

Tourist Assist App

Final Project Report

Team Members (Team M)

Mohammed Shethwala (A20368337)

Nishant Sharma (A20368125)

Sai Ravali Nannuru (A20354346)

Gan Xu (A20338959)

Ericson Akatakpo (A20349354)

Table of Contents

1		Objective	3
2		Categories of Users	3
3		Requirements Specification	4
	3.1	Use Case Diagram	6
	3.2	Use Case Scenarios	9
	3.3	Functional Requirements	15
	3.4	Non-Functional Requirement	20
	3.5	System Requirements	21
4		High Level Design	22
	4.1	Component Model	22
	4.2	System Architecture Model	23
	4.3	Storage Model	24
	4.4	Data Model	25
5		Test Planning	28
6		Test Cases & Test Results	29
	6.1	Functional Requirements Test Cases & Test Results	29
	6.2	Non-Functional Requirements Test Cases & Test Results	62
	6.3	Usability Test Cases & Test Results	64
7		Quality & Value	68
8		Marketing	69
9		Analysis of Project Team Performance	70
10		Prototype Documentation & User Guide	71

1. Objective

The Tourism industry has been the most rapidly growing industry and is by far the most exciting industry in the modern era. Every day, thousands of tourists visit the places of their interest with their family and friends. Almost every tourist needs some sort of assistance for a particular trip. The Tourist Assist App is a proposed application which will benefit every tourist for planning their trip and being assisted throughout the trip. The ***Tourist Assist App*** will be a mobile app which can be downloaded by the users on their *iPhone* and *Android* Devices.

The application will have a variety of functionalities which will be divided into modules for the ease of designing, prototyping, developing and testing the app.

2. Categories of Users

The Tourist Assist App will have two categories of users which are as follows:

1. Tourists (End-Users): There are a wide number of end users who will be using the tourist assist app. Even more, the types of users are also diverse based on the cultural, educational and geographical diversity. So, the users who will use the application are categorized into different categories as follows:

1. Nationality

- **National:** For a tourist location in any country, these are the tourists which are the people living in the country itself and thus are local to the country. So, these tourists are more familiar about the national language, culture and the currency of that location.
- **International:** These tourists are not local to the country of the destination and thus are unfamiliar with the culture, climate, currency and language of the country. They need more assistance regarding the language, culture, diversity and climate of the location in a foreign country.
- **Local:** These tourists are from the same state of the planned destination and are almost completely familiar with the local surroundings.

2. Age

- **Young (12-24 years):** This type of tourist travel with friends mostly. They are enthusiastic, adventurous and energetic. They are very well familiar with the mobile applications and would easily understand the app without much help needed to educate them.
- **Adults (24-50 years):** Majority of them married, and like to travel with their families. Prefer more facilities and the app may need to provide certain help to such type of tourists regarding the usage of the application.
- **Seniors (>50 years):** Mostly retired and like to travel with their spouses or families. May need help at many points while using the app.

3. Others

- **Adventurous:** Highly adventurous and energetic tourists. They are mostly interested in adventurous trips and would prefer camping, mountaineering, trekking rather than local sight-seeing.
- **Families:** Typically interested in local attractions and would prefer nice hotels with facilities for accommodation.
- **Business:** Some people might want to use this app while on a business trip. These users are widely interested in hotels, cafés and restaurants rather than attractions.

2. Admin: The Admin has the main authority of managing the entire app. The Admin adds the destinations, attractions, restaurants and other such information which is required by the app. Admin's role is very important in the back-end of the app.

3. Requirements Specification:

The User Analysis and Requirements Specification has been divided into the following sections:

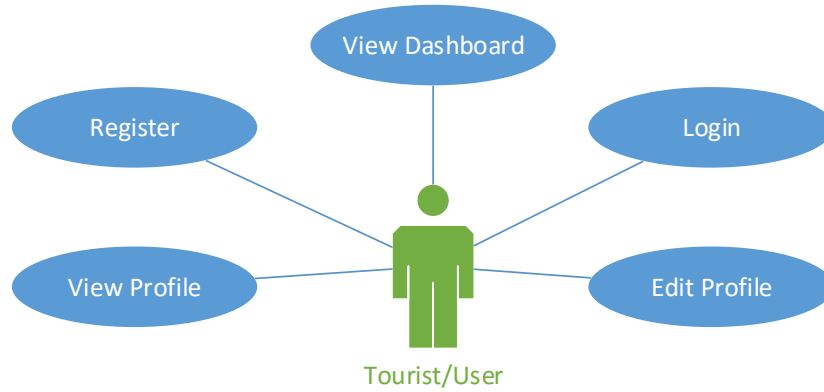
- **Use Case Diagram:** The Use Case Diagram depicts the overall view of the system consisting of use cases for different components of the system. It gives a complete overview of high level user requirements through the user's interaction with different modules.
- **Use Case Scenarios:** the Use Case Scenarios further describes in detail each of the use cases mentioned in the Use Case Diagram. This enhances the requirements specification and makes it easier to specify the functional and non-functional requirements further.
- **Functional Requirements:** The functional requirements are then written down based on the use cases and components specified in the use case diagram

- **Non-Functional Requirements:** Non-Functional requirements are then written considering the system as a whole.
- **System Requirements:** System Requirements deals with the type of environment which the application will run in.

Each of the above discussed sections are further elaborated in the following pages.

3.1 Use Case Diagram

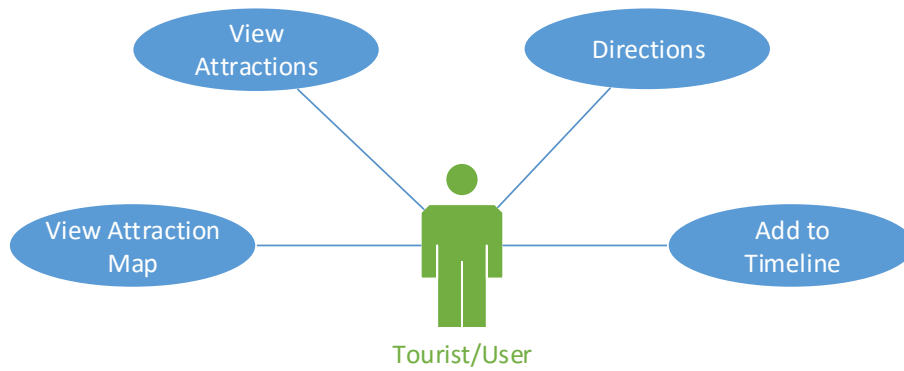
Profile



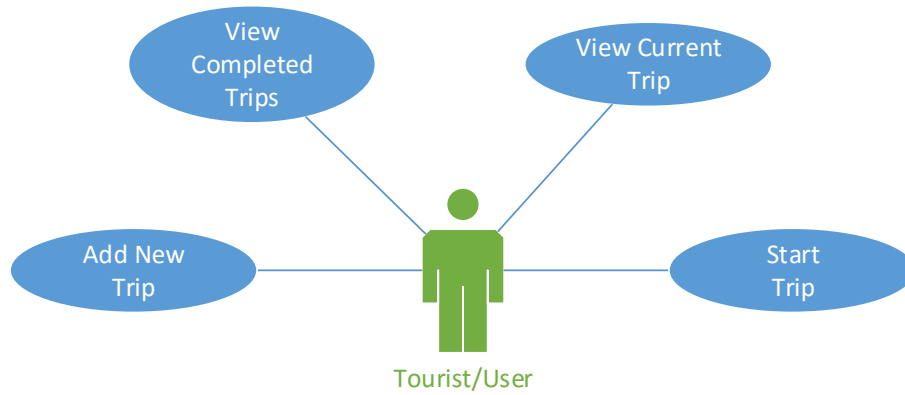
Destinations



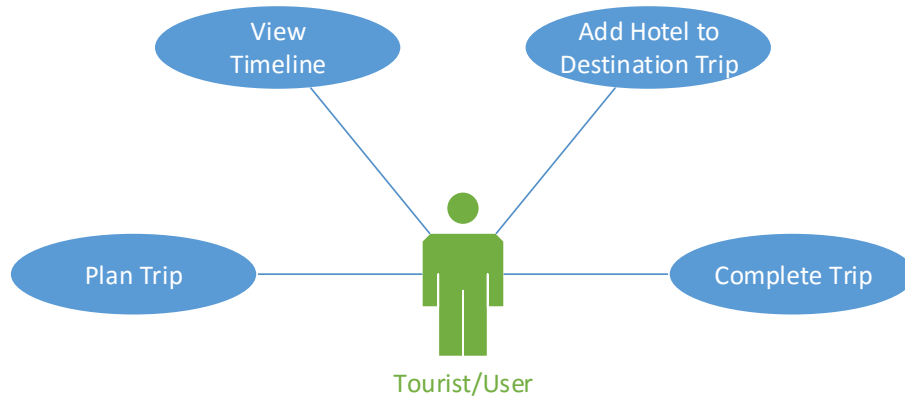
Attractions



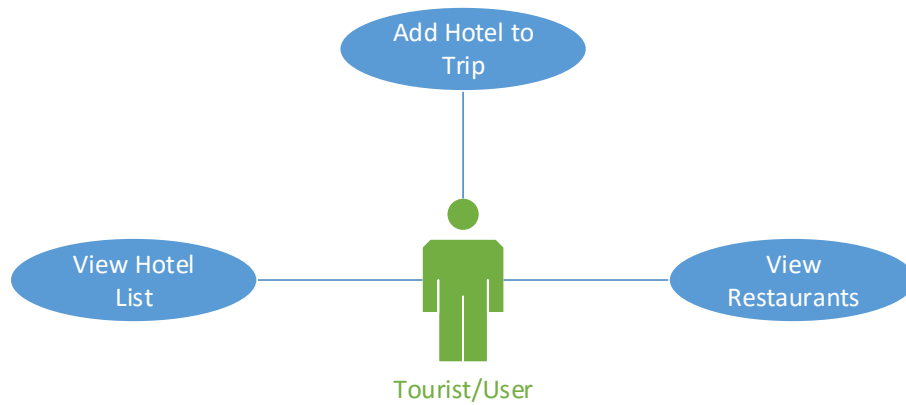
Trips



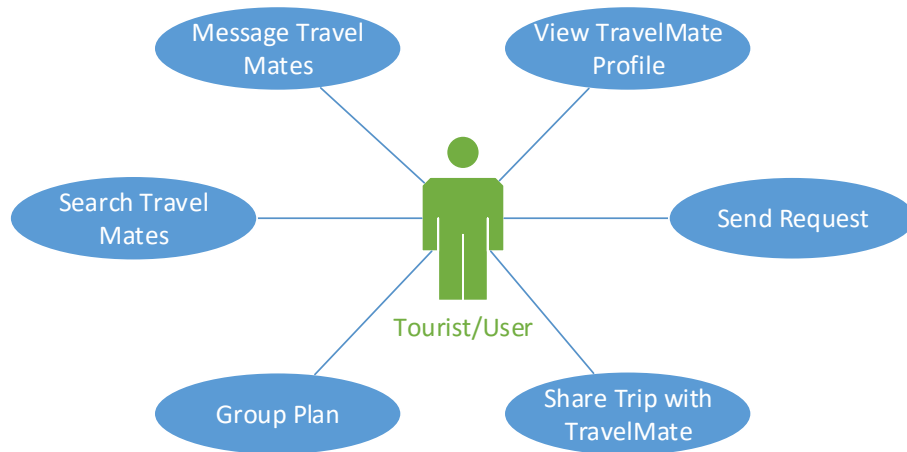
Current Trip



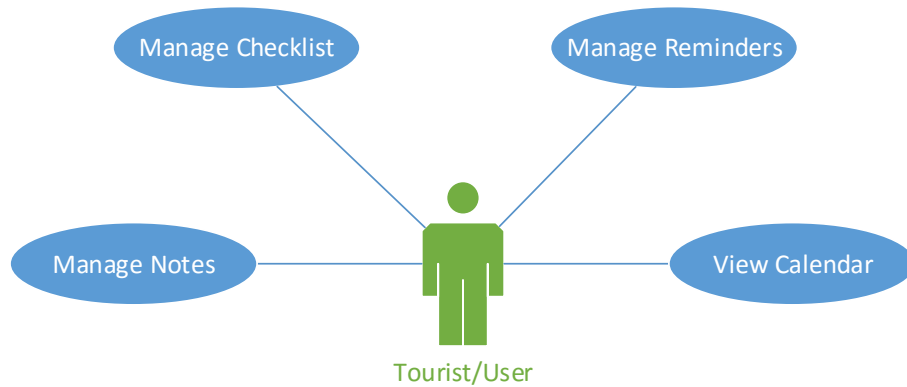
Accommodation & Food



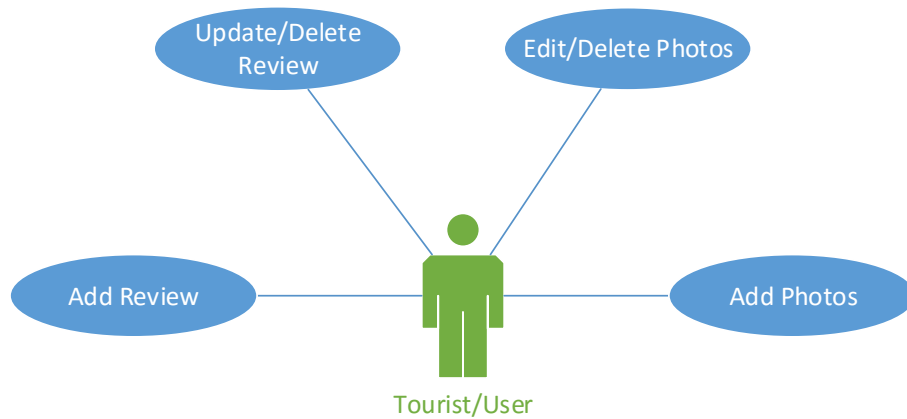
Travel Mates

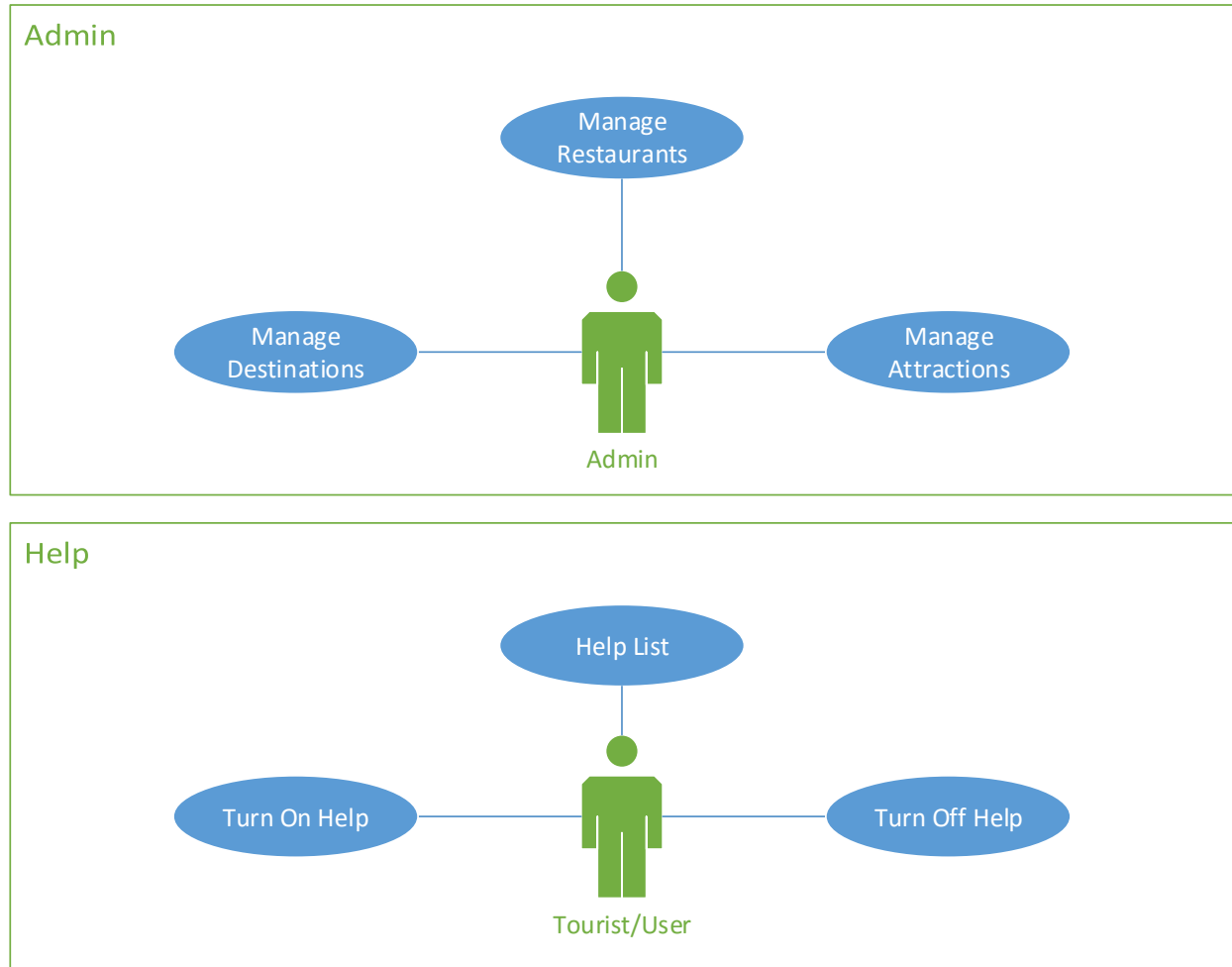


Utilities



Reviews & Photos





3.2 Use Case Scenarios:

Profile:

1. **Register:** A new user enters name, email address, age, country, state, city, phone number in the respective fields and uploads a profile photo if the user wants to upload. The username is the email address provided by the user. User submits the registration. System sends a verification link to the email address specified by the user. User clicks the link in email sent to the user. After link is clicked, system shows successful registration in the app.
2. **Login:** User enters username and password. System validates the credentials and displays the user's dashboard if the credentials are valid. If not, then system displays "Login failed" and prompts to enter the valid username and password for login.

3. **View Profile:** User clicks on 'Profile'. System displays the user's profile which has user's information like Name, Country, State, City, Profile Photo, Total Trips, Total Reviews and Total Destinations.
4. **Edit Profile:** A user clicks on 'Edit Profile'. System displays editable user's information. User then edits and updates his information and clicks 'Save Changes'. System successfully saves the updated user profile to the database and displays the profile.
5. **View Dashboard:** User clicks on 'Dashboard'. System displays the dashboard which includes completed trips, current trips, reminders, notes, checklists and trip timeline.

Destinations:

1. **Search destinations:** The user enters the name of a tourist location. The system checks if the tourist location exists. If yes, then the app displays the destination information to the user. If not a valid destination, the app displays "destination not found".
2. **View Destination:** The user selects a destination. The app displays destination information, destination overview, photos, reviews, attractions, cultural diversity, climatic conditions, and currency.

Attractions:

1. **View attractions:** The user clicks on the attractions in a destination. The system displays the list of attractions at the selected destination. The attraction information includes the description, photos, rating and reviews of the attraction. It also displays the distance from the user's hotel to the attraction.
2. **View Map of attractions:** User clicks on the 'View Map'. The system displays a map which includes all the attractions around this attraction in that destination.
3. **Add attraction to trip timeline:** User selects an attraction. User clicks 'Add attraction to trip'. User selects the trip in which the attraction is to be added to the destination. System adds the selected attraction to the list of attractions which has to be visited by the user in the destination of a current trip.
4. **Directions:** User clicks on 'directions' in an attraction. System opens a GPS Map and prompts the user to enter the source address. User enters the source address. If the source address is valid, the system displays the directions from the source to the attraction in the Map.

Trips:

1. **Add New Trip:** User initiates a new trip. System displays the new trip page. User selects the destination of the trip. If the user wants to go more than one destination then the user adds more destinations to the trip. User adds the dates for every destination in the planned trip. User selects the hotel for all the destinations in which the user is going to stay during the trip. The user then clicks submit. System saves the user's new trip and updates the information on dashboard.
2. **View Completed Trips:** User clicks on 'Completed Trips'. System displays all the previously completed trips by the user in a list. User can then view any of the completed trip information.
3. **View Current Trip:** User clicks on 'Current Trip'. System displays the current trip of the user. It shows the timeline of the trip, completed destinations, remaining destinations, visited attractions, remaining attractions, photos from current trips and restaurants to visit.
4. **Start Trip:** The user selects a trip from the planned trips list. User clicks on 'start trip'. System moves this trip from planned to current trip. System updates the current trip section and makes this trip as the current trip. System updates it on dashboard also.

Current Trip:

1. **Plan Trip:** User select to plan the current trip. User can add or remove the destinations from the current trip. User can change and update the dates set for the trip and every destination. User can add attractions to visit, prioritize destinations, attractions and plan restaurants to go in calendar. User can plan the trip date wise in the calendar. User can add events in the calendar for the places to visit on that particular day.
2. **View Timeline:** System displays a complete timeline of the current trip in a chart fashion with all the completed destinations, attractions, restaurants and all the remaining ones as planned in the current trip.
3. **Add Hotel/Stay in selected destination:** User can add/remove hotels from the destinations which the user is currently going to visit.
4. **Complete Trip:** User selects a current trip. In current trip, user selects to complete the trip. System confirms the user if he surely wants to complete the trip. User confirms and completes

the current trip. System moves the current trip to the completed trip and updates the dashboard and removes current trip from the dashboard.

Accommodation & Food:

1. **View Hotel List:** System displays all the available hotels in the destination selected by the user.
2. **Add Hotel to Trip:** User clicks 'Add hotel to trip'. System then adds the hotel to the destination in the trip.
3. **View Restaurants:** User selects to view the list of restaurants in the destination. System displays all the restaurants located at the particular destination of the trip. User selects the restaurant. System displays restaurant info like contact details, operating hours, directions, menu, reviews and photos.

TravelMates:

1. **Search TravelMates:** In a planned or current trip, the user searches for the TravelMates going to the same destination at the same time as the user. System checks for all the TravelMates travelling at the same destination on same dates as the user and displays the list of TravelMates.
2. **Message TravelMates:** In the TravelMates profile, user selects to send message to the TravelMate. System opens up a message box. User enters the message and sends the message. System sends the message to the TravelMate.
3. **View TravelMate Profile:** User selects a TravelMate from the list of available TravelMates for a selected trip. System displays the profile of the selected TravelMate.
4. **Send/Accept Request to TravelMate:** User clicks on 'send request' in the profile of a selected TravelMate. System displays a message box for the user. User writes about the trip and submits the request to the TravelMate. System sends the request to the TravelMate. Similarly, the user also can accept or decline a friend request, sent by another user.
5. **Share trip with TravelMate:** After acceptance of the request from a TravelMate, User adds the TravelMate to a planned trip. System adds the selected planned trip to the dashboard of the TravelMates. The TravelMate can also edit the trip along with the user. After sharing a trip with TravelMates, all the users can simultaneously make changes to the trip and share their thoughts about the trip.

6. **Group Plan:** Here, all the TravelMates in a trip can plan and discuss about the trip. User selects Group Plan from the tab. System opens up a group conversation. User enters the topic of discussion and submits. System adds all the users to this group conversation and also adds the conversation to each and every group member's profile.

Utilities:

1. **Manage Notes:** User can create a note anywhere in the planned trip or current trip. User creates a new note, edit a previous note or delete a note from the planned or current trip. System creates, edits or deletes the notes as specified by the user.
2. **Manage reminders:** User can add reminders for any event during a trip. User selects to set a reminder from the Reminders page. User enters the date, time and reminder text for the reminder. System saves the reminder and notifies the user at the time set for the reminder through the system.
3. **Manage checklist:** User can add a checklist for a current trip or a planned trip. User selects to add a checklist from the checklist tab. User adds the items to the checklist one by one. System displays the items of checklist in a list which can be edited by the user once the item is completed. User checks the item which is completed. System removes it from the checklist and adds it to the completed items.
4. **View Calendar:** User selects the calendar tab. System displays monthly calendar with the trips and events planned by the user in the planned trips. Planned trips are showed in the calendar from starting date to end date. User can select a trip from list of planned trips to view more detailed calendar for that trip.

Reviews & Photos:

1. **Add Review:** User selects to add a review of destination, attraction, restaurant, TravelMate. The review section is provided in destination, attraction and restaurant pages. The user can write a review for any of the sections. System saves the review and updates the review to the section to which review was added and also adds it to the user's profile.
2. **Update/Delete Review:** User selects the review from any of the places reviewed by the user. User selects to delete or update the review. System saves the deletion or updating of the review and updates it to the list of reviews as well as to the user's profile.

3. **Add Photos:** Add photos is supported for the following pages: destination, attraction, restaurant and user's personal photos for trip memories. User selects to add photos from any section from above. System prompts the user to add photos from the photo gallery. User selects the number of photos from the gallery and submits. User can also write caption for the photos. System saves the photos and updates it to the places where the user added photos. Also the photos are added in the user profile.

4. **Edit/Delete Photos:** User selects the photos to be edited or deleted. User edits or selects to delete the selected photos. System deletes the photos if the user selected to delete or updates the photos if the user edited the photos.

5. **Comment:** User selects any review or photos from the list of reviews or photos of a destination, attraction, restaurant or a different user. User selects to comment on the review or photo selected. System provides a text box for the user to comment on review or photo. User comments and submits the comment. System accepts the comment and updates it to the comments section of the review or photo.

Admin:

1. **Manage Destinations:** The Admin can add, remove and edit the destinations and edit the destination information. Admin adds, edits or deletes a destination. System saves it and updates it to the list of destinations

2. **Manage Attractions:** Admin adds, updates or deletes an attraction inside a particular destination. System updates the attraction and adds it to the list of attractions.

3. **Manage Restaurants:** Admin adds, deletes or updates a restaurant. Admin adds menu, location, photos, timing to the new restaurant. System saves and updates to the list of restaurants.

Help:

1. **Help List:** The user can navigate through the list of help sections provided for each and every functionality of the app. The user selects the help list from the tab. System displays module wise collapsed help sections. User selects one help module. System displays the help sections of that module. There are more help images which can be navigated by the user if the user wants to understand more about a functionality.

2. **Turn on Help:** User turns on help from profile. System helps user in every part of the application. Whenever a user clicks on a particular button or enters in a module, the system displays the help section for the app and also displays usability view for that section. User can read and see the help or user can dismiss the help for that section.
3. **Turn off Help:** User turns off help from profile. System stops adding help sections to the user.

3.3 Functional Requirements:

1. Profile

- 1.1. User shall register with the system after entering name, email address, phone number, password, country, state, city and uploading profile picture. Email address of the user will serve as username for future login purposes.
- 1.2. After getting a registration request from the user, system shall send an email to the email address specified by the user with a link to successfully register with the system. The system shall store the user information in the database.
- 1.3. The user shall be able to successfully register with the system after clicking on the link specified in the email address. System shall successfully register the user and add it to the database.
- 1.4. The user should be able to login with Facebook if the user does not want to create a new email id profile. System shall display user with the Facebook login options and if selected by the user, the system should authenticate the user as a verified Facebook user and add it to the database.
- 1.5. The user shall be able to login to the system successfully after providing valid username and password to the system. The system shall successfully login the user if the credentials are valid. If username and password are not valid, the system shall prompt the user to enter the proper combination of username and passwords. The system shall display option of 'Forgot Password' if the user forgets password. After clicking forget password, system shall send a password link to the email address of the user. The user shall use that password for first time and after logging in, the system shall prompt the user to enter new password again.

1.6. After successful login to the system, the user shall be able to see the dashboard on the home screen. The dashboard screen is the home screen. The dashboard shall display current trip, completed trips, upcoming trips, notes, checklists and reminders related to the trips. If user clicks on any of the above tabs then system shall take the user to the selected page of notes, reminders, checklist etc.

1.7. Once the user clicks on 'View Profile', the system shall display the User profile to the user. The User profile must have user's profile photo, name, country, state, city, TravelMates, total trips, total reviews and total photos of the user.

1.8. The user shall be able to edit the user profile. System should display editable fields if the user selects to edit the user profile. The system shall save the edited information to the database and update the user profile in the app.

2. Destinations

2.1. The user shall be able to search destinations after entering a valid destination name in the search bar. The user can also select to view destinations based on the filters provided by the user. The system shall be able to filter destinations based on country, state, city, destination type, cost estimates, currency, weather conditions and destination types (Adventure, Family, Trekking, Hiking, Beach, Mountain, Casino and Heritage). After filtering the result, the system shall display all the destinations which come in the specified filter.

2.2. After searching for one destination or list of destinations, the user shall click any one destination and clicks on the destination. The system shall display the home page for the selected destination. The destination page should include cover page of destination, destination information, photos, reviews, total app users visited and attractions of that destination.

3. Attractions

3.1. User shall be able to click on any attraction from the list of attractions in a destination. User shall also filter the attractions based on the type of attractions (Family, adventurous, beach, mountain and hiking). User shall click on any one attraction. System display the homepage of

that attraction. The homepage must consist cover page, photos, reviews, attraction information and timings.

3.2. The user shall be able to see the map of attractions in a destination. User shall click on view map of attractions. The system shall display the map of attractions which should include all the attractions in a destination in a map. The Map should have pins at the attractions in a destination. The map should be easy to understand and navigate by the user.

3.3. User shall be able to add an attraction to a trip, if he wants to visit a place in future, he. He shall plan his trip accordingly then, and should help him when he actually visits the place.

3.4 User shall be able to use GPS, if he wants to drive up to the attraction. GPS would help the user in reaching the destination easily and would give him the option to choose between different routes.

4. Trips

4.1 User shall be able to add a new trip, if he is planning a trip in future. He shall then be able to see attractions and restaurants for the place he wants to visit. This shall help the user to plan his trip, decide the attractions, look for hotels, restaurants etc.

4.2 User shall be able to see all his completed trips, if he wants to see the previously completed trips. He shall be able to see all the information of previously completed trips like places he visited, restaurants etc.

4.3 User shall be able to details of his current trip. He shall be able to check the places of attractions, restaurants and hotels for his current trip. He shall be able to see his completed destinations and remaining destinations of the current trip.

4.4 User shall be able to start his planned trip, when he is ready to start his planned trip. Now his planned trip which he has started will be changed to current trip.

5. Current Trip

5.1 User shall be able to modify his current trip. He shall be able to add/remove/edit places he wants to visit. He shall be able to change date and time, priority, for any destination.

5.2 User shall be able to see the complete timeline of his trip. He shall be able to see the places he has visited and the places remaining. He shall also be able to see the day to day plan of his trip.

5.3 User shall be able to modify his stay during the trip. He shall be able to add and delete hotels.

5.4 User shall be able to complete his current trip, after he has completed his trip. The current trip will now be changed to completed trips.

6. Accommodation and Food

6.1 User shall be able to view the list of the hotels available in the destination area selected by him. He shall be able to see all the information about the hotel i.e. address, phone etc.

6.2 User shall be able to add a hotel to his trip. He shall be able to add the hotel in which he wants to stay.

6.3 User shall be able to see the list of restaurants in the destination area selected by him. He shall be able to sort by restaurants type i.e. Italian, Chinese, Indian etc. He shall also be able to see the contact information, working hours, etc.

7. TravelMates

7.1 User shall be able to search for TravelMates for his planned or current trip. He shall be able to the list of the people who are visiting the same destination at the same time as the user.

7.2 User should be able to send messages to the TravelMate, so that he is able to interact with his TravelMate.

7.3 User should be able to view the profiles of all the TravelMates.

7.4 User shall be able to add his TravelMates to his profile, he shall be able to send friend request to his TravelMate. Similarly the user, shall also be able to respond to a request sent by another user.

7.5 User shall be able to share his trip with his TravelMates. He shall be able to add a TravelMate to a trip. The TravelMate shall also be able to see all the details of the trip that was created by the user.

7.6 User shall be able to discuss about the trip with all his TravelMates. He shall be able to communicate with all the members of the group.

8. Utilities

8.1 User shall be able to create a note in a planned trip or a current trip. He shall be able to modify previously created notes.

8.2 User shall be able to create a reminder for a trip by selecting the reminder date, time and adding reminder text. He shall be able to modify the previously created reminders.

8.3 User shall be able to manage his checklist for a planned or current trip. He shall be able to add or delete items in the checklist.

8.4 User shall be able to view the calendar for the current month with the trips planned in that current month.

9. Reviews & Photos

9.1 User shall be able to add a review for a destination, attraction, restaurant or TravelMate.

9.2 User shall be able to modify or delete his previous review.

9.3 User shall be able to add photos for destination, attraction, restaurants or his personal pictures from his trip. User shall also be able to upload multiple photos at the same time and also write a caption for every picture.

9.4 User shall be able to modify or delete his previously added pictures.

9.5 User shall be able to comment on a review or a photo uploaded by a different user.

10. Admin

10.1 Admin shall be able to manage all the destinations. He shall be able to add or delete a new destination. He shall also be able to modify the information of a destination.

10.2 Admin shall be able to manage attractions. He shall be able to add, delete or modify attractions.

10.3 Admin shall be able to manage restaurants. He shall be able to add, delete or modify restaurants.

11. Help

11.1 User shall be able to navigate through the help sections provided for every functionality of the application.

11.2 User shall be able to turn on help for a particular or every section of the application. The user should be able to get help from the application for every step that the user performs.

11.3 User shall be able to turn off the help.

3.4 Non-Functional Requirements

1. **Performance (response time and throughput):** The system should have short response time especially in the case of multi-user interaction (e.g. thousands of user's login or search concurrently).

2. **Capacity and Scalability:** The system should guarantee sufficient storage space for future extensions. The system can change its size accordingly to meet the user's demand.

3. **Usability:** The interface should be user-friendly, even children, elders or people without much experience in operating smartphones can enjoy the app without difficulties.

4. **Recoverable:** If the application terminates incidentally, its services should be able to recover.

5. **Security:** The system must guarantee information security to avoid user profile from leaking.

6. Authentication, Monitoring and Administration: The application should have different privilege level for the system, admins and users, which means they can access different kind of data and operations. The system should monitor the application process and take actions if errors take place. Admins may add and remove users as well as their privileges.

7. Data Integrity: The system must provide correct, reliable and up-to-date information.

8. Portability and Accessibility: The application should support different platforms (Android and iOS).

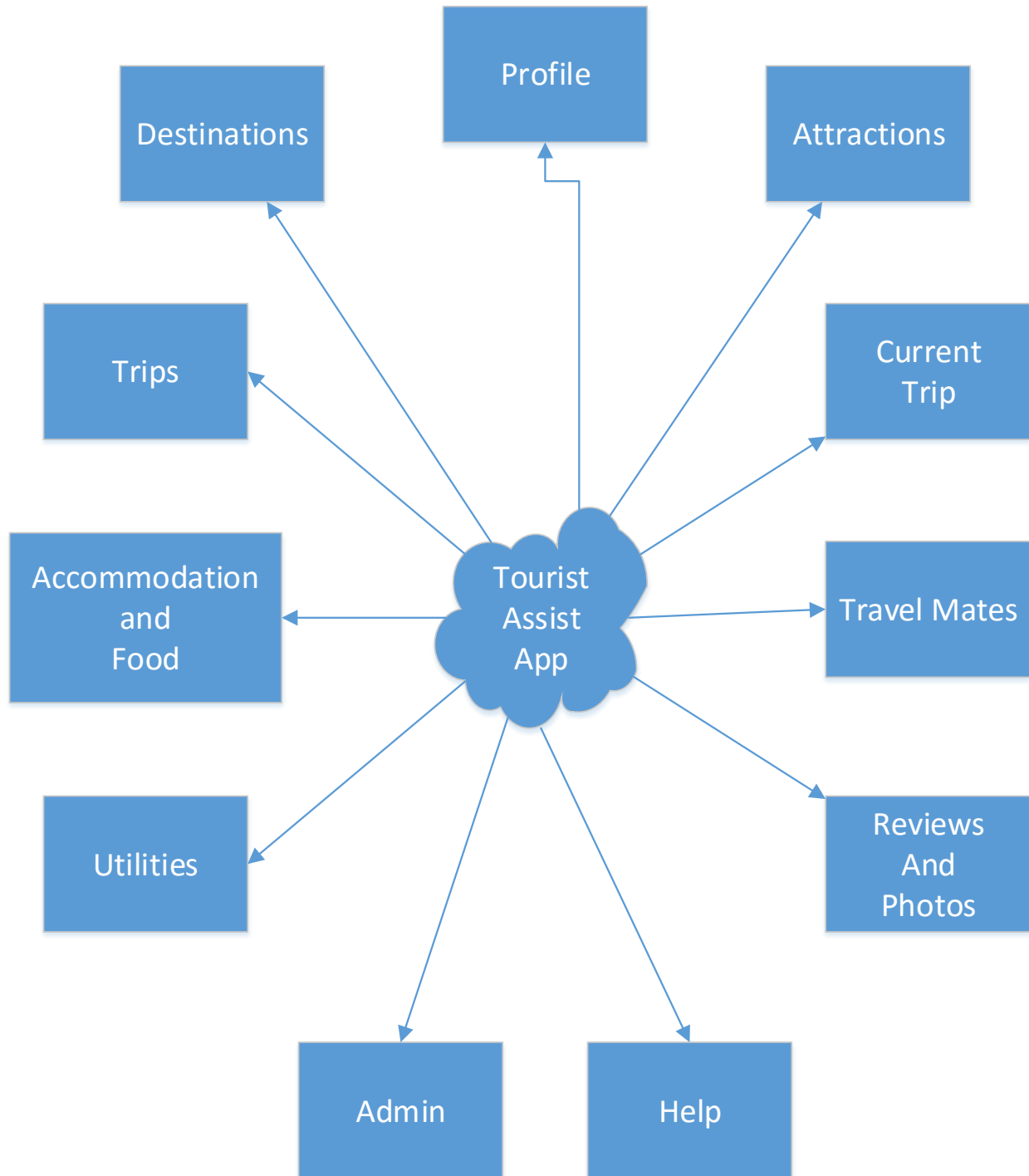
9. Reusability: Modules can be used in other places and add new features with slight modification.

3.5 System Requirements:

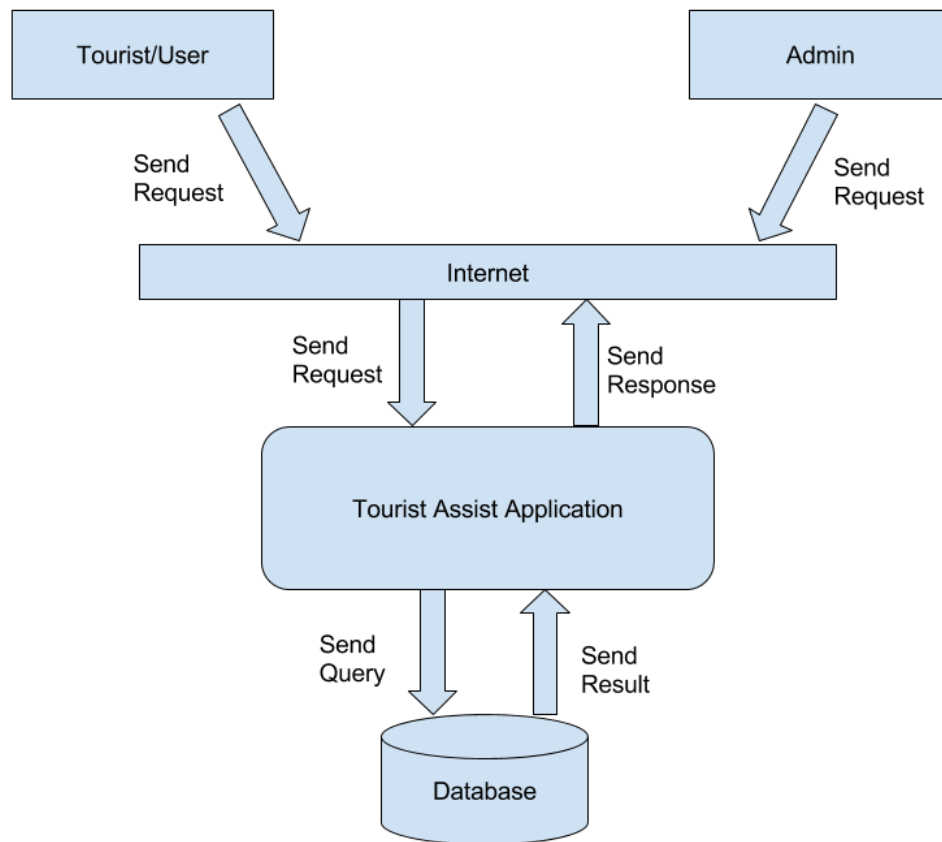
1. The application will run on all iPhones having iOS 8.1 and later version. The phone must have minimum of 1GB memory.
2. The application will run on all Android devices having version Jellybean and later. The Android phones must have minimum of 1GB memory.
3. Both Android and iPhone users must have minimum of 50mb space left for the application.

4. High Level Design

4.1 Component Model:

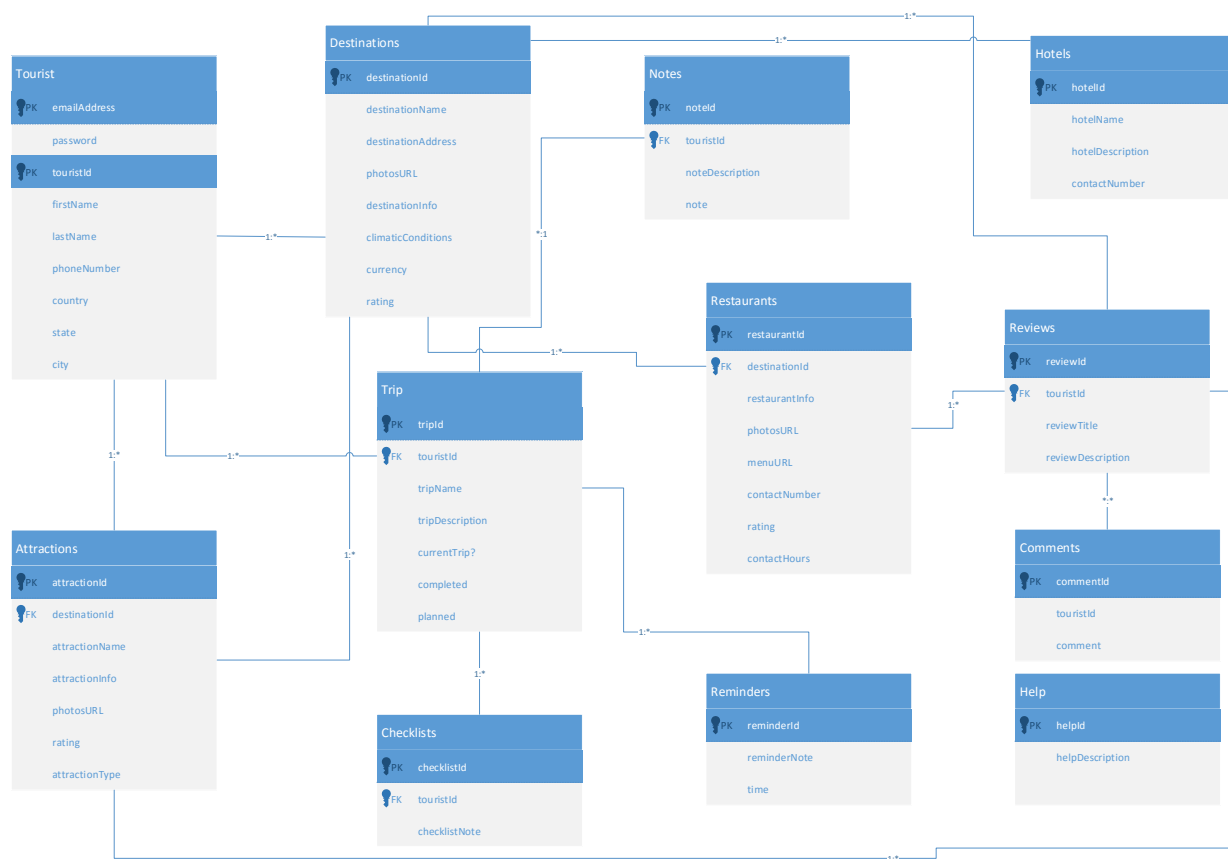


4.2 System Architecture Model



4.3 Storage Model (Entity-Relationship Diagram)

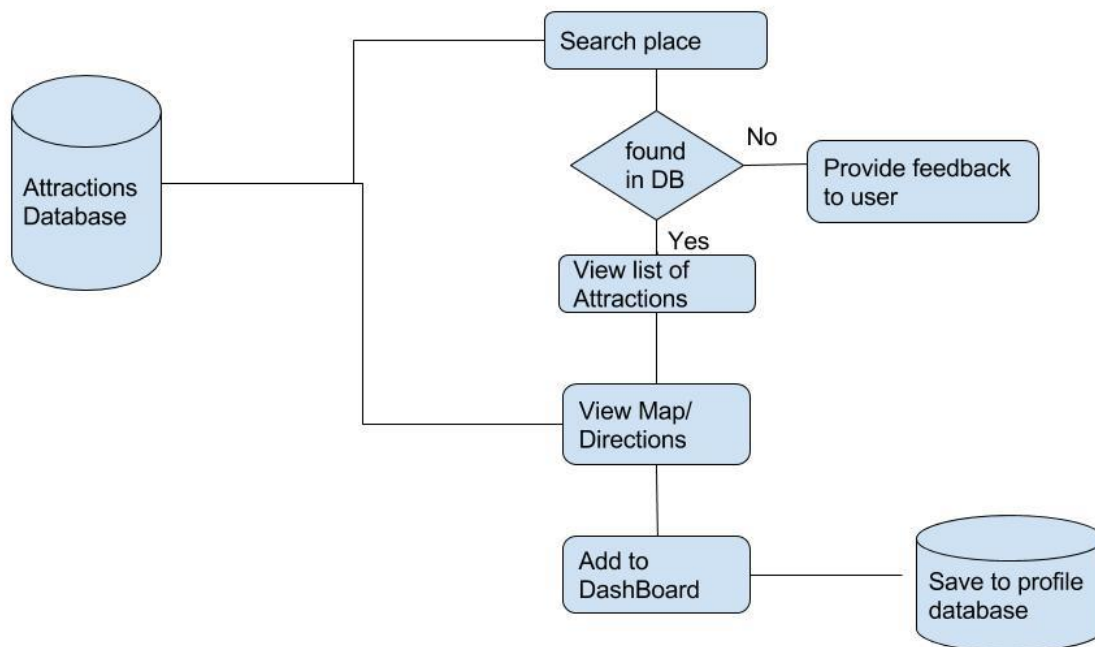
The storage and retrieval is done in a relational database. The design for the storage model is made as an entity relationship diagram.



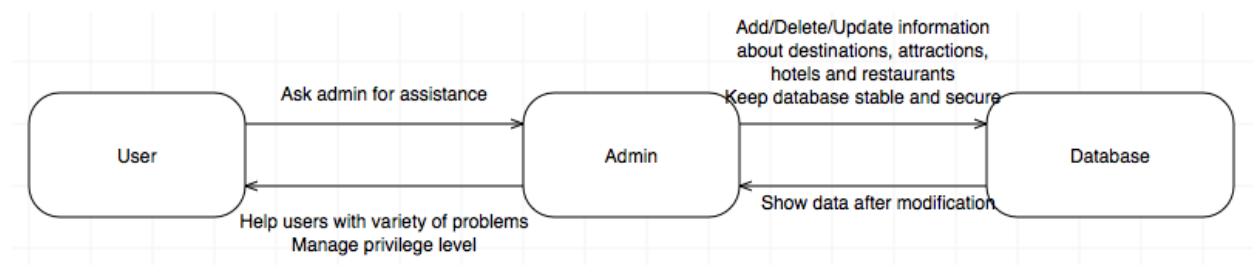
4.4 Data Model (Data Flow Diagrams)

The data flow diagram is used to depict the flow of data in different modules and components of the system. The data flow diagrams for some components of the system are as follows:

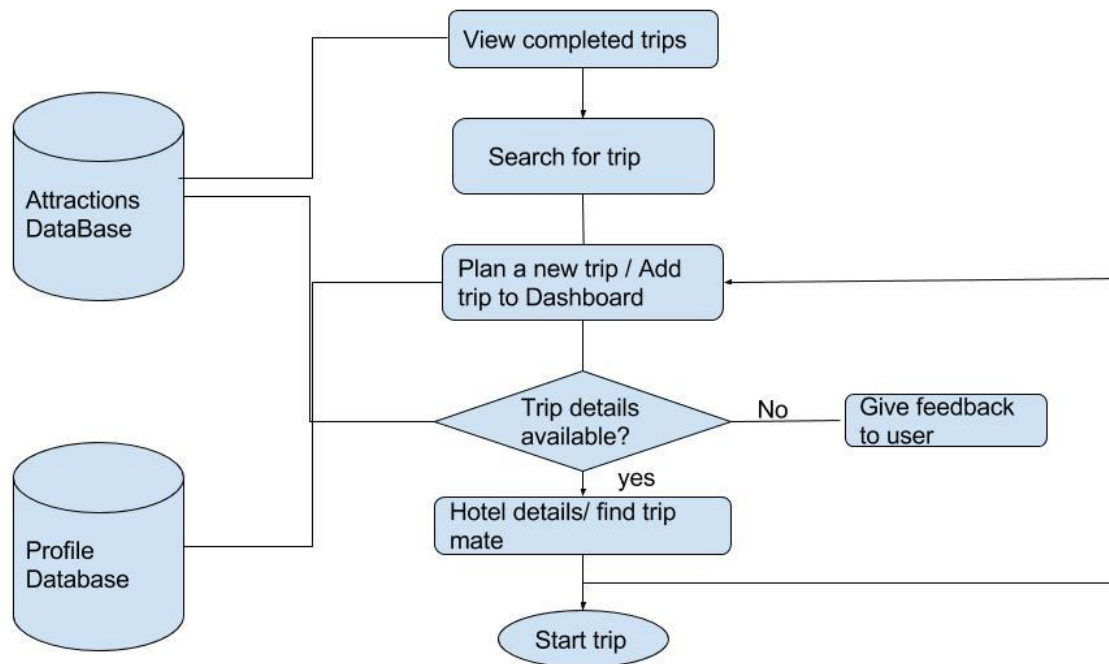
1. Attractions:



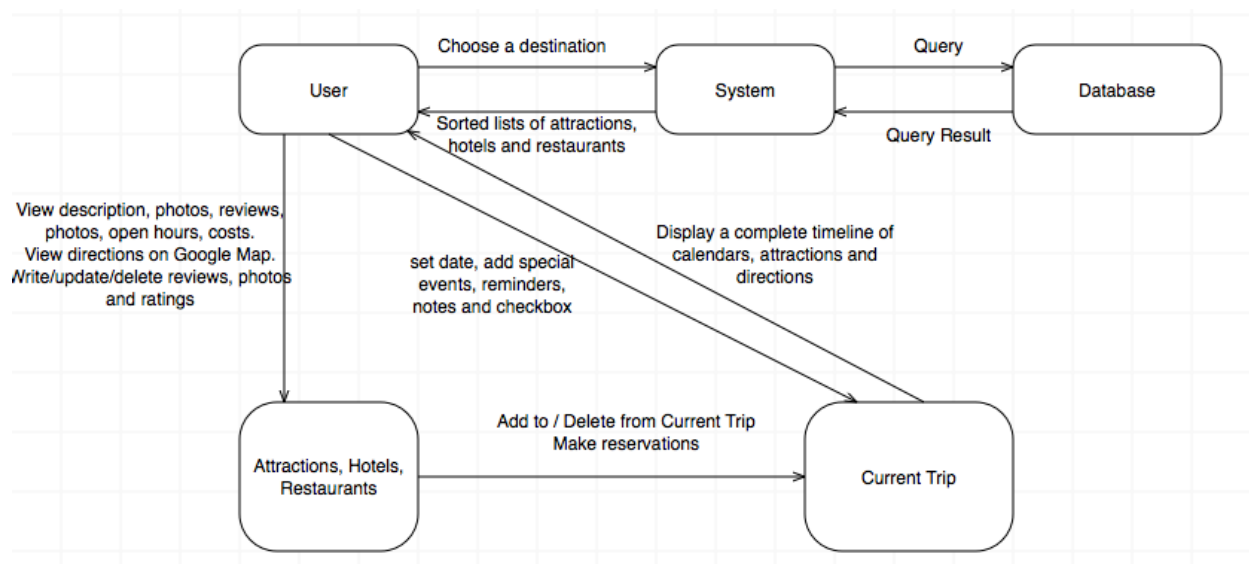
2. Admin

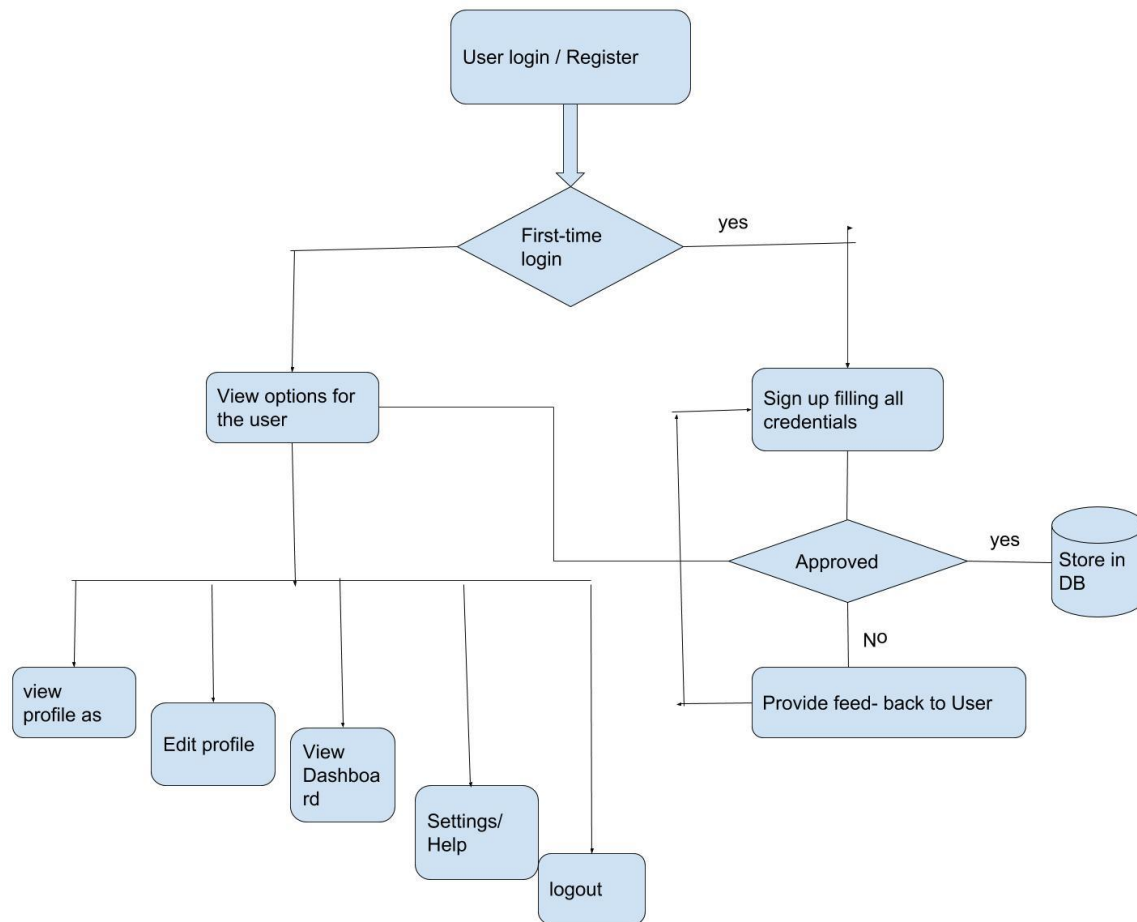


3. Trips



4. Multiple Component View



5. Profile:

5. Test Planning

Traceability

The use cases which we have made for the system will be used to test the functional requirements of the system. This way the test cases which will be run by the tester will easily be traced back to the use cases for which the testing is to be done. Thus, the traceability of the requirements to the use cases will be maintained.

Some advantage of using traceability matrix are:

1. It will ensure that all the requirements are met.
2. If the requirement changes, then it will be easier to update the test cases which are written for the use case or requirement.

Scenarios, Use Cases & Exception Handling

Use Cases depicts scenarios of user using the system. This can be seen as a real life situation in which the end-user/customer uses/interacts with the system to check for various failures.

Use cases are designed when requirements are clear. Use cases makes clear the overall flow of the system to developers first and then they can design the system accordingly.

Generally, test case consists of a unique identifier, use case name, actors involved, description stating what purpose of that particular test case is, precondition, post-condition, series of steps to follow, possible exception or risks that could occur during execution of that test.

Following are some test cases derived from the use case scenarios we developed in the requirements analysis phase for different modules of our Tourist Assist Application.

Usability Testing

Usability testing gives direct input on how end users use the system and can be seen as a technique used in user-centered interaction design to evaluate a system.

We have two types of system users. So the usability testing for both the users will be done as follows:

1. Usability from tourist's point of view:

Tourist is the primary user of the Tourist Assist application and this application can be found usable to tourist as they can get all information about their desired location easily also they can connect with other tourist travelling to the same location. Tourist can access all his trip related functionalities from this application. Each and every category of the user must find the system easy to use.

2. Usability from Admin's point of view:

Admin must be able to easily manage restaurants, destinations and attractions. After 1 hour of training given to the admin, admin must then not find any difficulty in using the system software.

Thus Tourist Assist Application passes usability test as it is really helpful for all its intended users.

Exception Handling

All the abnormal, unexpected or unprecedented events which can lead to any sort of exception in the system are handled in the test cases defined above. For example in Tourist Assist application, examples of exception are server failure, software failure, insufficient memory, database failure, web pages not working properly, entries not getting saved in database are some of the exception. We need to test such exception during code development. These exceptions can be avoided by proper testing and re-testing. Every time there is change in system requirement, we must perform testing to avoid these exceptions.

6. Test Cases & Test Results

6.1 Functional Requirements Test Cases & Test Results:

Test Case: Registration

Test Case #: 1 System: Tourist Assist App Designed by: Gan Xu Short Description: This test case verifies if Tourist registration function works properly.	Test Case Name: Tourist Registration
--	---

Pre-conditions:

The user hasn't registered yet and has entered the registration page to create an account.

Step	Action	Expected System Response	Actual Result
1	User fills in the field of Email, address, phone number and date of birth with wrong value. Example: Email: abc.com Address: a@b Phone number: 312 OIL 1234 Date of birth: 32, Jan, 2000	The system should not permit the user to register, instead, it will point out invalid field and prompt the user to enter the correct information.	As expected.

2	Confirm password does not match with password	The system should detect this error and indicate it to the user until they match.	As expected
3	The user fills every field with correct information and clicks “submit” button	The system should first check if the email is already registered, if it is, the system should reject this request and ask the user to provide another email. Else, the system will send a confirmation letter to user’s email.	As expected
4	User clicks the confirmation link in email	The system should store user’s information in database, informing the success to the user. Then return to login page and auto-fill user’s email.	As expected

Exceptions:

1. Invalid data causes exceptions
2. Data is not saved in database

Post-conditions:

The user successfully created an account.

Test Case: Login

Test Case #: 2 System: Tourist Assist App Designed by: Gan Xu Short Description: This test case verifies if Tourist login function works properly..	Test Case Name: Tourist Login
--	--------------------------------------

Pre-conditions:

The user has not logged in yet and the system is in the login page.

Step	Action	Expected System Response	Actual Result
1	The user clicks “forgot password” and enters his/her email in the popping dialog.	The system should send an email to user’s mailbox in order to let the user change his/her password	As expected
2	The user chooses to login with Facebook or Google+ and enters a wrong Facebook or Google+ account.	The system should verify account information with Facebook or Google and reject the login. The user shall get an error message.	As expected

3	The user chooses to login with Facebook or Google+ and enters a correct Facebook or Google+ account.	The system should verify account information with Facebook or Google and permit the login.	As expected
4	The user chooses to login with TouristAssist account and enters wrong information	The system should check the email&password combination in database and then reject the login. The user shall get an error message.	As expected
5	The user chooses to login with TouristAssist account and enters correct information	The system should check the email&password combination in database and then permit the login.	As expected
6	The user has logged in successfully.	The system should jump to the user's dashboard.	As expected
7	The user closes the app and re-opens it later.	The user should stay his/her login status without entering account and password again.	As expected
8	The user clicks "logout" button	The user should not be logged in any more.	As expected

Exceptions:

1. Login information not found in database
2. User doesn't stay logged in when he/she closes the app and reopen it.

Post-conditions:

Tourist should be able to login and logout.

Test Case: Dashboard

Test Case #: 3 System: Tourist Assist App Designed by: Gan Xu Short Description: This test case verifies if the user can view his/her dashboard correctly.	Test Case Name: View dashboard
---	---------------------------------------

Pre-conditions:

The user has logged in successfully.

Step	Action	Expected System Response	Actual Result
1	The user logs in and stays on dashboard.	The system should display the user's completed trips, reminders, notes, checklists and trip timeline in the dashboard.	As expected
2	The user modifies information on dashboard.	The dashboard should reflect these updates.	As expected.

	Example: a. Adds a reminder b. Deletes a note c. Changes trip timeline Then the user comes back to dashboard.		
3	The user clicks “logout”.	The system will pull back to login page.	As expected

Exceptions:

1. Dashboard information doesn't refresh in time.

Post-conditions:

The user can view updated information on dashboard after login.

Test Case: Profile

Test Case #: 4 System: Tourist Assist App Designed by: Gan Xu Short Description: This test case verifies if the user can view and update his/her own profile.	Test Case Name: User's Profile
--	---------------------------------------

Pre-conditions:

The user has logged in and stays on dashboard.

Step	Action	Expected System Response	Actual Result
1	The user clicks “Profile” button	The system should go to user's profile page, where the user can view his/her email, name, address, phone number, photo, date of birth, friends, previous trips and previous reviews. While email and name are not editable, friends, previous trips and reviews are auto-generated by other modules of this app, the other fields are editable.	As expected
2	The user updates his/her information with wrong value. Example: Address: a@b Phone number: 312 OIL 1234	The system should reject the update and indicate invalid fields.	As expected

	Date of birth: 32, Jan, 2000 Photo: Wrong format of image		
3	The user updates his/her information with correct information and clicks “Save Changes”	The system should ask the user to enter a PIN or password to confirm the change, and then update data in database.	As expected
4	The user deletes a friend in the Travelmate module, marks a trip as “finished” and modified a previous review. Then the user comes back to the profile page.	The system should display updated information.	As expected.
5	The user clicks “logout”	The system will not allow the user to view profile	As expected
6	The user sets his/her profile to be “invisible to others”	Other users will not see the user’s personal profile.	As expected

Exceptions:

1. Information may not be up-to-date
2. The editable property of these fields may be mixed

Post-conditions:

The user should be able to view and update his/her own profile.

Test Case: Search Travel Mates

Test Case #: 5 System: Tourist Assist App Designed by: Gan Xu Short Description: This test case verifies if the user can find travel mates correctly.	Test Case Name: Search Travel Mates
--	--

Pre-conditions:

The user has a planned trip and would like to find travel mates.

Step	Action	Expected System Response	Actual Result
1	In the current trip module, the user confirms his/her destinations and timetable, then clicks “Find Mates”.	The system should search in the database for other users who’s going to the same destination at the same time. Then the system will list all the searching results. If no users are found,	As expected.

	Test with different data for many times.	the system will recommend mates with close timetable and ask the user if he/she would to change destinations or timetable.	
2	The user has got a list of possible travel mates. Then the user click one of them.	The system should display the selected mate's profile and timetable if the mate doesn't set it to be invisible.	As expected
3	The user clicks "send message", fills in a message and click "Confirm".	The selected mate should receive the message in the mailbox, he/she should be able to reply or delete the message.	As expected

Exceptions:

1. The system doesn't find any proper travel mates.
2. The system displays user's profile even if it's set invisible.

Post-conditions:

The user should be able to find proper travel mates and talk with them.

Test Case: Communicates with other users

Test Case #: 6 System: Tourist Assist App Designed by: Gan Xu Short Description: This test case verifies if the user can add friends and create groups correctly.	Test Case Name: Friends & Groups
--	---

Pre-conditions:

The user is viewing another user's profile and they are not friends yet.

Step	Action	Expected System Response	Actual Result
1	The user clicks "Send friend request", enters a message in the popping message box and clicks "submit".	The other user should receive the request and be able to choose from accept, ignore and decline. The user should get correspondence feedback.	As expected
2	The user clicks "Create a group", enters a group name in the popping box and clicks "submit".	The system should create a group for the user, set the user as default group owner and allow the user to invite his/her friends to the group.	As expected

3	The user would like to talk with a friend or in a group	The system should provide chatting service just like whatsapp or any other social communication app.	As expected
4.	The user talks about travel plan with a friend or in a group	The user can create a plan chart in the chat page, everyone involved in the chat can modify it (just like Google drive) until they all agreed on the plan, then they should be allowed to add the plan to everyone's dashboard.	As expected

Exceptions:

1. Mutual plan is not correctly generated (synchronize problem, etc)
2. Chatting-related problems (lost wifi connection, chatting history, etc)

Post-conditions:

The user should be able to add friends, create groups and talk about travel plan.

Test Case: Help

Test Case #: 7 System: Tourist Assist App Designed by: Gan Xu Short Description: This test cases verifies the user can get necessary help in the Help module.	Test Case Name: Help
--	-----------------------------

Pre-conditions:

The user has logged in and stays on dashboard.

Step	Action	Expected System Response	Actual Result
1	User clicks Help module	The system should go to the Help page, including a list of FAQ, a search bar, a slider bar to control the beginner's mode, admin's linkified phone & email and an edit text view for sending in-app message to admin.	As expected
2	User searches some keywords in the search bar.	The system should list FAQs concerning keywords.	As expected
3	User clicks admin's phone	The system should dial the number automatically for the user.	As expected
4	User clicks admin's email	The system should open the phone's default mail app and autofill a letter's receiver with admin's email address.	As expected

5	User enters a message in the edit text view and clicks “send”	The admin should receive a notification about the in-app message and reply it.	As expected.
6	The user activates the “beginner’s mode”.	The system should provide guide when the user enters each module at the first time.	As expected
7	The user deactivates the “beginner’s mode”	The system will no longer show guide messages	As expected

Exceptions:

1. FAQ not correctly searched.
2. App doesn’t have permission to make phone calls and send emails.

Post-conditions:

The user should be able to get help when needed.

Test Case: Admin

Test Case #: 8 System: Tourist Assist App Designed by: Gan Xu Short Description: This test case verifies if the admin can modify database and provide help to users.	Test Case Name: Admin
---	------------------------------

Pre-conditions:

The admin has logged in with admin’s account

Step	Action	Expected System Response	Actual Result
1	The admin adds/deletes/updates data on destinations, attractions, restaurants, hotels and FAQ	The database should reflect admin’s modification.	As expected
2	The admin enters the mail box.	The admin should see messages from users and reply them to provide help.	As expected

Exceptions:

1. Database update error

Post-conditions:

The admin should be able to serve normally.

Test Case: Adding new trips.

Test Case #: 9 System: Tourist Assist App Designed by: Nishant Sharma Short Description: This test case verifies if a user can add a new trip.	Test Case Name: Adding new trip.
---	---

Pre-conditions:

The user has logged in with his account.

Step	Action	Expected System Response	Actual Result
1	User clicks on the new trip information tab.	System opens a new page where user can add a new trip.	As expected
2	The tourist enters the desired destination, date, hotel details etc.	User can see his review in the text box.	As expected
3	User clicks on save after completing all the details.	New trip gets saved to the system database.	As expected

Exceptions:

1. Database update error.
2. The desire location was not found in the database.

Post-conditions:

The user is able to add a new trip normally.

Test Case: Adding reviews.

Test Case #: 10 System: Tourist Assist App Designed by: Nishant Sharma Short Description: This test case verifies if a user can add reviews for destination, attraction, restaurant or travel mate.	Test Case Name: Review
--	-------------------------------

Pre-conditions:

The user has logged in with his account.

Step	Action	Expected System Response	Actual Result
1	User opens a destination, restaurants or an attraction page.	System opens the desired page.	As expected.
2	User clicks on the add a review tab.	System opens a text box where user can add his reviews.	As expected
3	User enters his review.	User can see his review in the text box.	As expected
4	User clicks on save after writing the review.	The review gets saved to the system database.	As expected

Exceptions:

1. Database update error
2. The user enters a spam content

Post-conditions:

The user is able to add his reviews normally.

Test Case: Modifying/Deleting reviews.

Test Case #: 11 System: Tourist Assist App Designed by: Nishant Sharma Short Description: This test case verifies if a user can modify/delete his previous reviews.	Test Case Name: Review
--	-------------------------------

Pre-conditions:

The user has logged in with his account.

Step	Action	Expected System Response	Actual Result
1	User selects the review from any of the places reviewed by him.	System displays the option to edit or delete the review.	As expected
2	User edits or deletes his previous review.	System edits or deletes user review.	As expected

Exceptions:

1. Database update error
2. The user enters a spam content

Post-conditions:

The user is able to modify/delete reviews normally.

Test Case: Adding photos.

Test Case #: 12 System: Tourist Assist App Designed by: Nishant Sharma Short Description: This test case verifies if a user can add photos for destination, attraction, and restaurant.	Test Case Name: Adding photos.
--	---------------------------------------

Pre-conditions:

The user has logged in with his account.

Step	Action	Expected System Response	Actual Result
1	User opens a destination, restaurants or an attraction page.	System opens the desired page.	As expected
2	User clicks on add a photo tab.	System opens a new box for browsing the photo in users phone.	As expected
3	User browse and select the file from the computer and clicks on upload tab.	The photo gets saved to the system database.	As expected

Exceptions:

1. Database update error
2. The user enters a spam content

Post-conditions:

The user is able to add photos normally.

Test Case: Edit/Delete photos.

Test Case #: 13 System: Tourist Assist App Designed by: Nishant Sharma Short Description: This test case verifies if a user can edit/delete his previously uploaded photos.	Test Case Name: Edit/delete photos.
--	--

Pre-conditions:

The user has logged in with his account.

Step	Action	Expected System Response	Actual Result
1	User opens a page where he has previously added a photo.	System opens the desired page.	As expected
2	User clicks on edit/delete photo tab.	System edits/deletes the desired photo.	As expected

Exceptions:

1. Database update error
2. The user enters a spam content

Post-conditions:

The user is able to add photos normally.

Test Case: Comment on reviews.

Test Case #: 14 System: Tourist Assist App Designed by: Nishant Sharma Short Description: This test case verifies if a user can comment on a review by other user.	Test Case Name: Comment
---	--------------------------------

Pre-conditions:

The user has logged in with his account.

Step	Action	Expected System Response	Actual Result
1	User selects any review or photos from the list of reviews or photos of a destination, attraction, restaurant or a different user.	System shows the option of replying to a review.	As expected
2	User types the comment to a reply.	Comment gets saved in the database.	As expected

Exceptions:

1. Database update error
2. The user enters a spam content

Post-conditions:

The user is able to add a comment to review normally.

Test Case: Search destinations.

Test Case #: 15 System: Tourist Assist App Designed by: Nishant Sharma Short Description: This test case verifies if a user can search a desired destination.	Test Case Name: Search destination
--	---

Pre-conditions:

The user has logged in with his account.

Step	Action	Expected System Response	Actual Result
1	User clicks the search destination tab.	System opens a text box for user to type the desired destination.	As expected
2	User enters the desired destination and click on search tab.	System displays information about the destination.	As expected

Exceptions:

1. User location not found
2. User location does not exist in the database

Post-conditions:

The user is able to search a destination normally.

Test Case: View destinations.

Test Case #: 16 System: Tourist Assist App Designed by: Nishant Sharma Short Description: This test case verifies if a user can view a desired destination.	Test Case Name: Search destination
--	---

Pre-conditions:

The user has logged in with his account.

Step	Action	Expected System Response	Actual Result
1	User clicks the search destination tab.	System opens a textbox for user to type the desired destination.	As expected
2	User selects the desired destination.	System displays information about the selected destination.	As expected

Exceptions:

1. User location not found
2. User location does not exist in the database

Post-conditions:

The user is able to search a destination normally.

Test Case: View attraction.

Test Case #: 17 System: Tourist Assist App Designed by: Nishant Sharma Short Description: This test case verifies if a tourist is able to view an attraction.	Test Case Name: View attraction
--	--

Pre-conditions:

The user has logged in with his account.

Step	Action	Expected System Response	Actual Result
1	User search for his desired location	System search for user's desired location.	As expected
2	User select's particular attraction.	System opens a page with all details of that attraction.	As expected

Exceptions:

1. The direction does not exist
2. The direction is not found in the database

Post-conditions:

The user is able to see details of a desired attraction.

Test Case: Direction to attraction.

Test Case #: 18 System: Tourist Assist App Designed by: Nishant Sharma Short Description: This test case verifies if a tourist is able to view the direction to his desired attraction location	Test Case Name: Direction
--	----------------------------------

Pre-conditions:

The user has logged in with his account.

Step	Action	Expected System Response	Actual Result
1	User search for his desired location	System search for user's desired location.	As expected
2	User select's particular attraction.	System opens a page with all details of that attraction.	As expected
3	User clicks on direction to attraction.	System displays the direction to the tourist desired attraction.	As expected

Exceptions:

1. The direction does not exist
2. The direction is not found in the database

Post-conditions:

The user is able to see directions to the desired attraction.

Test Case: Add attraction to trip timeline:

Test Case #: 19 System: Tourist Assist App Designed by: Nishant Sharma Short Description: This test case verifies if a user can add attraction to trip timeline.	Test Case Name: Add attraction to trip timeline
---	--

Pre-conditions:

The user has logged in with his account.

Step	Action	Expected System Response	Actual Result
1	User selects the desired attraction.	System displays information and options about the attraction.	As expected
2	User clicks on the Add attraction to trip timeline tab.	System adds the selected attraction to the trip timeline.	As expected

Exceptions:

1. User attraction not found
2. User attraction does not exist in the database

Post-conditions:

The user is able to search add attraction to trip timeline normally.

Test Case: View map of attraction.

Test Case #: 20 System: Tourist Assist App Designed by: Nishant Sharma Short Description: This test case verifies if a tourist is able to view a map of all the attractions near his attraction.	Test Case Name: Map of attraction
---	--

Pre-conditions:

The user has logged in with his account.

Step	Action	Expected System Response	Actual Result
1	User search for his desired location	System search for user's desired location.	As expected
2	User select's particular attraction.	System opens a page with all details of that attraction.	As expected
3	User clicks on view map.	System displays the map of all the attractions near user's attraction.	As expected

Exceptions:

1. The direction does not exist
2. The direction is not found in the database

Post-conditions:

The user is able to see directions to the desired attraction.

Test Case: View hotel list.

Test Case #: 21 System: Tourist Assist App Designed by: Nishant Sharma Short Description: This test case verifies if a tourist is able to view list of hotels near his attraction.	Test Case Name: Hotel list
---	-----------------------------------

Pre-conditions: The user has logged in with his account.
--

Step	Action	Expected System Response	Actual Result
1	User search for his desired location	System search for user's desired location.	As expected
2	User select's particular attraction.	System opens a page with all details of that attraction.	As expected
3	User clicks on view hotel list tab.	System displays the list of all the hotels near user's attraction.	As expected

Exceptions: 1. No hotels found near user's location.
--

Post-conditions: The user is able to see list of hotels near his attraction.
--

Test Case: Add hotel to trip.

Test Case #: 22 System: Tourist Assist App Designed by: Nishant Sharma Short Description: This test case verifies if a tourist is able to add a hotel to his trip.	Test Case Name: Hotel list
---	-----------------------------------

Pre-conditions: The user has logged in with his account.
--

Step	Action	Expected System Response	Actual Result
1	User searches for the trip to which he wants to add.	System displays the trip.	As expected

2	User select's add a hotel to a trip tab.	System shows the list of the hotels available.	As expected
3	User selects the hotels from the list.	System adds selected hotel to the trip.	As expected

Exceptions:

1. Database update error

Post-conditions:

The user is able to add hotel to his trip.

Test Case: View restaurants.

Test Case #: 23 System: Tourist Assist App Designed by: Nishant Sharma Short Description: This test case verifies if a tourist is able to view a list of restaurants.	Test Case Name: View restaurants
--	---

Pre-conditions:

The user has logged in with his account.

Step	Action	Expected System Response	Actual Result
1	User search for his desired location	System search for user's desired location.	As expected
2	User select's particular attraction.	System opens a page with all details of that attraction.	As expected
3	User clicks on view restaurants list tab.	System displays the list of all the restaurants near user's attraction.	As expected

Exceptions:

1. No restaurants near user's attraction.

Post-conditions:

The user is able to view list of restaurants near his attraction.

Test Case: View Completed Trips

Test Case #: 24 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: The test is to verify that the user can view his previous trips	Test Case Name: Completed Trips:
--	---

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Trip page

Step	Action	Expected System Response	Actual Result
1	The User clicks on the trips module and click on the completed trip	The system is expected to display all store previous trip by the user on the page	As expected
2	The user selects one from the option and clicks to view more	The system is expected to display all details of the trip including the time, dates and location	As expected
3	The user clicks the close button to return back to the previous page	The system is expected to take the user back to the previous viewed page	As expected

Exceptions: The user travel history is not stored on the data base The Admin restricts the user from viewing his completed trip history

Post-conditions: The user views his completed trip history page
--

Test Case: View current trip

Test Case #: 25 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify the user is able to view his View current trip	Test Case Name: View current trips
--	---

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Trip page

Step	Action	Expected System Response	Actual Result
1	The User clicks on the trips module and click on view current trip	The system is expected to take him to the page where he can find timeline of the trip, completed destinations, remaining destinations, visited attractions, remaining attractions, photos from current trips and restaurants to visit.	As expected
2	The user selects one from the option and clicks to view more	The system is expected to display all details of the current trip including all details	As expected
3	The user clicks the close button to return back to the previous page	The system is expected to take the user back to the previous viewed page	As expected

Exceptions: The user current history details is not stored on the database
The Admin restricts the user from viewing his current history details
The user is not logged in

Post-conditions: The user views is current trip details

Test Case: Start Trip

Test Case #: 26 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify the user is able to select and update his trip from the planned trips	Test Case Name: start trip
---	-----------------------------------

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Trip page

Step	Action	Expected System Response	Actual Result
1	The User clicks on the trips module and click on start trip from the planned trips	The system updates the trip details from planned trip to current trip	As Expected
2	The user views his current trip section	The system is expected to update the user current trip section to the selected trip	As Expected

3	The user views his dashboard to also check his current trip	The system is expected to update the user dashboard with the selected star trip	As Expected
---	---	---	-------------

Exceptions: Database error
 The user selects a location blocked by the Admin
 The user did not click the start trip button

Post-conditions: The user current trip will be updated with the chosen trip from the start trip button

Test Case: Plan Trip:

Test Case #: 27 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify that the Plan trip module work correctly	Test Case Name: Plan trip
--	----------------------------------

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Current Trip Page

Step	Action	Expected System Response	Actual Result
1	The user clicks the tab to plan new tip by editing the date, time, destination and more	The user should be able to plan his new trip by editing all necessary information	As Expected
2	The user clicks the tab to update the set date and destination for a planned trip	The user should be able to update the set information and the system is expected to store the new information	As Expected
3	The users clicks the tab to add destination, attractions and plan restaurants	The user should be able to add new details to his planned trip and the system is expected to store all information	As Expected
4	The user clicks the tab to add events to the calendar	The user should be able to add new events to his calendar and the system is expected to store all information	As Expected

Exceptions: Database Error
 The admin restricts the user's account the features it can perform

The user is not logged in

Post-conditions: The user is able to add new plan trip, edit and update all information about his trip

Test Case: View Timeline

Test Case #: 28
System: Tourist Assist App
Designed by: Ericson Akatakpo
Short Description: This test is to verify that the view timeline module work correctly

Test Case Name: View Timeline

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Current Trip Page

Step	Action	Expected System Response	Actual Result
1	The user clicks on the Timeline tab to view a highlight details of his trip	The system is expected to display all trip information in a Chart fashion	As Expected

Exceptions: Database Error
 The user haven't added any destination details
 The user is not logged in
 The user account is restricted by the Amin

Post-conditions: The User is expected to view the details of his trip information

Test Case: Add Hotel/Stay in selected destination:

Test Case #: 29
System: Tourist Assist App
Designed by: Ericson Akatakpo
Short Description: This test is to verify that the user can add/stay in hotels in selected location

Test Case Name:
 Add Hotel/Stay in selected destination:

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Current Trip Page

Step	Action	Expected System Response	Actual Result
1	The user click the link to add/stay in a hotel in a selected location	The system is expected to add the selected hotel detail to his travel details	As Expected

Exceptions: Database Error
 The User is not logged in
 The User account is restricted from performing this action

Post-conditions: The User is able to add a hotel he wants to stay to his selected destination

Test Case: Complete Trip

Test Case #: 30 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify that the user can complete the trip	Test Case Name: complete trip.
---	---------------------------------------

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Current Trip Page

Step	Action	Expected System Response	Actual Result
1	The user clicks the link to complete a current trip	The system is expected to confirm from the user if he really want to complete the trip before removing the trip from the current trip to completed trip	As Expected

Exceptions: Database Error
 The user is not logged in
 The Admin restrict the functionality from the users account

Post-conditions: The user completed the trip and his dashboard is updated with all details

Test Case: Manage Notes

Test Case #: 31 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify that the user can manage Notes	Test Case Name: Manage Notes
--	-------------------------------------

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Utilities Page

Step	Action	Expected System Response	Actual Result
1	The user click the link to add a new note to the planned trip or current trip	The system is expected to create a note, ask the user to name it and store the new note to the database	As Expected
2	The user clicks the link to edit/delete his created note	The system is expected to edit the required note to the new details and store it to the database	As Expected

Exceptions: Database Error
 The user is not logged in
 The Admin restrict the functionality from the users account

Post-conditions: The user is able to perform his desired activity either to create, edit or delete his notes

Test Case: Manage Reminder

Test Case #: 32 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify that the user can manage Reminder	Test Case Name: Manage Reminder
---	--

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Utilities Page

Step	Action	Expected System Response	Actual Result
1	The user click the link to add a new reminder	The system is expected to add a new reminder and store it on the database	As Expected
2	The user click the link to add the date, time and reminder text	The system is expected to save to the details to the database and remind the user on the desired time	As Expected

Exceptions: Database Error
 The user is not logged in
 The Admin restrict the functionality from the user's account

Post-conditions: The user sets a reminder and the reminder notifies the user at the desired time

Test Case: Manage checklist

Test Case #: 33 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify that the user can manage Reminder	Test Case Name: Manage Checklist
---	---

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Utilities Page

Step	Action	Expected System Response	Actual Result
1	The user click the link to add a new checklist to the planned or current trip	The system is expected a new checklist and store the details when the user is done adding all details	As Expected
2	The user clicks the link to edit the checklist	The system is expected to allow the user to edit the created check list and store the new details in the database	As expected

Exceptions: Database Error
 The user is not logged in
 The Admin restrict the functionality from the user's account

Post-conditions: The user is able to add, edit and delete his checklist

Test Case: View Calendar

Test Case #: 34
System: Tourist Assist App
Designed by: Ericson Akatakpo
Short Description: This test is to verify that the user can View the Calendar

Test Case Name: View Calendar

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Utilities Page

Step	Action	Expected System Response	Actual Result
1	The user clicks the link to view the calendar events	The system is expected to show all details of the planned trip to the user	As Expected
2	The user clicks on any trip for more information	The system is expected to show more information to the user	As Expected

Exceptions: Database Error
 The user is not logged in
 The Admin restrict the functionality from the user's account

Post-conditions: The user is able to view his calendar and details about his trips

Test Case: View Completed Trips

Test Case #: 35 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: The test is to verify that the user can view his previous trips	Test Case Name: Completed Trips:
--	---

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Trip page

Step	Action	Expected System Response	Actual Result
1	The User clicks on the trips module and click on the completed trip	The system is expected to display all store previous trip by the user on the page	As expected
2	The user selects one from the option and clicks to view more	The system is expected to display all details of the trip including the time, dates and location	As expected
3	The user clicks the close button to return back to the previous page	The system is expected to take the user back to the previous viewed page	As expected

Exceptions: The user travel history is not stored on the data base
 The Admin restricts the user from viewing his completed trip history

Post-conditions: The user views his completed trip history page

Test Case: View current trip

Test Case #: 36 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify the user is able to view his View current trip	Test Case Name: View current trips
--	---

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Trip page

Step	Action	Expected System Response	Actual Result
1	The User clicks on the trips module and click on view current trip	The system is expected to take him to the page where he can find timeline of the trip, completed destinations, remaining destinations, visited attractions, remaining attractions, photos from current trips and restaurants to visit.	As expected
2	The user selects one from the option and clicks to view more	The system is expected to display all details of the current trip including all details	As expected
3	The user clicks the close button to return back to the previous page	The system is expected to take the user back to the previous viewed page	As expected

Exceptions: The user current history details is not stored on the database
The Admin restricts the user from viewing his current history details
The user is not logged in

Post-conditions: The user views is current trip details

Test Case: Start Trip

Test Case #: 37 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify the user is able to select and update his trip from the planned trips	Test Case Name: start trip
---	-----------------------------------

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Trip page

Step	Action	Expected System Response	Actual Result
1	The User clicks on the trips module and click on start trip from the planned trips	The system updates the trip details from planned trip to current trip	As Expected
2	The user views his current trip section	The system is expected to update the user current trip section to the selected trip	As Expected

3	The user views his dashboard to also check his current trip	The system is expected to update the user dashboard with the selected star trip	As Expected
---	---	---	-------------

Exceptions: Database error
 The user selects a location blocked by the Admin
 The user did not click the start trip button

Post-conditions: The user current trip will be updated with the chosen trip from the start trip button

Test Case: Plan Trip:

Test Case #: 38 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify that the Plan trip module work correctly	Test Case Name: Plan trip
--	----------------------------------

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Current Trip Page

Step	Action	Expected System Response	Actual Result
1	The user clicks the tab to plan new tip by editing the date, time, destination and more	The user should be able to plan his new trip by editing all necessary information	As Expected
2	The user clicks the tab to update the set date and destination for a planned trip	The user should be able to update the set information and the system is expected to store the new information	As Expected
3	The users clicks the tab to add destination, attractions and plan restaurants	The user should be able to add new details to his planned trip and the system is expected to store all information	As Expected
4	The user clicks the tab to add events to the calendar	The user should be able to add new events to his calendar and the system is expected to store all information	As Expected
Exceptions: Database Error The admin restricts the user's account the the features it can perform The user is not logged in			

Post-conditions: The user is able to add new plan trip, edit and update all information about his trip

Test Case: View Timeline

Test Case #: 39 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify that the view timeline module work correctly	Test Case Name: View Timeline
--	--------------------------------------

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Current Trip Page

Step	Action	Expected System Response	Actual Result
1	The user clicks on the Timeline tab to view a highlight details of his trip	The system is expected to display all trip information in a Chart fashion	As Expected

Exceptions: Database Error
 The user haven't added any destination details
 The user is not logged in
 The user account is restricted by the Amin

Post-conditions: The User is expected to view the details of his trip information

Test Case: Add Hotel/Stay in selected destination:

Test Case #: 40 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify that the user can add/stay in hotels in selected location	Test Case Name: Add Hotel/Stay in selected destination:
---	---

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Current Trip Page

Step	Action	Expected System Response	Actual Result
1	The user click the link to add/stay in a hotel in a selected location	The system is expected to add the selected hotel detail to his travel details	As Expected

Exceptions: Database Error
The User is not logged in
The User account is restricted from performing this action

Post-conditions: The User is able to add a hotel he wants to stay to his selected destination

Test Case: Complete Trip

Test Case #: 41 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify that the user can complete the trip	Test Case Name: complete trip.
---	---------------------------------------

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Current Trip Page

Step	Action	Expected System Response	Actual Result
1	The user clicks the link to complete a current trip	The system is expected to confirm from the user if he really want to complete the trip before removing the trip from the current trip to completed trip	As Expected

Exceptions: Database Error
The user is not logged in
The Admin restrict the functionality from the user's account

Post-conditions: The user completed the trip and his dashboard is updated with **all details**

Test Case: Manage Notes

Test Case #: 42 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify that the user can manage Notes	Test Case Name: Manage Notes
--	-------------------------------------

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Utilities Page

Step	Action	Expected System Response	Actual Result
1	The user click the link to add a new note to the planned trip or current trip	The system is expected to create a note, ask the user to name it and store the new note to the database	As Expected
2	The user clicks the link to edit/delete his created note	The system is expected to edit the required note to the new details and store it to the database	As Expected

Exceptions: Database Error
 The user is not logged in
 The Admin restrict the functionality from the user's account

Post-conditions: The user is able to perform his desired activity either to create, edit or delete his notes

Test Case: Manage Reminder

Test Case #: 43 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify that the user can manage Reminder	Test Case Name: Manage Reminder
---	--

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Utilities Page

Step	Action	Expected System Response	Actual Result
1	The user click the link to add a new reminder	The system is expected to add a new reminder and store it on the database	As Expected
2	The user click the link to add the date, time and reminder text	The system is expected to save to the details to the database and remind the user on the desired time	As Expected

Exceptions: Database Error
 The user is not logged in
 The Admin restrict the functionality from the user's account

Post-conditions: The user sets a reminder and the reminder notifies the user at the desired time

Test Case: Manage checklist

Test Case #: 44 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify that the user can manage Reminder	Test Case Name:
---	------------------------

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Utilities Page

Step	Action	Expected System Response	Actual Result
1	The user click the link to add a new checklist to the planned or current trip	The system is expected a new checklist and store the details when the user is done adding all details	As Expected
2	The user clicks the link to edit the checklist	The system is expected to allow the user to edit the created check list and store the new details in the database	As expected

Exceptions: Database Error
 The user is not logged in
 The Admin restrict the functionality from the user's account

Post-conditions: The user is able to add, edit and delete his checklist

Test Case: View Calendar

Test Case #: 45 System: Tourist Assist App Designed by: Ericson Akatakpo Short Description: This test is to verify that the user can View the Calendar	Test Case Name: View Calendar
---	--------------------------------------

Pre-conditions: The user must be logged in with a valid Username and password and must be on the Utilities Page

Step	Action	Expected System Response	Actual Result
1	The user clicks the link to view the calendar events	The system is expected to show all details of the planned trip to the user	As Expected
2	The user clicks on any trip for more information	The system is expected to show more information to the user	As Expected

Exceptions: Database Error
 The user is not logged in
 The Admin restrict the functionality from the user's account

Post-conditions: The user is able to view his calendar and details about his trips

6.2 Non-Functional Requirements Test Cases & Test Results:

Test Case: Performance (Response time & Throughput)

Test Case #: P-1 System: Tourist Assist App Designed by: Mohammed Short Description: Test the response time of the screen when the user logs in and app displays the dashboard.	Test Case Name: Login Response Time
--	--

Pre-conditions: The user is a registered user and has valid login credentials like username and password.

Step	Action	Expected System Response	Expected Response Time
1	Enter username	System displays username	The transition from the Login page to the Dashboard page takes less than 1 second to display the dashboard completely.
2	Enter password	System displays hidden password	
3	Click on Login button	System take the user to dashboard	

Exceptions:

- Connection Error
- Another App Interruption
- Login Credentials entered wrong

Post-conditions: After clicking the login page, the user sees the dashboard with all the functionalities in less than a second.

Results: After testing for 100 different types of users from 4 different regions, the dashboard was displayed in less than one second.

Test Case: Recoverability

Test Case #: R-1 System: Tourist Assist App Designed by: Mohammed Short Description: Test if the app can come back to its previous state in case of a system shut down	Test Case Name: System Shut Down Recovery
---	--

Pre-conditions: The user is logged in to the app and currently adding a new destination to a trip

Step	Action	Expected System Response
1	Select Destination	The destination is selected and destination profile page is shown
2	Shut Down Phone	Phone shuts down and app is closed unexpectedly
3	Restart Phone	Phone restarts
4	Open Tourist Assist App	App displays the destination profile page and saved all the previous entered information before the phone was shut down.

Exceptions:

- No Battery and so cannot restart the phone immediately
- Internet Connection Error before shutting down the phone

Post-conditions: After a rapid shut down of the system, the app saves and displays the previous state when app is reopened after restarting the system

Results: After testing the test case for 10 times, every time the state was saved and information was not lost when the app was restarted

Test Case: Security

Test Case #: S-1

System: Tourist Assist App

Designed by: Mohammed

Short Description: Test if any fake username and password can open the app on the user's phone.

Test Case Name: Login Security Test

Pre-conditions: The Username and Password for the user is "admin" & "admin123456#" respectively and the app is installed on the phone.

Step	Action	Expected System Response
1	Enter 'admin' in username	App displays 'admin' in username field
2	Enter 'admin' in password	App displays '*****' in password field
3	Click on Login button	App displays 'Password Incorrect'

Post-conditions: The user is not allowed to login if the password is not correct

Results: 200 different combinations of the username "admin" with different fake passwords were tried to login to the system. The system did not login to any of those combination

6.3 Usability (Ease-of-use) Test Cases & Test Results:

The Tourist Assist App has a wide variety of users which are distinguished based on age, educational background, phone usability and region from where they are coming. So, in order to test the app with all the different types of users, we selected 50 users for testing which were all from different user class. The user categories is mentioned in the second chapter of the report.

We took 5 users from every category and then performed the usability (ease-of-use) testing on them.

Test Case: Add New Trip Usability

Test Case #: UT-1 System: Tourist Assist Designed by: Mohammed Short Description: To check if the user adds a new trip from dashboard to planned trips in 1.5 to 2 minutes	Test Case Name: Add New Trip Usability Test
---	--

Pre-conditions: The user is already logged in to the system and currently on the dashboard screen of the application.

Step	Action	Expected Time Taken by User (seconds)
1	Click on the '+' button on the right corner of dashboard	2
2	Enter New Trip Name	10
3	Select Start Date	10
4	Select End Date	10
5	Select Planned Trip checkbox	5
6	Click on 'Add Destination'	2
7	Search for the destination 'Chicago' in search bar	10
8	Click on 'Chicago' in displayed results	2
9	View Chicago Destination Profile	10-60
10	Click '+' button on top right corner (User will be taken to previous 'Add New Trip' screen)	5
11	Click Save button	2

Exceptions:

- App crashes in between
- User clicks anything else in between the regular flow of the functions
- User gets diverted by any other app on the phone

Post-conditions: 80% of the users tested must complete the function between 1.5 to 2 minutes

Results: After testing 50 different types of users, 42 users completed the function between 1.5 to 2 minutes which results in a successful usability test result.

Test Case: Register Usability

Test Case #: UT-1 System: Tourist Assist Designed by: Mohammed Short Description: Test the time taken by a user to register with the app for the first time	Test Case Name: Register Usability test
--	--

Pre-conditions: The user has installed the app for the first time and is on the first screen of the app where Login and Register options are given.

Step	Action	Expected Time Taken by User (seconds)
1	Click Register	2
2	Enter First Name	5
3	Enter Last Name	5
4	Enter Email Address	5
5	Select Age	5
6	Enter Phone Number	5
7	Select Country	5

Exceptions: User clicks Login instead of register (In this case, a warning will be displayed of wrong username and password). This may take extra 5 or 10 seconds.

Post-conditions: 85% of the users successfully registers for the app in 30-35 seconds

Results: After testing 50 different types of users, 45 users completed the task in an average of 32 seconds

Test Case: Add Destination Photo Usability

Test Case #: UT-1 System: Tourist Assist Designed by: Mohammed Short Description: Test if the user can add a photo from gallery for the destination in 10-20 seconds	Test Case Name: Add Destination Photo Usability Test
---	---

Pre-conditions: The user is already logged in to the application and is on the currently selected Chicago Trip of the user. The user is on the current trips page

Step	Action	Expected Time Taken by User (seconds)
1	Press the 'Add Photos' button on the current trip screen	2
2	Press the 'Gallery' button (The photo gallery will open up)	5-10
3	Select a photo from the gallery	2
4	Press 'Upload' button (Photo is uploaded and user is taken back to current trip photos screen)	2

Exceptions:

- User has no photos in the photo gallery.
- User sits and sees all the photos rather than selecting a photo to upload

Post-conditions: A User successfully uploads a photo in 10-20 seconds

Results: After testing 50 users, 39 users successfully completed the task in an average of 18 seconds.

7. Quality & Value:

The Tourist Assist App is designed by keeping in mind the diverse user base which will be using the app. We have designed our app such that every category of the user mentioned in the 'User Categories' section can benefit from the app. These are some key points which highlight the star features of the app:

1. Help Mechanisms:

- We have made different type of Help Mechanisms for different types of users. The User enters the Age at the time of registration. By getting the age of the user, we display different types of help to different types of user.
- A user whose age is 24 finds using an app on the phone very easy to use and is also familiar with the common User Interface Designs. So, for this category of users, help is kept minimal. For some exceptional cases where the user needs help, the user can see the Help section specifically designed to help users with all the functions of the system.
- On the other hand, a user whose age is 55 and wants to use the app but it is not familiar with the phone apps and also not familiar with the common User Interface Designs. For such type of users, we have designed a full help starting from when the user logs in to the system. The user is navigated to the whole app by a step of tutorials which then makes the functions easier to understand by the user.

2. User Interface Design:

- Our main goal is to make users happy by providing an easy to use interface for the application. We have done a thorough research on the different user interface designs and have extracted the most elegant and easy to use designs for our app. Most of the functions have a user interface similar to majority of the phone apps. This makes it easier for the user to comprehend about certain functions.
- We have used the best and easy to use icons and buttons for the features of the system. A user can easily understand what a particular icon or button is meant for.
- A thorough testing is done with a wide variety of user base and the results are exceptionally positive. The users who tested the app for ease-of-use loved the interface and found it very easy-to-use.

3. Organization of Features:

- The Tourist Assist app is an app which has a lot of features and sometimes it gets difficult for a user to use certain features and ultimately some features are always unnoticed by the user.
- In our design, we have elegantly categorized the features into modules and organized the features in the app such that each and every feature can be used by the user easily.

8. Marketing:

Marketing plays a major and a vital role in success of a mobile application. No matter how good an app is, the end users must get to know that there is an app which will make the life easier for them if they use that app for a particular purpose. And because marketing is so important, we came up with our marketing scheme which will not only promote the app but will also make the users download and use the app whenever they want to go on a tour. We want our users to love the app.

Our Mantra for marketing is very simple. Whenever a person thinks about planning a vacation or a business trip, the first thing that comes to their mind is our Tourist Assist App. The following are the marketing schemes which we came up with:

1) Social Media Marketing

- a) Facebook: There will be a tourist assist page which will have a link to download the app from the app store. The page will promote the app to specified category of users. The age group from 15 to 35 will be targeted the most because these will be the majority of the users using our app. The page will also post features of the app so that the users will get to know about the app and will use it more often.
- b) Twitter: Similar to Facebook, twitter will be used to promote the app by posting tweets about the trip planner and other functionalities.
- c) Instagram: An Instagram profile will post photos from different destinations and will also post trip planning tips so that the users get to know about the app.

2) On-Site Marketing

After getting corporate sponsors, the following places will be targeted for on-site marketing and live interaction of the app with the functionalities will be displayed for the users. This way, users will get to know about the app and will let their friends and family know about it too.

- a) Shopping Malls
- b) Airports
- c) Train Stations

Apart from these marketing schemes, we are also planning to have a share feature which will allow the users to share their trips, destinations, attractions etc on Facebook, Twitter and Instagram. This will inform their friends about the app and thus will help in overall marketing of the app.

We have also included a Rate & Review feature in the app. The users can rate & review their thoughts about the app on the app store. This will increase the quality and ranking of the app and thus help in marketing as well.

9. Analysis of Project Team Performance:

The entire project work was divided into modules and every team member was assigned a module to work on. The deliverables targeted for each and every module were as follows:

1. Use Cases & Use Case Diagram
2. Functional, Non-Functional & Environmental Requirements
3. System Model & System Architecture Model
4. Data Model
5. Component Model
6. Test Plan & Test Cases

All the above deliverables were assigned to every team member for a particular module of the application. The Team Members and the assigned Modules are as follows:

Team Member	Application Module
Mohammed Shethwala	Trip Planner & Dashboard
Ravali Nannuru	Profile & Admin
Ericson Akatakpo	Accommodation, Food & Help
Nishant Sharma	Attractions & Destinations
Gan Xu	Utilities & Travel Mates

The following deliverables were analyzed and done by the team as a collaborative effort by brainstorming together and providing their own views:

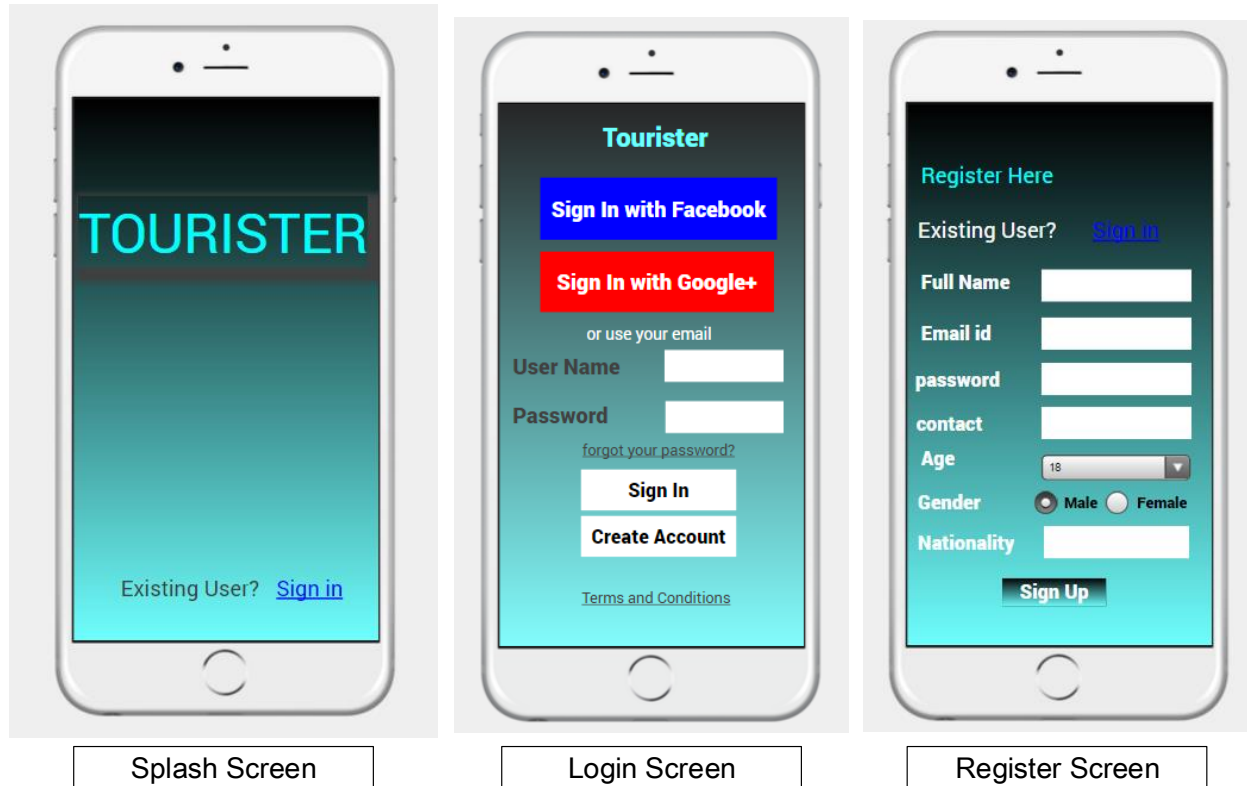
1. Entity Relationship Diagram (Data Model)
2. Component Diagram
3. System Architecture
4. Working Prototype

From the inception of the project, every Thursday the team met and did brainstorming on the upcoming deliverable in the project lifecycle. The team also reviewed each other's previous work and gave any suggestions to improve. Everyone's views and opinions were taken and a final decision on the topic under discussion was made. The whole week after that would be spent in individual work on the assigned modules.

Thus each and every team member's equal contribution to the project made sure that the performance was not compromised and a highly competitive final application was successfully designed & developed.

10. Prototype Documentation & User Guide:

Below are some screenshots of the working prototype developed by the team and a user guide explaining how the prototype works.

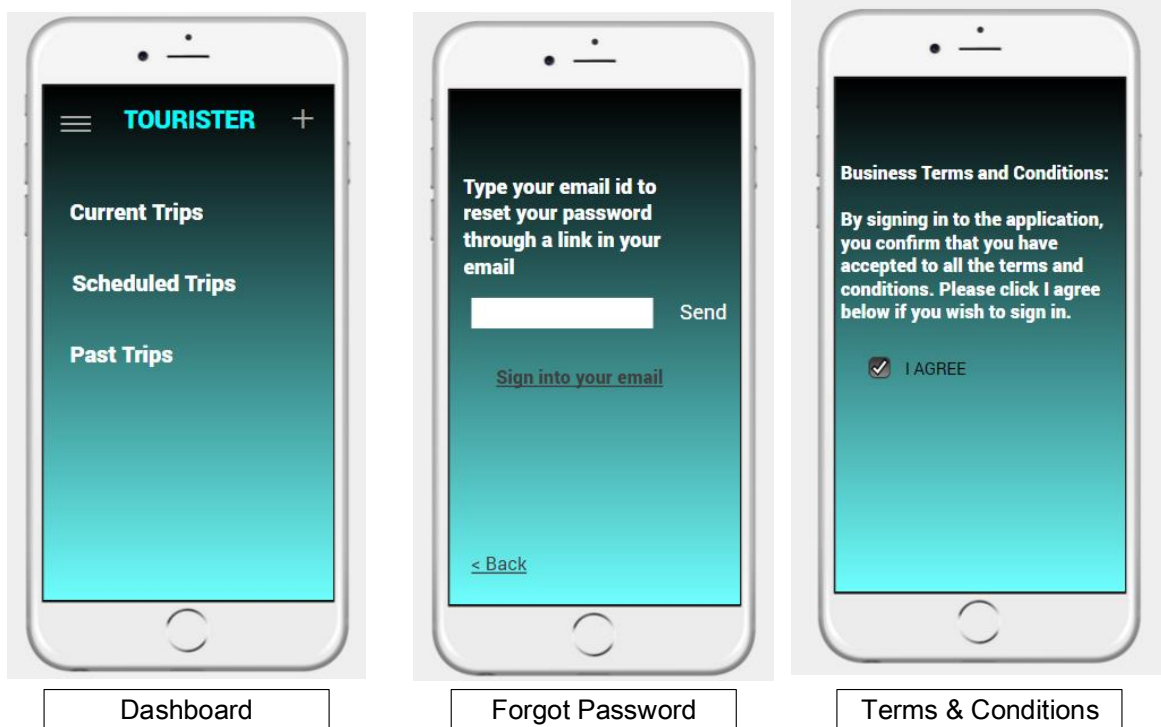


1. Splash Screen: The Splash Screen is the first screen when you open the Tourist Assist App from your iPhone. Clicking on to 'Sign In' button will take you to the Login Screen.

2. Login Screen: This is the screen from where you can login to the app if you are a previously registered user. Entering your username and password will take you to your dashboard. For the first time users, there are 3 options to register with the app

- Sign in with Facebook: This will take you to the Login with Facebook Screen
- Sign in with Google+: This will take you to the Login with Google+ Screen
- Register with your email address: This will take you to the Register Screen

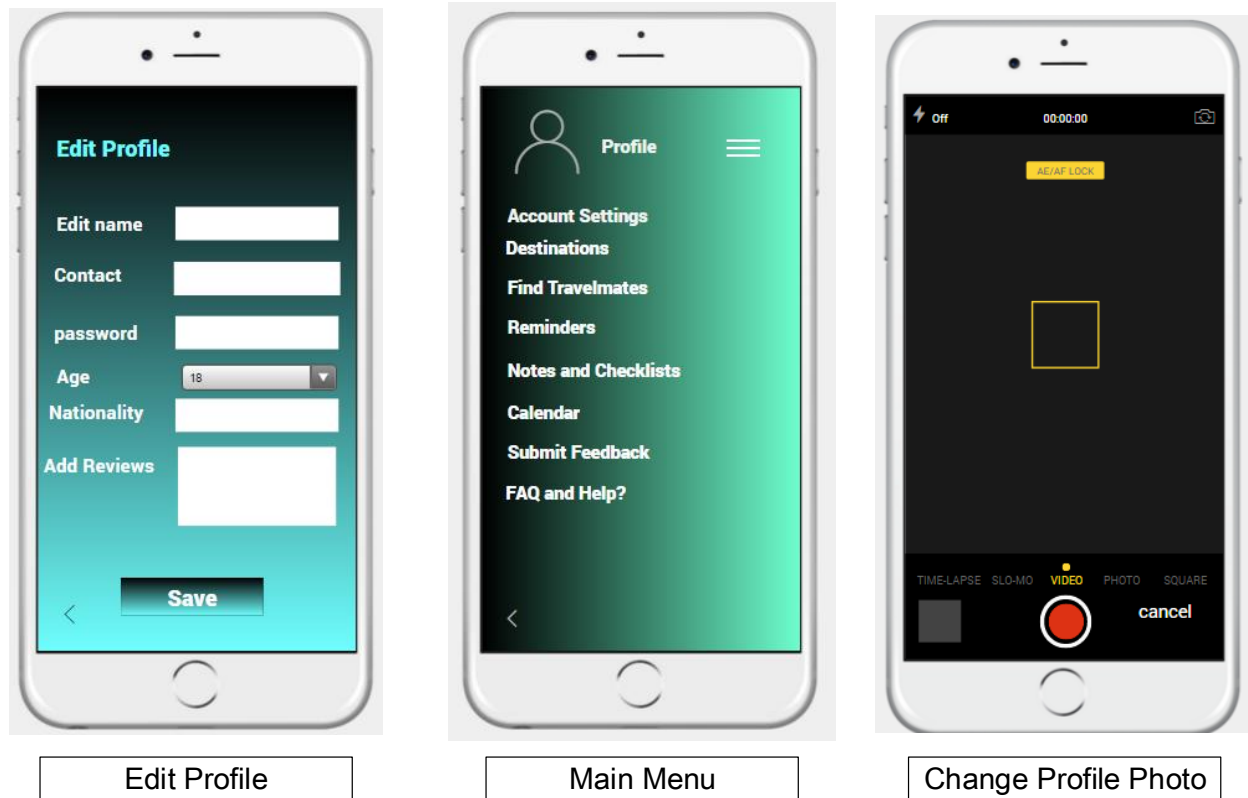
3. Register Screen: Enter your Full Name, Email id, Password, Phone Number, Age, Gender and Nationality in the fields. Click on Sign Up. You will be successfully registered and taken to your dashboard page



4. Dashboard: This is the home page of your app also called as the dashboard in the app. This displays the links to the Current Trips, Scheduled Trips and Past Trips. Clicking on each of the link will take you to the respected pages where the list of trips will be displayed. The '+' on the top right corner is for creating a new trip and will take you to the new trip page. The Menu icon on the top left will take you to the Main Menu of the app.

5. Forgot Password: When you click on the 'Forgot Password' button on the login page, you will be redirected to this page. Enter your email address and press the send button. A temporary password will be sent to your email address which you can use to login and then can change the password again.

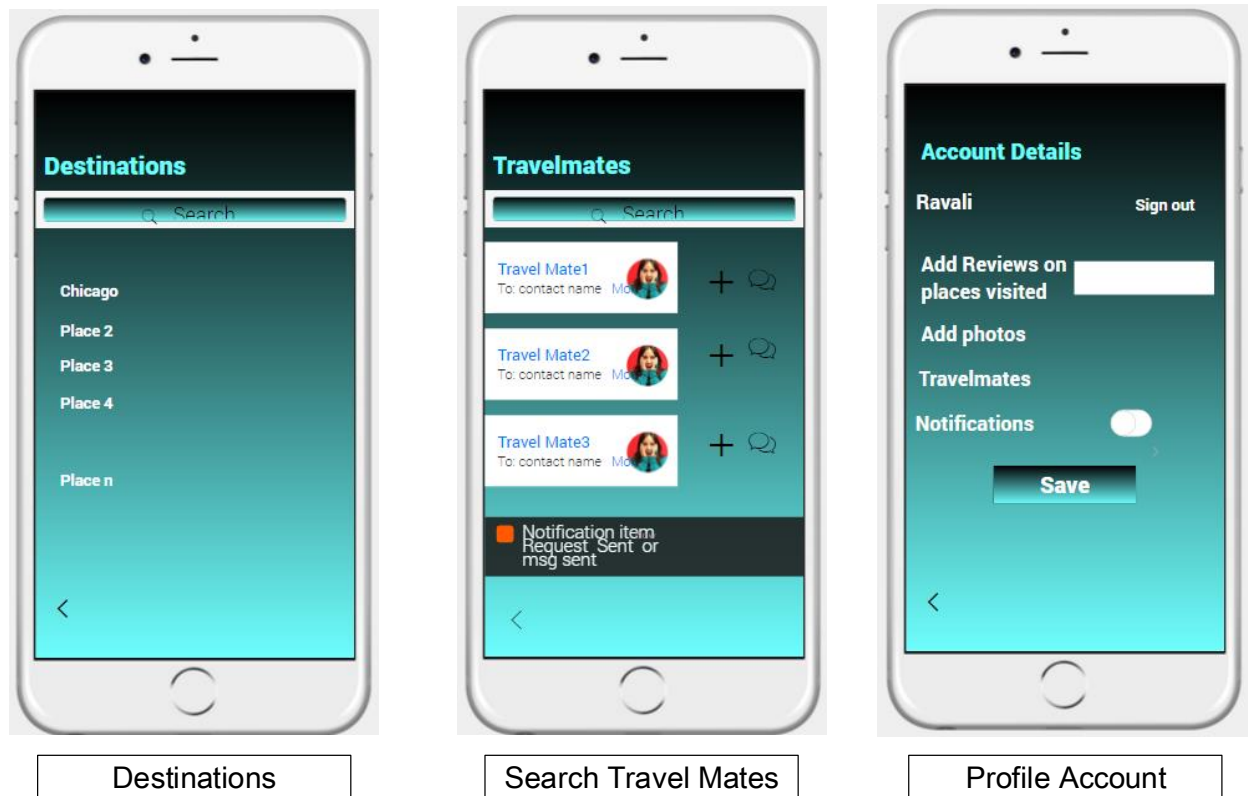
6. Terms & Conditions: Before registering with the app, you will have to read the terms & conditions of the app and check the 'I Agree' button for yourself to be registered successfully.



7. Edit Profile: When you click the profile on the Main Menu, you will be redirected to this page. This page lets you edit all your personal information like name, contact, password, age, nationality and also lets you add reviews. Clicking on the Save button will save all the edited information and will be updated to the profile.

8. Main Menu: This is the main menu of the app. From here you can navigate all the available features of the application. Clicking on the particular feature will take you to the particular feature page. For example, if you click the Find Travelmates button, you will be redirected to the Find Travelmates page.

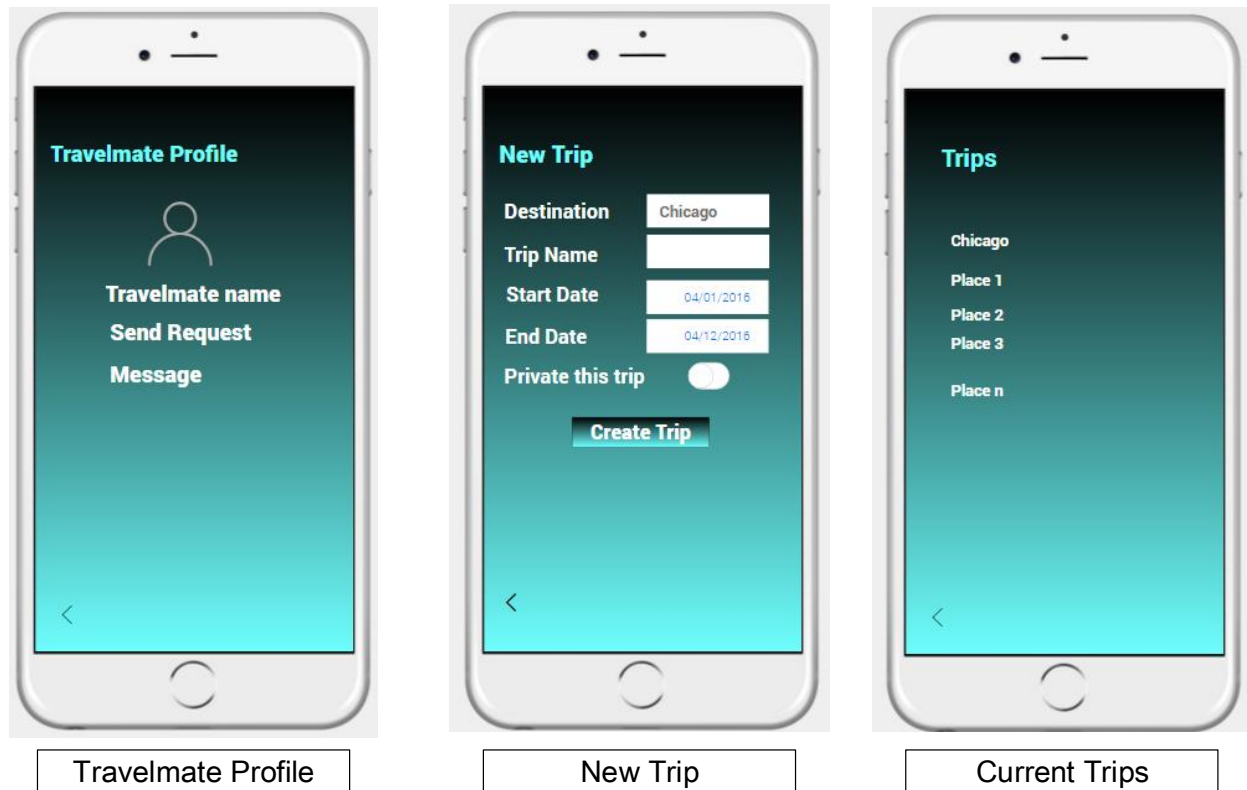
9. Change Profile Photo: When you click the Photo icon on the Main Menu, this camera will open up and will let you select a photo from the gallery or let you capture a new image from the camera.



10. Destinations: The Destinations page shows the featured destinations and also allows you to search for any valid destination. Click on the search bar and enter the name of the destination. After clicking enter, the list of places related to the search key will be displayed in the space below the search bar. From there you can select a particular destination to see the destination related information.

11. Search Travel Mates: This feature lets you see the different users who have planned the trip on the same days as yours. You can message them or send them a friend request. You can also view the TravelMate's profile by clicking on the TravelMate row. You can also search for travel mates by providing the name of the destination in the search bar. All the TravelMates will be shown in the list

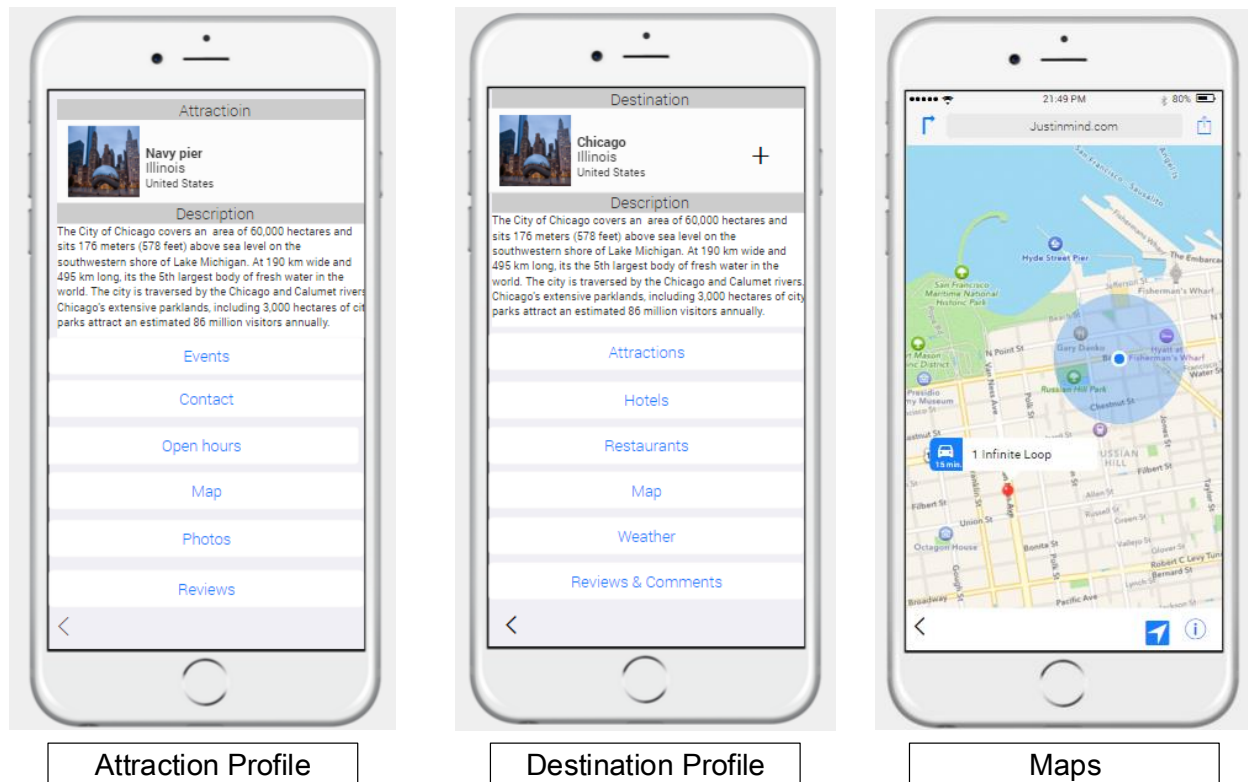
12. Profile Account: When you click the Account Details tab on the Main Menu, you will be redirected to this page. Here you can edit the account details like adding reviews, photos, turning on or off the notifications and you can sign out of the app from this page by selecting the sign out button. Clicking the Save button will save the changed information and will be applied and displayed on the same page again.



13. TravelMate Profile: This page shows all the information about the TravelMate like the Name, Send Request button, Message Button. Based on the security settings of the travelmate you can also see the reviews, photos, destinations visited etc of the travel mate.

14. New trip: This feature lets you add a new trip to your planned trips or set your trip to a current trip. Enter the destination name, trip name, start date, end date and select Private This Trip if you don't want to be displayed in the travel mates section of other users. Pressing the Create trip button will save the trip and will be also shown in the planned trips.

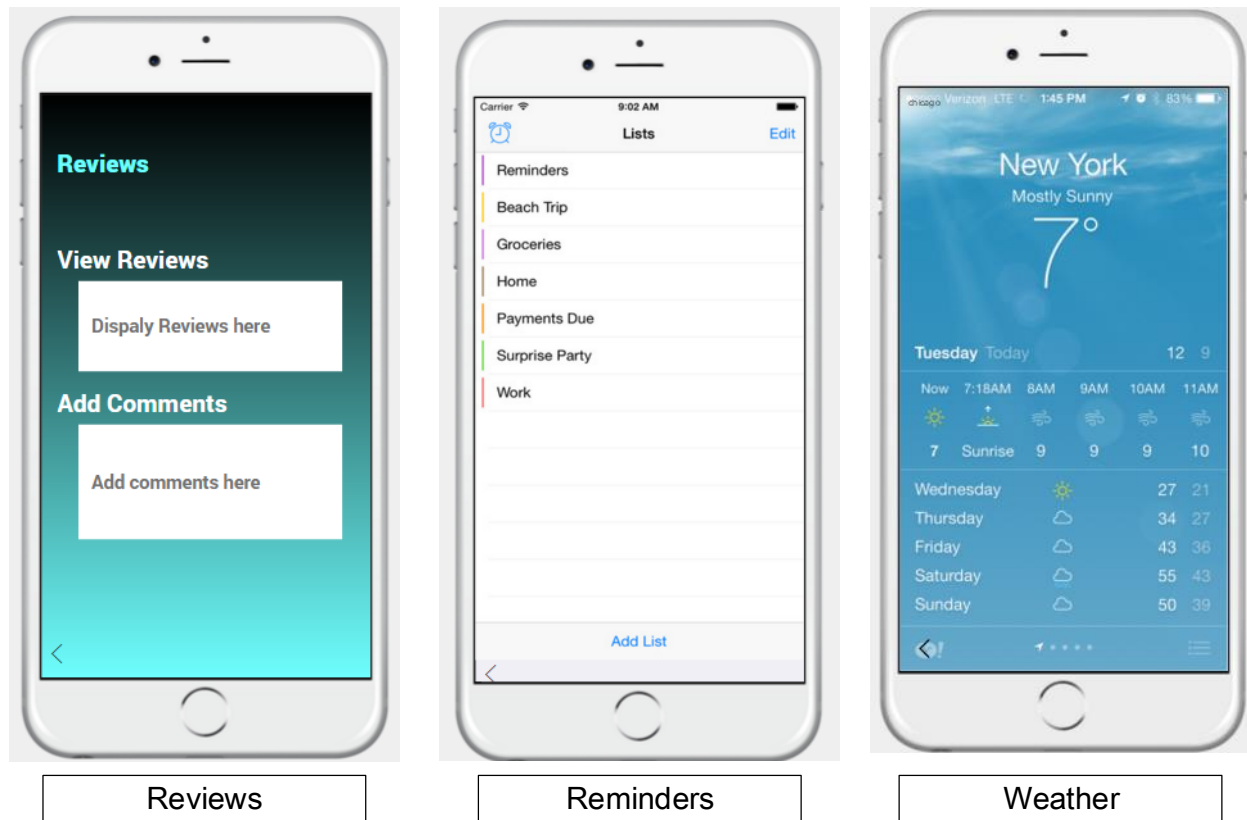
15. Current Trip: This page contains all the details related to your current destinations. All the planned destinations, attractions, photos, reviews etc are all displayed in this page.



16. Attraction Profile: When you click the Attractions button on the Destination Profile page, you will see this page. This page contains all the information about a particular attraction like the attraction name, description of the attraction, events, contact, open-hours and map of the attraction, photos and reviews.

17. Destination Profile: This is the place where you can get complete information about a destination. The Description, Attractions in the destination, Hotels, restaurants, map, weather, reviews and comments are displayed in this section. Clicking on each button will redirect you to that particular page. You can also directly add this destination to your planned or current trip by clicking on the '+' on the top right button.

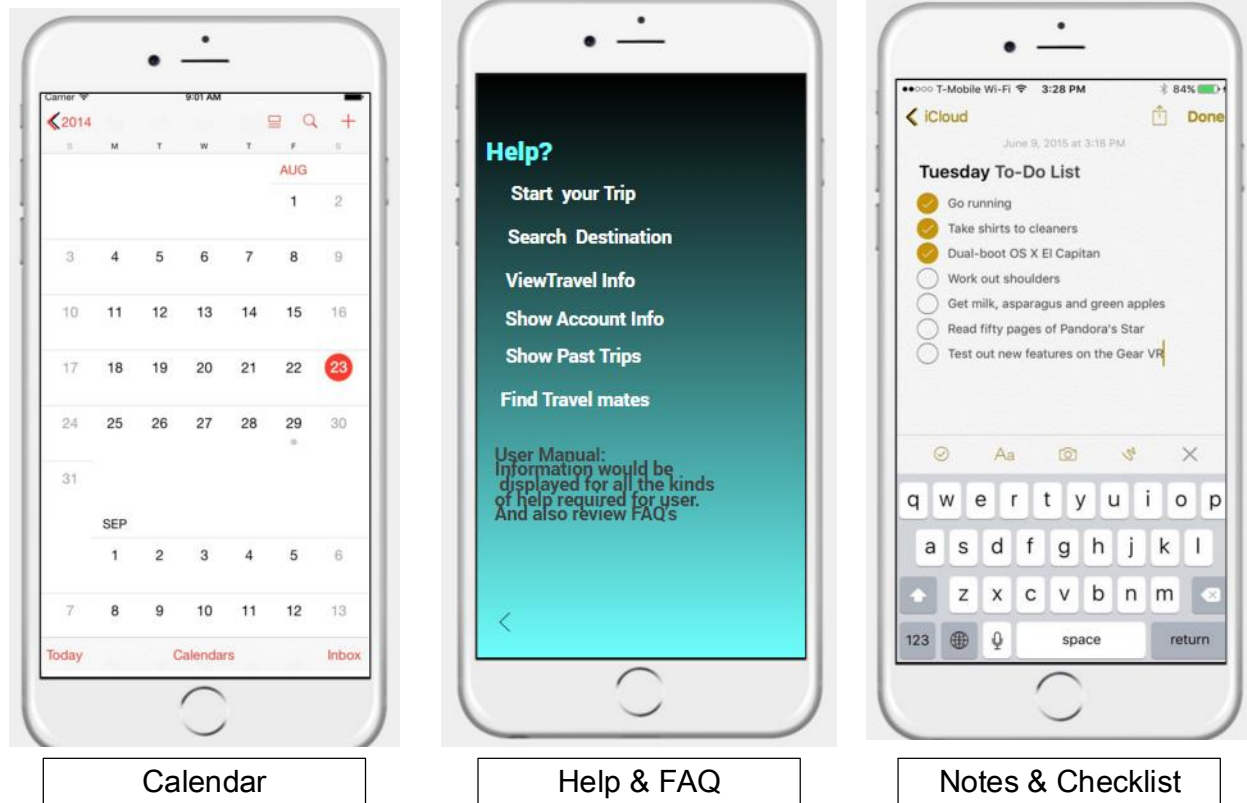
18. Maps: When you click on the Maps from the Destinations or the Attractions, this page will be opened and the map for that particular destination or the attraction will be shown. You can navigate the map and also get directions to attractions in a destination.



19. Reviews: This page is used to add reviews and comments about any destination, attraction, hotel or a restaurant. Write your review in the review section and write the comments in the comments section. That will be saved and added to the list of reviews and comments for whichever thing you selected.

20. Reminders: Reminders lets you remind any important event during a trip. You can add, delete or edit reminders form this page. You can also add reminders to particular trips whether it is a planned trip or a current trip. At the time of the reminder, a reminder notification will be shown on the phone.

21. Weather: If you want to check the weather of a particular destinations you can click on the weather button on the destination and this page will be shown. It shows the complete weather about the destination and also shows 5 day prediction of the weather.



22. Calendar: Calendar lets you see all your planned trips, completed trips, events, reminders in one place. It acts as a timeline and makes your life easier by providing easy to understand calendar to plan things.

23. Help & FAQ: This section is for the users who find it difficult to understand certain features. You can come to this page and can search or select any of the helps provided in the list. Clicking on the help will open a screenshot help which will show you how to use that particular feature.

24. Notes & Checklist: If you want to write some notes during a trip or want to make a checklist of the things you don't want to forget to take with you to the trip, you can do that in this section. By clicking the '+' button you can add notes and checklist and that will be saved on your phone and the trips for which you made the notes or the checklist.