SW Engineering CSC648-848-05 Spring 2023

DebugMe

Team 02

Emilee Padilla - Team Lead
emann@mail.sfsu.edu
Jijeong Lee - Frontend Lead
Cristobal Padilla - GitHub Lead
Matthew Bush - Backend Lead
Khayotbek Azimov - Cloud Lead
Jeffrey Ma - Database Lead
Hector Magallanes - Document Editor Lead

Milestone 2

03/30/23

History Versions Table

Revision	Date
m2V1	03/31/30
m1V2	
m1V1	02/23/23

Table of Contents

1. Data Definitions

1. **User types:**
    - 1.1. **Unregistered Users:** An unregistered user is a person who accesses DebugMe without creating an account. Unregistered users have limited access to the site and functions compared to registered users.
    - 1.2. **Registered Users:** A registered user is a person who has registered an account on DebugMe by providing an email and creating a password. Registered users have access to more features and functions compared to unregistered users.
        - 1.2.1. **General Users:** A general user is a subcategory of a registered user who has the basic level of access to DebugMe features and functions. General users have less permission than Premium users and Mentors but more permissions than unregistered users.
        - 1.2.2. **Premium Users:** A premium user is a subcategory of a registered user who has paid for access to additional features or content that are not available to general users. Premium users have access to exclusive content such as premium guides at the price of a one time cost or subscription fee depending on the feature.
        - 1.2.3. **Mentors:** Mentors are the authors of DebugMe's Premium Guides. They also provide Premium Users with specialized services, such as resume reviews and mock interviews, during Mentoring Sessions.
        - 1.2.4. **Admin:** Admins handle customer service requests, from reports about inappropriate content to refund requests.
2. **User Profile:** User profiles contain personal information about each registered user, including their profile picture and items related to their activity within the site.
    - 2.1. **Subscription Plan:** Free or Premium plan
    - 2.2. **List of upcoming Mentoring Sessions:** Premium users only.
    - 2.3. **List of upcoming Events:** Events the user is participating in.
    - 2.4. **List of saved Posts**
    - 2.5. **List of saved Premium Guides**
    - 2.6. **Notifications:** For activity related to the services the user participates on. e.g. comments on their posts, for reviews left on Mentors' Premium Guides, etc.
    - 2.7. **Admin user profile:** Admins have an additional section on their profile to manage customer service requests
        - 2.7.1. **List of user reports about inappropriate content:** Admins have the responsibility to remove user made content that violates the site's policies

2.7.2. **List of user refund requests:** Admins process and approve/deny user requests for refunds after unsatisfactory Premium services like Mentoring Sessions

2.8. **Mentor User Profile:** Mentors have an additional section on their profile dedicated to the feedback their Mentees provide

2.8.1. **Mentor Rating:** Value that indicates overall performance of the Mentor

2.8.2. **Mentor Reviews:** Individual comments left by Mentees after meeting with a Mentor

3. **Forum:** Main gathering place where all Registered Users connect with each other through posts and comments.

3.1. **Post:** Registered Users create posts to share ideas or start discussions with the entire DebugMe community.

3.1.1. **Post Comment:** Left by users that wish to engage in the Post discussion.

4. **Premium Guide:** Resources created by Mentors to share their advanced knowledge about internship success. Paid service included in Premium subscription plan.

4.1. **Premium Guide Comment:** Premium users can discuss the contents of Premium Guides by leaving comments on them.

4.2. **Premium Guide Rating:** Premium users can rate Premium Guides based on their quality.

5. **Mentoring Session:** Service for Premium Users to connect with Mentors on one-on-one meetings. Currently, Mentoring Session details like date/time and location (virtual/in person) are determined through direct messages between Mentor-Mentee.

6. **Calendar of Events:** Collection of all Events created by Registered users.

6.1. **Event:** Tool for all Registered users to coordinate meetings (virtual/in person) with other Registered users. The nature of these events must be related to internship help, success.

6.1.1. **Participant List:** List of Registered Users that wish to participate in the Event.

6.1.2. **Event Comment:** For participants to discuss event details, ask questions to the Event creator.

7. **Direct Messaging:** Means of direct communication between any two Registered Users. Any registered user can send a direct message to any other Registered User.

7.1. **Conversation:** Collection of messages between two Registered users

7.1.1. **Message:** Can contain text and/or images

2. Prioritized Functional Requirements

Priority 1

- **Unregistered Users**
    1. Unregistered users shall be able to create an account

- **Registered Users**
    1. Registered users shall be able to delete their accounts
    2. Registered users shall be able to log in
    3. Registered users shall be able to log out
    4. Registered users shall be able to request a password reset
    5. Registered users shall be able to see their profiles
    6. Registered users shall be able to update their personal information
    7. Registered users shall be able to upload a profile picture
    8. Registered users shall be able to change their subscription plans (Free/Premium)
    9. Registered users shall be able to update their financial information
    10. Registered users shall be able to see all posts in the Forum
    11. Registered users shall be able to search Forum posts
    12. Registered users shall be able to create posts
        12.1. Registered users shall be able to add text to their post body
        12.2. Registered users shall be able to add images to their post body
    13. Registered users shall be able to delete their posts
    14. Registered users shall be able to see any post's comments
    15. Registered users shall be able to leave a comment on any post
    16. Registered users shall be able to delete their post comments
    17. Registered users shall be able to report any post
    18. Registered users shall be able to report any post comment
    19. Registered users shall be able to become Mentors
    20. Registered users shall be able to send a direct message to any other registered user
    21. Registered users shall be able to receive direct messages from any other registered user
    22. Registered users shall be able to see all events listed in the calendar
    23. Registered users shall be able to create new events
        23.1. Registered users shall be able to add text to their event body
        23.2. Registered users shall be able to add images to their event body
        23.3. Registered users shall be able to make their event's participant list public or private
    24. Registered users shall be able to edit their events

25. Registered users shall be able to delete their events
26. Registered users shall be able to join any event's participant list
27. Registered users shall be able to leave any event's participant list
28. Registered users shall be able to filter events listed in the calendar
29. Registered users shall be able to receive notifications

- **Premium Users**
  1. Premium users shall be able to see a list of their upcoming Mentoring Sessions
  2. Premium users shall be able to see all Premium Guides
  3. Premium users shall be able to access any Premium Guide
  4. Premium users shall be able to leave a comment on any Premium Guide
  5. Premium users shall be able to rate any Premium Guide
  6. Premium users shall be able to report any Premium Guide
  7. Premium users shall be able to request a Mentoring Session from any Mentor
  8. Premium users shall be able to cancel their Mentoring Session requests
  9. Premium users shall be able to leave a rating on a Mentor's profile
  10. Premium users shall be able to leave a review on a Mentor's profile

- **Mentors**
  1. Mentors shall be able to see their rating
  2. Mentors shall be able to see their reviews
  3. Mentors shall be able to create Premium Guides
  4. Mentors shall be able to add text to their Premium Guides
  5. Mentors shall be able to add images to their Premium Guides
  6. Mentors shall be able to edit their Premium Guides
  7. Mentors shall be able to delete their Premium Guides
  8. Mentors shall be able to reply to comments on their Premium Guides
  9. Mentors shall be able to see their Mentoring Session requests
  10. Mentors shall be able to approve Mentoring Session requests
  11. Mentors shall be able to refuse Mentoring Session requests

- **Admin**
  1. Admins shall be able to see all their customer service requests
     - 1.1. Admins shall be able to receive reports about inappropriate user-made content (Forum posts, Premium Guides, etc.)
     - 1.2. Admins shall be able to receive refund requests for Premium services
  2. Admins shall be able to approve/deny customer requests for refunds
  3. Admins shall be able to process reports about inappropriate content
     - 3.1. Admins shall be able to remove any Forum post
     - 3.2. Admins shall be able to remove any Forum post comment
     - 3.3. Admins shall be able to remove any Calendar Event
     - 3.4. Admins shall be able to remove any Premium Guide

Priority 2

- **Registered Users**
    1. Registered users shall be able to upvote any post
    2. Registered users shall be able to embed images on their post comments
    3. Registered users shall be able to upvote any post comment
    4. Registered users shall be able to edit their post comments
    5. Registered users shall be able to add any post to their list of saved posts
    6. Registered users shall be able to remove any post from their list of saved posts
    7. Registered users shall be able to block any other registered user from messaging them
    8. Registered users shall be able to unblock any other registered user from messaging them
    9. Registered users shall be able to embed images on their direct messages
    10. Registered users shall be able to recover their deleted content (posts, comments, events)

    - **Premium Users**
        1. Premium users shall be able to add any Premium Guide to their list of saved Premium Guides
        2. Premium users shall be able to remove any Premium Guide from their list of saved Premium Guides

    - **Mentors**
        1. Mentors shall be able to embed video content in their Premium Guides
        2. Mentors shall be able to change the font style in their Premium Guides
        3. Mentors shall be able to attach a custom tag to their username
        4. Mentors shall be able to recover their deleted Premium Guides

    - **Admins**
        1. Admins shall be able to temporarily ban registered users
        2. Admins shall be able permanently ban registered users

Priority 3

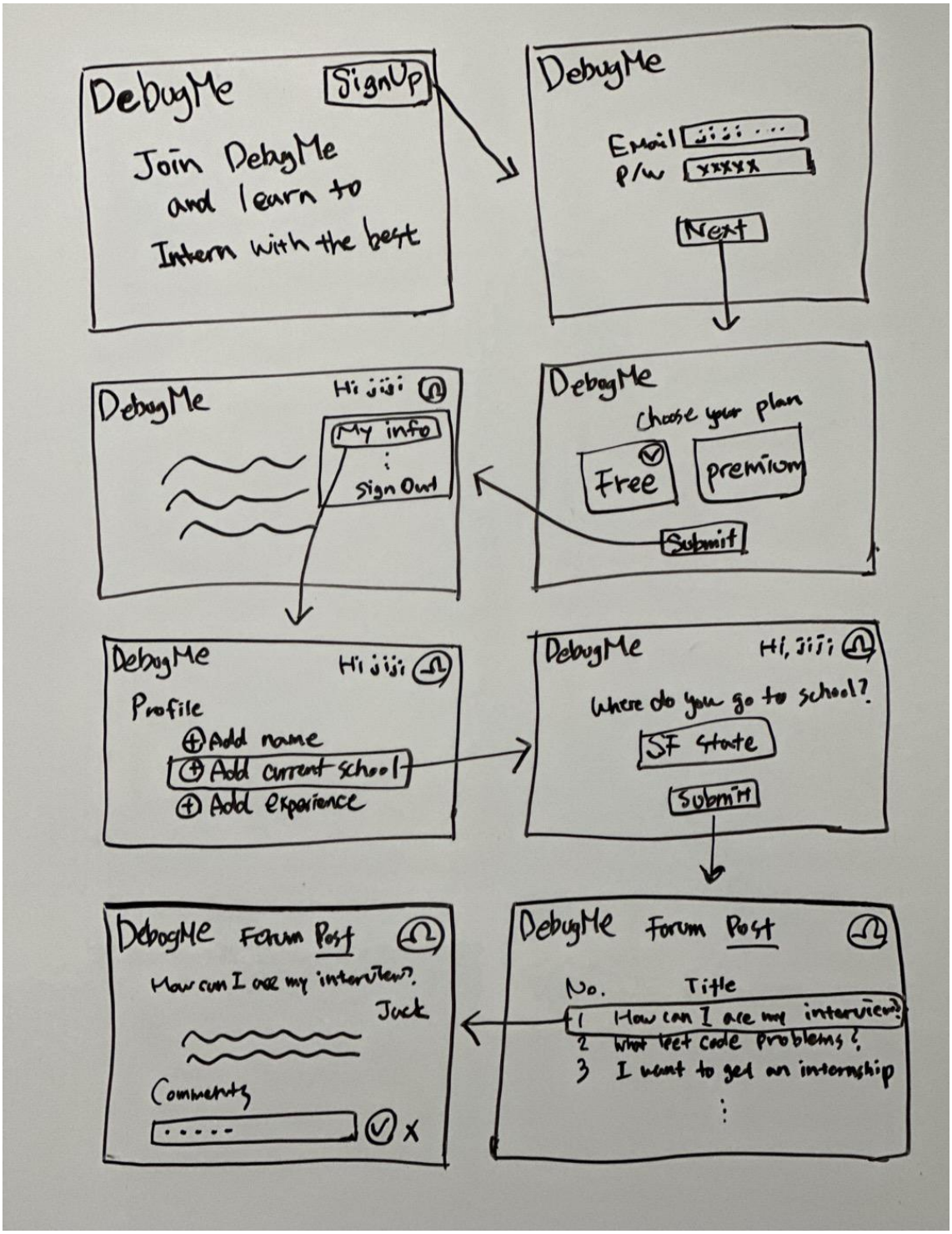
- **Registered Users**
    1. Registered users shall be able to to change DebugMe's theme (Light/Dark)
    2. Registered users shall be able to link their social media accounts to their DebugMe account
    3. Registered users shall have an icon next to their username to display if they're currently online
    4. Registered users shall be able to share DebugMe content through their linked social

media accounts

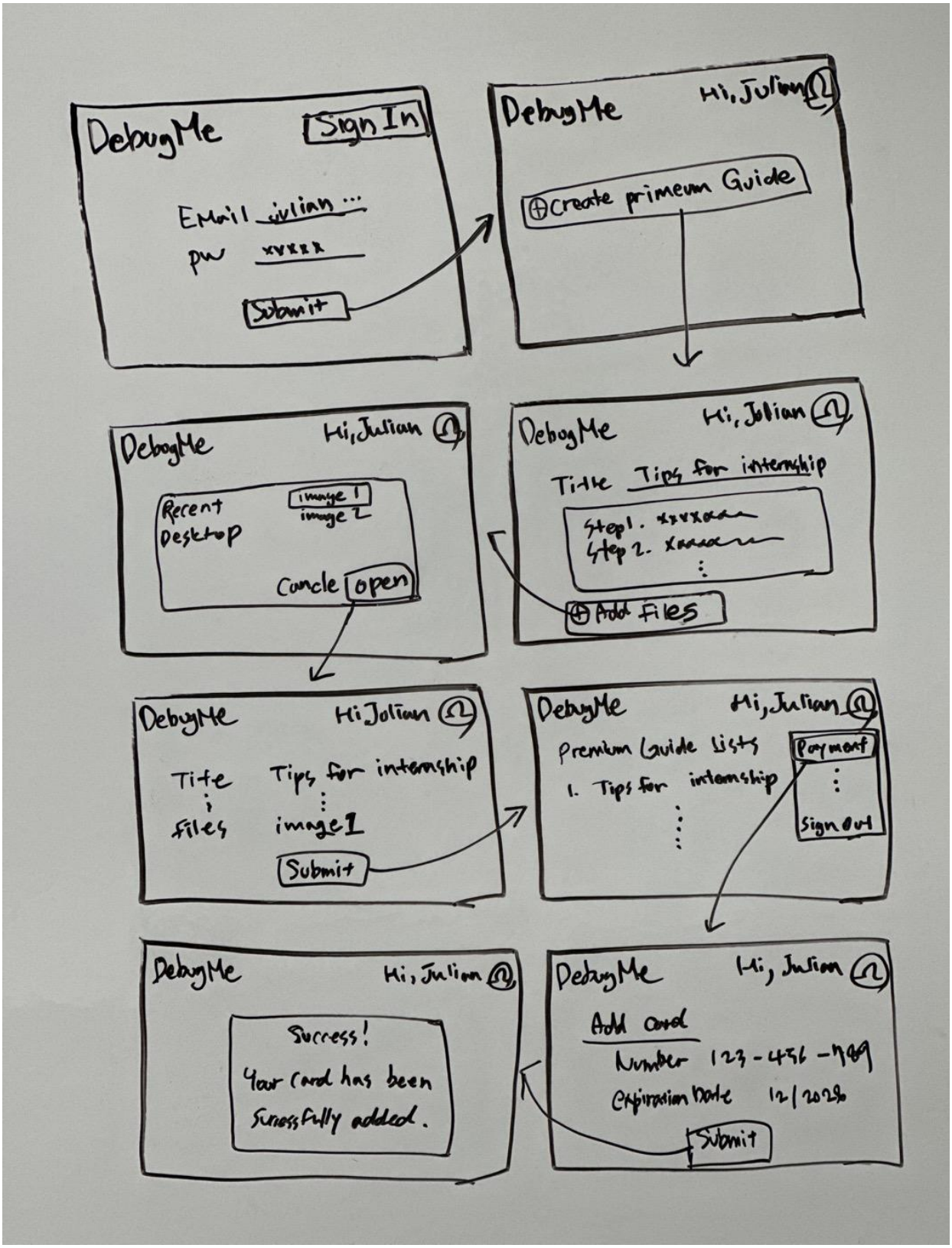
5. Registered users shall be able to see their history of interactions within DebugMe
6. Registered users shall be able to set reminders for upcoming Events or Mentoring Sessions

3. UI Mockups and Storyboards (high level only)

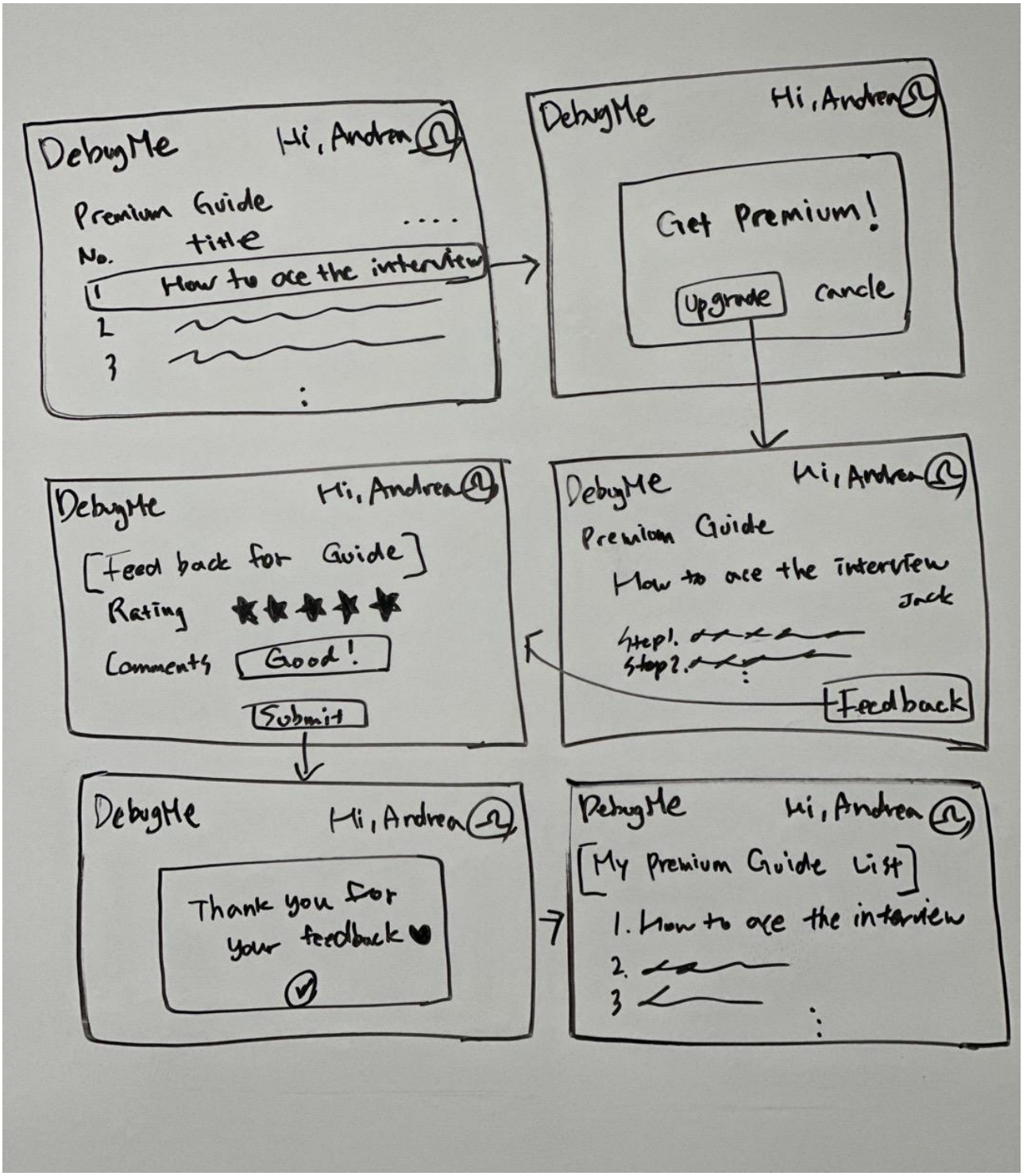
1. Mike visits DebugMe for the first time. He creates an account and updates his personal information. Finally, he interacts with a Forum post by leaving a comment.



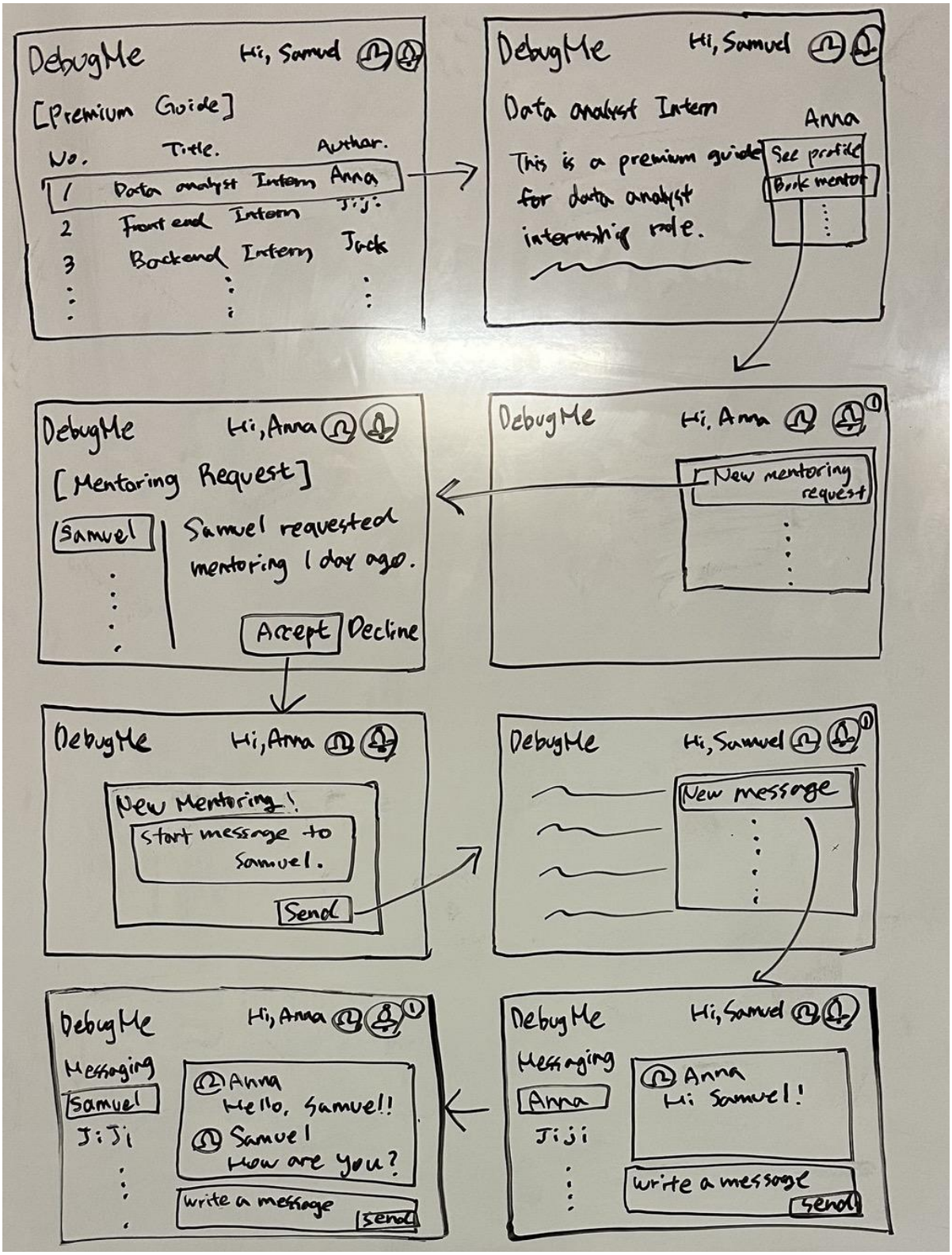
2. Julian logs in and creates a Premium Guide, he adds text and images to the guide body. After submitting his guide, he updates his financial information in his user profile.



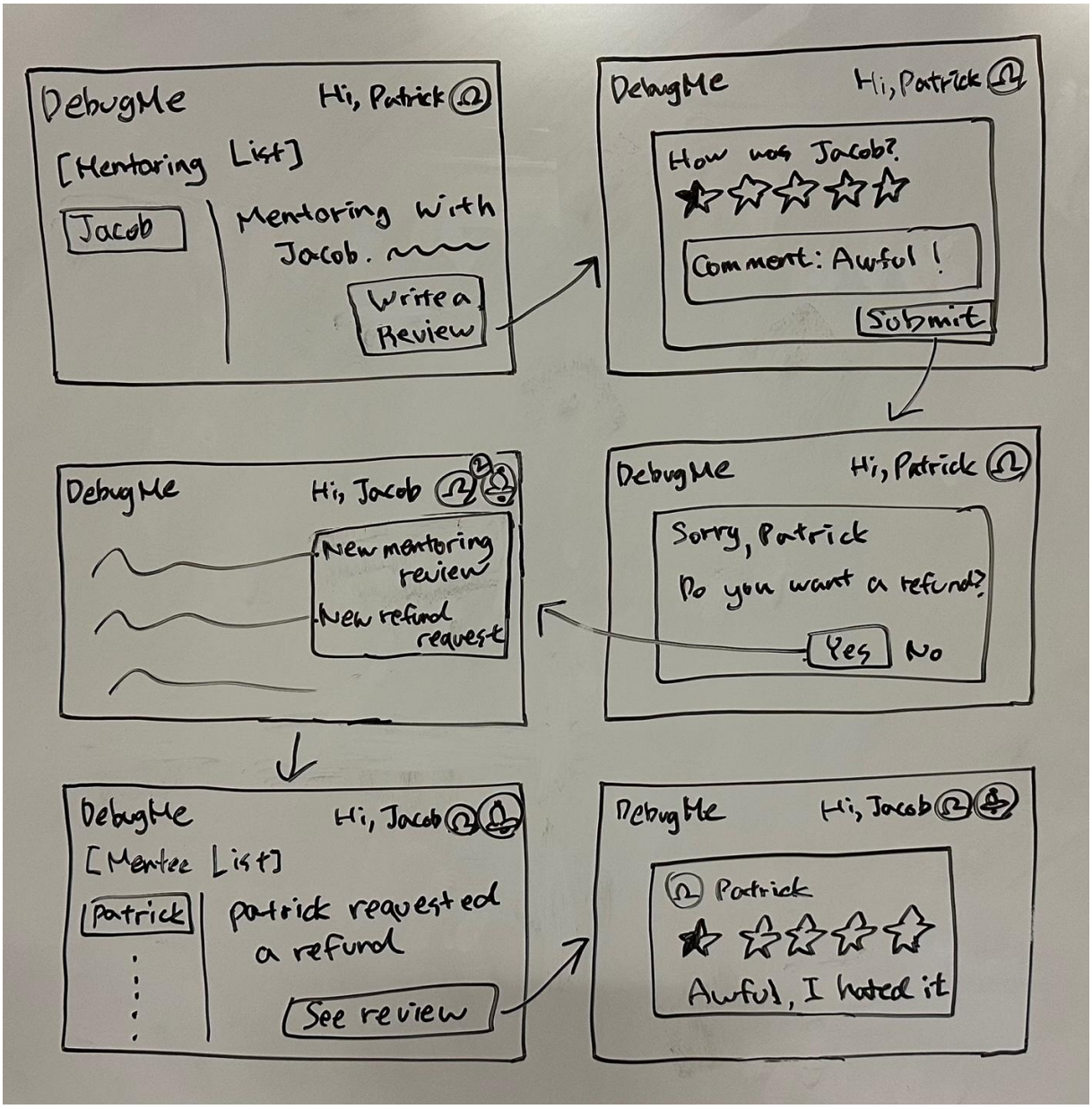
3. Andrea upgrades to the Premium subscription plan. She accesses a Premium Guide and likes it, so she leaves a positive rating and a short feedback comment.



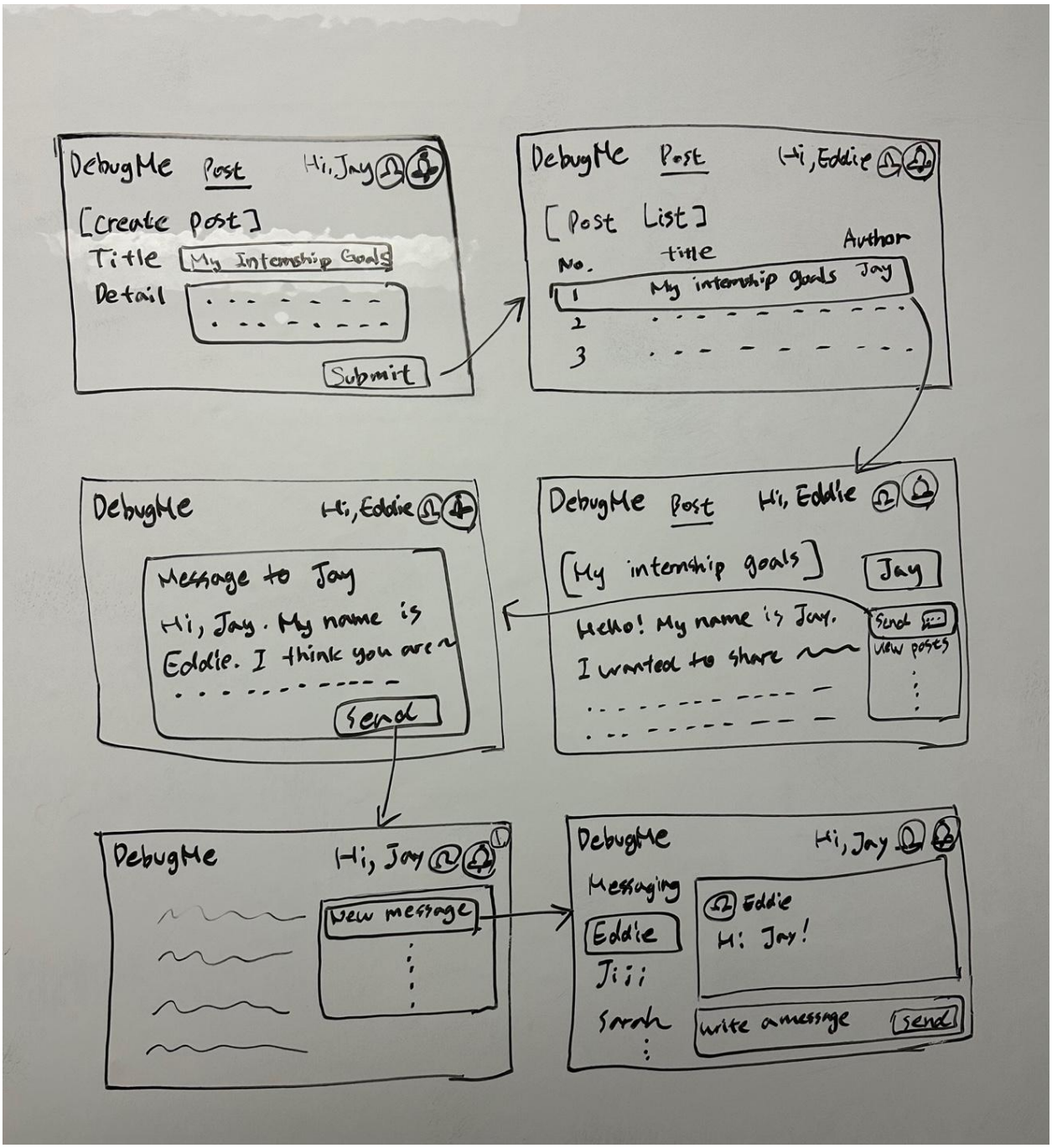
4. Samuel read Anna's Premium Guide and decided to book a Mentoring Session with her. Anna receives and accepts Samuel's mentoring request. They then engage in a direct message conversation to agree on the details of their meeting.



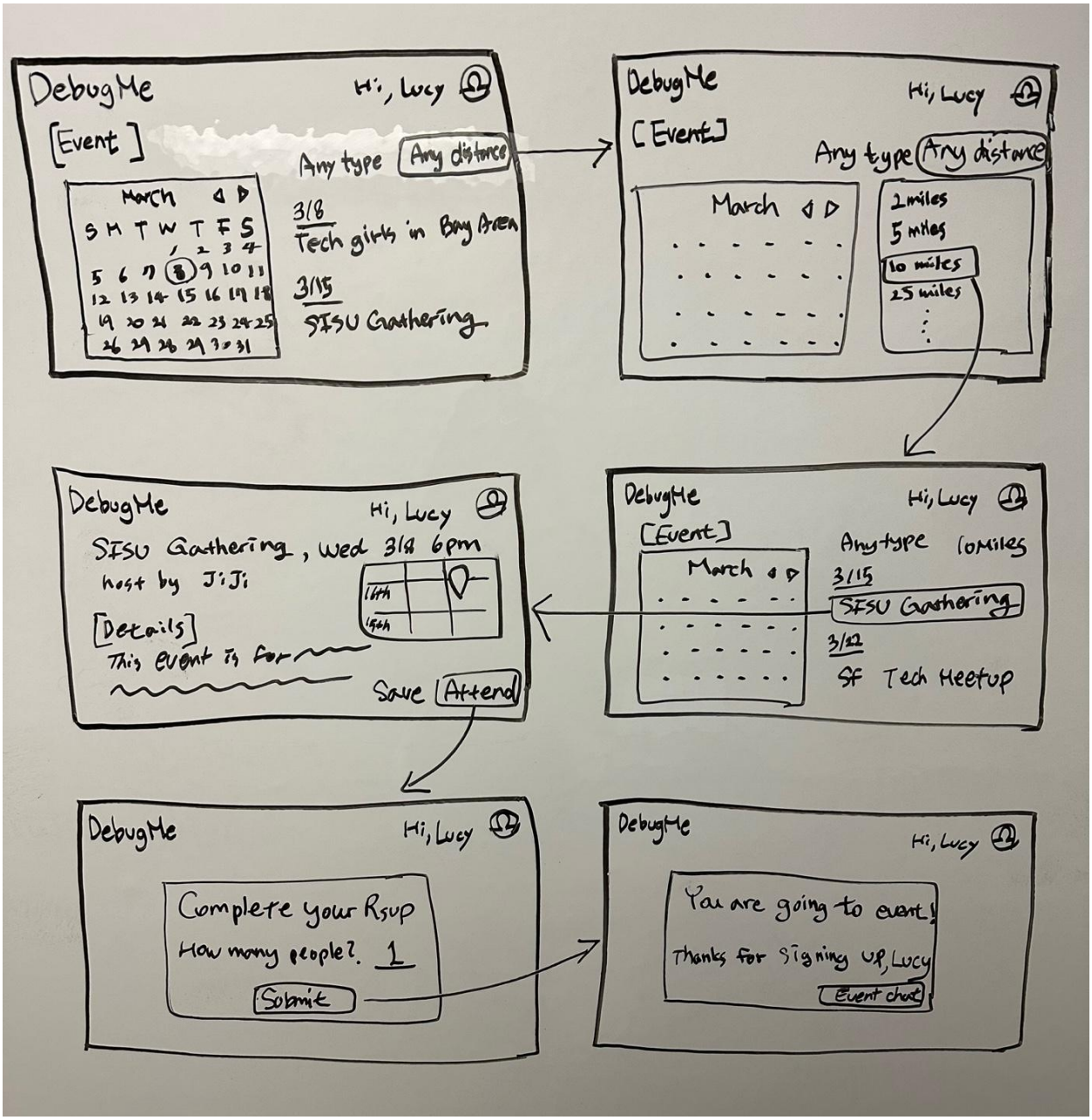
5. Patrick had a negative experience during a Mentoring Session, so he decides to leave a negative review and ask for a refund. Jacob, a DebugMe admin, receives Patrick's refund request and approves it.



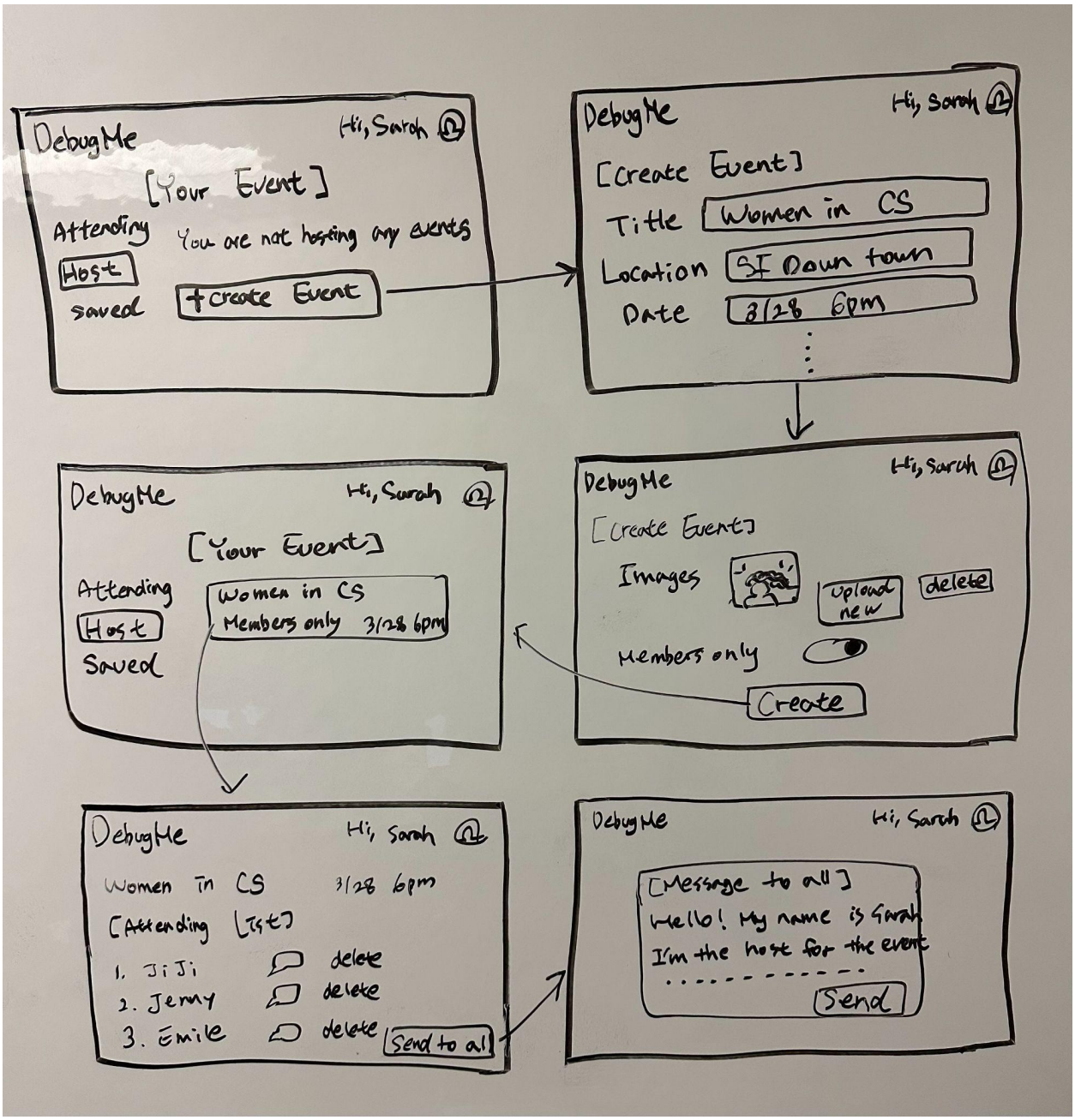
6. Jay creates a new post in the Forum. Eddie sees Jay's post and wishes to talk to him directly. Eddie then starts a direct message conversation with Jay.



7. Lucy filters the events from the Calendar of Events by distance. She then finds an event she is interested in so she joins the event's participants list.



8. Sarah wants to get to know other women that are also looking for internships. She creates a new event, filling out the event description and adding images. She sends a welcome message to other users as they join her event's participant list.



4. High level database architecture and organization

1) 10 Database requirements from Section 2 Part 1:

1. Username - For the login information
2. Password - For the login confirmation
3. Email - To subscribe users to notification when implemented
4. Post - Tracks post by ID so we can differentiate them
5. Post descriptions - Post message so they can have messages
6. Forum - Holds all the posts together, can have many forums with many posts
7. Reply - Holds all the replies to posts
8. Calendar - Holds all the Events on the Calender
9. Calendar Events - Holds all event information such as dates, location, description
10. Event participants - Holds participant list for the Events.

2) Entities and Relationship

User table:
This table stores information about registered users of the system, including their unique ID, name, email, password, rank, and creation timestamp.
id (INT, NOT NULL, AUTO_INCREMENT)
name (VARCHAR(255), NOT NULL)
email (VARCHAR(255), NOT NULL)
password (VARCHAR(255), NOT NULL)
userRank (VARCHAR(255))
created_at (TIMESTAMP, NOT NULL, DEFAULT CURRENT_TIMESTAMP)

Forum table:
This table stores information about forums within the system, including a unique ID, description of the forum, and creation and update timestamps.
id (INT, NOT NULL, AUTO_INCREMENT)
description (TEXT, NOT NULL)
created_at (TIMESTAMP, NOT NULL, DEFAULT CURRENT_TIMESTAMP)
updated_at (TIMESTAMP, NOT NULL, DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP)

Post table:
This table stores information about posts made within a forum, including a unique ID, the content of the post, the ID of the user who made the post, the ID of the forum it belongs to, and creation and update timestamps.
id (INT, NOT NULL, AUTO_INCREMENT)

content (TEXT, NOT NULL)
user_id (INT, NOT NULL)
forum_id (INT, NOT NULL)
created_at (TIMESTAMP, NOT NULL, DEFAULT CURRENT_TIMESTAMP)
updated_at (TIMESTAMP, NOT NULL, DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP)

Reply table:
This table stores information about replies made to a post, including a unique ID, the content of the reply, the ID of the user who made the reply, the ID of the post it belongs to, and creation and update timestamps.
id (INT, NOT NULL, AUTO_INCREMENT)
content (TEXT, NOT NULL)
user_id (INT, NOT NULL)
post_id (INT, NOT NULL)
created_at (TIMESTAMP, NOT NULL, DEFAULT CURRENT_TIMESTAMP)
updated_at (TIMESTAMP, NOT NULL, DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP)

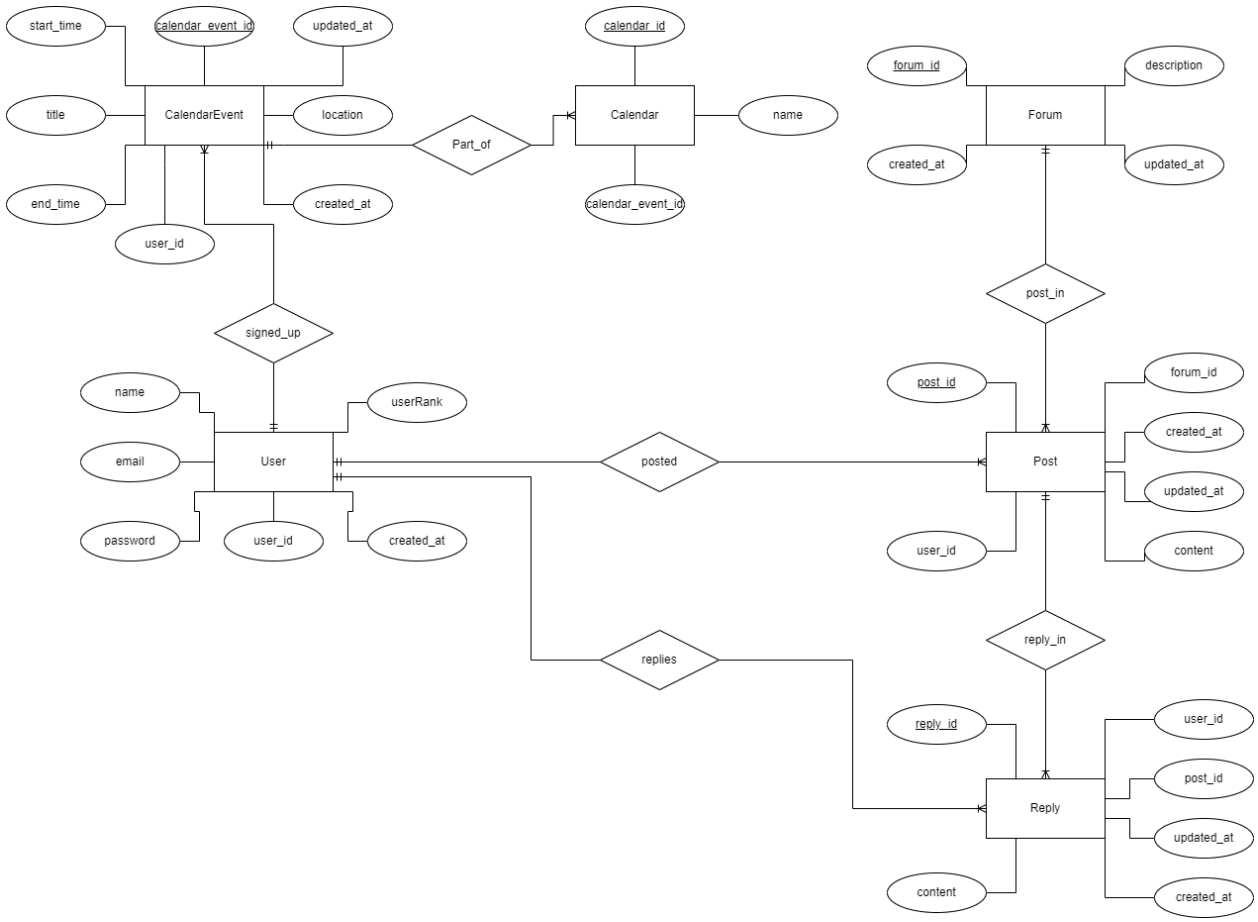
CalendarEvent table:
This table stores information about events on a calendar, including a unique ID, the title of the event, description, start and end times, location, the ID of the user who created the event, and creation and update timestamps.
id (INT, NOT NULL, AUTO_INCREMENT)
title (VARCHAR(255), NOT NULL)
description (TEXT)
start_time (DATETIME, NOT NULL)
end_time (DATETIME)
location (VARCHAR(255))
created_at (TIMESTAMP, NOT NULL, DEFAULT CURRENT_TIMESTAMP)
updated_at (TIMESTAMP, NOT NULL, DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP)
user_id (INT, NOT NULL)

Calendar table:
This table stores information about calendars, including a unique ID and the ID of the calendar event associated with it.
id (INT, NOT NULL, AUTO_INCREMENT)
name (VARCHAR(255), NOT NULL)

Calendar_event_id (INT, NOT NULL)

EventParticipants table:
This table stores information about the participants of a calendar event, including the ID of the event and the ID of the user who is participating in the event. The primary key is a combination of both IDs, to prevent duplicate entries.
event_id (INT, NOT NULL)
user_id (INT, NOT NULL)

3) ER Diagram
Sample ER diagram for DebugMe's Forum and Calendar of Events



4) DBMS
We have decided to use MySQL as the DBMS and will connect to it using MySQL Workbench.

Media storage: Images/Video/Audio will be stored in the file system and the file path to the files will be saved inside the database. Currently the special requirements we have for

Images/Video/Audio would be the standard file types like png, jpeg, m4a, mp3, mp4 only for now. No plans to implement GPS etc if needed we can export it to Google Maps links for location;

Search/filter architecture and implementation:

- Search in specific tables (posts, etc,)
- Serve results based on regular expression matching
- What do we do if there is no matches?
- How many results per search (top 10)

5. High Level APIs and Main Algorithms

- **Major APIs and Non-Trivial Algorithms**
    - **Adding users to the database**
        - This will be done by first checking that all of the fields that are required are filled out. Next we will check that all the fields have properly formatted information. For example we will check that the user email follows the format ^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$ We will also check to make sure that the two passwords that they entered are matching and follow our basic security requirements such as having at least one capital letter and one special character. Lastly when adding the user into the database we will hash their password for extra security.
    - **Making valid forum posts**
        - When a user tries to make a form post we first must make sure that they are a user that has the proper permissions to do so such as being a registered user. Next we check that the post follows are minimum requirements and does not contain any explicit language such as curse words.
    - **Deleting forum posts**
        - Registered users can only delete their own forum posts. In order to ensure this, the system will only allow the option of removal on posts created by the user.
    - **Search for forum posts**
        - When a user searches for posts we will only do keyword searches for the posts content, posts timestamp, form description, form timestamp and user name. This is so we can protect user information.
    - **Adding valid calendar events**
        - This will be done first checking that the user has proper permissions to do so with only registered users and above being able to do so. Next we will need to check that the information is valid such as the start time is before the end time and that the start time is after the current time. Lastly we will need to check that the event follows our basic format and does not contain any explicit language such as curse words.
    - **Registered users filter a calendar event**
        - When a user filters calendar events we will only get the items in the

database that fit the filter value such as only finding events that are on certain dates and times. We will also let the user filter by description, this will search threw the events and return all events that match that keyword

- **Registered users becoming Mentors by creating Premium Guide**
  - Registered users create and upload a guide and then they become mentors, which will alter their privileges within the site and allow them the use of more features.

- **Registered users requesting Mentoring Sessions**
  - When a registered user wants to schedule a mentoring session, the system will send a request for a mentoring session to a mentor either randomly, or to a mentor of his/her own choosing.

- **Registered users rating guides**
  - When a user rates a guide we add their review to the database and also take the average of all user ratings for that guide and then also update the guide's overall rating with the new average rating.

- **Providing Guide feedback**
  - When a user tries to leave feedback on a guide, we first must make sure that they are a user that has the proper permissions to do so such as being a registered user with a Premium account. Next we check that the post follows are minimum requirements and does not contain any explicit language such as curse words.

- **Registered users rating mentors**
  - When a user rates a mentor we add their review to the database and also take the average of all user ratings for that mentor and then also update the mentor's overall rating with the new average rating.

- **Payment Processing**
  - Stripe provides a great api for processing payments. It can accept dozens of payment methods in more than 100 currencies. Stripe also provides advanced developer tools. With Stripe, a user can pay for the premium subscription.

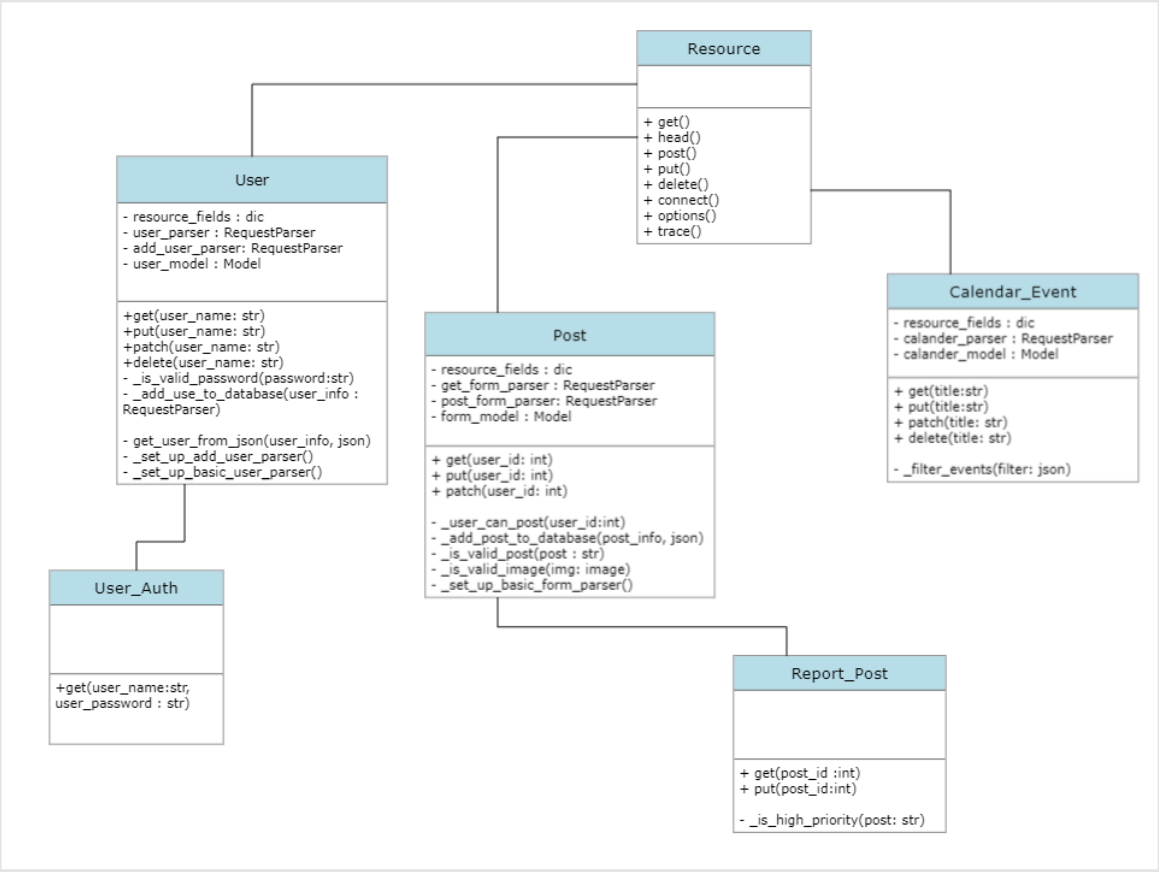
- **Upgrading user’s account from basic to premium**
  - When a user with a basic account upgrades to a premium account, the payment will be processed by the Stripe api. Once the payment clears, the user will get access to all the features available to premium account users.

- **Processing Refunds, or Canceling Payments**

        - When a refund, or a payment cancellation, is required, the system will use Stripe's api to perform the task.

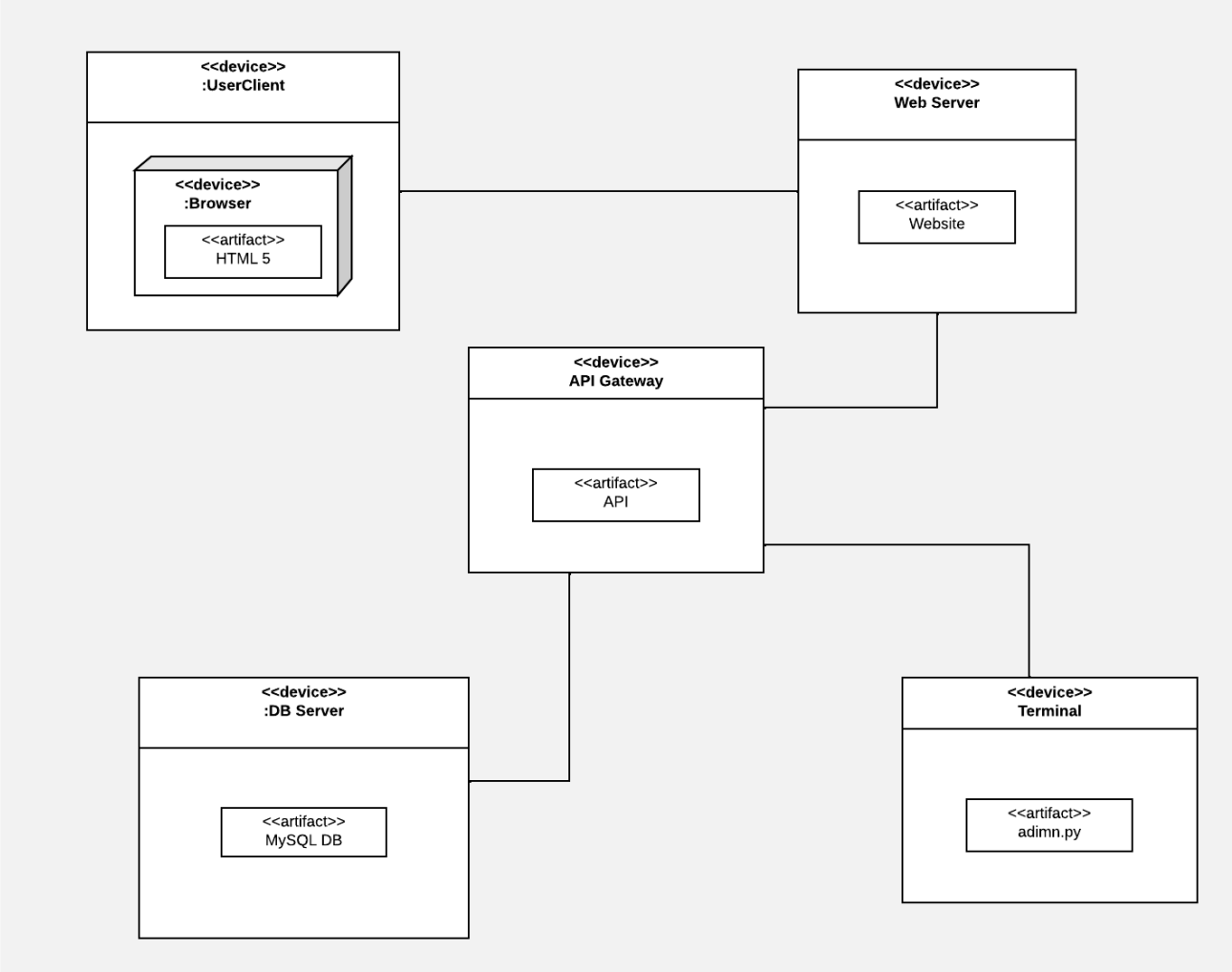
- **Direct Messaging (Chat)**
    - Registered users can chat with other registered users. Each chat session will be unique. Once a chat between two registered users is created, the chat will be reused for the same two users. Chat sessions between users are stored in the DB so users can view previous communication sessions.

6. High Level UML Diagrams

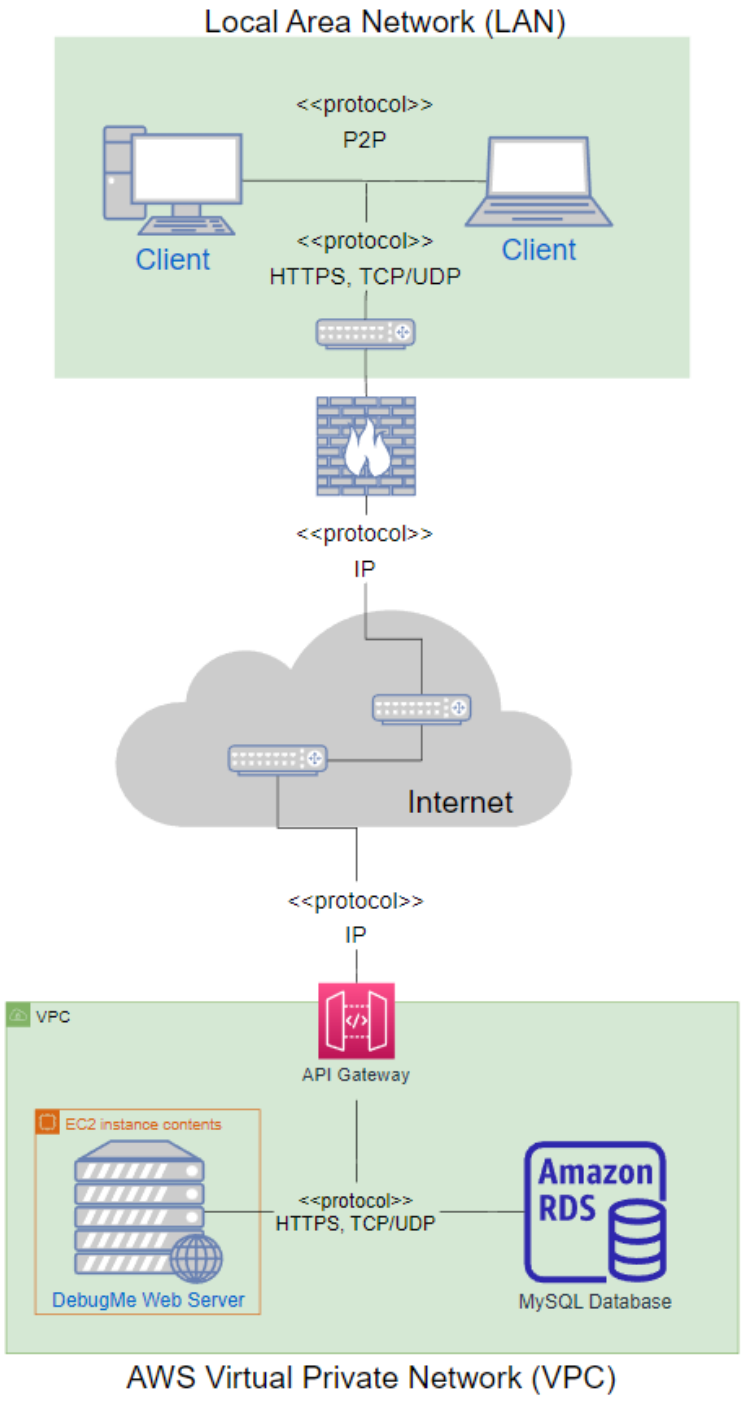


7. High Level Application Network and Deployment Diagrams

Deployment Diagram



Network Diagram



8 .Identify *actual* key risks for your project at this time

- Schedule Risks:
    - We had just changed our meeting structure to shorter meetings and were also facing an upcoming Spring Break. So, we were worried about having enough time to work together to work on the project, especially since Milestone #2 seemed so overwhelming. The Professor kept describing that Milestone #2 was going to be 10 times harder than Milestone #1 was. So we were very nervous about how all of this would play out together.
    - The way we planned to overcome this risk was by really good constant communication and check-ins through Discord. Also, we all talked about this together as a group right away to make it clear that when we did have time to work together that it really needed to be focused and used wisely. Everyone seemed to agree and understand the gravity of the situation.
- Time Management/Organization Risks:
    - One very real risk is that we had a hard time getting going and finding our "stride" together as a group in Milestone #1 which in turn caused us to turn it in very late with the help of an extension. Also, to expand on this, we felt that we were not working very efficiently.
    - The Team Leader took it on to really organize a Trello board for us to use and work off of in order to help us work more efficiently as a team. This so far seems to be a very effective and helpful resolution and has definitely increased our working efficiency as a team.
- Skills Risks:
    - We had a few aspects that we were concerned about, specifically, we were concerned if we had enough knowledge to work on the parts relating to Computer Networks and Databases.
    - The solution to this was communicating this openly as a team and we had several classmates that are taking those exact classes and they felt confident through those other classes that they would have the resources and skills to help guide us as a group.
- Technical Risks:
    - We were worried if we had made the right choices in regards to our stack in Milestone #1 and how that would affect us when working on Milestone #2.
    - By implementing the Vertical Prototype we were able to test that our chosen stack was the right one.
- Teamwork Risks:
    - We were worried about having everyone participate more consistently and in particular did not want to have team members no show to meetings without communication.
    - The way we handled this was by communicating more with each other so that

we could adjust meeting times, wait for team members and or catch them up afterwards if they truly could not make it. This really improved and therefore helped us work better together as a team.

- Legal/Content Risks:
    - The progress of our project up to Milestone 2 has not encountered any legal liabilities.

9. Project management

For Milestone #2 I was very excited and re-invigorated to get to work in a better and more efficient manner than was experienced in Milestone #1. The tool I decided to use to help me achieve more efficient project management is Trello. The first thing that I did was read through the new directions once they were available and take notes on them. I also took very detailed notes in class when we went over the directions. I then utilized both of these actions by inputting all of the instructions and my notes into tasks, etc on Trello. I had a pretty good idea of who should be assigned to what but I wanted to make sure that my team was on board with me and had them look everything over on their own and then assign themselves to the tasks that they thought were best suited to their roles as well as skill level. After this, I had everyone check again that they felt comfortable with what they needed to work on. On Trello, I used the color feature to make it visually easy to sort through by assigning each task a different color. I also came up with a plan that by being so detailed on Trello it would be easier for me to keep track of the team member's contributions, etc. I would say overall, these new approaches worked very well and we definitely seem to have progressed as a group in terms of work efficiency and communication.

I am really happy with the improvement in the project management, however, I definitely have several takeaways to further improve upon for the next Milestone. I want to make sure that I do more one-on-one check-ins with team members, possibly once a week to check in on how they are doing with their task as well as personally if they feel inclined so that we can maintain an even more efficient working pace, etc. I would also like to post meeting notes at the end of every meeting on Discord so that everyone has access to our series of events as well as who shows up and not so that it is "public knowledge" so to speak for the group and everyone can

keep better track collectively. Finally, I would like to create a simple meeting outline before each meeting so that individually and as a group we are more prepared for what we want to go over and really utilize our meetings much more efficiently. I am really looking forward to implementing these new ideas for the next milestone. I really believe they will encourage even more positive work within our team.

10. Detailed list of contributions

1. Jiji/Jijeong Lee (Frontend Leader) - Score: 9

- Made UI Mockups relating to Use cases
- Used all the entities from Functional, Data Entries, and Use cases
- Used consistent naming
- Walked us through all the mockups and storyboards
- Worked with Backend to agree on common interfaces
- Worked with Backend to establish rules for CSS and UI development
- Worked with Backend to agree on a common way to connect UI with Backend (this is documented?)
- Tested AWS with Backend to make sure it's running correctly
- Has participated in all the meetings she attended, even when sick
- Communicated very well about everything she was involved in

2. Mat/Matthew Bush (Backend Leader) - Score: 9

- Made a testing branch and integrated with Mysql
- Worked on our special feature through High level API and main algorithms
- Described and defined major API’s that were created
- Described any significant process
- Described changes to SW tools and or frameworks after getting it approved
- Used competitive analysis to create high level UML diagrams
- Created App Network Diagram (AND)
- Created Deployment Diagram (DD)

- Worked with Frontend to agree on common interfaces
- Worked with Frontend to establish rules for CSS and UI development
- Worked with Frontend to agree on a common way to connect UI with Backend (this is documented?)
- Tested AWS with Backend to make sure it's running correctly
- Has really participated in attended meetings and helped with brainstorming

3. Cris/Cristobal Padilla (Github Leader) - Score: 9

- Worked on our special feature through High level API and main algorithms
- Described and defined major API's that were created
- Described any significant process
- Described changes to SW tools and or frameworks after getting it approved
- Used competitive analysis to create high level UML diagrams
- Has attended all meetings and been a very active and constructive participant in all of them

4. Jef/Jeffrey Ma (Database Leader) - Score: 9

- Created a high level conceptual database design based on one of our unique features
- Used at least 10 functional requirements to create database requirements
- Clearly defined the 10 database requirements
- Described the entities, their attributes, relationship, and domains at a high level
- Created an Entity Relationship Diagram (ERD) with a software tool

- In one sentence defined which DBMS will be used to create the Database and why
- Lead the team in deciding if images and video/audio will be kept in file systems or in DB BLOB's
- Described any special data format requirements for video/audio/GPS, etc
- Worked on creating the algorithm/SW for the search bar and how it is organized, what DB terms will be searched, how it will be coded and how it will be organized in the DB
- Has attended all meetings and been a very active and constructive participant in all of them

5. Hayat/Khayotbek Azimov (Cloud Leader) - Score: 7

- Defined major data items
- Listed formats and max size for images and videos
- Worked on unique and important data items for the application
- Used competitive analysis to work on data definitions
- Fully defined user privileges and registration info and other main info
- Worked on Priority List #1
- Worked on Priority List #2
- Worked on Priority List #3
- Expanded functional requirements from Milestone #1
- Has participated in most attended meetings

6. Hector Magallanes (Document Leader) - Score: 10

- Defined major data items
- Worked on unique and important data items for the application
- Used competitive analysis to work on data definitions
- Fully defined user privileges and registration info and other main info
- Worked on Priority List #1
- Worked on Priority List #2
- Worked on Priority List #3
- Expanded functional requirements from Milestone #1
- Formatted all of Step #1: Data Definitions
- Formatted all of Step #2: Prioritized Functional Requirements
- Created the document history versions table
- Created the document table of contents
- Edited and formatted the entire final version document
- Helped with Computer Networks aspects of Milestone #2
- Has attended all meetings and been a very active and constructive participant in all of them