

**Team Number:** 202-2

**Team Name:** Probable Hedgehogs

**Members:** Patrick Faulkner, Michelle Tran, Kevin Lawton, John Ortiz, Kai Hueske-VanCeylon

### **Project Feature List:**

- **Account creation:**

- Description: Users create their accounts and fill out the necessary information such as their name, school email, password, subjects, year in school, and their school name.
- Requirements:
  - User can create an account (Functional)
  - User can choose a student and/or tutor role (functional)
  - If user chooses a tutor, then they must choose what year and subject they are (functional)
  - User must create their account with an .edu email (functional)
  - Application warns user when information for account is invalid (functional)
  - Stores user information into a graph (non-functional)

- **Login:**

- Description: User can login using their email and password
- Requirements:
  - User inserts email and password (functional)
  - Users email and password input will be verified by searching for them in our database. (non-functional)

- **Push Notifications for tutor matches and meeting times:**

- Description: When tutors and students get messages or when there is interest, users will get a notification.
- Requirements:
  - Sends tutors/ students a notification when someone is interested (functional)
  - Sends tutors/students a notification when their meeting time is approaching (functional)
  - When a user gets a message, an alert will be sent through or application and received by the other user (Non-functional)

- **Feedback System:**

- Description: Facilitates feedback between students and tutors through a rating system, similar to Uber. Users can also report accounts for various rules violations and other unfavorable events. Users will be able to rate the app as well, and to give feedback for development purposes
- Requirements:
  - Allows students to rate tutors based on their experience (functional)
  - Allows tutors to rate students based on their experience (functional)
  - Allows users to report accounts (functional)

- Allows users to rate the app and give personal feedback on positive/negative aspects (functional)
  - Stores ratings in database, along with their associated user (non-functional)
- **Tutor Filter:**
  - Description: Users w/ student roles will be able to view a list of tutors and filter that list based on subject, rating, year etc.
  - Requirements:
    - User will have a drop down to decide how they will filter (functional)
    - There will be a sort function on the back end (C++) that sorts all information.(Non-functional)
- **Student Filter:**
  - Description: Tutors will be able to access a menu that will be able to narrow down their search as to which students would be a good match for them.
  - Requirements:
    - Users (tutors) can filter out students based on ratings, subject, & other factors (functional)
    - There will be a sort function that will sort through which students will match best with a tutor. Will need to pass in information for both students and tutors. (non-functional)
- **Calendar Integration:**
  - Description: User can view their own personally generated calendar with dates and times for scheduled meetings between tutors and students.
  - Requirements:
    - Automatically schedules meeting after the date and time are chosen (non-functional)
    - Notifies both tutors and students the morning of the meeting (functional)
- **Messaging:**
  - Description: Users can send messages between one another when they match.
  - Requirements:
    - Allows either the student or tutor to begin a chat with the other (Functional)
    - Users can block anyone they choose from messaging them (Functional)
    - Creates a chatroom with the tutor and student (Non-functional)