Ralphie's Tutor
Project Part 4 - Final Report
Ruhi Saraf, Samarpita Debnath, Santhanakrishnan Ramani

1. What features were implemented?

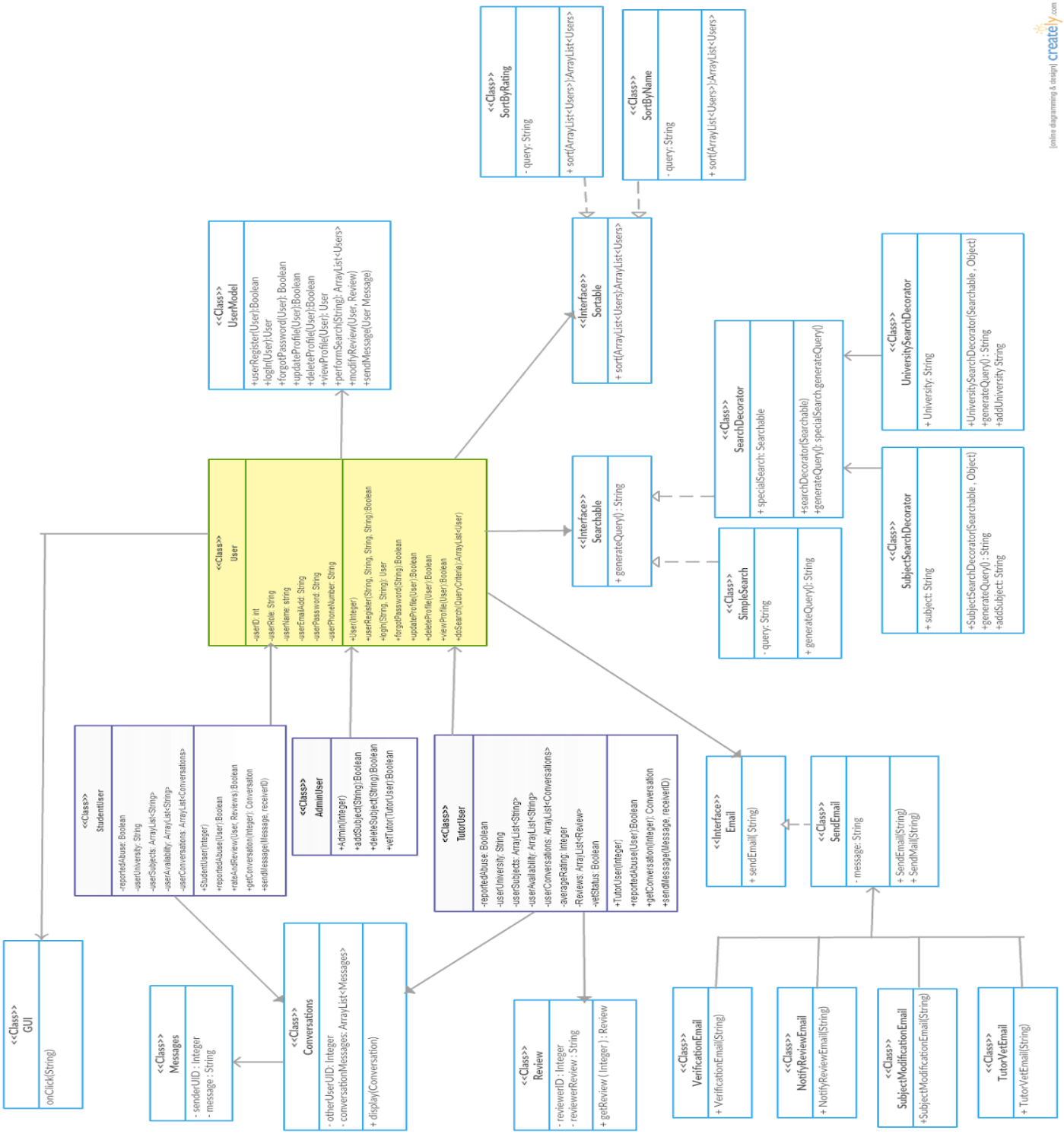
Based on our Project Part 1 we implemented required functionality and 1 stretch goal of rating and reviewing.

Use Case #	Use Case Name
UC-01	Create Account
UC-02	Login
UC-03	Forgot Password
UC-04	Update Profile
UC-05	Delete Profile
UC-06	Search Tutors/Students
UC-11	Report Abuse
UC-12	View Profile
UC-13	Rate and Review Tutor
UC-14	Add Subjects
UC-15	Delete Subjects
UC-16	Vet Tutors

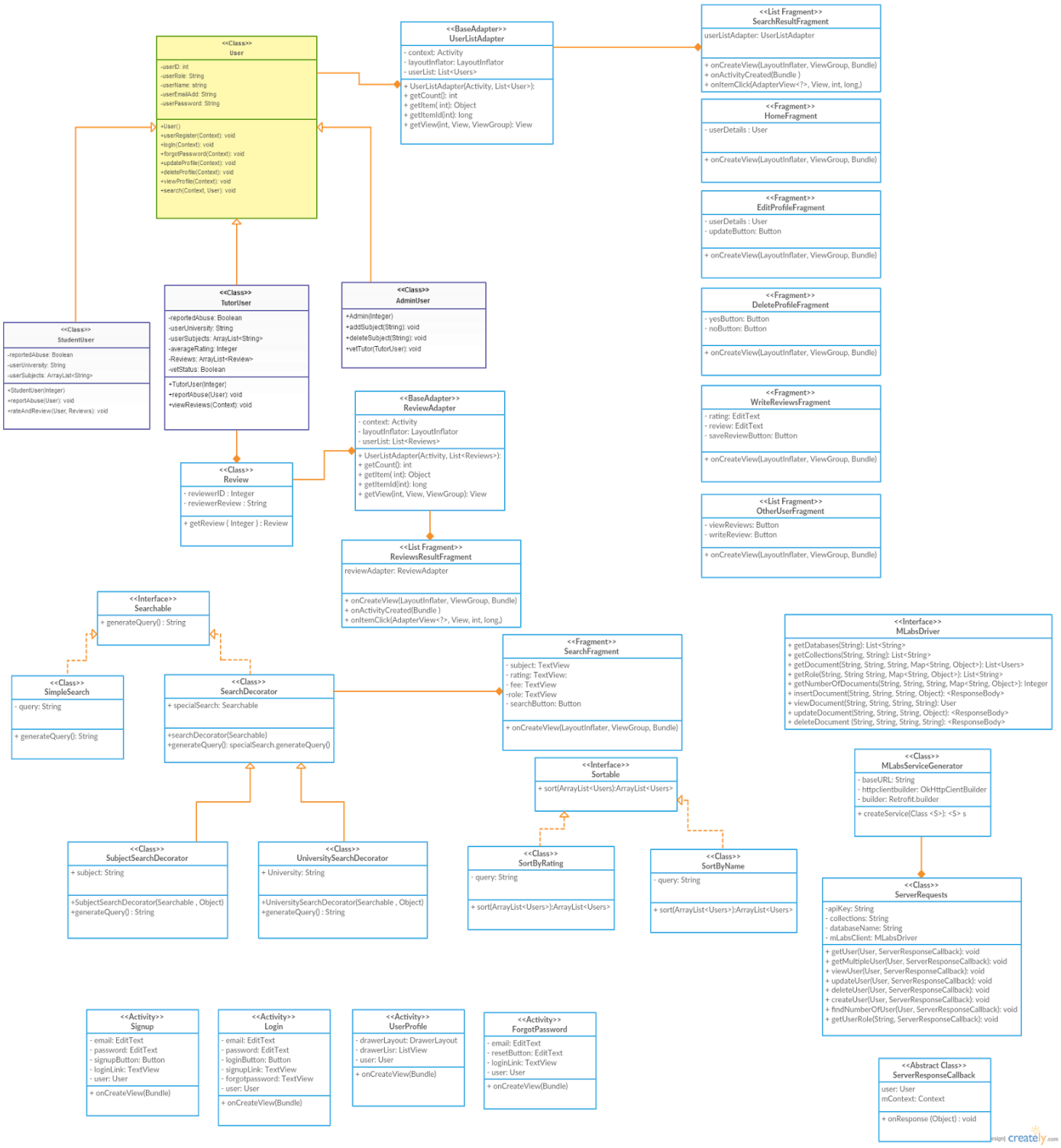
2. Which features were not implemented from Part 2?

UR - 007: Messaging was not implemented
FR - 001, 002, 004, 005, 006, 007: Emailing were also not implemented.
Android did not have support to send system emails from mobile devices - like web applications have. We had not anticipated this bottleneck while drawing initial design of our system.

3) Initial Class Diagram (Link)



Final Class Diagram (Link)



Show your Part 2 class diagram and your final class diagram. What changed? Why? If it did not change much, then discuss how doing the design up front helped in the development.

Changes:

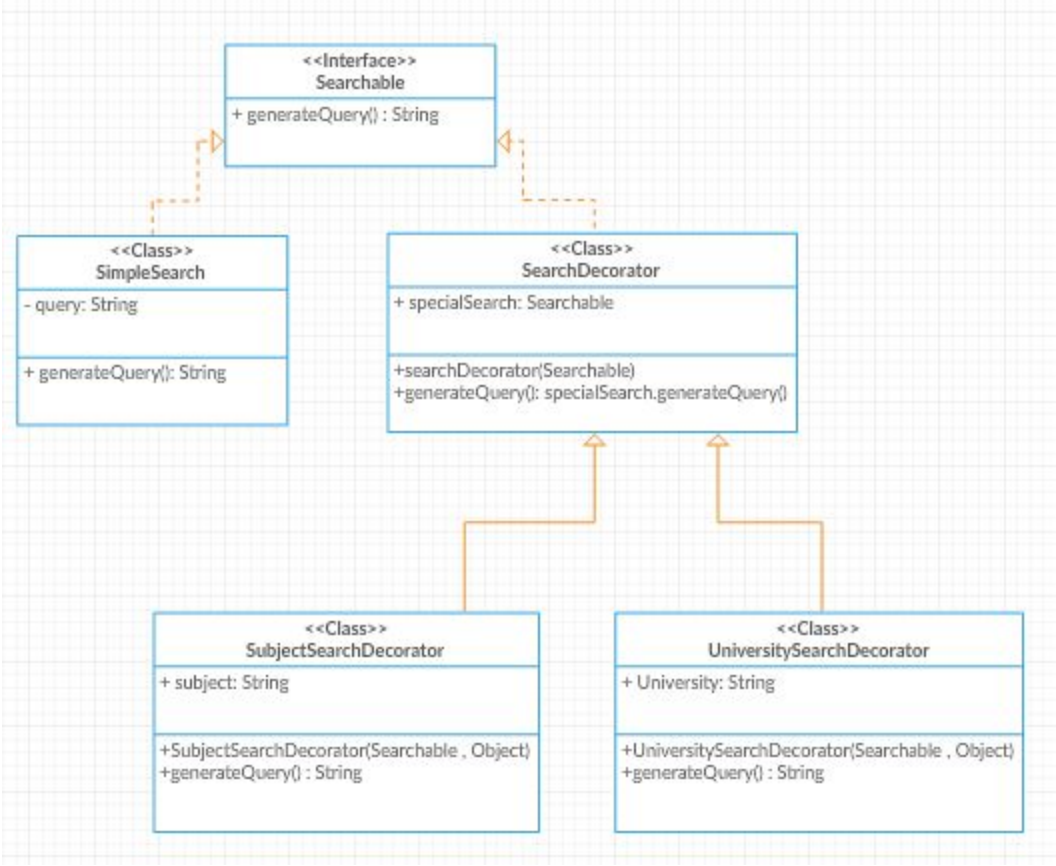
1. No Messaging and Emailing classes - Since we could not implement these two features.
2. Classes we did not not include in our initial diagram.
    a) Activities & Fragments - These are classes provided by Android for Android Development, as we did not have experience with Android before, this was not incorporated in our when we first formulated our design. These are classes specific to Android that handle the GUI.
    b) MLabs Drivers & Generator - We did not have a driver, hence had to write our own to access the REST API.
    c) ServerRequests & ServerResponseCallback in place of the UserModel: To decouple the HTTP requests from the model class, we then created these two classes that implement the calls from the MLabsDriver interface. This helped make coding much more simpler - we had the choice of experimenting with different online databases / REST API's without changing our User class.

Things that did not change:

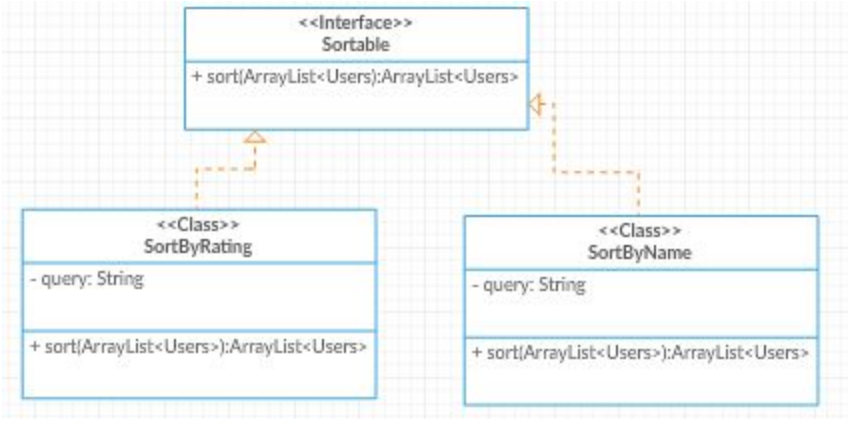
1. User Model: Having this class helped in 3 ways:
    a) ORM, although we had not learnt about ORM until then in class, when we began coding understanding the Retrofit library became much easier with our User Model in hand.
    b) Working with Activities & Fragments: We could easily map view data into our model.
2. Search & Sorting Patterns:
    a) The search decorator pattern helped us add the extra functionality of rating and reviewing quickly. We could initially test the code on basic User Attributes and then just extend the patterns to accommodate ratings and reviews.

4) Design Patterns used in the final Prototype

- Decorator: We used the decorator design pattern, to implement search in our project.



- Strategy: We used the Strategy design pattern, to sort the results coming out of search according to how the user wants it.



- Adapter: We used the Adapter design pattern, to adapt the inbuilt ListView, RecycleView available in android according to our need.

- Observable: We used the Observable design pattern, to allow the different fragments (Fragment represents a behavior or a portion of user interface in an Activity) we created as part of an activity to communicate between each other, so if there is any change in any of the fragment other fragments are notified.

5) Our project has been a big learning curve for us, as it has helped us in learning the concepts and understand the importance of Object Oriented Programming, Design & Analysis. Initially our project started by gathering the requirements needed for the project and designing them in a way how we wanted it to work, this was the most time consuming part in our project. Once we started coding, it was a bit easier initially to be get the basic setup up and running as we knew already what we want do. But since our's was an android app, we had to play according to its rule, and analyzing it we found out we could make things work the way we wanted it to. The Object Oriented Programming helped us to make the code look modular, and easier to change or extend them when required.

So the main learning is that a project isn't a single iteration of Design, Programming and Analysis, it is a final product of several iterations of above stated three principles.