SWENG 500 – Software Engineering Studio
Team 3 Requirements Document

Application Overview

There are many applications available for quickly finding a place to eat or something to do in a new area, but few are designed to help a group of people manage these simple tasks together. Teams traveling for work may have several individuals all figuring out plans on their own without an effective way to see who else is traveling with them or a mobile platform to share information about their trip with fellow teammates. Often a team finds themselves traveling to an unfamiliar location or needing to track and coordinate with team members quickly. Our system intends to meet this need by providing an easy way for teams, especially those on the road, to easily find contact information and schedule meetings and other events.

Our project will take advantage of mobile technology by implementing a native Android application and providing a web-based interface. The primary features that the system will provide to teams will include:

- Sharing team member contacts
- Tracking team member locations
- Sharing schedules
- Setting group tasks
- Tracking Travel Expenses

The system will allow users to participate in and/or manage several teams, sharing information and providing a centralized location for team contact information that is up-to-date as team members join or leave a team. Flexibility is designed into the system so that the application is useful to teams that are traveling or just teams that want to keep a centralized cache of information.

Objectives

Deploy a networked Android application and web browser client that will allow teams to coordinate with each other and improve the ability for active teams to work together and organize meetings and events.

Targeted System

The system has several components, each will be deployed on a specific system that they will be designed for:

- Android Application: Android mobile device running version 4.0 or above with at least 1GB of RAM, 4.8 inch HDPI screen, and run the standard suite of Google applications (Google Maps, Gmail, Contacts, Dialer, etc). Additional configurations including RAM less than 1GB, different pixel densities, and large tablet screen layouts will be supported and tested through emulation, but will not be considered the target system.

- Web Client: Internet Explorer, Google Chrome, Mozilla Firefox, or Apple Safari web browsers. Free disk space: 350 MB. RAM: 512 MB.
- Server: 2 CPU Cores - Intel(R) Xeon(R) 2.80GHz, 2GB RAM, 48GB SSD HD, Debian Linux 64-bit server running Apache2 Web Server, MySQL, and PHP
- Database: We will be using a MySQL database to store all the pertinent information that we want to store. The MySQL database will reside on our server.

Nonfunctional Requirements

- The system shall be able to store information on at least 100 teams
- A managed team shall have at least one manager
- A managed team shall require permission granted by a manager for each user who joins
- The system shall support the use of managed and unmanaged teams

Functional Requirements

- The system shall utilize username and password to identify users
- The system shall use username to reset a user's password using email notification
- The system shall collect user location at intervals
- The system shall allow users to act as members of one or more teams
- Users shall be able to create, join, and leave teams
- Team managers shall be able to authorize users to join a team, remove users from a team, assign current team members as additional team managers, and leave the team

Use Cases

Use Case	**Creating an Account on the App**
Description	When a user first joins the system, they must create an account so that their information can be tracked by the system.
Steps	1. Open application 2. Select option to create a new account 3. Enter username 3.1. If username is taken, user is prompted to enter another name 4. Enter password 5. Confirm password 6. Enter first and last name 7. Enter contact information 8. Submit information 9. User account is created 10. User is logged into app and may begin use
Assumptions	User is creating an account using the Android Application

Use Case	**Log into App**
Description	The user must enter authentication to log into the system the first time they use the app, if the “remember me” option is not selected, or if they have logged out from the application. If the user chooses to remain logged in, the app will remember their authentication details and log in automatically.
Steps	1. Open application 2. Select option to log in to an existing account 3. Enter username 4. Enter password 5. Select preference for staying logged in on the app (default is yes) 6. Select log in 6.1. If login is rejected, the user is provided a message and must enter authentication again 7. User is authorized and may use system
Assumptions	User is logging into Android app where they are currently logged out

Use Case	**Creating a Team**
Description	Any user may create a team. Teams are either unmanaged (public) or managed (private). Managed teams will require permission for users to join.
Steps	1. Select option to create a new team 2. Enter team name 3. Enter team location name 4. Enter team description 5. Select if the team will be managed 5.1. If managed team is selected, user is informed that they will be automatically assigned as a manager and that a manager must authorize each user who wishes to join the team 6. Submits team information 7. Team is created and may be discovered by other users
Assumptions	User has already opened and logged into the Android application

Use Case	**User Joins a Team**

Description	A user must join a team to access team information. Teams may either be unmanaged (public) or managed (private).
Steps	1. Select option to join an existing team 2. Application displays a list of existing teams 3. Find team to join 4. Select team 4.1. If team is managed, user is informed that they will need manager permission before they can view team information 4.2. User team membership enters a pending phase awaiting approval 4.3. Manager approves or disapproves the user’s request to join 4.3.1. If request is rejected, user request is removed 4.3.2. If request is accepted, continue with step 5 5. User is added as a team member 6. User accesses team information
Assumptions	User has already opened and logged into the Android application

Use Case	**User Leaves a Team**
Description	A user who is a member of a team may leave the team at any point. Depending on the user role or if this is the last user leaving a team, the team may be deleted as a result.
Steps	1. User selects option to leave team 1.1. If user is the only manager of the team, they receive a warning that this action will delete the team and are given the option to continue, cancel, or select a new manager 1.1.1. If the user chooses to select a new manager, a list of team members is displayed 1.1.2. User selects team member as manager 1.1.3. Team member is set as manager 1.1.4. Team member receives an email notifying them of manager status 1.2. If user is the last team member, they receive a warning that this action will delete the team and are given the option to continue or cancel 2. User confirms action 3. User is removed from team

	3.1. If user was the last manager or the last user, the team is deleted
Assumptions	User has already opened and logged into the Android application and is viewing the team information for the team they wish to leave.

Use Case	**Team Member Views Team Contact Information**
Description	A common feature that team members will use is accessing the contact information for other team members. Other team information will be accessed in a similar process.
Steps	1. Open application 1.1. If user is logged out, select log in option 1.2. Enter authentication credentials 2. Select desired team 3. Application displays team overview information 4. Select option for team member list 5. Application displays list of team members with their contact information and contact options
Assumptions	User is already a member of the team and has an account. User is accessing the Android application.

Use Case	**Manager Removes a Team Member**
Description	Any team manager has the ability to remove team members from a team. This can be used to remove members that have left a team and forgot to remove themselves or as disciplinary action.
Steps	1. Manager selects option for team member list 2. Application displays a list of current team members 3. Manager selects the member they wish to remove 4. Application displays team member details 5. Manager selects option to remove member 6. Manager confirms action 7. Team member is removed from the team
Assumptions	User is a manager of the team and is viewing the team information for the team he/she wishes to remove a member from. User has opened and logged in to the Android application.

Use Case	**Manager Deletes a Team**

Description	Any manager may delete a team. This can be completed to remove an incorrect or redundant team or at the termination of the team’s campaign.
Steps	1. Manager selects option to delete team 2. Manager confirms action 3. Team is deleted and all members are removed
Assumptions	User is a manager of the team and is viewing the team information for the team he/she wishes to delete. User has opened and logged into the Android application.

Use Case	**User Wishes to Add Expense Item (without OCR)**
Description	Any user, using the Android application, should be able to add an expense item to their expense report.
Steps	1. User selects option to add a receipt to the expense report 2. User is presented with camera functionality to take picture of receipt 3. User inputs data for the following: location of the purchase, type of purchase, dollar amount, and an optional description or note on the purchase. 4. User confirms all pricing or may re-edit the form if information is incorrect. 5. User confirms changes and submits expense and reciept.
Assumptions	User is logged into the Android application, is viewing the team information where the report should be uploaded, and has a receipt to collect information from

Use Case	**User Wishes to Add Expense Item (with OCR)**
Description	Any user, using the Android application, should be able to use the phone’s camera in order to take a photo of a receipt in order to add it to their expense report.
Steps	1. User selects option to add a receipt to the expense report 2. User is presented with camera functionality to take picture of receipt 3. System pulls description and price information from the receipt and presents the information to the user in the shape of a form 4. User confirms all pricing or may edit the form if pricing is incorrect. 5. User confirms changes and submits receipt.

Assumptions	User is logged into the Android application, is viewing the team information where the report should be uploaded, and has a receipt to take a picture of.

Use Case	**Team Member Views another Team Member’s GPS Location**
Description	A common feature that team members will use is accessing the latest GPS position for other team members.
Steps	1. Open application 1.1. If user is logged out, select log in option 1.2. Enter authentication credentials 2. Select desired team 3. Application displays team overview information 4. Select option for team member list 5. Application displays list of team members with their contact information and contact options 6. Select “Show GPS Location” option 7. Application displays geographical map with location of the team member along with the current location of the user
Assumptions	User is already a member of the team and has an account. User is accessing the Android application.

Use Case	**Team Member Suggests Dinner Location to the Team**
Description	Team members will be able to use the app to find different restaurants to go to for breakfast, lunch, and dinner.
Steps	1. Open application 1.1. If user is logged out, select log in option 1.2. Enter authentication credentials 2. Select desired team 3. Select communications option 4. Select send idea for food. 5. Application contacts server using the user’s GPS location to identify resturant locations within a user defined distance away from the user. 6. Server uses External API to find locations within specifications. 7. Server sends response back to user with different locations. 8. The user chooses a restaurant and selects “send to team.” 9. The team is notified of a restaurant choice and the team members choose if it is acceptable or not.

Assumptions	User is already a member of the team and has an account. User is accessing the Android application.