

Team Star Project

Final Documentation

Mubarak Ayantayo, Tim Hunt, Josh Adams, and Jack

12/6/2024

1 Introduction

- 1.1 Our client provided us with a solid outline of their needs and expectations for this project, with their being a heavy emphasis on functionality and workability rather than just purely aesthetics. Going from the core points and directions our client provided us, they clearly put an emphasis on player equality in regards to the beginning of the game, as well as heavy emphasis on player and server management. Along with this baseline functionality, they asked for us to put together, they also wanted us to create and add a form of bot to the game to simulate another player, potentially utilizing some form of chat-bot or online artificial intelligence to help streamline the process. Our software solution itself was centered around creating a top down functionality of the code to provide clearly divisible and testable functions to determine individual functionalities and efforts where possible, and to attempt to integrate this clearly testable framework into a UI utilizing firebase as a hosting platform and database. As for the rest of our documentation they can be found under their corresponding sections. The outline of the rest of our methods and solutions can be found in the order and under the sections as follows: Requirements-Functional, Requirements-Nonfunctional, Iterations, Architecture, Testing, and Reflection.

2 Requirements

2.1 User Roles

The types of users for our app can be broken down into three basic types, Users, Guests, and Admins. Two of these User Roles have been further developed and cemented with very specific permissions and allowances. Users and Guests only have permissions to open and join lobbies, whereas admins have permissions to moderate the game and the users to troubleshoot their issues and help keep the game running smoothly.

2.1.1 User: to login and play the game with stat trackers

2.1.2 Guest: to play the game

2.1.3 Admin: to login and moderate other players accounts and games

2.2 Functional Requirement User Stories and Tasks

2.2.1 User

ID: Play BS (CRUD 8 points)

1. Priority: High
2. I want to play BS
3. So that I can play BS with other people
4. Definition of Done: When a game can be played to the natural win conditions
5. Depends On: Other players
6. Tasks:
 - (a) Task ID: 1, Title: Deal Cards, Description: Deal Cards according to the total amount of players in the lobby
 - (b) Task ID: 2, Title: Play cards, Description: allow the players to play cards and progress the game

- (c) Task ID: 3, Title: Call BS, Description: Allow the players to call BS on another player in the game based on who played last
- (d) Task ID: 4, Title: Winners and Losers, Description: create the win condition of having no cards and be declared the winner or loser

ID: Create and Join Lobbies (CRUD 8 points)

1. Priority: High
2. I want to create or join someone elses lobby
3. So that I play with other players
4. Definition of Done: When a player can create or join a lobby
5. Depends On: Number of Lobbies and Players
6. Tasks:
 - (a) Task ID: 5, Title: Lobby Creation, Description: Create an online lobby capable of housing players to play BS
 - (b) Task ID: 6, Title: Join Lobby, Description: allow players to join previously established lobbies

ID: Public and Private Messaging (CRUD 8 points)

1. Priority: High
2. I want to message in a lobby or privately message someone
3. So that I can talk with other users
4. Definition of Done: When users can be messaged both privately and publicly
5. Depends On: Other players
6. Tasks:
 - (a) Task ID: 7, Title: Public messaging, Description: Messages all players contained within a lobby

- (b) Task ID: 8, Title: Private messaging, Description: Privately messages a single player of choice

ID: User Profiles and Login (CRUD 8 points)

1. Priority: High
2. **I want to login and have a username and profile**
3. **So that I can see my statistics and manage how I am viewed**
4. **Definition of Done: When I can login, logout, and see my lifetime statistics for the game**
5. **Depends On: Playing games**
6. Tasks:
 - (a) Task ID: 9, Title: Login, Description: Login to the profile and save user data such as profile name and password
 - (b) Task ID: 10, Title: Logout, Description: Logout of the game and required to log back in to access the profile
 - (c) Task ID: 11, Title: Record and Display all User Lifetime info, Description: Allow the user to see all relevant statistics to what they've done in games

ID: Friends List (CRUD 8 points)

1. Priority: High
2. **I want to add and remove friends from my friends list**
3. **So that I can moderate my friends list**
4. **Definition of Done: When I can view, add, and remove friends from my friends list**
5. **Depends On: Other players**
6. Tasks:
 - (a) Task ID: 12, Title: Add friends, Description: Add friends to the friends list and view them
 - (b) Task ID: 13, Title: Remove Friends, Description: Remove friends from friends list

2.3 Non-Functional Requirements

2.3.1 User Interface

ID: 14 Title: Appearance of the Cards and Table

1. Description: Looks futuristic and has a purple emphasis
2. Affects: How the game and UI is stylized

3 Iterations

3.1 Sprint 1

3.1.1 Plan

Create baseline code from which to work from, primarily centered around game logic, basic database initialization, UI design and planning

3.1.2 Activities

Tim Hunt: Working on the bones of the game logic through the deck, lobby, and player information

Mubarak Ayantayo: Working through friends list and getting the database up and running

Josh: getting the database up and running

Jake:

3.1.3 Retrospective

Tim Hunt: Accomplished creating the groundwork from which to build the game logic and data collection, used JUNIT, only person to have testing

Mubarak Ayantayo: got the friends list working

Josh: got the database up

Jake:

3.2 Sprint 2

3.2.1 Plan

Improve the previously established foundations and try to bring code closer to integration with a newly created UI while looking at multiplayer functionality

3.2.2 Activities

Tim Hunt: Continue progress on the game logic, further developing towards a playable gamestate and refactoring some game logic to be more top down, alongside extensive testing

Mubarak Ayantayo: Refactoring, database rules, and firestore collections creation to make friends functionality working. Completed sending of friend requests and created the first workable form of UI and the associated login functions as well as the login page information surrounding it. Acclimation to testing frameworks and tools.

Josh: Working on Database

Jake: Looking at Hosting stuff

3.2.3 Retrospective

Tim Hunt: Made substantial progress in refactoring and had fully tested and functional code

Mubarak Ayantayo: Created the first form of the user interface and created some login functions along with it from older code. Did major refactoring and decent progress to testing. first instance of multi player interaction created.

Josh: Working on chat

Jake:

3.3 Sprint 3

3.3.1 Plan

Finish all game logic, integrate with UI, resolve lack of testing not associated with game logic

3.3.2 Activities

Tim Hunt: Complete all game logic and testing, create a playable game-state and a working game with extensive testing

Mubarak Ayantayo: Working on integrating the UI with the working game-state. Made: viewing friend requests, accepting friends, removal of friends, messaging friends, messaging public. creating lobbies. Made progress joining lobbies, improved the database user fields

Josh: Worked on messages within this sprint. Got closer to getting friends working while attempting to refactor a bit

Jake: ?

3.3.3 Retrospective

Tim Hunt: Fully completed game logic and testing, created a fully functional game which can be played to completion with no outside assistance

Mubarak Ayantayo: Was injured severely and unable to continue work on pace, but still made substantial strides toward goals. Completed josh and jake tasks and still working on integrating the UI with the game logic in Java

Josh: Refactoring could have been a bit better when coming to my stories but ran out of time

Jake: ?

4 Architecture

The architecture layer from UI to Java is structured to go through HTTP code calling the java code from Firebase, and from the java code side of it it is made from a top down approach focused on having small scale functions being integrated into larger functions letting problems be more easily diagnosed and moderated. The most import and only testing we had was using JUnit on the Java code.

5 Testing

Testing was done exclusively in java code by Tim Hunt, all of which was done utilizing Junit which proved invaluable for testing functions. Testing was used to test every single function of every single class, thoroughly ensuring

it was functional. Testing consistently and fully went extremely well. The problem is there is no testing done outside the java side of the code.

6 Reflection

We would not have used firebase at all, we would have used a servlet and have designated clear roles for people who struggled to contribute. The main barriers were integrating completed code with the UI and associated older code. The main facilitators were Tim Hunt and Mubarak Ayantayo. The team was attempted to be organized, but elements within the group prevented this. Tim Hunt was the designated leader, who handled most coordination efforts as well as all documentation, testing, game logic, and most functional code. There was also an attempt by him to keep group members focused and on task. The team was not well organized. A summary of what we have learned about the SE process is that no amount of effort of a single member can make up for a groups worth of work, as well as when designing applications, comprehensive evaluation and planning for major design elements is absolutely critical to the success of an application.

7 AI Impact

Tim Hunt: did not use AI as it would slow down and convolute the code and weaken the understanding of the code

Mubarak Ayantayo: Used AI for organization, arrangement, suggestions. manually debugged AI errors, fixed interdependency and accessing the right files to import. AI was a very helpful tool.

Josh: Used AI exclusively for debugging but that's it.

Jake: