JACOBS UNIVERSITY

Software Engineering Project

SPRINT 03 - DOCUMENTATION

April 13, 2021

Team 26
Nayan Man Singh Pradhan
Aabishkar Karki

Contents

1 Team Members

- Aabishkar Karki
- Nayan Man Singh Pradhan

2 Introduction

For this Sprint, we received a simple project with just a login and signup page built on React and CSS for frontend and no proper backend. Hence, first we connected the frontend to our database using Fask. After creating a functional backend, we changed the overall user interface for the web applciation by adding responsive and clean background and buttons with navbars and footers. Since the code we received only had 1 page (login/signup), we had to create several other pages such that the project is up to date with the Sprint. We added functions where Instructors can create and host games as well as monitor the games they created. We also added the functionality for Players to join a game created by the Insturctor and a Game Screen where the Players get an idea of how the game is played.

3 Software Requirements

- The software must provide a clean, simple to follow, and scalable frontend GUI.
- The software must redirect clients to the specified pages and forms requested by the clients through links and buttons.
- The software must provide a platform for Players and Instructors to securely SignUp through the 'SignUp' form.
- The software must provide a platform for Players and Instructors to securely LogIn through the 'LogIn' form.
- The software must provide a platform for the Instructors to create/host a game with an autogenerated unique game id, custom game password, and game attributes.
- The software must provide a platform for the Instructors to monitor the games they are hosting.
- The software must provide a platform for the Instructors to edit and update the game details for the games that they are hosting.
- The software must provide logical constraints for game parameters (eg. number of rounds must be greater than 0).
- The software must provide a platform for the Players to join a game using the unique game id and the password set by the Instructor hosting the game.

- The software must provide different home pages for the Instructors and Players after they have logged in.
- The software must provide a page where Players can order items in order to play the game.
- The software must provide necessary backend components in order to store the game details.

4 Setup and Deployment

1. Clone the repository into your local machine:

   ```
   $ git clone https://github.com/lorenzorota/se-03-team-25
   ```

2. Make sure you have python virtual env installed. To install virtual env run:

   For Linux Systems

   ```
   $ pip3 install virtualenv
   $ virtualenv venv
   $ source venv/bin/activate
   $ pip3 install -r requirements.txt
   ```

   For Windows Systems:

   ```
   $ python3 -m venv new_venv
   $ new-venv\Scripts\activate.bat
   $ pip3 install -r path\to\requirements.txt
   ```

3. Run the inital db migration from the root backend directory:

   ```
   $ yoyo apply
   ```

4. From the root backend directory run:

   ```
   $ python3 main.py
   ```

5. Make sure you have 'ReactJS' installed. To install 'ReactJS' run:

   ```
   $ sudo apt install npm
   $ sudo npm -g install create-react-app
   ```

6. In frontend directory install the required packages and enter development mode by running:

   ```
   $ npm install
   $ npm start
   ```

7. In order to check the database, run:

   ```
   $ python3 view_table_data.py
   ```

5 Testing

In order to start the unit test for the backend, go to the backend directory run:

```
$ python3 connection_test.py
```

In order to start the unit test for the frontend, go to the frontend directory run:

```
$ npm test
```

6 Libraries and Frameworks Used

- python 3.8.5
- virtualenv
- flask
- flask-cors
- yoyo-migrations
- react
- bcryptjs
- axios
- enzyme

7 Document Structure

```
se-03-team-25
    backend
        db_migrations
            createtables.sql
        src
            connection.py
        venv
        connection_test.py
        beer_game.db
        main.py
        requirements.txt
        view_table_data.py
        yoyo.ini
    docs
        backend_documentation
        PREVIOUS_SPRINT_documentation.pdf
        Sprint03_Documentation_Team25
    frontend
        public
            contains all images and videos used in this project
        src
            contains all JavaScript and CSS for App and index
            components
                contains all JavaScript and CSS for component elements
                pages
                    contains all JavaScript and CSS for all pages
                    __test__
                        contains all frontend tests
```

8 Back-End

All the files for the Back-End can be found in the 'backend' directory. Please refer to the Document Structure for help in navigating inside the directories. The files for the Back-End are:

- createtables.sql
- connection.py
- beer_game.db
- connection_test.py
- main.py
- requirements.txt
- view_table_data.py
- yoyo.ini

A detailed overview of the functions used in the backend files 'connection.py', 'connection_test.html', and 'main.py' can be found inside 'docs/backend_documentation' as html files with the original file names (see Document Structure for reference).

9 Front-End

All the files for the Front-End can be found in the 'frontend' directory. Please refer to the Document Structure for help in navigating inside the directories. The files for the Front-End are:

- App.js/css
- Button.js/css
- Footer.js/css
- HeroSection.js/css
- Navbar.js/css
- About.js/css
- animation.js/css
- CardItem.js/css
- creategame.js

- editgame_table.js
- Form.js/css
- game.css
- Home.js
- HomePage.js
- instructor_landing.js
- LearnMore.js/css
- login.js
- MOCK_DATA.json
- player_landing.js
- playergamescreen.css
- playerscreen.js
- SignUp.js
- style.css
- viewgame_table_columns.js
- viewgame_table.css
- viewgame.js

10 Database

The database has been switched from django (from the previous sprint) to a mysqli database using Flask. The database can be checked by running the following inside the backend directory:

```
$ python3 view_table_data.py
```

11 User Manual

Follow the following steps in order to navigate through the user interface.

1. Open a new terminal inside the directory 'se-03-team-25'.
2. Navigate inside the 'frontend' directory.

```
$ cd frontend
```

3. Run the following command to start the frontend.

```
$ npm install
$ npm start
```

4. Open another new terminal inside the directory ’se-03-team-25’.

5. Navigate inside the ’backend’ directory.

```
$ cd backend
```

6. Run the following commands to start the backend.

```
$ pip3 install virtualenv
$ virtualenv venv
$ source venv/bin/activate
$ pip3 install -r requirements.txt
$ yoyo apply
$ python3 main.py
```

7. Open localhost in your browser.

8. Click on the ’Get Started’ button.

9. Register an account as an Instructor.

10. Login as an Instructor.

11. You now have the option to ’Create Game’ or ’View Game’ in the Instructor landing page.

12. Click on the ’Create Game’ card.

13. Create your game by filling the game details and a custom game password (usefull later).

14. After you create a game you will be redirected back into the Instructor landing page.

15. Click on the ’View Game’ card.

16. You should see the game you just created. Copy the GameId.

17. Open a new tab for localhost in your browser.

18. Click on the ’Get Started’ button again.

19. Register an account as a Player.

20. Login as a Player.
21. You now have to enter the GameId you copied in step 16 and the password you set in step 13.
22. You will now see 4 cards for the roles you can play as. Select any role.
23. You will then be redirected to a Game Screen. The Game Screen displays mock data for now.

12 Added Features

- We setup entire the backend using flask as the previous team just had the django server running and did nothing at the backend so we replaced it.
- We setup SQL codes for the database in backenddb_migrations.
- We setup database in SQLITE3 for testing purposes.
- We connected frontend to backend succesfully.
- We added home page to the frontend and also the About Page and Instruction page.
- We added a landing page for player after login where player can enter game credentials given by Instructor and go into the game.
- We added a role choosing page for player inside a game which enables the player to choose a role once they login into the specific game.
- We added a player screen page where the player can view the graphs and also the last weeks data and enter the beer they want to order and check if others have ordered beer or not.
- We added a landing page for instructor after login where the instructor can choose to either create a new game or view their currents games data.
- We added a create game page for instructor where the instructor can create games which is directly stored in the database.
- We added an instructors game viewing page where they can view datas of all their games which have been extracted real time from the database.
- We connected view games page of Instructor to backend to extract real data.
- We added backend testing which checks whether the registration of player, instructor and game is working correctly or not.
- We added frontend testing.

- We used password hash instead of plaintext password for safety in frontend.
- We did user authentication in login page such that only verified Players and Instructors can login.
- We added new documentation called 'DocumentationSprint3.pdf'.
- We updated the Readme such that all the frontend, backend, testing can be easily run following the steps given and also wrote.
- We changed the folder structure of the project to make it more professional than before.

13 Tips for Sprint 04

- Connect the Game Screen with the backend.
- Connect the Edit/Update Game Screen with the backend.
- Start implementing the logic for the actual Beer Game where the Players can order Beers and have to wait for their turns (according to the specification).