User Story

1 Privacy Statement: As a user, I want to be able to view and agree to the Privacy statement, so that I can use the app with confidence.

Acceptance Criteria

1 Privacy statement must be displayed in a visible location within the app.

2 User must be able to access the privacy statement at any time.

3 User must be able to read and understand the privacy statement.

4 User must be able to agree to the privacy statement before using the app.

5 Once agreed, the privacy statement must not be displayed again unless there are updates to the statement.

Tasks

1 Design the layout for the privacy statement page.

2Implement the privacy statement page in the app.

3 Create a button for users to access the privacy statement.

4 Ensure that the privacy statement is displayed in a visible location.

5 Write the privacy statement in clear and understandable language.

6 Create a checkbox for users to agree to the privacy statement.

User Story

2.User account system: As a new user, I want to be able to create an account and provide my basic information, so that I can use the app and connect with others.

Acceptance Criteria

1 User must be able to create a new account with a unique email and password.

2 User must be able to provide their basic information, including their name and date of birth.

3 User must be able to edit their basic information at any time.

4 User must receive a confirmation email after creating an account.

5 User must be able to log in and log out of their account."

Tasks

1 Design the layout for the account creation page.

2 Implement the account creation functionality, including validation of unique email and password.

3 Create the database table to store user information.

4 Implement the functionality to save user information to the database.

5 Design the layout for the user profile page.

6 Implement the functionality for the user to edit their basic information."

Team Joint As a user, I want to be able to search for teams within a 5km distance, so that I can apply to join and connect with nearby teams.

1 User must be able to enter their location and search for teams within a 5km distance.

2Teams within the 5km distance must be displayed in a list, sorted by distance.

3 User must be able to see the name, location, and team details for each team in the list.

4 User must be able to apply to join a team from the list.

5 Team administrators must receive an email notification when a user applies to join their team.

1 Design the layout for the team search page.

2 Implement the search functionality that allows the user to enter their location and search for teams within a 5km distance.

3 Create a database table to store team information, including team name, location, and details.

4 Implement the functionality to save team information to the database.

5 Implement the functionality to display a list of teams within a 5km distance, sorted by distance.

6 Design the layout for the team details page.

7 Implement the functionality to display the team name, location, and details on the team details page.

8 Implement the functionality for the user to apply to join a team from the team details page.

9 Create a database table to store user applications to join teams."

Location share open As a team member, I want to be able to start share my location with my team, so that we can coordinate and communicate effectively.

1 Team member must be able to turn on their location sharing feature.

2 Team member's location must be visible to other team members who have also enabled the location sharing feature.

3 Team member must be able to disable their location sharing feature at any time.

4 Team member's location must be accurate and update in real-time.

5 Location data must be stored securely and not shared with any third parties.

1 Design the layout for the location sharing feature.

2 Implement the functionality to enable and disable the location sharing feature.

3 Create a database table to store the location data of team members who have enabled the location sharing feature.

4 Implement the functionality to store location data securely in the database.

5 Implement the functionality to display the location of team members who have enabled the location sharing feature to other team members.

6 Implement the functionality to update the location of team members in real-time.

7 Implement the functionality to disable the location sharing feature for a team member.

8 Test the functionality to ensure it works as expected."

Location share As a team member, I want to be able to see the locations of other team members, so that I can coordinate with them effectively.

1.The user can view a map that shows the locations of all team members who have opted to share their location.

2.The map should be zoomable and scrollable to allow the user to view different areas in detail.

3.The user can click on a team member's location to see more information about them, such as their name, role, and contact information.

4.The map updates in real-time to reflect changes in the location of team members.

1.Identify location tracking requirements(For example,discuss with stakeholders to identify the level of accuracy required for location tracking and etermine how often locations should be updated)

2.Implement a database to store and retrieve location data.

3.Create a user interface for displaying locations

4.Test and debug location tracking and display

5.Create user documentation explaining how to use the location tracking feature."

Location share close As a team member, I want to be able to end the location sharing service at any time, so that I can protect my privacy.

1.The user can easily access a ""End sharing"" button within the app that is providing the location sharing feature.

2.Clicking the ""End sharing"" button will immediately end the user's participation in the location sharing service.

3.When the user stops sharing their location, the location data associated with their account is deleted from the system and is no longer visible to other team members.

4.The user should receive a confirmation message or notification indicating that their location sharing has been stopped.

5.If the user later decides to resume location sharing, they should be able to easily opt back in and resume sharing their location with the team.

1.Add a button to the user interface that allows team members to end location sharing at any time.

2.Modify the location database to stop collecting and storing location data for team members who have ended location sharing.

Ensure that the location data is securely deleted.

3.Update the user documentation to include information about how to end location sharing.

4.Test the ""End Sharing"" button and confirm that it stops location sharing for the appropriate team member."

Team leader position Transfer: As a team leader, I want to be able to transfer team leader position, and add or remove members, so that I can manage my team effectively.

1.To transfer the team leader position, the current team leader must select a new team leader from the list of existing team members, and the new team leader must accept the transfer.

2.If the new team leader declines the transfer, the transfer is cancelled and the current team leader retains their position.

3.The team leader should be able to remove members from the team by selecting them from a list of current team members and confirming the removal.

4.When a member is removed from the team, their access to team resources and data should be immediately revoked.

5.The system should provide appropriate notifications or alerts to team members when they are added or removed from the team, or when the team leader position is transferred.

1.Design the user interface for managing team membership and leadership changes.

2.Implement the ability to transfer team leader position to another team member.

3.Ensure that the appropriate team member permissions are updated when there are changes in team leadership or membership.

4.Design and create a database schema to store team membership and leadership data.

5.Update documentation to reflect the new functionality."

Team building system. As a user, I want to be able to create my own team and I can choose a team leader including myself, so that I can coordinate and manage the team.

1.To create a new team, the user must provide a unique team name and select a team leader from a list of available team members, including themselves.

2.The team name must not already exist in the system, and must meet any specified requirements for length, characters, or format.

3.The team leader must accept their role as team leader before the team is fully created.

4.The user should be able to remove members from the team by selecting them from a list of current team members and confirming the removal.

5.When a member is removed from the team, their access to team resources and data should be immediately revoked。

1.Design and develop a user interface for creating a new team, which includes a form to enter the team's name, description, and other relevant information.

2.Implement a feature to allow the user to choose the team leader, including themselves.

3.Design and create a database to store the new team information.

4.Test the creation of a new team, adding or removing team members.

5.Create documentation and user guides to help users understand how to create and manage teams.

"

Meeting Point 9. As a team leader, I want to be able to set a meeting point for my team, so that we can easily find each other.

1. The user interface should have an option for the team leader to set a meeting point.

2. The meeting point should be visible to all team members.

3. The meeting point should be marked on the map.

4. The meeting point should be easily accessible to team members.

5. The meeting point should be customizable and editable by the team leader.

6. The meeting point should be saved for future reference.

1. Add a ""Set Meeting Point"" button to the user interface.

2. Implement the functionality to set the meeting point on the map.

3. Ensure that the meeting point is visible to all team members.

4. Implement a system to save and retrieve the meeting point data.

5. Add an option for the team leader to customize and edit the meeting point.

6. Test the meeting point functionality to ensure it meets the acceptance criteria.

7. Deploy the updated application to the production environment.

Chatting 10. As a team member, I want to be able to chat with other team members, so that we can communicate effectively and share information.

1. The chat feature is integrated within the application.

2. Team members are able to send and receive messages with other members of their group.

3. Team members can create new chat groups and invite other members to join.

4. Messages are displayed in real-time and team members are notified of new messages.

5. Team members are able to delete or leave chat groups they no longer wish to be a part of.

6. The chat feature is intuitive and easy to use."

1. Research and select a chat framework to integrate within the application.

2. Implement the chat feature within the application.

3. Allow users to create new chat groups and invite other users to join.

4. Implement real-time messaging and notifications

5. Allow users to delete or leave chat groups.

6. Test and debug the chat feature to ensure it is working properly.

7. Incorporate user feedback to improve the usability of the chat feature."

"

Notification As a user, I want to be able to receive notifications when I am invited to join a team, so that I can stay informed.

1. When invited to join a team, the user should receive a notification on their mobile device.

2. The notification should include the name of the team and a message inviting the user to join.

3. The notification should be clear and easy to understand, with a prominent call-to-action button to accept or decline the invitation.

4. If the user accepts the invitation, they should be directed to the team page within the app.

5. If the user declines the invitation, they should be removed from the invitation list and the team should be notified.

1. Implement a notification system within the app that can send notifications to users.

2. Create a database schema for storing teams and team invitations.

3. Implement a user interface for sending invitations to join a team.

4. Implement logic for sending notifications to users when they are invited to join a team.

5. Implement a user interface for viewing and responding to team invitations.

6. Implement logic for accepting or declining team invitations.

7. Implement a user interface for viewing teams and team members.

8. Update the team page to show new members who have accepted invitations.

Activity schedule 12. As a team member, I want to be able to view the team's activity schedule, so that I can plan accordingly.

1. The activity schedule should be accessible from the main menu of the application.

2. The activity schedule should display the date, time, location and description of each scheduled activity.

3. The activity schedule should be up-to-date and accurate, reflecting any changes or cancellations.

4. The activity schedule should be customizable, allowing users to add or remove activities as necessary.

5. The activity schedule should be easy to read and navigate, with a clear layout and font size.

1. Determine the appropriate format and layout for the activity schedule.

2. Implement a new menu item for the activity schedule in the application's main menu.

3. Develop a data model for storing the activity schedule information.

4. Implement functionality to add, edit and delete activities in the schedule.

5. Integrate the activity schedule with the rest of the application's features and data.

6. Design and implement the user interface for the activity schedule, including a clear layout and font size.

7. Conduct testing and debugging to ensure that the activity schedule functions as intended and is free from errors.

8. Incorporate feedback from users and stakeholders to refine and improve the activity schedule."

"

Team information check 13. As a team member, I want to be able to see the team list and information like announcement, schedule, requirements etc, so that I can get all information I need.

1 The team member can access the team list and see all team members' names and roles.

2 The team member can view announcements related to the team.

3 The team member can see the schedule of upcoming team events and meetings.

4 The team member can view any requirements or tasks assigned to them or the team as a whole.

5 The team member can filter the team list by role, location, or other criteria.

1 Design the user interface for the team list and information display.

2 Implement the backend to retrieve and display the team list, announcements, schedule, and requirements.

3 Add filtering functionality to the team list.

Create a database schema for storing team information.

4 Develop a notification system for new announcements and upcoming events.

5 Add a search bar to allow team members to search for specific team members or announcements.

6 Write unit tests to ensure the functionality is working as intended.

7 Conduct user testing to gather feedback and make necessary improvements.

8 Integrate the feature into the main application.

9 Provide documentation for team members on how to access and use the feature."

Account Delete 14. As a user, I want to be able to easily delete my account, so that my personal information is no longer stored in the app.

1 The user can find the option to delete their account under setting menu.

2 Clicking on the delete account option will prompt the user to confirm their decision.

3 Deleting the account will remove all personal information associated with the account.

4 The user will receive a confirmation message that their account has been successfully deleted.

1 Design the user interface for the delete account feature.

2 Implement the backend to delete the user's account and all associated data.

3 Create a confirmation prompt that verifies the user's intention to delete their account.

4 Ensure that all data associated with the user is deleted, including profile information and location data.

5 Provide a confirmation message to the user once their account has been successfully deleted.

6 Add logging functionality to record when and why an account was deleted.

7 Write unit tests to ensure the functionality is working as intended.

8 Conduct user testing to gather feedback and make necessary improvements.

9 Integrate the feature into the main application.

10 Provide documentation for users on how to delete their account and what to expect when they do so.

Friends system 15. As a user, I want to be able to search, add and delete friends on the app, so that I can connect with people I know.

1 The user can search for other users by name or username.

2 The user can send a friend request to another user.

3 The recipient of the friend request can accept or decline the request.

4 The user can see their list of friends and their online/offline status.

5 The user can delete a friend from their friend list.

1 Design the user interface for the friend search and management features.

2 Implement the backend to handle friend requests, acceptances, and deletions.

3 Create a database schema for storing friend relationships.

4 Implement a search functionality that allows users to search for other users by name or username.

5 Add a friend request system that allows users to send and receive friend requests.

6 Allow recipients of friend requests to accept or decline the request.

7 Create a list of friends that shows the user's current friends and their online/offline status.

8 Allow users to delete friends from their friend list.

9 Write unit tests to ensure the functionality is working as intended.

10 Conduct user testing to gather feedback and make necessary improvements.

11 Integrate the feature into the main application.

12 Provide documentation for users on how to search, add, and delete friends on the app."