

CS242 Final Project Rubric (ykim164)

Week 1 – Client User Interface

In the first week of the final project, I will be developing the client-side user interface of the application where users can see the scores, match-ups, and league standings. I will also try to connect the interface with the Firebase database so that it displays/updates the data on the database. For this week, since I am not going to implement the admin-side, I will try to test these by manually updating the DB.

Requirement 1: Create Client User Interface

- Starting screen with the app logo
- Mode selection screen (admin or regular user)
- "Games" page with dummy game data
- Using TabLayout and NavigaionLayout together:
 - TabBar for other menus including:
 - "Live Games" : Shows only the live games
 - "Favorite Games" : Shows only the "starred" games
 - "League Standings" : Shows league table and scorer table.
 - NavigationLayout for other menus including:
 - "Select Channel" : Users can change channel
 - "Team List" : Shows the list of participating teams
 - "League Information" : Shows the basic information of the tournament/league

Requirement 2: Responsive Design

- Create a responsive design for landscape view
- Create a responsive design for X-Large (tablet) view
- Create a responsive design for X-Large Landscape view.

Requirement 3 : Database connection

- Set up a real-time database that stores all information of the tournament
- Connect all relevant data to the views
 - Game data → Game ListView
 - Channel data → Channel Selection
 - Team Data → Team Information Screen
 - Posting "Starred" Status: Pressing the "Star" imageView updates the "followed" status of the game.
- Responsive view
 - All views should update immediately when data change in the real-time database. This is for the live score update function that I will be implementing in the admin-side.
 - For this week, since I am not going to implement the admin-side, I will try to test these by manually updating the DB.

Requirement 4: Testing

- Create a manual test plan including the test of client-side UI

Grading

Category	Weight	Scoring	Requirements
Basic Preparation	2	0-1	Ready to go at the start of section
Cleverness	2	0-2	The hardest points on the rubric
Code Submission	4	0-2	Submitted correct content on time and to the correct location in the repository
Decomposition	4	0-2	Project is adequately decomposed to different classes and methods
Documentation	4	0-2	Comments for each class and each function are clear and are following style guides
Effort	2	0-2	Perform considerable amount of work

Category	Weight	Scoring	Requirements
Naming	2	0-2	Variable names and method names are readable and are following the naming conventions of the language you chose
Overall Design	5	0-2.5	Have nice approaches and structures in overall
Participation	5	0-2.5	Interact with the group 2 times (ask a question, make a comment, help answer a question, etc.)
Presentation	4	0-2	Present the code clearly
Requirements - Interface Design	5	0-2.5	2 points: Basic UI's for all the views have been defined. 2.5 points: Able to recycle views properly and have an application with tabs + navigation drawer for quicker access to different views.
Requirements - Database Connection	5	0-2.5	2 points: Set up a real-time database and connect it to its corresponding views. 2.5 points: Make the views responsive to database updates.
Requirement - Adaptive	4	0-2	1 point: Use some styling to make the application look clear and presentable. Use a nontrivial amount of styling 2 points: Modify the styling or theme to make those fit the different categories of pages, and make the application look nice, beautiful and adaptive
Testing	4	0-2	0 points: No test plan 2 points: Manual test plan
Total	52		

Week 2 – Admin User Interface & Mechanisms

In the second week of the final project, I will be developing the admin-side user interface of the application where the administrator can create and manage a competition. The admin will be able to set up the details of the tournament, set up the game schedule, and most importantly, start a live game and update scores of a live game.

Requirement 1: Create Admin User Interface

- Log-in & Registration page
- Tournament set-up page
- Game set-up page
- Game Management Page
- Buttons for easily updating game status
 - Start Game (Changing game status to live)
 - Update Score (Update score & add scorer)
 - End Game

Requirement 2: Functionality for starting/updating live games

- Mechanism for live games
 - Automatically updating game time (0 min ~ 90 min)
 - Half time / Full Time
 - Animation Effect for Live Status Icon

Requirement 3: League Table/Score Table

- Make a league table that is automatically sorted based on the rankings.
- Make a score table that is automatically sorted based on number of goals scored.
- Make them update/sort the information whenever the data changes in the DB.

Requirement 4: Testing

- Manual test plan for admin user interface
- Unit test for league table/score table logic