

CS242 Final Project Rubric (ykim164)

Week 4 – Finalize the Product

In the last week of the final project, I will be finalizing my app design and fixing some remaining bugs. Although this is a class project, if I assume that this is a real product, this week would be the most important week because it is the week before publication of the product. I would focus especially on the polishing user interface design and test the product as if I were a real user of the product. Some additional features that I will add include app sharing feature (just a message for now) and displaying the actual game times.

- **Requirement 1: Finalize design and make it fully adaptable for all conditions**
 - Polish the design of the app to make it pretty (use better fonts, change colors, fix ugly layouts, set background images).
 - Change all layouts to LinearLayout so that it is perfectly responsive to all kinds of sizes.
 - Minor debugs
 - ◆ Time conflict handling when creating tournaments or games (check if the tournament term is valid)
 - ◆ Ask twice when the user tries to delete an item.
 - ◆ Prevent making invalid actions:
 - Ending the game twice
 - Updating the score after the game is over
 - Updating the score of a game that is not LIVE
- **Requirement 2: Provide details on list item clicks**
 - Users should be able to view detailed information of items of all list views when clicked.
 - ◆ Team information (game statistics) on League Standings.

- ◆ Scorer information (when did the player score) on Score Table.
 - ◆ Updates the current time of a game when clicked
- **Requirement 3: Live Game mechanism**
 - Implement a way to display exceeded time for each game (make it both pause-able and resume-able).
 - Capture the current game time in minute in the client side
 - **Requirement 4: Share Application**
 - Make an option for sharing the app information (text would be fine for now) through email, messenger, Facebook, etc.
 - **Requirement 5: Testing**
 - Manual test plan for time conflict and delete confirmation
 - Unit Test for date handling

Grading

Category	Weight	Scoring	Requirements
Basic Preparation	2	0-1	Ready to go at the start of section
Cleverness	4	0-2	The hardest points on the rubric
Rubric	2	0-1	
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

Category	Weight	Scoring	Requirements
Effort	2	0-2	Perform considerable amount of work
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 – Improve design and fix remaining bugs	5	0-2.5	2 points: Make the design more pretty and adaptive. 2.5 points: Fix remaining bugs such as time conflicts and invalid actions (ending game twice and updating score after the game is over, etc.)
Requirements - Provide details for ListView items	5	0-2.5	2 points: Provide team stats on league table item click. Display detailed game stats. 2.5 points: Display goal records on score table item click. Include the team that the scorer scored on.
Requirement – Live game mechanism	5	0-2.5	2 point: Make a clear indicator for live games in the client side. 2.5 points: Put the current time (exceeded time) of live games on the client side.
Requirement – App Sharing	5	0-2.5	2 point: Options for sharing app information.

Category	Weight	Scoring	Requirements
			2.5 points: Sharing works for multiple media including text-message, email, messengers, etc.
Testing	5	0-2.5	2: points: Manual Test Plan 2.5 points: Tests for date conflict and time difference.
Total	63		