Team 10 - Design Document

Table of Contents:

Purpose:

Life as a college student can be challenging. New students need time to adapt to the often confusing credit system, while continuing students repeatedly require assistance in the process of selecting courses and buying/selling books. *GraduatR* guides each student by providing them with a personalized profile, access to an elaborate rating system, a marketplace for book exchange, and a social media outlet for communication with other students all in one location.

A. Our **functional requirements** include the following items:

1. Creating personalized profiles

   <u>As a user,</u>

   1.1. I would like to be able to create a *GraduatR* account using Google, Facebook and/or customized login credentials.

   1.2. I would like to login and manage my *GraduatR* account.

   1.3. I would like to customize my *GraduatR* profile.

   1.4. I would like to provide a profile picture for my profile.

   1.5. I would like to be able to set my current status as an undergraduate student, graduate student or tutor.

   1.6. I would like to be able to delete my account when I want.

   <u>As a student,</u>

   1.7. I would like to be able to select previous courses that I have taken.

   1.8. I would like to be able to choose to keep my grades hidden from other users.

2. Accessing an elaborate rating system and course information

   <u>As a student,</u>

   2.1. I would like to able to search for courses and professors.

   2.2. I would like to be able to look at all courses available at Purdue.

   2.3. I would like to be able to rate professors on a five-star scale with respect to the specific class.

2.4. I would like to be able to rate courses on a five-star scale based on difficulty.

2.5. I would like to be able to give comments on classes I have taken and professors I have had.

2.6. I would like to be able to view the courses that other students have taken.

2.7. I would like to be able to give ratings anonymously.

2.8. I would like to view the ratings of a professor/course.

2.9. I would like to be able to see the average grades/GPA received in a class.

2.10. I would like to see the average exam difficulty of the exams in a class.

2.11. I would like to be able to flag inappropriate and irrelevant comments.

<u>As a parent,</u>

2.12. I would like to be able to view course details for classes.

2.13. I would like to able to view the final grade averages for classes.

3. Providing resources to users:

<u>As a user,</u>

3.1. I would like to be able to view on-campus events.

3.2. (If time allows) I would like to view past exams for any class.

3.3. (If time allows) I would like to be able to view any study material/notes for my classes.

3.4. (If time allows) I would like to see a map of my campus.

<u>As a student,</u>

3.5. I would like to be able to be able to sell books to other students.

3.6. I would like to be able to buy books from other students/users.

3.7. I would like to be able to look for tutors for a particular course that I need help with.

As a tutor,

3.8. I would like to provide my contact information to the students in need of help.

3.9. I would like other users to contact me directly through the application.

4. Giving the ability to communicate with other users:

As a student,

4.1. I would like to contact other students who have taken the class that I wish to take.

4.2. I would like to create group chats to reach multiple people at once.

4.3. I would like to be able to create study groups for a specific course with other students who are enrolled.

As a tutor,

4.4. I would like other users to contact me directly through the application.

B. Our **non-functional requirements** include the following items:

As a developer,

1.1. I would like this application to run on iOS devices.

1.2. I would like to improve the user interface and make it easy to understand.

1.3. I would initially like to handle 100 users using the app simultaneously.

1.4. I would like to protect the information that my users provide.

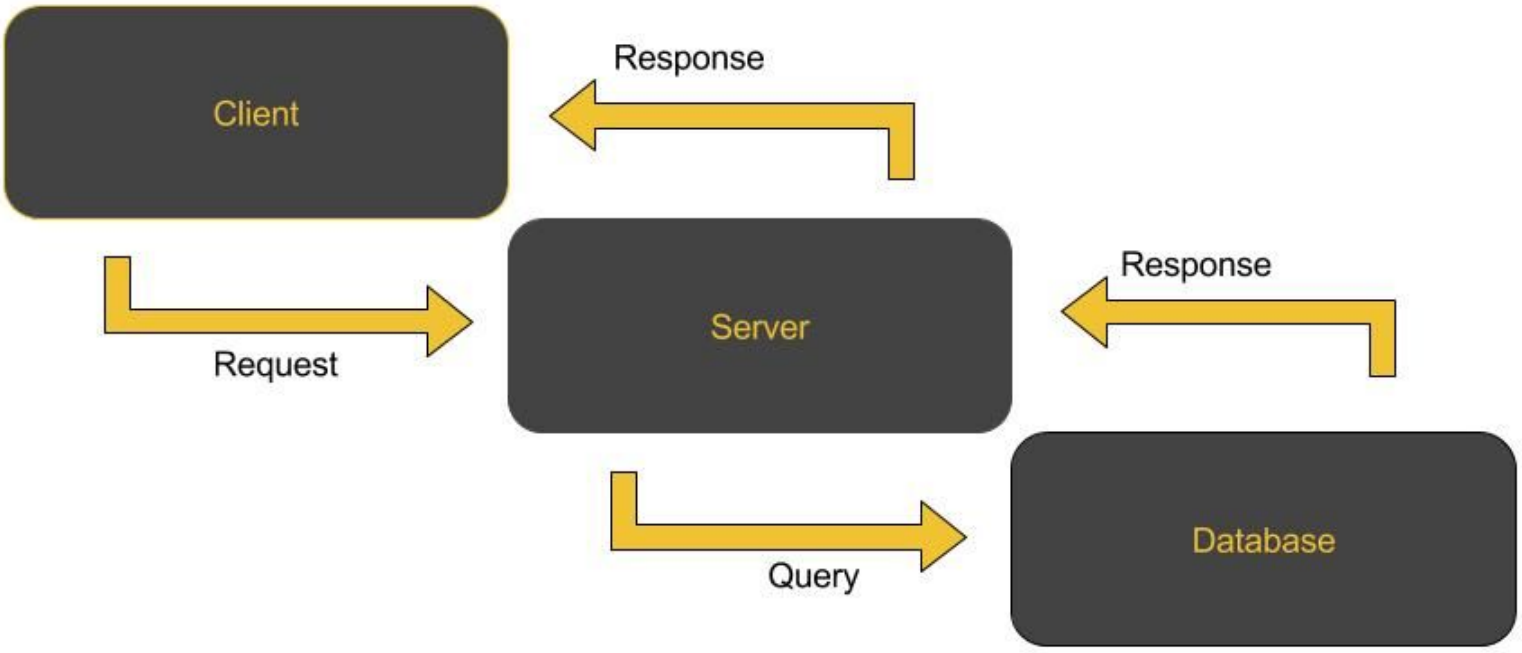
As a user,

1.5. I would like to access information through the cloud instead of updating the application periodically.

Design Outline:

GraduatR will be using a client-server model. The server will request and access information stored in several different databases namely a user profile database and a course and professor information/rating database. The following diagram represents the high-level overview of the same:

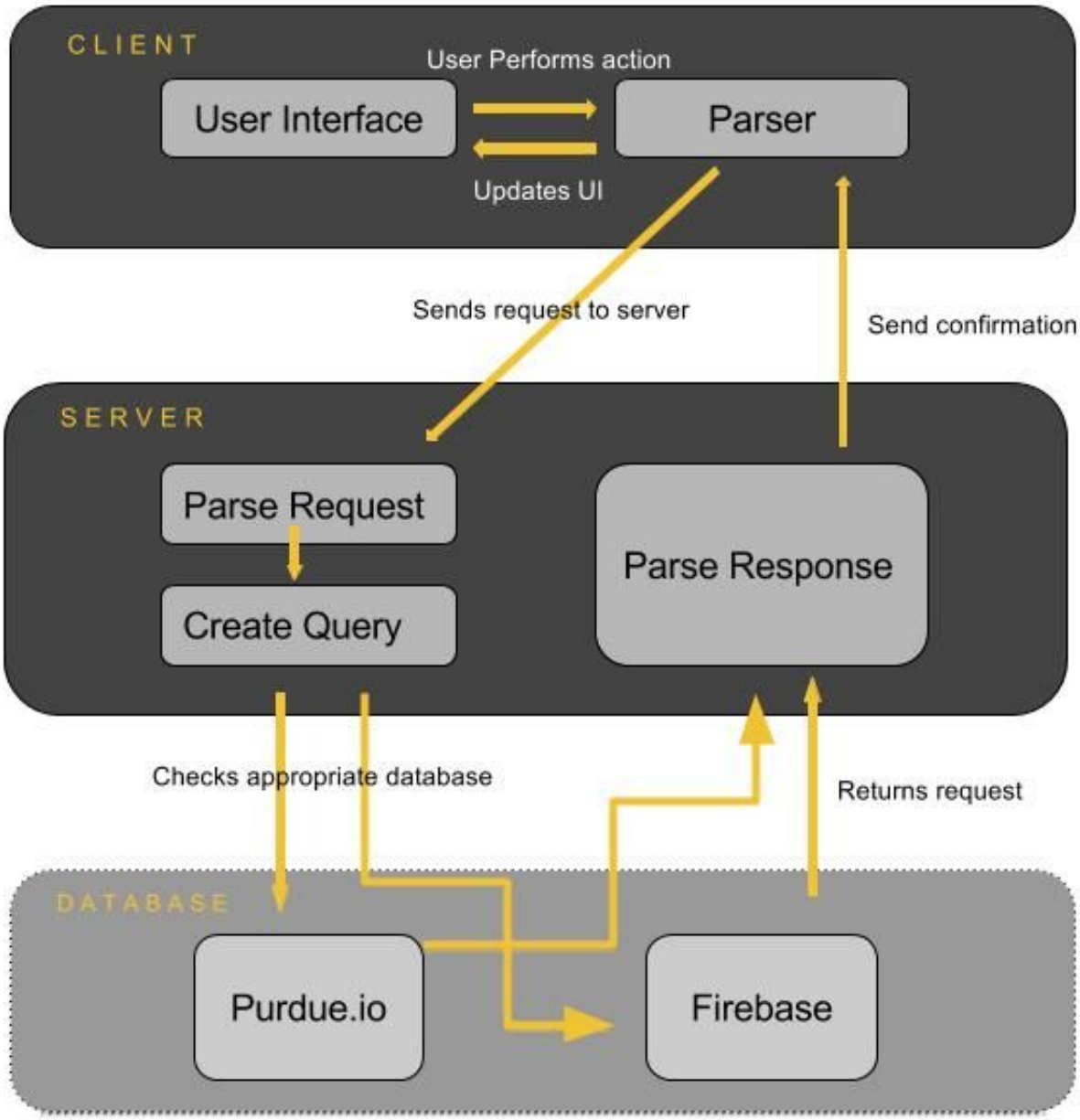
High Level Structure:



Components:

- *Client:*
    - Contains the user interface through an iOS application this will include everything from search bars, course based UI, rating UI etc.
    - Sends requests to the server through API keys, makes basic HTTP requests in JSON format.
    - Responses received from the server will then be parsed and formatted through the client to be put for display purposes.
- *Server:*
    - Receives and handles all client requests.
    - Translates client API requests to the respective database queries.
    - Return responses to requests from the client using the data acquired from the respective database queries.
- *Database:*
    - Firebase
        - Stores user information such as authentication, reviews, profile settings, etc.
        - Will handle authentication for Google and Facebook login
    - Purdue.io
        - Stores all Purdue University course listings and professors who have taught the course

UML Diagram:


CLIENT
User Performs action
User Interface
Parser
Updates UI
Sends request to server
Send confirmation
SERVER
Parse Request
Create Query
Parse Response
Checks appropriate database
Returns request
DATABASE
Purdue.io
Firebase

Design Issues:

A. Functional Issues:

1. Issue: How should users create an account with GraduatR?
    a. Option 1: Link existing user social media accounts (i.e. Google/Facebook) to create a GraduatR account.
    b. Option 2: Create a customized account system which will take in user credentials.
    c. **Option 3:** Allow user to choose either the social media link login or use the customized login process by GraduatR.

    Decision: We chose to give the user the power to decide whether they want to link existing user social media accounts to create a GraduatR account or use our customized login process. We felt that this would be more user friendly and would allow them to use whichever process they felt more comfortable with.

2. Issue: What kind of rating system would be used to rate professors and courses?
    a. **Option 1:** Rating system out of five stars
    b. Option 2: Rating using number system (i.e. 0-10)
    c. Option 3: Rating using word categories (i.e. Poor, Good, Excellent etc.)

    Decision: We chose to make use of a five star rating system as our rating system involves a review/comment section for each professor/course and hence this makes use of option three redundant. Finally we chose the five star rating system over the number based rating system as we feel the star based rating system is more user-friendly and visually appealing.

3. Issue: Should the announcement board be segregated with respect to different elements in address or should it be unified?
    a. Option 1: Have tutor updates, book sellers updates and on-campus events updates on one unified announcement board sorted chronologically.
    b. **Option 2:** Have the announcement board segregated into three tabs one for each of the elements listed above and announcements in each tab sorted chronologically.

    Decision: We chose to have the announcement board segregated into three tabs as this way we are able to have more information in each of the tabs while also making the announcement board more organized and user friendly.

4. Issue: How should the courses be listed?
    a. Option 1: Have the courses displayed under the college the course is administered from (i.e. Computer Science courses from College of Science)
    b. Option 2: Have the classes be displayed in alphabetical order independent of the college where the course is administered.
    c. Option 3: Have courses displayed numerically by ranges i.e. (100-200 level, etc.)
    d. Option 4: Have courses displayed by their categories (i.e General Education, STS, Great Issues, etc.)
    e. **Option 5:** Have courses displayed primarily under subject covered and have a secondary numeric division.

    Decision: We chose to have the courses listed under subject covered and then through numeric division as the Purdue.io API follows a similar

format which would make it easier to program. Further we could numerically order courses within each subject as students with lower class standings would want to have direct access to lower numeric class whereas students with higher class standings could quickly scroll down to their respective required courses. We chose not to have courses displayed by categories even though this is helpful because the lists of each category gets updated every subsequent semester which would lead to client being updated often. Further, the Purdue.io API does not segregate based on category hence we would have to manually update courses into each category.

5. Issue: How would inappropriate messages/comments/reviews be handled?
    a. **Option 1:** Have a flagging system wherein users can flag a message/comment/review and upon crossing a minimum threshold number of flags, the message/comment/review will be deleted by the administrator and a warning will be sent to the specific user who posted the deleted message/comment/review.
    b. Option 2: Delete the message/comment/review immediately after it is flagged once and a warning will be sent to the specific user who posted the deleted message/comment/review.
    c. Option 3: Give one user moderator privileges in each thread, chatroom and review section and allow this moderator to remove users, send warning messages and delete comments.
    d. Option 4: Have a filter system set up to automatically detect inappropriate language and disallow posts which have them.

    Decision: We chose to have a flagging system with a minimum threshold number of flags. This was preferred to option two as a single user's judgement of a message/comment/review is not sufficient enough to delete

a message/comment/review. Similarly, option three would involve the process of selecting a moderator and then relying on the moderator's judgement regarding the comments made which may or may not be universal. Lastly, implementing an automatic filter system would involve training a machine to detect foul language and could prove to be challenging.

6. Issue: Should students not enrolled in a particular course have access to the course chatrooms?
    a. **Option 1:** Yes, they should have complete access.
    b. Option 2: Yes, but they can only read messages.
    c. Option 3: No access, students will not be allowed to join chat rooms for courses they are not enrolled.

    Decision: GraduatR allows students who may have taken the course before to help solve doubts and concerns of students currently taking the course or planning on taking the course in the future. Hence we chose to allow each student user to have complete access to chat rooms but not misuse this ability.

B. Non-Functional Issues:

1. Issue: Fat Client vs Thin Client
    a. Option 1: Fat Client
    b. **Option 2:** Thin Client

    Decision: We chose to have a thin client as this would be the application size significantly smaller and make it easier to download for users. Further, application downtime would be reduced as minimal changes would be required to the client unless they are UI based updates. In which case, the users would have to re-downloaded the updated client, which would be a faster process for the smaller thin client.

2. Issue: Choice of database
    a. **Option 1:** Firebase realtime database
    b. Option 2: mySQL

    Decision: We chose Firebase over SQL as Firebase is a cloud based real-time document store and as we are creating an iOS application we can share one realtime database instance so that the client can automatically receive updates with new data.

3. Issue: Choice of platform
    a. **Option 1:** iOS application
    b. Option 2: Android application
    c. Option 3: Web application

    Decision: We chose to create an iOS application over an Android or Web application due to us product developers having more experience with the

iOS development environment. Further, the market for iOS application for college students at Purdue is larger than that of Android. Lastly, we chose not to create a web application as it difficult to update due to the response time for updates which can often be large.

4. Issue: Choice of coding language
    a. **Option 1:** Swift
    b. Option 2: Objective-C

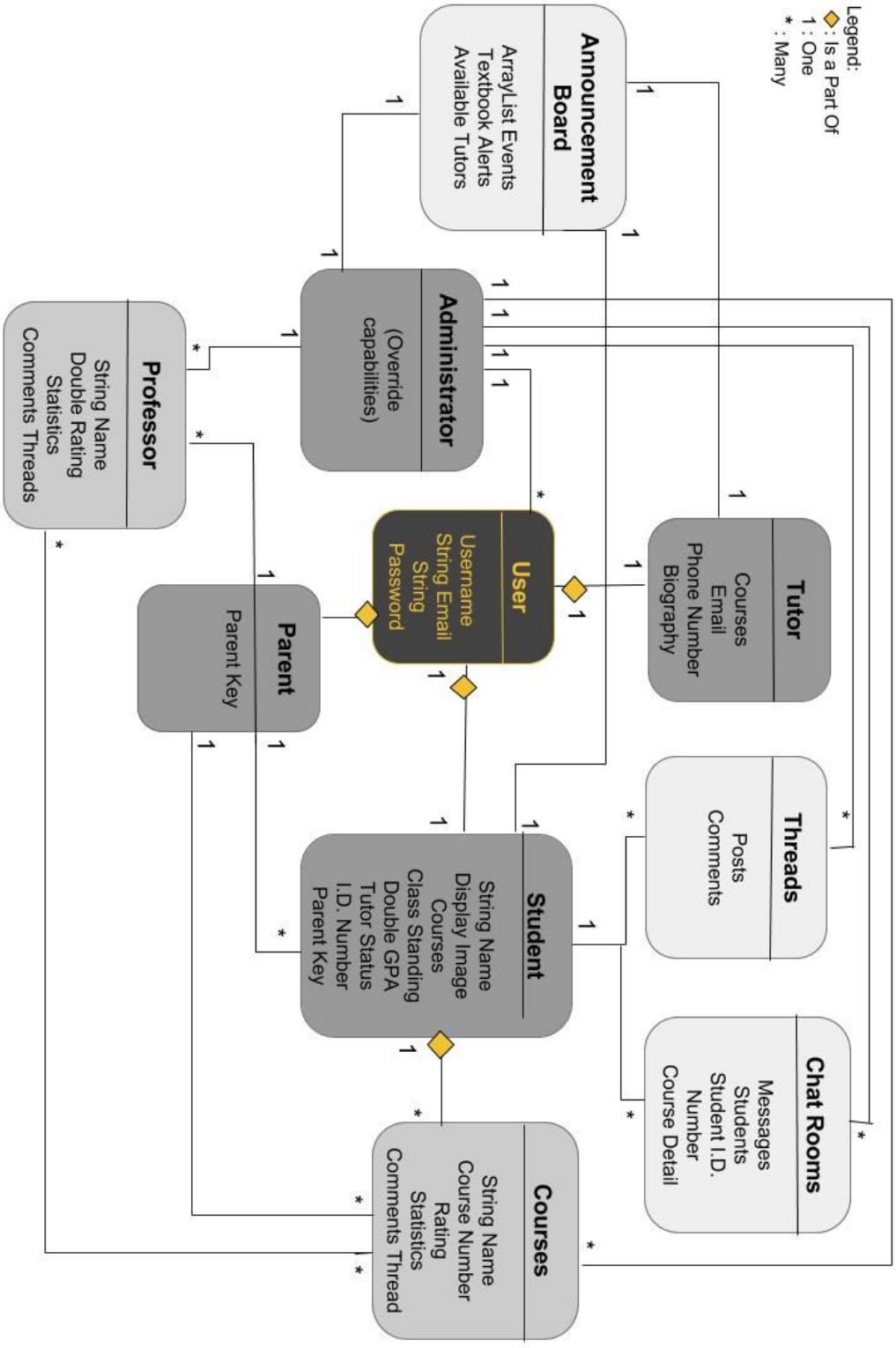
    Decision: We chose to use swift over objective-C as it faster than objective-C, easier and enables developers to be more productive. Further, most of the team members are familiar with swift over objective-C.

5. Issue: Distribution of Privileges
    a. Option 1: Have one user class one with multiple privilege levels depending on type of user.
    b. **Option 2:** Have seperate classes for different types of users and provide each type with a different privilege level.

    Decision: We chose option 2 as we felt it would be easier and more organized to have different classes as there are a lot of different variables for different classes which would be difficult to put into a single user class.

Design Details:

Class Diagram:



Class Description:

The following is our tentative database schema and is subject to change. The different colors represent the different types of classes. Chat rooms, threads and announcement boards are lighter as they are utilities and represent the social media aspect. Professors and Courses represent the databases for the same. The three different types of users - students, tutors, parents and administrators are the same shade as the represent particular entities accessing the features. Lastly, these four types of entities come under a large super user class which is given by the darkest shade.

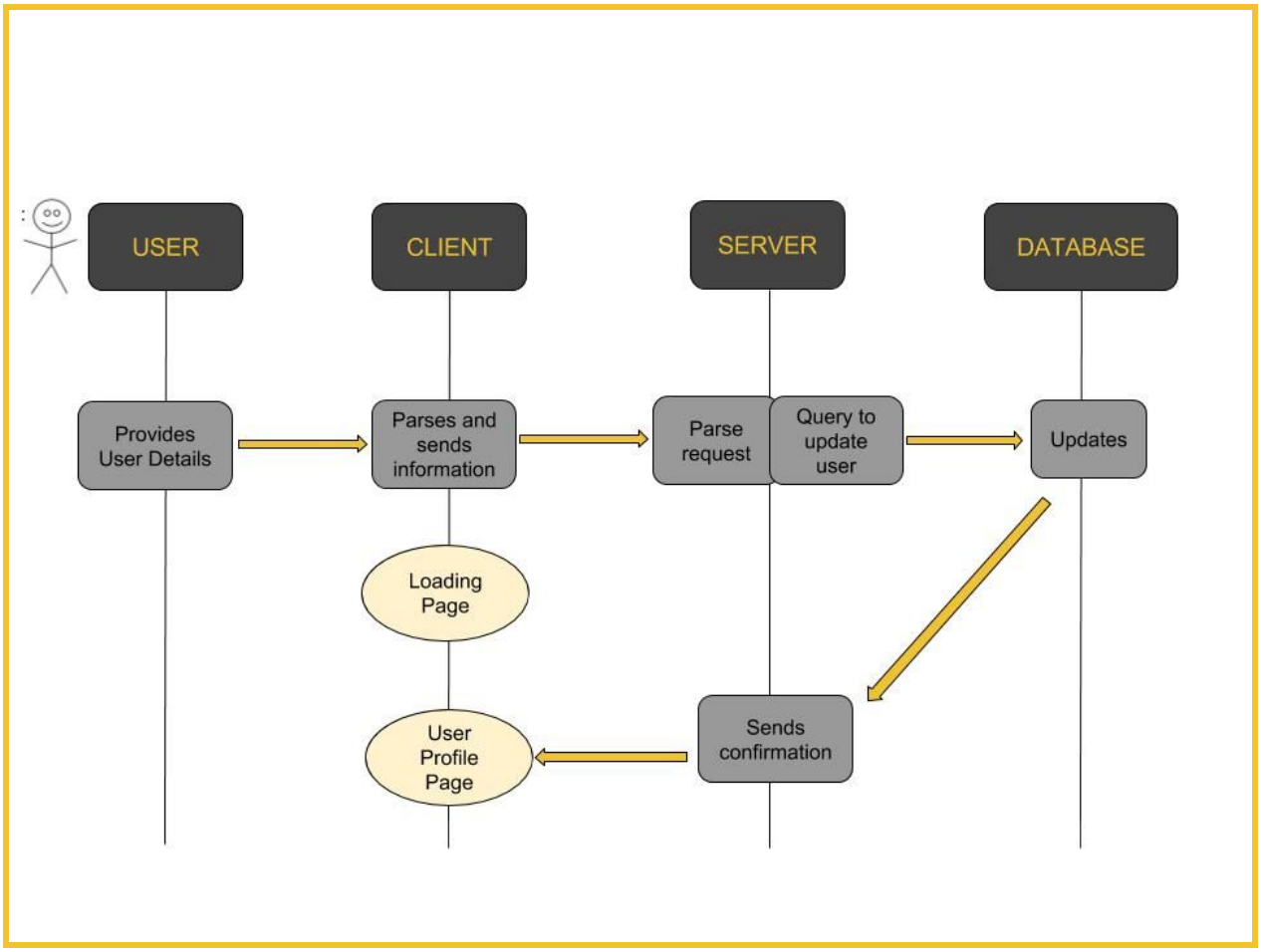
1. **User**
    a. All users are part of this user class, this forms the super class for all different types of users. These users consist of students, student tutors, tutors and parents.
    b. Contains each user's username, password and email address
2. **Student**
    a. Student contains a student identification number which is used for identification purposes.
    b. A student contains name, display picture, list of courses taken, class standing, student tutor status, GPA and parent key.
3. **Tutor**
    a. List of courses for which the tutor is tutoring, phone number, biography and email address.
    b. This class exists as privileges of a tutor are lower than that of a student.
4. **Parent**
    a. Parent contains a parent key which links the parent to the student(s), based on student approval.
    b. Parent just has access to course information and professor ratings.
5. **Administrator:**
    a. Administrator has access to announcement boards, threads, professor, chatrooms, reviews/comments and course reviews.

b. It has the ability to shut down, override comments and threads and can also delete users upon reports.

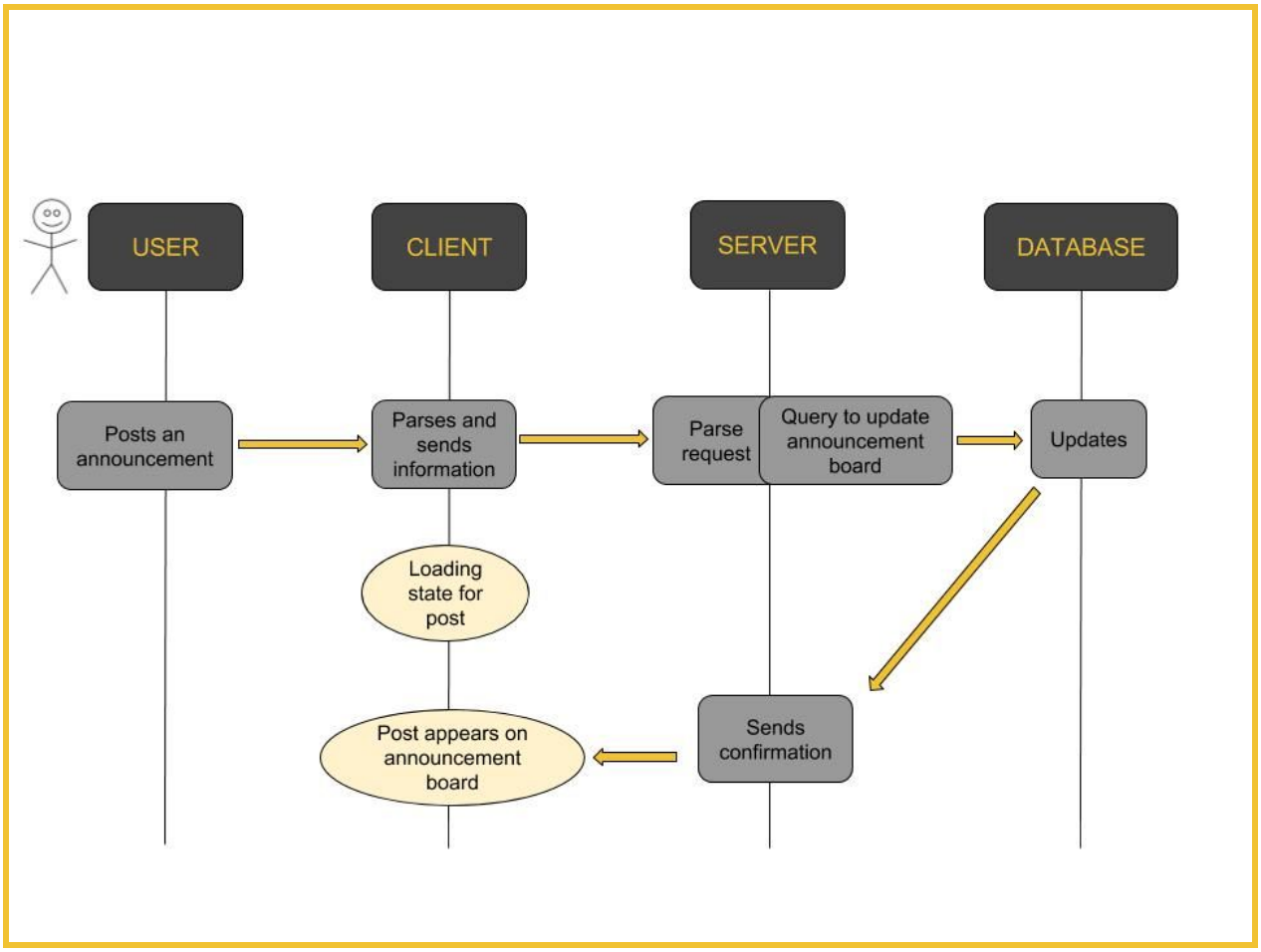
6. **Announcement Board**
    a. Announcement board contains announcements including on-campus events, textbook selling/buying alerts and available tutors.
    b. Announcements boards can be updated by tutors and students and be viewed by students.
7. **Discussion Thread**
    a. Students have access to many discussion threads for distinct topics where they are able to post questions and comments about anything.
    b. Discussion threads will contain students posts, comments and student identification numbers of students who are part of the thread.
8. **Professors**
    a. Professors contain name and a list of ratings given by students. They also contain an average rating of all ratings combined with respect to particular courses.
    b. Professors have a thread of comments/reviews provided by students.
    c. The ratings and reviews/comments are accessible by students and parents.
9. **Courses**
    a. Courses contain a name and a course number.
    b. Courses contain a list of ratings as well as an average rating of the class. They also contain statistics such as exam difficulty, average grade received, etc.
    c. Courses have a thread of comments provided by students and accessible by parents and students.
10. **Chat Rooms**
    a. Students are able to create chat rooms based on the course to communicate with others in their classes.
    b. A chat room will contain a list of messages, students, user identification numbers and course detail.

Sequence Diagrams:

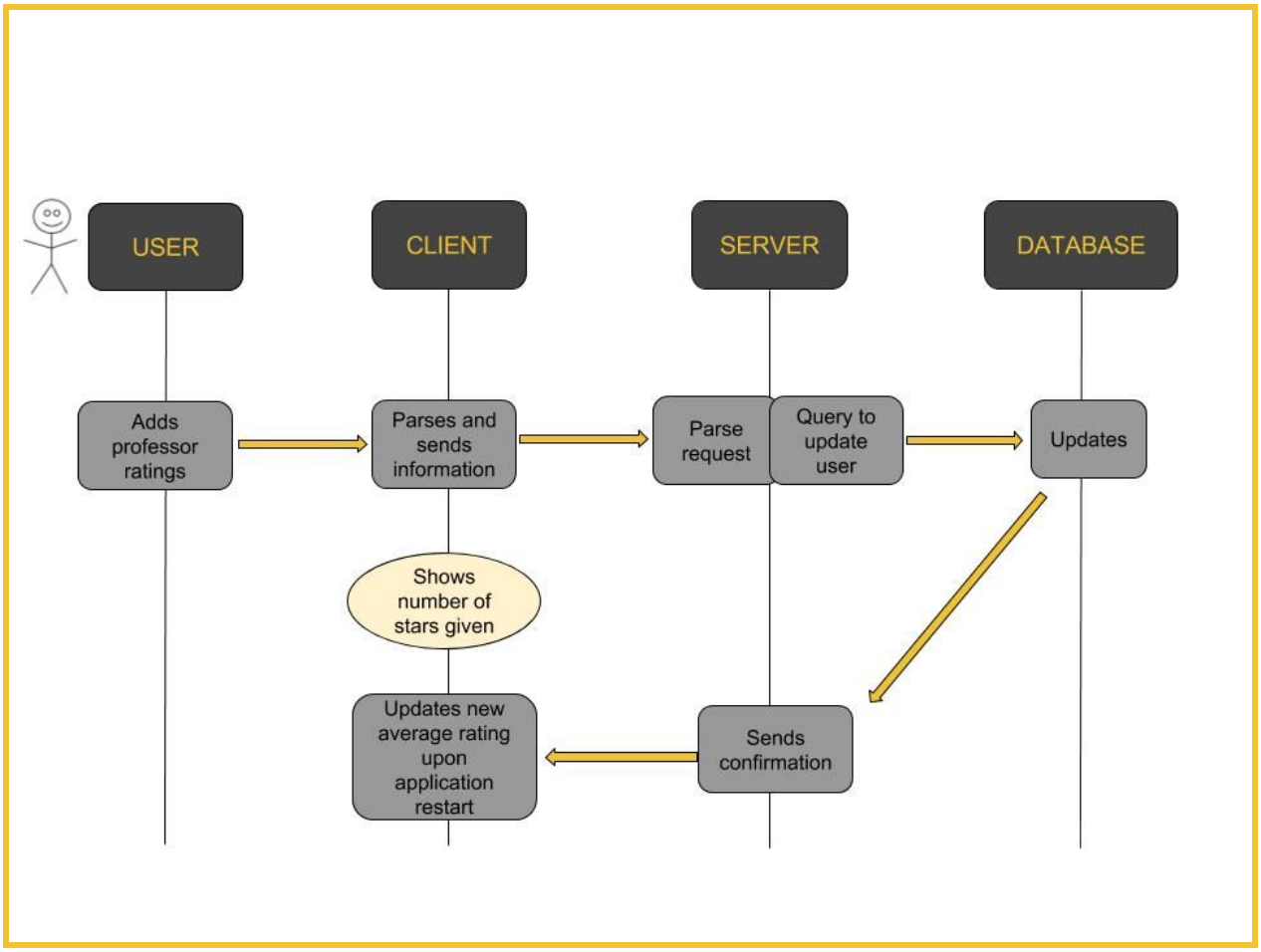
Sequence of events when a user creates a profile



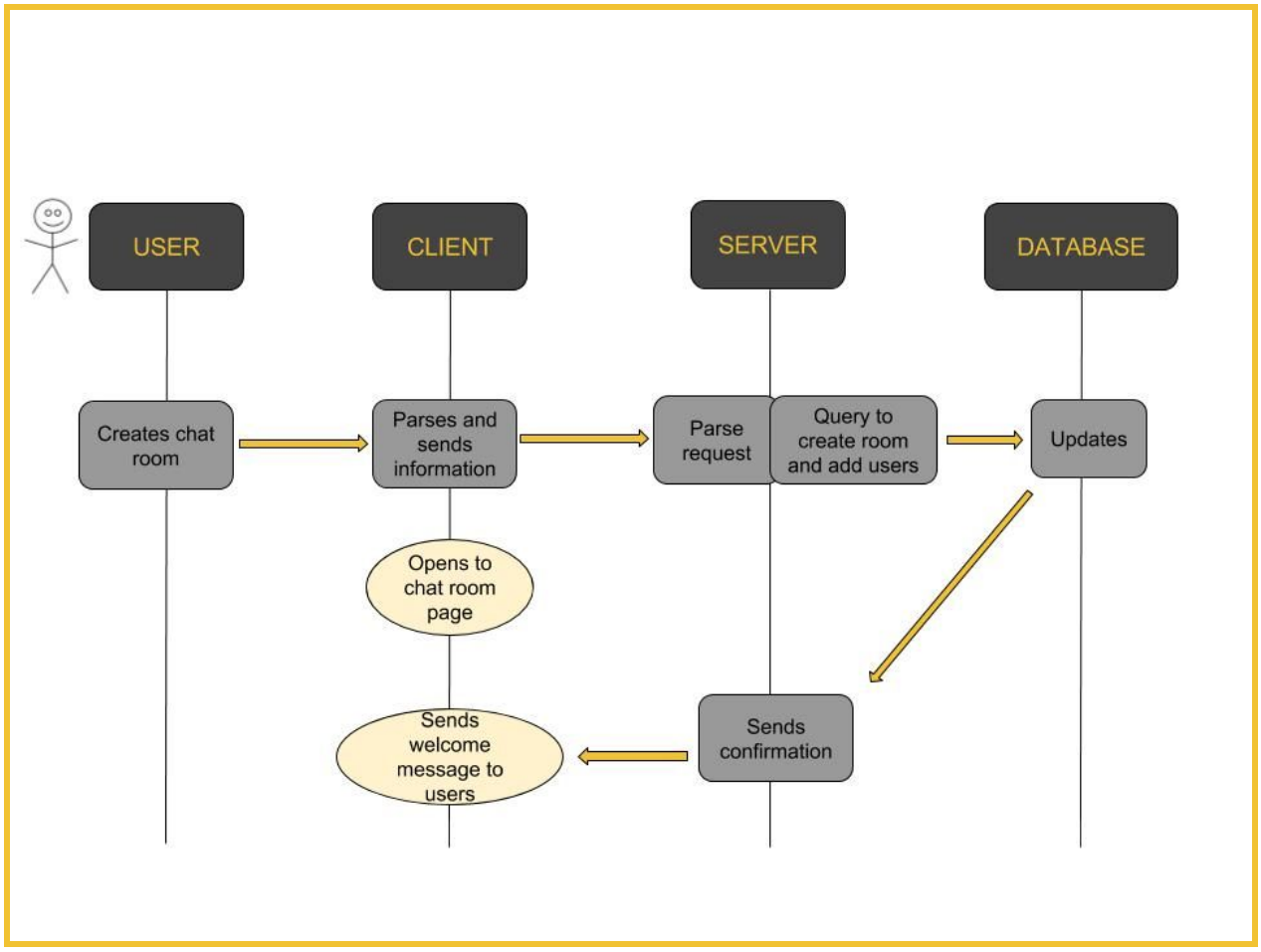
Sequence of events when announcement boards are updated by users



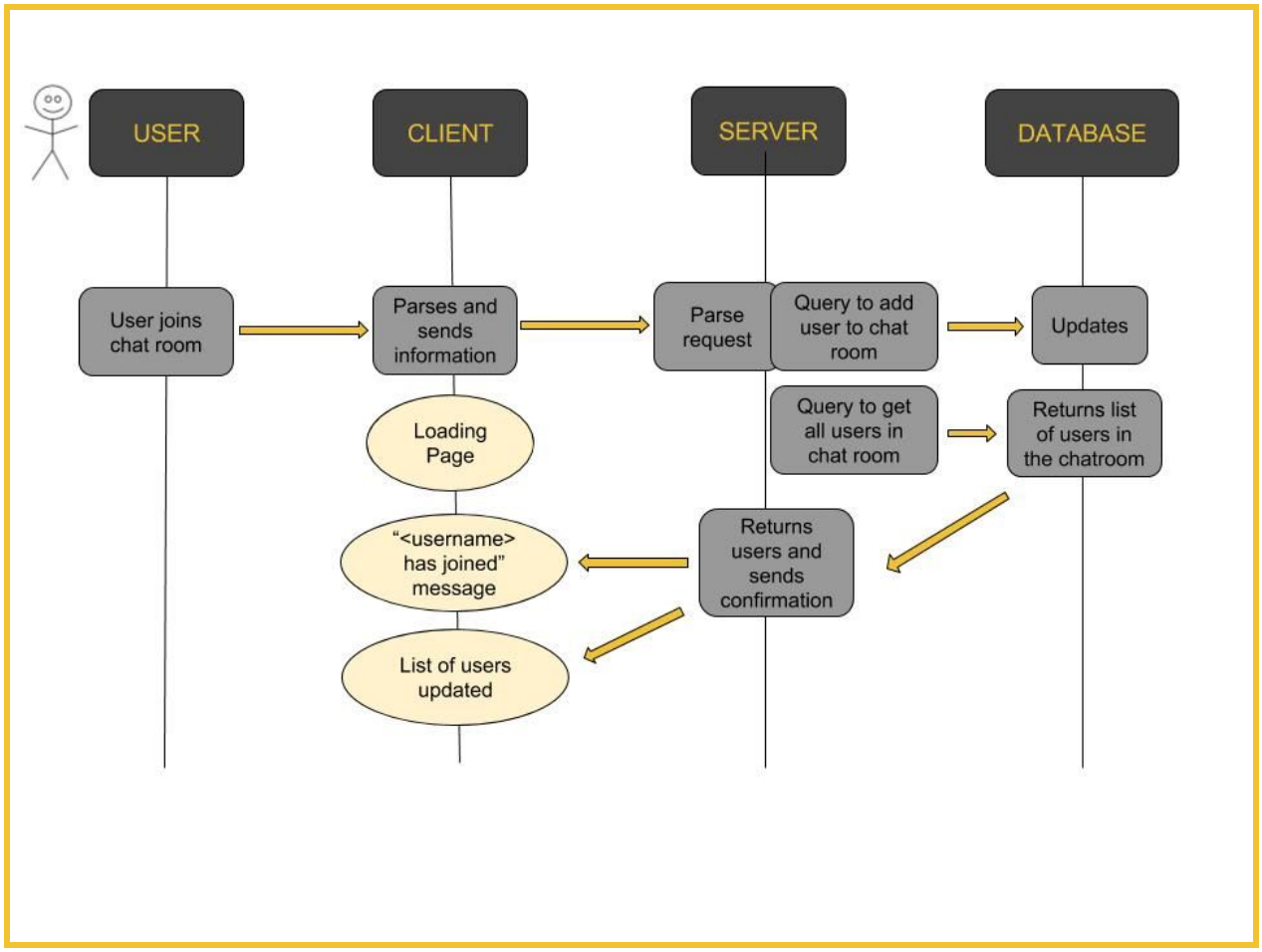
Sequence of events when professor ratings are added



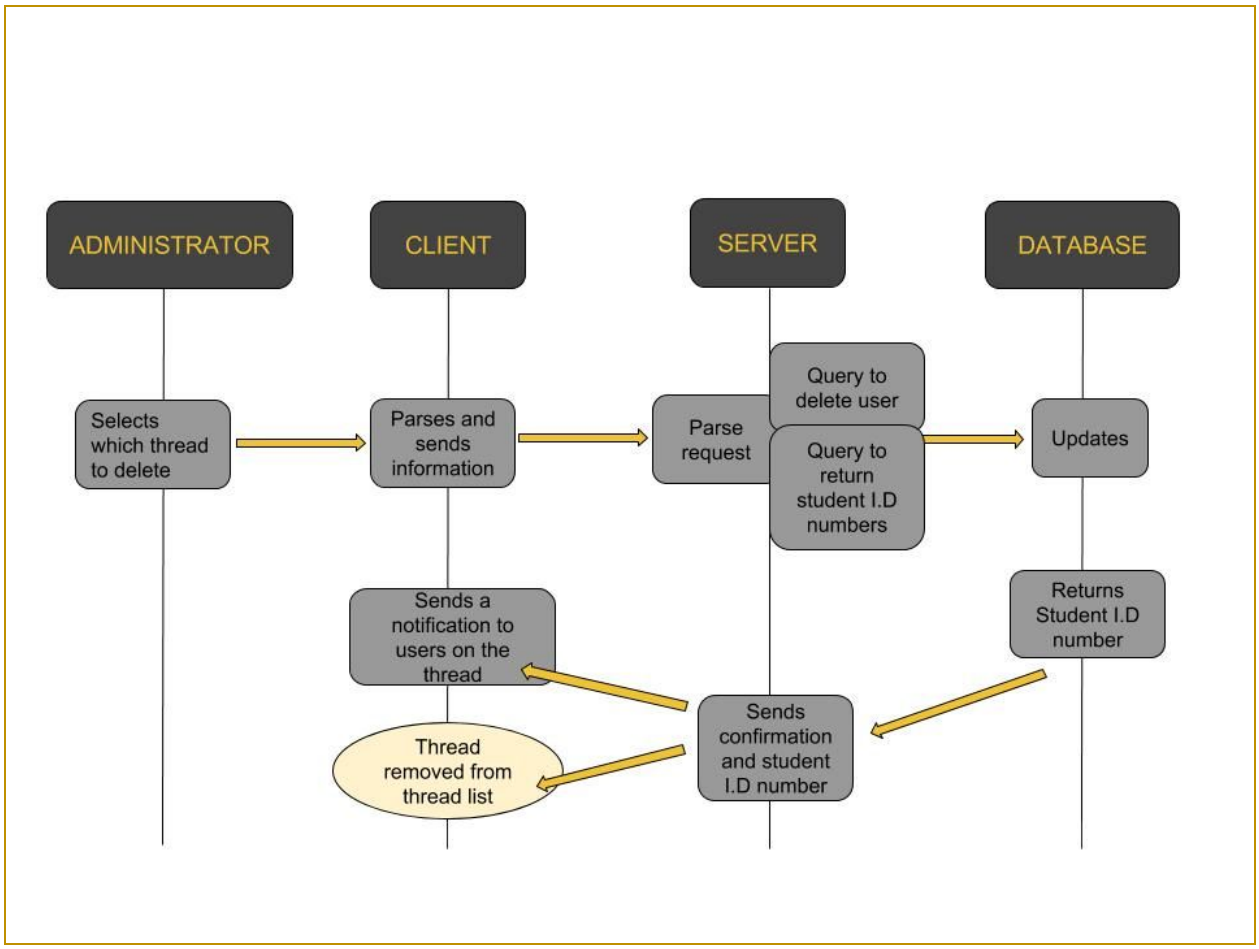
Sequence of events when a chat room is created



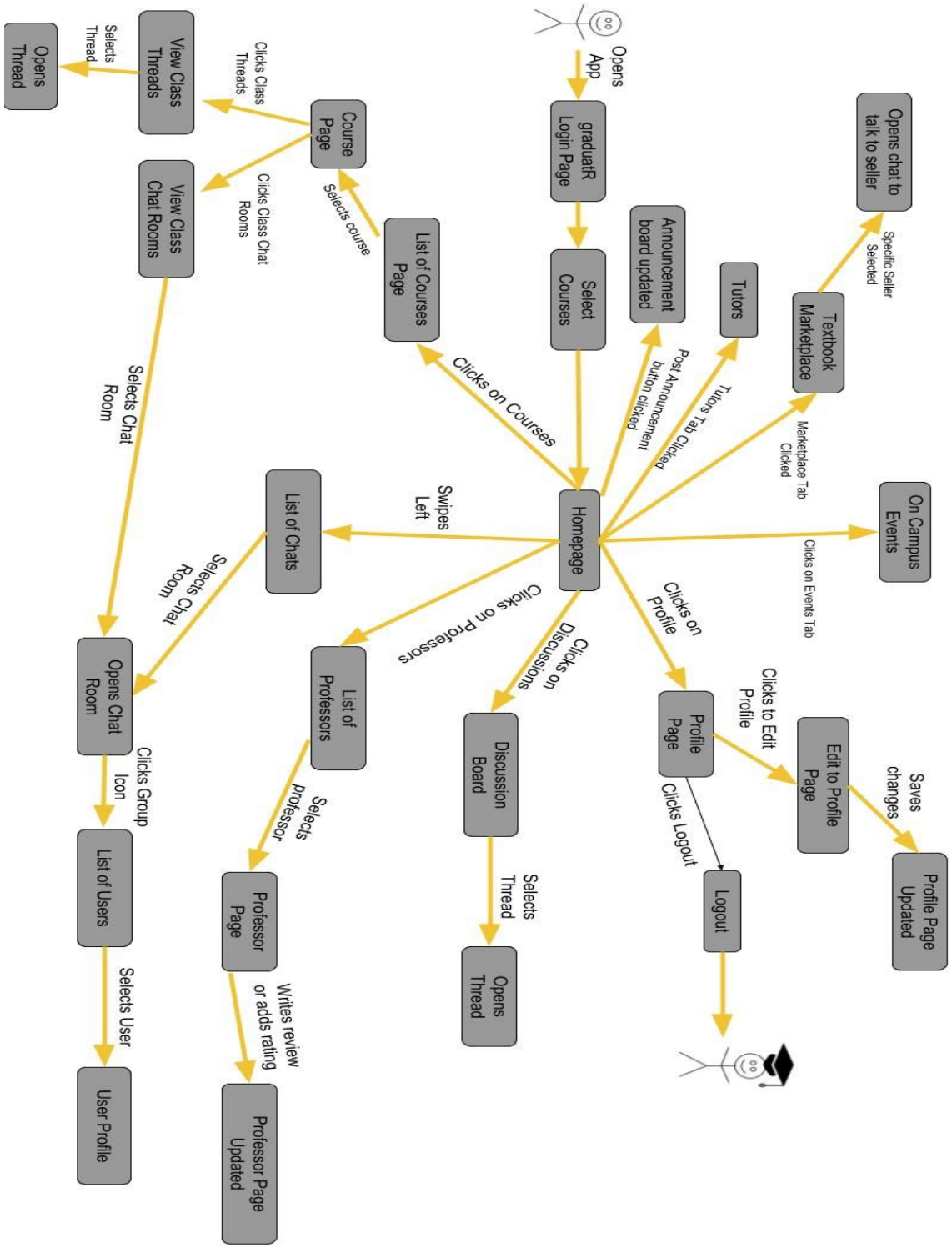
Sequence of events when user joins a chatroom



Sequence of events when an administrator shuts down a thread

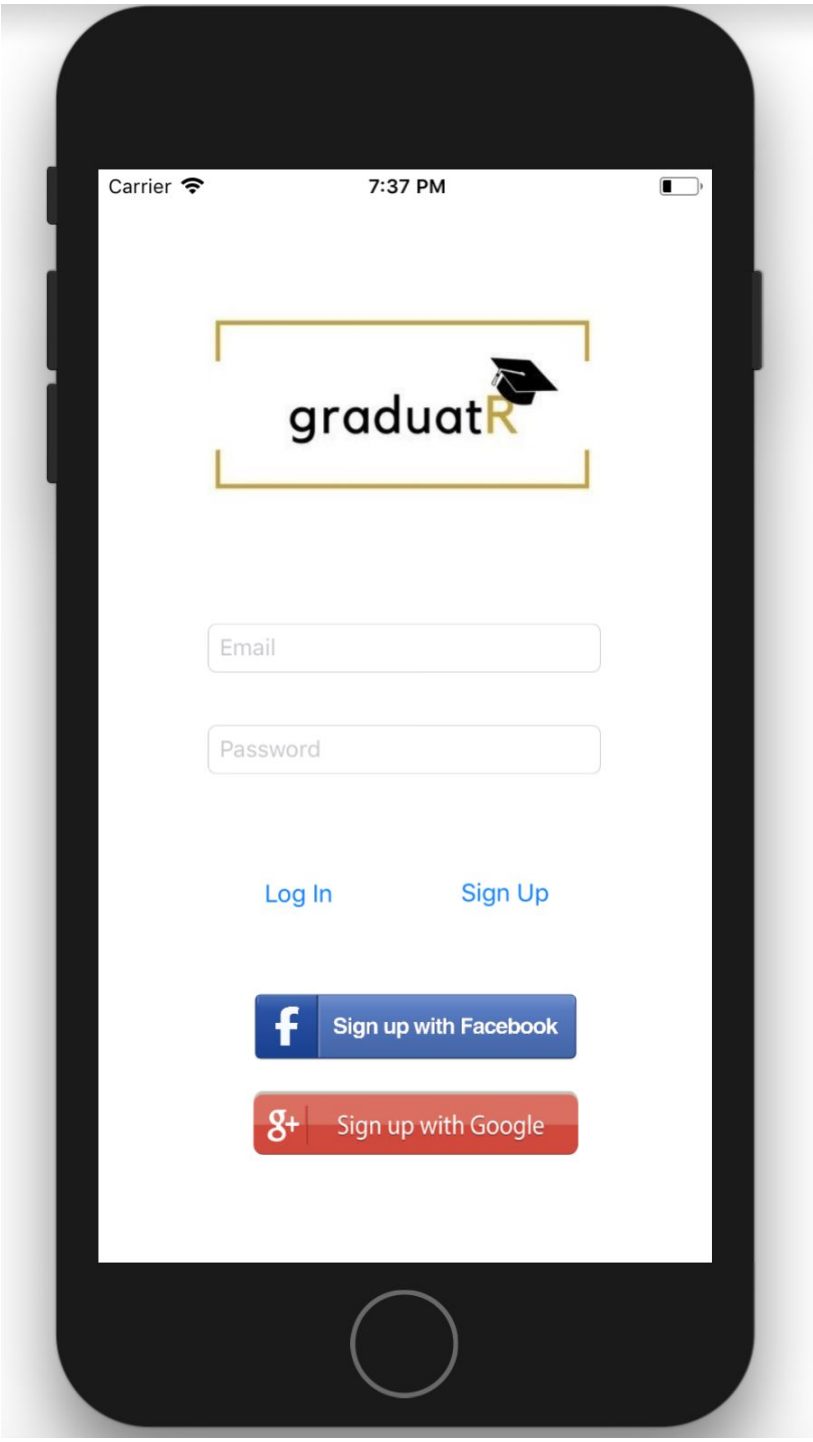


Activity Diagram

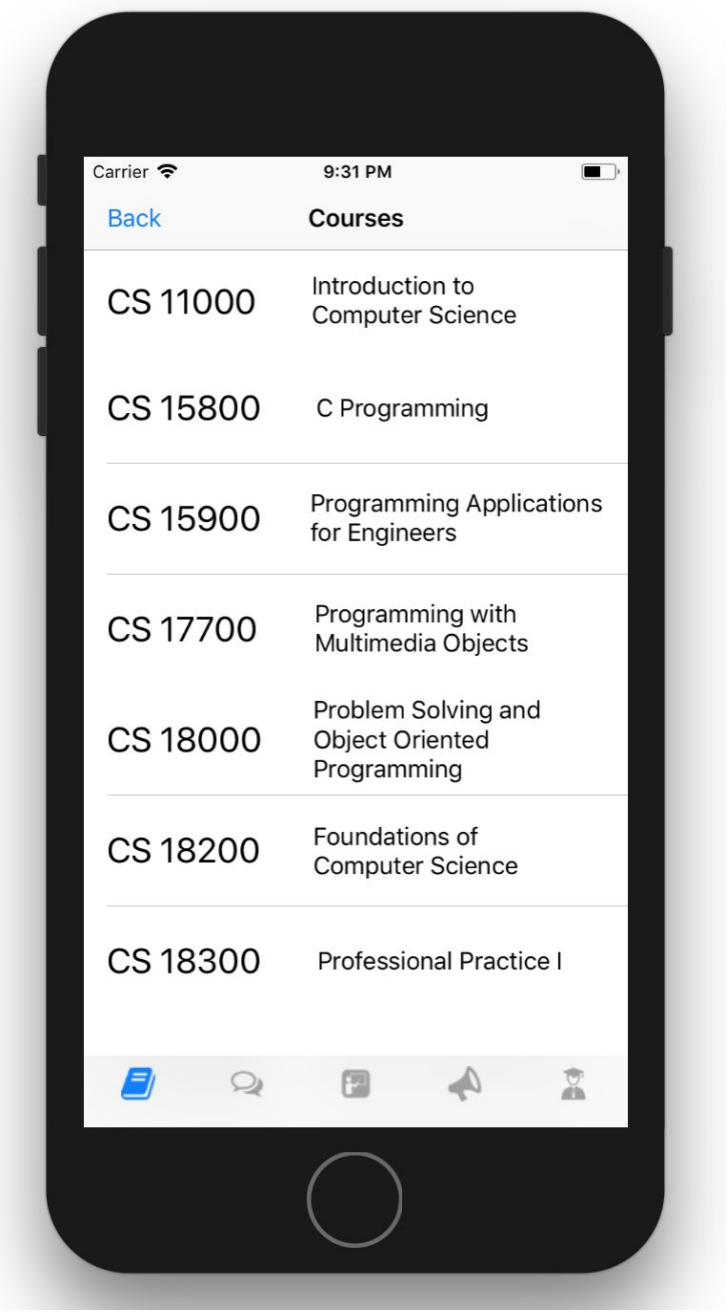


UI Mockups

The user can log in.



The user can look at the courses available at Purdue.



The user can view his/her profile.