Team 18 - Sprint Overview

Study Buddy

SCRUM Leader: Andy Chen
Team Members: Abhijay Gupta, Joseph Miller, Nathan Scott, Zeb Wezensky
Meeting Schedules:

- During (former) CS307 class time (12:30p-1:20p)
- Saturday Afternoon Coding Session (2p-4p)

Overview:
In this sprint, we plan on first learning the languages (Android Java, Django, Python, JSON, etc.) that will be needed to implement the project components. We plan on doing some back end tasks, including being able to get the server up and running by the end of the sprint, as well as getting the database up and running. We plan on doing some front end tasks as well, such as linking the classes and setting up navigation, setting up the Purdue.io aspects of the program, setting up user bios (not linking with the database yet), and start getting the search function to work.

Setup (Abhijay, Andy, Joey, Nate, Zeb)

- As a developer, I want to become familiar with the environment I will be working in and the technologies I will be working with.
  (**75 hours**)
  - Download development environment (Android Studio), get server with database up and configured, learn Android programming, implement small app with database connectivity to become familiar with coding, debugging, and deploying processes.
  - Acceptance: Given an Android Studio session, when I try and implement an application that connects to a database, then I have the skills to do so.

Front End (Andy, Zeb, Nate)

- Create all of the basic classes with navigation UI and linking the classes
  (**15 hours**)
  - Create the main page, profile page, edit picture page, course page, search course page, chat page, contact pages, settings, page, setting theme page, setting notifications page, setting

    blocklist page, settings reset page, feedback page, about page, pre-login page,
    - Acceptance: Given any page within the application, when I press a navigation button, then I am able to switch to the desired page.
- As developers, we want to use the Purdue.io API to retrieve a list of current classes.
  (**10 hours**)
    - Using the Purdue.io API, we want to be able to retrieve and parse the JSON output
    - Acceptance: Given a class search page, when I type in a class name, then I retrieve the existing class in Purdue.io.
- As a user, I want to be able to manage my profile (bio, etc.).
  (**5 hours**)
    - On the profile page, we want to edit basic information about the user: major, classification, courses, interests.
    - Acceptance: Given a user status, when I want to manage my profile, I am able to manage my profile.
- As a user, I want to be able to enter a short bio to describe myself.
  (**5 hours**)
    - On the profile page, we want the users to be able to a short description about themselves.
    - Acceptance: Given a user status, when I want to introduce myself to classmates, I am able to write a short bio about myself.
- As a user, I want to see a list of suggested classes while typing a class in.
  (10 hours)
    - As users search for classes in the search box, a dropdown box should show up with similar class names.
    - AutoCompleteTextView
    - Acceptance: Given a class name, when I type into the search box, then a list of classes with similar name will show

Back End (Abhijay, Joey)

- Become familiar with the technologies
  (**20 hours**)
    - Learn basic Python programming, implement small programs to test functionality and understanding
    - Set up Django project, learn how to create APIs using Django’s libraries
    - Set up MySQL database, learn how to manipulate the database tables using Django/Python

    - Acceptance: Given our server, when I need to complete a task related to Django or MySQL, Then I know how to do it or know how to find out.

- Implement storage and retrieval of profile information for users (**10 hours**)
    - Acceptance: Given an api call related to profile information manipulation/retrieval, when the server receives the request, then the client receives correct data

Mixed/User Stories to be Accomplished

- As a user, I want to be able to login using my Purdue email and password.
    - Frontend: Create a login class UI
    - Backend: query database for user email ID
    - Acceptance: Given a purdue student, when they open the app, then they are presented with the Purdue login page and can login successfully
- As a user, I want to enter my current classes.
    - Frontend: Course search UI and search button
    - Backend: Implement storage and retrieval of class list
    - Acceptance: Given a list of classes, when I want to enter them on the app, then I am able to and the change propagates to the database.
- As a user, I want to be able to enter classes that I'm not currently taking.
    - Backend: Implement storage and retrieval of class list
    - Acceptance: Given a list of classes, when I want to enter them on the app, then I am able to and the change propagates to the database.
- As a user, I want to create a profile for other students to view.
    - Frontend: create and manage profile
    - Backend: store profile information
    - Acceptance: Given some profile information, when a user enters that information on the manage profile screen, then the changes propagate to the database.

Remaining Backlog

Functional:

1. As a user, I want to enter other interests so that I can be matched with a more compatible individual.
2. As a user, I want to be able to specify whether I am looking for a group of students or just one.
3. As a user, I want a “feed” of announcements about classes I’ve enrolled in.
4. As a user, I want to be able to post to the announcements feed for each of my classes.
5. As a user, I want to view other people’s profile.
6. As a user, I want to communicate with other students before meeting with them.
7. As a user, I want to be able to meet with students as a group.
8. As a user, I want to see students who are in the same classes as me.
9. As a user, I want to see contact information for the students I am matched with.
10. As a user, I want to be able to customize my notification settings.
11. As a user, I want to get a notification when contacted.
12. As a user, I want to be able to chat with and send messages to other students.
13. As a user, I want to view my chat history.
14. As a user, I want to be able to report other users for inappropriate conduct.
15. As a user, I want to be able to block other users so that I do not see their messages.
16. As a user, I want to be able to customize the look and feel of the application.
17. As a developer, I want to match users based on what classes they are taking and interests they may have.
18. As a developer, I want users to be able to provide feedback.

Non-functional:

1. As developers, we want our application to be simple, straightforward, and easy to use with little to no introduction.
2. As developers, we want our application to utilize a clean and professional user interface.
3. As developers, we want our application to have no bugs if possible.
4. As users, we want our application to be as fast as possible without sacrificing features of the app.
5. As developers, we must be able to accommodate as many users as possible.
6. As developers, we want to store as little data on the client device as possible.

7. As a user, I want to be able to see all relevant information with minimal effort.
8. As a user, I want to always be able to access the app.
9. As a developer, I want to reduce server downtime as much as possible.