

User Requirement Document For Android Tourist App

Introduction

This document is to demonstrate the user requirements and use cases for the Android Tourist App.

- Document Scope

This document includes the system solution for the App's user interface and backend database.

- Intended Audience

The primary audience:

Users: Tourists that are not willing to type in every single word to search for weather, tourist attractions, hotels, restaurants and so on during their trips.

Professors and Developers

- References

3 - Sample Use Cases and User Requirements Document.pdf

System Overview

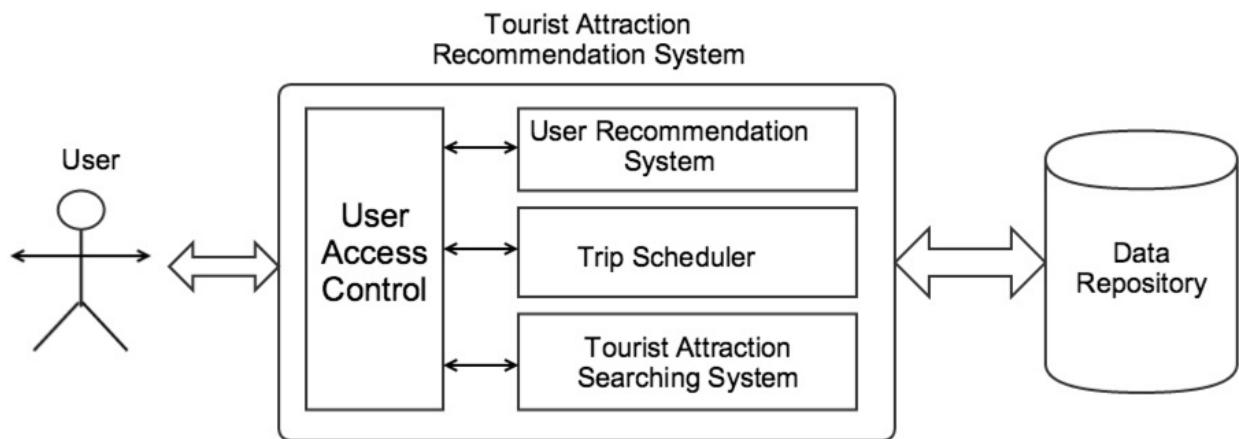


Fig 1. An overview of Tourist Attraction Recommendation System

- Purpose

Tourists are busy, no matter where they are: busy moving from attraction to attraction, busy planning for tomorrow's agenda when back to hotel. We want to build an App for those tourists who are exhausted searching for hotels,

restaurants and attractions from time to time. With this tourist App, users can just enter a location and the time they want to travel, and get all the information prepared to them.

- **Scope**

This App provides users with the weather, local attractions, hotel and restaurant information all within. Logged in users can get more: recommended destinations from similar tourists, previous search results and clicked attractions. Users can even search for users who are planning to go to the same destination on the same day!

General Requirements

In this project, we want to have a free-to-use App with user-friendly user interface. Users want to have the most useful information with least keywords entered. The efficiency of generating the recommended attractions is also very important.

- **System In Context**

The tourist App would realize the following functionalities:

- 1) User Registration and logging in
- 2) Database schema for storing users' profile and previous search results
- 3) User interface that shows search results and the recommended content

- **User characteristics**

The target users are non-technical individuals that want to enter as least information as possible to get most useful searching results and recommended content.

- 1) Want to enter just the time and destination city/area, but get sufficient content that can be used for planning an exciting trip.
- 2) Want to be informed of some recommended attractions/ restaurants/ hotels.

- **Constraints**

- 1) Reliable search results with sufficient data
- 2) Secure login mechanism
- 3) Easy-to-use user interface

- **System-wide Requirements (Received)**

Actors

The App system recognizes three types of users: Not logged-in users, logged-in users and administrators.

Events

The App aims to provide tourist related information given the destination and travel date. The most critical events are:

- 1) Creating a new user
 - 2) User log in
 - 3) Search for attractions/ weather/ hotels/ restaurants/ nearby attractions
 - 4) Calculate the scores of the attractions
 - 5) Add to/ update/ delete from schedule
 - 6) Verify email address
 - 7) Recommend attractions
 - 8) Update profile
- System-wide Requirements (Derived)

Actors

1. Not logged-in users
 - Creating a new user
 - User log in
 - Search for attractions/ weather/ hotels/ restaurants/ nearby attractions
 - Calculate the scores of the attractions
 - Verify email address
2. Logged-in users
 - Search for attractions/ weather/ hotels/ restaurants/ nearby attractions
 - Calculate the scores of the attractions
 - Add to/ update/ delete from schedule
 - Recommend attractions
 - Update profile
3. Administrators
 - Calculate the scores of the attractions
 - Verify email address

Detailed Requirements

- Functional Requirements

Table 1

Use Case ID: CRT_USR	Use Case Name: Create user
Primary Actor(s):	Not logged-in user
Secondary Actor(s):	N/A

Description:	Request username and password to create an account
Preconditions:	User sign up screen is available
Normal Flow of Events:	<ol style="list-style-type: none"> 1. User enters username and password 2. Check if the username is good to use 3. Add user account to database
Postconditions:	After signing up, prompt user to enter profile information, including the email address for further verification.
Frequency of Use:	High
Alternative Flows:	User can click cancel to quit the process.
Exceptions:	If the username is already in use, quit the process to ask for the user to enter another one. If the re-entered password is not the same as former one, prompt the user to enter it again.
Assumptions:	The App is on and connected to the internet.
Issues:	User has already registered the same email address.
Source:	TBD
Includes:	N/A
Associated Requirements:	User log in

Table 2

Use Case ID: LOG_IN	Use Case Name: User log in
Primary Actor(s):	Not logged-in user
Secondary Actor(s):	N/A

Description:	If user wants to use the functionalities that only logged-in users can use, he/she will have to log-in first.
Preconditions:	The user already has an account
Normal Flow of Events:	<ol style="list-style-type: none"> 1. User enters username/ email address and password 2. System verifies the user's information and approves the log-in process 3. User is logged-in and the status is updated on the system side.
Postconditions:	User is logged in and all those functionalities are open to this user.
Frequency of Use:	High
Alternative Flows:	User can cancel the process in the meantime.
Exceptions:	If user has entered wrong information, prompts the user to enter it again. If trying too many times, abort the user's privilege to log in for certain amount of time and send a warning email to the email address.
Assumptions:	The App is on and connected to the internet, and the user already has an account.
Issues:	If the user has no account, then this process can't be done.
Source:	TBD
Includes:	N/A
Associated Requirements:	Create user

Table 3

Use Case ID: SCH_ATT	Use Case Name: Search for attractions
----------------------	---------------------------------------

Primary Actor(s):	Logged-in user or Not logged-in user
Secondary Actor(s):	N/A
Description:	The user enters a destination and travel time, the system will return back the necessary information for the user to plan his/her trip, of which the attractions around that destination is of core importance.
Preconditions:	The user has to enter the destination and travel time.
Normal Flow of Events:	<ol style="list-style-type: none"> 1. User enters destination and travel time 2. System use google API to search for the attractions near the destination and returns the introduction of them. 3. The UI will show the abstract of the results in a user-friendly way.
Postconditions:	User will choose several interested attractions to see the details of them, and user will choose some of them and add to his/her schedule.
Frequency of Use:	High
Alternative Flows:	User may modify what he/she entered and re-enter the information. If no attractions are found near the destination, ask the user if he/she wants to re-enter a different destination.
Exceptions:	If no results can be shown, ask the user to enter another destination.
Assumptions:	The destination is a valid address and there are attractions nearby.
Issues:	Some destination city has no attractions.
Source:	TBD
Includes:	N/A

Associated Requirements:	Add to schedule/ Calculate site score
--------------------------	---------------------------------------

Table 4

Use Case ID: SCH_WTH	Use Case Name: Search for weather
Primary Actor(s):	Logged-in user or Not logged-in user
Secondary Actor(s):	N/A
Description:	Weather is very important for travellers. User can search for the weather forecast of the destination city on given date.
Preconditions:	Destination is a valid address.
Normal Flow of Events:	<ol style="list-style-type: none"> 1. User enters destination and time 2. System search for the weather information 3. UI shows the information to the user
Postconditions:	N/A
Frequency of Use:	High
Alternative Flows:	User can cancel the process
Exceptions:	The destination is not a valid address.
Assumptions:	The user entered a correct destination
Issues:	The destination is not a valid address.
Source:	TBD
Includes:	N/A
Associated Requirements:	Calculate weather score

Table 5

Use Case ID: SCH_RSTA	Use Case Name: Search for restaurants
-----------------------	---------------------------------------

Primary Actor(s):	Logged-in user or Not logged-in user
Secondary Actor(s):	N/A
Description:	User will want to search nearby restaurants when travelling.
Preconditions:	User has entered correct destination.
Normal Flow of Events:	<ol style="list-style-type: none"> 1. User enters the destination and time 2. System search for restaurants and show them to user
Postconditions:	N/A
Frequency of Use:	High
Alternative Flows:	User can cancel this process
Exceptions:	The destination is not a valid address.
Assumptions:	The user entered a correct destination
Issues:	The destination is not a valid address.
Source:	TBD
Includes:	N/A
Associated Requirements:	Calculate site score

Table 6

Use Case ID: SCH_HOTEL	Use Case Name: Search for hotels
Primary Actor(s):	Logged-in user or Not logged-in user
Secondary Actor(s):	N/A
Description:	Search for nearby hotels
Preconditions:	User has entered correct destination.
Normal Flow of Events:	<ol style="list-style-type: none"> 1. User enters the destination and time

	2. System search for nearby hotels and show them to user
Postconditions:	N/A
Frequency of Use:	High
Alternative Flows:	User can cancel this process
Exceptions:	The destination is not a valid address.
Assumptions:	The user entered a correct destination
Issues:	The destination is not a valid address.
Source:	TBD
Includes:	N/A
Associated Requirements:	Calculate site score

Table 7

Use Case ID: SCH_NEAR_ATT	Use Case Name: Search for nearby attractions
Primary Actor(s):	Logged-in user or Not logged-in user
Secondary Actor(s):	N/A
Description:	Search for nearby attractions when a user click certain attraction and add it to his/her schedule.
Preconditions:	User has entered correct destination and select one of the attractions.
Normal Flow of Events:	<ol style="list-style-type: none"> 1. User enters the destination and time 2. System search for attractions in the given area 3. User select one of the attractions and add it to his/her schedule. 4. System search for attractions that are near to the selected attraction and recommend them to the user.

Postconditions:	N/A
Frequency of Use:	High
Alternative Flows:	User can cancel this process
Exceptions:	The destination is not a valid address.
Assumptions:	The user selects one of the attractions.
Issues:	The destination is not a valid address.
Source:	TBD
Includes:	N/A
Associated Requirements:	Calculate site score

Table 8

Use Case ID: CALCPH-SCRE	Use Case Name: Calculate comprehensive score
Primary Actor(s):	Administrator
Secondary Actor(s):	N/A
Description:	The function is used to calculate a comprehensive score of recommendation for visiting a specific city at a specific time
Preconditions:	Users should input and submit the name of city and time correctively.
Normal Flow of Events:	1. Users input the city and time to visit. 2. Click submit to submit the form.
Postconditions:	It shows the comprehensive score of recommendation on the next page.
Frequency of Use:	Frequently
Alternative Flows:	The user click reset and it will not jump to the next page.
Exceptions:	If the user submit a wrong city or invalid time, it will give an exception.

Assumptions:	The user input the city and time correctly
Issues:	TBD
Source:	TBD
Includes:	N/A
Associated Requirements:	TBD

Table 9

Use Case ID: CALWEA-SCRE	Use Case Name: Calculate weather score
Primary Actor(s):	Administrator
Secondary Actor(s):	N/A
Description:	The function is used to calculate a weather score of recommendation for a specific city at a specific time.
Preconditions:	The city exists and the weather report for that time is valid.
Normal Flow of Events:	1. Users input the city and time to visit. 2. Click submit to submit the form.
Postconditions:	It shows the weather score of recommendation on the next page.
Frequency of Use:	Frequently
Alternative Flows:	The user click reset and it will not jump to the next page.
Exceptions:	If the city name is non-existing or the weather at that time is invalid, it will run an exception.
Assumptions:	The user input the name of the city and time to visit correctly.
Issues:	TBD
Source:	TBD

Includes:	N/A
Associated Requirements:	TBD

Table 10

Use Case ID: CALSIT_SCRE	Use Case Name: Calculate site score
Primary Actor(s):	Administrator
Secondary Actor(s):	N/A
Description:	The function is used to calculate the score of recommendation for a specific city at a specific time for a specific tourist attraction.
Preconditions:	The related resources can be found for the tourist attraction.
Normal Flow of Events:	1. Users input the city and time to visit. 2. Click submit to submit the form.
Postconditions:	It shows scores of recommendation for all the tourist attractions available on the next page.
Frequency of Use:	Frequently
Alternative Flows:	The user click reset and it will not jump to the next page.
Exceptions:	N/A
Assumptions:	Enough related resource for a site can be found to make the score right.
Issues:	If the related resource for a site is limited, it may not give the score objectively.
Source:	TBD
Includes:	N/A
Associated Requirements:	N/A

Table 11

Use Case ID: DEL_SCHEALL	Use Case Name: Delete all schedules
Primary Actor(s):	Logged-in user
Secondary Actor(s):	N/A
Description:	The function is used to delete all the schedules from a user's trip scheduler.
Preconditions:	The user is logged-in.
Normal Flow of Events:	Click Delete All in the trip scheduler page
Postconditions:	The related content of the trip schedule would be delete.
Frequency of Use:	Medium
Alternative Flows:	The user choose to update or add a schedule.
Exceptions:	TBD
Assumptions:	There are at least one schedule exited in the trip scheduler for the logged-in user.
Issues:	TBD
Source:	TBD
Includes:	N/A
Associated Requirements:	TBD

Table 12

Use Case ID: ADD_SCHE	Use Case Name: Add to schedule
Primary Actor(s):	Logged-in user.
Secondary Actor(s):	N/A
Description:	The function is used to add a travel schedule in the trip scheduler
Preconditions:	The user is logged-in.

Normal Flow of Events:	The user click “add it to the trip scheduler” in the site page.
Postconditions:	It jumps to the page for adding a schedule of trip scheduler.
Frequency of Use:	Medium
Alternative Flows:	The user returns from the site page
Exceptions:	TBD
Assumptions:	The site page is valid.
Issues:	TBD
Source:	N/A
Includes:	TBD
Associated Requirements:	TBD

Table 13

Use Case ID: UPD_SHCE	Use Case Name: Update schedule
Primary Actor(s):	Logged-in user
Secondary Actor(s):	N/A
Description:	The function is used to update a schedule from a user’s trip scheduler.
Preconditions:	The user has logged in and at least one schedule has existed.
Normal Flow of Events:	1.Choose a schedule from the trip scheduler. 2.Click update
Postconditions:	It jumps to the page to update the trip schedule
Frequency of Use:	Medium

Alternative Flows:	The user choose to add or delete a trip schedule.
Exceptions:	TBD
Assumptions:	The user has at least one existed travel schedule.
Issues:	TBD
Source:	N/A
Includes:	TBD
Associated Requirements:	TBD

Table 14

Use Case ID: DEL_SCHE	Use Case Name: Delete from schedule
Primary Actor(s):	Logged-in users
Secondary Actor(s):	N/A
Description:	The function is used to delete a specific schedule from the trip scheduler.
Preconditions:	The user has logged in.
Normal Flow of Events:	1.Select a trip schedule which has been made. 2.Click Delete.
Postconditions:	The corresponding entry will be deleted.
Frequency of Use:	Medium.
Alternative Flows:	The user choose to update or add a schedule
Exceptions:	TBD
Assumptions:	The user has at least one trip schedule.
Issues:	TBD
Source:	N/A

Includes:	TBD
Associated Requirements:	TBD

Table 15

Use Case ID:	Use Case Name: Verify email address
Primary Actor(s):	administrator
Secondary Actor(s):	N/A
Description:	Verify the user's email address when they try to log in
Preconditions:	Has already stored signed up users' email address in database, and the use has submit his email address when he try to log in
Normal Flow of Events:	<ol style="list-style-type: none"> 1. User enters user ID in ID field textbox and then taps or clicks into password field and enters password. 2. User clicks <i>ok</i>. 3. Administrator verify the input email with the signed up users email in database, is true, exceed the process.
Postconditions:	The user successfully signed in.
Frequency of Use:	High
Alternative Flows:	TBD
Exceptions:	The input email does match with the emails in database. User is given feedback that the email address is not right and return step 1 to re-input.
Assumptions:	To verify user's email address schedule is under the administrator's access right.
Issues:	TBD
Source:	APP Model and database

Includes:	TBD
Associated Requirements:	TBD

Table 16

Use Case ID:	Use Case Name: Recommend attractions
Primary Actor(s):	administrator
Secondary Actor(s):	N/A
Description:	Recommend some populations scenic spots for users
Preconditions:	Has already stored users' travel records in database, and has summed up some popular spots
Normal Flow of Events:	<ol style="list-style-type: none"> 1. User selects <i>Recommend attractions</i>. 2. APP displays list of scenic spots which are popular.
Postconditions:	Return the result of step 2
Frequency of Use:	High
Alternative Flows:	TBD
Exceptions:	Not logged-in user click on <i>Recommend attractions</i> . User is given feedback that user must login first.
Assumptions:	To recommend attractions is under the administrator's access rights.
Issues:	TBD
Source:	APP Model and database
Includes:	TBD
Associated Requirements:	TBD

Table 17

Use Case ID:	Use Case Name: Search for similar tourists
Primary Actor(s):	logged-in users
Secondary Actor(s):	N/A
Description:	Users could search for tourists who has similar travel schedule with them
Preconditions:	User has already logged-in and scheduled their travel
Normal Flow of Events:	<ol style="list-style-type: none"> 1. User selects <i>Search for similar tourists</i>. 2. APP displays list of attractions which may in similar tourists' schedule.
Postconditions:	Return the result of step 2
Frequency of Use:	Medium
Alternative Flows:	TBD
Exceptions:	<p>There is no similar tourist with the user. User is given feedback that selection is needed. Returns to step 1.</p> <p>Not logged-in user click on <i>Recommend attractions</i>. User is given feedback that user must login first.</p>
Assumptions:	To search for similar tourists is under the logged-in user's access rights.
Issues:	TBD
Source:	APP Model and database
Includes:	TBD
Associated Requirements:	TBD

Table 18

Use Case ID:	Use Case Name: Update profile
--------------	-------------------------------

Primary Actor(s):	logged-in users
Secondary Actor(s):	N/A
Description:	Logged-in user could change their profile
Preconditions:	User has already logged-in and the user profile page is displayed
Normal Flow of Events:	<ol style="list-style-type: none"> 1. User selects <i>Update profile</i>. 2. APP displays list of user information. User selects one and hits a <i>edit</i> button. 3. The selected item turn into a blank and user need to input new information. 4. User click <i>return</i> on keyboard to proceed with update.
Postconditions:	Returns to the screen displayed on step 2, user is able to update another schedule.
Frequency of Use:	Medium
Alternative Flows:	TBD
Exceptions:	User's new information inputted does not meet that item's input requirements. User is given feedback that selection is now allowed. Returns to step 2.
Assumptions:	To update profile is under the logged-in user's access rights.
Issues:	TBD
Source:	APP Model and database
Includes:	TBD
Associated Requirements:	TBD

Table 19

Use Case ID:	Use Case Name: User logout
--------------	----------------------------

Primary Actor(s):	logged-in users
Secondary Actor(s):	N/A
Description:	Logged-in users could log out their accounts
Preconditions:	User has already logged-in
Normal Flow of Events:	1. User clicks <i>logout</i>
Postconditions:	turn to the page with not logged-in status
Frequency of Use:	High
Alternative Flows:	TBD
Exceptions:	TBD
Assumptions:	To logout is under the logged-in user's access rights.
Issues:	TBD
Source:	APP Model and database
Includes:	TBD
Associated Requirements:	TBD

Table 20

Use Case ID:	Use Case Name: Download schedule
Primary Actor(s):	logged-in users
Secondary Actor(s):	N/A
Description:	Download travel records from database
Preconditions:	User is already log in and the download page is displayed
Normal Flow of Events:	<ol style="list-style-type: none"> 1. User selects <i>travel records</i>. 2. APP displays list of schedules. User selects one and hits a <i>Download</i> button.

	<ol style="list-style-type: none"> 3. APP asks for confirmation to <i>Download</i>. 4. User chooses to proceed with download.
Postconditions:	Returns to the screen displayed on step 2, user is able to download another schedule.
Frequency of Use:	Medium
Alternative Flows:	<ol style="list-style-type: none"> 1. If there are no records displayed in step 2, user can only select Back. 2. User selects to not proceed with download confirmation. User is returned to step 2.
Exceptions:	User does not select a record to download. User is given feedback that selection is needed. Returns to step 2.
Assumptions:	Download schedule is under the logged-in user's access rights.
Issues:	TBD
Source:	APP Model and database
Includes:	TBD
Associated Requirements:	TBD

- Technical Requirements

Operational Environment

The system will be running on Android devices such as Android smartphones and tablets. The App will be connected to system servers via secure internet connection.

Development Environment

Developers must have access to a modern object oriented development environment and support libraries with their corresponding APIs. Developers should have access to GUI development tools for efficiency.