

Purdue Course Ratings

Team 17: Elvin Uthuppan, Kenny Zheng, Manasi Goel, Kendall Z Bowles

Work done so far in sprint 2

In sprint 2 we have focused more on backend functionality and how different modules are connected. The following use cases have been completed:

1. As a user, I would like to view a course's individual reviews.
2. As a user, I would like to view an instructor's individual reviews.
3. As a user, I would like to search for a course or instructor.
4. As a user, I would like to view statistics for a course based on reviews.
5. As a user, I would like to view statistics for an instructor based on reviews.

Note: Throughout this document, we will use a severity of 1 as the lowest severity. As the numbers increase, so does the severity, up to 5.

Design Inspection Defects

Product	Purdue Rating App Design Inspection		
Date	03/03/2017		
Author	Manasi Goel		
Moderator	Elvin Uthuppan		
Inspector	Kenny Zheng		
Defect #	Description	Severity	How Corrected
1	The listview showing rows of reviews wasn't scrollable when too many objects were in view.	2	Was able to disable the scrollView whenever the ListView was being scrolled through.
2	When the user searched for a course/instructor that did not exist in our database, nothing would happen.	2	We added an error message that says that the searched for item does not exist and we added buttons that only become visible in this scenario and let users write the first review.
3	When a user viewed the course/instructor reviews of a specific course/instructor, there was no easy way to add a review from there.	3	We added a button that takes a user to the add review view with the course/instructor name already typed in for them.
4	The screen with all the information about a course/instructor was becoming very crowded and it was hard to fit in everything and read it all.	4	We created a new page for reviews. This way a person will first see all the statistics, which is the focus of our reviewing system, and then click on a button to see all the written reviews.
5	Our error message and buttons to add a new course/instructor were always visible in search. They should only have been visible when the item searched for did not	2	We added several conditions and checks and only displayed them when it was needed and hid them when it wasn't.

	exist in the database.		
6	Error message for when successfully searching for a professor/course was inconsistent and often display wrong message	1	Add several checks and condition to prevent this issue
7	When adding a new review the numbers shown representing the value of the SeekBar will have delays and often will not reflect the actual value.	2	Assigning each SeekBar with different TextView which correspond to different value changes has allowed the numbers to be updated with the actual value right away.
8	When adding a new review, users are unable to scroll through their text in the comments box when the comments are too long.	1	We intend on adding a scroll view to the comment area which would allow users to scroll within the comments section and write longer comments.
9	When a user added a review from the page where they could see the reviews of a course/instructor, the add a review page would be completely blank. Instead it should have opened with the course/instructor name already filled out.	2	We added the instructor/course name when the user tries to add a review from a page that displays reviews. We also added a few checks so that this would not cause problems for when someone wanted to add a review for a non-existing course/instructor.

Code Inspection Defects

Product	Purdue Rating App Inspection		
Date	03/03/2017		
Author	Manasi Goel		
Moderator	Elvin Uthuppan		
Inspector	Kenny Zheng		
Defect #	Description	Severity	How Corrected
1	A typo was preventing our code from accessing the instructor section of our database.	4	We simply changed from "message/review/instruct ore" to "message/review/instruct or".
2	An incorrect object type was being used when setting the variables used for instructor reviews.	4	We simply changed the object type from CourseReview to InstructorReview, which caused the correct data to be set.
3	Three attributes set when submitted a review were inadvertently always set as false.	2	We fixed this by correctly setting them to being true when they should be.
4	A reference to an ImageView object that we had previously deleted from the project was causing null pointer errors.	3	We took out all leftover references to this object.
5	The search says a course/instructor does not exist in the database when it does. However, if a user clicks on the search button again, it works.	4	This issue is because of latency in Firebase requests. We intend on using threads to wait for Firebase to finish before displaying anything to users.
6	The test for the buttons to add a	2	We needed to switch the

	course/instructor review when the search does not return any results would fail.		rule from ActionIntent to Intent at the top of our search java file.
7	Title of a review and other stats stay on top which blocked a user from reaching the reviews in the listView	3	Made the Scrollview primary but disabled when scrolling through the listview.
8	We initially had global variables to pass information between different screens but this caused the app to crash in different scenarios and other issues because it was not what Android typically does.	3	We replaced the global variables with information passing between intents when we start a new activity.
9	When user enters an input, the program will search an ArrayList that contains all the course name and professor name to see if the input is valid or not, however, the result will always be false even if we know the input is valid.	4	We figured out that whenever we pass the ArrayList from CourseDisplay to Search page, the ArrayList gets reset back to empty. Instead of getting the ArrayList from CourseDisplay page, we passed the user's input to CourseDisplay page for it to be checked if the input is valid or not
10	When a user enters an instructor's name, "Buster Dunsmore" and "buster dunsmore" would be stored as two different people.	2	We converted every instructor name to be upper case. We also added test cases to evaluate this.

Unit Testing Defects

We automated our unit testing by using Espresso, which helps automate unit testing for Android Studio. We created rules for different activities and intents and then wrote tests for various UI components using those rules. This automation can be seen in the following folder:

[“CS_408_Project/app/src/androidTest/java/com/mango/cs_408_project/”](#)

We’ve tested everything from UI features to our backend functionality.

Product	Purdue Rating App Unit Testing			
Date	03/03/2017			
Author	Manasi Goel			
Moderator	Elvin Uthuppan			
Inspector	Kenny Zheng			
Defect #	Module	Description	Severe	How Corrected
1	Viewing course reviews	Course reviews were not showing up on the listView	4	Had to make a separate xml file for the items in a single review, and connect that to the listView. Made it work with a custom arrayadapter which took in review objects and put them into an arrayList. Was able to set each review for each object in the arrayList. We wrote unit tests to check if we could view the reviews in the listView and add a review from there.
2	Viewing instructor reviews	Instructor reviews were not showing up in the listView	4	Similar with the course reviews, but had to add different fields from the course reviews. We wrote unit tests to check if we could view the reviews in the listView and add a review from there.

3	Searching for reviews	The search had trouble with different cases. For example if someone searched for a course with lower case which only existed in uppercase, the search results would be empty.	2	We converted everything to uppercase behind the scenes. This allowed uniformity in the code no matter what the user types in. We also added tests for every possible combination of letter cases.
4	Viewing statistics for course reviews	There was some trouble calculating various statistics for each course related item. We were calculating the wrong numbers given our data for certain parameters.	3	We added a few simple examples to the database which we could calculate the statistics for by hand. We also added tests that check that the calculations are what we expect for certain test courses.
5	Viewing statistics for instructor reviews	The progress bar showing toughness was incorrectly displayed because we had toughness on a scale out of 5 which is not what is expected of Android Studio.	2	We did some research and found that we can set the range for a progress bar and did exactly that. We also added test cases to make sure that our toughness bar is on a scale of 1 to 5.