Purdue Course Ratings

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

Use cases worked on so far in sprint 1:

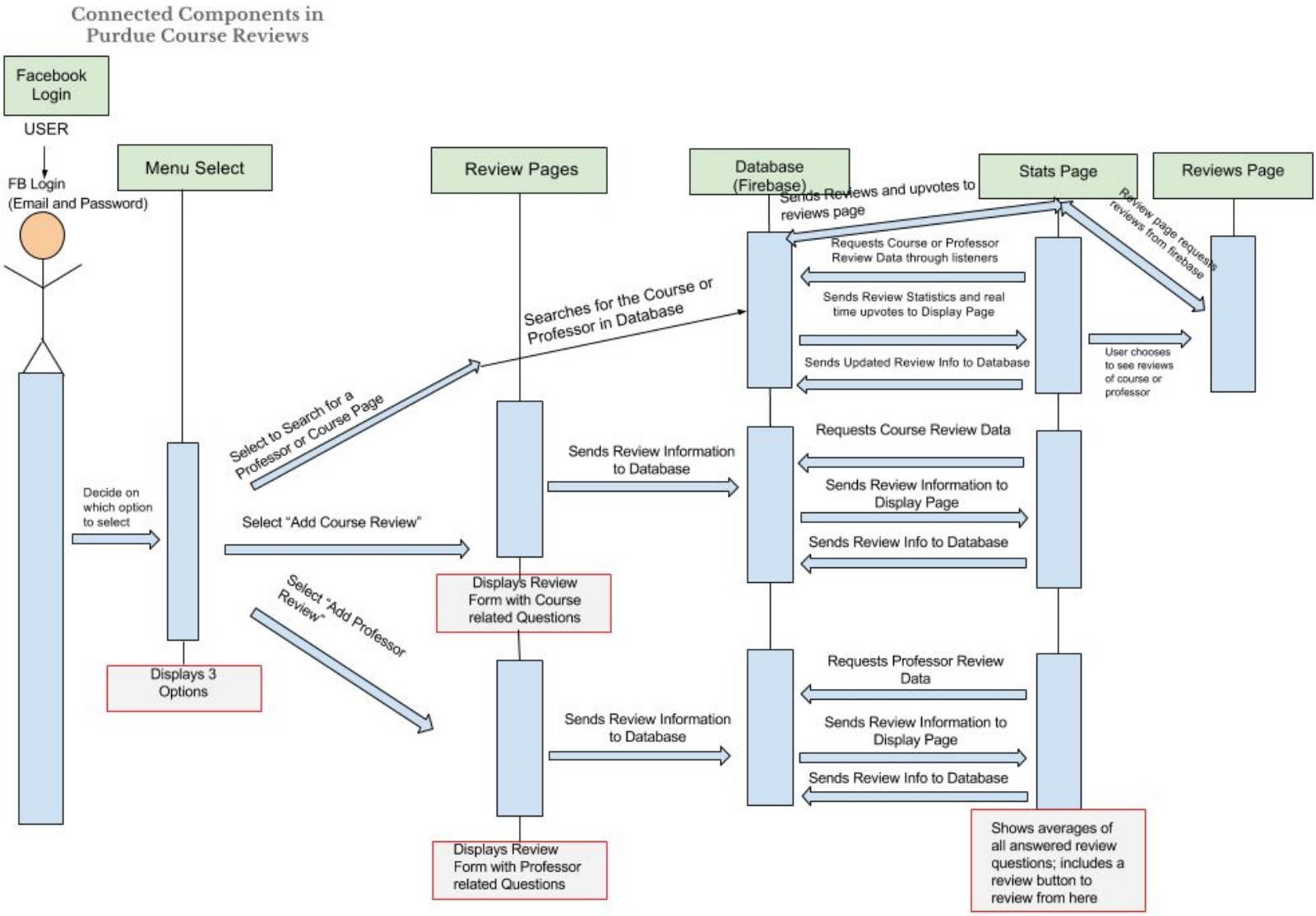
In sprint 2 we have focused more on backend functionality and how different modules are connected. The following use cases have been completed:

1. As a user, I would like to view a course's reviews.
2. As a user, I would like to view an instructor's 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.
6. As a user, I would like my searches to be auto-completed.
7. As a user, I would like to sort reviews based on recency, rating and helpfulness.
8. As a user, I would like to vote on the helpfulness of reviews.

Note: Throughout this document, we will use a severity of 1 as the lowest severity. As the numbers increase, so does the severity. The highest severity we have is 5.

Classification of Components

Our components are database (Firebase), facebook login, writing an instructor/course review, selecting the type of review from login, search for a course or instructor, displaying an instructor/course statistics, displaying an instructor/course reviews, voting on helpfulness of reviews, and sorting reviews. Their interactions, inputs, and outputs are outlined in the following diagram:



We are using top down incremental testing. We have completed more frontend work than backend work and so we can more easily work top down. We have created a lot more fields where data from our database should place the averages from reviews. We could substitute unavailable components with stubs.

Incremental and Regression Testing

We automated testing by using Espresso, which helps automate unit testing for Android Studio. As we added more modules, we tested if going through them was smooth and without error. We created rules for different activities/pages and intents and then wrote tests for various UI components to make sure they are functioning properly altogether as we add more tests for regression testing. This automation can be seen in the folder: "CS_408_Project/app/src/androidTest/java/com/mango/cs_408_project/"

Module	Voting on helpfulness of reviews		
Incremental Testing			
Defect No.	**Description**	**Severity**	**How to correct**
1	Adding or subtracting from the database likesCount made the page go back to the CourseDisplay.	2	Make the activities a singleInstance at launch to prevent them from making more pages when changing the database.
2	Using the like button replaced the current review object with a separate one in the database.	5	Had to specify the child's firebase key. To do that, we had to put the key as a child of each review so that it could be accessed later and edit that particular review rather than replace a review.
Regression Testing			
Defect No.	**Description**	**Severity**	**How to correct**
1	After using the like button, there would be a ton of pages to go back through to get back to the Search Bar.	3	The listeners in the Search activity were noticing every time the data changed in the database, so changing their type of listener stopped this behavior.
2	A user could press the like button for a single review multiple times. Hitting the like button could be abused and mess up tests that expected button to upvote and then reset.	2	Fixed this by grabbing the facebook login token and placing a List of tokens with each review object that can be checked to see if a user has already reviewed.

Module	View Statistics of Professors and Courses		
Incremental Testing			
Defect No.	**Description**	**Severity**	**How to correct**
1	The review list would get in the way of the stats since you could only scroll through the list and not the page.	3	We placed the review list with individual comments into its own separate page/activity. Then we disabled the scrollview if we were scrolling through the list view.
2	Adding a review from that page could lead to a user creating a new professor or course.	2	Placed the name as presented in the database already into the enter course or professor name field.
Regression Testing			
Defect No.	**Description**	**Severity**	**How to correct**
1	A user could only see the reviews and not the statistics page after having searched for the instructor/course.	2	We changed the flow of events so that after searching the user would be taken to the statistics page and then could click a button to see the written reviews.
2	The calculation of statistics failed if the user did not answer every single question when writing a review.	3	We checked that a user had answered all questions before adding a review to our database. We also added tests to make sure a review that does not have every question answered does not get accepted.

Module	View Reviews of Professors and Courses		
Incremental Testing			
Defect No.	**Description**	**Severity**	**How to correct**
1	Adding a review from that page could lead to a user creating a new professor or course.	2	Placed the name as presented in the database already into the enter course or professor name field.
2	Adding a review for an existing course/instructor caused both courseDisplay and instructorDisplay activities to show more reviews than there really were. This was because the data was being written into as strings and they would not go inside each other.	4	Set up the reviews to be pushed into the database as objects so that the object's fields were shown elegantly through firebase's database. Reviews were then showing under each respective course or professor name.
Regression Testing			
Defect No.	**Description**	**Severity**	**How to correct**
1	Found that adding reviews to this page through the Course and Instructor Review Display activities were disrupting the likes count values.	3	Had to make separate listeners for the likes and the Review Display pages to not interfere with each other. The likes would listen to when a child was changed while the review display would listen to when the page was visited by a user.

Module	Autocomplete Searches		
Incremental Testing			
Defect No.	**Description**	**Severity**	**How to correct**
1	Search bar was only showing instructor and course as the words on the list.	3	Had to add to an ArrayList of Strings by going through the child of the instructor reviews and the child of course reviews separately, and then they would show up on the search bar.
Regression Testing			
Defect No.	**Description**	**Severity**	**How to correct**
1	After using the like button, the autocomplete in the search bar would show up 6x the amount of reviews the review object had.	3	Had to set listeners in the Search activity to single value event listeners so they wouldn't go off while editing the database in realtime.
2	Tests for autofill was working on the emulator but not phone, and we found the problem to be we used pressButton function to test our code and pressButton is not available for phone.	4	Change all pressButton function so instead of choosing correct autofill by press of button, it now chooses correct autofill by clicking the correct choice

Module	Search for a course or professor review page		
Incremental Testing			
Defect No.	**Description**	**Severity**	**How to correct**
1	Searching for professor or course name containing capitalization would yield different result if searching for names with no capitalization	2	Convert all search inputs to capitalization and use new inputs to search our database
2	Search for valid professor or course name would sometimes not success first try because the lag in Firebase	3	When searching for a professor or course name, click “submit search” twice with Thresh.sleep() in between to make sure no database lag
Regression Testing			
Defect No.	**Description**	**Severity**	**How to correct**
1	Tests for viewing review page would fail for some version of emulator because the button to see reviews are off the screen for smaller phone	1	Instead of clicking the “view review” button, we perform scrollTo and then click the button

Module	Sorting Reviews		
Incremental Testing			
Defect No.	**Description**	**Severity**	**How to correct**
1	We were not storing the time at which a user had written each review, which made sorting reviews by recency hard.	3	We made sure that the order of reviews after they were written into the database was not changed. Therefore, the database will always contain entries from oldest to newest and we could use that.
2	The application crashed if no sort was selected by default, when a user first opens the reviews page.	5	We set the default order to be oldest to newest, which would be the fastest. The user can then change this.
Regression Testing			
Defect No.	**Description**	**Severity**	**How to correct**
1	If the user did not add a rating when writing a review, it would not be possible to sort by rating because there wasn't one.	3	We made it mandatory for users to add a rating when writing reviews. We also added tests to check this.
2	A user could write several reviews and since sorting is slower if there are more elements to sort, this could really slow down sort.	2	We used a facebook login token and placed a List of tokens with each review object. This can then be checked to make sure only one review can be written per user.

Updated Product Backlog

The updated list of requirements can be found in the updated backlog in our repository. The following is a copy of the table. Note: One of our group members is no longer taking this class so we have changed the status for some requirements to “if time permits”. It should also be noted that we had certain features in mind in the beginning of the semester which would no longer fit into our project because of how it has been implemented.

ID	Functional Requirement	Hours	Status
1	As a user, I would like to login through Facebook.	5	Completed in sprint 1
2	As a user, I would like to search for a course/professor/TA (through information pulled from the Purdue directory).	20	Completed in sprint 2
3	As a user, I would like to filter my search further after seeing the initial results using different keywords .	5	No longer applicable
4	As a user I would like to view statistics (averages of all students for different criterias) and reviews of a professor/course/TA.	20	Completed in sprint 2
5	As a user, I would like to write a review	10	Completed in sprint 1
6	As a user, I would like to upvote or downvote a review as helpful.	10	Completed in sprint 2
7	As a user, I would like to comment on reviews.	10	Planned for spring break
8	As a user, I would like to sort the list of reviews based on recency, how helpful they are and rating.	10	Completed in sprint 2
9	As a user, I would like to sort search results by ratings.	5	No longer applicable
10	As a user, I would like to edit/delete my own reviews or ratings.	10	Planned for spring break

11	As a user, I would like to answer yes or no questions related to the professor/course/TA (e.g. is the professor easy to understand, do you need a textbook, etc.).	5	Completed in sprint 1
12	As a user I would like my searches to be autocompleted.	2	Completed in sprint 2
13	As a user, I would like to view different courses a professor teaches.	5	If time permits
14	As a user, I would like to share a review on Facebook.	5	If time permits
15	As a user, I would like to see who the highest rated professor for a specific course is.	5	If time permits