο τραπέζι φανερά και οι τρεις κάρτες τερματισμού κρυφά στην θέση τους, μετέπειτα ενώνονται οι κάρτες μονοπατιών και οι κάρτες ενεργειών και μοιράζεται ο ανάλογος αριθμός καρτών στους παίκτες με βάση τους κανόνες.

- 3 έως 5 παίκτες: Κάθε παίκτης παίρνει 6 κάρτες
- 6 έως 7 παίκτες: Κάθε παίκτης παίρνει 5 κάρτες
- 8 έως 10 παίκτες: Κάθε παίκτης παίρνει 4 κάρτες

Η αρχική τοποθέτηση των καρτών γίνεται με συγκεκριμένο τρόπο στο πλέγμα του τραπεζιού. Οι κάρτες τοποθετούνται μόνο κάθετα στο τραπέζι και δεν έχει σημασία η φορά. Επομένως ο παίκτης μπορεί να περιστρέψει την κάρτα. Το παιχνίδι παίζεται σε 2 ομάδες ανάμεσα σε χρυσοθήρες και σε σαμποτέρ. Οι παίκτες δεν γνωρίζουν ποιος ανήκει στην ομάδα τους, αλλά μπορούν να το υποψιαστούν στην πορεία του παιχνιδιού.

Όταν έρθει η σειρά του παίκτη τότε παίρνει μία κάρτα από την τράπουλα και έχει το δικαίωμα να τοποθετήσει μία κάρτα μονοπατιού στο τραπέζι, να χρησιμοποιήσει μια κάρτα ενέργειας σε κάποιο παίκτη ή να πάει πάσο και να κάψει μία κάρτα του στην στοίβα με τα σκάρτα. Αν ο παίκτης έχει κάποιο μέρος από το καρότσι του σπασμένο τότε δεν μπορεί να τοποθετήσει κάρτα μονοπατιού μέχρι να το φτιάξει ο ίδιος ή κάποιος συμπαίκτης τους με κάρτα ενέργειας. Αν τελειώσουν οι κάρτες και οι χρυσοθήρες δεν έφτασαν στο χρυσό, τότε οι παίκτες συνεχίζουν να παίζουν μέχρι να τελειώσουν οι κάρτες από το χέρι τους και να ολοκληρωθεί το παιχνίδι.

### Νικητής:

Ο στόχος του παιχνιδιού για τους χρυσοθήρες είναι να δημιουργήσουν ένα συνεχές μονοπάτι για να φτάσουν στο χρυσό ενώ για τους σαμποτέρ, να αποτρέψουν τους χρυσοθήρες από τον στόχο τους αυτό. Μία ομάδα μπορεί να νικήσει. Αν νικήσουν οι χρυσοθήρες τότε παίρνουν από μία κάρτα χρυσού ενώ αν νικήσουν οι σαμποτέρ τις μοιράζονται με τον εξής τρόπο:

- 1 σαμποτέρ παίρνει 4 κάρτες χρυσού



- 2 ή 3 σαμποτέρ παίρνουν από 3 κάρτες χρυσού
- 4 σαμποτέρ παίρνουν από 2 κάρτες χρυσού

Εφόσον ολοκληρωθεί ο γύρος του παιχνιδιού, τότε το παιχνίδι παίζεται για ακόμη 2 γύρους. Ο παίχτης που έχει μαζέψει το περισσότερο χρυσό τότε χρίζεται και νικητής του παιχνιδιού.

#### Συμπέρασμα:

- Διαφορετική τράπουλα από την κλασσική τράπουλα χαρτιών(ειδική)
- Υπάρχουν 4 διαφορετικές τράπουλες στο παιχνίδι: 1) οι αρχικές κάρτες του τραπέζιου, 2) οι κάρτες με τους ρόλους για τους παίχτες, 3) οι κάρτες μονοπατιών και ενεργειών και 4) οι κάρτες χρυσού
- Υπάρχει η τράπουλα όπου οι παίχτες τραβάνε κάρτες
- Υπάρχει στοίβα με τα σκάρτα όπου οι κάρτες δεν χρησιμοποιούνται στο παιχνίδι ξανά
- Υπάρχει ειδικό πλέγμα στο τραπέζι για το πως θα τοποθετούνται οι κάρτες στο τραπέζι
- Στο τραπέζι υπάρχουν ανοιχτές και κλειστές κάρτες
- Ο παίχτης μπορεί να μην έχει το δικαίωμα να τοποθετήσει κάρτες στο τραπέζι άρα πρέπει να έχει ένα μικρό πλέγμα μπροστά του που να δείχνει τις ενέργειες που δεν του επιτρέπονται
- Στο τέλος κάθε γύρου ο GM μοιράζει κάρτες από την τράπουλα του χρυσού στους παίχτες που νίκησαν
- Μέγιστοι παίχτες 10
- Τα μέγιστα χαρτιά που κρατάει στο χέρι του ένας παίχτης είναι 7 και είναι κρυφά

### **3.4 Λειτουργικές απαιτήσεις πλατφόρμας**

Καθώς ολοκληρώθηκαν όλες οι αναλύσεις ως προς τις παρόμοιες πλατφόρμες, τους ρόλους που θα έχουν οι χρήστες και τα πιο δημοφιλή ΠΜΚ, θα αναφερθούν οι απαραίτητες λειτουργικότητες που θα πρέπει να έχει η πλατφόρμα.

Η πλατφόρμα θα πρέπει:

- Να χειρίζεται την εγγραφή, είσοδο και έξοδο του χρήστη στην πλατφόρμα
- Να παρέχει διαφορετικά δικαιώματα σε κάθε χρήστη ανάλογα με τον ρόλο του
- Να εμφανίζει στον χρήστη διαθέσιμα ΗΤ
- Να δημιουργεί ΠΜΚ
- Να δημιουργεί ΗΤ
- Να πραγματοποιεί την είσοδο του χρήστη σε ΗΤ
- Να αλληλοεπιδρούν και να επικοινωνούν οι παίχτες στο ΗΤ σε πραγματικό χρόνο

Πέραν των βασικών αυτών λειτουργικοτήτων που αναφέρθηκαν πιο πάνω, υπάρχουν συγκεκριμένες λειτουργικότητες ως προς το ΗΤ, τον GM και τον GD που είναι αναγκαίο

να συμπεριληφθούν στις κύριες λειτουργικότητες, καθώς ο πυρήνας της πλατφόρμας θα έχει να κάνει με τα ΠΜΚ.

### 3.4.1 Λειτουργικές απαιτήσεις ηλεκτρονικού τραπέζιού

Μετά τα παιχνίδια που αναλύθηκαν αποφασίστηκαν οι παρακάτω λειτουργικές απαιτήσεις του ΗΤ.

- Κάθε ΗΤ θα αποτελείται από τουλάχιστο δύο παίκτες και υποχρεωτικά μόνο ένας από αυτούς θα είναι ο GM, ο δημιουργός του τραπέζιού
- Στο ΗΤ θα εμφανίζεται το όνομα του κάθε χρήστη, ο ρόλος του ή η ομάδα του αν έχει και η κατάσταση του παίχτη αν υπάρχει
- Είναι απαραίτητο να υπάρχει ένα συγκεκριμένο πλέγμα στην οθόνη του ΗΤ που θα περιέχει τις κάρτες του συνδεδεμένου χρήστη
- Οι κάρτες στο χέρι του παίχτη δεν έχουν μέγιστο όριο
- Θα πρέπει να παρέχεται η λίστα με τους παίκτες που είναι συνδεδεμένοι στο τραπέζι και να είναι με την σειρά που θα παίζουν οι παίκτες
- Στο ΗΤ θα πρέπει να προσδιοριστεί χώρος που θα μπορούν να είναι εμφανείς οι κάρτες ή οι τράπουλες του τραπέζιού και οι παίκτες να μπορούν να αλληλοεπιδρούν σε αυτό το σημείο. Οι συνήθεις τράπουλες που έχει ένα παιχνίδι είναι η ενεργή βασική τράπουλα και η τράπουλα με τα σκάρτα
- Ένας παίκτης μπορεί να ρίξει μια κάρτα είτε μπροστά του, είτε στο τραπέζι, είτε σε άλλο παίχτη, είτε να την ξεσκαρτάρει.
- Το τραπέζι μπορεί να έχει ελεύθερο πλέγμα ή ένα συγκεκριμένο πλέγμα που μπορεί να καθοριστεί με συγκεκριμένο αριθμό σε γραμμές και στήλες ή να μην υπάρχει τραπέζι πέρα από τις τράπουλες και τα χέρια των παιχτών.
- Ένας παίκτης μπορεί να ρίξει κάρτα ανοικτή ή κλειστή (χωρίς οι άλλοι να βλέπουν τι συμβολίζει, αλλά να βλέπουν ότι έχει ρίξει)
- Οι παίκτες πέρα από την αλληλοεπίδραση που θα έχουν στο τραπέζι μέσα από τις κάρτες, θα πρέπει επίσης να τους παρέχεται ένας πίνακας ενέργειων που ο χρήστης να προσδιορίζει μία ενέργεια. Για παράδειγμα, δώσε χαρτί, πάσο, UNO, κτλ.
- Με την δήλωση ενέργειας «Παίζω» θα κλειδώνονται οι οποιοσδήποτε ενέργειες των άλλων παιχτών μέχρι να ενημερώσει όλους η πλατφόρμα με την ενέργεια «Τέλος γύρου» και να παίξει ο επόμενος παίχτης. Η ενέργεια «Παίζω» θα υπάρχει λόγω του ότι σε κάποια παιχνίδια δεν παίζει πάντα ο επόμενος παίχτης με βάση την σειρά και έτσι πρέπει να γνωρίζουν οι υπόλοιποι παίκτες ποιος παίχτης παίζει
- Το σύνολο από τις τράπουλες που χρησιμοποιούνται στο παιχνίδι θα πρέπει να είναι εμφανές στο τραπέζι, το ίδιο και το πλήθος των καρτών τους
- Ένας παίκτης μπορεί να μαζέψει όλες τις κάρτες από τραπέζι ή από κάποια τράπουλα σε μία κίνηση

### 3.4.2 Λειτουργικές απαιτήσεις Game Master

Ο ρόλος του GM θεωρείται εξαιρετικά σημαντικός στο HT και θα πρέπει να έχει επιπλέον ενέργειες και δυνατότητες στο τραπέζι ώστε να μπορεί να βοηθήσει στην εξέλιξη και στην ολοκλήρωση του παιχνιδιού. Ο ρόλος του GM θα ανατίθεται ταυτόχρονα στον χρήστη που έχει δημιουργήσει το τραπέζι. Ο GM θα πρέπει:

- Να έχει ένα έξτρα επιπλέον πίνακα ενεργειών
- Να έχει την δυνατότητα να ξεκινάει, να διακόπτει και να σταματάει το παιχνίδι
- Να αναιρεί κινήσεις κατά την διάρκεια του παιχνιδιού
- Να ανακατεύει την τράπουλα ή συγκεκριμένες τράπουλες αν το παιχνίδι χρησιμοποιεί διαφορετικές τράπουλες ταυτόχρονα
- Να μοιράζει τα φύλα της τράπουλας στους παίκτες με ένα πάτημα
- Με την ολοκλήρωση του παιχνιδιού να μπορεί να προβάλει τις κάρτες από τα χέρια των παιχτών ή του τραπεζιού σε λίστα ώστε να καταμετρηθούν για βαθμολογία
- Να ενημερώνει τον πίνακα βαθμολογίας
- Να μπορεί να διώξει παίκτη από το παιχνίδι
- Να αλλάζει την κατάσταση του HT. Για παράδειγμα να θέτει το τραπέζι σε κατάσταση «Αναζήτησης παίκτη» ή «Παιχνίδι σε εξέλιξη»
- Να προσδιορίζει τον χρήστη που παίζει πρώτος αλλά να μπορεί να το κάνει και στην πορεία του παιχνιδιού όταν δεν είναι σαφές ποιου παίκτη είναι η σειρά να παίζει
- Να ορίζει και να επεξεργάζεται τον ρόλο, την ομάδα και την κατάσταση ενός παίκτη
- Να θέτει το τραπέζι σε κατάσταση επεξεργασίας, το οποίο θα είναι ορατό μόνο από τον ίδιο

### 3.4.3 Λειτουργικές απαιτήσεις Game Designer

Για να δημιουργηθεί ένα HT είναι απαραίτητο ο GM να φορτώσει ένα παιχνίδι σε αυτό. Τα παιχνίδια θα κατασκευάζονται από τους GD και θα προϋποθέτουν κάποιες υποχρεωτικές ρυθμίσεις ώστε να είναι σε θέση να δημιουργηθούν. Οι ρυθμίσεις του είναι οι εξής:

- Επιλέγει ή δημιουργεί τράπουλες με κάρτες
- Θέτει και ονομάζει τους διαθέσιμους ρόλους που θα έχουν οι παίκτες
- Ορίζει το μέγιστο αριθμό παιχτών για το παιχνίδι
- Ορίζει το αρχικό μοίρασμα στους παίκτες ή στις άδειες τράπουλες και από ποια τράπουλα θα μοιραστούν (αν υπάρχουν περισσότερες από μια τράπουλες)
- Ορίζει αν το παιχνίδι απαιτεί ειδικό πλέγμα ή όχι
- Ορίζει τις ονομασίες των ομάδων αν υπάρχουν
- Ορίζει νέες καταστάσεις παιχτών αν και εφόσον υπάρχουν περισσότερες καταστάσεις πέραν από την προκαθορισμένη κατάσταση παίκτη, «Ενεργός»

### 3.5 Προδιαγραφές πλατφόρμας

Για την επιθυμητή και ομαλή ανάπτυξη της πλατφόρμας, απαιτείται να καταγράφουν οι προδιαγραφές της πλατφόρμας. Οι προδιαγραφές βοηθάνε στην ορθή επιλογή τεχνολογιών και εργαλείων για την ανάπτυξη της πλατφόρμας αλλά και την χρήση που θα έχουν.

Οι απαιτούμενες προδιαγραφές είναι οι εξής:

- **Σύνδεση στο διαδίκτυο:** Η πλατφόρμα δεν μπορεί να εξυπηρετήσει μόνο ένα χρήστη, χρειάζονται τουλάχιστο δύο χρήστες ώστε να μπορούν να αλληλοεπιδρούν μεταξύ τους και να χρησιμοποιούν τις δυνατότητες της πλατφόρμας. Επομένως είναι απαραίτητη η σύνδεση στο διαδίκτυο για να μπορεί η πλατφόρμα να φιλοξενεί αρκετούς χρήστες και να καλύπτει έτσι και την ανάγκη των απομακρυσμένων χρηστών, να αλληλοεπιδρούν μεταξύ τους σε πραγματικό χρόνο .
- **Επικοινωνία σε πραγματικό χρόνο:** Στο ΗΤ οι κινήσεις των παιχτών θα πρέπει να ενημερώνονται ταυτόχρονα σε όλες τις οθόνες των παιχτών, για αυτό και θα πρέπει να γίνει ειδική ανάπτυξη στην πλατφόρμα ώστε να διασφαλιστεί η άμεση επικοινωνία των παιχτών χωρίς καθυστέρηση.
- **Αυθεντία του χρήστη:** Καθώς θα υπάρχει αλληλοεπίδραση των χρηστών σε πραγματικό χρόνο, θα πρέπει να διασφαλίσουμε την εγκυρότητα του εγγεγραμμένου χρήστη με το προσωπικό του email, ώστε οι χρήστες να μην έχουν πολλαπλούς λογαριασμούς και να υπάρχουν ανεπίτρεπτες συμπεριφορές στις συνομιλίες.
- **Ασφάλεια συστήματος:** Κατά την εγγραφή του εγγεγραμμένου χρήστη, θα αποθηκεύονται στην βάση δεδομένων διάφορα προσωπικά στοιχεία του χρήστη και είναι απαραίτητο να διασφαλιστεί ότι αυτά τα δεδομένα δεν θα καταλήξουν σε λάθος χέρια. Επίσης θα πρέπει να δοθεί η κατάλληλη προσοχή ώστε οι χρήστες να μπορούν να έχουν πρόσβαση μόνο στις σελίδες που τους επιτρέπει ο ρόλος τους.
- **Φιλικό περιβάλλον:** Η πλατφόρμα δεν έχει ελάχιστο ή μέγιστο όριο ηλικίας καθώς τα ΠΜΚ παίζονται από ανθρώπους διαφόρων ηλικιών . Το περιβάλλον της πλατφόρμας επιβάλλεται να είναι φιλικό για όλες τις ηλικίες και να μπορεί να χρησιμοποιηθεί από διαφορετικές συσκευές, όπως τηλέφωνα και ταμπλέτες. Για την επιτυχία του responsive layout θα χρησιμοποιηθεί η βιβλιοθήκη Tailwind CSS όπου μπορεί να αναπτυχθεί παράλληλα διαφορετική σχεδίαση της ιστοσελίδας ανάλογα με το πλάτος της οθόνης σε μονάδα μέτρησης, pixel.
- **Ανέβασμα πολυμέσων εικόνες:** Η πλατφόρμα θα εξυπηρετεί διάφορα ΠΜΚ πέρα από την κλασική τράπουλα με χαρτιά. Ο GM θα έχει την δυνατότητα να ανεβάσει στην πλατφόρμα εικόνες με τις κάρτες που θα είναι μέρος της τράπουλας του παιχνιδιού που θέλει. Για να διασφαλιστεί ότι οι φωτογραφίες δεν θα καταναλώνουν μεγάλο χώρο αποθήκευσης στον διακοσμητή που θα φιλοξενείται η πλατφόρμα, θα πρέπει να γίνεται η κατάλληλη συμπίεση πριν ανέβουν.

### 3.6 Περιπτώσεις χρήσης

Σε αυτή την ενότητα θα αναλυθούν οι περιπτώσεις χρήσης για τον κάθε χρήστη, σε σημαντικές λειτουργίες της πλατφόρμας. Ο προσδιορισμός των περιπτώσεων χρήσης βοηθάει στην κατανόηση όλων των λειτουργιών της πλατφόρμας και καθορίζει την αντιμετώπιση που θα έχει η πλατφόρμα σε κάθε διαφορετικό χρήστη ανάλογα με τον ρόλο του.

Η συγκεκριμένη πλατφόρμα χωρίζεται σε δύο μέρη. Το ένα αφορά τους κύριους ρόλους που θα έχουν οι χρήστες, και αποτελείται από τους ανώνυμους χρήστες, τους εγγεγραμμένους χρήστες και τον διαχειριστή και το άλλο μέρος της πλατφόρμας αφορά το ΗΤ όπου χρήστες αποκτούν ένα νέο ρόλο με την είσοδο τους σαυτό, του GM και των παικτών.

Οι περιπτώσεις χρήσης που θα αναλυθούν για τον κάθε ρόλο είναι οι εξής:

- Εγγραφή του χρήστη στην διαδικτυακή πλατφόρμα
- Είσοδος του χρήστη στην διαδικτυακή πλατφόρμα
- Αποσύνδεση του χρήστη από την διαδικτυακή πλατφόρμα
- Επαναφορά κωδικού πρόσβασης
- Επεξεργασία προσωπικών δεδομένων του χρήστη
- Δημιουργία τράπουλας
- Δημιουργία παιχνιδιού
- Δημιουργία ηλεκτρονικού τραπεζιού
- Είσοδος στον ηλεκτρονικό τραπέζι
- Προβολή και επεξεργασία όλων των χρηστών

#### 3.6.1 Εγγραφή του χρήστη στην διαδικτυακή πλατφόρμα

Ρόλος χρήστη: Ανώνυμος χρήστης

Περιγραφή: Δεν απαιτείται η εγγραφή του χρήστη για αυτό τον ρόλο.

Ρόλος χρήστη: Εγγεγραμμένος χρήστης

Περιγραφή: Ο χρήστης πρέπει να εγγραφεί στην πλατφόρμα με τα προσωπικά του δεδομένα ώστε να είναι τακτοποιημένος χρήστης της πλατφόρμας.

Ροή:

1. Επιλέγει από το μενού της ιστοσελίδας την επιλογή εγγραφή
2. Η πλατφόρμα τον κατευθύνει στην σελίδα εγγραφή χρήστη
3. Ο χρήστης συμπληρώνει τα κατάλληλα πεδία (Ψευδώνυμο, email και κωδικό πρόσβασης)
4. Ο χρήστης επιλέγει το πεδίο εγγραφή

5. Εμφανίζεται μήνυμα επιτυχίας και η πλατφόρμα ενημερώνει τον χρήστη ότι του στάλθηκε email επιβεβαίωσης λογαριασμού
6. Ο χρήστης ανοίγει το email επιβεβαίωσης και επιλέγει το πεδίο επιβεβαίωσης εγγραφής
7. Ο χρήστης κατευθύνεται πίσω στην πλατφόρμα μέσω του email και του εμφανίζει επιτυχές μήνυμα ολοκλήρωσης της επιβεβαίωσης λογαριασμού
8. Η πλατφόρμα εμφανίζει στο χρήστη το πεδίο «Είσοδος» και τον κατευθύνει στην φόρμα εισόδου της πλατφόρμας

#### Εναλλακτική Ροή:

- 5.1. Εμφανίζεται μήνυμα απόρριψης εγγραφής καθώς δεν συμπληρώθηκαν σωστά τα στοιχεία. Η πλατφόρμα ενημερώνει τον χρήστη σε ποια πεδία έχει κάνει λάθος
- 7.1. Ο χρήστης κατευθύνεται πίσω στην πλατφόρμα και του εμφανίζει μήνυμα αποτυχίας καθώς ο χρόνος επιβεβαίωσης έληξε
- 7.2. Ο χρήστης κατευθύνεται πίσω στην πλατφόρμα και του εμφανίζει μήνυμα αποτυχίας καθώς δεν είναι σωστό το token που υπάρχει στο σύνδεσμο

#### Ρόλος χρήστη: Διαχειριστής

Περιγραφή: Στην πλατφόρμα θα υπάρχει μόνο ένας διαχειριστής. Ο ρόλος του διαχειριστή θα δημιουργηθεί από τον προγραμματιστή με εντολή SQL στην βάση δεδομένων.

### **3.6.2 Είσοδος του χρήστη στην διαδικτυακή πλατφόρμα**

#### Ρόλος χρήστη: Ανώνυμος χρήστης

Περιγραφή: Ο χρήστης επιλέγει να εισέλθει στην πλατφόρμα χωρίς να έχει κάνει εγγραφή, με περιορισμένες λειτουργίες.

#### Ροή:

1. Επιλέγει από το μενού της ιστοσελίδας την επιλογή «Προθάλαμος χρήστη»
2. Η πλατφόρμα τον κατευθύνει στην σελίδα και εμφανίζει ένα αναδυόμενο παράθυρο εάν και εφόσον ο υπάρχον χρήστης δεν είναι συνδεδεμένος στην πλατφόρμα
3. Ο χρήστης συμπληρώνει το ψευδώνυμο του και επιβεβαιώνει την φόρμα
4. Η πλατφόρμα εμφανίζει μήνυμα επιτυχίας

#### Εναλλακτική Ροή:

- 3.1 Ο χρήστης δεν συμπληρώνει κάποιο ψευδώνυμο και κλείνει το αναδυόμενο παράθυρο
- 3.2 Ο χρήστης επιλέγει το κουμπί «Προσθήκη ψευδώνυμου» από την σελίδα «Προθάλαμος χρήστη» για να δηλώσει ένα ψευδώνυμο και να του επιτραπεί η είσοδος σε κάποιο ΗΤ
- 4.2 Η πλατφόρμα εμφανίζει μήνυμα αποτυχίας καθώς ο χρήστης δεν συμπλήρωσε έγκυρο ψευδώνυμο ή γιατί το ψευδώνυμο χρησιμοποιείται από κάποιο άλλο χρήστη

### Ρόλος χρήστη: Εγγεγραμμένος χρήστης, Διαχειριστής

Περιγραφή: Ο χρήστης επιλέγει να εισέλθει στην πλατφόρμα ως εγγεγραμμένος χρήστης με τα προσωπικά του στοιχεία.

Ροή:

1. Επιλέγει από το μενού της ιστοσελίδας την επιλογή Είσοδος
2. Η πλατφόρμα τον κατευθύνει στην σελίδα είσοδος χρήστη
3. Ο χρήστης συμπληρώνει τα πεδία email, κωδικό πρόσβασης και «Κράτησε με σε σύνδεση» και ολοκληρώνει την φόρμα με την επιλογή του πεδίου «Είσοδος»
4. Η πλατφόρμα εμφανίζει μήνυμα επιτυχίας και κατευθύνει το χρήστη στην σελίδα που είναι η αίθουσα αναμονής και εύρεσης παιχνιδιού

Εναλλακτική Ροή:

- 4.1. Η πλατφόρμα εμφανίζει μήνυμα αποτυχίας για λάθος συμπλήρωση των προσωπικών του δεδομένων
- 4.2. Η πλατφόρμα εμφανίζει μήνυμα αποτυχίας ότι ο χρήστης δεν υπάρχει στην βάση δεδομένων
- 4.3. Η πλατφόρμα εμφανίζει μήνυμα αποτυχίας για λάθος κωδικό πρόσβασης
- 4.4. Η πλατφόρμα εμφανίζει μήνυμα αποτυχίας γιατί ο χρήστης δεν επιβεβαίωσε το email του. Η πλατφόρμα του στέλνει νέο email επιβεβαίωσης και τον ενημερώνει για την αυτή την διαδικασία

### **3.6.3 Αποσύνδεση του χρήστη από την διαδικτυακή πλατφόρμα**

Ρόλος χρήστη: Ανώνυμος χρήστης

Περιγραφή: Ο ανώνυμος χρήστης θέλει να αποχωρήσει από την πλατφόρμα. Ο χρήστης επειδή δεν είναι εγγεγραμμένος δεν χρειάζεται κάποια ενέργεια εκτός από το να κλείσει την ιστοσελίδα

Ροή:

1. Ο χρήστης κλείνει το παράθυρο της ιστοσελίδας από το πρόγραμμα περιήγησης
2. Η πλατφόρμα κρατάει το ψευδώνυμο του στο χώρο αποθήκευσης του προγράμματος περιήγησης

Ρόλος χρήστη: Εγγεγραμμένος χρήστης, Διαχειριστής

Περιγραφή: Ο εγγεγραμμένος χρήστης θέλει να αποχωρήσει από την πλατφόρμα.

Ροή:

1. Ο χρήστης επιλέγει την επιλογή «Αποσύνδεση» από το μενού της ιστοσελίδας.

2. Η πλατφόρμα αποσυνδέει τον χρήστη και τον κατευθύνει στην αρχική σελίδα

Εναλλακτική Ροή:

- 1.1.Ο χρήστης κλείνει το παράθυρο της ιστοσελίδας από το πρόγραμμα περιήγησης
- 1.2.Αν επαρκεί ο χρόνος του από το JWT Token τότε θα εισέλθει στην πλατφόρμα χωρίς να κάνει είσοδο εφόσον χρησιμοποιήσει το ίδιο πρόγραμμα περιήγησης

### 3.6.4 Επαναφορά κωδικού πρόσβασης

Ρόλος χρήστη: Ανώνυμος χρήστης

Περιγραφή: Ο ανώνυμος χρήστης δεν έχει κωδικό πρόσβασης για να μπορεί να κάνει επαναφορά κωδικού πρόσβασης

Ρόλος χρήστη: Εγγεγραμμένος χρήστης, Διαχειριστής

Περιγραφή: Ο χρήστης έχει ξεχάσει τον κωδικό του πρόσβασης και θέλει να εισέλθει στην πλατφόρμα. Του δίνεται η επιλογή να δημιουργήσει νέο κωδικό και να εισέλθει στην πλατφόρμα

Ροή:

1. Επιλέγει από το μενού της ιστοσελίδας την επιλογή Είσοδος
2. Η πλατφόρμα τον κατευθύνει στην σελίδα είσοδος χρήστη
3. Ο χρήστης επιλέγει την επιλογή «Επαναφορά κωδικού πρόσβασης»
4. Η πλατφόρμα τον κατευθύνει στην σελίδα επαναφορά κωδικού πρόσβασης
5. Ο χρήστης συμπληρώνει το email του και ολοκληρώνει την φόρμα την επιλογή του πεδίου «Υποβολή»
6. Η πλατφόρμα εμφανίζει μήνυμα επιτυχίας και ενημερώνει τον χρήστη ότι του στάλθηκε email για την επαναφορά του κωδικού του
7. Ο χρήστης επιλέγει στο email την επιλογή «Επαναφορά κωδικού πρόσβασης»
8. Η πλατφόρμα κατευθύνει τον χρήστη στην σελίδα δημιουργία νέου κωδικού πρόσβασης
9. Ο χρήστης συμπληρώνει του νέους κωδικούς πρόσβασης
10. Η πλατφόρμα εμφανίζει μήνυμα επιβεβαίωσης και δίνει την επιλογή του χρήστη να εισέλθει στην πλατφόρμα μέσω της επιλογής «Είσοδος»

Εναλλακτική Ροή:

- 6.1. Η πλατφόρμα εμφανίζει μήνυμα αποτυχίας ότι το email καταχώρησης δεν υπάρχει
- 6.2. Η πλατφόρμα εμφανίζει μήνυμα αποτυχίας ότι το email καταχώρησης δεν επιβεβαιώθηκε και η πλατφόρμα στέλνει email επιβεβαίωσης στον χρήστη και τον ενημερώνει
- 8.1. Η πλατφόρμα εμφανίζει μήνυμα αποτυχίας καθώς το token στον σύνδεσμο δεν αντιστοιχεί στην πλατφόρμα



8.2. Η πλατφόρμα εμφανίζει μήνυμα αποτυχίας καθώς έχει λήξει το χρονικό περιθώριο για αλλαγή κωδικού πρόσβασης

10.1 Η πλατφόρμα εμφανίζει μήνυμα αποτυχίας καθώς οι κωδικοί δεν είναι οι ίδιοι ή δεν τηρούν τις προδιαγραφές της πλατφόρμας

### 3.6.5 Επεξεργασία προσωπικών δεδομένων του χρήστη

Ρόλος χρήστη: Ανώνυμος χρήστης

Περιγραφή: Δεν υπάρχουν πληροφορίες στην βάση δεδομένων για τον ανώνυμο χρήστη για να μπορεί να τις επεξεργαστεί

Ρόλος χρήστη: Εγγεγραμμένος χρήστης, Διαχειριστής

Περιγραφή: Ο χρήστης είναι συνδεδεμένος στην πλατφόρμα και επιθυμεί να αλλάξει κωδικό πρόσβασης.

Ροή:

1. Επιλέγει από το μενού της ιστοσελίδας το «Ταμπλό χρήστη»
2. Η πλατφόρμα τον κατευθύνει σε μία νέα σελίδα και του εμφανίζει το ταμπλό του
3. Από το κάθετο μενού επιλέγει τις «Ρυθμίσεις»
4. Επιλέγει το πεδίο αλλαγή κωδικού πρόσβασης
5. Ο χρήστης προσθέτει τον νέο κωδικό πρόσβασης και τον προϋπάρχον κωδικό πρόσβασης
6. Η πλατφόρμα αποθηκεύει τον νέο κωδικό πρόσβασης και ενημερώνει το χρήστη με μήνυμα επιτυχίας

Εναλλακτική Ροή:

- 6.1. Η πλατφόρμα επιστρέφει μήνυμα αποτυχίας καθώς οι δύο κωδικοί δεν είναι οι ίδιοι
- 6.2. Η πλατφόρμα επιστρέφει μήνυμα αποτυχίας καθώς ο κωδικός δεν έχει τους κατάλληλους συνδυασμούς συμβόλων

### 3.6.6 Δημιουργία κάρτας

Ρόλος χρήστη: Ανώνυμος χρήστης

Περιγραφή: Δεν υπάρχει αυτή η δυνατότητα για τον ανώνυμο χρήστη

Ρόλος χρήστη: Εγγεγραμμένος χρήστης, Διαχειριστής

Περιγραφή: Ο χρήστης επιλέγει να δημιουργήσει μία κοινόχρηστη κάρτα για την πλατφόρμα

Ροή:

1. Από το μενού της ιστοσελίδας επιλέγει τον «Προθάλαμο χρήστη»

2. Από το μενού της σελίδας «Προθάλαμος χρήστη» επιλέγει την επιλογή «Δημιουργία κάρτας»
3. Ο χρήστης δίνει ένα όνομα στην κάρτα και θέτει την κάρτα ως κοινόχρηστη.
4. Επιλέγει την εικόνα της κάρτας από τον αποθηκευτικό χώρο του υπολογιστή τύπου jpg, jpeg ή png και καταχωρεί την κάρτα στην πλατφόρμα.
5. Η πλατφόρμα επιστρέφει μήνυμα επιτυχίας στο χρήστη

Εναλλακτική Ροή:

5.1 Η πλατφόρμα εμφανίζει μήνυμα αποτυχίας με το λόγο που απορρίφθηκε η καταχώρηση του χρήστη

### 3.6.7 Δημιουργία τράπουλας

Ρόλος χρήστη: Ανώνυμος χρήστης

Περιγραφή: Δεν υπάρχει αυτή η δυνατότητα για τον ανώνυμο χρήστη

Ρόλος χρήστη: Εγγεγραμμένος χρήστης, Διαχειριστής

Περιγραφή: Ο χρήστης επιλέγει να δημιουργήσει κοινόχρηστη τράπουλα για την πλατφόρμα

Ροή:

1. Από το μενού της ιστοσελίδας επιλέγει τον «Προθάλαμο χρήστη»
2. Από το μενού της σελίδας «Προθάλαμος χρήστη» επιλέγει την επιλογή «Δημιουργία τράπουλας»
3. Ο χρήστης δίνει υποχρεωτικά όνομα για την τράπουλα
4. Ο χρήστης επιλέγει η τράπουλα να είναι κοινόχρηστη
5. Ο χρήστης επιλέγει το πεδίο «Επεξεργασία καρτών»
6. Η πλατφόρμα του εμφανίζει ένα αναδυόμενο παράθυρο με τις κάρτες που ανέβασε ο ίδιος και τις κοινόχρηστες κάρτες της πλατφόρμας
7. Ο χρήστης επιλέγει τις κάρτες που θέλει για την τράπουλα του και κλείνει το αναδυόμενο παράθυρο
8. Ο χρήστης καταχωρεί την τράπουλα
9. Η πλατφόρμα επιστρέφει μήνυμα επιτυχίας

Εναλλακτική Ροή:

6.1. Η πλατφόρμα εμφανίζει μήνυμα αποτυχίας καθώς δεν υπάρχει καμία κάρτα ανεβασμένη στην πλατφόρμα

9.1. Η πλατφόρμα εμφανίζει μήνυμα αποτυχίας

### 3.6.8 Δημιουργία παιχνιδιού

Ρόλος χρήστη: Ανώνυμος χρήστης

Περιγραφή: Στον ανώνυμο χρήστη δεν θα είναι διαθέσιμη η παραπάνω λειτουργικότητα

Ρόλος χρήστη: Εγγεγραμμένος χρήστης, Διαχειριστής

Περιγραφή: Ο χρήστης θέλει να δημιουργήσει παιχνίδι στην πλατφόρμα

Ροή:

1. Από το μενού της ιστοσελίδας επιλέγει τον «Προθάλαμο χρήστη»
2. Από το μενού της σελίδας «Προθάλαμος χρήστη» επιλέγει την επιλογή «Δημιουργία παιχνιδιού»
3. Η πλατφόρμα εμφανίζει την φόρμα που αποτελείται από τρεις (3) σελίδες
4. Ο χρήστης συμπληρώνει τις κατάλληλες ρυθμίσεις (\*Τα πεδία της φόρμας θα αναλυθούν λεπτομερώς στην επομένη παράγραφο)
5. Ο χρήστης επιλέγει το πεδίο «Δημιουργία παιχνιδιού» για ολοκλήρωση της φόρμας
6. Η πλατφόρμα επιστρέφει μήνυμα επιτυχίας

Εναλλακτική Ροή:

6.1. Η πλατφόρμα εμφανίζει μήνυμα αποτυχίας

\*Η δημιουργία του παιχνιδιού θα είναι από τις δύσκολες υλοποιήσεις της πλατφόρμας και αν συμπεριλαμβανόντουσαν στην παραπάνω περίπτωση χρήσης και τα πεδία της φόρμας, θα ήταν τεράστια χωρίς λόγο. Έτσι είναι μία καλή ευκαιρία να αναλυθούν περισσότερο οι επιλογές των πεδίων της συγκεκριμένης φόρμας. Οι επιλογές που θα έχει ο χρήστης σε αυτή την φόρμα θα είναι οι εξής:

- Να υπάρχει ένα μοναδικό όνομα για το παιχνίδι
- Να υπάρχει προαιρετική περιγραφή παιχνιδιού ώστε να προσδιορίζει στους χρήστες τους κανόνες και το πως παίζεται
- Επιλογή τράπουλας ή τράπουλες με κάρτες
- Να επιλέγει το πλέγμα που θα έχει το τραπέζι σε στήλες και γραμμές
- Να επιλέγει αν το παιχνίδι είναι κοινόχρηστο
- Να επιλέγει αν το τραπέζι θα θέτει αυτόματα το επόμενο παίχτη με βάση την σειρά που κάθονται στο τραπέζι οι παίχτες
- Θέτει του μέγιστους παίχτες που μπορούν να παίξουν το παιχνίδι
- Ρόλοι στο παιχνίδι: Δημιουργία των ρόλων των παιχτών που θα υπάρχουν στο παιχνίδι όπως και τα ονόματα τους ώστε να διακρίνονται οι παίχτες μεταξύ τους
- Άδειες τράπουλες: Δημιουργία άδειων τράπουλών που θα είναι διαθέσιμες στο παιχνίδι
- Κατάσταση παιχτών: Αν οι παίχτες θα έχουν κάποιες καταστάσεις πέρα από το ενεργός. Για παράδειγμα στο παιχνίδι saboteur, ο παίχτης μπορεί να μην έχει το δικαίωμα να βάλει φύλλο στο πλέγμα γιατί είναι σπασμένο το καρτόσι του
- Ομάδες: Δημιουργία των ομάδων αν θα υπάρχουν στο παιχνίδι
- Αρχικό μοίρασμα: Θα επιλέγει τον αριθμό των καρτών που θα πάρουν οι παίχτες, το τραπέζι, η στοίβα με τα σκάρτα, οι άδειες τράπουλες και αν θα είναι φανερές οι κρυφές αντίστοιχα

### 3.6.9 Δημιουργία ηλεκτρονικού τραπεζιού

Ρόλος χρήστη: Ανώνυμος χρήστης

Περιγραφή: Δεν θα προσφέρεται αυτή η δυνατότητα σε ανώνυμους χρήστες

Ρόλος χρήστη: Εγγεγραμμένος χρήστης, Διαχειριστής

Περιγραφή: Ένα συνδεδεμένος χρήστης θέλει να δημιουργήσει ένα ιδιωτικό HT και να ξεκινήσει ένα παιχνίδι

Ροή:

1. Από το μενού της ιστοσελίδας επιλέγει τον «Προθάλαμο χρήστη»
2. Από το μενού της σελίδας «Προθάλαμος χρήστη» επιλέγει την επιλογή «Δημιουργία τραπεζιού»
3. Η πλατφόρμα εμφανίζει ένα αναδυόμενο παράθυρο για τις ρυθμίσεις του παιχνιδιού
4. Δίνει όνομα στο τραπέζι
5. Επιλέγει παιχνίδι από τα κοινόχρηστα ή από τα παιχνίδια του
6. Επιλέγει το τραπέζι να είναι ιδιωτικό και θέτει ένα κωδικό πρόσβασης
7. Αφού ολοκληρώσει την εισαγωγή δεδομένων των ρυθμίσεων τότε επιλέγει το πεδίο «Δημιουργία Παιχνιδιού»
8. Η πλατφόρμα τον κατευθύνει στο HT και ο GM αναμένει τους υπόλοιπους παίκτες για να ξεκινήσει το παιχνίδι

Εναλλακτική Ροή:

- 8.1 Η πλατφόρμα εμφανίζει μήνυμα αποτυχίας

### 3.6.10 Είσοδος στο ηλεκτρονικό τραπέζι

Ρόλος χρήστη: Ανώνυμος χρήστης, Εγγεγραμμένος χρήστης, Διαχειριστής

Περιγραφή: Ο χρήστης βρίσκεται στην σελίδα αναμονής και εύρεσης τραπεζιού και επιθυμεί την εισαγωγή του σε ένα HT ώστε να λάβει μέρος σε παιχνίδι

Ροή:

1. Ο χρήστης βλέπει τα διαθέσιμα τραπέζια που ψάχνουν για παίκτες
2. Ο χρήστης επιλέγει το πεδίο «Είσοδος» στο HT που θέλει να εισέλθει
3. Η πλατφόρμα κατευθύνει τον χρήστη στο HT που επέλεξε και ενημερώνει το υπόλοιπο HT και πλατφόρμα

Εναλλακτική Ροή:

- 3.1. Η πλατφόρμα εμφανίζει μήνυμα αποτυχίας καθώς το τραπέζι είναι πλήρες (Ταυτόχρονη είσοδο με άλλο παίκτη)

### 3.6.11 Προβολή και επεξεργασία όλων των χρηστών

Ρόλος χρήστη: Ανώνυμος και εγγεγραμμένος χρήστης

Περιγραφή: Δεν έχουν το δικαίωμα για την προβολή αυτών των στοιχείων

Ρόλος χρήστη: Διαχειριστής

Περιγραφή: Ο διαχειριστής επιθυμεί να προβάλει στην οθόνη όλους τους διαθέσιμους εγγεγραμμένους χρήστες

Ροή:

1. Από το μενού της ιστοσελίδας επιλέγει το «Ταμπλό του διαχειριστή»
2. Η πλατφόρμα τον κατευθύνει στην σελίδα του διαχειριστή
3. Επιλέγει από το μενού που είναι στο πλάι το πεδίο «Προβολή Εγγεγραμμένων Χρηστών»
4. Η πλατφόρμα του εμφανίζει ένα πίνακα με τα στοιχεία των χρηστών

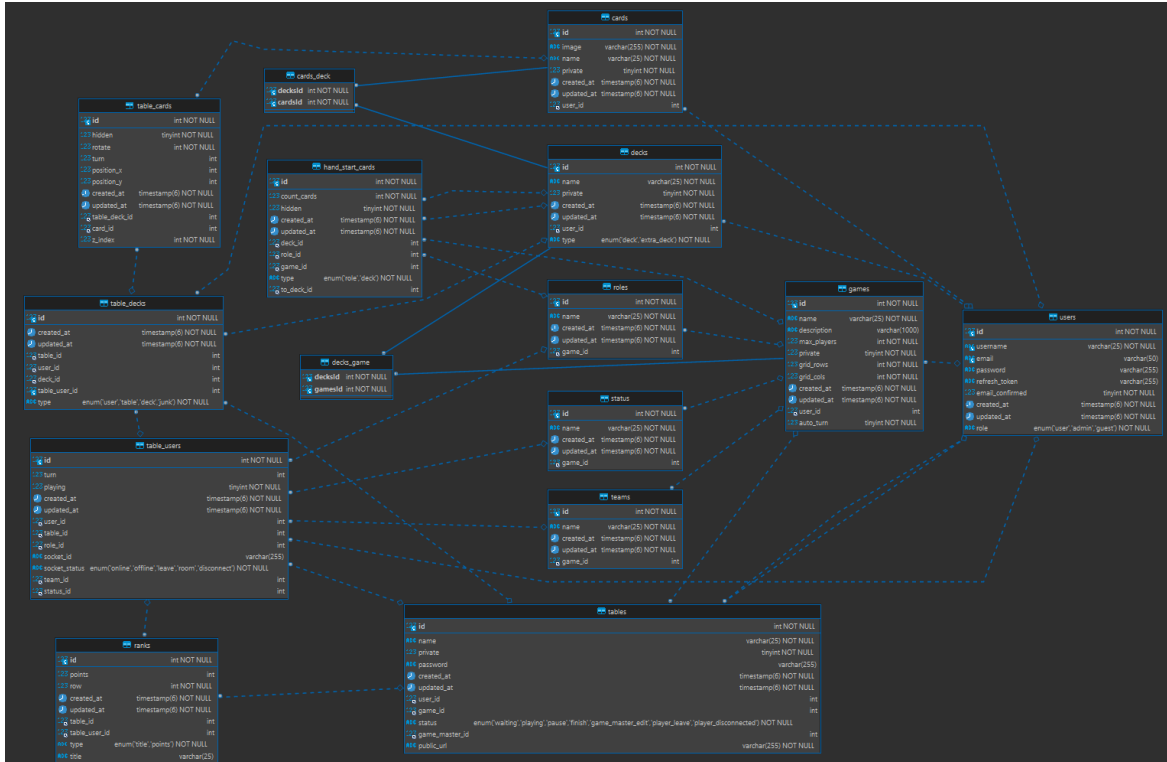
## 3.7 Σχεδιασμός και ανάλυση βάσης δεδομένων

Αφού ολοκληρώθηκε η ανάλυση της πλατφόρμας ως προς τις λειτουργίες της και τον ρόλο που θα έχουν οι χρήστες απομένει ο σχεδιασμός και η ανάλυση της βάσης δεδομένων. Η κύρια τεχνική που χρησιμοποιήθηκε είναι ως προς την σχέση που έχει ένας πίνακας (ή αντικείμενο) με άλλο πίνακα ώστε να σχεδιαστεί μια σχεσιακή βάση δεδομένων.

Για την κατασκευή των πινάκων χρησιμοποιήθηκε η βιβλιοθήκη της nest.js, Type ORM. Οι πίνακες δημιουργούνται από αρχεία κώδικα τα οποία και ορίζουν τα πεδία του κάθε πίνακα, τον τύπο τους καθώς και την σχέση που έχει ο υφιστάμενος πίνακας με ένα άλλο. Αυτή η ΠΕ αποτελείται από 15 πίνακες οι οποίοι είναι:

1. users
2. cards
3. decks
4. cards\_deck
5. games
6. roles
7. status
8. teams
9. hand\_start\_cards
10. decks\_game
11. tables
12. table\_decks
13. table\_cards
14. table\_users
15. ranks

Παρακάτω παρουσιάζεται το διάγραμμα της σχεσιακής βάσης δεδομένων για την καλύτερη αναπαράσταση και κατανόηση της βάσης δεδομένων. Στις επόμενες ενότητες θα αναλυθεί ο κάθε πίνακας ξεχωριστά ως προς την χρήση του και τα πεδία που θα αποτελείται.



Εικόνα 11 - Διάγραμμα σχέσεων οντοτήτων (ER Diagram) βάσης δεδομένων

### 3.7.1 Πίνακας users

Ο πίνακας users περιέχει τους εγγεγραμμένους χρήστες που ανήκουν στην πλατφόρμα μαζί με τα προσωπικά τους στοιχεία, τον ρόλο τους και το αναγνωριστικό κλειδί για την αλληλοεπίδραση του με την πλατφόρμα.

users	
123 id	int NOT NULL
ABC username	varchar(25) NOT NULL
ABC email	varchar(50)
ABC password	varchar(255)
ABC refresh_token	varchar(255)
123 email_confirmed	tinyint NOT NULL
🕒 created_at	timestamp(6) NOT NULL
🕒 updated_at	timestamp(6) NOT NULL
ABC role	enum('user','admin','guest') NOT NULL

Εικόνα 12 - Πίνακας users

Ανάλυση πεδίων του πίνακα:

- **id:** Δηλώνει το αναγνωριστικό κωδικό του χρήστη. Αποτελεί το πρωτεύων κλειδί (primary key) του πίνακα και ο τύπος του είναι ακέραιος (int).
- **username:** Δηλώνει το όνομα του χρήστη στην πλατφόρμα. Ο τύπος του πεδίου είναι αλφαριθμητικό (varchar) με μέγεθος 25 χαρακτήρες και η τιμή που παίρνει το πεδίο είναι μοναδική (unique).
- **email:** Δηλώνει το ηλεκτρονικό ταχυδρομείο του χρήστη στην πλατφόρμα. Ο τύπος του πεδίου είναι αλφαριθμητικό (varchar) με μέγεθος 50 χαρακτήρες και η τιμή που παίρνει το πεδίο είναι μοναδική (unique). Η τιμή του πεδίου μπορεί να παραμείνει κενή (nullable) όταν ο χρήστης είναι ανώνυμος.
- **password:** Δηλώνει το κρυπτογραφημένο (hash) κωδικό πρόσβασης του χρήστη και ο τύπος του είναι αλφαριθμητικό (varchar) και η τιμή του πεδίου μπορεί να παραμείνει κενή (nullable).
- **role:** Δηλώνει το ρόλο που έχει ο εγγεγραμμένος χρήστης στην πλατφόρμα και είναι τύπου αντικειμένου αλφαριθμητικό (ENUM [29]). Ένα αντικείμενο τύπου ENUM αποτελείται από μία λίστα επιτρεπόμενων τιμών που απαριθμούνται στην προδιαγραφή της στήλης κατά την δημιουργία του πίνακα. Στους εγγεγραμμένους χρήστες οι ρόλοι αποτελούνται από τους `user` και τον διαχειριστή (`admin`) με προκαθορισμένη (default) τιμή `“user”`. Οι ανώνυμοι χρήστες θα έχουν την τιμή `“guest”`.
- **email\_confirmed:** Δηλώνει αν ο χρήστης έχει επιβεβαιώσει το ηλεκτρονικό του ταχυδρομείο με την εγγραφή του στην πλατφόρμα. Ο τύπος του πεδίου είναι boolean που δέχεται μόνο δύο τιμές, αλήθεια ή ψευδές (true or false) με προκαθορισμένη τιμή το false.
- **refresh\_token:** Δηλώνει την κρυπτογραφημένη (hash) συμβολοσειρά του χρήστη που χρησιμοποιείται σε κάθε αίτημα του χρήστη ως προς την εξακρίβωση της ταυτότητας του προς την βάση δεδομένων. Ο τύπος του είναι αλφαριθμητικό (varchar) και η τιμή του πεδίου μπορεί να παραμείνει κενή (nullable).
- **created\_at:** Δηλώνει την ημερομηνία και την ώρα που δημιουργήθηκε η εγγραφή του χρήστη. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime) και καταγράφεται αυτόματα με την εγγραφή του χρήστη μέσω της συνάρτησης NOW().
- **updated\_at:** Δηλώνει την ημερομηνία και την ώρα που τροποποιήθηκε η εγγραφή του χρήστη. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime) και καταγράφεται αυτόματα με την τροποποίηση ή διαγραφή δεδομένων του χρήστη μέσω της συνάρτησης NOW().

#### Ανάλυση σχέσεων με άλλους πίνακες:

- **cards:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα cards με όνομα user\_id
- **decks:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα decks με όνομα user\_id
- **games:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα games με όνομα user\_id
- **tables:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα cards με όνομα user\_id και game\_master\_id
- **table\_decks:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_decks με όνομα user\_id
- **table\_users:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_users με όνομα user\_id

Κώδικας για δημιουργία πίνακα και ορισμό σχέσεων με άλλους πίνακες σε γλώσσα nest.js:

```
@Entity('users')
export class UsersEntity {
  @PrimaryGeneratedColumn()
  id: number;

  @Column({ unique: true, length: 25 })
  username: string;

  @Column({ unique: true, length: 50, nullable: true })
  email: string;

  @Column({ select: false, nullable: true })
  password: string;

  @Column({ nullable: true })
  refresh_token: string;

  @Column({ default: false })
  email_confirmed: boolean;
}
```



```
@CreateDateColumn({ type: 'timestamp', default: () =>
"CURRENT_TIMESTAMP(6)" })
created_at: Date;

@UpdateDateColumn({ type: "timestamp", default: () =>
"CURRENT_TIMESTAMP(6)", onUpdate: "CURRENT_TIMESTAMP(6)" })
updated_at: Date;

@Column({ type: 'enum', enum: Role, default: Role.USER })
role: Role;

@OneToMany(() => CardsEntity, (cardsEntity) => cardsEntity.creator)
cards: CardsEntity

@OneToMany(() => DecksEntity, (decksEntity) => decksEntity.creator)
decks: DecksEntity

@OneToMany(() => GamesEntity, (gamesEntity) => gamesEntity.creator)
game_id: GamesEntity

@OneToMany(() => TablesEntity, (tablesEntity) => tablesEntity.creator)
tables: GamesEntity

@OneToMany(() => TablesDecksEntity, (tablesDecksEntity) =>
tablesDecksEntity.user)
table_decks: TablesDecksEntity

@OneToMany(() => TableUsersEntity, (tableUsersEntity) =>
tableUsersEntity.user, { cascade: true })
table_users_id: TableUsersEntity
```

```
@OneToMany(() => TablesEntity, (tablesEntity) =>
tablesEntity.game_master)
  tables_game_master: TablesEntity
}
table_users_id: TableUsersEntity

@OneToMany(() => RankEntity, (rankEntity) => rankEntity.user_id)
  ranks: RankEntity
}
```

### 3.7.2 Πίνακας cards

Ο πίνακας cards περιέχει τις κάρτες του παιχνιδιού που μπορούν να χρησιμοποιήσουν οι χρήστες για την δημιουργία μίας τράπουλας. Στην πλατφόρμα θα δίνεται η δυνατότητα να δημιουργήσουν κάρτες μόνο οι εγγεγραμμένοι χρήστες.

Παρακάτω παρουσιάζεται η εικόνα από τον πίνακα του διαγράμματος.

Column Name	Data Type	Constraints
id	int	NOT NULL, Primary Key
image	varchar(255)	NOT NULL
name	varchar(25)	NOT NULL
private	tinyint	NOT NULL
created_at	timestamp(6)	NOT NULL
updated_at	timestamp(6)	NOT NULL
user_id	int	

Εικόνα 13 - Πίνακας cards

#### Ανάλυση πεδίων του πίνακα:

- **id:** Δηλώνει το αναγνωριστικό κωδικό της κάρτας. Αποτελεί το πρωτεύων κλειδί (primary key) του πίνακα και ο τύπος του είναι ακέραιος (int).
- **image:** Δηλώνει το όνομα και τον τύπο του αρχείου της εικόνας ώστε με την προσθήκη της διεύθυνσης του backend, η εικόνα να εμφανίζεται στο πρόγραμμα περιήγησης. Ο τύπος του πεδίου είναι αλφαριθμητικό (varchar) με μέγεθος 255 χαρακτήρες.
- **name:** Δηλώνει το όνομα της κάρτας. Ο τύπος του πεδίου είναι αλφαριθμητικό (varchar) με μέγεθος 25 χαρακτήρες.
- **private:** Δηλώνει αν η κάρτα είναι διαθέσιμη και σε άλλους χρήστες ή μόνο στον δημιουργό της. Ο τύπος του πεδίου είναι boolean με προκαθορισμένη τιμή το ψέμα (false).

- **created\_at:** Δηλώνει την ημερομηνία και την ώρα που δημιουργήθηκε η κάρτα. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **updated\_at:** Δηλώνει την ημερομηνία και την ώρα που τροποποιήθηκε η εγγραφή του χρήστη. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **user\_id:** Δηλώνει το αναγνωριστικό κωδικό του χρήστη που το δημιούργησε. Αποτελεί ξένο κλειδί στον πίνακα users και ο τύπος του είναι ακέραιος (int).

#### Ανάλυση σχέσεων με άλλους πίνακες:

- **users:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα cards με όνομα user\_id
- **table\_cards:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_cards με όνομα card\_id
- **decks:** Έχει σχέση πολλά προς πολλά και δημιουργείται ένας νέος πίνακα με όνομα cards\_deck

Κώδικας για δημιουργία πίνακα και ορισμό σχέσεων με άλλους πίνακες σε γλώσσα nest.js:

```
@Entity('cards')
export class CardsEntity {
  @PrimaryGeneratedColumn()
  id: number;

  @Column()
  image: string;

  @Column({ length: 25 })
  name: string;

  @Column({ default: false })
  private: boolean;

  @CreateDateColumn({ type: 'timestamp', default: () =>
"CURRENT_TIMESTAMP(6)" })
  created_at: Date;
```

```

@UpdateDateColumn({ type: "timestamp", default: () =>
"CURRENT_TIMESTAMP(6)", onUpdate: "CURRENT_TIMESTAMP(6)" })
updated_at: Date;

@ManyToMany(() => DecksEntity, (decksEntity) => decksEntity.cards)
deck: DecksEntity[]

@OneToMany(() => TablesCardsEntity, (tablesCardsEntity) =>
tablesCardsEntity.card_id)
table_cards_id: TablesCardsEntity

@ManyToOne(() => UsersEntity, (usersEntity) => usersEntity.cards, {
onDelete: 'SET NULL' })
@JoinColumn({ name: 'user_id' })
creator: UsersEntity
}

```

### 3.7.3 Πίνακας decks

Ο πίνακας decks περιέχει τις τράπουλες της πλατφόρμας που μπορούν να χρησιμοποιηθούν στα παιχνίδια. Τράπουλα θα μπορούν να δημιουργήσουν οι εγγεγραμμένοι χρήστες της πλατφόρμας.

Παρακάτω παρουσιάζεται η εικόνα από τον πίνακα του διαγράμματος.

decks	
id	int NOT NULL
name	varchar(25) NOT NULL
private	tinyint NOT NULL
created_at	timestamp(6) NOT NULL
updated_at	timestamp(6) NOT NULL
user_id	int
type	enum('deck','extra_deck') NOT NULL

Εικόνα 14 - Πίνακας decks

#### Ανάλυση πεδίων του πίνακα:

- **id:** Δηλώνει το αναγνωριστικό κωδικό της τράπουλας. Αποτελεί το πρωτεύον κλειδί (primary key) του πίνακα και ο τύπος του είναι ακέραιος (int).
- **name:** Δηλώνει το όνομα της τράπουλας. Ο τύπος του πεδίου είναι αλφαριθμητικό (varchar) με μέγεθος 25 χαρακτήρες.
- **private:** Δηλώνει αν η κάρτα είναι διαθέσιμη και σε άλλους χρήστες ή μόνο στον δημιουργό της. Ο τύπος του πεδίου είναι boolean με προκαθορισμένη τιμή το ψέμα (false).
- **type:** Δηλώνει το τύπο της τράπουλας και είναι τύπου αντικειμένου αλφαριθμητικό (ENUM). Δέχεται δύο τιμές – “deck”, “extra\_deck” – με προκαθορισμένη τιμή το “deck”.
- **created\_at:** Δηλώνει την ημερομηνία και την ώρα που δημιουργήθηκε η τράπουλα. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **updated\_at:** Δηλώνει την ημερομηνία και την ώρα που τροποποιήθηκε η τράπουλα. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **user\_id:** Δηλώνει το αναγνωριστικό κωδικό του χρήστη που το δημιούργησε. Αποτελεί ξένο κλειδί στον πίνακα users και ο τύπος του είναι ακέραιος (int).

#### Ανάλυση σχέσεων με άλλους πίνακες:

- **users:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα decks με όνομα user\_id
- **cards:** Έχει σχέση πολλά προς πολλά και δημιουργείται ένας νέος πίνακα με όνομα cards\_deck
- **hand\_start\_cards:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα hand\_start\_cards με όνομα deck\_id
- **games:** Έχει σχέση πολλά προς πολλά και δημιουργείται ένας νέος πίνακας με το όνομα decks\_game
- **tables\_decks:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_decks με όνομα deck\_id

#### Κώδικας για δημιουργία πίνακα και ορισμό σχέσεων με άλλους πίνακες σε γλώσσα nest.js:

```
@Entity('decks')
export class DecksEntity {
  @PrimaryGeneratedColumn()
  id: number;

  @Column({ length: 25 })
  name: string;
```

```
@Column({ default: false })
private: boolean;

@Column({ type: 'enum', enum: DeckType, default: DeckType.DECK })
type: DeckType

@CreateDateColumn({ type: 'timestamp', default: () =>
"CURRENT_TIMESTAMP(6)" })
created_at: Date;

@UpdateDateColumn({ type: "timestamp", default: () =>
"CURRENT_TIMESTAMP(6)", onUpdate: "CURRENT_TIMESTAMP(6)" })
updated_at: Date;

@ManyToOne(() => UsersEntity, (userEntity) => userEntity.decks, {
onDelete: 'SET NULL' })
@JoinColumn({ name: 'user_id' })
creator: UsersEntity

@ManyToMany(() => CardsEntity, (cardsEntity) => cardsEntity.deck)
@JoinTable({ name: 'cards_deck' })
cards: CardsEntity[]

@OneToMany(() => HandStartCardsEntity, (handStartCardsEntity) =>
handStartCardsEntity.deck)
hand_start_cards: HandStartCardsEntity

@OneToMany(() => HandStartCardsEntity, (handStartCardsEntity) =>
handStartCardsEntity.toDeck)
hand_start_cards_to_deck: HandStartCardsEntity
```

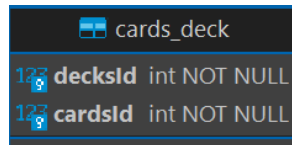
```
@ManyToMany(() => GamesEntity, (gamesEntity) => gamesEntity.deck)
@JoinTable({ name: 'decks_game' })
games: GamesEntity[]

@OneToMany(() => TablesDecksEntity, (tablesDecksEntity) =>
tablesDecksEntity.deck)
table_deck_id: TablesDecksEntity;
}
```

### 3.7.4 Πίνακας cards\_deck

Με την δημιουργία της σχέσης πολλά προς πολλά μεταξύ του πίνακα cards και decks, δημιουργείται αυτόματα στην βάση δεδομένων ένας νέος πίνακας με όνομα cards\_deck. Αυτός ο πίνακας προσδιορίζει τις κάρτες που έχει μία τράπουλα. Όταν προθέτονται κάρτες σε μία τράπουλα τότε ενημερώνεται ο συγκεκριμένος πίνακας.

Στην παρακάτω εικόνα παρουσιάζονται τα πεδία του πίνακα.



cards_deck
deckId int NOT NULL
cardId int NOT NULL

Εικόνα 15 - Πίνακας cards\_deck

#### Ανάλυση πεδίων του πίνακα:

- **deckId:** Δηλώνει το αναγνωριστικό κωδικό της τράπουλας. Αποτελεί το ξένο κλειδί (foreign key) του πίνακα decks και ο τύπος του είναι ακέραιος (int).
- **cardId:** Δηλώνει το αναγνωριστικό κωδικό της κάρτας. Αποτελεί το ξένο κλειδί (foreign key) του πίνακα cards και ο τύπος του είναι ακέραιος (int).

### 3.7.5 Πίνακας games

Ο πίνακας games περιέχει όλα τα παιχνίδια της πλατφόρμας που είναι διαθέσιμα προς του χρήστες ώστε να τα φορτώσουν στο ΗΤ. Για την δημιουργία ενός παιχνιδιού χρειάζονται αρκετά δεδομένα και κρίθηκε ως καλύτερη υλοποίηση, κάποια δεδομένα να υπάρχουν σε άλλους νέους πίνακες που συσχετίζονται με τον υφιστάμενο πίνακα.

Στην παρακάτω εικόνα παρουσιάζονται τα πεδία του πίνακα με τα κύρια χαρακτηριστικά:

games	
123 id	int NOT NULL
ABC name	varchar(25) NOT NULL
ABC description	varchar(1000)
123 max_players	int NOT NULL
123 private	tinyint NOT NULL
123 grid_rows	int NOT NULL
123 grid_cols	int NOT NULL
↓ created_at	timestamp(6) NOT NULL
↓ updated_at	timestamp(6) NOT NULL
123 user_id	int
123 auto_turn	tinyint NOT NULL

Εικόνα 16 - Πίνακας games

Ανάλυση πεδίων του πίνακα:

- **id:** Δηλώνει το αναγνωριστικό κωδικό του παιχνιδιού. Αποτελεί το πρωτεύων κλειδί (primary key) του πίνακα και ο τύπος του είναι ακέραιος (int).
- **name:** Δηλώνει το όνομα το παιχνιδιού. Ο τύπος του πεδίου είναι αλφαριθμητικό (varchar) με μέγεθος 25 χαρακτήρες.
- **description:** Δηλώνει την περιγραφή του παιχνιδιού ώστε οι χρήστες να γνωρίζουν τις οδηγίες και του κανόνες του παιχνιδιού. Ο τύπος του πεδίου είναι αλφαριθμητικό (varchar) με μέγεθος 1000 χαρακτήρες και η τιμή του πεδίου μπορεί να παραμείνει κενή (nullable).
- **max\_players:** Δηλώνει το μέγιστο αριθμό παικτών που θα έχει το παιχνίδι. Ο τύπος του πεδίου είναι ακέραιος (int).
- **auto\_rank:** Δηλώνει αν το παιχνίδι θα θέτει αυτόματα τον επόμενο παίχτη όταν ο χρήστης που παίζει τερματίσει τον γύρο του. Ο τύπος του πεδίου είναι boolean με προκαθορισμένη τιμή το ψέμα (false).
- **private:** Δηλώνει αν το παιχνίδι είναι ελεύθερο για χρήση και από άλλους χρήστες στα HT. Ο τύπος του πεδίου είναι boolean με προκαθορισμένη τιμή το ψέμα (false).
- **grid\_rows:** Δηλώνει σε πόσες γραμμές θα είναι χωρισμένο το HT. Ο τύπος του πεδίου είναι ακέραιος (int).
- **grid\_cols:** Δηλώνει σε πόσες στήλες θα είναι χωρισμένο το HT. Ο τύπος του πεδίου είναι ακέραιος (int).
- **created\_at:** Δηλώνει την ημερομηνία και την ώρα που δημιουργήθηκε το παιχνίδι. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **updated\_at:** Δηλώνει την ημερομηνία και την ώρα που τροποποιήθηκε το παιχνίδι. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **user\_id:** Δηλώνει το αναγνωριστικό κωδικό του χρήστη που δημιούργησε το παιχνίδι. Αποτελεί ξένο κλειδί στον πίνακα users και ο τύπος του είναι ακέραιος (int).



#### Ανάλυση σχέσεων με άλλους πίνακες:

- roles: Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα roles με όνομα game\_id
- hand\_start\_cards: Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα hand\_start\_cards με όνομα game\_id
- decks: Έχει σχέση πολλά προς πολλά και δημιουργείται ένας νέος πίνακα με όνομα decks\_game
- users: Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα games με όνομα user\_id
- tables: Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα tables με όνομα game\_id
- status: Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα status με όνομα game\_id
- teams: Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα teams με όνομα game\_id

Κώδικας για δημιουργία πίνακα και ορισμό σχέσεων με άλλους πίνακες σε γλώσσα nest.js:

```
@Entity('games')
export class GamesEntity {
  @PrimaryGeneratedColumn()
  id: number;

  @Column({ length: 25 })
  name: string;

  @Column({ length: 1000 , nullable: true })
  description: string;

  @Column()
  max_players: number;

  @Column({ default: false })
  auto_turn: boolean;
```

```
@Column({ default: false })
private: boolean;

@Column()
grid_rows: number;

@Column()
grid_cols: number;

@CreateDateColumn({ type: 'timestamp', default: () =>
"CURRENT_TIMESTAMP(6)" })
created_at: Date;

@UpdateDateColumn({ type: "timestamp", default: () =>
"CURRENT_TIMESTAMP(6)", onUpdate: "CURRENT_TIMESTAMP(6)" })
updated_at: Date;

@OneToMany(() => RolesEntity, (rolesEntity) => rolesEntity.game, {
cascade: true })
roles: RolesEntity[]

@OneToMany(() => HandStartCardsEntity, (handStartCardsEntity) =>
handStartCardsEntity.game, { cascade: true })
hand_start_cards: HandStartCardsEntity[]

@ManyToMany(() => DecksEntity, (decksEntity) => decksEntity.games, {
onDelete: 'CASCADE' })
deck: DecksEntity[]

@ManyToOne(() => UsersEntity, (usersEntity) => usersEntity.id, {
onDelete: 'SET NULL' })

@JoinColumn({ name: 'user_id' })
```

```

creator: UsersEntity

@OneToMany(() => TablesEntity, (tablesEntity) => tablesEntity.game)
table_id: TablesEntity

@OneToMany(() => StatusEntity, (statusEntity) => statusEntity.game, {
cascade: true })
status: StatusEntity[]

@OneToMany(() => TeamsEntity, (teamsEntity) => teamsEntity.game, {
cascade: true })
teams: TeamsEntity[]
}

```

### 3.7.6 Πίνακας roles

Ο πίνακας roles είναι μέρος της δημιουργίας του παιχνιδιού. Αν ο GD επιλέξει την επιλογή ώστε να υπάρχουν περισσότεροι ρόλοι στο παιχνίδι από τους προκαθορισμένους τότε θα δημιουργούνται οι ρόλοι του παιχνιδιού σε αυτό τον πίνακα.

Η παρακάτω εικόνα προσδιορίζει τα στοιχεία του πίνακα από το διάγραμμα της βάσης δεδομένων.

roles	
id	int NOT NULL
name	varchar(25) NOT NULL
created_at	timestamp(6) NOT NULL
updated_at	timestamp(6) NOT NULL
game_id	int

Εικόνα 17 - Πίνακας roles

#### Ανάλυση πεδίων του πίνακα:

- **id:** Δηλώνει το αναγνωριστικό κωδικό του ρόλου. Αποτελεί το πρωτεύων κλειδί (primary key) του πίνακα και ο τύπος του είναι ακέραιος (int).
- **name:** Δηλώνει το όνομα του ρόλου. Ο τύπος του πεδίου είναι αλφαριθμητικό (varchar) με μέγεθος 25 χαρακτήρες.
- **created\_at:** Δηλώνει την ημερομηνία και την ώρα που δημιουργήθηκε ο ρόλος. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).

- **updated\_at:** Δηλώνει την ημερομηνία και την ώρα που τροποποιήθηκε ο ρόλος. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **game\_id:** Δηλώνει το αναγνωριστικό κωδικό του παιχνιδιού. Αποτελεί ξένο κλειδί στον πίνακα games και ο τύπος του είναι ακέραιος (int).

#### Ανάλυση σχέσεων με άλλους πίνακες:

- **games:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα roles με όνομα game\_id
- **hand\_start\_cards:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα hand\_start\_cards με όνομα role\_id
- **table\_users:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_users με όνομα role\_id

#### Κώδικας για δημιουργία πίνακα και ορισμό σχέσεων με άλλους πίνακες σε γλώσσα nest.js:

```
@Entity('roles')
export class RolesEntity {
  @PrimaryGeneratedColumn()
  id: number;

  @Column({ length: 25 })
  name: string;

  @CreateDateColumn({ type: 'timestamp', default: () =>
"CURRENT_TIMESTAMP(6)" })
  created_at: Date;

  @UpdateDateColumn({ type: "timestamp", default: () =>
"CURRENT_TIMESTAMP(6)", onUpdate: "CURRENT_TIMESTAMP(6)" })
  updated_at: Date;

  @ManyToOne(() => GamesEntity, (gamesEntity) => gamesEntity.roles, {
onDelete: 'CASCADE' })
  @JoinColumn({ name: 'game_id' })
  game: GamesEntity
```

```
@OneToMany(() => HandStartCardsEntity, (handStartCardsEntity) =>
handStartCardsEntity.role, { onDelete: 'CASCADE' })
hand_start_deck_id: HandStartCardsEntity

@OneToMany(() => TableUsersEntity, (tableUsersEntity) =>
tableUsersEntity.role)
table_user_id: TableUsersEntity;
}
```

### 3.7.7 Πίνακας status

Ο πίνακας status ανήκει στην διαδικασία της δημιουργίας του παιχνιδιού. Εφόσον ο GD επιλέξει να προσθέσει νέες καταστάσεις στους παίχτες, τότε οι νέες καταστάσεις αποθηκεύονται στον πίνακα status, με την κατάλληλη συσχέτιση τους με τον πίνακα games.

Η παρακάτω εικόνα προσδιορίζει τα στοιχεία του πίνακα από το διάγραμμα της βάσης δεδομένων.

Column	Data Type
id	int NOT NULL
name	varchar(25) NOT NULL
created_at	timestamp(6) NOT NULL
updated_at	timestamp(6) NOT NULL
game_id	int

Εικόνα 18 - Πίνακας status

#### Ανάλυση πεδίων του πίνακα:

- **id:** Δηλώνει το αναγνωριστικό κωδικό της κατάστασης. Αποτελεί το πρωτεύων κλειδί (primary key) του πίνακα και ο τύπος του είναι ακέραιος (int).
- **name:** Δηλώνει το όνομα της κατάστασης. Ο τύπος του πεδίου είναι αλφαριθμητικό (varchar) με μέγεθος 25 χαρακτήρες.
- **created\_at:** Δηλώνει την ημερομηνία και την ώρα που δημιουργήθηκε η κατάσταση παίχτη. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **updated\_at:** Δηλώνει την ημερομηνία και την ώρα που τροποποιήθηκε η κατάσταση παίχτη. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **game\_id:** Δηλώνει το αναγνωριστικό κωδικό του παιχνιδιού. Αποτελεί ξένο κλειδί στον πίνακα games και ο τύπος του είναι ακέραιος (int).

Ανάλυση σχέσεων με άλλους πίνακες:

- **games:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα status με όνομα game\_id
- **table\_users:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_users με όνομα status\_id

Κώδικας για δημιουργία πίνακα και ορισμό σχέσεων με άλλους πίνακες σε γλώσσα nest.js:

```
@Entity('status')
export class StatusEntity {
  @PrimaryGeneratedColumn()
  id: number;

  @Column({ length: 25 })
  name: string;

  @CreateDateColumn({ type: 'timestamp', default: () =>
"CURRENT_TIMESTAMP(6)" })
  created_at: Date;

  @UpdateDateColumn({ type: "timestamp", default: () =>
"CURRENT_TIMESTAMP(6)", onUpdate: "CURRENT_TIMESTAMP(6)" })
  updated_at: Date;

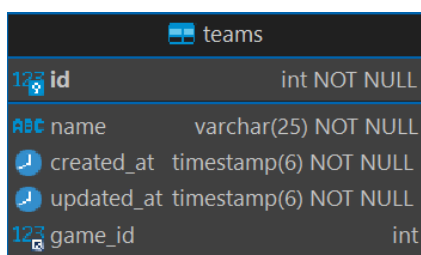
  @ManyToOne(() => GamesEntity, (gamesEntity) => gamesEntity.status, {
onDelete: 'CASCADE' })
  @JoinColumn({ name: 'game_id' })
  game: GamesEntity

  @OneToMany(() => TableUsersEntity, (tableUsersEntity) =>
tableUsersEntity.status)
  table_user: TableUsersEntity
}
```

### 3.7.8 Πίνακας teams

Ο πίνακας teams είναι και αυτός επέκταση της δημιουργίας παιχνιδιού. Αν ο GD επιλέξει ότι οι παίκτες θα χωρίζονται σε ομάδες τότε τα στοιχεία της ομάδας θα αποθηκεύονται σε αυτό το πίνακα.

Η παρακάτω εικόνα προσδιορίζει τα στοιχεία του πίνακα από το διάγραμμα της βάσης δεδομένων.



teams	
id	int NOT NULL
name	varchar(25) NOT NULL
created_at	timestamp(6) NOT NULL
updated_at	timestamp(6) NOT NULL
game_id	int

Εικόνα 19 - Πίνακας teams

#### Ανάλυση πεδίων του πίνακα:

- **id:** Δηλώνει το αναγνωριστικό κωδικό της ομάδας. Αποτελεί το πρωτεύων κλειδί (primary key) του πίνακα και ο τύπος του είναι ακέραιος (int).
- **name:** Δηλώνει το όνομα της ομάδας. Ο τύπος του πεδίου είναι αλφαριθμητικό (varchar) με μέγεθος 25 χαρακτήρες.
- **created\_at:** Δηλώνει την ημερομηνία και την ώρα που δημιουργήθηκε η ομάδα. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **updated\_at:** Δηλώνει την ημερομηνία και την ώρα που τροποποιήθηκε η ομάδα. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **game\_id:** Δηλώνει το αναγνωριστικό κωδικό του παιχνιδιού. Αποτελεί ξένο κλειδί στον πίνακα games και ο τύπος του είναι ακέραιος (int).

#### Ανάλυση σχέσεων με άλλους πίνακες:

- **games:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα teams με όνομα game\_id
- **table\_users:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_users με όνομα team\_id

Κώδικας για δημιουργία πίνακα και ορισμό σχέσεων με άλλους πίνακες σε γλώσσα nest.js:

```
@Entity('teams')
export class TeamsEntity {
  @PrimaryGeneratedColumn()
  id: number;

  @Column({ length: 25 })
  name: string;

  @CreateDateColumn({ type: 'timestamp', default: () =>
"CURRENT_TIMESTAMP(6)" })
  created_at: Date;

  @UpdateDateColumn({ type: "timestamp", default: () =>
"CURRENT_TIMESTAMP(6)", onUpdate: "CURRENT_TIMESTAMP(6)" })
  updated_at: Date;

  @ManyToOne(() => GamesEntity, (gamesEntity) => gamesEntity.teams, {
onDelete: 'CASCADE' })
  @JoinColumn({ name: 'game_id' })
  game: GamesEntity

  @OneToMany(() => TableUsersEntity, (tableUsersEntity) =>
tableUsersEntity.team)
  table_user: TableUsersEntity
}
```

### 3.7.9 Πίνακας hand\_start\_cards

Ο πίνακας hand\_start\_cards αφορά τις ρυθμίσεις που θα θέσει ο GD ώστε να είναι σε θέση ο GM να δώσει τις κάρτες στους παίκτες. Ο GD θα θέτει πόσες κάρτες θα πάρει ο κάθε ρόλος ή το τραπέζι, αν θα είναι κρυφές και επίσης από ποια τράπουλα του παιχνιδιού θα τις πάρει .



Η παρακάτω εικόνα προσδιορίζει τα στοιχεία του πίνακα από το διάγραμμα της βάσης δεδομένων.

hand_start_cards	
id	int NOT NULL
count_cards	int NOT NULL
hidden	tinyint NOT NULL
created_at	timestamp(6) NOT NULL
updated_at	timestamp(6) NOT NULL
deck_id	int
role_id	int
game_id	int
type	enum('role','deck') NOT NULL
to_deck_id	int

Εικόνα 20 - Πίνακας hand\_start\_cards

#### Ανάλυση πεδίων του πίνακα:

- **id:** Δηλώνει το αναγνωριστικό κωδικό της κάθε εντολής για μοίρασμα καρτών. Αποτελεί το πρωτεύων κλειδί (primary key) του πίνακα και ο τύπος του είναι ακέραιος (int).
- **count\_cards:** Δηλώνει το πλήθος των καρτών που θα μοιραστούν. Ο τύπος του πεδίου είναι ακέραιος (int).
- **hidden:** Δηλώνει αν η κάρτα θα είναι κλειστή (κρυφή). Ο τύπος του πεδίου είναι boolean με προκαθορισμένη τιμή το ψέμα (false).
- **type:** Δηλώνει το τύπο του κανόνα για το αν είναι κανόνας που αφορά κάποιο ρόλο παίχτη ή κάποια άδεια τράπουλα και είναι τύπου αντικειμένου αλφαριθμητικό (ENUM). Το αντικείμενο δέχεται δύο τιμές – role, deck – με προκαθορισμένη τιμή “role”.
- **created\_at:** Δηλώνει την ημερομηνία και την ώρα που δημιουργήθηκε η εντολή για μοίρασμα καρτών. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **updated\_at:** Δηλώνει την ημερομηνία και την ώρα που τροποποιήθηκε η εντολή μοιράσματος καρτών. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **game\_id:** Δηλώνει το αναγνωριστικό κωδικό του παιχνιδιού που ανήκει η εντολή. Αποτελεί ξένο κλειδί στον πίνακα games και ο τύπος του είναι ακέραιος (int).
- **deck\_id:** Δηλώνει το αναγνωριστικό κωδικό της τράπουλας που προσδιορίζει από ποια τράπουλα θα μοιράσει τις κάρτες. Αποτελεί ξένο κλειδί στον πίνακα decks και ο τύπος του είναι ακέραιος (int).
- **role\_id:** Δηλώνει το αναγνωριστικό κωδικό του ρόλου που αφορά σε ποιους θα μοιράσει τις ανάλογες κάρτες. Αποτελεί ξένο κλειδί στον πίνακα games και ο τύπος του είναι ακέραιος (int).
- **to\_deck\_id:** Δηλώνει το αναγνωριστικό κωδικό της τράπουλας που θα μοιράσει τις κάρτες. Αποτελεί ξένο κλειδί στο πίνακα decks και ο τύπος του είναι ακέραιος (int) και η τιμή του πεδίου μπορεί να παραμείνει κενή (nullable).

#### Ανάλυση σχέσεων με άλλους πίνακες:

- **games:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα `hand_start_cards` με όνομα `game_id`
- **decks:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα `hand_start_cards` με όνομα `deck_id` και `to_deck_id`
- **roles:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα `hand_start_cards` με όνομα `role_id`

#### Κώδικας για δημιουργία πίνακα και ορισμό σχέσεων με άλλους πίνακες σε γλώσσα nest.js:

```
@Entity('hand_start_cards')
export class HandStartCardsEntity {
  @PrimaryGeneratedColumn()
  id: number;

  @Column()
  count_cards: number

  @Column({ default: false })
  hidden: boolean;

  @Column({ type: 'enum', enum: HandStartCardsRuleType, default:
HandStartCardsRuleType.ROLE })
  type: HandStartCardsRuleType

  @CreateDateColumn({ type: 'timestamp', default: () =>
"CURRENT_TIMESTAMP(6)" })
  created_at: Date;

  @UpdateDateColumn({ type: "timestamp", default: () =>
"CURRENT_TIMESTAMP(6)", onUpdate: "CURRENT_TIMESTAMP(6)" })
  updated_at: Date;
```

```

    @ManyToOne(() => DecksEntity, (decksEntity) =>
decksEntity.hand_start_cards)
    @JoinColumn({ name: 'deck_id' })
    deck: DecksEntity

    @ManyToOne(() => DecksEntity, (decksEntity) =>
decksEntity.hand_start_cards_to_deck, { nullable: true })
    @JoinColumn({ name: 'to_deck_id' })
    toDeck: DecksEntity

    @ManyToOne(() => RolesEntity, (rolesEntity) =>
rolesEntity.hand_start_deck_id, { onDelete: 'CASCADE' })
    @JoinColumn({ name: 'role_id' })
    role: RolesEntity

    @ManyToOne(() => GamesEntity, (gamesEntity) =>
gamesEntity.hand_start_cards, { onDelete: 'CASCADE' })
    @JoinColumn({ name: 'game_id' })
    game: GamesEntity
}

```

### 3.7.10 Πίνακας decks\_games

Ο πίνακας decks\_games προκύπτει από την σχέση πολλά προς πολλά που έχουν οι πίνακες decks και games. Αυτός ο πίνακας προσδιορίζει τις τράπουλες που θα έχει το παιχνίδι.

Στην παρακάτω εικόνα παρουσιάζονται τα πεδία του πίνακα.

decks_game	
12	decksId int NOT NULL
12	gamesId int NOT NULL

Εικόνα 21 - Πίνακας decks\_game

### Ανάλυση πεδίων του πίνακα:

- **deckId:** Δηλώνει το αναγνωριστικό κωδικό της τράπουλας. Αποτελεί το ξένο κλειδί (foreign key) του πίνακα decks και ο τύπος του είναι ακέραιος (int).
- **gameId:** Δηλώνει το αναγνωριστικό κωδικό του παιχνιδιού. Αποτελεί το ξένο κλειδί (foreign key) του πίνακα games και ο τύπος του είναι ακέραιος (int).

### 3.7.11 Πίνακας tables

Ο πίνακας tables αφορά αποκλειστικά τα βασικά στοιχεία των ΗΤ που θα δημιουργούνται από τους εγγεγραμμένους χρήστες. Οι επόμενοι πίνακες που θα αναλυθούν στην συνέχεια αφορούν τα ΗΤ προς την προσομοίωση τους στα ΠΜΚ.

Στην παρακάτω εικόνα παρουσιάζονται τα πεδία του πίνακα με τα κύρια χαρακτηριστικά:

Field Name	Data Type	Constraints
id	int	NOT NULL
name	varchar(25)	NOT NULL
private	tinyint	NOT NULL
password	varchar(255)	
created_at	timestamp(6)	NOT NULL
updated_at	timestamp(6)	NOT NULL
user_id	int	
game_id	int	
status	enum('waiting','playing','pause','finish','game_master_edit','player_leave','player_disconnected')	NOT NULL
game_master_id	int	
public_url	varchar(255)	NOT NULL

Εικόνα 22 - Πίνακας tables

### Ανάλυση πεδίων του πίνακα:

- **id:** Δηλώνει το αναγνωριστικό κωδικό του ΗΤ. Αποτελεί το πρωτεύων κλειδί (primary key) του πίνακα και ο τύπος του είναι ακέραιος (int).
- **name:** Δηλώνει το όνομα το ΗΤ. Ο τύπος του πεδίου είναι αλφαριθμητικό (varchar) με μέγεθος 25 χαρακτήρες.
- **private:** Δηλώνει αν το ΗΤ είναι ιδιωτικό. Ο τύπος του πεδίου είναι boolean με προκαθορισμένη τιμή το ψέμα (false).
- **password:** Δηλώνει τον κωδικό του ΗΤ αν υπάρχει. Ο τύπος του πεδίου είναι αλφαριθμητικό (varchar) με μέγεθος 255 χαρακτήρες και η τιμή του πεδίου μπορεί να παραμείνει κενή (nullable).
- **status:** Δηλώνει την κατάσταση που βρίσκεται το τραπέζι και είναι τύπου αντικείμενου αλφαριθμητικό (ENUM). Το αντικείμενο δέχεται τις εξής τιμές: waiting, playing, pause, finish, game\_master\_edit, player\_leave και player\_disconnected. Η προκαθορισμένη τιμή του είναι “waiting”.
- **game\_master\_id:** Δηλώνει το αναγνωριστικό κωδικό του χρήστη που είναι ο GM του τραπεζιού. Αποτελεί ξένο κλειδί στον πίνακα users και ο τύπος του είναι ακέραιος (int).

- **public\_url:** Δηλώνει το κοινόχρηστο αριθμό του δωματιού που δημιουργείται από το πακέτο UUID (Universal Unique Identifier). Ο τύπος του πεδίου είναι αλφαριθμητικό (varchar) με μέγεθος 255 χαρακτήρες.
- **created\_at:** Δηλώνει την ημερομηνία και την ώρα που δημιουργήθηκε το HT. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **updated\_at:** Δηλώνει την ημερομηνία και την ώρα που τροποποιήθηκε το HT. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **user\_id:** Δηλώνει το αναγνωριστικό κωδικό του χρήστη που δημιούργησε το παιχνίδι. Αποτελεί ξένο κλειδί στον πίνακα users και ο τύπος του είναι ακέραιος (int).
- **game\_id:** Δηλώνει το αναγνωριστικό κωδικό του παιχνιδιού που θα προσομοιώσει το HT. Αποτελεί ξένο κλειδί στον πίνακα games και ο τύπος του είναι ακέραιος (int).

#### Ανάλυση σχέσεων με άλλους πίνακες:

- **users:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα tables με όνομα user\_id και game\_master\_id
- **games:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα tables με όνομα game\_id
- **table\_decks:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_decks με όνομα table\_id
- **table\_users:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_users με όνομα table\_id
- **ranks:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα ranks με όνομα table\_id

#### Κώδικας για δημιουργία πίνακα και ορισμό σχέσεων με άλλους πίνακες σε γλώσσα nest.js:

```
@Entity('tables')
export class TablesEntity {
  @PrimaryGeneratedColumn()
  id: number;

  @Column({ length: 25 })
  name: string;

  @Column({ default: false })
  private: boolean;
}
```

```
@Column({ nullable: true })
password: string;

@Column({ type: 'enum', enum: TableStatus, default: TableStatus.WAITING
})
status: TableStatus;

@Column()
public_url: string;

@CreateDateColumn({ type: 'timestamp', default: () =>
"CURRENT_TIMESTAMP(6)" })
created_at: Date;

@UpdateDateColumn({ type: "timestamp", default: () =>
"CURRENT_TIMESTAMP(6)", onUpdate: "CURRENT_TIMESTAMP(6)" })
updated_at: Date;

@ManyToOne(() => UsersEntity, (usersEntity) => usersEntity.tables, {
onDelete: 'SET NULL' })
@JoinColumn({ name: 'user_id' })
creator: UsersEntity

@ManyToOne(() => GamesEntity, (gamesEntity) => gamesEntity.table_id)
@JoinColumn({ name: 'game_id' })
game: GamesEntity

@OneToMany(() => TablesDecksEntity, (tablesDecksEntity) =>
tablesDecksEntity.table)
table_decks: TablesDecksEntity[]

@OneToMany(() => RankEntity, (rankEntity) => rankEntity.table)
```

```

ranks: RankEntity[]

@OneToMany(() => TableUsersEntity, (tableUsersEntity) =>
tableUsersEntity.table, { onDelete: 'CASCADE' })
table_users: TableUsersEntity[]

@ManyToOne(() => UsersEntity, (usersEntity) =>
usersEntity.tables_game_master)
@JoinColumn({ name: 'game_master_id' })
game_master: UsersEntity
}

```

### 3.7.12 Πίνακας table\_users

Ο πίνακας table\_users αφορά του χρήστες που θα ανήκουν στο HT. Στον πίνακα αυτό θα καταγράφονται οι χρήστες που έχουν σύνδεση με web socket σύνδεση ώστε να τους επιτραπεί η είσοδος σε κάποιο τραπέζι. Επίσης, σ' αυτόν τον πίνακα θα υπάρχουν και στοιχεία που αφορούν το τραπέζι που συμμετέχουν.

Στην παρακάτω εικόνα παρουσιάζονται τα πεδία του πίνακα με τα κύρια χαρακτηριστικά:

Field Name	Data Type / Constraint
id	int NOT NULL
turn	int
playing	tinyint NOT NULL
created_at	timestamp(6) NOT NULL
updated_at	timestamp(6) NOT NULL
user_id	int
table_id	int
role_id	int
socket_id	varchar(255)
socket_status	enum('online','offline','leave','room','disconnect') NOT NULL
team_id	int
status_id	int

Εικόνα 23 - Πίνακας table\_users

#### Ανάλυση πεδίων του πίνακα:

- **id:** Δηλώνει το αναγνωριστικό κωδικό χρήστη στο HT. Αποτελεί το πρωτεύων κλειδί (primary key) του πίνακα και ο τύπος του είναι ακέραιος (int).
- **turn:** Δηλώνει την σειρά που θα παίζουν οι παίχτες στο HT. Ο τύπος του πεδίου είναι ακέραιος (int) και η τιμή του πεδίου μπορεί να παραμείνει κενή (nullable).
- **playing:** Δηλώνει αν παίχτης είναι η σειρά του να παίξει. Ο τύπος του πεδίου είναι Boolean με προκαθορισμένη τιμή την αλήθεια (false).

- **socket\_id:** Δηλώνει το αναγνωριστικό κωδικό της σύνδεσης του με τον web socket server. Ο τύπος του πεδίου είναι αλφαριθμητικό (varchar) με μέγεθος 255 χαρακτήρες και η τιμή του πεδίου μπορεί να παραμείνει κενή (nullable).
- **socket\_status:** Δηλώνει το τύπο της σύνδεσης του χρήστη που έχει με την πλατφόρμα και είναι τύπου αντικειμένου αλφαριθμητικό (ENUM). Το αντικείμενο δέχεται τις τιμές – online, offline, leave, room, disconnect – με προκαθορισμένη τιμή την “online”.
- **created\_at:** Δηλώνει την ημερομηνία και την ώρα που εισήλθε ο χρήστης στο HT. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **updated\_at:** Δηλώνει την ημερομηνία και την ώρα που τροποποιήθηκε ο χρήστης στο HT. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **user\_id:** Δηλώνει το αναγνωριστικό κωδικό του χρήστη. Αποτελεί ξένο κλειδί στον πίνακα users και ο τύπος του είναι ακέραιος (int).
- **role\_id:** Δηλώνει το αναγνωριστικό κωδικό ρόλου που θα έχει ο παίχτης στο παιχνίδι. Αποτελεί ξένο κλειδί στον πίνακα roles και ο τύπος του είναι ακέραιος (int). Η τιμή του πεδίου μπορεί να είναι κενή (nullable).
- **status\_id:** Δηλώνει το αναγνωριστικό κωδικό κατάστασης που θα έχει ο παίχτης στο παιχνίδι αν έχει. Αποτελεί ξένο κλειδί στο πίνακα status και ο τύπος του είναι ακέραιος (int). Η τιμή του πεδίου μπορεί να είναι κενή (nullable).
- **team\_id:** Δηλώνει το αναγνωριστικό κωδικό ομάδας που θα έχει ο παίχτης στο παιχνίδι αν έχει. Αποτελεί ξένο κλειδί στο πίνακα teams και ο τύπος του είναι ακέραιος (int). Η τιμή του πεδίου μπορεί να είναι κενή (nullable).

#### Ανάλυση σχέσεων με άλλους πίνακες:

- **tables:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_users με όνομα table\_id.
- **users:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_users με όνομα user\_id.
- **roles:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_users με όνομα role\_id.
- **teams:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_users με όνομα team\_id.
- **status:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_users με όνομα status\_id.
- **table\_deck:** Έχει σχέση ένα προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_deck με όνομα table\_user\_id.
- **Rank:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα ranks με όνομα table\_user\_id.



Κώδικας για δημιουργία πίνακα και ορισμό σχέσεων με άλλους πίνακες σε γλώσσα nest.js:

```
@Entity('table_users')
export class TableUsersEntity {
  @PrimaryGeneratedColumn()
  id: number;

  @Column({ nullable: true })
  turn: number;

  @Column({ default: false })
  playing: boolean;

  @Column({ nullable: true })
  socket_id: string;

  @Column({ type: 'enum', enum: SocketStatus, default:
SocketStatus.ONLINE })
  socket_status: SocketStatus

  @CreateDateColumn({ type: 'timestamp', default: () =>
"CURRENT_TIMESTAMP(6)" })
  created_at: Date;

  @UpdateDateColumn({ type: "timestamp", default: () =>
"CURRENT_TIMESTAMP(6)", onUpdate: "CURRENT_TIMESTAMP(6)" })
  updated_at: Date;

  @ManyToOne(() => UsersEntity, (usersEntity) =>
usersEntity.table_users_id, { onDelete: 'CASCADE' })
  @JoinColumn({ name: 'user_id' })
  user: UsersEntity
```

```
@ManyToOne(() => TablesEntity, (tableEntity) =>
tableEntity.table_users, { cascade: true, nullable: true })
@JoinColumn({ name: 'table_id' })
table: TablesEntity

@ManyToOne(() => RolesEntity, (rolesEntity) =>
rolesEntity.table_user_id, { cascade: true, nullable: true })
@JoinColumn({ name: 'role_id' })
role: RolesEntity

@ManyToOne(() => TeamsEntity, (teamsEntity) => teamsEntity.table_user,
{ cascade: true, nullable: true })
@JoinColumn({ name: 'team_id' })
team: TeamsEntity

@ManyToOne(() => StatusEntity, (statusEntity) =>
statusEntity.table_user, { cascade: true, nullable: true })
@JoinColumn({ name: 'status_id' })
status: StatusEntity

@OneToOne(() => TablesDecksEntity, (tablesDecksEntity) =>
tablesDecksEntity.table_user, { cascade: true })
table_deck: TablesDecksEntity

@OneToMany(() => RankEntity, (randEntity) => randEntity.table_user, {
cascade: true })
rank: RankEntity
}
```

### 3.7.13 Πίνακας `table_decks`

Ο πίνακας `table_decks` θα περιέχει τις στοίβες από τα χαρτιά των παιχτών, από τις τράπουλες και από το HT. Με την έναρξη του παιχνιδιού θα δημιουργούνται οι απαραίτητες στοίβες με κάρτες ώστε να καταγράφονται οι κινήσεις των παιχτών και να ενημερώνονται και οι υπόλοιποι παίχτες για τις αλλαγές που προκύπτουν στο HT.

Στην παρακάτω εικόνα παρουσιάζονται τα πεδία του πίνακα με τα κύρια χαρακτηριστικά:

Field Name	Field Type
<code>id</code>	<code>int NOT NULL</code>
<code>created_at</code>	<code>timestamp(6) NOT NULL</code>
<code>updated_at</code>	<code>timestamp(6) NOT NULL</code>
<code>table_id</code>	<code>int</code>
<code>user_id</code>	<code>int</code>
<code>deck_id</code>	<code>int</code>
<code>table_user_id</code>	<code>int</code>
<code>type</code>	<code>enum('user','table','deck','junk') NOT NULL</code>

Εικόνα 24 - Πίνακας `table_decks`

#### Ανάλυση πεδίων του πίνακα:

- **id:** Δηλώνει το αναγνωριστικό κωδικό της στοίβας στο HT. Αποτελεί το πρωτεύων κλειδί (primary key) του πίνακα και ο τύπος του είναι ακέραιος (`int`).
- **type:** Δηλώνει το τύπο της στοίβας της τράπουλας και είναι τύπου αντικειμένου αλφαριθμητικό (ENUM). Το αντικείμενο αποτελείται από τις εξής τιμές: `user`, `table`, `deck` και `junk`. Η προκαθορισμένη τιμή είναι “`user`”.
- **created\_at:** Δηλώνει την ημερομηνία και την ώρα που δημιουργήθηκε η στοίβα του HT. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (`datetime`).
- **updated\_at:** Δηλώνει την ημερομηνία και την ώρα που τροποποιήθηκε η στοίβα στο HT. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (`datetime`).
- **table\_id:** Δηλώνει το αναγνωριστικό κωδικό του HT που ανήκει η στοίβα. Αποτελεί ξένο κλειδί στον πίνακα `tables` και ο τύπος του είναι ακέραιος (`int`).
- **table\_user\_id:** Δηλώνει το αναγνωριστικό κωδικό του χρήστη που ανήκει σε κάποιο τραπέζι. Αποτελεί ξένο κλειδί στο πίνακα `table_users` και ο τύπος του είναι ακέραιος (`int`). Η τιμή μπορεί να είναι κενή (`nullable`).
- **user\_id:** Δηλώνει το αναγνωριστικό κωδικό του χρήστη που ανήκει η στοίβα. Αποτελεί ξένο κλειδί στον πίνακα `users` και ο τύπος του είναι ακέραιος (`int`). Η τιμή μπορεί να είναι κενή για τον λόγο ότι η στοίβα θα ανήκει στο τραπέζι.
- **deck\_id:** Δηλώνει το αναγνωριστικό κωδικό της τράπουλας που θα προσδιορίσει ποιες κάρτες θα έχει η στοίβα. Η τιμή μπορεί να είναι κενή (`nullable`).

#### Ανάλυση σχέσεων με άλλους πίνακες:

- **tables:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα `table_decks` με όνομα `table_id`

- **users:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα `table_decks` με όνομα `user_id`
- **table\_cards:** Έχει σχέση ένα προς πολλά και το πεδίο συσχέτισης δημιουργείται στο πίνακα `table_cards` με όνομα `table_deck_id`
- **decks:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα `table_decks` με όνομα `deck_id`
- **table\_user:** Έχει σχέση ένα προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα `table_decks` με όνομα `table_user_id`

Κώδικας για δημιουργία πίνακα και ορισμό σχέσεων με άλλους πίνακες σε γλώσσα `nest.js`:

```
@Entity('table_decks')
export class TablesDecksEntity {
  @PrimaryGeneratedColumn()
  id: number;

  @Column({ type: 'enum', enum: TableDeckType, default:
TableDeckType.USER })
  type: TableDeckType

  @CreateDateColumn({ type: 'timestamp', default: () =>
"CURRENT_TIMESTAMP(6)" })
  created_at: Date;

  @UpdateDateColumn({ type: "timestamp", default: () =>
"CURRENT_TIMESTAMP(6)", onUpdate: "CURRENT_TIMESTAMP(6)" })
  updated_at: Date;

  @ManyToOne(() => TablesEntity, (tablesEntity) =>
tablesEntity.table_decks)
  @JoinColumn({ name: 'table_id' })
  table: TablesEntity;

  @ManyToOne(() => UsersEntity, (usersEntity) => usersEntity.table_decks,
{ nullable: true })
```

```

@JoinColumn({ name: 'user_id' })
user?: UsersEntity;

@ManyToOne(() => DecksEntity, (decksEntity) =>
decksEntity.table_deck_id, { nullable: true })
@JoinColumn({ name: 'deck_id' })
deck?: DecksEntity;

@OneToMany(() => TablesCardsEntity, (tablesCardsEntity) =>
tablesCardsEntity.table_deck)
table_cards: TablesCardsEntity[];

@OneToOne(() => TableUsersEntity, (tableUsersEntity) =>
tableUsersEntity.table_deck, { nullable: true })
@JoinColumn({ name: 'table_user_id' })
table_user?: TableUsersEntity
}

```

### 3.7.14 Πίνακας table\_cards

Ο πίνακας table\_cards θα περιέχει στοιχεία της κάρτας ως προς την εμφάνιση της στο HT και σε ποια στοίβα θα ανήκει στο τραπέζι. Οι εγγραφές του πίνακα θα δημιουργούνται με την έναρξη του παιχνιδιού και θα ενημερώνεται κατά την διάρκεια του παιχνιδιού.

Στην παρακάτω εικόνα παρουσιάζονται τα πεδία του πίνακα με τα κύρια χαρακτηριστικά:

table_cards	
123 id	int NOT NULL
123 hidden	tinyint NOT NULL
123 rotate	int NOT NULL
123 turn	int
123 position_x	int
123 position_y	int
123 created_at	timestamp(6) NOT NULL
123 updated_at	timestamp(6) NOT NULL
123 table_deck_id	int
123 card_id	int
123 z_index	int NOT NULL

Εικόνα 25 - Πίνακας table\_cards

#### Ανάλυση πεδίων του πίνακα:

- **id:** Δηλώνει το αναγνωριστικό κωδικό της κάρτας του HT. Αποτελεί το πρωτεύων κλειδί (primary key) του πίνακα και ο τύπος του είναι ακέραιος (int).
- **rotate:** Δηλώνει την κλίση που θα έχει η κάρτα στο τραπέζι. Ο τύπος του πεδίου είναι ακέραιος (int) που θα προσδιορίζει τις μοίρες. Η προκαθορισμένη (default) είναι το 0.
- **hidden:** Δηλώνει αν η κάρτα θα είναι κλειστή (κρυφή). Ο τύπος του πεδίου είναι boolean με προκαθορισμένη τιμή την αλήθεια (true).
- **turn:** Δηλώνει την σειρά που είναι οι κάρτες στην τράπουλα. Ο τύπος του πεδίου είναι ακέραιος (int). Η τιμή μπορεί να είναι κενή γιατί δεν είναι απαραίτητη η σειρά των καρτών σε κάποιες περιπτώσεις όπως στο τραπέζι. Η τιμή μπορεί να είναι κενή (nullable).
- **position\_x:** Δηλώνει το σημείο που βρίσκεται η κάρτα στον άξονα X της οθόνης. Ο τύπος του πεδίου είναι ακέραιος (int). Η τιμή μπορεί να είναι κενή γιατί όταν η κάρτα βρίσκεται στην τράπουλα ή στο χρήστη θα βρίσκεται σε συγκεκριμένα σημεία στην οθόνη.
- **position\_y:** Δηλώνει το σημείο που βρίσκεται η κάρτα στον άξονα Y της οθόνης. Ο τύπος του πεδίου είναι ακέραιος (int). Η τιμή μπορεί να είναι κενή γιατί όταν η κάρτα βρίσκεται στην τράπουλα ή στο χρήστη θα βρίσκεται σε συγκεκριμένα σημεία στην οθόνη.
- **z\_index:** Δηλώνει τον αριθμό που καθορίζει τη σειρά των επικαλυπτόμενων στοιχείων HTML. Ο τύπος του πεδίου είναι ακέραιος (int). Η προκαθορισμένη τιμή είναι το 0.
- **created\_at:** Δηλώνει την ημερομηνία και την ώρα που δημιουργήθηκε η κάρτα της στοίβας. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **updated\_at:** Δηλώνει την ημερομηνία και την ώρα που τροποποιήθηκε η κάρτα της στοίβας. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **table\_deck\_id:** Δηλώνει το αναγνωριστικό κωδικό της στοίβας του HT. Αποτελεί ξένο κλειδί στον πίνακα table\_decks και ο τύπος του είναι ακέραιος (int).
- **card\_id:** Δηλώνει το αναγνωριστικό κωδικό της κάρτας ώστε να είναι διαθέσιμη η εικόνα της κάρτας. Αποτελεί ξένο κλειδί στον πίνακα cards και ο τύπος του είναι ακέραιος (int).

#### Ανάλυση σχέσεων με άλλους πίνακες:

- **table\_deck:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_cards με όνομα table\_deck\_id
- **cards:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα table\_cards με όνομα card\_id

Κώδικας για δημιουργία πίνακα και ορισμό σχέσεων με άλλους πίνακες σε γλώσσα nest.js:

```
@Entity('table_cards')
export class TablesCardsEntity {
  @PrimaryGeneratedColumn()
  id: number;

  @Column({ default: false })
  hidden: boolean;

  @Column({ default: 0 })
  rotate: number;

  @Column({ nullable: true })
  turn: number;

  @Column({ nullable: true, default: 0 })
  position_x: number;

  @Column({ nullable: true, default: 0 })
  position_y: number;

  @Column({ default: 0 })
  z_index: number;

  @CreateDateColumn({ type: 'timestamp', default: () =>
"CURRENT_TIMESTAMP(6)" })
  created_at: Date;

  @UpdateDateColumn({ type: "timestamp", default: () =>
"CURRENT_TIMESTAMP(6)", onUpdate: "CURRENT_TIMESTAMP(6)" })
  updated_at: Date;
```

```

    @ManyToOne(() => TablesDecksEntity, (tablesDecksEntity) =>
tablesDecksEntity.table_cards)

    @JoinColumn({ name: 'table_deck_id' })
    table_deck: TablesDecksEntity

    @ManyToOne(() => CardsEntity, (cardsEntity) =>
cardsEntity.table_cards_id)

    @JoinColumn({ name: 'card_id' })
    card: CardsEntity
}

```

### 3.7.15 Πίνακας ranks

Ο πίνακας ranks αφορά τις βαθμολογίες που θα έχουν οι παίκτες στο τραπέζι. Οι βαθμολογίες θα δημιουργούνται με την έναρξη του παιχνιδιού και θα τροποποιούνται μόνο από τον GM.

Στην παρακάτω εικόνα παρουσιάζονται τα πεδία του πίνακα με τα κύρια χαρακτηριστικά:

ranks	
id	int NOT NULL
points	int
row	int NOT NULL
created_at	timestamp(6) NOT NULL
updated_at	timestamp(6) NOT NULL
table_id	int
table_user_id	int
type	enum('title','points') NOT NULL
title	varchar(25)

Εικόνα 26 - Πίνακας ranks

#### Ανάλυση πεδίων του πίνακα:

- **id:** Δηλώνει το αναγνωριστικό κωδικό βαθμολογίας στο HT. Αποτελεί το πρωτεύων κλειδί (primary key) του πίνακα και ο τύπος του είναι ακέραιος (int).
- **points:** Δηλώνει το πλήθος των πόντων που έχει ο παίχτης. Ο τύπος του πεδίου είναι ακέραιος (int) και η τιμή του πεδίου μπορεί να παραμείνει κενή (nullable).
- **row:** Δηλώνει τον αριθμό της γραμμής που ανήκει η συγκεκριμένη βαθμολογία. Ο τύπος του πεδίου είναι ακέραιος (int).



- **type:** Δηλώνει το τύπο της βαθμολογίας αν ανήκει σε βαθμολογία σε κάποιο παίχτη ή ονομασία της γραμμής και είναι τύπου αντικειμένου αλφαριθμητικό (ENUM). Το αντικείμενο δέχεται τις τιμές “title” και “points”. Η προκαθορισμένη τιμή είναι “points”.
- **title:** Δηλώνει την ονομασία της γραμμής της βαθμολογίας. Ο τύπος του πεδίου είναι αλφαριθμητικό (varchar) με μέγεθος 25 χαρακτήρες και η τιμή του πεδίου μπορεί να παραμείνει κενή (nullable).
- **created\_at:** Δηλώνει την ημερομηνία και την ώρα που δημιουργήθηκε η βαθμολογία. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **updated\_at:** Δηλώνει την ημερομηνία και την ώρα που τροποποιήθηκε η βαθμολογία. Ο τύπος του πεδίου είναι ημερομηνία και ώρα (datetime).
- **table\_user\_id:** Δηλώνει το αναγνωριστικό κωδικό του χρήστη που ανήκει η βαθμολογία. Αποτελεί ξένο κλειδί στον πίνακα table\_users και ο τύπος του είναι ακέραιος (int).
- **table\_id:** Δηλώνει το αναγνωριστικό κωδικό του HT που ανήκει η βαθμολογία. Αποτελεί ξένο κλειδί στον πίνακα tables και ο τύπος του είναι ακέραιος (int).

#### Ανάλυση σχέσεων με άλλους πίνακες:

- **tables:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα ranks με όνομα table\_id
- **table\_users:** Έχει σχέση πολλά προς ένα και το πεδίο συσχέτισης δημιουργείται στο πίνακα ranks με όνομα table\_user\_id

Κώδικας για δημιουργία πίνακα και ορισμό σχέσεων με άλλους πίνακες σε γλώσσα nest.js:

```
@Entity('ranks')
export class RankEntity {
  @PrimaryGeneratedColumn()
  id: number;

  @Column()
  points: number;

  @Column()
  row: number;
```

```
@CreateDateColumn({ type: 'timestamp', default: () =>
"CURRENT_TIMESTAMP(6)" })
created_at: Date;

@UpdateDateColumn({ type: "timestamp", default: () =>
"CURRENT_TIMESTAMP(6)", onUpdate: "CURRENT_TIMESTAMP(6)" })
updated_at: Date;

@ManyToOne(() => TablesEntity, (tablesEntity) => tablesEntity.ranks)
@JoinColumn({ name: 'table_id' })
table: TablesEntity

@ManyToOne(() => UsersEntity, (usersEntity) => usersEntity.ranks)
@JoinColumn({ name: 'user_id' })
user_id: UsersEntity
}
```

### 3.8 Ασφάλεια συστήματος

Η ασφάλεια στις εφαρμογές διαδικτύου είναι υψίστης σημασίας, καθώς συχνά γίνονται στόχος από κακόβουλους παράγοντες που επιδιώκουν να εκμεταλλευτούν αδύνατα σημεία της εφαρμογής και να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση σε ευαίσθητες πληροφορίες.

Με την αυξανόμενη παραγωγή διαδικτυακών εφαρμογών τόσο για προσωπική όσο και για επαγγελματική χρήση, είναι απαραίτητη ανάγκη να κατανοηθούν οι διάφοροι κίνδυνοι που αντιμετωπίζουν αυτές οι εφαρμογές και να εφαρμοστούν μέτρα για τον περιορισμό τους. Αυτό περιλαμβάνει την εφαρμογή ασφαλών πρακτικών κωδικοποίησης, και τη χρήση ισχυρών μηχανισμών ελέγχου ταυτότητας και εξουσιοδότησης για την προστασία από μη εξουσιοδοτημένη πρόσβαση. Επιπλέον, οι τακτικές δοκιμές ασφαλείας είναι ζωτικής σημασίας για τον εντοπισμό και την αποκατάσταση πιθανών αδυναμιών του συστήματος. Λαμβάνοντας μια προληπτική προσέγγιση για την ασφάλεια των εφαρμογών αυτών, είναι δυνατό να ελαχιστοποιηθεί ο κίνδυνος επιτυχημένων επιθέσεων και να προστατεύονται έτσι καλύτερα ευαίσθητες πληροφορίες παρά να πέφτουν σε λάθος χέρια.

Συνοπτικά, η ασφάλεια διαδικτυακών εφαρμογών είναι μια περίπλοκη και δύσκολη διαδικασία καθώς είναι αδύνατον να υπάρξει απόλυτη ασφάλεια. Ωστόσο, με την κατανόηση των κινδύνων και την εφαρμογή μεθοδολογιών, είναι δυνατό να

επιτευχθεί μια υψηλή προστασία και να διατηρηθούν οι ευαίσθητες πληροφορίες ασφαλείς. Μερικές τεχνικές ασφαλείας είναι οι εξής:

**Κρυπτογράφηση κωδικών πρόσβασης:** Η κρυπτογράφηση κωδικών πρόσβασης είναι ένα ουσιαστικό βήμα για την ασφάλεια των εφαρμογών διαδικτύου. Διασφαλίζει ότι ακόμη και αν ένας εισβολέας αποκτήσει πρόσβαση στη βάση δεδομένων της εφαρμογής, δεν θα μπορεί να δει ή να χρησιμοποιήσει τους αποθηκευμένους κωδικούς πρόσβασης. Μία από τις πιο ευρέως χρησιμοποιούμενες μεθόδους κρυπτογράφησης για κωδικούς πρόσβασης είναι η bcrypt. Το bcrypt είναι μια βιβλιοθήκη που χρησιμοποιεί μια τεχνική που ονομάζεται «επέκταση κλειδιού» (key stretching) για να κάνει αδύνατο το σπάσιμο των κρυπτογραφημένων κωδικών πρόσβασης, ακόμη και με τη χρήση ισχυρού υλικού από κακοπροαίρετες πηγές.

**Έλεγχος ταυτότητας και εξουσιοδότησης:** Για την ταυτοποίηση του χρήστη μεταξύ του χρήστη και του διακομιστή, χρησιμοποιούνται ένα κλειδί (JWT – Json Web Token). Αυτά τα κλειδιά περιέχουν πληροφορίες για τον χρήστη, όπως την ταυτότητά του, και υπογράφονται από τον διακομιστή, επιτρέποντας στον πελάτη να εμπιστευτεί ότι οι πληροφορίες είναι αυθεντικές. Μπορούν επίσης να κρυπτογραφηθούν για την αποφυγή παραποίησης και πλαστογραφίας, παρέχοντας έτσι ένα επιπρόσθετο επίπεδο ασφαλείας στην εφαρμογή και προστατεύοντας την από επιθέσεις.

**Ασφάλεια από επιθέσεις “ενέσεων” εντολών SQL (SQL injection):** Το SQL injection είναι ένας τύπος επίθεσης στις εφαρμογές διαδικτύου όπου ένας εισβολέας εισάγει κακόβουλο κώδικα σε ένα πεδίο εισαγωγής, και αυτή του η κίνηση του επιτρέπει να αποκτήσει μη εξουσιοδοτημένη πρόσβαση σε μια βάση δεδομένων. Για να αποφευχθεί το SQL injection, θα πρέπει να χρησιμοποιηθούν έλεγχοι στα πεδία εισαγωγής ώστε να διασφαλιστεί ότι η είσοδος που παρέχεται από τον χρήστη δεν είναι κακόβουλη. Επιπλέον, η χρήση ενός πλαισίου ORM (Object-Relational Mapping) μπορεί να βοηθήσει στον περιορισμό των κινδύνων από SQL injection επιθέσεις.

### 3.9 Σύνοψη κεφαλαίου

Στο τρίτο κεφάλαιο αρχικά πραγματοποιήθηκε η ανάλυση και η καταγραφή των απαιτήσεων της πλατφόρμας και των δημοφιλείς παιχνιδιών που θα υπάρχουν σ' αυτή. Καταγράφηκαν οι περιπτώσεις χρήσης για τον κάθε τύπου χρήστη που θα υπάρχει στην πλατφόρμα. Στην συνέχεια αναλύθηκε λεπτομερώς η βάση δεδομένων ως προς τους πίνακες που απαρτίζεται, τις σχέσεις που θα έχουν μεταξύ τους, τα πεδία τους και οι τύποι των πεδίων τους. Τέλος, έγινε αναφορά σε πιθανούς κινδύνους που μπορεί να έχει μια εφαρμογή διαδικτύου αλλά και σε μεθοδολογίες για επίλυση τους ώστε να χρησιμοποιηθούν στην πλατφόρμα για καλύτερη προστασία της.

## 4. Λειτουργίες και διεπαφή χρήστη

Σε αυτό το κεφάλαιο, παρουσιάζονται οι λειτουργίες και οι δυνατότητες της πλατφόρμας μέσω της διεπαφής χρήστη (User Interface). Η διεπαφή χρήστη μιας εφαρμογής διαδικτύου είναι τα οπτικά στοιχεία με τα οποία αλληλοεπιδρά ο χρήστης. Αυτό περιλαμβάνει τη διάταξη, τα κουμπιά, το κείμενο, τις εικόνες και άλλα στοιχεία που εμφανίζονται στην οθόνη. Η σχεδίαση διεπαφής χρήστη θα πρέπει να είναι αισθητικά εμφανίσιμη και εύκολη προς τον χρήστη (User friendly) αλλιώς θα οδηγήσει στην απομάκρυνση των επισκεπτών. Επίσης, η διεπαφή περιέχει τμήματα τόσο του υλικού του συστήματος όσο και του λογισμικού.

Με βάση αυτό το σημαντικό κριτήριο, θα δοθεί η απαραίτητη σημασία ώστε η διεπαφή της πλατφόρμας να είναι εύχρηστη και φιλική προς τον χρήστη.

### 4.1 Αρχική Σελίδα

Όταν ο χρήστης πληκτρολογήσει την διεύθυνση URL στο πρόγραμμα περιήγησης, θα φορτωθεί η αρχική σελίδα της πλατφόρμας. Στο πάνω μέρος της σελίδας υπάρχει το μενού πλοήγησης της πλατφόρμας, που θα είναι ορατό σε όλες τις σελίδες ώστε να έχει την δυνατότητα ο χρήστης να έχει άμεση πλοήγηση στην πλατφόρμα. Μετά την είσοδο του χρήστη με τα προσωπικά του στοιχεία, στο μενού πλοήγησης θα υπάρχουν διαφορετικές επιλογές.

Στην μέση της σελίδας αναγράφεται το όνομα της σελίδας καθώς και λίγα λόγια σχετικά με την πλατφόρμα. Υπάρχει και μία εικόνα με κάρτες ώστε να γίνει πιο κατανοητό στον χρήστη τι προσφέρει η πλατφόρμα. Στην συνέχεια υπάρχουν δύο κουμπιά πλοήγησης, το ένα κατευθύνει τον χρήστη στην σελίδα «Register» ώστε να έχει στην διάθεση του όλες τις λειτουργίες της πλατφόρμας και το άλλο κουμπί τον κατευθύνει στην σελίδα «Lobby» ώστε να λάβει μέρος σε ΠΜΚ.

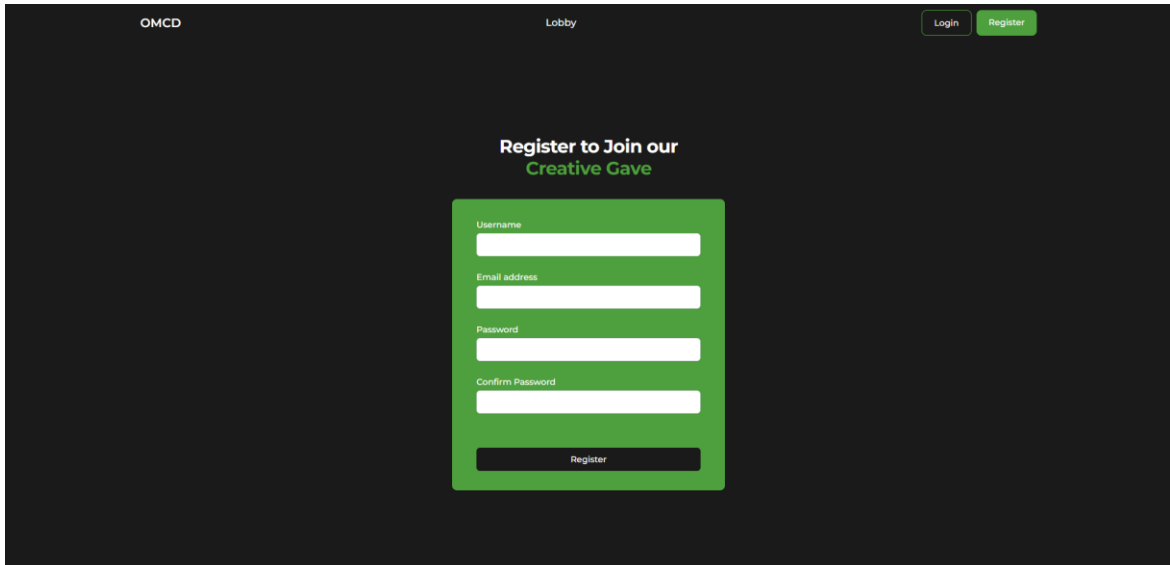
Στην παρακάτω εικόνα, παρουσιάζεται η αρχική σελίδα της πλατφόρμας.



Εικόνα 27 – Στιγμιότυπο από την σελίδα Home page

## 4.2 Εγγραφή χρήστη

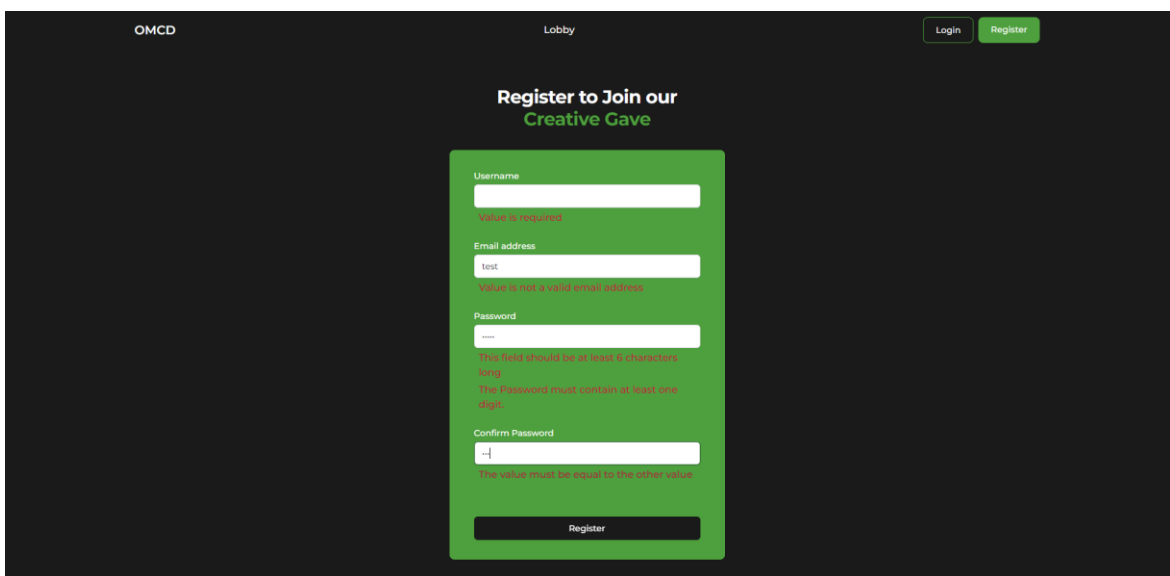
Στην σελίδα “Register” υπάρχει μία φόρμα με συγκεκριμένα πεδία που είναι όλα υποχρεωτικά και ο χρήστης έχει την δυνατότητα να εγγραφεί στην πλατφόρμα με τα προσωπικά του στοιχεία.



The screenshot shows a registration form titled "Register to Join our Creative Gave". The form is centered on a dark background. It contains four input fields: "Username", "Email address", "Password", and "Confirm Password". Below the fields is a "Register" button. The top navigation bar includes "OMCD", "Lobby", and "Login" and "Register" buttons.

Εικόνα 28 - Στιγμιότυπο από την σελίδα Register

Αφού ο χρήστης συμπληρώσει τα στοιχεία του στην φόρμα και υποβάλει την αίτηση του, τότε γίνονται οι κατάλληλοι έλεγχοι στο frontend κατά πόσο τα στοιχεία που δοθήκανε είναι σωστά. Στην περίπτωση που εισάχθηκαν λάθος στοιχεία τότε εμφανίζονται τα μηνύματα αποτυχίας κάτω από τα πεδία που έγινε λάθος εισαγωγή όπως φαίνεται στην παρακάτω εικόνα [Εικόνα 29].

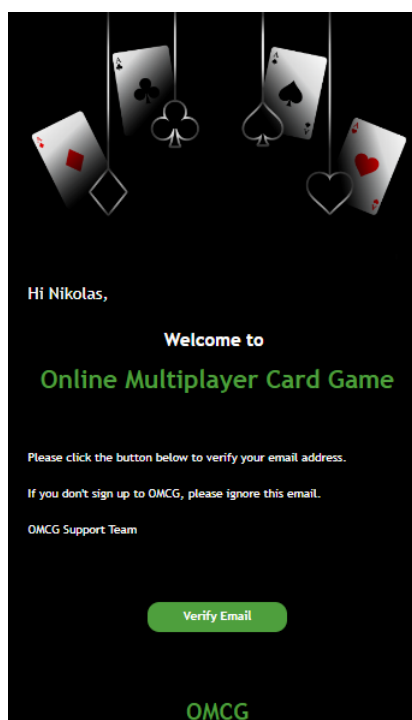


The screenshot shows the same registration form as in Figure 28, but with error messages displayed below each input field. The "Username" field has the message "value is required". The "Email address" field has the message "value is not a valid email address". The "Password" field has two messages: "This field should be at least 6 characters long" and "The Password must contain at least one digit". The "Confirm Password" field has the message "The value must be equal to the other value". The "Register" button is still visible at the bottom.

Εικόνα 29 - Στιγμιότυπο από την σελίδα Register με μηνύματα αποτυχίας

Έλεγχοι πραγματοποιούνται και στο backend, τόσο για να διασφαλιστεί ότι τα δεδομένα είναι τα σωστά προς αποφυγήν ανεπιθύμητων επιθέσεων όσο και για τον έλεγχο των στοιχείων δεδομένου του χρήστη που δεν μπορούν να γίνουν στο frontend. Για παράδειγμα, τα πεδία username και email οφείλουν να είναι μοναδικά, επομένως σε περίπτωση που χρησιμοποιηθούν στοιχεία που υπάρχουν ήδη στην βάση δεδομένων τότε θα εμφανίσει τα κατάλληλα μηνύματα αποτυχίας.

Αφού ο χρήστης συμπληρώσει σωστά τα στοιχεία του, τότε εμφανίζεται ένα αναδυόμενο παράθυρο (popup) με το μήνυμα επιτυχία. Ενημερώνει τον χρήστη ότι έχει λάβει μήνυμα στο ηλεκτρονικό του ταχυδρομείο και το παροτρύνει να το επιβεβαιώσει. Το μήνυμα που λαμβάνει είναι το εξής:

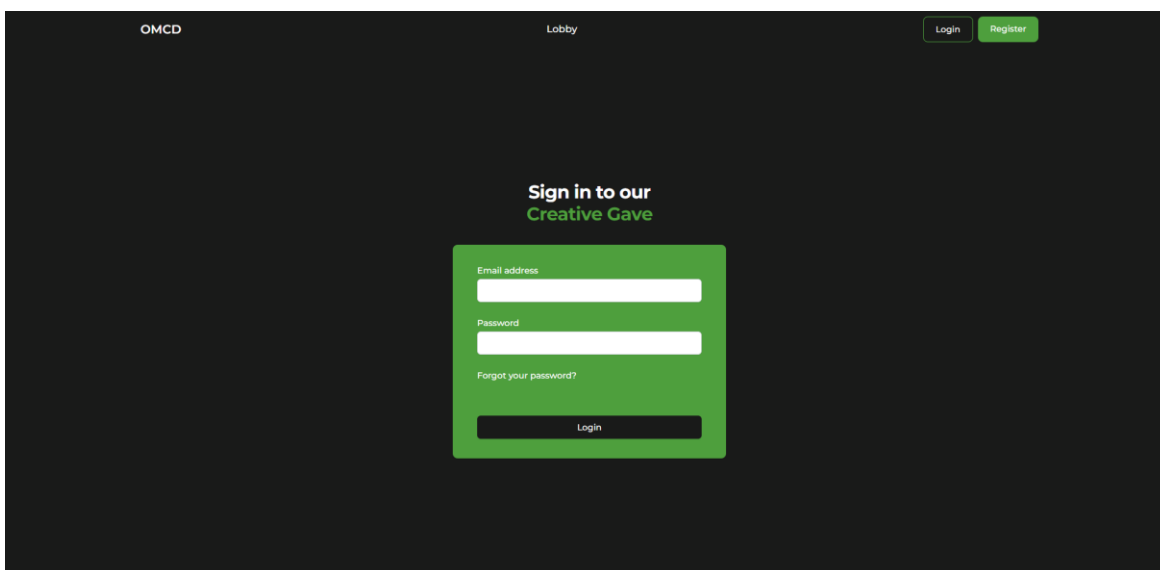


Εικόνα 30 - Στιγμιότυπο από το email επιβεβαίωσης ηλεκτρονικού ταχυδρομείου

Με την επιλογή “Verify Email” εντός της επόμενης μίας ώρας από την χρονική στιγμή που έκανε την εγγραφή, ο χρήστης μπορεί να εισέλθει στην πλατφόρμα χωρίς κάποιο πρόβλημα. Εφόσον δεν υπάρχει κάποιο πρόβλημα με την επαλήθευση, τότε η πλατφόρμα εμφανίζει μήνυμα επιτυχίας με την χρήση popup και κατευθύνει τον χρήστη στην σελίδα “Login”.

### 4.3 Είσοδος χρήστη

Στην σελίδα “Login” υπάρχει μία φόρμα για να συμπληρώσει ο χρήστης και να καταχωρήσει τα απαραίτητα στοιχεία για να συνδεθεί στην πλατφόρμα. Και σε αυτή την φόρμα υπάρχουν έλεγχοι στα πεδία της, που εμφανίζουν τα κατάλληλα μηνύματα σε περίπτωση επιτυχίας ή αποτυχίας. Αφού εισάγει σωστά τα στοιχεία του ο χρήστης τότε κατευθύνεται από την πλατφόρμα στην σελίδα “Lobby”.



Εικόνα 31 - Στιγμιότυπο από την σελίδα Login

Αφού ο χρήστης εισέλθει στην πλατφόρμα, το backend επιστρέφει στο frontend ένα κρυπτογραφημένο κλειδί που από αυτό το σημείο και έπειτα σε κάθε κλήση του χρήστη προς τον διακομιστή θα πρέπει να ενσωματώνεται στο αίτημα. Αυτό το κλειδί περιέχει στοιχεία του χρήστη που χρησιμοποιούνται για την ταυτοποίηση του χρήστη με τον διακομιστή.

Στο backend υπάρχει ενδιάμεσο λογισμικό (middleware) που μέσω αυτού γίνεται η ταυτοποίηση του χρήστη. Η Nest.js παρέχει φρουρούς (guards) ώστε να χρησιμοποιούνται οι κατάλληλοι στα APIs ανάλογα με την πρόσβαση που έχει ο ρόλος του κάθε χρήστη. Οι φρουροί που χρησιμοποιήθηκαν στην υφιστάμενη πλατφόρμα είναι οι εξής:

- **Ταυτοποίηση JWT**

Όταν γίνεται η κρυπτογράφηση του JWT, προσθέτετε μια συμβολοσειρά ώστε να διασφαλιστεί ότι το JWT δημιουργήθηκε από την υφιστάμενη πλατφόρμα. Αυτός ο φρουρός ελέγχει αν το JWT που παρέχει ο χρήστης στο αίτημα του είναι το ίδιο με την συμβολοσειρά που το δημιούργησε μέσω των πακέτων passport και passport-jwt.

Κώδικας:

```
@Injectable()
export class JwtStrategy extends PassportStrategy(Strategy) {
  constructor() {
    super({
      jwtFromRequest: ExtractJwt.fromAuthHeaderAsBearerToken(),
      ignoreExpiration: false,
      secretOrKey: process.env.JWT_SECRET,
    })
  }
}
```

```
}  
  async validate(payload: any) {  
    return { ...payload.user };  
  }  
}
```

- Έλεγχος refresh\_token

Με την είσοδο του χρήστη δημιουργείται ένας κωδικός μέσω της βιβλιοθήκης UUID (Universal unique Identifier), κρυπτογραφείται και αποθηκεύεται στην βάση δεδομένων. Αυτός ο φρουρός ελέγχει αν το κλειδί που έχει ο χρήστης είναι το ίδιο με αυτό.

#### Κώδικας:

```
@Injectable()  
export class RefreshToken implements CanActivate {  
  constructor(private reflector: Reflector,  
    private jwtService: JwtService,  
    @InjectRepository(UsersEntity)  
    private readonly usersRepository: Repository<UsersEntity>  
  ) { }  
  
  async canActivate(  
    context: ExecutionContext,  
  ): Promise<boolean> {  
    // Get the user from the request  
    const { user } = context.switchToHttp().getRequest();  
    // Verify user with refresh token  
    if (user.refresh_token) {  
      return await this.verifyToken(user);  
    }  
  }  
  
  async verifyToken(user: User) {  
    try {  
      // Verify JWT token  
      const jwt_rt = await this.jwtService.verifyAsync(  
        user.refresh_token,  
        { secret: process.env.JWT_RT_SECRET }  
      );  
    } catch {  
      return false;  
    }  
  }  
}
```



```
);  
  
// Hash the refresh token to check if is the same with the refresh  
token at database  
  
const refresh_token = this.hashPassword(jwt_rt.refresh_token)  
const refresh_token_exists = await this.usersRepository.findOne({  
where: { refresh_token } });  
  
// If refresh token exists accept the user  
if (refresh_token_exists) {  
    return true  
}  
  
} catch (error) {  
    throw new HttpException({ status: HttpStatus.FORBIDDEN, error:  
'Unauthorized' }, HttpStatus.FORBIDDEN);  
}  
  
}  
  
hashPassword(token: string) {  
    return crypto.createHash('sha256').update(token).digest('hex');  
}  
  
}
```

- Έλεγχος ρόλου χρήστη

Ο συγκεκριμένος φρουρός διαβάζει τον ρόλο που έχει ο χρήστης, και αν ο ρόλος του είναι αυτός που αναγράφεται πάνω από το API, τότε δίνει πρόσβαση στο χρήστη να το χρησιμοποιήσει.

#### Κώδικας:

```
@Injectable()  
export class RolesGuard implements CanActivate {  
    constructor(private reflector: Reflector) { }  
  
    canActivate(  
        context: ExecutionContext,  
    ): boolean | Promise<boolean> | Observable<boolean> {  
        // Get the require roles from the request  
        const requiredRoles = this.reflector.getAllAndOverride<Role[]>(ROLES_KEYS, [  
            context.getHandler(),
```

```

    context.getClass()
  ])
  // if function does not have roles permissions return true
  if (!requiredRoles) {
    return true;
  }
  // Get the user from the request
  const { user } = context.switchToHttp().getRequest();
  // check if the user has any from the require roles
  return requiredRoles.some(role => user.role?.includes(role));
}
}

```

- Έλεγχος ανώνυμου χρήστη

Καθώς πλέον στην εφαρμογή τα περισσότερα APIs γίνονται από εξουσιοδοτημένους χρήστες, ο φρουρός για το έλεγχο του JWT υπάρχει σε όλες τις κλήσεις χωρίς να προστεθεί επιπλέον. Ο φρουρός δημιουργήθηκε για τους ανώνυμους χρήστες ώστε να μπορούν να κάνουν αίτημα στο διακομιστή για τα APIs που δεν χρειάζονται ταυτοποίηση.

Κώδικας:

```

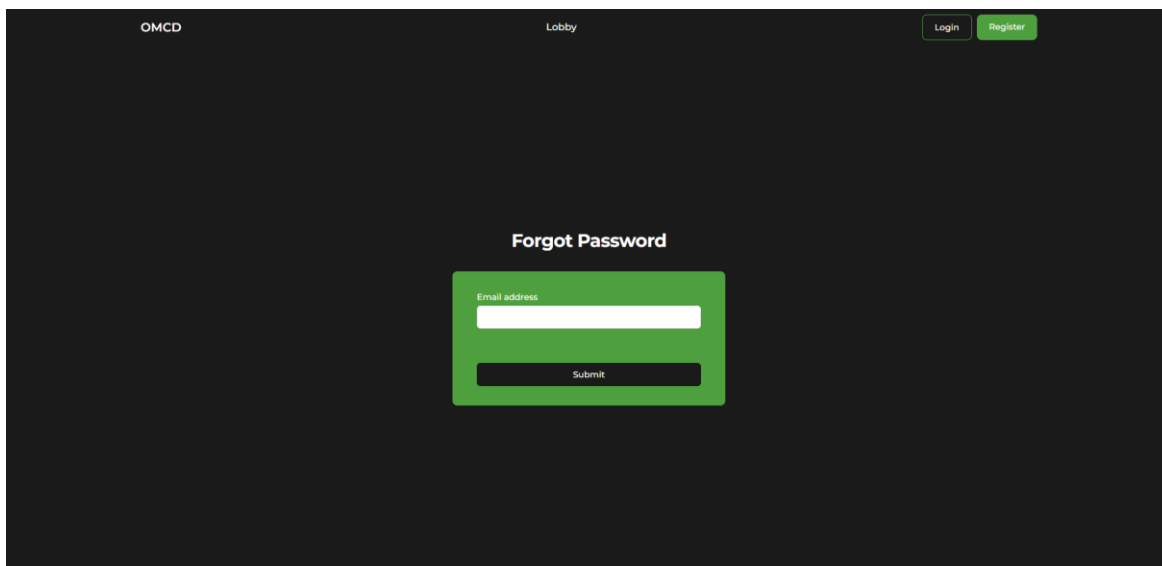
@Injectable()
export class JwtGuard extends AuthGuard('jwt') {
  constructor(private reflector: Reflector) {
    super();
  }

  canActivate(context: ExecutionContext) {
    const isPublic = this.reflector.getAllAndOverride<boolean>(IS_PUBLIC_KEY, [
      context.getHandler(),
      context.getClass(),
    ]);
    if (isPublic) {
      return true;
    }
    return super.canActivate(context);
  }
}

```

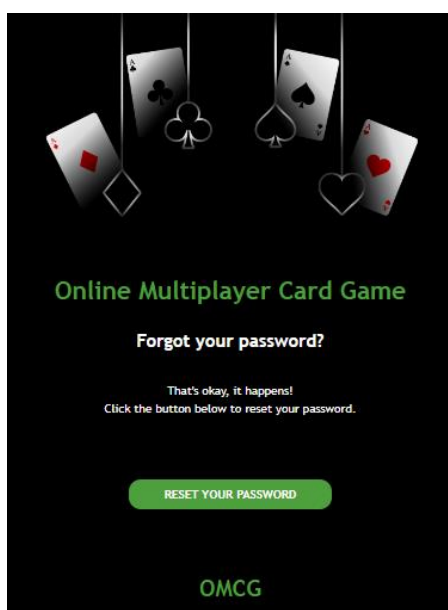
## 4.4 Επανάδοση κωδικού πρόσβασης

Αν ο χρήστης δεν μπορεί να συνδεθεί στην πλατφόρμα γιατί έχει ξεχάσει τον κωδικό πρόσβασης του, τότε η πλατφόρμα του δίνει την δυνατότητα να αιτηθεί την επανάδοση του κωδικού πρόσβασης. Αυτή την δυνατότητα μπορεί να την αιτηθεί από την σελίδα “Login”, με την επιλογή “Forgot your password?”.



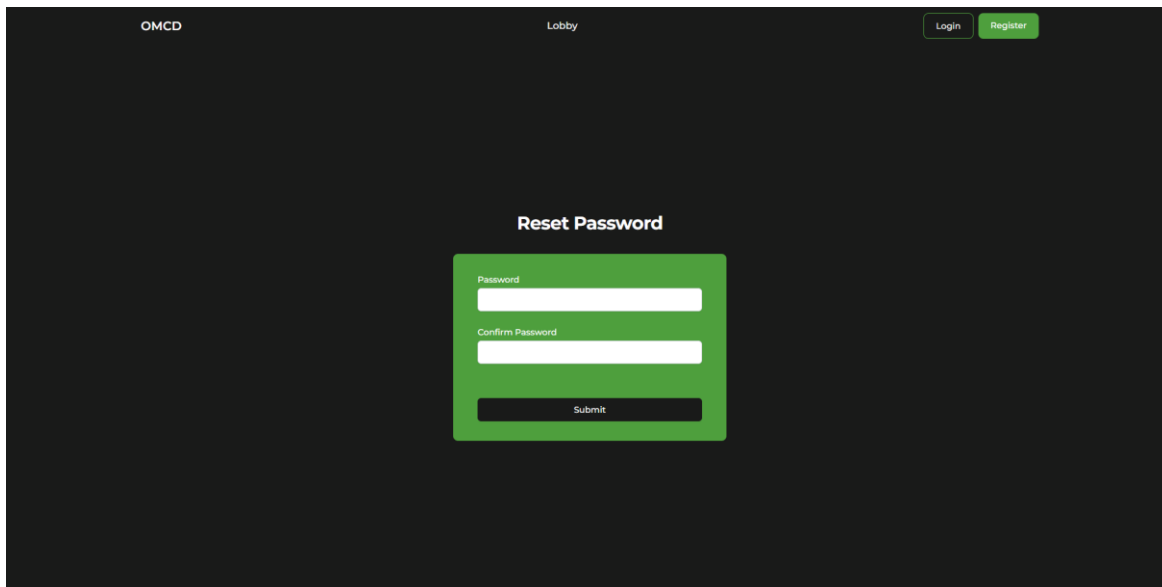
Εικόνα 32 - Στιγμιότυπο από την σελίδα Forgot Password

Ο χρήστης πρέπει να συμπληρώσει το ηλεκτρονικό του ταχυδρομείο. Αν το ηλεκτρονικό του ταχυδρομείο υπάρχει στην βάση δεδομένων τότε θα του αποσταλεί μήνυμα στο ηλεκτρονικό ταχυδρομείο ώστε να επιβεβαιώσει την ενέργειά αυτή και να κατευθύνει στην σελίδα “Reset Password”. Αν δεν υπάρχει τότε θα ενημερωθεί με το κατάλληλο μήνυμα. Το μήνυμα που λαμβάνει είναι το εξής:



Εικόνα 33 - Στιγμιότυπο από το email επιβεβαίωσης αλλαγής κωδικού πρόσβασης

Ο χρήστης πρέπει να επιλέξει την επιλογή “Reset your password” εντός της επόμενης μίας ώρας από την χρονική στιγμή που έκανε το αίτημα και στην συνέχεια η πλατφόρμα τον κατευθύνει στην σελίδα που φαίνεται παρακάτω, “Reset password”.

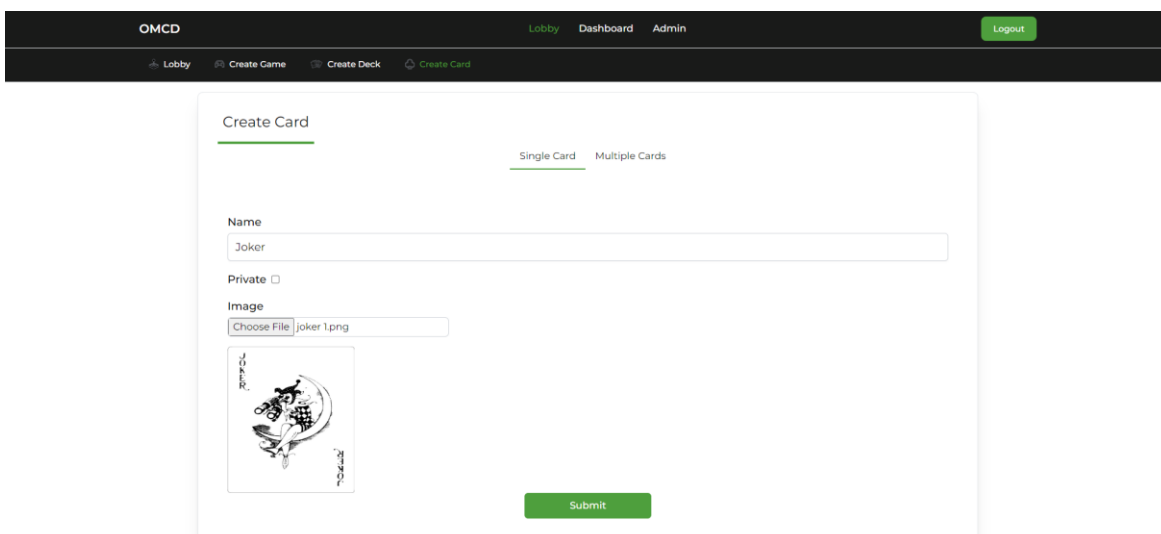


Εικόνα 34 - Στιγμιότυπο από την σελίδα Reset Password

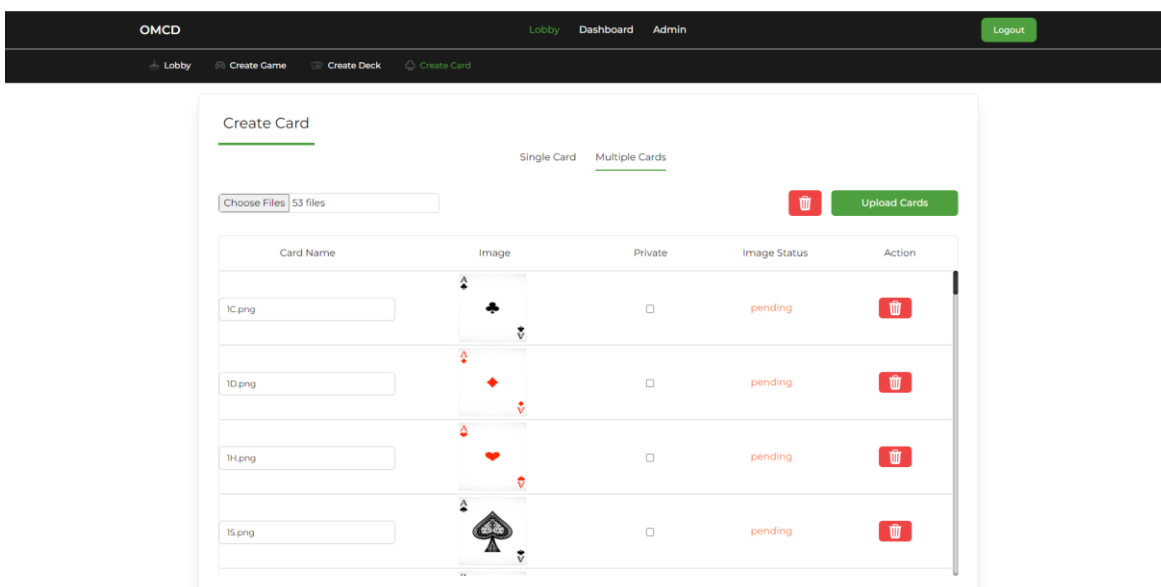
Σε αυτή την σελίδα ο χρήστης πρέπει να συμπληρώσει την φόρμα για την αλλαγή του κωδικού πρόσβασης του και να υποβάλει την αίτηση του. Θα ειδοποιηθεί με μήνυμα αποτυχίας ή επιτυχίας ανάλογα με την αίτηση του και αν συμπληρωθεί επιτυχώς, τότε θα κατευθυνθεί στην σελίδα “Login”.

## 4.5 Δημιουργία κάρτας

Στην σελίδα δημιουργία κάρτας έχουν πρόσβαση μόνο οι εγγεγραμμένοι χρήστες καθώς η δημιουργία κάρτας είναι ικανότητα που δεν προσφέρεται στους ανώνυμους χρήστες. Ο χρήστης μπορεί να ανεβάσει μία-μία την κάρτα ή να επιλέξει να ανεβάσει πολλαπλές εικόνες ταυτόχρονα. Για οποιαδήποτε από τις πιο πάνω επιλογές που έχει ο χρήστης, πρέπει να δώσει ένα όνομα στην κάρτα και να την ορίσει αν θα είναι διαθέσιμη μόνο για εκείνο ή και για άλλους χρήστες. Αφού ολοκληρωθεί το ανέβασμα της κάρτας ή των καρτών, λαμβάνει μήνυμα επιτυχίας ή αποτυχίας.



Εικόνα 35 - Στιγμιότυπο από το ανέβασμα μίας κάρτας στην σελίδα Create Card

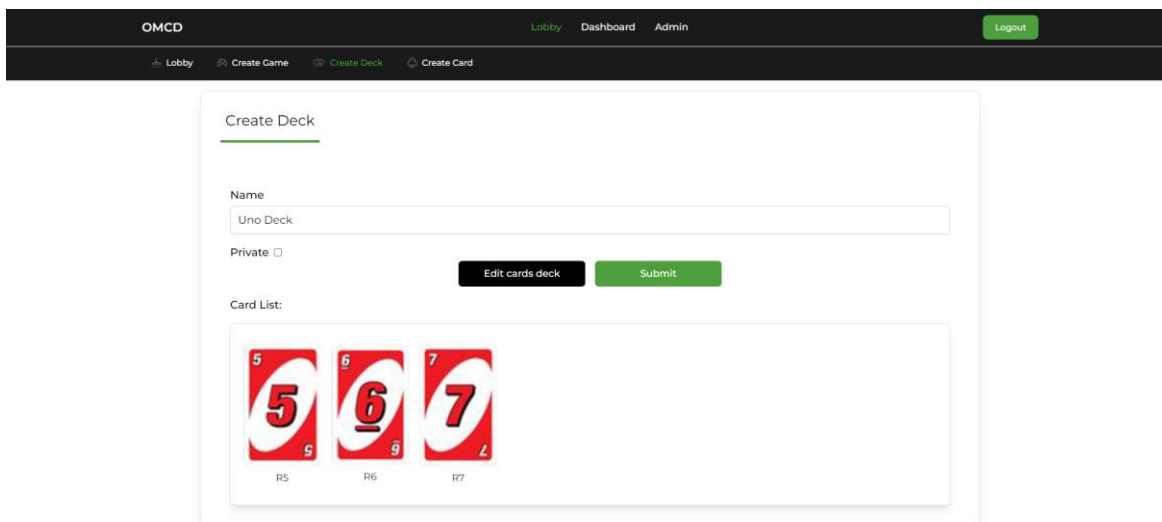


Εικόνα 36 - Στιγμιότυπο από το ανέβασμα πολλών καρτών στην σελίδα Create Card

Οι εικόνες αποθηκεύονται στο backend με ένα συγκεκριμένο όνομα με βάση την ημερομηνία που έγινε η εγγραφή της κάρτας και αυτό το όνομα αποθηκεύεται στην βάση δεδομένων. Πριν γίνει η αποθήκευση γίνεται συμπύκνωση στην εικόνα ώστε να διασφαλιστεί ότι δεν θα καταναλώνουν μεγάλο χώρο οι εικόνες και έτσι θα φορτώνονται και στις διεπαφές γρήγορα. Οι αποδεκτοί τύποι εικόνας είναι το jpg, jpeg και png με μέγιστο μέγεθος χωρητικότητας 2MB.

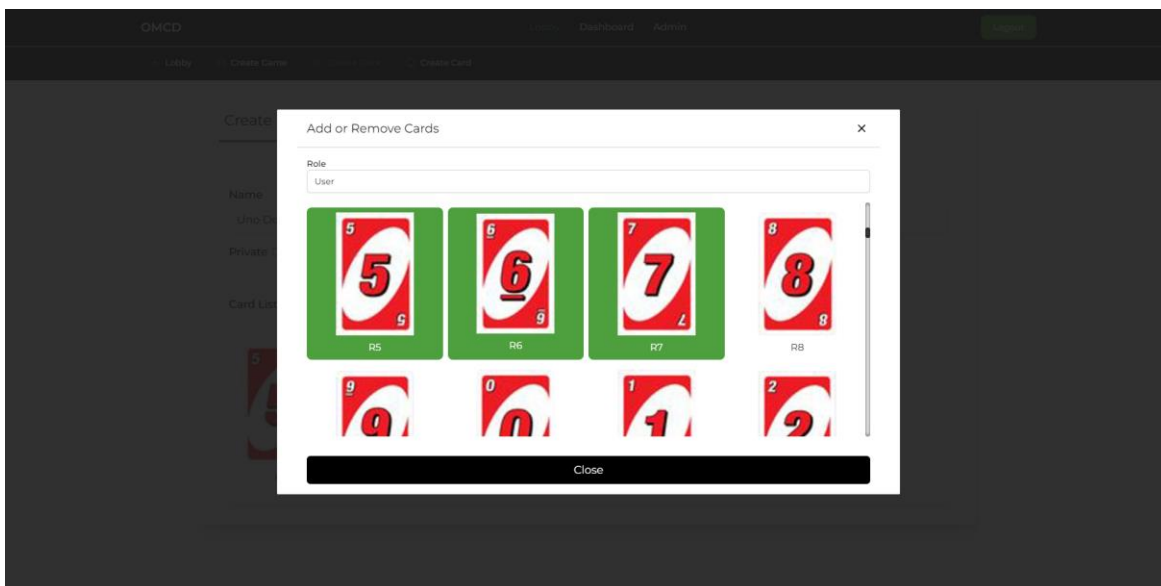
## 4.6 Δημιουργία τράπουλας

Στην σελίδα “Create Deck” ο εγγεγραμμένος χρήστης θα μπορεί να δημιουργήσει την δική του τράπουλα. Δεν είναι προϋπόθεση ο χρήστης να έχει ανεβάσει κάρτες για να δημιουργήσει μία τράπουλα, καθώς υπάρχουν και διαθέσιμες κάρτες από άλλους χρήστες.



Εικόνα 37 - Στιγμιότυπο από την σελίδα Create Deck

Όπως και στην φόρμα δημιουργία κάρτας, έτσι και τώρα ο χρήστης πρέπει να δώσει ένα όνομα στην τράπουλα και να επιλέξει αν θα είναι ιδιωτική. Στην συνέχεια επιλέγοντας το κουμπί “Edit cards deck” εμφανίζεται ένα αναδυόμενο παράθυρο με τις διαθέσιμες κάρτες που μπορεί να επιλέξει. Μπορεί να προσθέσει είτε κάρτες που ανέβασε ο ίδιος είτε κάρτες που ανέβασαν άλλοι χρήστες και δεν είναι ιδιωτικές. Ο χρήστης επιλεγεί κάρτες και αυτές προστίθενται αυτόματα στην τράπουλα. Με πράσινο χρώμα είναι οι επιλεγμένες κάρτες όπως φαίνονται στην παρακάτω εικόνα.



Εικόνα - Στιγμιότυπο από το αναδυόμενο παράθυρο για την επιλογή καρτών στην σελίδα Create deck

Εφόσον ολοκληρώσει την επιλογή του, κλείνει το popup και ο χρήστης πρέπει να ολοκληρώσει την αίτηση του με την επιλογή του κουμπιού “Submit”.

#### 4.7 Δημιουργία παιχνιδιού

Αυτό το σημείο είναι πολύ καθοριστικό καθώς σε αυτό βρίσκονται όλες οι ρυθμίσεις του παιχνιδιού που θα φορτωθούν σε ένα ΗΤ και αφορούν το ίδιο το τραπέζι, το παιχνίδι και τους παίκτες. Παιχνίδι μπορεί να δημιουργήσει ο κάθε εγγεγραμμένος χρήστης με τον ρόλο του Game Designer. Ο GD, με τις επιλογές του, έχει την δυνατότητα να δημιουργήσει ένα ευχάριστο και φιλικό τραπέζι προς τους χρήστες διευκολύνοντας έτσι και το έργο που θα έχει ο GM στο ΗΤ.

Η φόρμα “Create Game” περιέχει αρκετές ρυθμίσεις ώστε να μπορεί να καλύψει, όσο περισσότερο γίνεται, τις απαιτήσεις που έχουν τα περισσότερα ΠΜΚ. Για αυτό τον λόγο, η φόρμα χωρίζεται σε τρία βήματα, για να μην υπερβαίνει το μέγεθος της σελίδας και να είναι φιλική προς τον χρήστη.

Στο πρώτο βήμα (1. Game) ο χρήστης πρέπει να συμπληρώσει τα βασικά στοιχεία του παιχνιδιού σχετικά με το όνομα που θα έχει, τον μέγιστο αριθμό των παιχτών που θα το αποτελούν, και η εμφάνιση του πλέγματος του τραπέζιού. Με την επιλογή στο πλέγμα της τιμής μηδέν για τις στήλες και τις γραμμές, δεν θα υπάρχει πλέγμα στο τραπέζι παρά μόνο η στοίβα του παίχτη και οι στοίβες με τις τράπουλες. Στην συνέχεια έχει να προσθέσει τις τράπουλες με κάρτες που θα έχει το παιχνίδι, με ελάχιστη επιλογή αυτή της μίας τράπουλας. Ο χρήστης μπορεί να επιλέξει ανάμεσα σε δικές του τράπουλες ή σε κοινόχρηστες τράπουλες από άλλους χρήστες. Για την ολοκλήρωση του πρώτου βήματος απομένει ο χρήστης να επιλέξει αν το παιχνίδι θα είναι ιδιωτικό και αν θα επιλέγεται αυτόματα ο επόμενος παίχτης όταν ολοκληρώσει τον γύρο του ο προηγούμενος παίχτης.

Με την συμπλήρωση της φόρμας του πρώτου βήματος δημιουργείται στην βάση δεδομένων ένα στιγμιότυπο με τις απλές ρυθμίσεις του παιχνιδιού.

Εικόνα 38 - Φόρμα δημιουργίας παιχνιδιού - Βήμα 1

Στο επόμενο βήμα (2. More Settings) ο χρήστης έχει την δυνατότητα να συμπληρώσει την φόρμα ή να συνεχίσει με τις προκαθορισμένες ρυθμίσεις οι οποίες είναι υποχρεωτικές. Οι επιλογές που έχει ο χρήστης στην φόρμα είναι οι εξής:

- **Ρόλοι:** Προσδιορίζει τον ρόλο που θα έχει ο παίχτης στο τραπέζι. Για παράδειγμα σε ένα παιχνίδι Blackjack υπάρχουν οι παίχτες και ο ντίλερ. Στους ρόλους υπάρχει ο προκαθορισμένος ρόλος “Παίχτης” (“Player”) καθώς κάθε παιχνίδι αποτελείται από παίχτες.
- **Άδειες τράπουλες:** Στο πρώτο βήμα της φόρμας δημιουργίας του παιχνιδιού επιλέχθηκαν οι τράπουλες οι οποίες περιέχουν κάρτες και πρέπει να μοιραστούν σε παίχτες ή σε τράπουλες. Στην κατηγορία άδειες τράπουλες δίνεται η δυνατότητα στον GD να μοιράσει κάρτες σε επιπλέον τράπουλες με την έναρξη του παιχνιδιού. Στην κατηγορία αυτή ωστόσο υπάρχουν και ήδη δυο υπάρχουσες υποχρεωτικές άδειες τράπουλες, οι οποίες είναι η στοίβα του τραπέζιού (“Table”) και η στοίβα με τα σκάρτα (“Junk”). Ο χρήστης μπορεί να προσθέσει και επιπλέον άδειες τράπουλες. Για παράδειγμα σε ένα παιχνίδι όπως την μπιρίμπα, υπάρχουν δύο επιπλέον τράπουλες (μπιριμπάκια) που δεν ανήκουν σε κάποιο παίχτη. Έτσι με αυτή την επιλογή στην συγκεκριμένη περίπτωση θα υπάρχουν και αυτές οι δύο τράπουλες στο HT.
- **Καταστάσεις παιχτών:** Σε ένα παιχνίδι UNO ένας παίχτης οφείλει να ενημερώσει τους αντιπάλους του, όταν του απομείνει ένα χαρτί λέγοντας την λέξη “UNO”. Στην συγκεκριμένη πλατφόρμα ο παίχτης θα ενημερώνει μέσω της κατάστασης του τους υπόλοιπους παίχτες, επομένως ο Game Designer θα πρέπει να προσδιορίσει τις καταστάσεις των παιχτών που θα χρειαστούν σε ένα παιχνίδι σε αυτό το σημείο της φόρμας.



- **Ομάδες:** Σε αυτό το μέρος της φόρμας θα προστεθούν οι ονομασίες των ομάδων που έχει ένα παιχνίδι ώστε να γνωρίζουν οι παίκτες στο ΗΤ ποιοι θα είναι οι συμπαίχτες τους.

Με την ολοκλήρωση του δεύτερου βήματος οι πληροφορίες αποθηκεύονται στην βάση των δεδομένων και φορτώνονται στο επόμενο και τελευταίο βήμα της φόρμας.

Εικόνα 39 - Φόρμα δημιουργίας παιχνιδιού - Βήμα 2

Στο τελευταίο βήμα της φόρμας (3. Starting cards), ο GD θα δηλώσει τον τρόπο που θα μοιραστούν οι κάρτες στους παίκτες και στις άδειες τράπουλες με την έναρξη του παιχνιδιού. Αυτή η φόρμα περιέχει ένα πίνακα που αποτελείται από 6 στήλες. Όλες οι στήλες εκτός από την τρίτη είναι πάντα οι ίδιες σε οποιαδήποτε επιλογή του χρήστη. Οι επιλογές που έχει να συμπληρώσει ο χρήστης είναι οι εξής:

- 1η στήλη: Στην πρώτη στήλη ο GD έχει να επιλέξει αν οι κάρτες θα μοιραστούν σε κάποιο συγκεκριμένο ρόλο ή σε κάποια άδεια τράπουλα.
- 2η στήλη: Στην δεύτερη στήλη ο GD επιλέγει από ποια τράπουλα θα μοιράσει τις κάρτες.
- 3η στήλη: Αν ο GD έχει επιλέξει στην πρώτη στήλη να μοιράσει κάρτες σε κάποιο ρόλο τότε εδώ θα επιλέξει σε ποιο ρόλο παίχτη θα δώσει τις κάρτες.

Αν ο GD έχει επιλέξει στην πρώτη στήλη να μοιράσει κάρτες σε άδειες τράπουλες τότε θα πρέπει να επιλέξει σε ποια τράπουλα θα μοιραστούν οι κάρτες.

- **4η στήλη:** Στην τέταρτη στήλη ο GD επιλέγει αν οι κάρτες θα είναι κρυφές ή φανερές στους άλλους παίχτες.
- **5η στήλη:** Στην πέμπτη στήλη ο GD προσδιορίζει των αριθμών των καρτών που θα μοιραστούν σε κάποιο παίχτη ή τράπουλα.
- **6η στήλη:** Η τελευταία στήλη είναι ενέργειες που έχει ο GD σε αυτό το στάδιο της φόρμας. Μπορεί να προσθέσει ή να αφαιρέσει γραμμές του πίνακα του οποίου η κάθε γραμμή του σχετίζεται με μία οδηγία για το πως θα μοιραστούν οι κάρτες.

Εικόνα 40 - Φόρμα δημιουργίας παιχνιδιού - Βήμα 3

Στην παραπάνω εικόνα (Εικόνα 40) ο GD έχει δηλώσει ότι θα μοιραστούν επτά κάρτες σε κάθε χρήστη που έχει το ρόλο “Player” και θα είναι κρυφές. Μία κάρτα θα δοθεί στην τράπουλα “Junk” και θα είναι φανερή σε όλους του παίχτες.

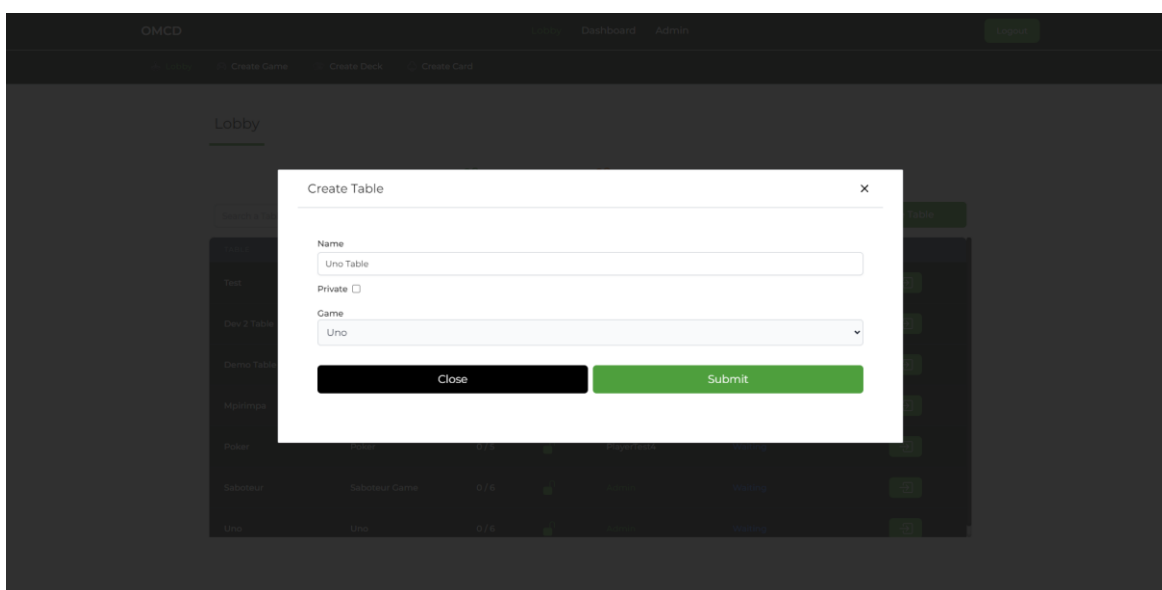
Σε αυτό το σημείο της φόρμας ο GD έχει την επιλογή να μην προσθέσει κάποια οδηγία για το πως θα μοιραστούν οι κάρτες και να προχωρήσει στην ολοκλήρωση της δημιουργίας του παιχνιδιού. Ο χρήστης ενημερώνεται με το κατάλληλο μήνυμα αν πραγματοποιήθηκε επιτυχώς η δημιουργία του παιχνιδιού ή με μήνυμα αποτυχίας για το σημείο που πρέπει να διορθώσει.

## 4.8 Δημιουργία ηλεκτρονικού τραπεζιού

Η φόρμα με την δημιουργία του HT βρίσκεται στον προθάλαμο του χρήστη με την επιλογή του κουμπιού “Create Table”. Με την επιλογή του κουμπιού, ανοίγει ένα αναδυόμενο παράθυρο που έχει την φόρμα της δημιουργίας του HT. Αυτή η φόρμα είναι πολύ απλή καθώς ο χρήστης έχει να συμπληρώσει υποχρεωτικά ένα όνομα για το HT και

να επιλέξει ένα παιχνίδι. Παιχνίδι μπορεί να επιλέξει είτε από αυτά που έχει δημιουργήσει ο ίδιος, είτε από αυτά που έχουν δημιουργήσει άλλοι χρήστες και τα έχουν ως κοινόχρηστα. Επιπλέον, ο χρήστης έχει την επιλογή να κάνει το HT ιδιωτικό και να προσθέσει ένα κωδικό πρόσβασης ώστε να προαπαιτείτε πριν την είσοδο του κάθε χρήστη στο συγκεκριμένο HT.

Εφόσον συμπληρωθεί η φόρμα σωστά τότε το HT έχει κατάσταση αναμονής (“waiting”) και προθέεται στην λίστα με τα HT που είναι διαθέσιμα για να εισέλθουν οι χρήστες στο προθάλαμο της πλατφόρμας. Για να επιτευχθεί αυτή η άμεση ενημέρωση στους άλλους χρήστες, η δημιουργία του τραπέζιου έγινε με την χρήση των Web Sockets. Με την δημιουργία του HT, ο χρήστης που δημιούργησε το τραπέζι ορίζεται ως Game Master και η πλατφόρμα τον κατευθύνει στο HT.

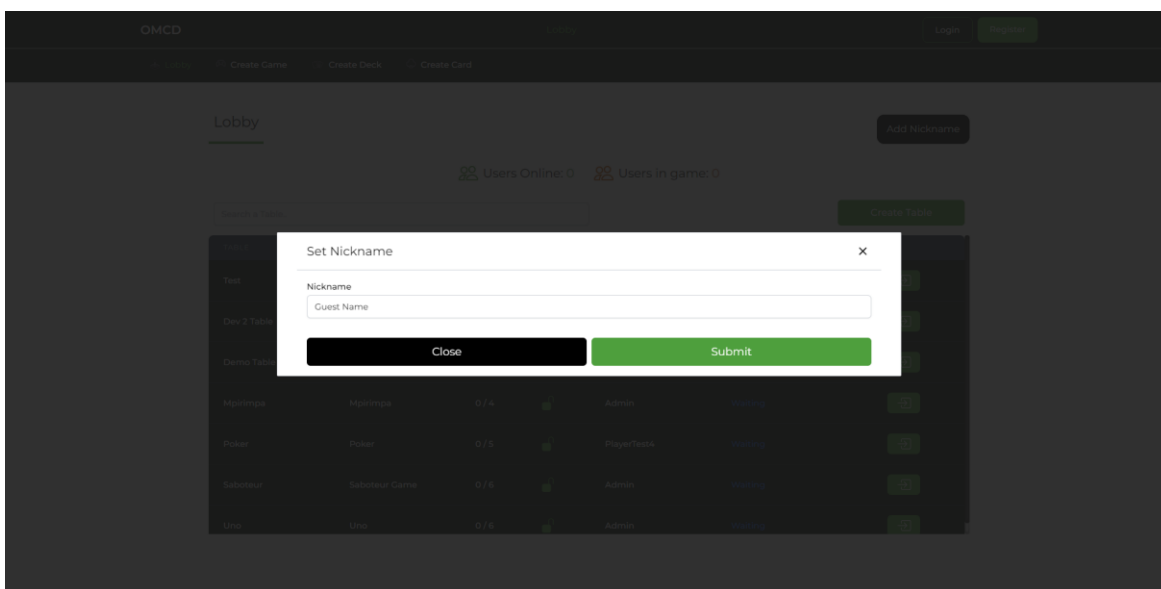


Εικόνα 41 - Φόρμα δημιουργίας ηλεκτρονικού τραπέζιου

## 4.9 Προθάλαμος χρήστη

Η σελίδα προθάλαμος χρήστη (“Lobby”) βρίσκεται στο μενού της σελίδας και έχουν πρόσβαση όλοι οι χρήστες της πλατφόρμας. Με το που εισέλθει κάποιος χρήστης στο “Lobby”, η πλατφόρμα ελέγχει αν ο χρήστης είναι συνδεδεμένος. Αν ο χρήστης είναι εγγεγραμμένος και είναι συνδεδεμένος στην πλατφόρμα τότε έχει κανονική πρόσβαση στην σελίδα και έχει την δυνατότητα να δημιουργήσει HT. Στην περίπτωση που ο χρήστης δεν είναι συνδεδεμένος και δεν έχει κάποιο λογαριασμό, τότε εμφανίζεται ένα αναδυόμενο παράθυρο στο οποίο ζητείτε από τον χρήστη να καταχωρήσει ένα όνομα ως ψευδώνυμο. Το ψευδώνυμο του χρήστη καταχωρείται στην βάση δεδομένων στο πεδίο username του πίνακα users με τον ρόλο του επισκέπτη (“guest”). Το πεδίο username δέχεται μοναδικές τιμές. Σε περίπτωση που το ψευδώνυμο χρησιμοποιείται από άλλο χρήστη τότε εκτυπώνεται το κατάλληλο μήνυμα ενημέρωσης στο χρήστη. Ο επισκέπτης αν απορρίψει το αναδυόμενο παράθυρο και δεν έχει δημιουργήσει λογαριασμό ως επισκέπτης, τότε σε κάθε είσοδο του συγκεκριμένου επισκέπτη σε κάποιο από τα HT θα του εμφανίζεται το αναδυόμενο παράθυρο για εισαγωγή ψευδώνυμου μέχρι να

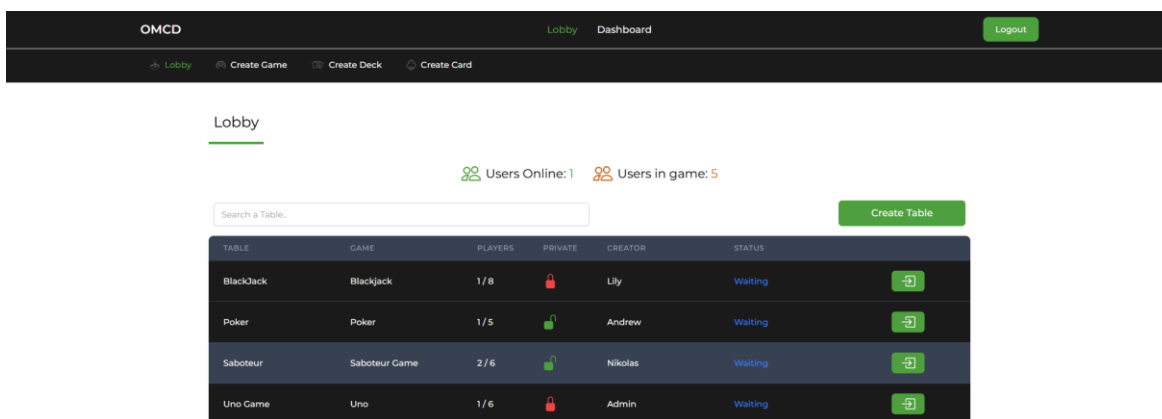
καταχωρήσει επιτυχώς ένα ψευδώνυμο. Το αναγνωριστικό του επισκέπτη αποθηκεύεται στο αποθηκευτικό χώρο του προγράμματος περιήγησης, και στην επόμενη είσοδο του επισκέπτη θα έχει το ίδιο όνομα. Ο επισκέπτης μπορεί ανά πάσα στιγμή να αλλάξει το ψευδώνυμο του από το κουμπί “Change Nickname” που βρίσκεται πάνω αριστερά στην σελίδα. Εφόσον ο επισκέπτης δημιουργήσει ψευδώνυμο επιτυχώς, τότε μπορεί να εισέλθει σε HT αλλά δεν μπορεί να δημιουργήσει κάποιο HT.



Εικόνα 42 - Φόρμα εισαγωγής ψευδώνυμου για τον ανώνυμο χρήστη

Στο πίνακα του προθάλαμου εμφανίζονται τα HT που είναι διαθέσιμα για να εισέλθουν χρήστες. Τα στοιχεία του HT που εμφανίζονται είναι το όνομα του, το όνομα του παιχνιδιού, πόσοι παίκτες υπάρχουν στο HT, αν το HT είναι ιδιωτικό, τον δημιουργό του και η κατάσταση του HT. Όλη η σελίδα του προθάλαμου χρησιμοποιεί Web Sockets σύνδεση, επομένως όλες οι αλληλοεπιδράσεις που έχουν οι χρήστες με την σελίδα γίνονται σε πραγματικό χρόνο όπως και η ενημέρωση όλων των χρηστών. Για παράδειγμα όταν ένα χρήστης εισέλθει σε κάποιο HT τότε θα αυξήσει αυτόματα ο αριθμός των χρηστών που υπάρχουν στο συγκεκριμένο HT.

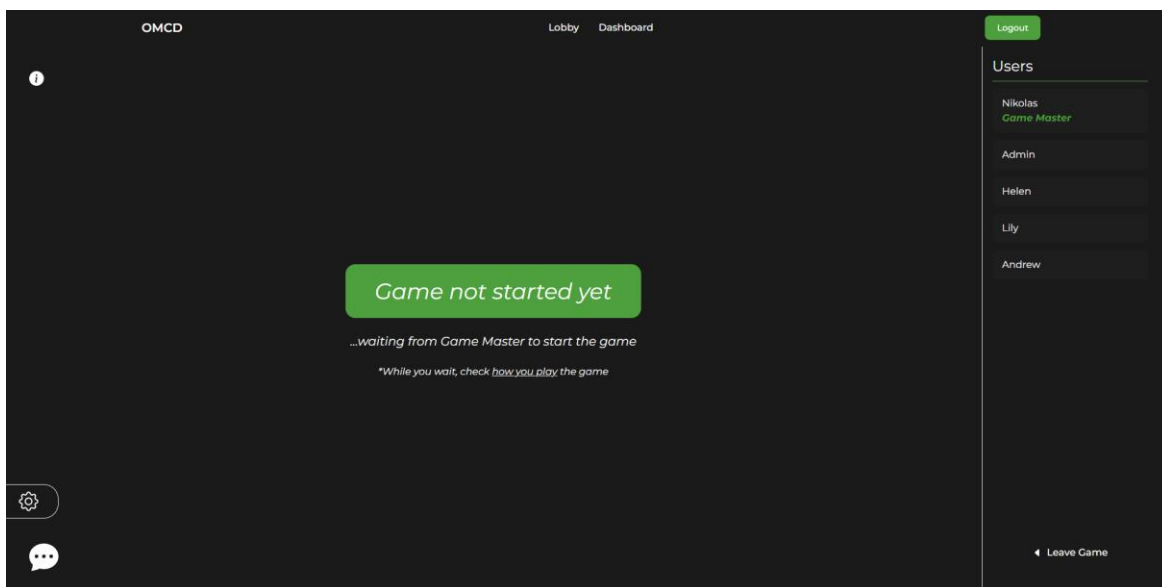
Για να εισέλθει κάποιος χρήστης σε κάποιο HT το μόνο που έχει να κάνει είναι να επιλέξει το τραπέζι και στην συνέχεια να επιλέξει το κουμπί “Join Room”. Αν το τραπέζι δεν είναι ιδιωτικό τότε θα εισέλθει αυτόματα αλλιώς θα του εμφανιστεί ένα αναδυόμενο παράθυρο για να πληκτρολογήσει το κωδικό πρόσβασης του HT. Αν είναι σωστός ο κωδικός πρόσβασης, τότε ο χρήστης θα εισέλθει στο HT εάν όχι, θα του εμφανιστεί το κατάλληλο μήνυμα για να δοκιμάσει πάλι. Επιπλέον, ο δημιουργός μπορεί να αφαιρέσει το HT από την λίστα με τα HT επιλέγοντας το κουμπί της διαγραφής που εμφανίζεται στο HT, υπό την προϋπόθεση ότι το έχει δημιουργήσει ο ίδιος.



Εικόνα 43 - Προθάλαμος χρήστη

## 4.10 Ηλεκτρονικό τραπέζι

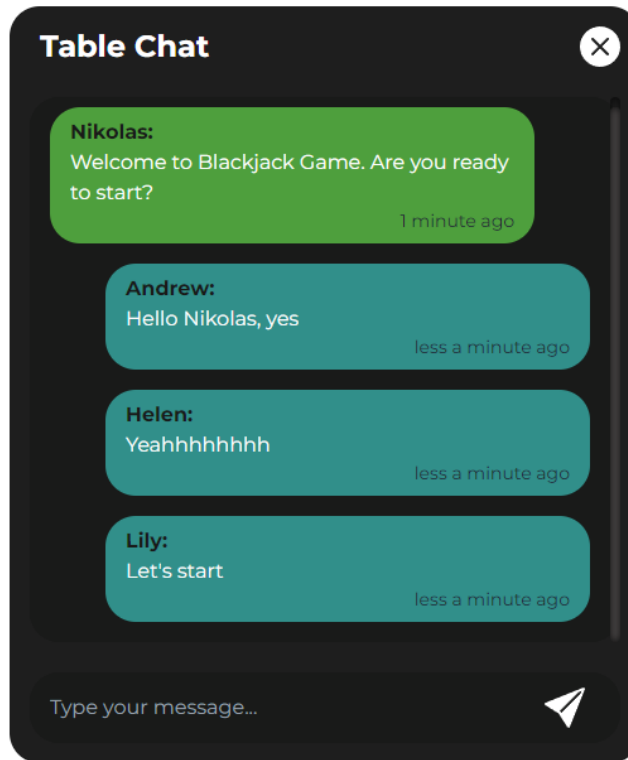
Το HT διακρίνεται από ένα μοναδικό κωδικό από το πακέτο UUID (Universally Unique Identifier) που προστίθεται στην διεύθυνση της σελίδας του δωματίου. Στο τραπέζι μπορεί να εισέλθει κάποιος μόνο μέσω του προθάλαμου καθώς η σελίδα του HT χρειάζεται κάποιες πληροφορίες που φορτώνονται μόνο από τον προθάλαμο για να διασφαλιστεί έτσι η ασφάλεια του HT. Εφόσον φορτωθεί το HT τότε εμφανίζεται στο χρήστη ένα μήνυμα αναμονής ότι το HT αναμένει από τον GM να ξεκινήσει το παιχνίδι.



Εικόνα 44 - Ηλεκτρονικό τραπέζι σε κατάσταση αναμονής

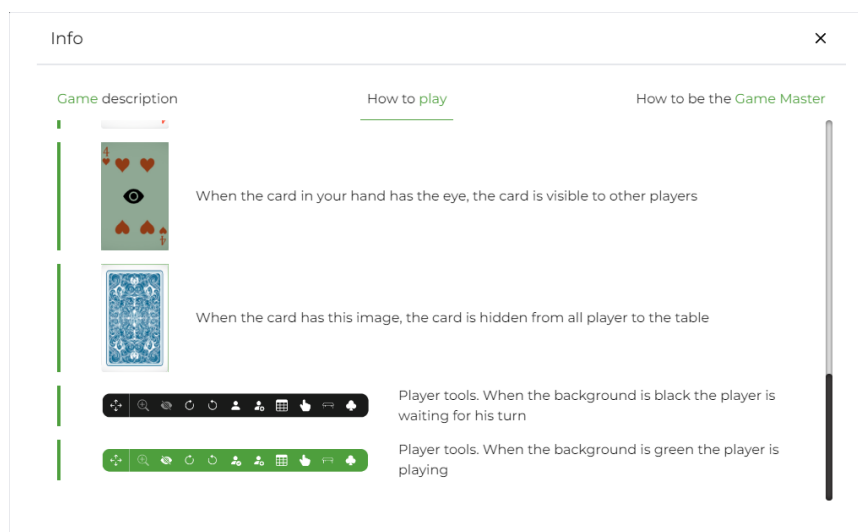
Στο δεξί μέρος της σελίδα υπάρχει ένα sidebar, που εμφανίζει τους χρήστες που είναι συνδεδεμένοι στο HT και ποιος είναι ο GM του HT. Στο αριστερό κάτω μέρος υπάρχει ένα εικονίδιο συνομιλίας που δίνει την δυνατότητα στους χρήστες που είναι στο τραπέζι να ανταλλάξουν μηνύματα μεταξύ τους καθ' όλη την διάρκεια του HT. Αν ο χρήστης είναι

ο GM του τραπέζιου, του εμφανίζεται ένα επιπλέον μενού με το εικονίδιο των ρυθμίσεων που μπορεί να προβεί σε περετέρων ρυθμίσεις που έχουν να κάνουν με το ΗΤ, το παιχνίδι και τους παίχτες.



Εικόνα 45 - Γραπτή συνομιλία του ηλεκτρονικού τραπεζιού

Στο πάνω αριστερά μέρος υπάρχει το εργαλείο των πληροφοριών που εμφανίζει ένα αναδυόμενο παράθυρο που αναλύει την περιγραφή που πρόσθεσε ο GD στο παιχνίδι και πως μπορεί ένας παίχτης ή ο GM να παίξει στο τραπέζι επεξηγώντας τους τα εργαλεία που έχουν στην διάθεση τους.



Εικόνα 46 - Αναδυόμενο παράθυρο των πληροφοριών του ηλεκτρονικού τραπεζιού

#### 4.10.1 Μενού Game Master

Όταν ο χρήστης μετακινήσει το ποντίκι του πάνω από το μενού το ρυθμίσεων του GM, τότε ανοίγει η μπάρα με τα εργαλεία που έχει ο GM στο HT όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 47 - Μενού εργαλείων του Game Master

Το μενού χωρίζεται σε 4 ομάδες, η πρώτη ομάδα αφορά εργαλεία σχετικά με το τραπέζι, η δεύτερη ενέργειες για το παιχνίδι και τους παίκτες, η τρίτη αφορά τους χρήστες και η τελευταία αφορά το μενού του GM. Όταν το ποντίκι είναι πάνω από κάποιο εργαλείο εμφανίζεται σε μήνυμα (tooltip) η ενέργεια που κάνει το εικονίδιο.










Οι ενέργειες των εργαλείων είναι οι παρακάτω:

⊕ Αυτό το εργαλείο είναι για την έναρξη του παιχνιδιού (“New Game”). Όταν ο GM το επιλέξει τότε θα εμφανιστεί το τραπέζι σε όλους του παίκτες και θα δοθούν οι κάρτες στους παίκτες και τις άδειες τράπουλες με βάση τις οδηγίες που έθεσε ο GD στις ρυθμίσεις του παιχνιδιού. Με την έναρξη του παιχνιδιού το HT αλλάζει σε κατάσταση παίζεται (“playing”).

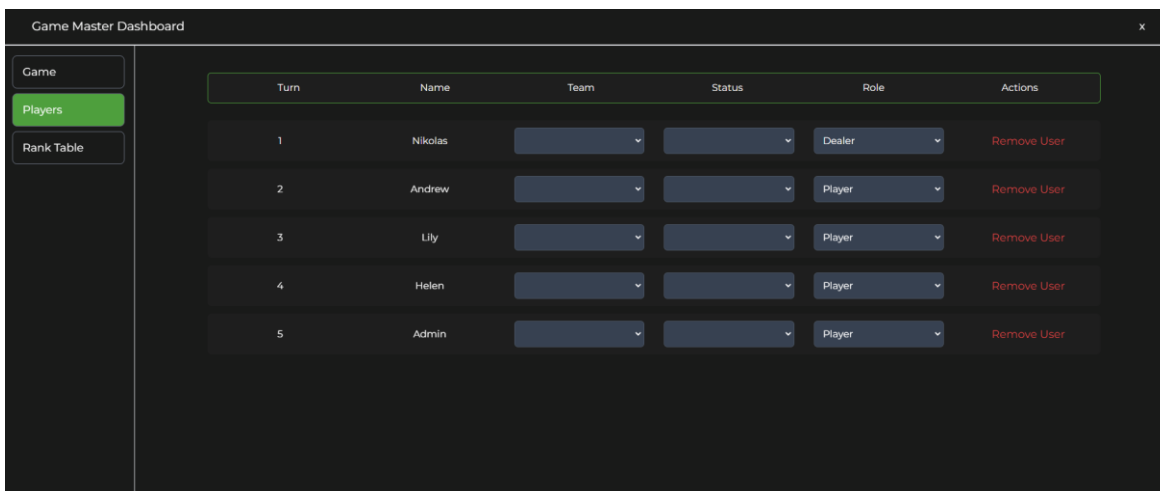
⏸ Αυτό το εργαλείο θέτει το HT σε κατάσταση παύσης (“pause”) και οι παίκτες δεν μπορούν να κάνουν κάποια κίνηση μέχρι το HT να επιστρέψει σε κατάσταση παίζεται (“playing”). Εφόσον το HT είναι σε κατάσταση παύσης τότε το εργαλείο αλλάζει σε ▶ ώστε να είναι κατανοητό στον χρήστη ότι με την επιλογή αυτού του εργαλείου θα πραγματοποιηθεί η επανέναρξη του παιχνιδιού.

✎ Το εργαλείο συμβολίζει την επεξεργασία του τραπέζιού. Όταν ο GM το επιλέξει, τότε το HT μεταβαίνει σε κατάσταση επεξεργασίας (“game\_master\_edit”) και το τραπέζι εμφανίζεται μόνο στον GM για να το επεξεργαστεί χωρίς να βλέπουν οι υπόλοιποι παίκτες τις ενέργειες που κάνει ο GM σ’ αυτό.

⌚ Το εργαλείο συμβολίζει την προσμονή για νέους παίκτες στο τραπέζι. Όταν ο GM το επιλέξει, τότε το HT μεταβαίνει σε κατάσταση αναμονής για νέους παίκτες στο τραπέζι (“waiting”) και ενημερώνει αυτόματα τους χρήστες στο προθάλαμο ότι το τραπέζι περιμένει νέους παίκτες.

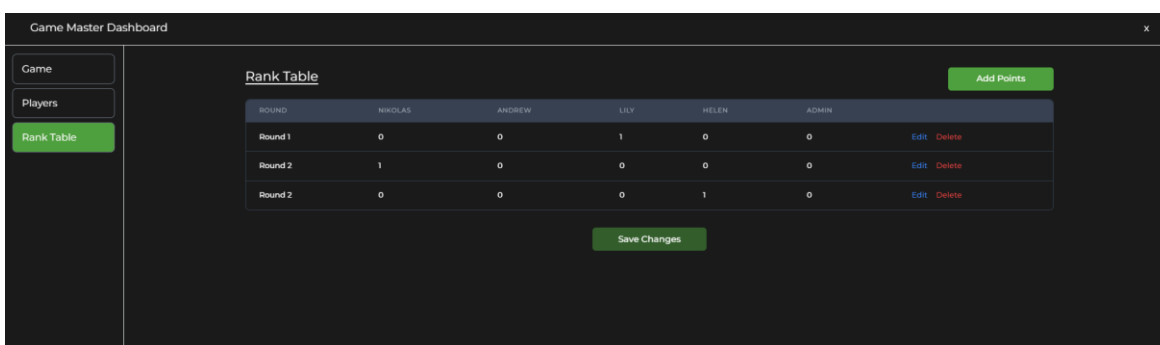
-  Το εργαλείο συμβολίζει τον τερματισμό του παιχνιδιού. Με την ενέργεια αυτή μαζεύονται όλες οι κάρτες από το HT και το τραπέζι αλλάζει σε κατάσταση τερματισμού (“finish”).
-  Με την επιλογή αυτού του εργαλείου όλοι οι χρήστες αποχωρούν αυτόματα από το HT και επιστρέφουν στον προθάλαμο χρήστη.
-  Οι παίχτες στο παιχνίδι κατατάσσονται σε μία σειρά με βάση την σειρά που μπήκαν στο HT ή από την νέα σειρά που τους έθεσε ο GM. Ο GM μπορεί να επιλέξει το προηγούμενο ή το επόμενο παίχτη που παίζει ανάλογα με την σειρά που έχουν. Αυτό το εργαλείο προσδιορίζει την επιλογή του προηγούμενου παίχτη.
-  Αυτό το εργαλείο είναι για την επιλογή του επόμενου παίχτη.
-  Αυτό το εργαλείο εμφανίζει ένα αναδυόμενο παράθυρο σε όλους του χρήστες στο οποίο προβάλλονται όλες οι κάρτες που έχει ο κάθε παίχτης και η κάθε τράπουλα. Αυτή η ενέργεια είναι χρήσιμη στο τέλος του παιχνιδιού όταν θα πρέπει να καταγραφούν οι πόντοι.
-  Με την επιλογή κάποιας τράπουλας αυτό το εργαλείο ενεργοποιείται και δίνει την δυνατότητα στον GM να ανακατέψει τις κάρτες της τράπουλας. Οι χρήστες λαμβάνουν μήνυμα ενημέρωσης για την τράπουλα που έχει ανακατευτεί.
-  Αυτό εργαλείο συμβολίζει την ενέργεια της αναίρεσης (“undo”). Ο GM έχει την δυνατότητα να αναιρέσει οποιαδήποτε κίνηση κάρτας οπουδήποτε παίχτη με όριο μέχρι 10 κινήσεις. Αυτό είναι ένα σημαντικό εργαλείο για τον GM γιατί μπορεί να διορθώσει λάθος κινήσεις των παιχτών.
-  Αυτό το εργαλείο συμβολίζει την αναίρεση της αναίρεσης (“redo”). Ο GM έχει την δυνατότητα να αναιρέσει κάποια λάθος αναίρεση που έχει κάνει. Και αυτή η ενέργεια έχει όριο μέχρι 10 κινήσεις.
-  Με την επιλογή αυτού του εργαλείου, ανοίγει ένα αναδυόμενο παράθυρο στον GM που περιέχει την λίστα με τους παίχτες που υπάρχουν στο τραπέζι , την οποία μπορεί να επεξεργαστεί. Οι δυνατότητες που του προσφέρονται είναι να μπορεί να αλλάξει την σειρά των παιχτών και να θέσει είτε ρόλο, είτε ομάδα, είτε κατάσταση σε κάποιο παίκτη. Επίσης, μπορεί να αφαιρέσει οποιοδήποτε παίχτη από το HT.





Εικόνα 48 - Ρυθμίσεις χρηστών του ηλεκτρονικού τραπεζιού

Με την επιλογή αυτού του εργαλείου, ανοίγει ένα αναδυόμενο παράθυρο στον GM που περιέχει το πίνακα βαθμολογίας του τραπεζιού. Ο GM μπορεί να προσθέσει, να επεξεργαστεί και να διαγράψει οποιαδήποτε βαθμολογία. Μπορεί να δώσει και όνομα για την ονομασία του κάθε γύρου. Όταν ο GM ενημερώνει το πίνακα, τότε οι υπόλοιποι χρήστες λαμβάνουν μήνυμα ότι ο πίνακας βαθμολογίας έχει υποστεί επεξεργασία.

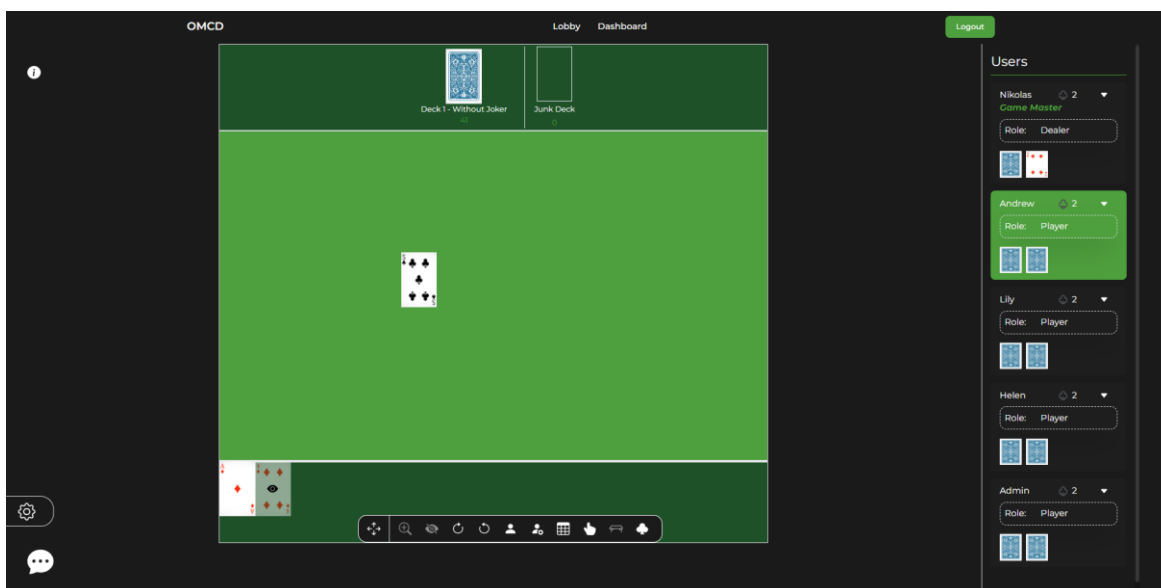


Εικόνα 49 - Ρυθμίσεις χρηστών του πίνακα βαθμολογίας του ηλεκτρονικού τραπεζιού

Με την επιλογή του εργαλείου των ρυθμίσεων παραμένει το μενού του GM ανοιχτό μέχρι να το επιλέξει πάλι και να κλείσει.

#### 4.10.2 Ηλεκτρονικό τραπέζι σε κατάσταση εξέλιξης

Με την έναρξη του παιχνιδιού μοιράζονται οι κάρτες στους παίκτες και στις άδειες τράπουλες όπως φαίνονται στην επόμενη εικόνα.



Εικόνα 50 - Blackjack παιχνίδι σε ηλεκτρονικό τραπέζι

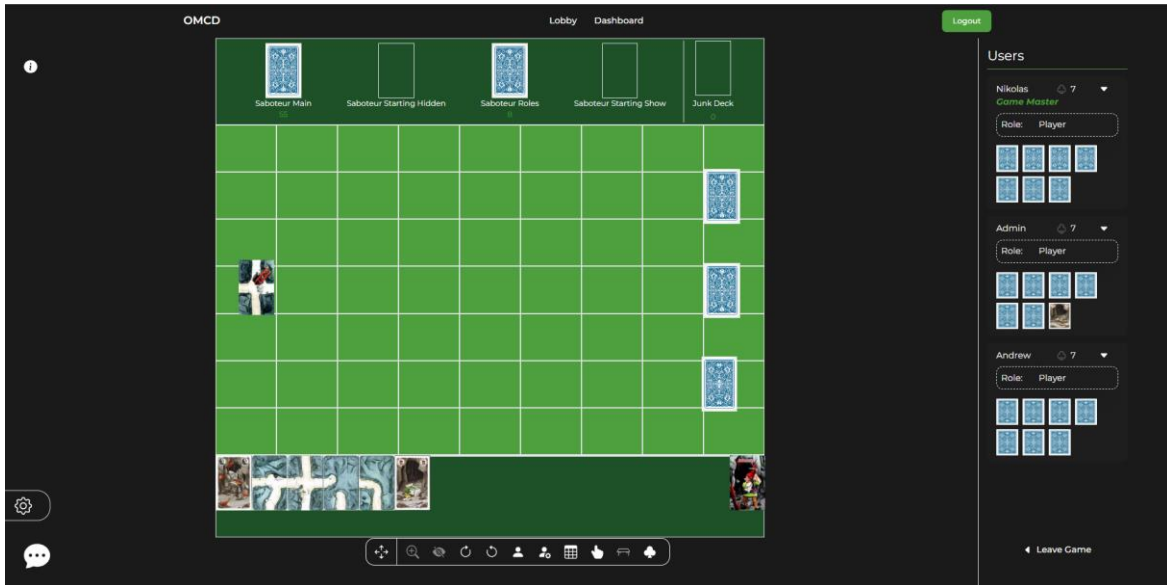
Το τραπέζι διαχωρίζεται σε 3 μέρη, το πρώτο μέρος αφορά όλες τις τράπουλες που περιέχουν ή δεν περιέχουν κάρτες. Οι τράπουλες που περιέχουν κάρτες είναι πάντα κρυφές από τους παίκτες, ενώ οι άδειες τράπουλες είναι στην επιλογή του GD αν θα είναι κρυφές ή φανερές. Επίσης στην στοιβία με τα σκάρτα οι κάρτες είναι πάντα φανερές.

Το δεύτερο μέρος είναι το τραπέζι που προσφέρεται στους παίκτες για να ρίχνουν τις κάρτες τους και να τις βλέπουν όλοι οι παίκτες. Το τραπέζι μπορεί να έχει και διαφορετικό πλέγμα όπως στην παρακάτω εικόνα [Εικόνα 51] με βάση τις ρυθμίσεις του GD. Στο τραπέζι ο παίκτης μπορεί να θέσει την κάρτα είτε φανερά, είτε κρυφά, είτε να γυρίσει την κάρτα σε οριζόντια θέση. Αυτές τις ενέργειες μπορεί να τις κάνει μέσω του μενού του χρήστη [Εικόνα 52] που θα αναλυθεί στην συνέχεια. Όταν μία κάρτα τραβηχτεί από την τράπουλα και πέσει στο τραπέζι, τότε η κάρτα αυτή είναι φανερή.

Στο τρίτο και τελευταίο μέρος είναι ο χώρος που έχει τις κάρτες του ο υφιστάμενος χρήστης. Οι κάρτες του χρήστη μπορεί να είναι φανερές ή κρυφές. Αν είναι φανερές τότε έχουν ένα θαμπό χρώμα όπως είναι ο αριθμός 4 στην από πάνω εικόνα [Εικόνα 50] και το εικονίδιο με το μάτι στην μέση. Αν οι κάρτες είναι κρυφές τότε η κάρτα εμφανίζεται κανονικά όπως είναι το 1 στην από πάνω εικόνα. Όταν ένας χρήστης τραβάει κάρτα από τράπουλα και την βάζει στο χέρι τότε η κάρτα είναι κρυφή ώστε να διασφαλιστεί ότι οι άλλοι παίκτες δεν θα την δούνε.

Στα δεξιά μέρος της σελίδας, το sidebar του τραπέζιου αποκτά νέες πληροφορίες σχετικά με τον ρόλο του, την κατάσταση και την ομάδα που έχει ο παίκτης. Αν ο παίκτης δεν έχει κάποια πληροφορία σε κάποια από αυτές τις τρεις ιδιότητες, τότε δεν εμφανίζεται. Επιπρόσθετα, σε κάθε παίκτη εμφανίζονται οι κάρτες που έχει ανάλογα στην κατάσταση που βρίσκονται (φανερές ή κρυφές).

Η εικόνα [Εικόνα 50] είναι ένα στιγμιότυπο από ένα παιχνίδι Blackjack όπου όλοι οι παίκτες έχουν από δύο κρυφά χαρτιά και ο ντίλερ έχει ένα κρυφό και ένα φανερό χαρτί. Ο παίκτης θα ζητάει κάρτα είτε μέσω της κατάστασης του είτε μέσω της γραπτής συνομιλίας. Ο ντίλερ θα τραβάει κάρτες από την τράπουλα και θα τις ρίχνει στο τραπέζι μέχρι ο παίκτης να πάει πάσο.



Εικόνα 51 - Saboteur παιχνίδι σε ηλεκτρονικό τραπέζι





Οι χρήστες μπορούν να επιλέξουν μία κάρτα, και αφήνοντας πεισμένο το αριστερό κουμπί του ποντικιού μπορούν να την μετακινήσουν και να την αφήσουν σε οποιοδήποτε επιτρεπόμενο μέρος του τραπεζιού. Αν την αφήσουν σε σημείο εκτός του τραπεζιού, τότε η κάρτα θα επιστρέψει στην προηγούμενη της θέση. Αυτή η υλοποίηση έγινε με την χρήση του HTML Drag and Drop API [30].


#### 4.10.3 Μενού παίχτη






Εικόνα 52 - Μενού εργαλείων παίχτη


Το μενού του παίχτη δημιουργήθηκε ώστε να προσφέρει κάποιες συγκεκριμένες ενέργειες σε οποιοδήποτε παίχτη (περιλαμβανόμενου και του GM) που αφορούν τις κάρτες του παιχνιδιού και τον ίδιο τον παίχτη. Οι ενέργειες που προσφέρονται είναι οι εξής:


-  Πιέζοντας αυτό το εργαλείο, ο χρήστης μπορεί να μετακινήσει το μενού σε οποιοδήποτε μέρος της σελίδας τον βολεύει.
-  Όταν ο χρήστης έχει επιλεγμένη μία κάρτα, επιλέγοντας αυτό το εργαλείο μπορεί να δει την κάρτα σε μεγαλύτερο μέγεθος σε ένα νέο αναδυόμενο παράθυρο.
-  Αυτό το εργαλείο ενεργοποιείται αφού ο χρήστης επιλέξει ή μετακινήσει μία κάρτα που είναι κρυφή για να την κάνει φανερή. Αν η κάρτα είναι φανερή τότε εμφανίζεται αυτό το εργαλείο  ώστε να μπορεί να την κάνει κρυφή.


 Αυτό το εργαλείο ενεργοποιείται με την επιλογή ή μετακίνηση μίας κάρτας από τον χρήστη, ώστε να μπορεί να την περιστρέψει προς τα δεξιά. Σε κάθε κλικ του χρήστη προστίθενται στην κάρτα 90 μοίρες.


 Αυτό το εργαλείο είναι το ίδιο με το προηγούμενο με την μονή διαφορά η κάρτα περιστρέφεται σε κάθε κλικ κατά 90 μοίρες προς τα αριστερά.


 Αυτό το εργαλείο προσδιορίζει τον ίδιο τον χρήστη. Αν είναι η σειρά του να παίξει τότε επιλέγει αυτό το εργαλείο ώστε να ενημερώσει τους άλλους παίκτες ότι ήρθε η σειρά του. Όταν είναι επιλεγμένο το εργαλείο αλλάζει το εικονίδιο στο εργαλείο  και με την επιλογή αυτού μπορεί να θέσει ότι έχει τερματίσει τις κινήσεις του.

 Αυτό το εργαλείο αφορά τις καταστάσεις του παίχτη. Επιλέγοντας αυτό το εργαλείο, τότε εμφανίζεται ένα μενού από τις διαθέσιμες επιλογές των καταστάσεων που ορίστηκαν στο παιχνίδι ώστε να δηλώσει στους υπόλοιπους παίκτες την κατάσταση που βρίσκεται.

 Επιλέγοντας αυτό το εργαλείο ο παίχτης μπορεί να δει τον πίνακα βαθμολογίας του τραπέζιού.

 Αυτό το εργαλείο εμφανίζει όλες τις διαθέσιμες τράπουλες του παιχνιδιού. Ο παίχτης μπορεί να μαζέψει όλες τις κάρτες από μία τράπουλα με μία κίνηση.

 Αυτό το εργαλείο εμφανίζει όλες τις διαθέσιμες τράπουλες του παιχνιδιού. Ο παίχτης μπορεί να επιλέξει να μετακινήσει όλες τις κάρτες του τραπέζιού σε οποιαδήποτε τράπουλα με μία κίνηση.

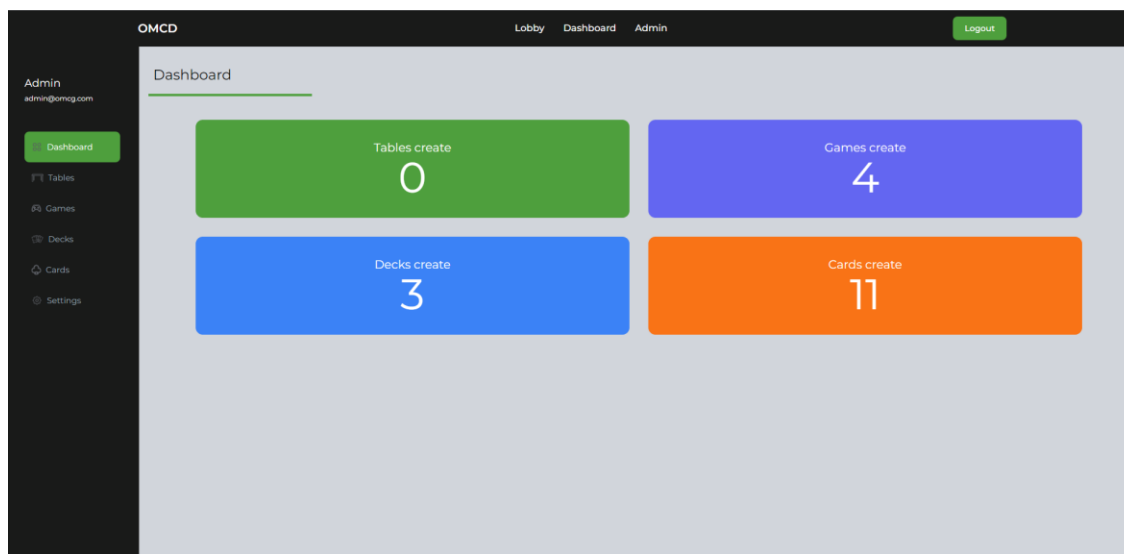
 Αυτό το εργαλείο δίνει την δυνατότητα στο χρήστη να δώσει μία κάρτα του σε κάποιο άλλο παίχτη. Πρώτα χρειάζεται να επιλέξει μία κάρτα για να ενεργοποιηθεί το εργαλείο, και στην συνέχεια να επιλέξει ένα παίχτη από το μενού για να του δώσει την κάρτα.



Εικόνα 53 - Μενού εργαλείων παίχτη όταν παίζει

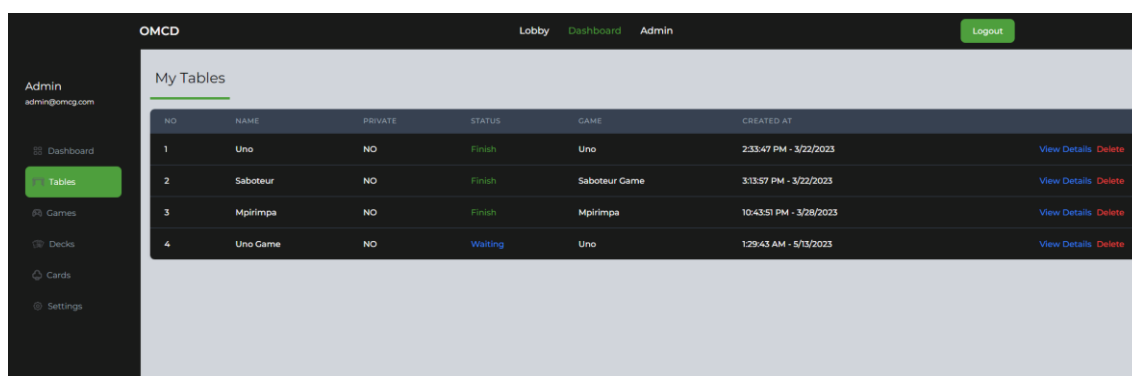
## 4.11 Ταμπλό χρήστη

Το ταμπλό του χρήστη (Dashboard) είναι μία ενότητα που αποτελείται από αρκετές σελίδες και είναι ορατή μόνο από εγγεγραμμένους χρήστες. Αυτή η σελίδα δίνει την δυνατότητα στον χρήστη να επεξεργαστεί τα προσωπικά δεδομένα του, τις κάρτες, τις τράπουλες, τα παιχνίδια και τα ΗΤ. Με την επιλογή την σελίδας Dashboard ο χρήστης κατευθύνεται σε αυτή την ενότητα και του παρουσιάζονται απλές ρυθμίσεις σχετικά με αυτόν και την πλατφόρμα όπως φαίνεται στην επόμενη εικόνα.



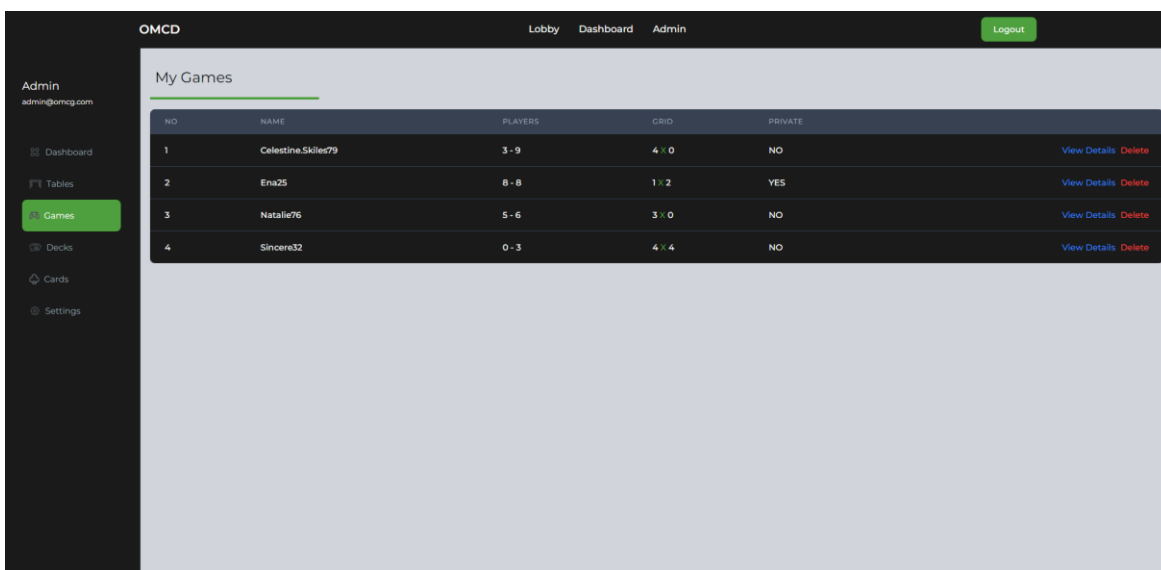
Εικόνα 54 - Στιγμιότυπο από την σελίδα Dashboard ενός χρήστη

Στο αριστερό μέρος της σελίδας υπάρχει μενού πλαϊνής γραμμής που περιέχει τις σελίδες της ενότητας που μπορεί να μεταβεί ο χρήστης. Στην σελίδα “Tables” ο χρήστης μπορεί να δει το ιστορικό του με όλα τα τραπέζια που δημιούργησε.

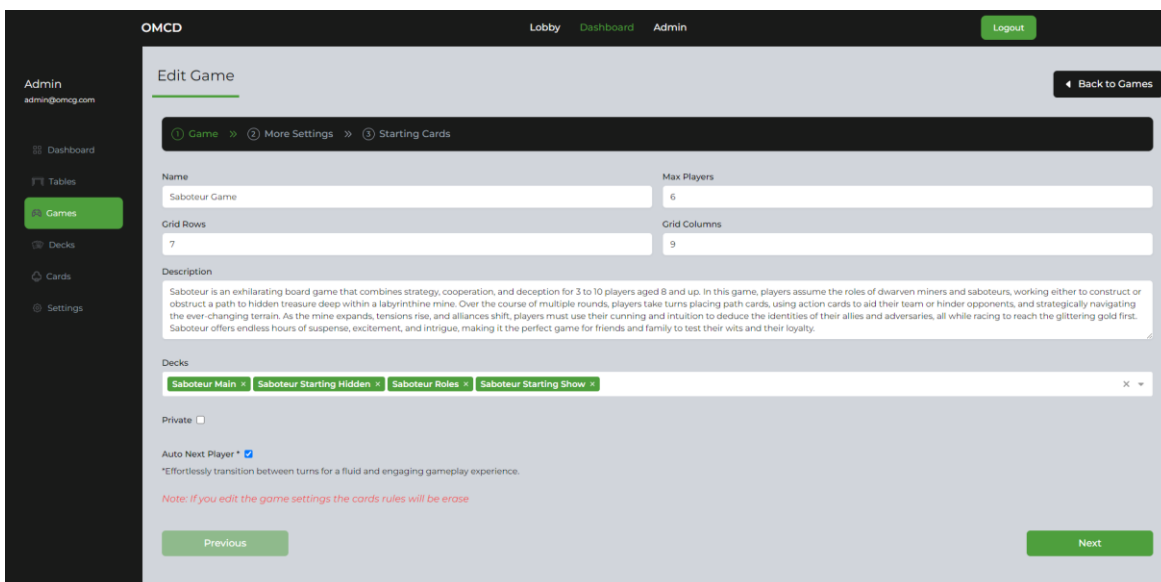


Εικόνα 55 - Στιγμιότυπο από την σελίδα Tables ενός χρήστη

Η επόμενη σελίδα “Games” περιέχει την λίστα με τα παιχνίδια που δημιούργησε ο υφιστάμενος χρήστης. Μπορεί να επεξεργαστεί τις ρυθμίσεις των παιχνιδιών του ή να διαγράψει τα παιχνίδια.

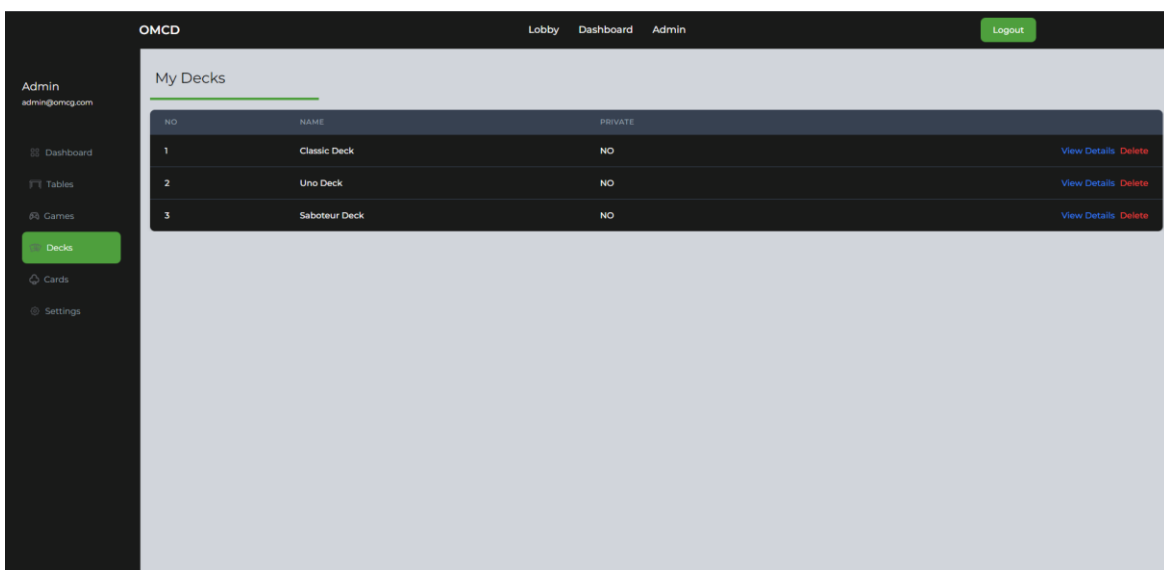


Εικόνα 56 - Στιγμιότυπο από την σελίδα προβολής παιχνιδιών του χρήστη



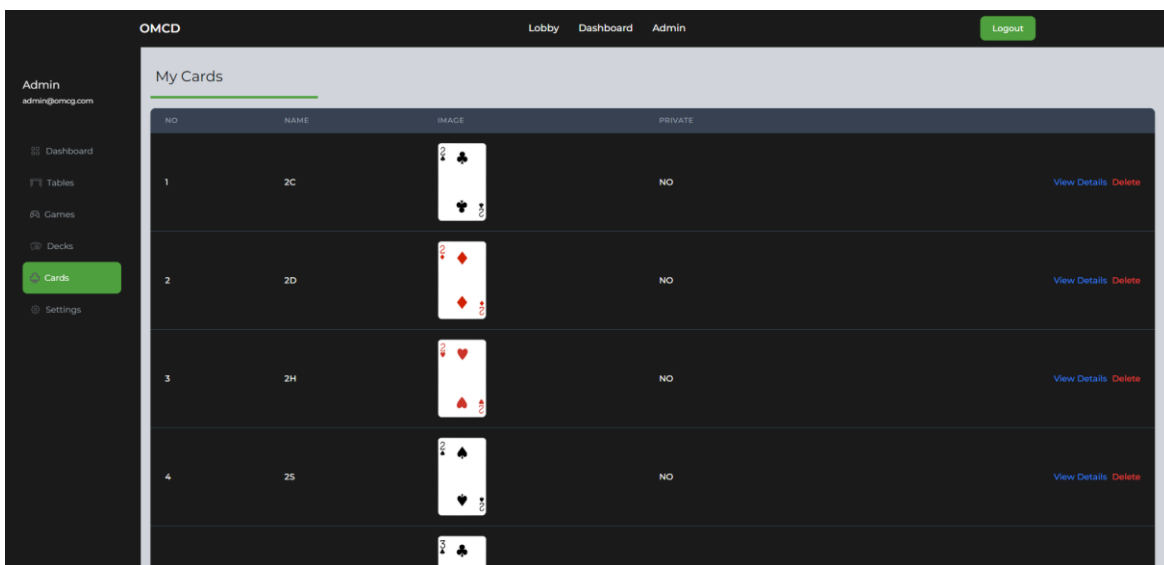
Εικόνα 57 - Στιγμιότυπο από την επεξεργασία παιχνιδιού από το ταμπλό του χρήστη

Η σελίδα “Decks” περιέχει την λίστα με τις τράπουλες που δημιουργήθηκαν από τον χρήστη. Στο πίνακα φαίνονται οι βασικές πληροφορίες της κάθε τράπουλας. Με την επιλογή του κουμπιού “View Details”, εμφανίζεται στην σελίδα η συγκεκριμένη τράπουλα μαζί με όλες τις πληροφορίες και με τις κάρτες που περιέχει. Σε αυτό το σημείο ο χρήστης μπορεί να αλλάξει τις πληροφορίες της τράπουλας ή να προσθέσει καθώς και να αφαιρέσει κάρτες από την τράπουλά. Η φόρμα που χρησιμοποιήθηκε στην επεξεργασία της τράπουλας είναι η ίδια με αυτή της δημιουργίας της, καθώς αυτή η φόρμα δημιουργήθηκε ως component στην Vue. Επιπρόσθετα ο χρήστης μπορεί να διαγράψει όποια τράπουλα επιθυμεί.



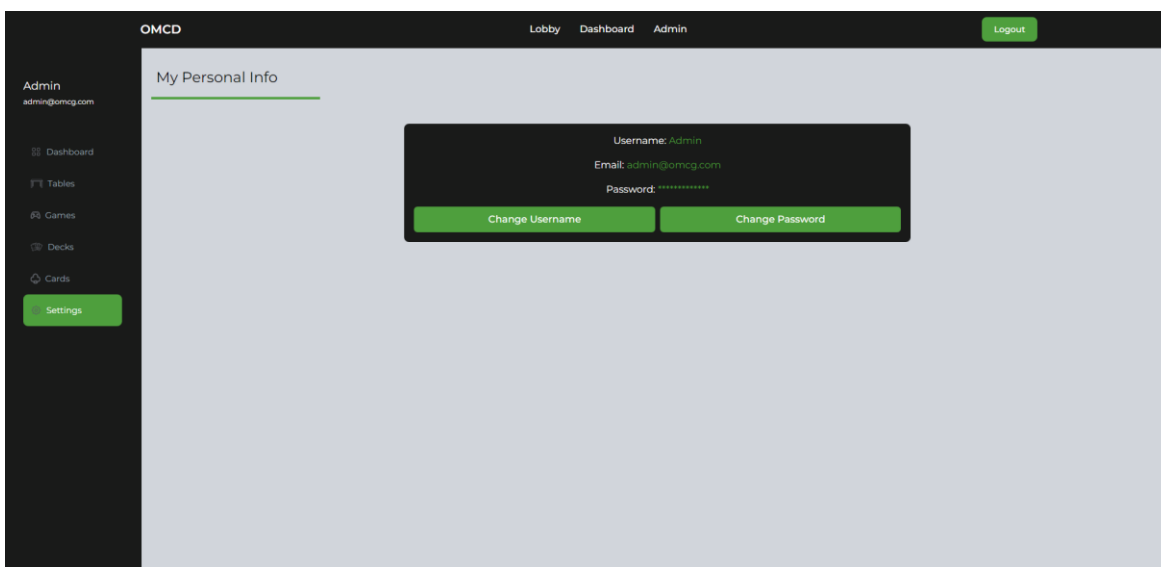
Εικόνα 58 - Στιγμιότυπο από την σελίδα προβολής τραπουλών του χρήστη

Η σελίδα "Cards" περιέχει τις λίστες με τις κάρτες του χρήστη. Με την ίδια μεθοδολογία από την πιο πάνω σελίδα έτσι και σε αυτή γίνεται η προβολή των καρτών. Οι κάρτες μπορούν να διαγραφούν ή να επεξεργαστούν μέσω της ίδιας φόρμας της σελίδας "Create card".

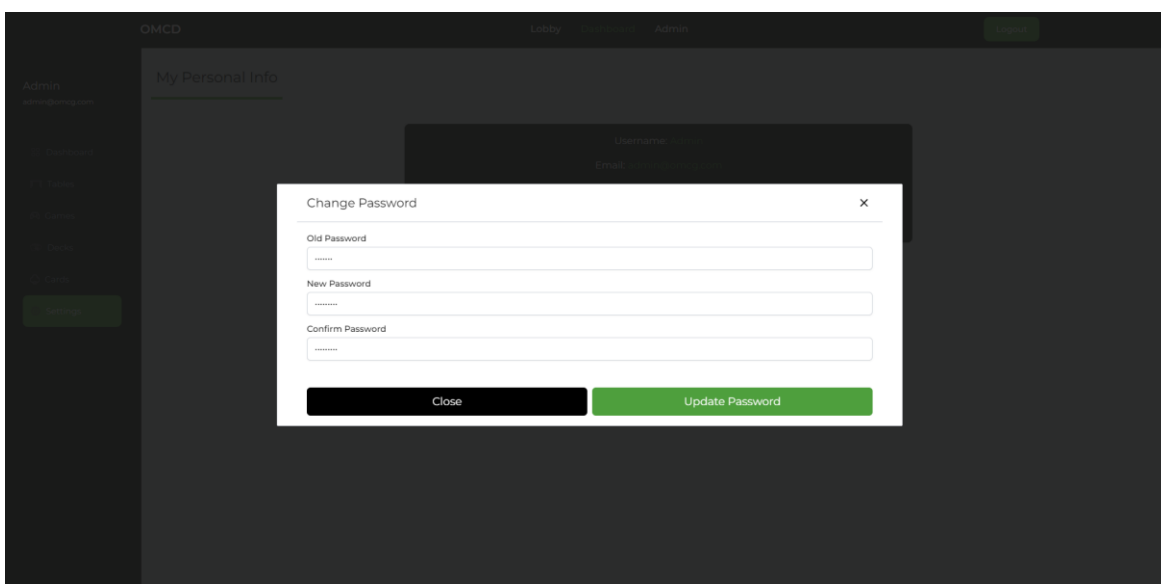


Εικόνα 59 - Στιγμιότυπο από την σελίδα προβολής καρτών του χρήστη

Η τελευταία σελίδα της ενότητας παρέχει στο χρήστη την δυνατότητα να επεξεργαστεί τα προσωπικά δεδομένα του. Μπορεί να αλλάξει το όνομα του username του ή τον κωδικό πρόσβασης του.



Εικόνα 60 - Στιγμιότυπο από την σελίδα Settings



Εικόνα 61 - Στιγμιότυπο από την αλλαγή του κωδικού πρόσβασης του χρήστη

## 4.12 Ταμπλό διαχειριστή

Στο βασικό μενού της ιστοσελίδας είναι ορατή η επιλογή “Admin” η οποία είναι ορατή μόνο από τους διαχειριστές της πλατφόρμας που περιέχει το ταμπλό του διαχειριστή. Όπως και το ταμπλό του χρήστη, έτσι και το ταμπλό του διαχειριστή είναι μία ενότητα που αποτελείται από αρκετές σελίδες. Οι σελίδες που το αποτελούν είναι “Dashboard”, “Users”, “Tables”, “Games”, “Decks” και “Cards”.

Οι σελίδες “Dashboard”, “Tables”, “Games”, “Decks” και “Cards” έχουν ακριβώς την ίδια λογική μαζί με αυτή του χρήστη με την μόνη διαφορά ότι ο διαχειριστής βλέπει όλες τις πληροφορίες, από όλους τους χρήστες της πλατφόρμας. Η σελίδα “Settings” δεν



υπάρχει στην ενότητα του διαχειριστή καθώς τα προσωπικά του δεδομένα μπορεί να τα τροποποιήσει από το ταμπλό του χρήστη.

Η μοναδική επιπλέον σελίδα που έχει είναι αυτή των χρηστών (“Users”), στην οποία είναι διαθέσιμη και η λίστα με όλους του χρήστες της πλατφόρμας. Ο διαχειριστής έχει την δυνατότητα να επεξεργαστεί τα δεδομένα των χρηστών αλλά και να τους διαγράψει από την βάση δεδομένων.

ID	USERNAME	EMAIL	ROLE	EMAIL CONFIRM	CREATED AT	UPDATED AT	
1	Admin	admin@omcg.com	admin	YES	4:07:14 PM - 1/15/2023	5:40:23 PM - 1/15/2023	<a href="#">View Details</a> <a href="#">Delete</a>
2	Agnes.Kulas	Guy53@yahoo.com	user	NO	4:07:15 PM - 1/15/2023	4:07:15 PM - 1/15/2023	<a href="#">View Details</a> <a href="#">Delete</a>
3	Bette_Armstrong15	Ciselle.Beer@hotmail.com	user	NO	4:07:15 PM - 1/15/2023	4:07:15 PM - 1/15/2023	<a href="#">View Details</a> <a href="#">Delete</a>
4	Kade_Leuschke66	Madisen.Cole37@yahoo.com	user	NO	4:07:15 PM - 1/15/2023	4:07:15 PM - 1/15/2023	<a href="#">View Details</a> <a href="#">Delete</a>
5	Hubert.Hammes	Ocie.Prica24@yahoo.com	user	YES	4:07:15 PM - 1/15/2023	4:07:15 PM - 1/15/2023	<a href="#">View Details</a> <a href="#">Delete</a>
6	Katherine.Ripplin	Mabelle63@hotmail.com	user	NO	4:07:16 PM - 1/15/2023	4:07:16 PM - 1/15/2023	<a href="#">View Details</a> <a href="#">Delete</a>
7	Aubrey.Stehr	Coit_Hills@gmail.com	user	NO	4:07:16 PM - 1/15/2023	4:07:16 PM - 1/15/2023	<a href="#">View Details</a> <a href="#">Delete</a>
8	Mozell10	Elbert45@gmail.com	user	NO	4:07:16 PM - 1/15/2023	4:07:16 PM - 1/15/2023	<a href="#">View Details</a> <a href="#">Delete</a>
9	Tanya.Moen67	Josue_Rolfson99@gmail.com	user	YES	4:07:16 PM - 1/15/2023	4:07:16 PM - 1/15/2023	<a href="#">View Details</a> <a href="#">Delete</a>
10	Boris.Rolfson	Sylvester.Erdman@gmail.com	user	YES	4:07:16 PM - 1/15/2023	4:07:16 PM - 1/15/2023	<a href="#">View Details</a> <a href="#">Delete</a>
11	Kristian69	Aaliyah60@hotmail.com	user	NO	4:07:17 PM - 1/15/2023	4:07:17 PM - 1/15/2023	<a href="#">View Details</a> <a href="#">Delete</a>
12	Curt69	Ubaldo65@hotmail.com	user	NO	4:07:17 PM - 1/15/2023	4:07:17 PM - 1/15/2023	<a href="#">View Details</a> <a href="#">Delete</a>
13	Destin80	Jacquelyn.Bosco@yahoo.com	user	NO	4:07:17 PM - 1/15/2023	4:07:17 PM - 1/15/2023	<a href="#">View Details</a> <a href="#">Delete</a>
14	Rex57	Evan_Hartmann@hotmail.com	user	NO	4:07:17 PM - 1/15/2023	4:07:17 PM - 1/15/2023	<a href="#">View Details</a> <a href="#">Delete</a>

Εικόνα 62 - Στιγμιότυπο από την σελίδα προβολή χρηστών του διαχειριστή

X

Username  
Admin

Email  
admin@omcg.com

Role  
Admin

Email Confirmed

Save Changes
Close

Εικόνα 63 - Στιγμιότυπο από την επεξεργασία ενός χρήστη από την σελίδα του διαχειριστή

## 4.13 Web Sockets Events

Για την πραγματοποίηση της σύνδεσης των χρηστών σε πραγματικό χρόνο για τα HT χρησιμοποιήθηκε σύνδεση με Web Sockets. Η διαφορά του με το HTTP πρωτόκολλο στην υλοποίηση είναι ότι για το HTTP, λαμβάνονται ή αποστέλλονται δεδομένα με την χρήση διευθύνσεων (APIs endpoints) και αιτείται πάντα ο χρήστης τα δεδομένα.

Για να επιτευχθεί μία σύνδεση με Web Socket, δημιουργούνται συμβάντα (events), που όλα τα sockets που είναι συνδεδεμένα ακούνε σε αυτά σε πραγματικό χρόνο. Για παράδειγμα, ο χρήστης ενημερώνει τον διακομιστή ότι έχει εισέλθει ένα νέος χρήστης, και αυτός με την σειρά του, στέλνει ένα συμβάν ότι ένα νέος χρήστης έχει

συνδεθεί στην πλατφόρμα και όλα τα socket λαμβάνουν αυτή την πληροφορία. Ο διακομιστής μπορεί να απαντήσει στο αίτημα το χρήστη, αλλά μπορεί να στείλει και διαφορετικό συμβάν για να λάβουν την ενημέρωση οι χρήστες, καθώς αν απαντήσει στο αίτημα του χρήστη θα το λάβει μόνο αυτός. Το front-end πρόγραμμα αναλαμβάνει πως θα διαχειριστεί αυτή την πληροφορία και πως θα την παρουσιάσει στον χρήστη. Επιπρόσθετα, ο διακομιστής έχει την επιλογή να ομαδοποιήσει τους χρήστες και να στέλνει πληροφορίες σε συγκεκριμένες ομάδες ή χρήστες.

Για τις ανάγκες της πλατφόρμας δημιουργήθηκαν αρκετά συμβάντα (events), κάποια από αυτά αφορούν το προθάλαμο του χρήστη και λαμβάνουν ενημέρωση όλοι οι συνδεδεμένοι χρήστες που είναι στην πλατφόρμα ενώ άλλα αφορούν αποκλειστικά το HT και λαμβάνουν ενημέρωση μόνο οι χρήστες που ανήκουν σ' αυτό. Στον παρακάτω πίνακα παρουσιάζονται όλα τα συμβάντα της πλατφόρμας. Στην πρώτη στήλη είναι αυτά που αιτείται ο χρήστης, και στην δεύτερη στήλη είναι τα συμβατά που στέλνει ο διακομιστής ώστε να ενημερωθεί κατάλληλα η πλατφόρμα.

Client	Server
createOnlineTable	getOnlineTable
removeTable	getRemoveTable
joinTable	getTableUsers, getUpdateTable
leaveTable	getTableUsers
removePlayer	getTableUsers
getAllOnlineTable	
setTurnTableUsers	getTurnTableUsers
setPlayerPlaying	getTurnTableUsers
setNextPlayer	getTurnTableUsers
setTeamTableUser	getUpdateTableUser
setRoleTableUser	getUpdateTableUser
setStatusTableUser	getUpdateTableUser
stopGame	getStartGameDetails
newGame	getStartGameDetails, getUpdateTable
exitTable	exitUsersFromTable, getUpdateTable
updateCard	getUpdateCard
updateTableGameStatus	getTableGameStatus, getUpdateTable
showAllCards	getShowAllCards
validateTablePassword	
shuffleDeck	getShuffleDeck
sendMessage	getSendMessage
getRankTable	
storeRankRow	getUpdateRankRow
updateRankRow	getUpdateRankRow

deleteRankRow	getUpdateRankRow
setOnlineSocketUser	getLastGame, getUsersOnline
setOfflineSocketUser	getUsersOnline

Πίνακας 1 - Web Socket Events

Σημείωση: Κάποια συμβατά στην δεύτερη στήλη είναι κενά καθώς λαμβάνει απάντηση στο συμβάν μόνο ο χρήστης που το κάλεσε.

Με την είσοδο του χρήστη στην πλατφόρμα, είτε ως εγγεγραμμένος είτε ως ανώνυμος, γίνεται καταγραφή της σύνδεσης μέσω του συμβάντος **setOnlineSocketUser** και ο διακομιστής ενημερώνει τους υπόλοιπους χρήστες μέσω του συμβάντος **getUsersOnline** για τον νέο αριθμό των συνδεδεμένων χρηστών. Με παρόμοιο τρόπο γίνεται και η ενημέρωση του χρήστη ως μη συνδεδεμένος με την χρήση του συμβάν **setOfflineSocketUser**.

Με την είσοδο του ο χρήστης στο προθάλαμο λαμβάνει όλα τα διαθέσιμα τραπέζια που είναι διαθέσιμα για παιχνίδι μέσω του συμβάν **getAllOnlineTable**. Ένας εγγεγραμμένος χρήστης μπορεί να δημιουργήσει ένα HT ή να το διαγράψει αν είναι ο δημιουργός του με την χρήση των **createOnlineTable** και **removeTable**. Το front-end λαμβάνει με συμβάν από το διακομιστή το τραπέζι που πρέπει με προσθέσει ή να αφαιρέσει μέσω των **getOnlineTable** και **getRemoveTable**. Ο προθάλαμος λαμβάνει και το συμβάν **getTableUsers**, που αφορά στοιχεία για τους χρήστες που έχουν συνδεθεί ή αποσυνδεθεί από ένα HT μέσω των **joinTable**, **leaveTable**, **removePlayer**. Επίσης λαμβάνει και το συμβάν **getTableUpdate** που ενημερώνει τις καταστάσεις των HT ώστε να αφαιρείται από την λίστα των HT όταν δεν είναι διαθέσιμο και να προστίθεται αν είναι πάλι διαθέσιμο για να δεχθεί νέους παίκτες.

Το συμβάν **getTableUsers** λαμβάνεται και στο HT ώστε να ενημερώνεται η λίστα με τους συνδεδεμένους παίκτες του. Οι παίκτες στο HT χρησιμοποιούν μόνο τα συμβάντα **setPlayerPlaying**, **updateCard** και **sendMessage**. Το πρώτο αφορά την ενημέρωση του παίκτη προς τους άλλους παίκτες ότι είναι η σειρά του για να παίξει, το δεύτερο σχετίζεται με οποιαδήποτε ενέργεια έχει ο παίκτης με μία κάρτα και το τελευταίο αφορά τα μηνύματα που στέλνει στην ομαδική συνομιλία του HT.

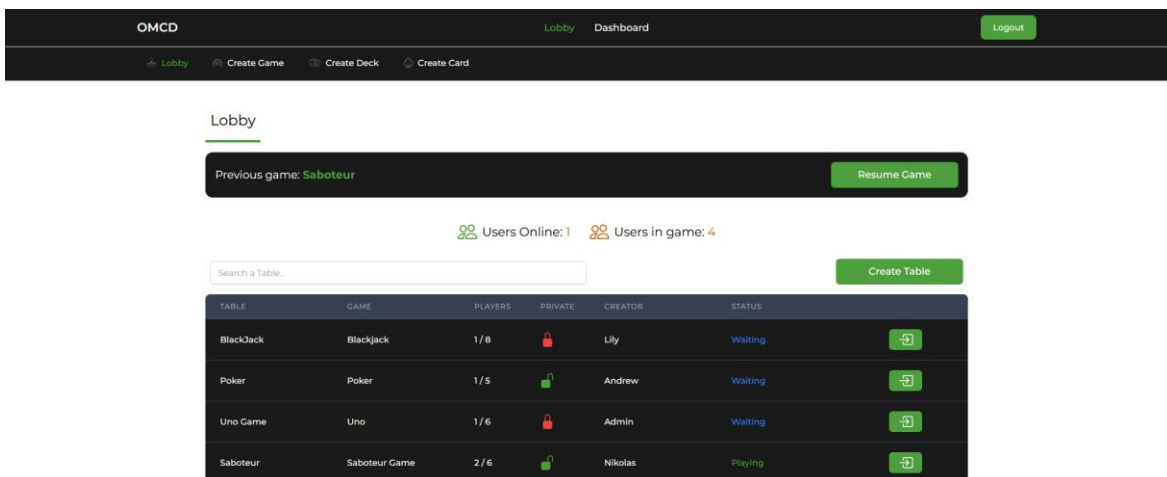
Ο GM, λόγω και του καθοριστικού ρόλου που έχει στο HT, χρησιμοποιεί επιπλέον συμβάντα που είναι τα εξής:

- **setTurnTableUsers:** Καθορίζει την σειρά που κάθονται και θα παίζουν οι παίκτες στο HT
- **setNextPlayer:** Ενημερώνει τον αμέσως επόμενο ή προηγούμενο παίκτη ότι παίζει
- **setTeamTableUser:** Θέτει την ομάδα που θα έχει ο παίκτης
- **setRoleTableUser:** Θέτει τον ρόλο που θα έχει ο παίκτης
- **setStatusTableUser:** Θέτει την κατάσταση που βρίσκεται ένας παίκτης
- **stopGame:** Σταματάει το παιχνίδι, μαζεύει τις κάρτες από τους παίκτες και οι παίκτες παραμένουν στο τραπέζι
- **newGame:** Ξεκινάει το παιχνίδι και μοιράζει τις κάρτες τους παίκτες και τις άδειες τράπουλες

- **exitTable**: Διώχνει όλους τους παίχτες από το HT και κλείνει το τραπέζι
- **updateTableGameStatus**: Αλλάζει την κατάσταση του HT είτε σε παύση, είτε σε επεξεργασία, είτε αναμονή για νέους παίχτες
- **shuffleDeck**: Ανακατεύει οποιαδήποτε τράπουλα της επιλογής του που δεν ανήκει σε κάποιο παίχτη
- **storeRankRow**, **updateRankRow**, **deleteRankRow**: Αφορούν την καταγραφή, ενημέρωση ή την διαγραφή της βαθμολογίας για τους παίχτες

Σε ένα παιχνίδι πραγματικού χρόνου μπορούν να προκύψουν διάφορα σενάρια και το παιχνίδι που είναι σε εξέλιξη να μην καταφέρει να ολοκληρωθεί επιτυχώς. Μερικές περιπτώσεις είναι όταν ο χρήστης αποχωρήσει με δική του βούληση από το παιχνίδι, είτε γιατί χάθηκε η σύνδεση είτε από απροσεξία του ίδιου. Ένα παιχνίδι με κάρτες είναι αδύνατον να συνεχίσει χωρίς την επιστροφή του παίχτη πίσω, καθώς ο παίχτης αντιπροσωπεύει κάποιες κάρτες στο παιχνίδι που θα ήταν άδικο απλά να επιστραφούν πίσω στην τράπουλα που ανήκουν.

Με την παραπάνω προσέγγιση, σε περίπτωση που ο χρήστης ανήκει σε κάποιο HT που είναι σε εξέλιξη και αποσυνδεθεί, τότε η πλατφόρμα καταγραφεί αυτό το γεγονός στην βάση δεδομένων και θέτει το παιχνίδι σε παύση και ενημερώνει τους υπόλοιπους χρήστες αν ο χρήστης έφυγε με δική του θέληση ή λόγω σύνδεσης. Με την είσοδο του αποσυνδεδεμένου χρήστη στην προθάλαμο της πλατφόρμας, του εμφανίζεται το παιχνίδι που έπαιζε ώστε να ξανασυνδεθεί και να συνεχίσει το παιχνίδι εφόσον ο GM αποφάσισε να περιμένει το χρήστη να επανέλθει πίσω.



Εικόνα 64 - Στιγμιότυπο πριν από την επανασύνδεση του παίχτη στο παιχνίδι στην σελίδα "Προθάλαμος χρήστη"

Η καταγραφή του τρόπου αποσύνδεσης επιτυγχάνεται με την χρήση της Vue.js και την Nest.js. Με την χρήση της συνάρτησης **unMounded()** που ανήκει στο οικοσύστημα της Vue, όταν καταστραφεί το component που αφορά το HT, τότε αποθηκεύεται ο χρήστης στην βάση δεδομένων ως leaver. Αν ο χρήστης κλείσει το

πρόγραμμα περιήγησης ή χαθεί η σύνδεση του τότε δεν καλείται η συνάρτηση **unMounded()** του component γιατί έχει κλείσει όλο το πρόγραμμα και όχι ένα μόνο κομμάτι του. Η Nest.js όμως, λαμβάνει ενημερώσει για κάθε χρήστη που συνδέεται ή αποσυνδέεται από την πλατφόρμα μέσω της σύνδεσης, Web Socket. Μέσω αυτής της δυνατότητας καταγράφεται ο χρήστης ότι έχει αποσυνδεθεί και όταν συνδεθεί λαμβάνει τα δεδομένα του παιχνιδιού ώστε να συνδεθεί πάλι μέσω του συμβάντος **getLastGame**, εάν εφόσον ο GM δεν έχει ξεκινήσει νέο παιχνίδι.

```
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "setOnlineSocketUser" message +52ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "setOfflineSocketUser" message +1ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "createOnlineTable" message +0ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "removeTable" message +1ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "joinTable" message +0ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "leaveTable" message +1ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "getAllOnlineTable" message +1ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "setTurnTableUsers" message +0ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "setStatusTableUser" message +1ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "setTeamTableUser" message +0ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "setRoleTableUser" message +1ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "stopGame" message +0ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "exitTable" message +1ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "newGame" message +0ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "updateCard" message +1ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "setPlayerPlaying" message +0ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "updateTableGameStatus" message +0ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "showAllCards" message +1ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "removePlayer" message +1ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "validateTablePassword" message +0ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "setNextPlayer" message +0ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "shuffleDeck" message +1ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "sendMessage" message +1ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "storeRankRow" message +0ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "getRankTable" message +0ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "updateRankRow" message +1ms
[Nest] 20632 - 05/13/2023, 9:02:14 PM LOG [WebSocketsController] OnlineTableGateway subscribed to the "deleteRankRow" message +0ms
```

Εικόνα 65 - Στιγμιότυπο από την κονσόλα του διακομιστή για τα Web Socket Events

## 5. Επίλογος

Το πέμπτο και τελευταίο κεφάλαιο ξεκινάει συνοψίζοντας την διαδικτυακή πλατφόρμα ως προς την ανάπτυξη των κύριων χαρακτηριστικών της, τις λειτουργίες και τα εργαλεία υλοποίησης της. Στο επόμενο κεφάλαιο θα παρουσιαστούν τα μετρικά του κώδικα για τις δύο εφαρμογές (frontend και backend). Στην συνέχεια θα αναλυθεί η έννοια του μοντέλου SWOT ώστε να προκύψει η ανάλυση της πλατφόρμας μέσω αυτού του μοντέλου. Στο κεφάλαιο «Πλάνο ελέγχου ορθής λειτουργίας» περιγράφονται τα βήματα που χρειάζονται να γίνουν ώστε η πλατφόρμα να φιλοξενηθεί σε ένα περιβάλλον παραγωγής και στην συνέχεια να πραγματοποιηθεί η διαδικασία δοκιμής της όπου καταγράφονται τυχόν προβλήματα ή νέες απαιτήσεις που προκύπτουν. Τέλος, γίνονται αναφορές σε μελλοντικές επεκτάσεις που θα έχει η πλατφόρμα καθώς και τα συμπεράσματα που προέκυψαν με την ολοκλήρωση αυτής της πτυχιακής εργασίας.

### 5.1 Ανακεφαλαίωση

Η παρούσα πτυχιακή εργασία αφορά την δημιουργία μίας διαδικτυακής πλατφόρμας για εξυπηρέτηση παιχνιδιών με κάρτες από απομακρυσμένους παίχτες. Κύριος σκοπός της πλατφόρμας είναι να δώσει την δυνατότητα στους χρήστες να δημιουργήσουν δικά τους ή προ υπάρχον ΠΜΚ και να τα παίξουν με άλλους χρήστες σε πραγματικό χρόνο. Ταυτόχρονα, η πλατφόρμα προσφέρει κοινωνική αλληλεπίδραση δημιουργώντας έτσι μία ζωντανή και ελκυστική κοινότητα για τους λάτρεις των ΠΜΚ.

Συνοπτικά, η πλατφόρμα υποστηρίζει τρεις (3) κατηγορίες χρηστών, των ανώνυμο, τον εγγεγραμμένο χρήστη και τον διαχειριστή. Η μοναδική δυνατότητα που έχει ο ανώνυμος χρήστης είναι να μπορεί να εισέλθει σε τραπέζια με πραγματικούς παίχτες και να παίξει ΠΜΚ. Ο εγγεγραμμένος χρήστης, μπορεί να δημιουργήσει ένα παιχνίδι με τον ρόλο του Game Designer από το μηδέν, μπορεί να φτιάξει μία τράπουλα ανεβάζοντας κάρτες από τον υπολογιστή του και στην συνέχεια να δημιουργήσει τα εργαλεία και τους κανόνες του παιχνιδιού μέσα από την φόρμα δημιουργίας του παιχνιδιού. Στην συνέχεια ο χρήστης έχει την δυνατότητα να δημιουργήσει ένα τραπέζι και να φορτώσει το παιχνίδι που έφτιαξε ο ίδιος ή άλλοι χρήστες. Όλοι οι χρήστες που εισέρχονται στο τραπέζι είναι απλοί παίχτες που μπορούν να αλληλοεπιδρούν με τις κάρτες στο παιχνίδι όταν έρθει η σειρά τους. Ο δημιουργός του τραπέζιού χρίζεται αυτόματα ως ο Game Master του παιχνιδιού και έχει πρόσβαση σε όλες τις δυνατότητες του τραπέζιού αφού είναι ο μαέστρος του. Μπορεί να θέσει την σειρά που θα καθίσουν οι παίχτες στο τραπέζι, την ομάδα, τον ρόλο ή την κατάσταση για ένα παίχτη ή ακόμα και να αποβάλει κάποιο χρήστη από το τραπέζι. Έχει την δυνατότητα να μετακινεί κάρτες στο παιχνίδι καθ' όλη την διάρκεια του, ανεξάρτητα με ποιανού είναι η σειρά να παίξει και να διορθώσει κινήσεις παιχτών με το να επιστρέψει κάρτες σε προηγούμενη κατάσταση τους. Ο πίνακας της βαθμολογίας μπορεί να ενημερωθεί μόνο από αυτόν αλλά έχουν πρόσβαση σε αυτόν όλοι οι παίχτες. Όλοι οι παίχτες του παιχνιδιού έχουν την δυνατότητα να συνομιλήσουν μεταξύ τους μέσω της γραπτής ομαδικής συνομιλίας. Τέλος, υπάρχει ο διαχειριστής που έχει πρόσβαση σε όλες τις δυνατότητες της πλατφόρμας, με κύρια δυνατότητα του την επεξεργασία ή διαγραφή οποιασδήποτε πληροφορίας ανά πάσα στιγμή.

Η πλατφόρμα κατασκευάστηκε με την χρήση σύγχρονων τεχνολογιών που οι βασικές είναι οι HTML, CSS και JavaScript. Για την εφαρμογή του frontend χρησιμοποιήθηκε η Vue.js που είναι βιβλιοθήκη της JavaScript και τρέχει την ιστοσελίδα σε εικονικό DOM. Στην Vue εφαρμογή χρησιμοποιήθηκε η TypeScript για καλύτερη χρήση των τύπων των μεταβλητών και η Tailwind CSS που προσφέρει έτοιμες κλάσεις της CSS γλώσσας. Για την backend εφαρμογή χρησιμοποιήθηκε η βιβλιοθήκη Nest.js που είναι βιβλιοθήκη της Node.js σε συνδυασμό με την TypeScript και με το πακέτο Express.js που είναι για την δημιουργία ενός διακομιστή. Η βάση δεδομένων που επιλέχθηκε για αυτή την πλατφόρμα είναι MySQL που είναι σχεσιακή βάση δεδομένων. Στην backend εφαρμογή γίνονται κλήσεις προς αυτή μέσω ειδικών εντολών αντικειμένων του πακέτου TypeORM και όχι με SQL εντολές. Σε κάθε φόρμα εισαγωγής δεδομένων, γίνεται έλεγχος δεδομένων πριν σταλθούν από την frontend στην backend εφαρμογή και γίνονται έλεγχοι και στην backend εφαρμογή πριν την είσοδο τους στους controller ώστε να διασφαλιστεί έτσι η ασφάλεια του συστήματος. Επίσης, οι κωδικοί πρόσβασης των χρηστών στην βάση δεδομένων είναι κρυπτογραφημένοι, όπως και τα κλειδιά JWT (Json Web Token) που απαιτούνται για να πραγματοποιηθούν κλήσεις από το frontend στο backend μεγαλώνοντας έτσι κι' άλλο το τοίχος προστασίας της πλατφόρμας.

## 5.2 Μετρικά του συστήματος

Σε αυτή την ενότητα αναφέρονται το πλήθος των φακέλων, των αρχείων ανά τον τύπο τους, οι γραμμές κώδικα, τα σχόλια, οι κενές γραμμές, το συνολικό μέγεθος των εφαρμογών και το συνολικό μέγεθος των μεταγλωττισμένων αρχείων.

### 5.2.1 Μετρικά κώδικα front-end

Στο παρακάτω πίνακα παρουσιάζονται οι μετρικές του κώδικα της frontend εφαρμογής με συνολικό μέγεθος 1.83MB.

Τύπος αρχείου	Αρχεία	Γραμμές κώδικα	Κενές γραμμές	Σχόλια
ENV	3	21	3	0
JavaScript	2	166	2	1
TypeScript	65	2207	92	80
JSON	4	6795	3	0
HTML	1	14	1	0
SCSS	2	60	17	2
Σύνολο	77	9263	118	83

Πίνακας 2 - Μετρικά κώδικα frontend εφαρμογής

Στο επόμενο πίνακα παρουσιάζονται οι μετρικές του μεταγλωττισμένου του κώδικα με συνολικό μέγεθος 1.10MB.

Τύπος αρχείου	Αρχεία	Γραμμές κώδικα	Κενές γραμμές	Σχόλια
JavaScript Files	66	94	66	0
HTML	1	15	2	0
Σύνολο	67	109	68	0

Πίνακας 3 - Μετρικά μεταγλωττισμένου κώδικα frontend εφαρμογής

## 5.2.2 Μετρικά κώδικα back-end

Στο παρακάτω πίνακα παρουσιάζονται οι μετρικές του κώδικα της backend εφαρμογής με συνολικό μέγεθος 1.19MB.

Τύπος αρχείου	Αρχεία	Γραμμές κώδικα	Κενές γραμμές	Σχόλια
ENV	3	60	18	0
JavaScript	1	25	1	0
TypeScript	103	4506	682	202
JSON	6	22567	6	0
HBS	2	570	0	10
Σύνολο	115	27728	707	212

Πίνακας 4 - Μετρικά κώδικα backend εφαρμογής

Στο επόμενο πίνακα παρουσιάζονται οι μετρικές του μεταγλωττισμένου του κώδικα με συνολικό μέγεθος 1.06MB.

Τύπος αρχείου	Αρχεία	Γραμμές κώδικα	Κενές γραμμές	Σχόλια
MAP	100	100	0	0
JavaScript	100	6523	0	100
TypeScript	100	1183	100	6
HBS	2	570	36	10
Σύνολο	302	8376	136	116

Πίνακας 5 - Μετρικά μεταγλωττισμένου κώδικα backend εφαρμογής

## 5.3 Ανάλυση SWOT

Η ανάλυση SWOT χρησιμοποιείται από αρκετούς οργανισμούς για την αξιολόγηση των εσωτερικών δυνατοτήτων, αδυναμιών, καθώς και εξωτερικών ευκαιριών και απειλών. Εξετάζοντας τις τέσσερις διαστάσεις του SWOT – δυνατά σημεία (Strengths), αδυναμίες (Weakness), ευκαιρίες (Opportunities) και απειλές (Threats) – οι οργανισμοί αποκτούν μια ολοκληρωμένη κατανόηση του ανταγωνιστικού τους τοπιού, δίνοντας τους έτσι την δυνατότητα να αξιοποιήσουν τα δυνατά τους σημεία, να αντιμετωπίσουν τις αδυναμίες, να εκμεταλλευτούν ευκαιρίες και να μετριάσουν πιθανές απειλές. Η ανάλυση SWOT χρησιμεύει ως πολύτιμο εργαλείο στους οργανισμούς για να αναπτύξουν



αποτελεσματικές στρατηγικές, να βελτιώσουν τη θέση τους στην αγορά και να παραμείνουν μπροστά σε ένα δυναμικό και ανταγωνιστικό επιχειρηματικό περιβάλλον.

### 5.3.1 Δυνατά σημεία

- Δημιουργία πρωτότυπων παιχνιδιών με κάρτες
- Δίνεται η δυνατότητα στους χρήστες να χρησιμοποιήσουν κάρτες ή τράπουλες ή παιχνίδια που δημιουργήθηκαν από άλλους
- Εύκολη περιήγηση στην πλατφόρμα καθώς υπάρχουν οι απαραίτητες λειτουργίες για ένα χρήστη ώστε να μην απαιτούνται ιδιαίτερες γνώσεις
- Πλήρης έλεγχος της πλατφόρμας από την σελίδα του διαχειριστή
- Ενημέρωση των δεδομένων στο προθάλαμο και στο HT σε πραγματικό χρόνο
- Δυνατότητα επικοινωνίας των παιχτών που ανήκουν στον ίδιο HT
- Πλήρης έλεγχος του HT από τον GM
- Δωρεάν χρήση της πλατφόρμας
- Δεν περιέχει διαφημίσεις που προκαλούν δυσάρεστη εμπειρία περιήγησης

### 5.3.2 Αδυναμίες

Μία από τις αδυναμίες της εφαρμογής είναι ότι ένας νέος εγγεγραμμένος χρήστης που θέλει να παίξει κάποιο ΠΜΚ το οποίο δεν υπάρχει στα κοινόχρηστα, θα χρειαστεί να το δημιουργήσει από την αρχή, κάτι που χρειάζεται κάποιο περισσότερο χρόνο από το να εισέλθει απλά σε ένα HT. Αυτό ίσως αποθαρρύνει κάποιους χρήστες από το να παραμείνουν στην πλατφόρμα. Ακόμα μία αδυναμία είναι ότι η πλατφόρμα δεν έχει δοκιμαστεί στην παράλληλη χρήση της από μεγάλο αριθμών χρηστών. Συνεπώς δεν μπορούν να προσδιοριστούν αστοχίες που ενδεχομένως να προκύψουν σε περίπτωση υπερφόρτωσης της εφαρμογής.

### 5.3.3 Ευκαιρίες

Στην σημερινή ψηφιακή εποχή, η ζήτηση για διαδραστικές εμπειρίες με ηλεκτρονικά παιχνίδια αυξάνεται συνεχώς. Η πλατφόρμα προσφέρει μια καινοτόμο λύση καθώς δίνει τη δυνατότητα στους χρήστες να απελευθερώσουν τη δημιουργικότητα τους σχεδιάζοντας τα δικά τους ΠΜΚ, προσαρμοσμένα στις προτιμήσεις και τα ενδιαφέροντα τους. Η πλατφόρμα υποστηρίζει την δυνατότητα πολλών παιχτών σε πραγματικό χρόνο και έτσι οι παίκτες μπορούν να συνδεθούν με άλλους λάτρεις των ΠΜΚ, δημιουργώντας μια ζωντανή και ελκυστική κοινότητα. Η πλατφόρμα πέρα από το παιχνίδι, προσφέρει κοινωνική αλληλεπίδραση και ανταγωνισμό για τους χρήστες που αναζητούν μια νέα και καθηλωτική εμπειρία παιχνιδιού.

### 5.3.4 Απειλές

Καθώς η πλατφόρμα είναι ανοικτή σε όλο το κοινό και για όλες τις ηλικίες μπορεί να προκύψουν ανεπιθύμητες συμπεριφορές κατά την διάρκεια του παιχνιδιού είτε για κακή συμπεριφορά, είτε χρήστες να κάνουν σαμποτάζ στο παιχνίδι και άλλα πολλά. Έχουν δοθεί αρκετές δυνατότητες στον GM του HT για να είναι σε θέση να διαχειριστεί τέτοιου είδους παίχτες από το HT. Ιδανικότερα, η διαχείριση αυτών των καταστάσεων θα έπρεπε να γινόταν αυτοματοποιημένα ώστε να απλοποιηθεί το έργο του GM και να καταγράφονται τα συμβάντα ώστε να αποβάλλονται οι κακόβουλοι χρήστες.

## 5.4 Πλάνο ελέγχου ορθής λειτουργίας

Σε αυτό το κεφάλαιο θα διασφαλιστεί η ποιότητα της πλατφόρμας τόσο για το backend, τόσο και για το frontend. Αυτό το κεφάλαιο θα καλύψει δύο κρίσιμες πτυχές: την ανάπτυξη των εφαρμογών σε περιβάλλον παραγωγής και την συνολική δοκιμή του. Η ανάπτυξη στην παραγωγή περιλαμβάνει προσεκτικό σχεδιασμό και εκτέλεση της διαδικασίας ανάπτυξης για να εγγυηθεί η ομαλή μετάβαση από την ανάπτυξη σε ένα ζωντανό περιβάλλον. Από την άλλη πλευρά οι δοκιμές διαδραματίζουν καθοριστικό ρόλο στον εντοπισμό και στην αντιμετώπιση τυχόν σφαλμάτων ή προβλημάτων, διασφαλίζοντας επιπλέον και ότι η εφαρμογή αποδίδει βέλτιστα και προσφέρει μια αξέχαστη εμπειρία χρήστη.

### 5.4.1 Περιβάλλον παραγωγής

Το επόμενο βήμα μετά την ολοκλήρωση της διαδικασίας ανάπτυξης της πλατφόρμας είναι να αναπτυχθεί το περιβάλλον φιλοξενίας της. Ο Hostinger είναι ένας δημοφιλής πάροχος φιλοξενίας για ιστοσελίδες και το Oracle Cloud προσφέρει μια ισχυρή πλατφόρμα για την φιλοξενία και εκτέλεση εφαρμογών, παρέχοντας επεκτασιμότητα, ασφάλεια και αξιοπιστία. Παρακάτω θα περιγράψει η διαδικασία βήμα προς βήμα τόσο για την frontend εφαρμογή, όσο και για το backend.

#### Frontend

Αρχικά πρέπει να δημιουργηθεί μία διεύθυνση που θα φιλοξενήσει την εφαρμογή. Στην συνέχεια από τις ρυθμίσεις του Hostinger και του GitHub θα πρέπει να πραγματοποιηθεί η ζεύξη ανάμεσα τις πλατφόρμες, ώστε η σελίδα να είναι πλήρως ενημερωμένη σε τυχόν νέες αλλαγές από το main branch. Έπειτα, πρέπει να γίνει η επεξεργασία του αρχείου .env.production ώστε να συμπληρωθούν οι απαραίτητες τιμές στις μεταβλητές που αφορούν την επικοινωνία των δύο εφαρμογών. Από τις ρυθμίσεις του Hostinger πρέπει να οριστεί να διαβάζεται η αρχική σελίδα από τον φάκελο dist, καθώς αυτός ο φάκελος περιέχει την μεταγλωττισμένη και βελτιστοποιημένη έκδοση της εφαρμογής Vue.js σε αρχεία JavaScript. Τέλος για να δημιουργηθεί ο μεταγλωττισμένος κώδικας αρκεί να τρέξει η εντολή `npm run build --prod` στην εφαρμογή και ακολούθως να ανέβουν οι αλλαγές στο main branch στο GitHub.

## Backend

Το Hostinger δεν προσφέρει υπηρεσίες για την node.js στο shared hosting πλάνο και για αυτό τον λόγο είναι αδύνατο να τρέξει το backend που δεν είναι απλά, μία ιστοσελίδα. Τέτοιου τύπου εφαρμογές χρειάζονται ιδιωτικούς εικονικούς διακομιστές (Virtual Private Server) που παρέχουν περισσότερες δυνατότητες στην εφαρμογή.

Η διαδικασία ξεκινάει με την δημιουργία μια παρουσίας (instance) Ubuntu στην πλατφόρμα Oracle Cloud. Αυτή η παρουσία διακρίνεται από μία μοναδική ιδιωτική διεύθυνση που για λόγους ασφάλειας ως προς την απόκρυψη της θα πρέπει δημιουργηθεί μία εύχρηστη διεύθυνση URL που θα συσχετίζεται με αυτή. Αυτό πραγματοποιείται από την επεξεργασία των ρυθμίσεων DNS (Domain Name System) όπου θα πρέπει να οριστεί και η πόρτα που θα τρέχει η εφαρμογή για να διασφαλιστεί η σωστή πρόσβαση σε αυτή.

Από το VPS γίνεται η εγκατάσταση της MySQL και η δημιουργία της βάσης δεδομένων, ώστε να τρέχει μαζί με την εφαρμογή και να αποθηκεύει τα δεδομένα σε αυτή.

Με την χρήση του GitHub θα γίνει η φόρτωση της εφαρμογής στο VPS από το main branch του repository. Έπειτα γίνεται η εγκατάσταση της node.js στο VPS ώστε να εγκατασταθούν όλα τα απαιτούμενα πακέτα που χρειάζονται για να τρέξει η εφαρμογή με την εντολή `npm install`. Στην συνέχεια πρέπει να δημιουργηθεί το `.env` αρχείο που περιέχει κλειδιά ή κωδικούς που σχετίζονται με την εφαρμογή ώστε να μπορεί να τρέξει η εφαρμογή και να είναι σε θέση να επικοινωνήσει με την frontend εφαρμογή. Για λόγους ευκολίας υπάρχει το αρχείο `.env.example` που περιέχει τις ονομασίες των μεταβλητών που θα πρέπει να υπάρχουν στο `.env` αρχείο. Όπως και στην Vue.js, η Nest.js δημιουργεί το dist φάκελο μέσω της εντολής `npm run build --prod` που περιέχει την μεταγλωττισμένη και βελτιστοποιημένη έκδοση της εφαρμογής σε αρχεία JavaScript.

Τέλος, για να συνεχίσει να τρέχει η εφαρμογή στο παρασκήνιο χωρίς να σταματήσει όταν τερματιστεί η σύνδεση με το VPS, απαιτείται η εγκατάσταση του πακέτου pm2. Μέσω της εντολής `pm2 start app-name` τρέχει η εφαρμογή στο παρασκήνιο μέχρι να τερματιστεί από τον διαχειριστή με την εντολή `pm2 stop app-name`.

### 5.4.2 Δοκιμή

Η διασφάλιση της ορθής λειτουργίας της πλατφόρμας αποτελεί σημαντικό παράγοντα για την επιτυχή σχεδίαση και ανάπτυξη ενός πληροφοριακού συστήματος. Σε πραγματικές εφαρμογές που χρησιμοποιούνται από μεγάλες εταιρείες υπάρχει ένα περιβάλλον που είναι μόνο για δοκιμές ώστε να διασφαλιστεί ότι η νέα έκδοση της εφαρμογής τους, τρέχει σωστά χωρίς προβλήματα πριν να γίνει η ενημέρωση στο περιβάλλον παραγωγής. Σε αυτό το στάδιο, μπορεί να προκύψουν προβλήματα που δεν υπήρχαν στο περιβάλλον του προγραμματιστή ή ακόμη να προκύψουν και νέες ανάγκες ώστε να προσφέρουν στον χρήστη καλύτερη εμπειρία με την πλατφόρμα τους.

Καθώς η παρούσα πλατφόρμα που δημιουργήθηκε βρίσκεται στην πρώτη έκδοση της, οι δοκιμές της πλατφόρμας έγιναν στο περιβάλλον παραγωγής.

Η πρώτη δοκιμή της πλατφόρμας πραγματοποιήθηκε αποκλειστικά από τον προγραμματιστή της με την χρήση δύο (2) ηλεκτρονικών υπολογιστών και από δύο (2)

προγράμματα περιήγησης στο κάθε υπολογιστή ώστε να πραγματοποιηθούν παιχνίδια με παράλληλους χρήστες. Αρχικά, δημιουργήθηκαν οι χρήστες για το κάθε πρόγραμμα περιήγησης και πραγματοποιήθηκε επιτυχώς η εγγραφή τους, η επανέκδοση κωδικού πρόσβασης, η σύνδεση τους και η έξοδος τους με την πλατφόρμα. Μετά την επιτυχή σύνδεση των εγγεγραμμένων χρηστών δημιουργήθηκαν οι κάρτες, οι τράπουλες και τα παιχνίδια τα οποία και αναλύθηκαν στο 3ο κεφάλαιο (Blackjack, Μπιρίμπα, Πόκερ, Υπο, Sabotuer) από διαφορετικούς χρήστες με επιτυχία. Στην συνέχεια πραγματοποιήθηκαν τα πρώτα παιχνίδια στο περιβάλλον παραγωγής με διαφορετικά σενάρια ώστε να διασφαλιστεί αν η πλατφόρμα λειτουργεί επιτυχώς χωρίς προβλήματα. Με την ολοκλήρωση της πρώτης δοκιμής, δυστυχώς η πλατφόρμα δεν δούλευε σωστά. Η δυνατότητα του Game Master να επιστρέφει τις κάρτες σε προηγούμενες καταστάσεις ώστε να διορθώνει κινήσεις παιχτών δεν δούλευε και επίσης όταν κάποιος χρήστης αποχωρούσε από το παιχνίδι που βρισκόταν σε εξέλιξη λόγω κακής σύνδεσης ή εσκεμμένα τότε το HT δεν ανταποκρινόταν. Επιπρόσθετα, κρίθηκε η ανάγκη να προστεθούν ακόμα κάποιες δυνατότητες στους παίχτες ώστε να μπορούν να δίνουν κάποια κάρτα τους σε άλλο παίχτη ή να μαζεύουν σε μία κίνηση όλες τις κάρτες από το τραπέζι ή από μία τράπουλα.

Καθώς διορθώθηκαν τα προβλήματα που προέκυψαν στην προηγούμενη δοκιμή και προστέθηκαν οι νέες δυνατότητες στο παίχτη, άρχισε ο δεύτερος γύρος δοκιμών της πλατφόρμας όπου ο χρήστης της πλατφόρμας με τον ρόλο του επισκέπτη ήταν ο επιβλέπων καθηγητής της ΠΕ, Μηνάς Δασυγένης. Πραγματοποιήθηκαν δύο παιχνίδια με παράλληλα προγράμματα περιήγησης και ολοκληρώθηκαν επιτυχώς χωρίς κάποιο πρόβλημα. Μετά την ολοκλήρωση του δεύτερου κύκλου δοκιμών αποφασίστηκε να δοθεί η δυνατότητα στον χρήστη να ανεβάζει πολλαπλές κάρτες ταυτόχρονα ώστε η πλατφόρμα να είναι πιο εύκολη και φιλική προς τον χρήστη.

Ο τρίτος και τελευταίος κύκλος δοκιμών έγινε με 5 χρήστες όπου κάποιοι από αυτούς ήταν και σε διαφορετικές χώρες και οι τέσσερις από αυτούς την χρησιμοποίησαν για πρώτη φορά. Ο ένας από τους νέους χρήστες, ανάλαβε να δημιουργήσει ένα παιχνίδι από κάποιο δημοφιλές ΠΜΚ και ένας άλλος χρήστης ένα τροποποιημένο δημοφιλές παιχνίδι. Πραγματοποιήθηκαν και ολοκληρώθηκαν τα παιχνίδια επιτυχώς χωρίς κάποιο πρόβλημα. Δοκιμάστηκε από τους χρήστες να αποσυνδεθούν από παιχνίδι που βρίσκεται σε εξέλιξη είτε εσκεμμένα, είτε να τερματίσουν την σύνδεση τους στο διαδίκτυο και η πλατφόρμα ενημέρωνε σωστά τους υπολοίπους χρήστες για την αποσύνδεση των χρηστών. Κάποιοι χρήστες λάμβαναν τα μηνύματα σε διαφορετικά χρονικά σημεία με μέγιστη καθυστέρηση τα 5 δευτερόλεπτα. Αυτό έχει να κάνει με το διακομιστή που φιλοξενείται η πλατφόρμα και έτσι αυτός ο κύκλος δοκιμών κρίνεται ως επιτυχής.

## 5.5 Μελλοντικές επεκτάσεις

Οι μελλοντικές επεκτάσεις μίας εφαρμογής είναι απαραίτητες ώστε να διορθώνονται τυχόν προβλήματα που εμφανίζονται από την χρήση των χρηστών αλλά και να δίνονται νέες βελτιώσεις όπου οι χρήστες θα θέλουν να παραμένουν χρήστες της πλατφόρμας και να μπορεί να αυξηθεί η κοινότητάς της. Μερικές επεκτάσεις που θα μεγάλωναν το ενδιαφέρον των χρηστών είναι οι εξής:

- **Η πλατφόρμα να υποστηρίζεται και σε μικρότερες οθόνες συσκευών:** Λόγω του μεγάλου όγκου της εφαρμογής δεν ήταν δυνατό να γίνει παράλληλα η ανάπτυξη της πλατφόρμας για συσκευές μικρότερες από 1000 pixels και θα ήταν σημαντικό προσόν για την πλατφόρμα αν υποστηριζόταν από όλες τις συσκευές
- **Ζωντανή συνομιλία των χρηστών με την χρήση του μικροφώνου:** Καθώς το παιχνίδι διαδραματίζεται σε πραγματικό χρόνο θα ήταν σημαντικό εργαλείο η ζωντανή ομιλία για τους παίκτες. Θα μπορούσαν να μιλήσουν για την ενέργεια τους αντί να το γράψουν σε γραπτή συνομιλία ή και με την δήλωση της κατάστασης του παίχτη.
- **Υποστήριξη δημοφιλών παιχνιδιών που θα τρέχουν αυτόματα χωρίς την χρήση του GM:** Με αυτή την επέκταση θα προσέγγιζε και παίκτες που θέλουν να παίξουν τα κλασσικά παιχνίδια με κάρτες που το παιχνίδι κινείται αυτόματα.
- **Δημιουργία δωματίου δοκιμής παιχνιδιού:** Θα υπάρχει ένας ειδικά διαμορφωμένος χώρος που η πλατφόρμα θα δίνει στο χρήστη δοκιμαστικούς λογαριασμούς σε ένα δωμάτιο και αυτός θα μπορεί να πάρει τον έλεγχο τους ώστε να δοκιμάσει το παιχνίδι μόνος του πριν το φορτώσει σε κάποιο ΗΤ.
- **Χρήση ηλεκτρονικών ρομπότ στο παιχνίδι:** Θα χρησιμοποιούνται ηλεκτρονικά ρομπότ στα παιχνίδια στην θέση του παίκτη ώστε να καλύπτεται ο ελάχιστος αριθμός των παιχτών όταν δεν υπάρχουν διαθέσιμοι παίκτες. Επιπρόσθετα, τα ηλεκτρονικά ρομπότ θα μπορούσαν να χρησιμοποιηθούν με τεχνική νοημοσύνη ώστε να μαθαίνουν μέσα από το παιχνίδι και να είναι σε θέση να παίζουν καλύτερα από ένα πραγματικό παίχτη.
- **Αξιολόγηση παιχνιδιών και Game Master:** Τα παιχνίδια θα δέχονται αξιολογήσεις ώστε να γνωρίζουν οι υπόλοιποι χρήστες αν είναι ενδιαφέροντα για να τα χρησιμοποιήσουν. Το παιχνίδι εξαρτάται σε ένα μεγάλο βαθμό από τον GM, επομένως θα ήταν χρήσιμο για τους παίκτες να γνωρίζουν την αξιολόγηση του GM από άλλους παίκτες.
- **Καταγραφή ανεπιθύμητων χρηστών:** Η εφαρμογή θα πρέπει να υποστηρίζει στο μέλλον αναφορές για ανεπιθύμητους χρήστες ώστε να αποβάλλονται από την πλατφόρμα.

## 5.6 Συμπεράσματα

Τα συμπεράσματα που προκύπτουν από την ολοκλήρωση της παρούσας πτυχιακής εργασίας είναι ποικίλα και σχετίζονται άμεσα με την λειτουργικότητα της πλατφόρμας. Αυτή η πλατφόρμα δίνει την δυνατότητα στους φανατικούς των παιχνιδιών με κάρτες να δημιουργήσουν τα δικά τους παιχνίδια, να τα δοκιμάσουν και να τα μοιραστούν με φίλους τους ή με άλλους χρήστες. Με την χρήση αυτού του εργαλείου οι χρήστες θα περάσουν την δοκιμασία της δημιουργίας του παιχνιδιού καταγράφοντας του κανόνες του, τις οδηγίες και εφαρμόζοντας την λειτουργικότητα του μέσω των εργαλείων που δίνονται από την πλατφόρμα.

## Παράρτημα

- **Προϋποθέσεις εγκατάστασης λογισμικών:**

Εγκατάσταση της MySQL

Εγκατάσταση της node.js (Version 16.10.0)

- **Οδηγίες Εγκατάστασης σε περιβάλλον ανάπτυξης:**

1. Αποσυμπίεση του αρχείου omcg.zip ή λήψη των 2 προγραμμάτων από το GitHub από τους συνδέσμους:

- **Frontend:** <https://github.com/Nikolas2610/OnlineMultiplayerCardGame-Frontend>

- **Backend:** <https://github.com/Nikolas2610/OnlineMultiplayerCardGame-Backend>

2. Δημιουργία βάσης δεδομένων με την εντολή “CREATE DATABASE online\_multiplayer\_card\_game;”

3. Άνοιγμα της κονσόλας στο υφιστάμενο φάκελο και ακολουθείτε τις εξής εντολές:

**Για το Backend project:**

```
cd omcg/online-multiplayer-card-game-backend
```

```
npm install ή npm i
```

**Για το Frontend project:**

```
cd omcg/online-multiplayer-card-game-frontend
```

```
npm install ή npm i
```

4. Στο αρχείο .env.development πρέπει να οριστούν οι κατάλληλες μεταβλητές που αφορούν την MySQL (MYSQL\_PASSWORD, MYSQL\_USERNAME, MYSQL\_DATABASE). Επιπρόσθετα, η πλατφόρμα χρησιμοποιεί την δυνατότητα να στέλνει email στους χρήστες, επομένως απαιτείτε να οριστούν και οι μεταβλητές MAIL\_HOST, MAIL\_USER, MAIL\_PASSWORD και APP\_EMAIL. Ο πάροχος email server που χρησιμοποιήθηκε σε αυτή την πτυχιακή εργασία είναι του [sendgrid](#).

- **Εισαγωγή ψεύτικων δεδομένων στην βάση δεδομένων:**

Υπάρχει η επιλογή να προσδεθούν ψεύτικα δεδομένα στην βάση δεδομένων ώστε να μπορεί ο προγραμματιστής να αλληλοεπιδράσει με την πλατφόρμα χωρίς να χρειαστεί να προσθέσει δεδομένα από τις φόρμες. Για να πραγματοποιηθεί αυτό απαιτείτε να έχει γίνει η εγκατάσταση της backend εφαρμογής και στην συνέχεια να τρέξει η εντολή **npm run seed**.

Αυτή η εντολή θα τρέξει την συνάρτηση fillData από το αρχείο:

```
online-multiplayer-card-game-backend/entities/seeders/fake-data.ts
```

Αφού εμφανιστούν τα μηνύματα επιτυχίας της εισαγωγής δεδομένων και το τελευταίο μήνυμα “Fake data added successfully” (όπως φαίνεται στην παρακάτω εικόνα), τότε έχει ολοκληρωθεί η διαδικασία.

```
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/admin/users, GET} route +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/admin/user/edit, PATCH} route +0ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/admin/user/delete, DELETE} route +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/admin/decks, GET} route +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/admin/deck, PATCH} route +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/admin/deck, DELETE} route +0ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/admin/games, GET} route +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/admin/tables, GET} route +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/admin/cards, GET} route +0ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/admin/card, PATCH} route +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/admin/card/image, PATCH} route +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/admin/card, DELETE} route +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RoutesResolver] UserController {/api/user}: +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/user/dashboard, GET} route +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/user/edit/username, PATCH} route +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/user/edit/password, PATCH} route +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/user/cards, GET} route +2ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/user/decks, GET} route +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/user/games, GET} route +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/user/edit/game, PATCH} route +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/user/delete/game, DELETE} route +0ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [RouterExplorer] Mapped {/api/user/tables, GET} route +1ms
[Nest] 18528 - 01/21/2023, 5:32:58 PM LOG [NestApplication] Nest application successfully started +5ms
Add users DONE
Add cards DONE
Add decks DONE
Add games DONE
Add tables DONE
Fake data added suffessfully
```

Εικόνα 66 – Στιγμιότυπο από την κονσόλα του backend λογισμικού με μήνυμα επιτυχής εισαγωγής ψεύτικων δεδομένων στην βάση δεδομένων

Σημείωση: Στην βάση δεδομένων δημιουργήθηκε ένας administrator με email: [admin@omcg.com](mailto:admin@omcg.com). Όλοι οι χρήστες που δημιουργήθηκαν (και ο administrator) έχουν κωδικό πρόσβασης: CardGame-0. Μερικά email απλών χρηστών είναι [player1@omcg.com](mailto:player1@omcg.com) μέχρι και τον αριθμό 8.

- **Τρέξτε τα λογισμικά**

Πρέπει να υπάρχουν δύο κονσόλες ανοιχτές ταυτόχρονα, που το ένα είναι για το backend λογισμικό και το άλλο για το frontend λογισμικό. Τρέξτε τις εξής εντολές από τον φάκελο omcg:

**Backend project**

cd online-multiplayer-card-game-backend

npm run start or npm run start:dev (Για να εμφανίζονται μηνύματα της κονσόλας)

**Frontend project**

cd online-multiplayer-card-game-frontend

npm run dev

Το backend τρέχει στην διεύθυνση <http://localhost:3000> και το frontend στην διεύθυνση <http://localhost:5173/>

## Βιβλιογραφία

- [1] N. Renu, "SAGE Journals," 9 3 2021. [Online]. Available: <https://journals.sagepub.com/doi/full/10.1177/205031212111000912>. [Accessed 10 10 2022].
- [2] D. R. – J. G. – J. Rydning, "The Digitization of the World - From Edge to Core," 2018.
- [3] R. R. Contreras, "Eurofound," 15 December 2021. [Online]. Available: <https://www.eurofound.europa.eu/data/digitalisation/research-digests/covid-19-and-digitalisation>. [Accessed 10 10 2022].
- [4] A. SEMUELS, "Time," 6 August 2020. [Online]. Available: <https://time.com/5876604/machines-jobs-coronavirus/>. [Accessed 10 10 2022].
- [5] N. Ko, "BestColleges," 11 9 2021. [Online]. Available: <https://www.bestcolleges.com/bootcamps/tracks/covid-impact-on-tech-hiring-trends/>. [Accessed 10 10 2022].
- [6] Werner Ballhaus, Wilson Chow, Emmanuelle Rivet, "PWC," PWC, 2022. [Online]. Available: <https://www.pwc.com/gx/en/industries/tmt/media/outlook/outlook-perspectives.html>. [Accessed 11 10 2022].
- [7] "Casino.org," Casino.org, 2022. [Online]. Available: <https://www.casino.org/features/gambling-statistics/>. [Accessed 12 10 2022].
- [8] "Dod.gr," Dod.gr, 10 10 22. [Online]. Available: <https://www.dod.gr/>. [Accessed 13 10 2022].
- [9] R. Kahn, "Britannica," 10 5 2022. [Online]. Available: <https://www.britannica.com/technology/Internet>. [Accessed 14 10 2022].
- [10] J. Assange, "Britannica," 28 7 2021. [Online]. Available: <https://www.britannica.com/technology/website>. [Accessed 16 10 2022].
- [11] him0000, "GeeksForGeeks," [Online]. Available: <https://www.geeksforgeeks.org/html-introduction/>. [Accessed 17 10 2022].
- [12] "W3Schools," W3Schools, [Online]. Available: <https://www.w3schools.com/css/>. [Accessed 17 10 2022].
- [13] A. Wathan, "Tailwindcss," 2022. [Online]. Available: <https://tailwindcss.com/>. [Accessed 17 10 2022].
- [14] "LaunchSchool," LaunchSchool, 2022. [Online]. Available: <https://launchschool.com/books/javascript/read/introduction>. [Accessed 17 10 2022].
- [15] "Stack overflow," 2022. [Online]. Available: <https://survey.stackoverflow.co/2022/#:~:text=Rust%20is%20on%20its%20seventh,typeScript%20running%20a%20close%20second.&text=Phoenix%20overtakes%20Svelte's%20spot%20as,Angular..> [Accessed 17 10 2022].
- [16] "W3Schools," 2022. [Online]. Available: <https://www.w3schools.com/typescript/index.php>. [Accessed 18 10 2022].
- [17] E. You, "Vue.js," 2022. [Online]. Available: <https://vuejs.org/>. [Accessed 19 10 2022].



- [18] E. You, "Vue.js," 2022. [Online]. Available: <https://vuejs.org/guide/essentials/lifecycle.html>. [Accessed 19 10 2022].
- [19] O. Media, "Ethan Brown, Web Development with Node and Express, 2th Edition," 2017.
- [20] K. Mysliwicz, "Nest.js," 2022. [Online]. Available: <https://docs.nestjs.com/>. [Accessed 20 10 2022].
- [21] "MySQL," Oracle, 2022. [Online]. Available: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>. [Accessed 26 10 2022].
- [22] "Type ORM," 2022. [Online]. Available: <https://typeorm.io/>. [Accessed 20 10 2022].
- [23] ArpitAsati, "GeekForGeeks," 2022. [Online]. Available: <https://www.geeksforgeeks.org/what-is-web-socket-and-how-it-is-different-from-the-http/>. [Accessed 21 10 2022].
- [24] "VS Code," Microsoft, 2022. [Online]. Available: <https://code.visualstudio.com/>. [Accessed 22 10 2022].
- [25] "Postman," Postman, 2022. [Online]. Available: <https://www.postman.com/>. [Accessed 22 10 2022].
- [26] "Figma," 2022. [Online]. Available: <https://www.figma.com/>. [Accessed 24 10 2022].
- [27] "SendGrid," SendGrid, 2022. [Online]. Available: <https://sendgrid.com/>. [Accessed 26 10 2022].
- [28] "Git," Software Freedom Conservancy, 2022. [Online]. Available: <https://git-scm.com/>. [Accessed 26 10 2022].
- [29] "MySQL," Oracle, 2022. [Online]. Available: <https://dev.mysql.com/doc/refman/8.0/en/enum.html>. [Accessed 20 10 2022].
- [30] "W3Schools," 2023. [Online]. Available: [https://www.w3schools.com/html/html5\\_draganddrop.asp](https://www.w3schools.com/html/html5_draganddrop.asp). [Accessed 25 3 2023].
- [31] Werner Ballhaus, Wilson Chow, Emmanuelle Rivet, "PWC," PWC, 2022. [Online]. Available: <https://www.pwc.com/gx/en/industries/tmt/media/outlook/outlook-perspectives.html>. [Accessed 12 10 2022].
- [32] "W3Schools," 2023. [Online]. Available: [https://www.w3schools.com/html/html\\_responsive.asp](https://www.w3schools.com/html/html_responsive.asp). [Accessed 26 3 2023].