CS222 Project Proposal
By Divya Sundar, Khushi Maheshwari, Viven Puthenpurayil, Zuhair Ali

Pitch: Want a study buddy? So do we! This is why the study buddy web application can quickly and easily allow you to find where another student is studying for a specific class or assignment. The study buddy app will also allow you to filter through Reddit posts to see what other students are saying about classes you might be enrolled in or interested in.

Functionality:

- Users can create a profile and login to website
- Course descriptions are available to users about the class
- Users can see filtered and cleaned Reddit posts relating to classes they are enrolled in
- Different filters are available based on classes a user is currently enrolled in and classes a user is interested in
    - Users can add classes to their profile
- Users have the ability to make a post about where they are studying or working on an assignment
- Users have the ability to search through posts made by other users to find a buddy to study with

Design:

- Four different pages
    - My Profile: user can view their profile and see their information
    - Find a Friend: user can make a post about where and what they are studying and view posts made by other users
    - Browse: user can look through courses available on campus and view Reddit posts for each
    - Learn More: user can see quick bios of creators (adds a more personal touch to the web application)
- Below is a design of what we based our web application on (3 of the pages):
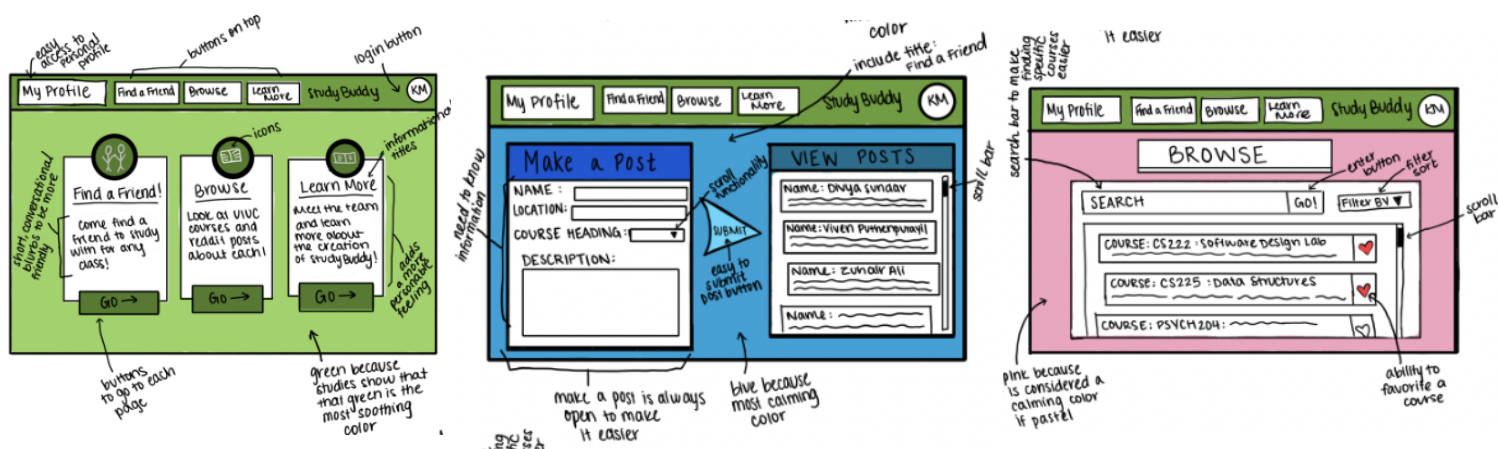
Components:

- Backend: The role of the backend component is to house and collect data when necessary to output it to the frontend
    - Store all of the user accounts, the classes and class descriptions, and previous chats
        - Coded in python using Flask
        - We will run the monolithic server on a separate .py file and connect the web application to it. This application will have a RESTful API so we can make simple HTTP requests between our own database and make requests to outside APIs like Reddit.
    - We will acquire filtered and cleaned Reddit posts in order to present real-time students' opinions on various classes. Using PRAW(Python Reddit API Wrapper), we will be able to scrape posts from different subreddits, as well as comments from a specific post.
        - [Python Reddit API Wrapper(PRAW)](#)
    - We will also acquire Wade's Dataset to get descriptions for each course for the course description page. We can parse the CSV file using Python and filter based on the course as well as the professor's name to interpret the data.
        - https://raw.githubusercontent.com/wadefagen/datasets/master/gpa/uiuc-gpa-dataset.csv
- Frontend: The role of the frontend is to make requests to the backend when it needs information to populate a certain page (such as a course page with the description and Reddit posts)
    - For the front end we are planning to create our code in Visual Studio Code using React JS, and we will easily be able to implement and test our code locally on local host
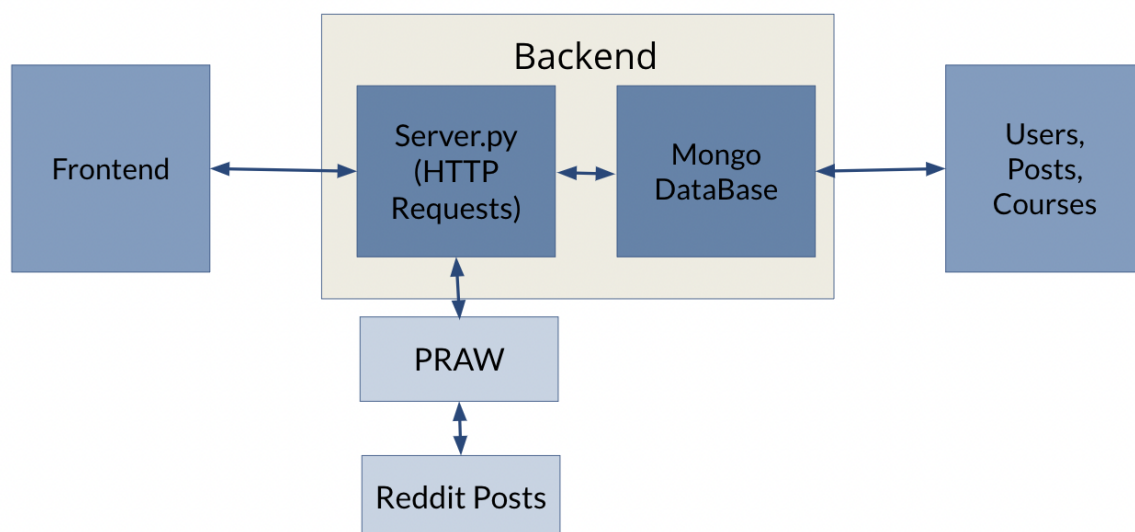
Continuous Integration:

- Unit tests are very important for development so we are going to test our components to make sure they are working correctly.
    - The [testing libraries](#) we are planning on using are Jest and Pytest which work very well with java and android studio to test applications
    - For the flask server, we will use Pytest and 'Coverage' to run tests and measure code coverage respectively
- In terms of an automatic linter, we are planning to use checkstyle because it works very well in Visual Studio Code for React JS and python
- We are planning on creating pull requests when we finish parts of a component or whole components depending on the time frame allotted for the component. The person who will be assigned the pull request for review will be the other person also working on that respective area (backend or frontend). If that person doesn't review the pull request add an alternative reviewer based on which person from the opposite area has more experience with what is in the pull request. To avoid merge conflicts there should be clear distinctions of what people are working on, so two people don't work on very similar items which reside in the same file.

Schedule/Weekly Planning:

- Week 1: 8/29 - 9/2: Finish project proposal draft
- Week 2: 9/5 - 9/9: Create GitHub and Visual Studio projects, and Testing structure
- Week 3: 9/12 - 9/16: Work on developing the server, designing the skeleton of the web application
- Week 4: 9/19 - 9/23: Begin developing the study buddy page by adding post features (including name, location, and what a user is working on)
- Week 5: 9/26 - 9/30: Populate the server with the courses and descriptions, and continue creating the study buddy page
- Week 6: 10/3 - 10/7: Acquire GPA and disparity information from Wade
- Week 7: 10/10 - 10/14: Implement PRAW so that the server can make requests to it for info from Reddit, test study buddy page
- Week 8: 10/17 - 10/21: Filter and clean Reddit posts
- Week 9: 10/24 - 10/28: Connect GPA and disparity information to the course Reddit page
- Week 10: 10/31 - 11/4: Add usability features such as easy-to-use buttons and create a visually appealing structure and logo
- Week 11: 11/7 - 11/11: This week will be used as a grace period because we know something will go wrong/get delayed so this week will be used to deal with the issue
- Week 12: 11/14 - 11/18: Conduct final testing (E2E/stress testing) of both the study buddy page and the course Reddit page
- Week 13: 11/28 - 12/2: Working on the final presentation
- Week 14: 12/5 - 12/9: Final Presentation

Stretch Goals:

- Directions to locations by implementing google maps and Illini Bus Routes
- Implementing a map feature to find locations near a user, as well as display major landmarks and study spaces on the UIUC campus
- Chat room implementation to allow users to directly talk to one another in a virtual chat room
- A feature to add friends to a user's profile

Risks:

- Some issues may be encountered with grabbing the data from the Reddit threads regarding the course descriptions. Reddit, being a social media platform and place for many to put their ideas as they are, can have lots of unnecessary and irrelevant information. Cleaning the data and outputting the correct and most appropriate data may be a challenge. The impact of this would be 1-2 days to find a method to get the most relevant data.
- Another source of risk may be connecting the server with React. This issue may stem from the lack of experience in using both of these components. We may struggle with adjusting to the use of cross-platforms and making everything work together smoothly. The impact of this would be 1-2 weeks when combining the different aspects, and we will utilize the help of our mentor for these issues.
- Stretch Goal:
  - The in-app chat room implementation may be difficult, as a result of the lack of experience in this as well. Connecting two users from the app without having a profile or a permanent mark on the app itself may cause issues. In order to solve this, we plan on researching more into the different options for a chat room in React. The impact of this will be ~1 week when creating the chat room feature, and we will utilize the help of our mentor and other documentation for this.

Teamwork:

Since we are creating a web application, our team is planning on utilizing MongoDB, Visual Studio Code, and GitHub. We will be creating different branches in GitHub that each member can push their work onto. Visual Studio Code will be the server we are using that will connect to our database in MongoDB. By pushing into different branches in GitHub, multiple people will be able to work simultaneously on the same project without disrupting each other's progress. We decided on these applications because all of the team members have experience to some extent working with them and the options for collaboration are available through them.

As a team of four, we will be divided up into two teams: one working primarily on the backend (Viven and Zuhair) and the other working more on the front end (Divya and Khushi)

development. We chose these teams based on our interest and current knowledge of languages and databases. Viven and Zuhair have experience working in Python and React and are looking to expand their knowledge on backend website development. Divya also has experience working with React JS and Khushi wants to work more towards the design and usability of the web application. Both the backend and frontend tasks will be split evenly among the different teams.

We will be meeting weekly and updating the other team members on work completed and in progress to avoid any friction or confusion while working on the project. We plan on creating a working document where team members can put down things they were confused about, discussion topics, and the next steps for a given task. This will help structure our meetings and organize our team's thoughts.