

Νικόλας Ψύλλου

Πτυχιακή Εργασία HOU-CS-UGP-2022-13



Σχεδίαση και Υλοποίηση διαδικτυακής πλατφόρμας για εξυπηρέτηση παιχνιδιών καρτών απομακρυσμένων παικτών

Επιβλέπων: Μηνάς Δασυγένης

Μέλη ΕΕ: Αθανάσιος Κακαρούντας, Αλέπης Ευθύμιος

Εισαγωγή

- ▶ Η πτυχιακή εργασία αφορά την δημιουργία μιας διαδικτυακής πλατφόρμας για παιχνίδια με κάρτες η οποία θα εξυπηρετεί απομακρυσμένους μεταξύ τους παίκτες, προσφέροντας τους έτσι επικοινωνία σε πραγματικό χρόνο.
- ▶ Οι χρήστες της πλατφόρμας θα έχουν την μοναδική δυνατότητα να δημιουργήσουν, να προσαρμόσουν και να απολαύσουν παιχνίδια καρτών τόσο παραδοσιακά όσο και δικής τους δημιουργίας.

Δήλωση προβλήματος

- ▶ Η έλλειψη ποικιλίας παιχνιδιών με κάρτες που υπάρχει.
- ▶ Το γεγονός ότι δεν υπάρχει δυνατότητα τροποποίησης σε ότι αφορά κανόνες υφιστάμενων παιχνιδιών.
- ▶ Το κενό που υπάρχει στην αγορά για μια πλατφόρμα που προσφέρει πρωτότυπα παιχνίδια με κάρτες σε πραγματικό χρόνο.

Σχεδιαστικές προδιαγραφές

- ▶ Ο χρήστης να ανεβάζει εικόνες από κάρτες και να δημιουργεί τις δικές του τράπουλες.
- ▶ Ο χρήστης να δημιουργεί το δικό του παιχνίδι, με τους δικούς του κανόνες.
- ▶ Ο χρήστης να φορτώνει το παιχνίδι του στο τραπέζι ώστε να μπορεί να το παίξει με άλλους χρήστες.
- ▶ Ο Game Master να έχει το έλεγχο του τραπεζιού.
- ▶ Ο χρήστης να αλληλοεπιδρά σε πραγματικό χρόνο με άλλους παίχτες

Ανάλυση προβλήματος

Τα παιχνίδια με κάρτες που επιλέχθηκαν για να καταγραφούν οι βασικές προδιαγραφές παιχνιδιών με κάρτες είναι:

- ▶ Blackjack
- ▶ Μπιρίμπα
- ▶ Πόκερ
- ▶ Uno
- ▶ Saboteur

Κοινές ιδιότητες παιχνιδιών με κάρτες

- ▶ Ένας παίχτης μοιράζει τις κάρτες στους υπόλοιπους παίχτες, σε έξτρα τράπουλες και στο τραπέζι.
- ▶ Μία κάρτα στο παιχνίδι μπορεί να είναι κρυφή ,φανερή ή να είναι περιστρεφόμενη.
- ▶ Ανακάτεμα της τράπουλας.
- ▶ Ένα παιχνίδι μπορεί να αποτελείται από περισσότερες από μία τράπουλες.
- ▶ Υπάρχει πάντα μία τράπουλα που οι παίχτες τραβάνε κάρτες και ακόμη μία που είναι η στοίβα με τα σκάρτα.

Ρόλοι εγγεγραμμένων χρηστών στην πλατφόρμα

- ▶ Player
- ▶ Game Designer
- ▶ Game Master

Λειτουργικές απαιτήσεις Player

- ▶ Να έχει ένα μπλοκ στο παιχνίδι που θα αποτελείται από τις κάρτες του χεριού του.
- ▶ Θέτει τον εαυτό ότι η σειρά του να παίξει ώστε να του επιτρέπεται να μετακινεί κάρτες στο τραπέζι.
- ▶ Να μετακινεί κάρτες στο τραπέζι όταν είναι ή σειρά του.
- ▶ Να περιστρέφει κάρτες στο παιχνίδι και να είτε να τις θέτει σε κρυφές ή φανερές.
- ▶ Να συνομιλεί με τους άλλους χρήστες του τραπεζιού μέσω μιας ομαδικής γραπτής συνομιλίας.
- ▶ Θέτει την κατάσταση που βρίσκεται στο παιχνίδι.

Λειτουργικές απαιτήσεις Game Designer

- ▶ Να δημιουργεί κάρτες και τράπουλες για το παιχνίδι
- ▶ Να ορίζει τους μέγιστους παίχτες στο τραπέζι
- ▶ Να ορίζει τους ρόλους που θα έχουν οι παίχτες στο τραπέζι
- ▶ Να ορίζει τράπουλες με κάρτες και χωρίς που απαιτούνται στο παιχνίδι
- ▶ Να ορίζει τις ομάδες των παιχτών
- ▶ Να ορίζει το πλέγμα του παιχνιδιού
- ▶ Να ορίζει το αρχικό μοίρασμα των παιχτών

Λειτουργικές απαιτήσεις Game Master

- ▶ Να έχει ένα έξτρα επιπλέον πίνακα ενεργειών.
- ▶ Να αλλάζει την κατάσταση του τραπεζιού.
- ▶ Να αναιρεί κινήσεις κατά την διάρκεια του παιχνιδιού.
- ▶ Να ορίζει την σειρά που κάθονται στο τραπέζι οι παίχτες.
- ▶ Να ανακατεύει τις τράπουλες.
- ▶ Να εμφανίζει τις κάρτες του παιχνιδιού για καταμέτρηση των πόντων.
- ▶ Να ενημερώνει τον πίνακα βαθμολογίας του παιχνιδιού.

Σχεδιαστικές Επιλογές

Για την ανάπτυξης της πλατφόρμας επιλέχθηκαν οι γλώσσες προγραμματισμού:

- ▶ HTML, CSS, JavaScript

Για το frontend και για το backend χρησιμοποιήθηκε ως κύρια γλώσσα προγραμματισμού η JavaScript με framework την Vue.js και την Nest.js αντίστοιχα.

Η πλατφόρμα αναπτύχθηκε με σχεσιακή βάση δεδομένων, την MySQL.

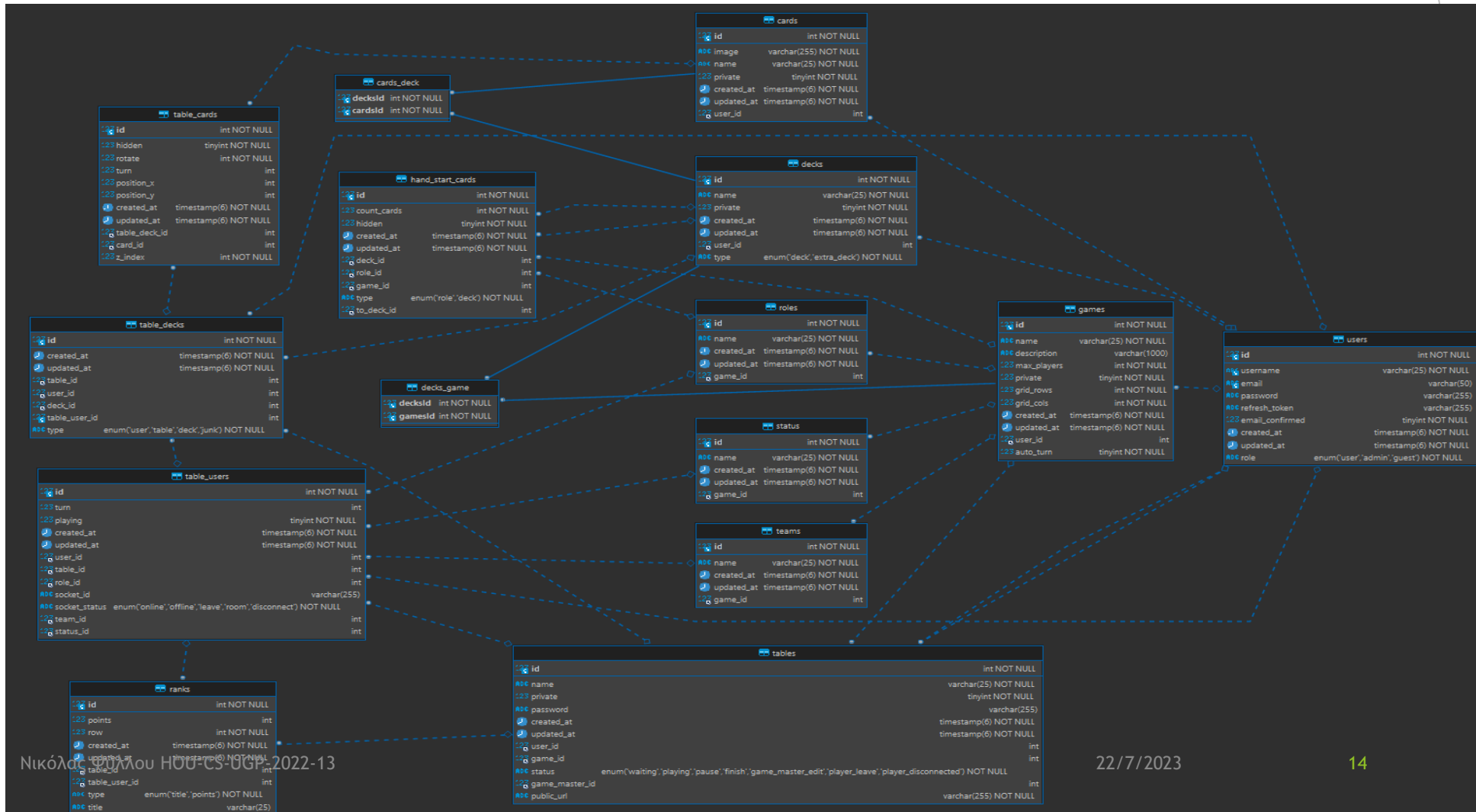
Vue.js

- ▶ Είναι μία ελαφριά βιβλιοθήκη της JavaScript, ιδανική για την δημιουργία γρήγορων αποδοτικών εφαρμογών.
- ▶ Όλες οι σελίδες της Vue τρέχουν σε μία σελίδα (Single Page App) και αυτό επιτυγχάνεται ότι η Vue τρέχει σε εικονικό DOM.
- ▶ Γίνεται η επαναχρησιμοποίηση τμημάτων κώδικα βελτιώνοντας την παραγωγικότητα και την συντηρητικότητα.
- ▶ Ένα αρχείο της Vue αποτελείται από HTML, CSS, JavaScript που γράφεται ο ανάλογος κώδικας στις τρεις (3) ετικέτες της: `<template>`, `<script>`, `<style>`.

Nest.js

- ▶ Η Nest.js είναι βιβλιοθήκη της Node.js, όπου με την εγκατάσταση της, παρέχεται το πακέτο Express για την δημιουργία εφαρμογών στην πλευρά του διακομιστή.
- ▶ Υποστηρίζεται με TypeScript και συνδυάζει στοιχεία OOP (Object Oriented Programming) και FP (Functional Programming).
- ▶ Υποστηρίζει αρχιτεκτονική επιπέδων που κάθε επίπεδο της αποτελείται από το Module, Controller και το Service
- ▶ Παρέχει πλούσιο οικοσύστημα τόσο για την χρήση των middleware, WebSocket's και άλλα.

Διάγραμμα Βάσης δεδομένων



Παράδειγμα επικοινωνίας του frontend με του backend

Vue.js

```
async fetchGames() {  
  try {  
    this.$state.loading = true;  
    const { data: games }: AxiosResponse = await axiosUser.get(`/${this.$state.role}/games`);  
    this.$state.games = games;  
    this.$state.loading = false;  
  } catch (error) {  
    process.env.NODE_ENV === 'development' ? console.log(error) : ''  
  }  
},
```

Παράδειγμα επικοινωνίας του frontend με του backend

Nest.js - Controller

```
@Get('games')
findAllGames() {
  return this.adminService.findAllGames();
}
```

Nest.js - Service

```
async findAllGames() {
  try {
    const games = await this.gamesRepository.find({
      relations: ['roles', 'hand_start_cards', 'hand_start_cards.role',
        'hand_start_cards.deck', 'hand_start_cards.toDeck', 'teams', 'status', 'deck'
      ],
      order: {
        created_at: 'DESC'
      }
    });
    return games;
  } catch (error) {
    throw new HttpException({
      status: HttpStatus.BAD_REQUEST, error
    },
      HttpStatus.BAD_REQUEST
    );
  }
}
```


Αλληλεπίδραση σε πραγματικό χρόνο

- ▶ Πραγματοποιήθηκε με την χρήση των WebSocket's από την πλευρά του διακομιστή και με την χρήση του πακέτου socket.io για το frontend.
- ▶ Τα WebSocket's πραγματοποιούν την σύνδεση (handshake) του πελάτη με το διακομιστή με το HTTP πρωτόκολλο αλλά η σύνδεση τους παραμένει ανοιχτή για να πραγματοποιηθεί η άμεση επικοινωνία τους.
- ▶ Τα WebSocket's χρησιμοποιούνται συνήθως για ζωντανές συνομιλίες, παιχνίδια πολλαπλών παικτών και άλλα πολλά.

Δοκιμή

- ▶ Οι δοκιμές της πλατφόρμας πραγματοποιήθηκαν σε περιβάλλον παραγωγής (production environment).
- ▶ Η πλατφόρμα δοκιμαστικέ από 4 χρήστες σε πραγματικό χρόνο όπου ολοκλήρωσαν δύο γύρους σε 3 παιχνίδια (Blackjack, Poker, Uno) και κρίθηκε η πλατφόρμα επιτυχής στην εφαρμογή της.

Συμπεράσματα

- ▶ **Δημιουργία πρωτοποριακής πλατφόρμας:** Η πλατφόρμα που αναπτύχθηκε παρέχει μια διαδραστική εμπειρία στους χρήστες, επιτρέποντάς τους να παίζουν παιχνίδια με κάρτες σε πραγματικό χρόνο, είτε με κανονικές τράπουλες είτε με ειδικές.
- ▶ **Ευρύτερη Πρόσβαση:** Η πλατφόρμα δίνει τη δυνατότητα σε χρήστες από όλο τον κόσμο να παίζουν τα αγαπημένα τους παιχνίδια με κάρτες, ανεξάρτητα από την απόσταση.
- ▶ **Καινοτομία στα Παιχνίδια με Κάρτες:** Οι χρήστες μπορούν να παίζουν παιχνίδια που δεν υπάρχουν στο διαδίκτυο, ή να προσθέσουν τροποποιήσεις σε υπάρχοντα παιχνίδια.

Εμπειρία

- ▶ Η εμπειρία από την ανάπτυξη αυτής της πλατφόρμας ήταν πολύτιμη. Αντιμετώπισα προκλήσεις, όπως η υλοποίηση της λειτουργικότητας σε πραγματικό χρόνο και η διασφάλιση της ομαλής λειτουργίας της πλατφόρμας για όλους τους χρήστες και η ασφάλεια των προσωπικών δεδομένων των χρηστών. Αυτές οι προκλήσεις με βοήθησαν να εμβαθύνω τις γνώσεις μου σε γλώσσες προγραμματισμού όπως η JavaScript και τεχνολογίες όπως η Vue.js και η Nest.js.

Μελλοντικές Επεκτάσεις

- ▶ Ζωντανή συνομιλία των χρηστών με την χρήση του μικροφώνου.
- ▶ Υποστήριξη δημοφιλών παιχνιδιών που θα τρέχουν αυτόματα χωρίς την χρήση του GM.
- ▶ Χρήση τεχνική νοημοσύνης για την προσφορά ειδικών αντιπάλων στους χρήστες.
- ▶ Δημιουργία δωματίου δοκιμής παιχνιδιού.

Demo Πλατφόρμας και Ερωτήσεις