Group 8 - Final Project Report

Submitted By

64130500231	Supasek	Dhanabordeephat
64130500202	Kyaw Swar	Hein
64130500206	Kantitat	Warawut
64130500265	Phongsaphak	Fongsamut
64130500255	Khush	Agarwal
63130500228	Sukanya	Chinwicha

Present

Assoc. Prof. Dr. VAJIRASAK VANIJJA
Course CSC234 User Centered Mobile Application Development

Semester 2, Academic Year 2022

Table of Contents

Concept and list of functions

WisMod is a mobile application that aims to connect people among various faculties to unite together in order to do something EPIC!!!. The goal of WisMod is to provide a platform for university students to find a group with the same interests to complete some particular tasks which require multi-disciplinary skills.

Our objectives for developing this application, includes:

1. Increase student engagement and create a student community where students and staff of KMUTT can find a partner on some tasks.
2. Accessible to all faculties and allows for inter-faculty communication
3. To create the platform for exchanging ideas and discussing in creative ways (e.g., discuss PM2.5 solutions among chemistry students and CS students, etc.)

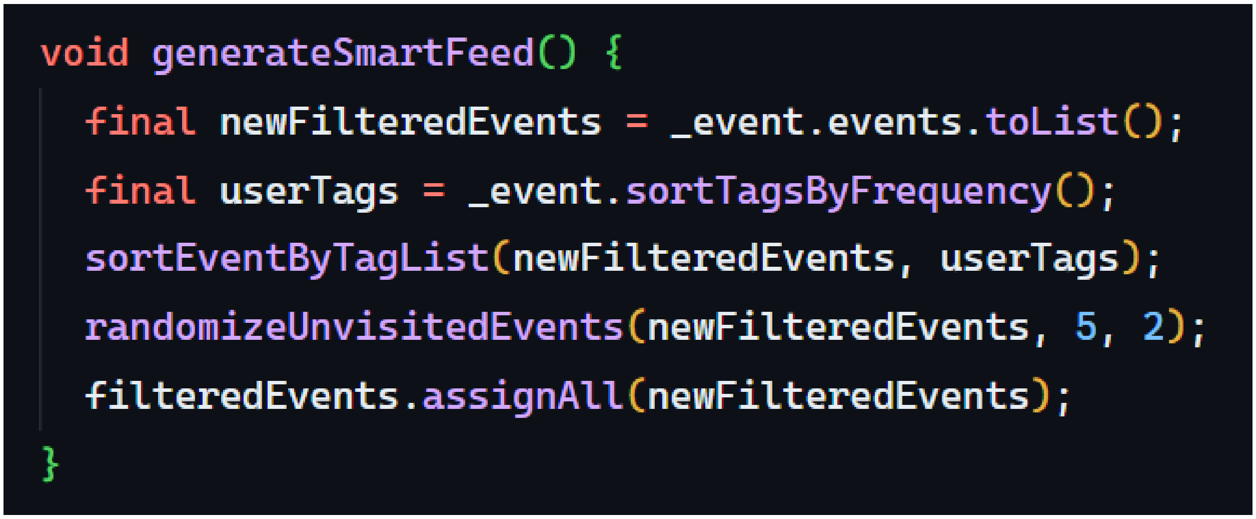
WisMod's core functionality works by allowing its users to create 'Events' or 'Activities' in the application which can be joined by other users and provide a convenient chatting interface in order for users to communicate with each other. Users who create the events in the application are referred to as 'Event Owners' and the users who join the event are referred to as 'Event Participants'. There are also 'Admin' who have the authority to remove inappropriate events.

Functional Requirements:

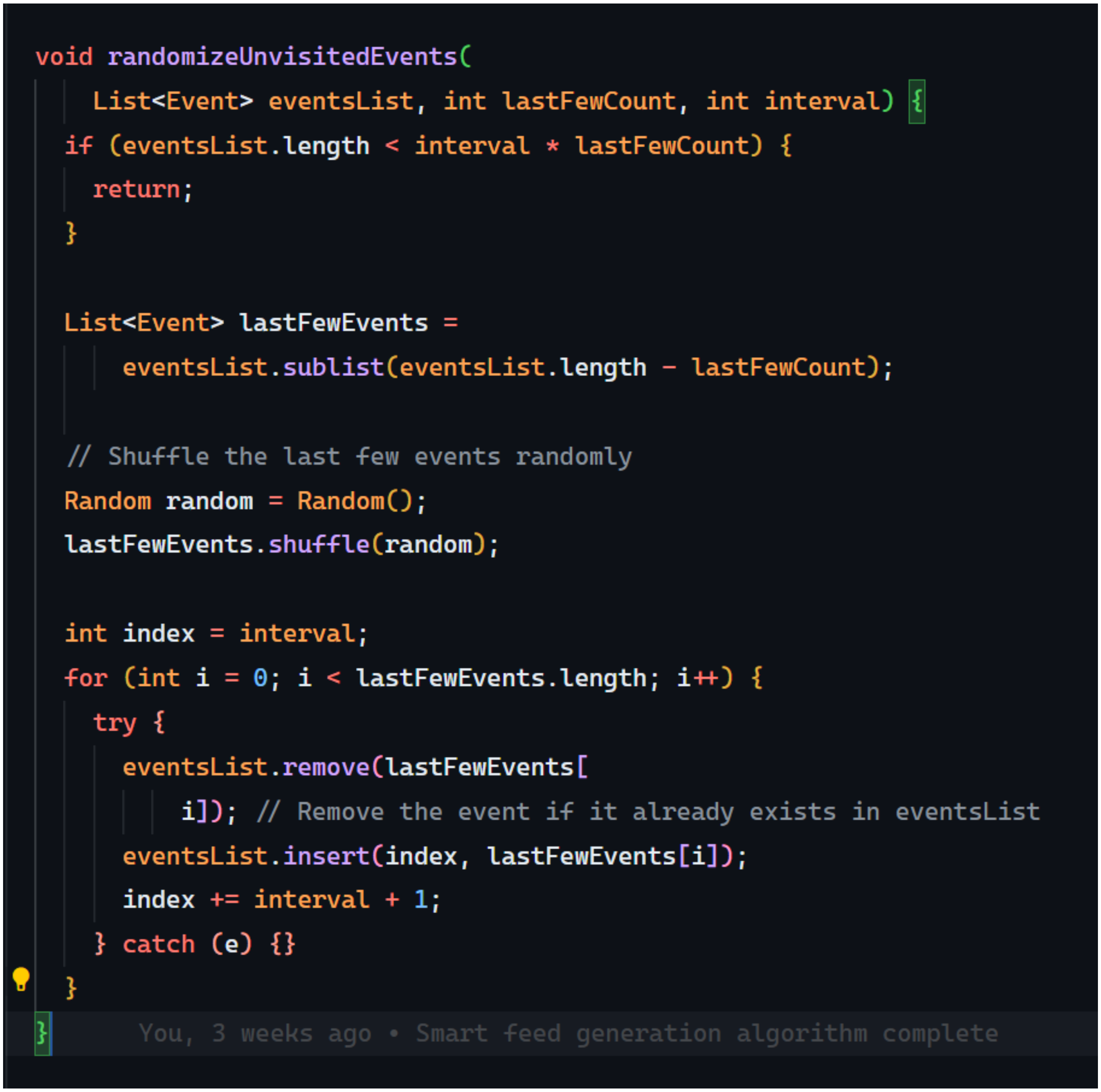
- Notification:
  - Users should receive notifications for relevant user interactions, including the creation of new events and received chat messages.
- Event Publishing:
  - Users should be able to create events within predefined categories.
  - Users should have the ability to edit or delete events they have created if needed.
- Authentication:
  - The application should authenticate user login and registration data using Firebase authentication or a similar mechanism.
- Edit Profile:
  - Users should be able to make changes to their profile information, including profile picture, name, department, year, and password.
- Chat:
  - Event owners and participants should be able to communicate with each other through a group chat function.
- Browsing, Reaction, and Join:
  - Users should be able to filter specific events based on their interests.
  - Users should have the ability to upvote and join other events.
- Reporting:
  - Users should be able to report inappropriate events or illegal activities to the administrators for appropriate action.

- Request Participant Approval:
    - Users hosting events should have the option to disable automatic joining and require participants to send a join request.
    - Event owners should be able to approve or reject join requests as they see fit.
- Filtering and Search:
    - Users should be able to search for specific events by typing keywords in the search bar or filter events based on category and date.
- Blocking Chats:
    - Users should have the ability to block any chat they do not wish to see
- Removing events:
    - Admins should be able to remove any event they deem inappropriate
- Smart Feed:
    - Users should be presented with events in the feed page based on a smart algorithm that prioritizes events relevant to them.

Actual Smart Feed implementation:



```
void generateSmartFeed() {
  final newFilteredEvents = _event.events.toList();
  final userTags = _event.sortTagsByFrequency();
  sortEventByTagList(newFilteredEvents, userTags);
  randomizeUnvisitedEvents(newFilteredEvents, 5, 2);
  filteredEvents.assignAll(newFilteredEvents);
}
```

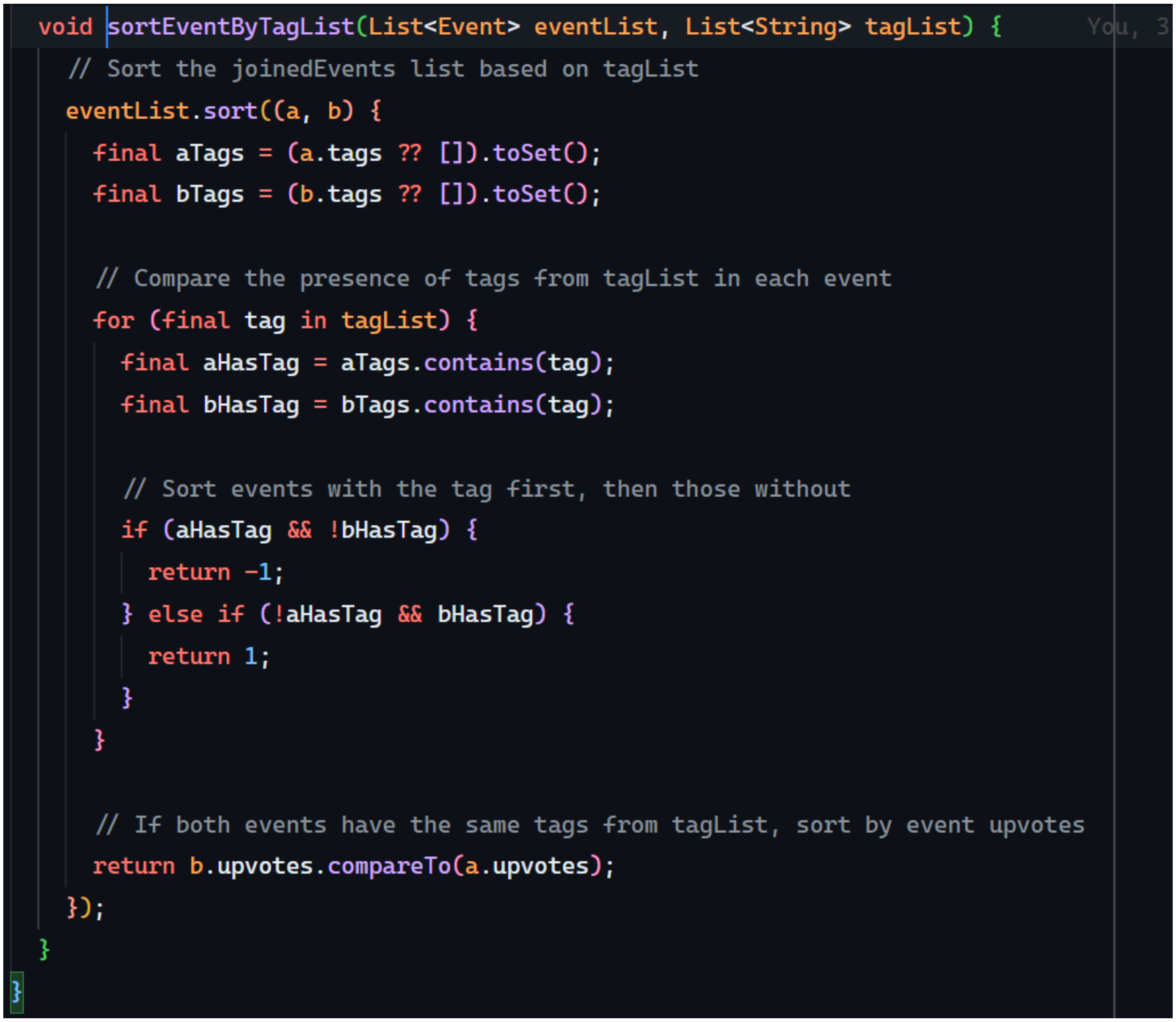


```
void randomizeUnvisitedEvents(
    List<Event> eventsList, int lastFewCount, int interval) {
  if (eventsList.length < interval * lastFewCount) {
    return;
  }

  List<Event> lastFewEvents =
      eventsList.sublist(eventsList.length - lastFewCount);

  // Shuffle the last few events randomly
  Random random = Random();
  lastFewEvents.shuffle(random);

  int index = interval;
  for (int i = 0; i < lastFewEvents.length; i++) {
    try {
      eventsList.remove(lastFewEvents[
          i]); // Remove the event if it already exists in eventsList
      eventsList.insert(index, lastFewEvents[i]);
      index += interval + 1;
    } catch (e) {}
  }
}      You, 3 weeks ago • Smart feed generation algorithm complete
```

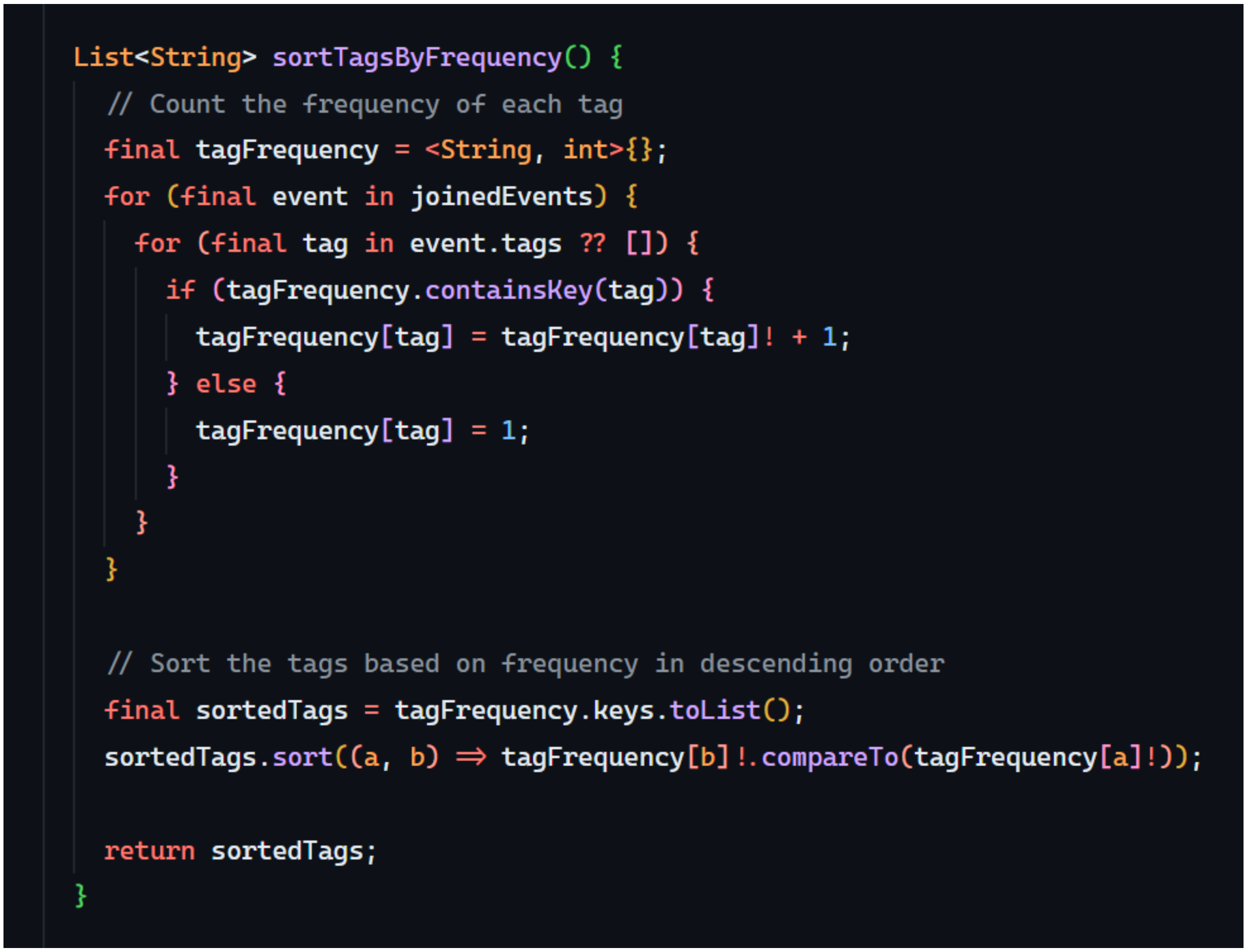


```
  void sortEventByTagList(List<Event> eventList, List<String> tagList) {
    // Sort the joinedEvents list based on tagList
    eventList.sort((a, b) {
      final aTags = (a.tags ?? []).toSet();
      final bTags = (b.tags ?? []).toSet();

      // Compare the presence of tags from tagList in each event
      for (final tag in tagList) {
        final aHasTag = aTags.contains(tag);
        final bHasTag = bTags.contains(tag);

        // Sort events with the tag first, then those without
        if (aHasTag && !bHasTag) {
          return -1;
        } else if (!aHasTag && bHasTag) {
          return 1;
        }
      }

      // If both events have the same tags from tagList, sort by event upvotes
      return b.upvotes.compareTo(a.upvotes);
    });
  }
}
```



```dart
List<String> sortTagsByFrequency() {
  // Count the frequency of each tag
  final tagFrequency = <String, int>{};
  for (final event in joinedEvents) {
    for (final tag in event.tags ?? []) {
      if (tagFrequency.containsKey(tag)) {
        tagFrequency[tag] = tagFrequency[tag]! + 1;
      } else {
        tagFrequency[tag] = 1;
      }
    }
  }

  // Sort the tags based on frequency in descending order
  final sortedTags = tagFrequency.keys.toList();
  sortedTags.sort((a, b) => tagFrequency[b]!.compareTo(tagFrequency[a]!));

  return sortedTags;
}
```

Persona

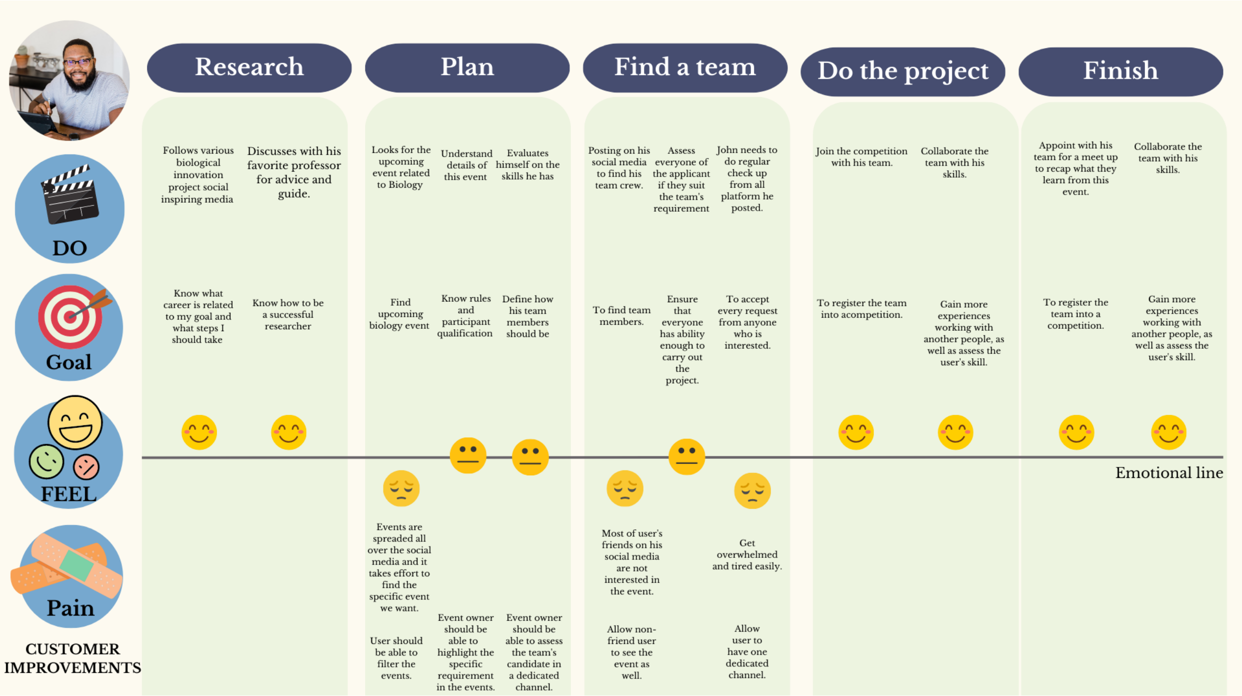
There are 2 personas for target user representation, whose names are John and Jane. John takes the role of ‘Event Owner’ and Jane takes the role of ‘Event Participant’.

1. John
    a. Background : He is a freshman student at KMUTT who is studying in the Biology faculty. He enjoys his studies and is dedicated to learning new things and making friends.
    b. Goals and Needs :
        i. Know more people who are interested in his target hackathon, bio hackathon
        ii. Meet with people from other department who have skills for a hackathon
        iii. Get help from other students to improve his skills
    c. Problems :
        i. Limited platform options for collaboration
        ii. Difficulty Finding People with Relevant Skills
        iii. Limited reach: Hard to find people outside the department.
        iv. Time spent to know someone outside department
    d. Challenges :
        i. Does not know anyone in other faculties.
        ii. Does not have any knowledge about engineering at all.
        iii. Other students are too shy to participate or help him.
        iv. Needs to balance his event planning with his coursework and other obligations.
2. Jane
    a. Background : She is a KMUTT student who wants to expand her friend circle and apply her knowledge somewhere while learning at KMUTT
    b. Goals and Needs :
        i. Improve skills by joining multidisciplinary events
        ii. Find events and make more friends and connection
        iii. Save time and cost to find events and join them
    c. Problems :
        i. Limited and hard to find opportunities outside the classroom.
        ii. Lack of Confidence or shyness when communicating verbally
    d. Challenges :
        i. Doesn't know how and where to find competitions
        ii. Doesn't have a strong network or connection
        iii. Have some skills but nowhere to show

User Journey Map

Below diagrams are user journey maps of John and Jane, which illustrate the steps and emotions when finding friends with similar interests.

John’s user journey map

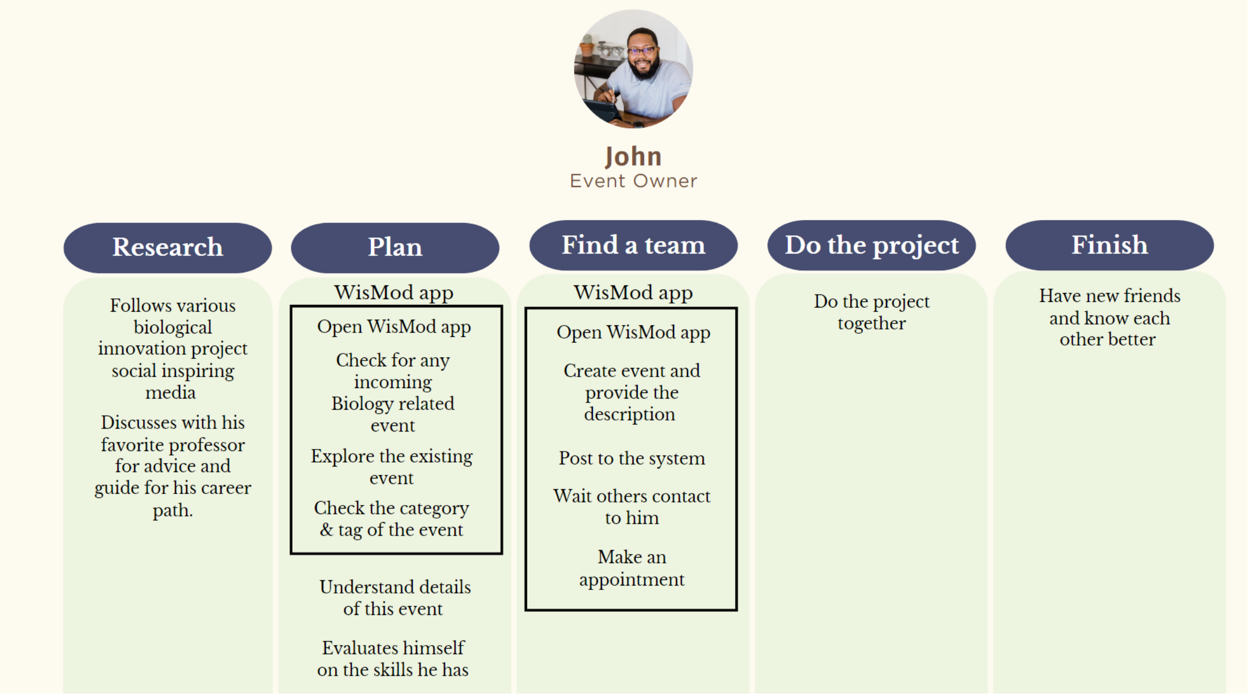


	Research		Plan			Find a team			Do the project		Finish	
DO	Follows various biological innovation project social inspiring media	Discusses with his favorite professor for advice and guide.	Looks for the upcoming event related to Biology	Understand details of this event	Evaluates himself on the skills he has	Posting on his social media to find his team crew.	Assess everyone of the applicant if they suit the team's requirement	John needs to do regular check up from all platform he posted.	Join the competition with his team.	Collaborate the team with his skills.	Appoint with his team for a meet up to recap what they learn from this event.	Collaborate the team with his skills.
Goal	Know what career is related to my goal and what steps I should take	Know how to be a successful researcher	Find upcoming biology event	Know rules and participant qualification	Define how his team members should be	To find team members.	Ensure that everyone has ability enough to carry out the project.	To accept every request from anyone who is interested.	To register the team into acompetition.	Gain more experiences working with another people, as well as assess the user's skill.	To register the team into a competition.	Gain more experiences working with another people, as well as assess the user's skill.
FEEL	😊	😊	😔	😐	😐	😔	😐	😔	😊	😊	😊	😊
Pain			Events are spreaded all over the social media and it takes effort to find the specific event we want.			Most of user's friends on his social media are not interested in the event.		Get overwhelmed and tired easily.				
CUSTOMER IMPROVEMENTS			User should be able to filter the events.	Event owner should be able to highlight the specific requirement in the events.	Event owner should be able to assess the team's candidate in a dedicated channel.	Allow non-friend user to see the event as well.		Allow user to have one dedicated channel.				

https://drive.google.com/file/d/1bk204TMkaITkw5piArdpqxFqa0i6jqO/view?usp=sharing

John’s user journey map is divided into 5 phases. Overall, there are quite a lot of negative feelings from John especially in the planning and finding a team phase.

This is John’s To-Be-Map:



John
Event Owner

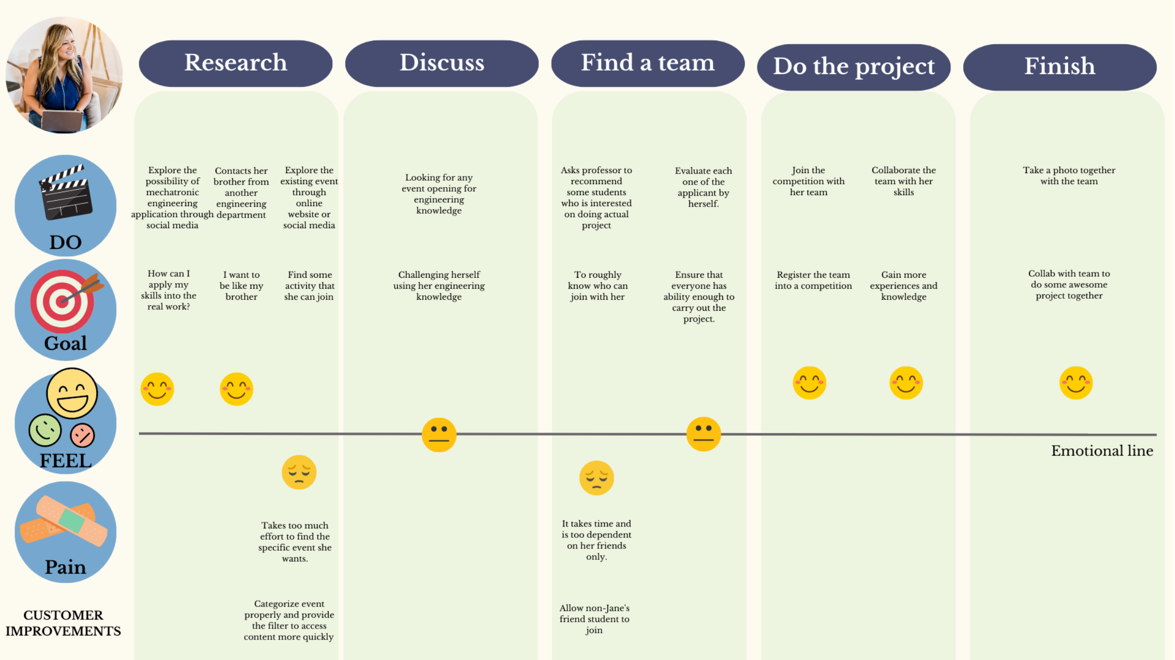
Research	Plan	Find a team	Do the project	Finish
Follows various biological innovation project social inspiring media Discusses with his favorite professor for advice and guide for his career path.	WisMod app Open WisMod app Check for any incoming Biology related event Explore the existing event Check the category & tag of the event Understand details of this event Evaluates himself on the skills he has	WisMod app Open WisMod app Create event and provide the description Post to the system Wait others contact to him Make an appointment	Do the project together	Have new friends and know each other better

https://drive.google.com/file/d/1ieilQgzj7s7rYOOoXOMf5a7qtdwZENW-/view?usp=sharing

Therefore, we propose two main features which can relieve finder's pains in our application. To illustrate more,

1. Filtering events, which can reduce the effort to access the specific events users want
2. Creating Events, Anybody can see the event which John creates which means he can reach out to more people, additionally, users will be provided with the functionality to join the event and chat with John easily through the application itself.

Jane's user journey map

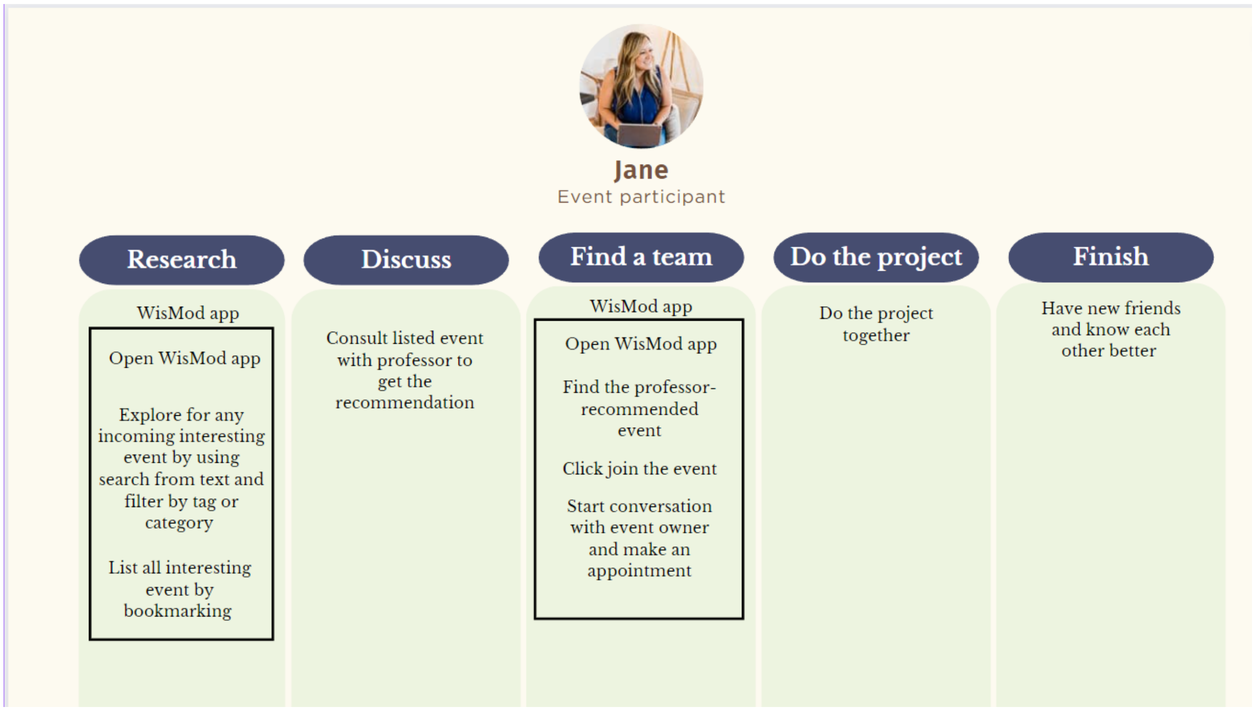


	Research	Discuss	Find a team	Do the project	Finish
DO	Explore the possibility of mechatronic engineering application through social media Contacts her brother from another engineering department Explore the existing event through online website or social media	Looking for any event opening for engineering knowledge	Asks professor to recommend some students who is interested on doing actual project Evaluate each one of the applicant by herself.	Join the competition with her team Collaborate the team with her skills	Take a photo together with the team
Goal	How can I apply my skills into the real work? I want to be like my brother Find some activity that she can join	Challenging herself using her engineering knowledge	To roughly know who can join with her Ensure that everyone has ability enough to carry out the project.	Register the team into a competition Gain more experiences and knowledge	Collab with team to do some awesome project together
FEEL	😊 😊 😔	😐	😔 😐	😊 😊	😊
Pain	Takes too much effort to find the specific event she wants.		It takes time and is too dependent on her friends only.		
CUSTOMER IMPROVEMENTS	Categorize event properly and provide the filter to access content more quickly		Allow non-Jane's friend student to join		

https://drive.google.com/file/d/1jc1L5X6Whzj7rXdgaKVPVep6kV9MWxoV/view?usp=sharing

Similar to John, Jane also faces issues in the Research and Find a team phase leading to negative emotions.

This is Jane's to-be Map:



Jane
Event participant

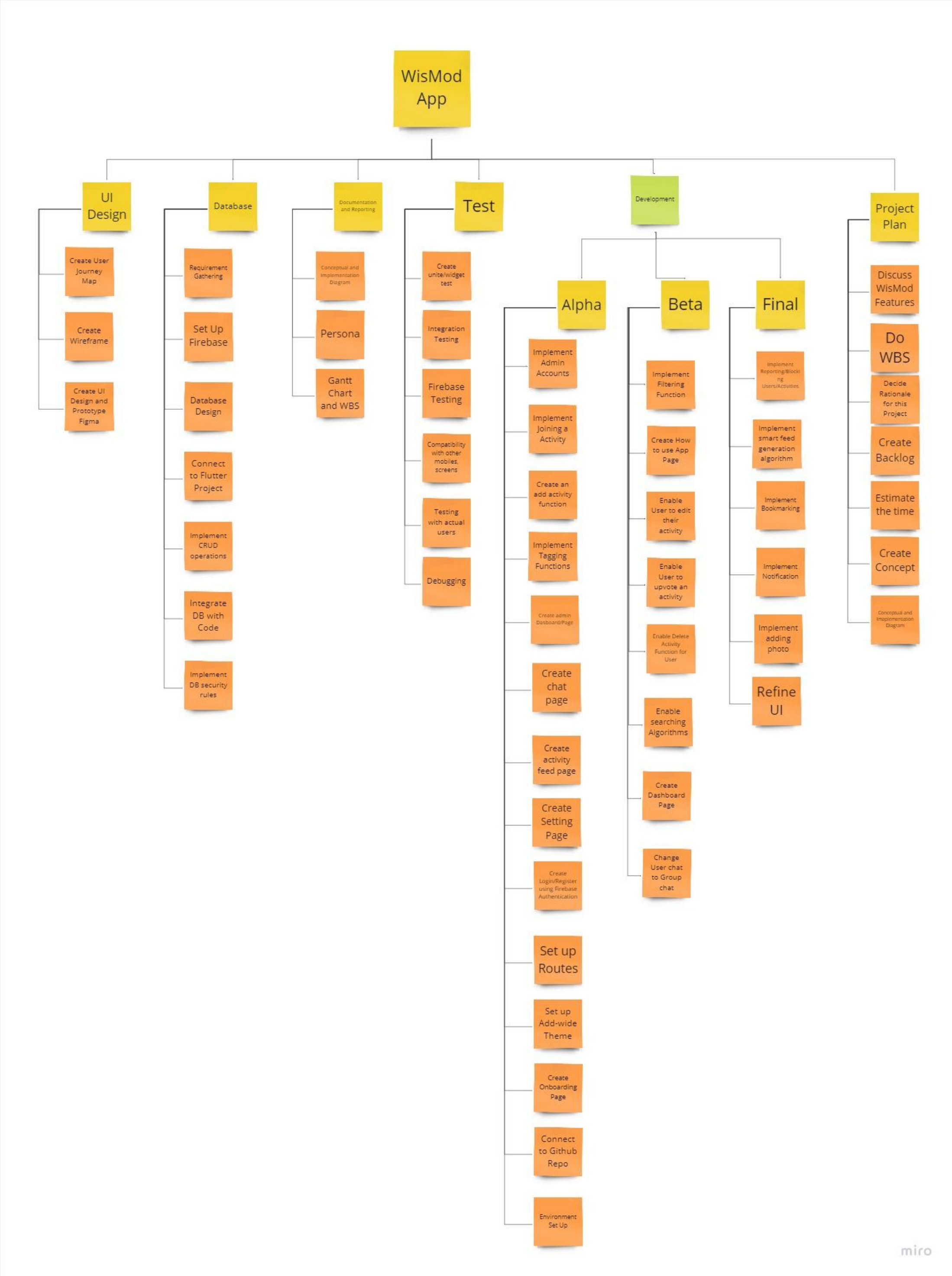
Research	Discuss	Find a team	Do the project	Finish
WisMod app Open WisMod app Explore for any incoming interesting event by using search from text and filter by tag or category List all interesting event by bookmarking	Consult listed event with professor to get the recommendation	WisMod app Open WisMod app Find the professor-recommended event Click join the event Start conversation with event owner and make an appointment	Do the project together	Have new friends and know each other better

https://drive.google.com/file/d/1-uF4CKuMRhCk-UmNlbUXS9s8N-yb3Jmx/view?usp=sharing

To help her, there are three main features which support participant’s pains, including

1. Searching and Filtering: Jane can quickly search got any interesting events
2. Bookmarking: Jane can save any event she is interested in so that she can look at them at a later date or for convenience,
3. Chatting: Easily get to know more about the event by directly chatting with the event owner and other participants.

Work Breakdown Structure


WisMod App
UI Design
Create User Journey Map
Create Wireframe
Create UI Design and Prototype Figma
Database
Requirement Gathering
Set Up Firebase
Database Design
Connect to Flutter Project
Implement CRUD operations
Integrate DB with Code
Implement DB security rules
Documentation and Reporting
Conceptual and Implementation Diagram
Persona
Gantt Chart and WBS
Test
Create unite/widget test
Integration Testing
Firebase Testing
Compatibility with other mobiles, screens
Testing with actual users
Debugging
Development
Alpha
Implement Admin Accounts
Implement Joining a Activity
Create an add activity function
Implement Tagging Functions
Create admin Dasboard/Page
Create chat page
Create activity feed page
Create Setting Page
Create Login/Register using Firebase Authentication
Set up Routes
Set up Add-wide Theme
Create Onboarding Page
Connect to Github Repo
Environment Set Up
Beta
Implement Filtering Function
Create How to use App Page
Enable User to edit their activity
Enable User to upvote an activity
Enable Delete Activity Function for User
Enable searching Algorithms
Create Dashboard Page
Change User chat to Group chat
Final
Implement Reporting/Blocking Users/Activities
Implement smart feed generation algorithm
Implement Bookmarking
Implement Notification
Implement adding photo
Refine UI
Project Plan
Discuss WisMod Features
Do WBS
Decide Rationale for this Project
Create Backlog
Estimate the time
Create Concept
Conceptual and Imaplementation Diagram
miro

https://imgur.com/a/DpoQtme

WBS Dictionary

1.UI design

1.1 Create Customer Journey within our Application
description: Creating a user journeymap to see the experience of the user.
Deliverable: Customer Journey Map
Acceptance Criteria:
Our customer journey should have

- Pain of the user
- Opportunity for our organization
- Our proposed features which will help solving user’s pain

1.2 Create wireframe
Description: Create and design prototype wireframe of our application
Deliverable: Prototype wire frame
Acceptance Criteria:

- Wire frame for base designing with UI

1.3 Define the concept and features
Description: Design what our app should be or how it should work
Deliverable: Concept and features of the application
Acceptance Criteria:

- The document of concept and features of the application

1.4 Create UI design and prototype figma
Description: Create the UI of our application in Figma
Deliverable: The complete UI of the application in Figma
Acceptance Criteria:

- Well and nice looking of the UI
- Can be implemented in coding

2.Database

2.1 Requirement gathering
Description: Research what data in our application that should be store in database.
Deliverable: Requirement Document
Acceptance Criteria:

- Category of all data within application
- Detailed description of all needed data

2.2 Set up Firestore
Description: Set up the Firebase Firestore servic
Deliverable: Authorized Firebase
Acceptance Criteria:

- Accessible Database Link for all team members
- Database added to Flutter project

2.3 Database Design

Description: Set the data collection of our application in Firebase Firestore.

Deliverable: Data storage diagram

Acceptance Criteria:

- Precise flow of the data within the app
- Create the collections for the database in firebase
- All information the app needed

2.4 Connect to flutter project

Description: Implemented the firestore in flutter.

Deliverable: The flutterfire that fully implemented with Firebase

Acceptance Criteria:

- The API can be called from Firebase

2.5 Implement CRUD operation

Description: Implemented CRUD operation in flutter

Deliverable: Backend with CRUD calls to database

Acceptance Criteria:

- Backend has CRUD implemented
- API calls have error handling
- Extreme state managed
- Reviewed by other team members

2.6 Integrate DB with UI

Description: Integrated database with UI

Acceptance Criteria:

- Front end can be interact with database

2.6 Implement DB security rule

Description: We implemented Firestore security rule in flutter.

Deliverable: Database with security rules

Acceptance Criteria:

- Every read and write request will only be completed if developer rules allow it

3.Documentation and Reporting

3.1 Conceptual and implementation Diagram

Description: Draw a Conceptual diagram of workflow in our application and Implementation diagram between mobile application and Firebase

Deliverable: Complete conceptual and implementation Diagram

Acceptance Criteria:

- Diagram is clear and can be follow the instruction

3.2 Persona

Goal: Do a customer segment by creating Persona of that customer group

Deliverable: Two Persona diagrams

Acceptance Criteria:

- Each persona represent the different type of customer
- Persona is relatable to the actual people

3.3 Gantt chart & WBS

Description: Make the Gantt chart & WBS chart for planning the time table of WBS.

Deliverable: Complete Gantt chart & WBS

Acceptance Criteria:

- The Gantt chart & WBS represented tasks in the project correctly
- The time of the sprint in chart is followed the ability to finished the task by team member

4.Test

4.1 Create Unit/Widget Test (Removed)

Description: Conduct unit/widget testing to ensure the UI units or components of the application function correctly and meet the expected behavior.

Deliverable: Unit/Widget Test Results

Acceptance Criteria:

- Develop and execute unit/widget tests for Global UI components.
- Validate that the units/widgets function as intended and produce the expected results.

Verdict:

- We were not able to write proper uni/widget tests as our application is highly intertwined with GetX and Firebase and there seems to be almost no resources on the internet on how to write tests when both are in use. We came to a conclusion that it is not worth it to spend the time learning Firebase and GetX tests so we conducted tests manually instead.

4.2 Integration Testing

Description: Perform integration testing to verify that different components of the application work together seamlessly and as expected. (Not coding tests but manually testing the components by running the application and doing as the test directs

Deliverable: Integration Test Results

Acceptance Criteria:

- Define integration test cases to validate the interactions between various components.
- Conduct those tests by mimicking those use cases and ensure the application works as expected.

4.3 Firebase Testing

Description: Conduct testing for features that are implemented using Firebase.

Deliverable: Firebase Test Results

Acceptance Criteria:

Identify features in the application that rely on Firebase services.

Perform testing to ensure proper integration, data handling, and functionality with Firebase.

4.4 Compatibility with Other Mobile Screens
Description: Test the application's compatibility and responsiveness with different mobile screen sizes and resolutions.
Deliverable: Compatibility Test Results
Acceptance Criteria:

- Test the application on various mobile devices with different screen sizes and resolutions.
- Verify that the application layout, elements, and interactions adapt well to different screen dimensions.

4.5 Testing with Actual Users
Description: Conduct testing sessions with real users to gather feedback and evaluate the application's usability and user experience.
Deliverable: User Testing Feedback and Reports
Acceptance Criteria:

- Plan and conduct user testing sessions with real users who are not part of the development team.
- Collect feedback on the application's usability, intuitiveness, and overall user experience.

4.6 Debugging
Description: Identify and resolve issues or bugs that arise during real-world usage of the application.
Deliverable: Bug Fixes and Issue Resolution
Acceptance Criteria:

- Analyze user feedback, bug reports, and error logs to identify issues.
- Debug and fix any identified issues to improve the application's stability and performance.

5.Alpha

5.1 Implement admin account
Description: Create the functionality to manage admin accounts.
Deliverable: Functional admin account feature.
Acceptance Criteria:

- Admin users can log in with their credentials.
- Admin users have access to additional administrative features and controls (denying report and deleting events)
- Only authorized admin users can access the admin account functionality.

5.2 Implement joining an activity
Description: Develop the functionality for users to join activities.
Deliverable: Functional join activity feature.
Acceptance Criteria:

- Users can join/request for activities they are interested in.
- Users can see the activities they have joined in their profile or dashboard.

5.3 Create an add activity function

Description: Implement the feature to add activities to the application.

Deliverable: Functional add activity feature.

Acceptance Criteria:

- Users can create new activities by providing necessary details such as title, description, date, time, and location.
- The add activity form validates input fields and displays appropriate error messages for missing or incorrect information.
- Created activities are saved in the system and become visible to other users.
- Users can edit or delete activities they have created.

5.4 Implement tagging function

Description: Add the capability to tag posts with relevant keywords.

Deliverable: Functional tagging feature.

Acceptance Criteria:

- Users can add tags to activities and posts.
- Activities and posts can be searched based on tags.
- Tags are available to be used in smart feed generation algorithm

5.5 Create an admin Dashboard/Page

Description: Create an admin page/dashboard for the admin account.

Deliverable: Functional admin dashboard/page.

Acceptance Criteria:

- The admin dashboard/page provides a list of reported events.
- Admin can deny report requests or remove events.

5.6 Create chat page

Description: Create a chat page for users to communicate.

Deliverable: Functional chat page.

Acceptance Criteria:

- Users can send and receive messages in real-time through the chat page.
- Chat messages are displayed in a threaded conversation format.
- Users can join new chat groups

5.7 Create activity feed page

Description: Create a feed page in the application.

Deliverable: Functional activity feed page.

Acceptance Criteria:

- The activity feed page displays activities or events.
- Users can see activities posted by others, including those they have joined or are interested in.

- The feed page provides options for filtering and sorting activities based on preferences.
- Users can interact with activities directly from the feed page, such as joining or commenting or upvoting.

5.8 Create setting page

Description: Create a settings page.

Deliverable: Functional settings page.

Acceptance Criteria:

- Users can access and modify their account settings.
- Users can view help page for more information
- Users can see blocked chat groups and can unblock them.

5.9 Create login/register using FirebaseAuth

Description: Create login and register pages and implement with FirebaseAuth.

Deliverable: Functional login and register pages integrated with FirebaseAuth.

Acceptance Criteria:

- Users can register new accounts using their email and password.
- Users can log in to the application using their registered credentials.
- FirebaseAuth handles user authentication securely and provides necessary validation checks.
- User login and registration processes are smooth and error-free.

5.10 Set up Routes

Description: Set up routing for each page in the application.

Deliverable: Functional page routing.

Acceptance Criteria:

- Each page of the application is accessible through its respective route.
- Navigation between pages is seamless and intuitive.
- Routing handles authentication checks and redirects users accordingly.
- The application URL reflects the current page being viewed.

5.11 Set up App-wide Theme

Description: Set up a consistent theme for the entire application.

Deliverable: App-wide theme setup.

Acceptance Criteria:

- The application follows a consistent design theme, including colors, typography, and layout.
- App-wide theme elements are defined and applied consistently across all pages.

5.12 Create onboarding page

Description: Create an onboarding page for new users.

Deliverable: Functional onboarding page.

Acceptance Criteria:

- The onboarding page provides a logo of WisMod and a button to get started.

5.13 Connect to GitHub repository

Description: Connect the application to a GitHub repository for version control and collaboration.

Deliverable: Connected GitHub repository.

Acceptance Criteria:

- The application codebase is connected to a remote GitHub repository.
- Version control is implemented, allowing for efficient tracking and management of code changes.
- Collaborative features of GitHub, such as pull requests and issue tracking, are utilized.
- Code is regularly pushed and pulled from the GitHub repository to ensure synchronization.
- All team members can push

5.14 Environment Setup

Description: Set up the necessary environment and assets for the application.

Deliverable: Prepared environment and preset theme assets in Flutter.

Acceptance Criteria:

- The development environment is properly configured with the required software and tools.
- The Flutter framework is installed and set up correctly.
- Preset theme assets, such as fonts, icons, and UI components, are integrated into the project.
- The application can be built and launched successfully in the designated environment

6. Beta

6.1 Implement filtering function

Description: Develop the functionality to filter activities based on specific criteria.

Deliverable: Functional filtering feature.

Acceptance Criteria:

- Users can filter by selecting a category and sort events by date
- The filtering function displays activities that match the selected criteria.
- Filtered activities are updated in real-time as users modify their filter preferences.

6.2 Create how-to-use app page

Description: Design and create a tutorial page to guide users on how to use the application.

Deliverable: How-to-use app page.

Acceptance Criteria:

- The how-to-use app page provides step-by-step instructions and explanations on various application features.

6.3 Enable users to edit their activity

Description: Implement the functionality for users to edit the details of their created activities.

Deliverable: Edit activity feature.

Acceptance Criteria:

- Users can access and modify the information of activities they have created.

- Editing options include updating activity details such as title, description, date, time, and location, image, allow automatic join etc.
- The edit activity feature validates input fields and displays appropriate error messages for missing or incorrect information.
- Changes made to the activity are saved and reflected in the application.

6.4 Enable users to upvote an activity

Description: Create the ability for users to upvote activities to show their appreciation or support.

Deliverable: Upvote activity feature.

Acceptance Criteria:

- Users can upvote activities they find interesting, useful, or enjoyable.
- Users can view the number of upvotes an activity has received.
- Users can remove upvotes.
- Changes reflected in application instantly
-

6.5 Enable delete activity function for users

Description: Develop the capability for users to delete activities they have created.

Deliverable: Delete activity feature.

Acceptance Criteria:

- Users can delete activities from the event details page.
- Deleting an activity removes it from the application and updates related information accordingly.
- Deleted activities are no longer visible to other users.

6.6 Enable searching algorithms

Description: Implement searching algorithms to allow the application's search functionality.

Deliverable: Search feature

Acceptance Criteria:

- The search function provides accurate and relevant results based on user queries.
- Search algorithms consider factors such as activity title, description, tags, and user preferences.
- Search results are presented in a clear and organized manner, prioritizing the most relevant activities.
- The search feature supports partial keyword matching

6.7 Create Dashboard page

Description: Design and create a dashboard page as the central hub for user activity and information.

Deliverable: Dashboard page.

Acceptance Criteria:

- The dashboard page displays personalized information
- Users can access and navigate various application features and sections from the dashboard.

- The dashboard provides an intuitive and user-friendly interface for managing user owned activities, managing join requests and viewing events.
- Users can change profile picture.

6.8 Change User chat to Group chat

Description: Modify the chat functionality to support group chat instead of one-on-one user chat.

Deliverable: Group chat feature.

Acceptance Criteria:

- Users can create and participate in group chats with multiple participants.
- Group chat allows for simultaneous communication and collaboration among participants.
- Users can join existing groups.

7. Final

7.1 Implement Reporting/Blocking/Users/Activities

Description: Develop the functionality for users to report activities or block other users within the application.

Deliverable: Reporting and blocking feature.

Acceptance Criteria:

- Users can report inappropriate activities or users through a dedicated reporting system.
- Reported activities are reviewed by the admin for further action, such as removal or denial.
- Users can block specific chats to prevent further interactions or communications.

7.2 Implement smart feed generation algorithm

Description: Develop an algorithm to generate a personalized and relevant feed on the feed page.

Deliverable: Smart feed generation algorithm.

Acceptance Criteria:

- The feed generation algorithm analyzes user preferences, activities they have interacted with, and other relevant factors to generate a customized feed.
- The algorithm considers factors such as activity categories, user interests, and upvoted activities.
- The smart feed also ensures the relevant discovery of activities which might be hidden due to feed generation.
- The smart feed algorithm adapts and improves over time based on user interactions.

7.3 Implement Bookmarking

Description: Enable users to bookmark activities for future reference or quick access.

Deliverable: Bookmarking feature.

Acceptance Criteria:

- Users can add activities to their bookmarks for easy retrieval.
- The bookmarked activities are stored in a designated section within the application.

- Users can view and manage their bookmarked activities, including removing bookmarks if desired.
- The bookmarking functionality provides a seamless and intuitive user experience.

7.4 Implement Notification

Description: Create a system for sending notifications to users' mobile devices.

Deliverable: Notification feature.

Acceptance Criteria:

- The application can send push notifications to users' devices for important updates, activity reminders, or new messages.

7.5 Implement adding photo

Description: Develop the functionality for users to add photos to their activity posts.

Deliverable: Photo upload feature.

Acceptance Criteria:

- Users can select and upload photos from their device's gallery or capture new photos within the application.
- The photo upload supports common image formats
- Added photos are displayed within the activity post or a designated photo section.

7.6 Refine UI

Description: Improve and refine the user interface (UI) design of the entire application.

Deliverable: Refined UI design.

Acceptance Criteria:

- Update UI to fix inconsistencies

8. Project Plan

8.1 Discuss WisMod Features

Description: Plan and define the scope what features our application should have

Deliverable: Feature List and Requirements

Acceptance Criteria:

- The feature descriptions are clear and unambiguous
- The features are align with the user needs of the software application
- The features has feasibility to implement within available resources

8.2 Do WBS

Description: Create WBS diagram in order to effective project planning, resource allocation, and project execution

Deliverable: WBS diagram

Acceptance Criteria:

- WBS comprehensively covers the entire project scope
- Clarity of the work package definitions within the WBS
- WBS effectively decomposes the project deliverables into manageable and understandable components.

8.3 Decide Rationale for this Project

Description: Create a clear and compelling reason or justification for undertaking the project

Deliverable: Guiding principle for undertaking project

Acceptance Criteria:

- Clarity of Purposes
- The rationale aligns with the needs and expectations of stakeholders

8.4 Create Backlog

Description: Create a prioritized and comprehensive list of features and tasks to be addressed in a project

Deliverable: Backlog

Acceptance Criteria:

- The backlog captures a comprehensive set of requirements and features for the project
- Be able to assess the prioritization and ranking of backlog items
- Clarity and granularity of backlog items

8.5 Estimate the time

Description: Provide an accurate and realistic assessment of the time required to complete project tasks

Deliverable: Gantt chart

Acceptance Criteria:

- Has reliability of the time estimation

8.6 Create Concept

Description: Create a concept is to develop a clear and compelling understanding of the project idea and solution.

Deliverable: Project Overview

Acceptance Criteria:

- Clearly value proposition, potential benefits and impact of the project
- Concept aligns with user needs and our goals

8.7 Conceptual and Implementation Diagram

Description: Provide a visual representation of the system architecture and design to understand overall structure, components and relationships of the system and implementation.

Deliverable: Conceptual Diagram and Implementation Diagram

Acceptance Criteria:

- Clear and accurate diagrams which can represent the intended structure and behavior
- Correctly capture the relationships, interactions, and dependencies among system components and modules

Backlog Planning

Task name	Status	Due	Sprint	Assigned To	Project	Final Time (Hours)	TIME EST (1st ROUND)	TIME EST (2nd Round)
Discuss WisMod Features	Done	17-Apr-23	Sprint 1	Faii, James, JoNote, Khush, Ulyssus, wara	Project Plan	1	1h, 1h,1h,1h, 1h,1h	1h,1h,1h,1h, 1h,1h
Decide Rationale for this Project	Done	17-Apr-23	Sprint 1	Faii, James, JoNote, Khush, Ulyssus, wara	Project Plan	1	1h, 1h,1h,1h, 1h,1h	1h,1h,1h,1h, 1h,1h
Create Concept	Done	18-Apr-23	Sprint 1	Khush	Project Plan	1	1h, 2h,1h,1h, 1.5h,2h	1h,1h,1h,1h, 1h,1h
Do WBS and figure out all the tasks	Done	18-Apr-23	Sprint 1	Faii, James, JoNote, Khush, Ulyssus, wara	Project Plan	1	1h, 1h,1h,1h,, 1.5h,1h	1h,1h,1h,1h, 1h,1h
Create Product and Sprint Backlogs	Done	19-Apr-23	Sprint 1	Khush	Project Plan	4	3h, 3h,3h,3h, 4h, 4h	3h,3h,3h,3h, 4h, 4h
Figure out est. time for each package	Done	19-Apr-23	Sprint 1	Faii, James, JoNote, Khush, Ulyssus, wara	Project Plan	4	3h, 4h,2h,3h, 4h, 4h	3h,3h,3h,4h, 4h, 4h

Create Presentation	Done	19-Apr-23	Sprint 1	Faii, James, JoNote, Ulyssus, wara	Project Plan	7	4h, 3h,4h,3h, 10h, 4h	5h,5h,5h,5h, 7h, 5h
Add tasks to sprint	Done	19-Apr-23	Sprint 1	Khush	Project Plan	1	1h, 1h,2h,1h, 0.5h, 1h	1h,1h,1h,1h, 1h,1h
Learning GetX	Done	29-Apr-23	Sprint 3	Faii, James, JoNote, Khush, Ulyssus, wara	Project Plan	7	6h, 3h,6h,3h, 8h, 6h	7h,6h,6h,6h, 7h, 4h
Backlog Items	Done	19-Apr-23	Sprint 1	Khush	Project Plan	2	1h, 2h,2h,2h, 1h,2h	1h,2h,1h,1h, 1h,1h
Gantt Chart and WBS	Done	10-May -23	Sprint 4	Faii, James	Documentation and Reporting	2	1h, 2h,2h,2h, 1.5h,2h	2h, 1h,1h, 1h, 1.5h,2h
Persona	Done	26-Apr-23	Sprint 1	Faii, James, Khush, Ulyssus	Documentation and Reporting	4	4h,2h,2h,3h, 4h,4h	3h, 4h,3h,3h, 4h,4h
Conceptual and Implementation Diagram Draft	Done	10-May -23	Sprint 4	Faii, James, JoNote, Khush, Ulyssus, wara	Project Plan	5	5h, 5h,4h,4h, 4h,4h	4h, 5h,4h,5h, 4h,4h
Conceptual and Implementation Diagram	Done	18-May -23	Sprint 5	Faii, James	Documentation and Reporting	5	5h, 5h,4h,4h, 4h,4h	4h, 5h,4h,5h, 4h,4h

Implement Filtering Function	Done	10-May-23	Sprint 4	Khush, wara	Beta	3	2h, 4h, 3h, 5h, 4h, 3h	3h, 3h, 2h, 3h, 3h, 3h
Create How to use App Page	Done	17-May-23	Sprint 5	JoNote	Beta	4	6h, 3h, 4h,3h, 3h, 3h	4h,3h,4h,3h, 3h, 3h
Enable User to edit their activity	Done	18-May-23	Sprint 4, Sprint 5 Delayed	James, JoNote	Beta	5	10h, 5h,5h,5h, 5h, 5h	5h,5h,5h,5h, 5h, 5h
Enable User to upvote an activity	Done	3-May-23	Sprint 3	Khush	Beta	5	4h, 3h,2h,3h, 5h, 5h	4h,3h,3h,3h, 5h, 5h
Enable Delete Activity Function for User	Done	18-May-23	Sprint 4, Sprint 5 Delayed	Khush	Beta	3	4h, 1h,2h,1h, 1h, 2h	3h,1h,1h,1h, 1h, 1h
Enable searching Algorithms	Done	10-May-23	Sprint 4	wara	Beta	6	7h, 5h,7h,5h, 4h, 5h	6h,5h,5h,5h, 5h, 5h
Create Dashboard Page	Done	15-May-23	Sprint 4, Sprint 5 Delayed	Ulysses	Beta	5	4h, 6h, 5h, 3h, 3h, 3h	4h, 5h, 4h. 4h. 4h, 4h
Change User chat to Group Chat	Done	10-May-23	Sprint 4	wara	Beta	4	3h,3h,4h,3h,4h,4h,4h	4h,4h,4h,4h,4h,4h

Reporting/Blocking Users/Activities	Done	14-May-23	Sprint 4, Sprint 5 Delayed	Faii, Khush	Final	6	7h, 5h,5h,5h, 7h,6h	5h, 6h,6h,5h, 6h,6h
Smart Feed generation algorithm	Done	17-May-23	Sprint 5	Khush	Final	8	4h, 8h,8h,8h, 8h,8h	8h, 8h,8h,8h, 8h,8h
Bookmarking/Save activities for later (Library)	Done	10-May-23	Sprint 4	JoNote, Ulyssus	Final	7	12h, 2h,2h,2h 5h,5h	2h, 7h,3h,3h, 5h,5h
Notifications Implementation	In Progress	18-May-23	Sprint 4, Sprint 5 Delayed	wara	Final	6	6h, 6h,6h,6h 6h,6h	6h, 6h,6h,6h, 6h,6h
Adding Photos to activity Functionality	Done	17-May-23	Sprint 3	Khush, wara	Final	5	4h, 7h, 5h,5h, 2h, 5h	5h, 5h,5h,5h, 5h,5h
Refine UI	Done	10-May-23	Sprint 4	James, Khush, Ulyssus	Final	10	10h, 8h,10h,8h, 12h,12h	8h, 10h,10h, 8h, 10h, 10h
Implement Admin Accounts	Done	3-May-23	Sprint 3	Khush	Alpha	2	2h, 2h,2h,2h, 2h, 2h	2h, 2h,2h, 2h, 2h, 2h
Implement Tagging Functions	Done	29-Apr-23	Sprint 3	Khush	Alpha	1	0.5h, 2h,1h,1h, 1h, 2h	1h, 1h,1h,1h,, 1h, 1h

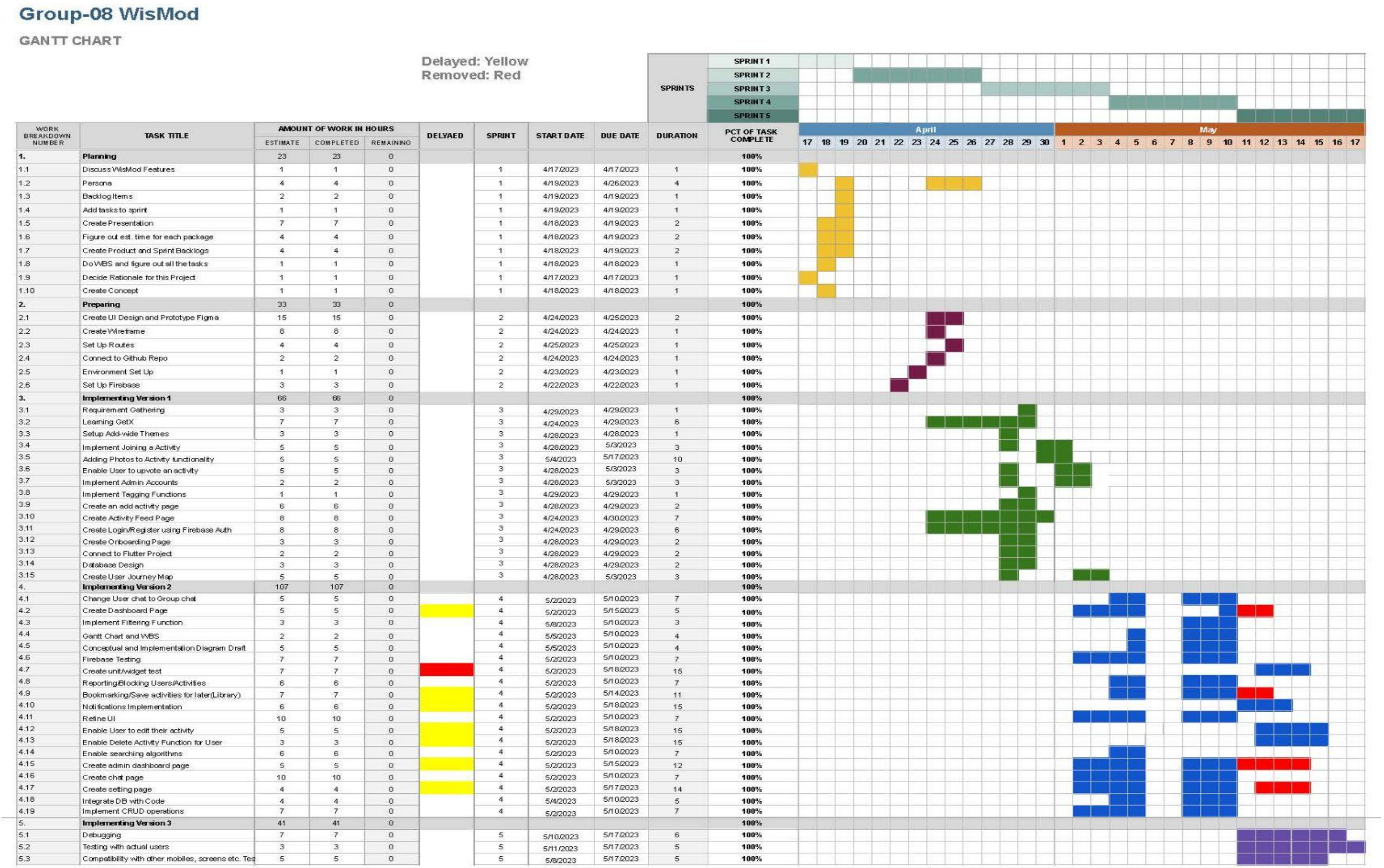
Implement Joining a Activity	Done	3-May-23	Sprint 3	Khush	Alpha	5	4h, 5h, 8h, 4h, 4h, 5h	4h, 4h, 5h, 4h, 4h, 4h
Create admin Dashboard/Page	Done	15-May-23	Sprint 4, Sprint 5 Delayed	Khush, Ulyssus	Alpha	5	5h, 5h,4h,5h, 4.5h,5h	4h, 5h,5h,4h, 4.5h,5h
Create an add activity page	Done	29-Apr-23	Sprint 3	Khush, wara	Alpha	6	8h, 2h,4h,4h, 3h,4h	4h, 6h,4h,4h, 4h,4h
Create chat page	Done	10-May-23	Sprint 4	wara	Alpha	10	15h, 8h,8h,8h, 10h, 10h	8h, 10h,10h, 8h, 8h, 8h
Create Activity Feed Page	Done	30-Apr-23	Sprint 3	Khush	Alpha	8	8h, 5h,8h,8h, 7h, 8h	8h, 8h,6h, 6h, 7h, 8h
Create Settings Page	Done	15-May-23	Sprint 4, Sprint 5 Delayed	JoNote	Alpha	4	4h, 3h,4h,3h, 3h, 3h	3h, 4h,4h,3h, 3h, 3h
Create Login/Register using Firebase Auth	Done	29-Apr-23	Sprint 3	Khush, Ulyssus	Alpha	8	7h, 7h,8h,7h, 8h, 8h	7h, 8h,7h,7h, 8h, 8h
Create Onboarding Page	Done	29-Apr-23	Sprint 3	Khush, Ulyssus	Alpha	3	3h, 3h,3h,3h, 3h, 3h	3h, 3h,3h,3h, 3h, 3h

Set Up Add-wide Themes	Done	28-Apr-23	Sprint 3	Khush	Alpha	3	3h, 4h, 3h, 5h, 1h, 1h	2h, 2h, 3h, 3h, 3h, 3h
Set Up Routes	Done	25-Apr-23	Sprint 2	Khush	Alpha	4	5h, 5h,2h,3h, 3h, 3h	3h, 4h,4h,4h, 3h, 3h
Connect to Github Repo	Done	24-Apr-23	Sprint 2	Faii, James, JoNote, Khush, Ulyssus, wara	Alpha	2	2h, 2h,2h,2h, 2h, 3h	2h,2h,2h,2h, 2h, 3h
Environment Set Up	Done	23-Apr-23	Sprint 2	Khush	Alpha	1	1h, 1h, 1h,1h, 1h, 1h	1h,1h,1h,1h, 1h, 1h
Integrate DB with Code	Done	10-May-23	Sprint 4	Khush, Ulyssus, wara	Database	4	4h, 3h,4h,4h,4h,4h	4h,4h,4h,4h, 4h,4h
Implement CRUD operations	Done	10-May-23	Sprint 4	Khush, Ulyssus, wara	Database	7	8h, 5h,6h,6h,5h,5h	7h,7h,6h,6h,6h,6h
Connect to Flutter Project	Done	29-Apr-23	Sprint 3	Khush	Database	2	2h, 2h,1h,2h,1h,1h	2h,2h,1h,2h,1h,1h
Database Design	Done	29-Apr-23	Sprint 3	Faii	Database	3	2h, 2h,2h,2h,3h,2h	2h,2h,2h,3h,3h,2h

Requirements Gathering	Done	29-Apr-23	Sprint 3	Faii	Database	3	2h, 3h,2h,3h,3h,2h	2h,2h,2h,3h,3h,2h
Create User Journey Map	Done	3-May-23	Sprint 3	James	UI Design	5	3h, 3h, 4h,3h, 4h, 3h	4h,4h, 4h, 3h, 4h, 5h
Create Wireframe	Done	24-Apr-23	Sprint 2	Faii, James, JoNote, Khush	UI Design	8	7h, 8h, 7h,7h, 8h, 8h	8h,8h, 8h, 8h, 8h, 8h
Create UI Design and Prototype Figma	Done	25-Apr-23	Sprint 2	James, JoNote, Ulyssus	UI Design	15	15h, 12h, 12h,15h, 24h, 24h	12h,13h, 12h, 12h, 15h, 14h
Set Up Firebase	Done	22-Apr-23	Sprint 2	Khush	Database	3	1h, 3h,1h,1h,3h,1h	2h,2h,2h,3h,3h,2h
Implement DB security rules	Done	17-May-23	Sprint 5	JoNote, Khush	Database	6	6h, 6h,3h,3h,6h,6h	6h,6h,6h,6h, 6h,6h
Create unit/widget test	Removed	18-May-23	Sprint 4, Sprint 5 Removed	Khush	Tests	7	10h, 7h, 7h,7h, 5h, 8h	7h,7h, 7h, 7h, 7h, 7h
Integration Testing	Done	17-May-23	Sprint 5	Khush, Ulyssus	Tests	8	19h, 8h, 8h,8h, 7h, 8h	8h,8h, 8h, 8h, 8h, 8h

Firebase Testing	Done	10-May-23	Sprint 4	James, Khush	Tests	7	6h, 6h, 6h,6h, 8h, 8h	6h,6h, 6h, 6h, 7h, 6h
Compatibility with other mobiles, screens etc. Testing	Done	17-May-23	Sprint 5	JoNote	Tests	5	5h, 4h, 4h,3h, 3h, 3h	3h,4h, 5h, 4h, 3h, 4h
Testing with actual users	Done	17-May-23	Sprint 5	Faii	Tests	3	3h, 3h, 3h,3h, 2h, 2h	3h,3h, 3h, 3h, 3h, 3h
Debugging	Done	17-May-23	Sprint 5	James, Khush, Ulyssus, wara	Tests	7	7h, 7h, 7h,7h, 7h, 7h	7h,7h, 7h, 7h, 7h, 7h

Gantt Chart

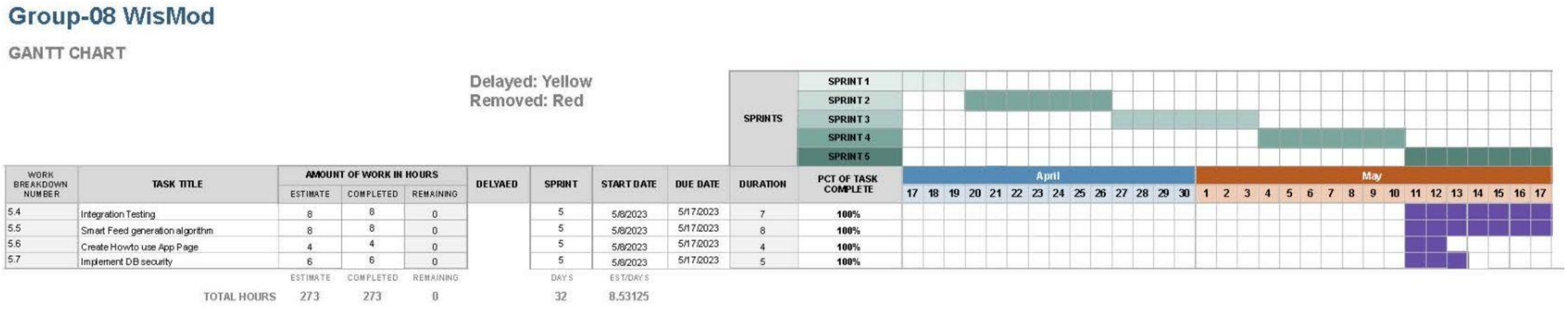


Group-08 WisMod

GANTT CHART

Delayed: Yellow
Removed: Red

WORK BREAKDOWN NUMBER	TASK TITLE	AMOUNT OF WORK IN HOURS			DELYAED	SPRINT	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
		ESTIMATE	COMPLETED	REMAINING						
1.	**Planning**	23	23	0						**100%**
1.1	Discuss WisMod Features	1	1	0		1	4/17/2023	4/17/2023	1	**100%**
1.2	Persona	4	4	0		1	4/19/2023	4/26/2023	4	**100%**
1.3	Backlog Items	2	2	0		1	4/19/2023	4/19/2023	1	**100%**
1.4	Add tasks to sprint	1	1	0		1	4/19/2023	4/19/2023	1	**100%**
1.5	Create Presentation	7	7	0		1	4/18/2023	4/19/2023	2	**100%**
1.6	Figure out est. time for each package	4	4	0		1	4/18/2023	4/19/2023	2	**100%**
1.7	Create Product and Sprint Backlogs	4	4	0		1	4/18/2023	4/19/2023	2	**100%**
1.8	Do WBS and figure out all the tasks	1	1	0		1	4/18/2023	4/18/2023	1	**100%**
1.9	Decide Rationale for this Project	1	1	0		1	4/17/2023	4/17/2023	1	**100%**
1.10	Create Concept	1	1	0		1	4/18/2023	4/18/2023	1	**100%**
2.	**Preparing**	33	33	0						**100%**
2.1	Create UI Design and Prototype Figma	15	15	0		2	4/24/2023	4/25/2023	2	**100%**
2.2	Create Wireframe	8	8	0		2	4/24/2023	4/24/2023	1	**100%**
2.3	Set Up Routes	4	4	0		2	4/25/2023	4/25/2023	1	**100%**
2.4	Connect to Github Repo	2	2	0		2	4/24/2023	4/24/2023	1	**100%**
2.5	Environment Set Up	1	1	0		2	4/23/2023	4/23/2023	1	**100%**
2.6	Set Up Firebase	3	3	0		2	4/22/2023	4/22/2023	1	**100%**
3.	**Implementing Version 1**	66	66	0						**100%**
3.1	Requirement Gathering	3	3	0		3	4/29/2023	4/29/2023	1	**100%**
3.2	Learning GetX	7	7	0		3	4/24/2023	4/29/2023	6	**100%**
3.3	Setup Add-wide Themes	3	3	0		3	4/28/2023	4/28/2023	1	**100%**
3.4	Implement Joining a Activity	5	5	0		3	4/28/2023	5/3/2023	3	**100%**
3.5	Adding Photos to Activity functionality	5	5	0		3	5/4/2023	5/17/2023	10	**100%**
3.6	Enable User to upvote an activity	5	5	0		3	4/28/2023	5/3/2023	3	**100%**
3.7	Implement Admin Accounts	2	2	0		3	4/28/2023	5/3/2023	3	**100%**
3.8	Implement Tagging Functions	1	1	0		3	4/29/2023	4/29/2023	1	**100%**
3.9	Create an add activity page	6	6	0		3	4/28/2023	4/29/2023	2	**100%**
3.10	Create Activity Feed Page	8	8	0		3	4/24/2023	4/30/2023	7	**100%**
3.11	Create Login/Register using Firebase Auth	8	8	0		3	4/24/2023	4/29/2023	6	**100%**
3.12	Create Onboarding Page	3	3	0		3	4/28/2023	4/29/2023	2	**100%**
3.13	Connect to Flutter Project	2	2	0		3	4/28/2023	4/29/2023	2	**100%**
3.14	Database Design	3	3	0		3	4/28/2023	4/29/2023	2	**100%**
3.15	Create User Journey Map	5	5	0		3	4/28/2023	5/3/2023	3	**100%**
4.	**Implementing Version 2**	107	107	0						**100%**
4.1	Change User chat to Group chat	5	5	0		4	5/2/2023	5/10/2023	7	**100%**
4.2	Create Dashboard Page	5	5	0	🟨	4	5/2/2023	5/15/2023	5	**100%**
4.3	Implement Filtering Function	3	3	0		4	5/8/2023	5/10/2023	3	**100%**
4.4	Gantt Chart and WBS	2	2	0		4	5/5/2023	5/10/2023	4	**100%**
4.5	Conceptual and Implementation Diagram Draft	5	5	0		4	5/5/2023	5/10/2023	4	**100%**
4.6	Firebase Testing	7	7	0		4	5/2/2023	5/10/2023	7	**100%**
4.7	Create unit/widget test	7	7	0	🟥	4	5/2/2023	5/18/2023	15	**100%**
4.8	Reporting/Blocking Users/Activities	6	6	0		4	5/2/2023	5/10/2023	7	**100%**
4.9	Bookmarking/Save activities for later(Library)	7	7	0	🟨	4	5/2/2023	5/14/2023	11	**100%**
4.10	Notifications Implementation	6	6	0	🟨	4	5/2/2023	5/18/2023	15	**100%**
4.11	Refine UI	10	10	0		4	5/2/2023	5/10/2023	7	**100%**
4.12	Enable User to edit their activity	5	5	0	🟨	4	5/2/2023	5/18/2023	15	**100%**
4.13	Enable Delete Activity Function for User	3	3	0	🟨	4	5/2/2023	5/18/2023	15	**100%**
4.14	Enable searching algorithms	6	6	0		4	5/2/2023	5/10/2023	7	**100%**
4.15	Create admin dashboard page	5	5	0	🟨	4	5/2/2023	5/15/2023	12	**100%**
4.16	Create chat page	10	10	0		4	5/2/2023	5/10/2023	7	**100%**
4.17	Create setting page	4	4	0	🟨	4	5/2/2023	5/17/2023	14	**100%**
4.18	Integrate DB with Code	4	4	0		4	5/4/2023	5/10/2023	5	**100%**
4.19	Implement CRUD operations	7	7	0		4	5/2/2023	5/10/2023	7	**100%**
5.	**Implementing Version 3**	41	41	0						**100%**
5.1	Debugging	7	7	0		5	5/10/2023	5/17/2023	6	**100%**
5.2	Testing with actual users	3	3	0		5	5/11/2023	5/17/2023	5	**100%**
5.3	Compatibility with other mobiles, screens etc. Tes	5	5	0		5	5/8/2023	5/17/2023	5	**100%**



Group-08 WisMod

GANTT CHART

Delayed: Yellow
Removed: Red

WORK BREAKDOWN NUMBER	TASK TITLE	AMOUNT OF WORK IN HOURS			DELYAED	SPRINT	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
		ESTIMATE	COMPLETED	REMAINING						
5.4	Integration Testing	8	8	0		5	5/8/2023	5/17/2023	7	100%
5.5	Smart Feed generation algorithm	8	8	0		5	5/8/2023	5/17/2023	8	100%
5.6	Create Howto use App Page	4	4	0		5	5/8/2023	5/17/2023	4	100%
5.7	Implement DB security	6	6	0		5	5/8/2023	5/17/2023	5	100%
		ESTIMATE	COMPLETED	REMAINING		DAYS	EST/DAYS			
	TOTAL HOURS	273	273	0		32	8.53125			

Gantt Chart Link: https://docs.google.com/spreadsheets/d/1YXa5546J7CJcJoedS2Py-4a4S71mH-IxbJwWOf9PJE4/edit?usp=sharing

We implemented WBS to Gantt chart by setting each work package to have the time table, Which is be able to be implemented by the sprint of each bucket in our project. The Gantt chart type we choose to implement with our project is Finish to Start(FS) which is Task A must be finished before Task B can be started.

1.Planning Task

In this task include of work packages that're in the planning state of the project The sprint of the this task is the sprint 1 ,which has the duration of 3 days. However, there is a persona work package that have took the space in sprint 2. This happened because there are some adjustment of the persona work package.

2.Preparing Task

In this task include of work packages that're in the set up state of the project The sprint of the this task is the sprint 2 ,which has the duration of 1 week.

3.Implementing Version1 Task

In this task include of work packages that're in the development of Alpha state of the project The sprint of the this task is the sprint 3 ,which has the duration of 1 week. Since the task of sprint 2 is not much, we move to this task early.

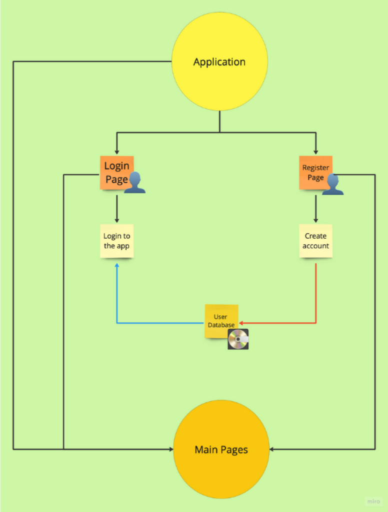
4.Implementing Version2 Task

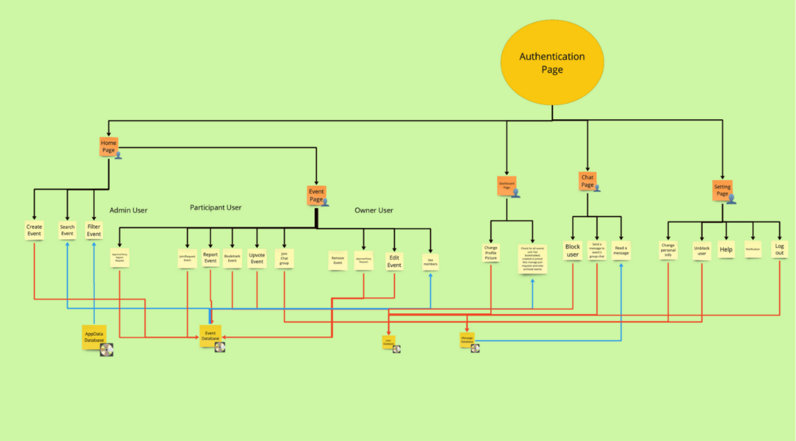
In this task include of work packages that're in the development of Beta state of the project The sprint of the this task is the sprint 4 ,which has the duration of 1 week. In this task, we delayed some work packages to be working in sprint 5, which is showing in yellow color. We removed some work packages from this tasks which are showing in red color.

5.Implementing Version3 Task

In this task include of work packages that're in the development of Final state of the project The sprint of the this task is the sprint 5 ,which has the duration of 1 week.

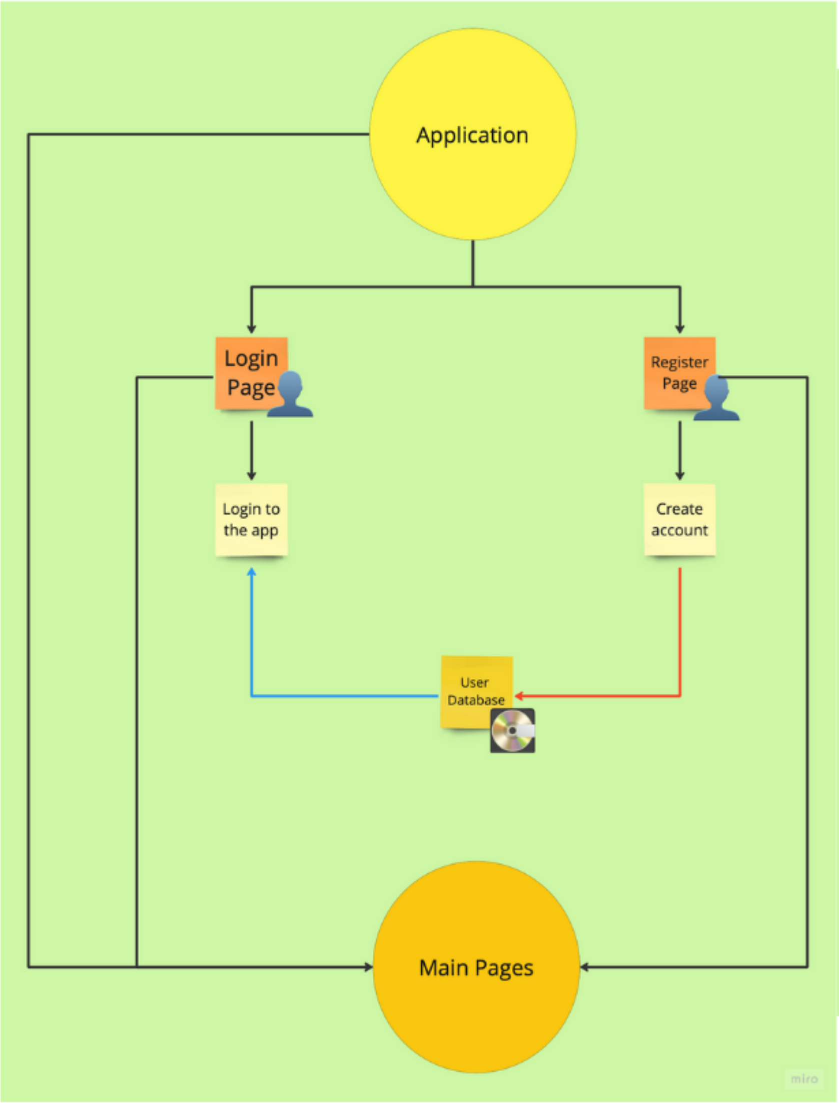
Conceptual Diagram





https://miro.com/app/board/uXjVMQJqdxA=/?moveToWidget=3458764556418524267&cot=14

Since our conceptual diagram is too large, we divide them into 2 parts; the Authentication page and the Main pages. We start the conceptual diagram at the authentication page where users first enter the application and then they proceed to the main pages.

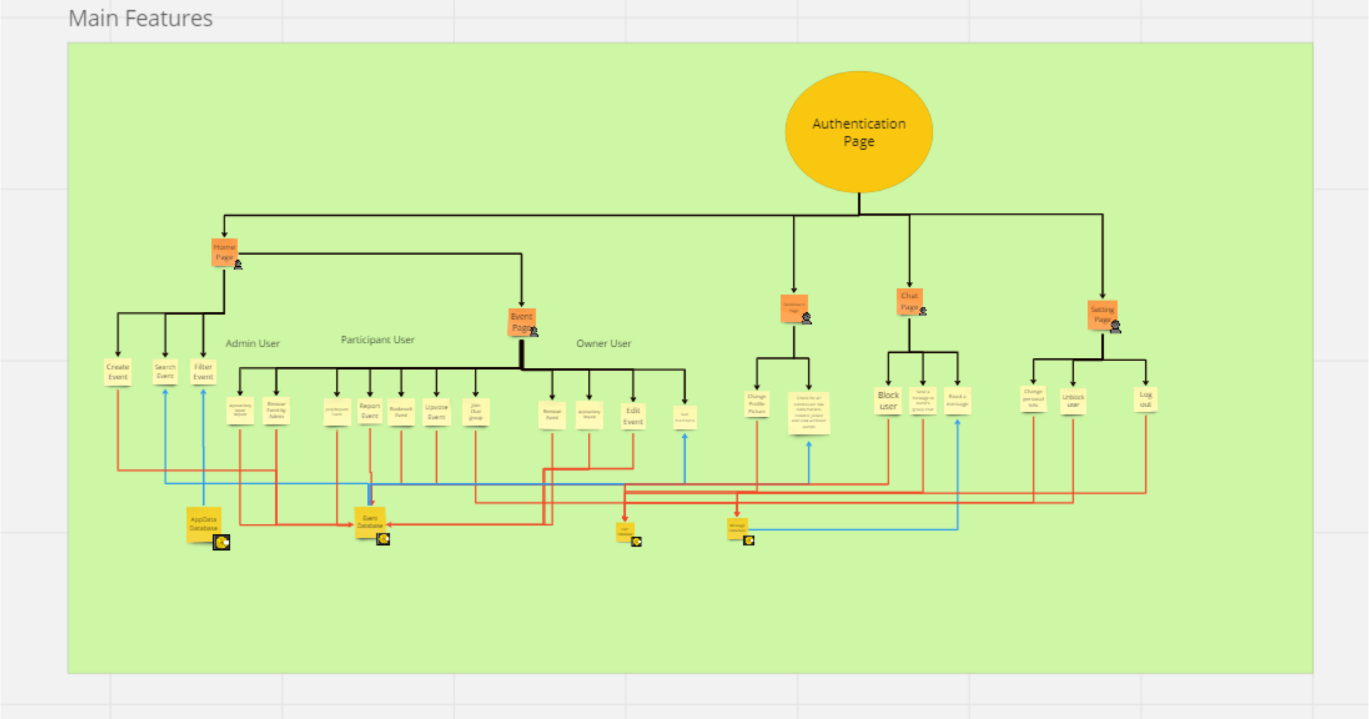


https://drive.google.com/file/d/1kKOVsvPl8Ngwi-Da4jkfuVNVtRSCU8Fm/view?usp=sharing

1. Authentication Page

When a user first enters the system, they have 2 choices whether to directly log in or register for a new account. Both actions require the use of a user information database which we stored in the firebase identified as User Database. The Register page will not automatically log in, it will direct the user after they create an account to the log in, so the Register page only pushes change to the database. While the Log in only retrieves data from the database but does not make any changes.

After the user logs in they will be directed to the main pages.



https://miro.com/app/board/uXjVMQJqdxA=/?moveToWidget=3458764556418524267&cot=14

2. Main Pages

2.1 Home Page

Feed page is designed to be the center of the app, where it has a major feature here. The major actions user can do are create event, search for event and filter for specific event. Create event requires making change to the database since it's adding a new event into the app, so database needs to be updated and then the screen of the application will display the event there. The search and filter function are very similar, they do not make any change to the database because their goal is to find the exact or related event to the users, therefore they only require the data to be put into their algorithm.

2.1.1 Event Page

Inside the home page, another feature user can do is to interact with the event owner through the event page. But over here, we have 3 groups of user; event participant, event owner & admin.

For event participant, they have many choices to do here. It could be

- Upvote Event: viewer user can upvote the event they are interested or want to promote. The action here pushes change into the Event Database. And it will refresh the app to show the current number of upvote.
- Join Event: normal user can join the event they wish. This action pushes change to the event database as well. But none of the content will be refreshed and shown until the owner accept the request, then the event page will show more current participants number.
- Report Event: in case the event detail looks threatening or harassment in some way, user can choose to report the event. This information will be sent to admin, so they can determine to keep this event or not. The action taken here requires change to the database.
- Bookmark Event: user can collect all event they like in one place via this bookmark function. It will send changes to the user database to update the list of bookmarked event of that person.
- Join chat group: user can choose not to join first but explore about that particular event by chatting with the group chat for that event first. This action will make change only to the message database where it contains the member of that chat room.

On the other side user which is owner user, they can do these actions

- See member: owner can check who joined their events, this action doesn't change anything to the database but it only takes data to show here.
- Edit event: very similar function to create but only that edit event is done after publish the event. This action makes change to the event database.
- Remove event, when user abort their event, they can choose to remove that particular event, Removal needs making change to the database.
- Approve/deny request: as we mentioned earlier that normal user will send request to join the event, but the result that they will be accepted is only up to owner. This action will change information inside the event database.

Lastly, in order to maintain safe space for people to enjoy the event being held within KMUTT society, the admin is recruited to ensure the secure environment. Their major action here is

- Approve/deny report request: admin will check for any report made by user, they need to check first whether the event is really harassing. This action will be updating the database, for approval it will terminate that event detail inside the event database. But for denial, the request which is sent will be terminated, but there will be no change on UI.
- Remove Event: admin has authority to check by their own in case they found the harassing event by themselves, they can remove that event immediately. This action makes direct change to the event database.

3. Dashboard Page

Dashboard page is where user can have a brief overview of all events they interacted with.

- Change Profile Picture: user will have white picture as their default profile first. But they are allowed to change the profile. This feature needs direct change to the user database where the app needs to retrieve new profile picture and render it to the current app.
- Check for all events (Bookmarked, upvoted, joined own): user can check for all events they have interacted with. These actions do not make any direct change to the event database nor the user database. But they pull data they want from those databases.

4. Chat Page

Chat page collects all chat that user have joined, it has both unjoined and joined events here.

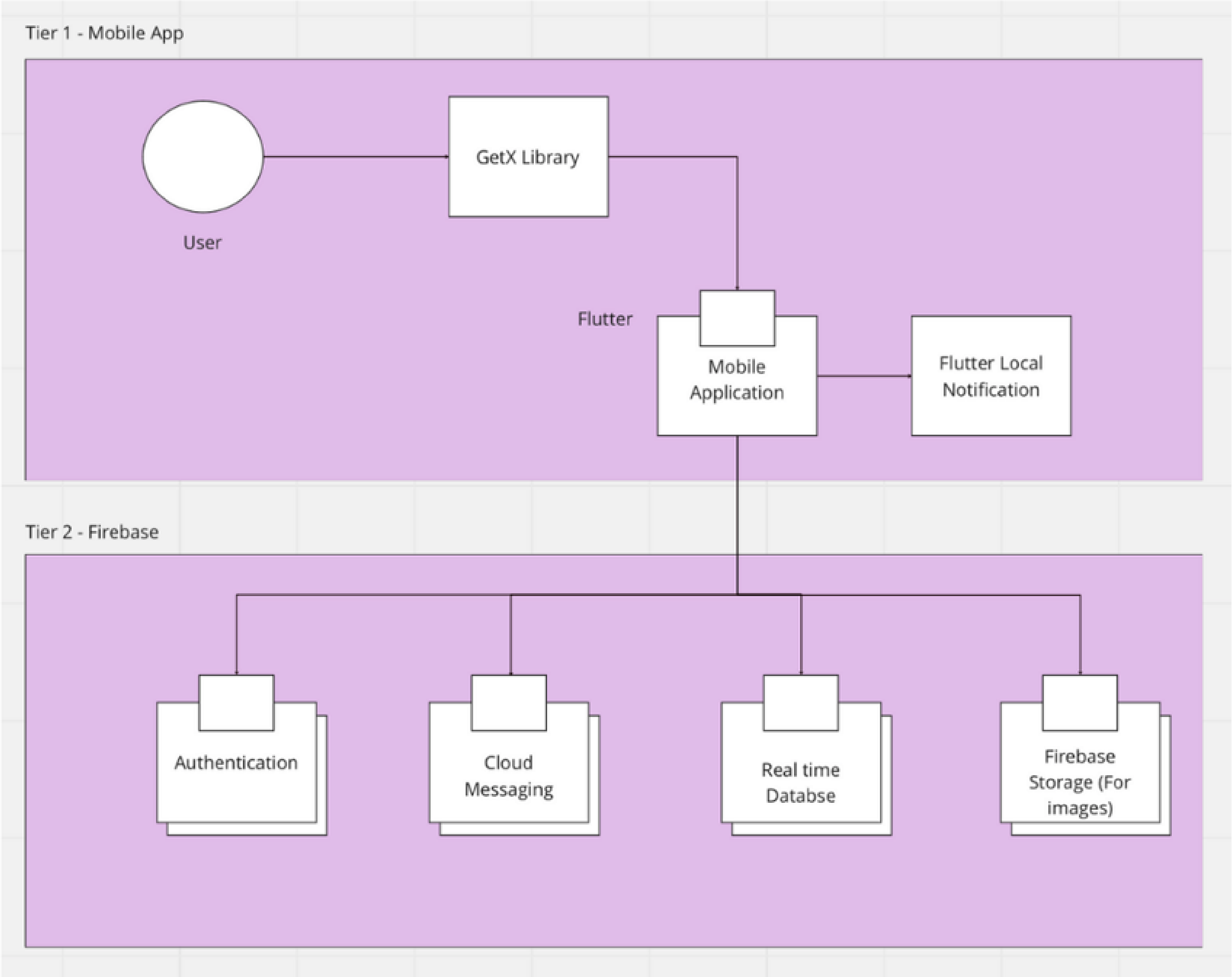
- Block User: users can choose to block anyone who is behaving badly or annoying. This makes change directly to both blocked users and the chat room owner. The user database will be updated since it's where the blocked user list is. While the blocked user will not be able to chat in that chat room anymore because they are removed from that room by the owner.
- Send message: users are allowed to communicate using group chat. Each message transfer pushes change into message database.
- Read a message: following from the send message, the user can read a message. All message shown in the chat are retrieved from the message database.

5. Setting Page

Setting page acts like a normal setting page people have. It offers user the change operation to all personal information. Most of the actions are related to the user database.

- Change personal information: this includes all name, year, faulty. This action makes direct changes to the user database.
- Unblock user: there can be mistakes between the conversation, so we offer user to reconcile with each other via the unblock feature. This feature changes the blocked user list in the user database.
- Log out: user can choose to log out if they want, this will update the user database for the log out time.

Implementation Diagram



https://drive.google.com/file/d/1CZre7n5dsaIkcIpqWU0YVUmdiQZkgPzY/view?usp=sharing

Implementation diagram shows how our applications are implemented. First, the user interact with the application the UI that user interact, are using GetX library to connect the fontend and backend, which are developed by using flutter framework. Flutter can send notification to user mobile with its local notification. When user use implemented functions in our application such as Login/Register, Chatting, Create/delete/edit event, upload image. The application will call api from the Firebase platform for processing that function.