



TriviaCast

Sprint 1 Planning Document

CS 30700, Team 15

Kyle Beck, Jimmy Carlson, Ronan Coyne, Ram Dent, Tan Lin, Kathleen Masterson

Sprint Overview	2
Current Sprint Detail	2
User Stories to be Implemented	2
User Story 1	2
User Story 2	3
User Story 3	3
User Story 4	4
User Story 5	5
User Story 7	6
User Story 8	7
User Story 9	7
Backlog	8
Functional	8
Non-functional	9

Sprint Overview

1. This initial sprint will aim to produce a basic working model of our application. We plan to create the basis for our game, that is: login and main menu interface, a small question bank for basic gameplay as well as initial question and answer displays. The performance and aesthetics are not a priority in this first sprint as we are simply focusing on producing a starting model to further develop in future sprints.
2. Scrum Master: **Kathleen Masterson**
3. Scrum Meeting Schedule: Tues, Thurs 3pm - 4:30pm (after CS 30700 lecture)
4. Challenges:
 - a. Keeping both iOS and Android applications on the same page. Since our project will result in an application for each platform, it is crucial that our development of one version remains synchronised with development on the other platform.
 - b. Working cohesively as a team. Having not worked together before, it may take us longer than expected to complete certain tasks as we adapt to different working styles and patterns. Similarly, splitting tasks as such is required by the class means that not everyone will be familiar with all aspects of the project; ensuring a cohesive design and resulting product will be challenging, as will debugging software that we've only written parts of.
 - c. Working with new technologies. As a team we have varied experience with the tools we will be using during development. We plan in this first sprint to invest some time getting up to speed with technologies such as MongoDB, iOS and Android app development using react native, and UI design in order to ensure improved efficiency and productivity in future sprints.

Current Sprint Detail

User Stories to be Implemented

- User Story 1

As a user, I want to be able to connect to my Chromecast.

Task	Workload Estimate	Developer
Implement Chromecast APK into iOS/Android	6 hours	Kyle Beck
Add UI to start the sync to the Chromecast in iOS/Android	2 hours	Kyle Beck
Create basic UI screen for Chromecast	2 hours	Kyle Beck

Acceptance criteria:

- i. Given that the Chromecast sync is implemented correctly, when a user starts the game and selects the Chromecast, the game will automatically display all information on the screen.
- ii. Given that the Chromecast APK is implemented correctly, the user will be able to choose from a list of Chromecast devices to stream to.

Workload estimate: 10 hours

- User Story 2

As a user, I want to be able to create and join game sessions

Task	Workload Estimate	Developer
Create TCP socket server to allow users to get real time game updates	8 hours	Jimmy Carlson
Create service to add/remove users from a game session	4 hours	Jimmy Carlson
Create a game manager class to control	4 hours	Jimmy Carlson
Update player stats, and game leaderboards at the end of each game.	2 hours	Jimmy Carlson

Acceptance Criteria:

i. Given that the socket server is implemented correctly, the user should be able to receiving live updates of the game.

ii. Given that the game service is created successfully, the user should be able to make a GET request that will respond with a port, and game ID to start listening to for updates.

iii. Given that the game manager class is created successfully, the DB will be updated accordingly, and games will be managed and send updates to ports.

Total Workload Estimate: 13 hours

- User Story 3

As a developer, I want to create a question user interface in React.

A general screen to display the question grabbed from the server. Will show answers based on type of question (Multiple Choice or short answer).

Task	Workload Estimate	Developer
Create segue from main menu to the game interface	1 hour	Kyle Beck
Create reusable UI to show each question and types of answers (MC/short answer)	4 hours	Kyle Beck
Create UI for showing the answer and each player's response	4 hours	Kyle Beck
Create UI elements showing cumulative users scores and game progress	4 hours	Ram Dent
Create UI element to display timer for each question	3 hours	Ronan Coyne

Acceptance Criteria:

- Given that the interface is implemented correctly, players should be able to select an answer and submit their response.
- Given that the UI elements are implemented correctly, the user should see the correct answer and game progress for each user after the timer for the question has passed.

Total Workload Estimate: 16 hours

- User Story 4

As a developer, I want to create a user login interface in React.

- The default launch screen; users need to register an account before using the app. Credentials should be stored and this screen should not appear again most cases.

Task	Workload Estimate	Developer
Create basic login UI	3 hours	Ronan Coyne
Display text fields for username and password	2 hours	Ronan Coyne
Validate existing credentials using database	2 hours	Ram Dent
Display button to register for new users	2 hours	Ronan Coyne
Build text fields to gather new credentials	4 hours	Ronan Coyne
Validate new password and email address	3 hours	Ronan Coyne
Implement forgotten password feature	4 hours	Ram Dent
Ensure login is retained after app quit/relaunch	4 hours	Ram Dent

Acceptance Criteria:

- Given that the login interface is implemented correctly, when a user enters valid credentials the main menu will be displayed.
- Given that the login validation is implemented correctly, when a user enters invalid credentials an error message will be displayed and the login interface will continue to be displayed.
- Given that the sign up interface is implemented correctly, when a user enters valid credentials the data will be written to the user database and the main menu will be displayed.
- Given that the sign up validation is implemented correctly, when a user enters invalid password or email address an error message will display and the user can try again.
- Given that the forgotten password tool is implemented correctly, when a user clicks the button they will be prompted to enter email address and a password reminder will be generated.

Total workload estimate: 24 hours

- User Story 5

As a developer, I want to create a main menu interface.

Task	Workload Estimate	Developer
Create a new user interface with React Native	1 hour	Kathleen Masterson
Enhance the user interface with all menu options	3 hours	Kathleen Masterson
Connect all menu options to functions that switch to the corresponding views for each option in the menu	4 hours	Kathleen Masterson
Ensure interface is scalable for different types of devices.	2 hours	Tan Lin
Ensure all buttons and aspects of the interface are functioning with no errors	1 hour	Tan Lin

Acceptance criteria:

- Given that this is the interface for both the iOS and Android devices, React Native will be used to create the interfaces.
- Given that users should only have to login a minimum number of times this will be the main page that opens when they launch the application.
- Given that this will be the main page for navigation users should be able to reach every view that is needed from this page.
- Given this will be one of the first impressions of the application, this should be designed in the overall theme for the application.

Total workload estimate: 12 hours

- User Story 7

As a developer, I want to create a database of potential questions.

Task	Workload Estimate	Developer
Create a new MongoDB instance	1 hour	Kathleen Masterson
Add a small bank of questions in both true/false and multiple choice format	3 hours	Kathleen Masterson
Ensure each question is given a category	2 hours	Kathleen Masterson
Ensure each question is given a difficulty ranking	3 hours	Tan Lin

Acceptance criteria:

- i. Given MongoDB's flexibility in storing records we will use that when creating a database of questions.
- ii. Given users will be given the option to customize the types of questions they would like to answers, all questions will need to be assigned to a category and difficulty ranking.

Workload estimate: 9 hours

- User Story 8

As a user, I want to be able to manage my account settings and view my statistics.

Task	Workload Estimate	Developer
Write wrapper to connect to mongoDB	2 hours	Jimmy Carlson
Add encryption methods to keep user passwords secure, before storing into mongoDB	4 hours	Jimmy Carlson
Create REST services for user management methods	4 hours	Kathleen Masterson
Create service to update new user scores whenever a game is finished.	2 hours	Jimmy Carlson

Acceptance Criteria:

- i. Given that the wrapper is created successfully, data should be successfully stored in the mongoDB instance.
- ii. Given that the encryption methods are successful, inside the database you should not be able to see everyone's password
- iii. Given that the REST services are implemented, there will be API calls to register a user successfully, and respond with a token verifying the user. Acceptance criteria:

Total Workload Estimate: 10 hours

- User Story 9

As a user, I want to be able to choose my avatar.

Allows for users to choose a picture associated with their account, and be their representation in game. Will likely need a mechanism to store avatars on the server, resize and compress to optimize space, etc.

Task	Workload Estimate	Developer
Add option to add avatar in account settings	2 hours	Tan Lin
Create uploading/resizing/etc interface	3 hours	Ram Dent
Create and host database of avatars	5 hours	Tan Lin

Acceptance criteria:

- i. Given that users should have full control over their game, the avatar would allow them to continue that.
- ii. Given that users will keep their game settings and statistics the same each time the application is being used the avatars need to be saved in a database.

Total Workload Estimate: 10 hours

Product Backlog

Functional

- As a developer, I want to create a database of potential questions.
- As a developer, I want to create a database for player statistics.
- As a developer, I want to create a question user interface.
- As a developer, I want to create a main menu interface.
- As a developer, I want to be able to easily add questions to the database.
- As a developer, I want solid exception handling.
- As a developer, I want to manage game states on the server.
- As a developer, I want the game state to reset to default each time the application is closed.
- As a user, I want to be able to create a game with other users.
- As a user, I want to be able to connect to my Chromecast.
- As a user, I want to be able to save my game.
- As a user, I want to be able to view my high scores.
- As a user, I want to be able to record my score history.
- As a user, I would like to chat with other users.
- As a user, I want to be able to quit my game.
- As a user, I want to be able to choose what types of questions I answer.
- As a user, I want the ability to hear music in the background.
- As a user, I want useful sound effects during gameplay.
- As a user, I want an easy navigation method.
- As a user, I want to compare my scores with others.
- As a user, I want to choose topics for my quiz game.
- As a user, I want to be able to have a menu to edit certain settings for game modes. (Time per question, etc.)
- As a user, I want to choose how many questions will appear in my quiz game.
- As a user, I want to be able to login to save my statistics. (If time allows)
- As a user, I want the option of different types of games. (If time allows)
- As a user, I want to choose question sets of varying difficulty.
- As a user, I want to be able to change my login/username. (If time allows)
- As a user, I want to be able to view high scores/statistics of each game mode.
- As a user, I want to be able to rejoin games if disconnected. (If time allows)
- As a user, I want to view a leaderboard comparing me with other players.



Non-functional

- As a developer, I want answer schemes to be securely hidden from users.
- As a developer, I want to make system scalable (10-20 users).
- As a developer, I want to have a backup server in order to maintain server downtime.
- As a developer, I want to be able to authenticate users.
- As a developer, I want to write modular code that is easily reusable.
- As a user, I want the game to have a fast response time with little to no lag.

