Preliminary Design Report

Objective:

The purpose of the application “Tourist Assist” is to guide the tourist with regard to the tourist spots in a city. This application allows the user to locate the nearby attractions based on the category the user selects. This category includes many options such as Gardens, Golf Courses, Mosques, Museums, Parks, Airports, etc. This application will also let the user pick a place based on the reviews, locate the route (Via Google Maps) and also make reservations in any hotel across the city if required (Via Expedia). The user also has an option to rate a place and give feedbacks.

Introduction:

The Tourist assist is a web based application. In this system, there are multiple users. Users are Tourists, admin, hotel manager, Restaurant manager.

Category of users:

1. End Users: Tourists, Hotel and Restaurant managers.
2. Backend Users: Admin

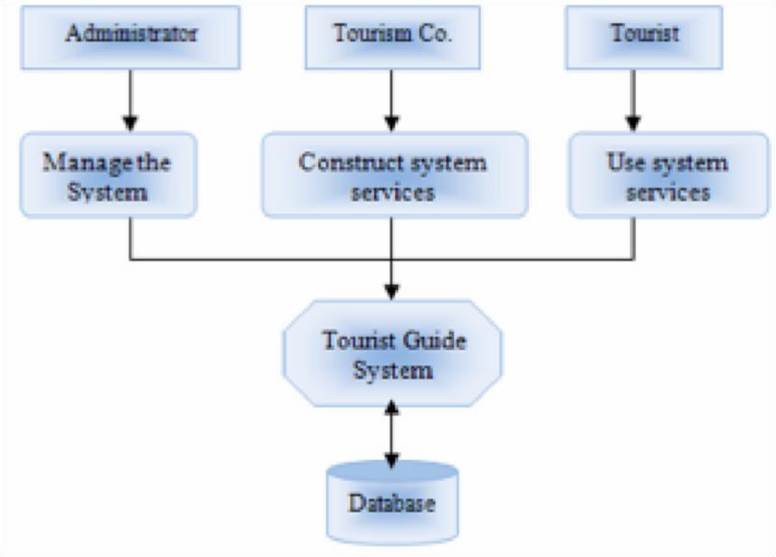


Figure 1: Category of Users

Users can perform following operations.

- Login/Logout
- Choose the language
- Find the places, hotels and restaurants.
- See the location on map.
- See the route.(Via Google Map)
- Book the hotels and transportation facilities.
- Read reviews.
- Give feedback.

Admin functionalities:

- Login/ Logout.
- Add, delete, and updates the hotels, restaurants, and places to visit.
- Delete inappropriate comments.

Requirements:

1. Functional Requirements:

The tourist assist system allows different types of users to interact with the system according to provided privileges.

There are three main users of the system: administrator, Tourism Company, and tourist.

- Authentication: System must authenticate the users. If the valid user id and password meets then only user can get login.
- Administrative functions: admin functions like keep the records of hotel, restaurant availability, system must do it properly.

2. Non-Functional Requirements:

- Usability: The system should provide a coherent and easy user interface that can be used easily with all kind of users.
- Supportability: The system should be easily modified or extended easily.
- Security: Is the system preventing unauthorized users to access the system?
- Scalability: System shall be scalable to add more hotels/restaurants and users.
- Security: System should keep the users information secure.
- Data Integrity: System should be easily integral with other partner application line mapping application.

System Requirement:

Software

- Front-End – HTML, CSS, Java Server Pages (JSP) and JavaScript(JS).
- Back-End - Java Serialization

Hardware

- Processor : Intel i7
- RAM: 8 GB.
- Browser : Chrome
- Server : Apache Tomcat 8.0
- Editor : Eclipse Mars 1

Minimum Hardware Required

- Processor : Intel i3
- RAM: 4 GB.
- Browser : Chrome/Firefox/Opera

System overview:

The "Tourist Assist" system represents a relatively new modern trend in the field of tourism. There is a user interface for front end and back end users to interact with the application.

Use case Diagram:

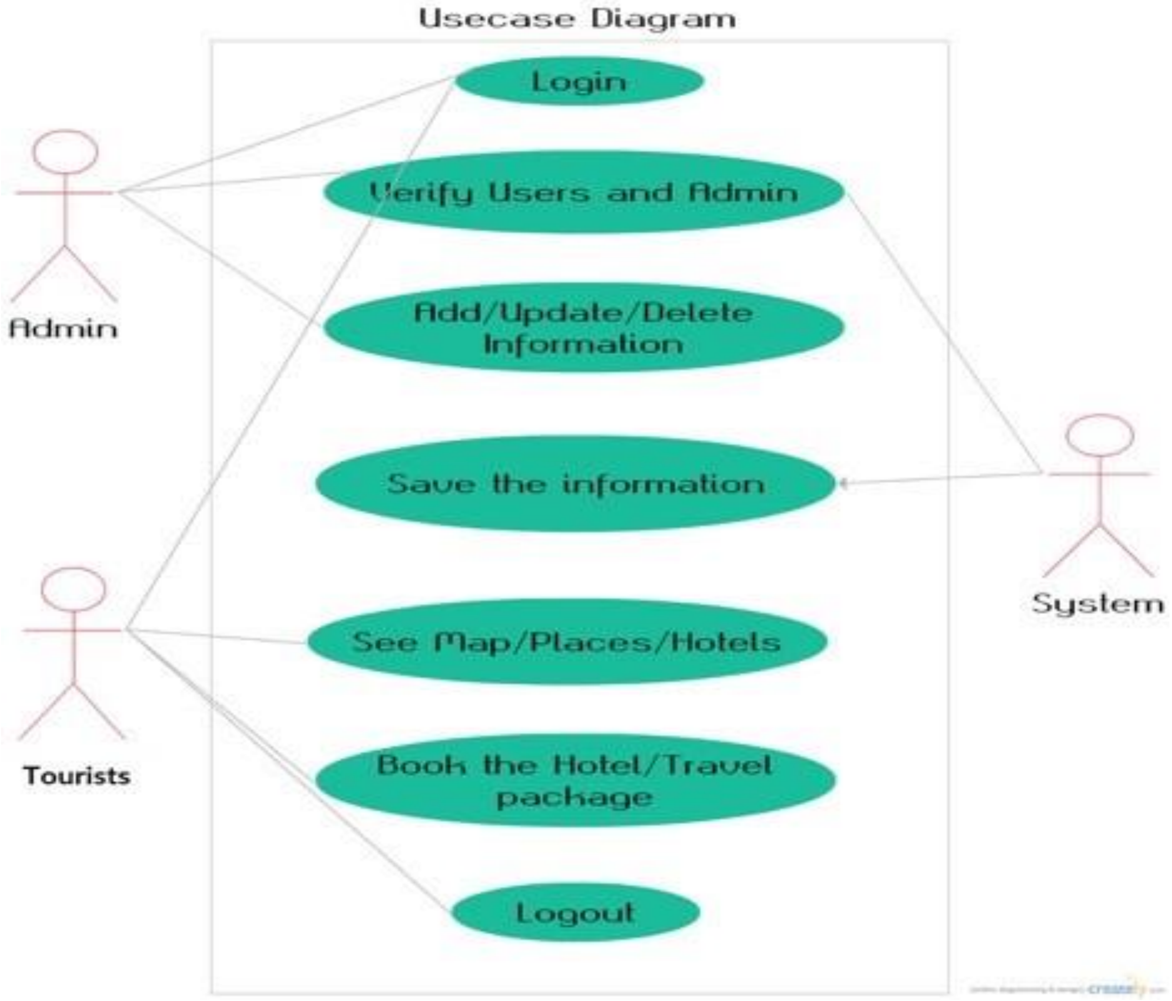


Figure 2: Use-case Diagram

Class Diagram:

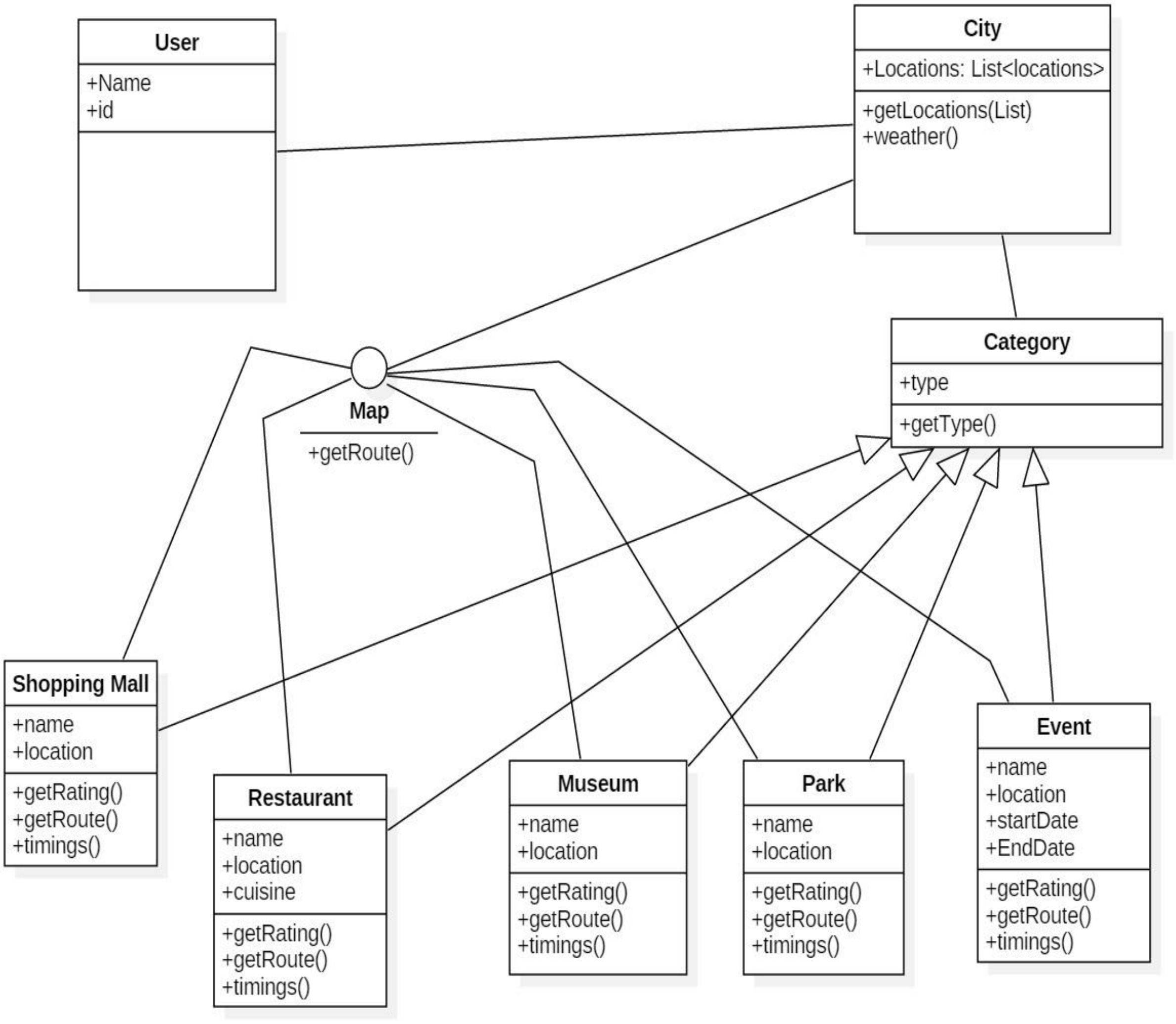


Figure 3: Class Diagram

Context Diagram:

Context Diagram gives the picture of the architecture of the system and connections between the systems and entities.

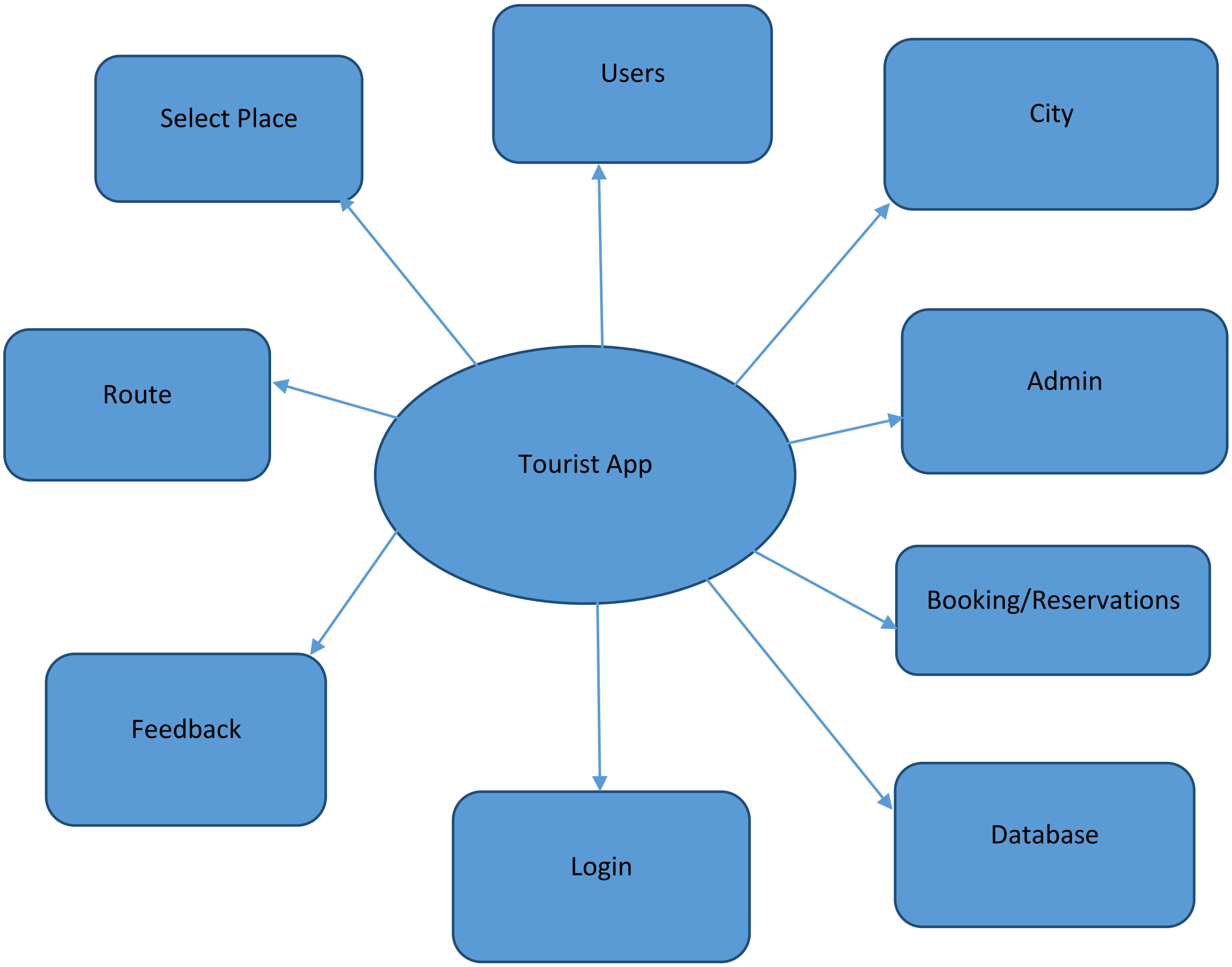


Figure 4: Context Diagram

Login Page:

Objective:

In this module we have given the login access for the tourist/admin. All the users can perform the operation by logging in with their credentials to use the different modules as specified.

Category of users

1. End user: Tourist.
2. Backend user: Admin

<u>User Analysis</u>

1. Tourist: For using the modules which are specified in the application, the tourist needs to login.
2. Admin: The Admin is categorized as the back end user. They maintain the overall functionality of the application by maintaining the database and all the interfaces. The admin is authorized to delete any data or add any data on to the application.

Requirement Specification:

1. Functional

- Both the users Tourist/Managers (hotel or restaurant) / Admin will have their own usernames and passwords to access the application.
- After the login is authenticated the users can access the application.
- Tourist can perform operations such as locating the nearby places to visit, pick a category that suits his search requirements, locate the distance from his current location(*via google maps, by clicking on their links)*, read reviews about the places that he/she has picked, give feedback, choose hotels, restaurants(*via links provided*) and make bookings if required.
- Admin has full access on the back end as well as on the front end of the system.
- Admin has the authority to make any changes in the database.

2. Non Functional

- The Login page should have good response time without any hassles.
- The login credentials provided by the users should be correct to grant the permission by the admin to use the application.
- Systems should be compatible to provide multiple users to perform multiple operations at the same time.

User and system interaction:

1. Tourist will be able to access the application through a web based application. They have to authenticate their login to access the app.
2. Admin is the only one who can have access to all the modules and the database wherein he can edit and maintain the application.

System and Data Models

High Level diagram

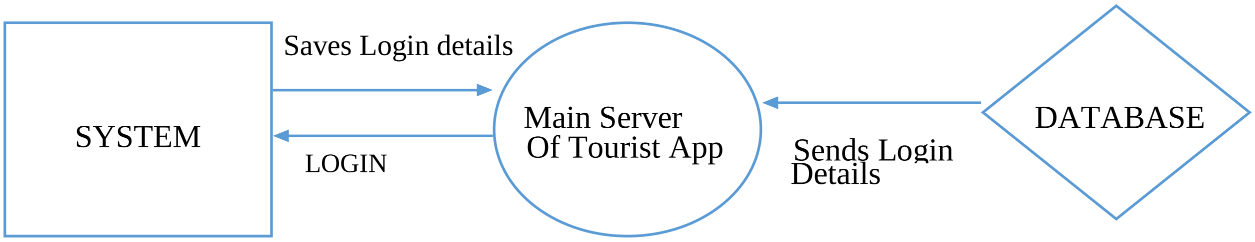


Figure 5: Login interface

Data models

Flow of information

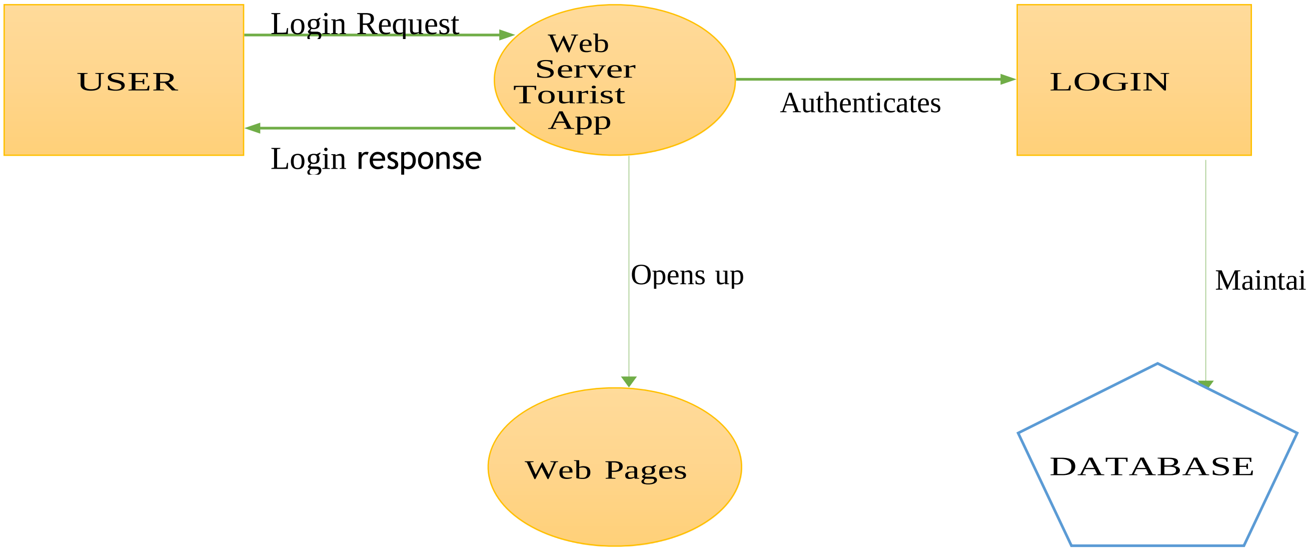


Figure 6: Flow Diagram

Storage and Retrieval:

We use a database system to store all data for the Login Page. Following are the attributes stored per user category within the database:

USER CATEGORY	ATTRIBUTES SAVED IN DATABASE
TOURIST	Login ID Password
ADMIN	Login ID Password

Category System:

Objective:
In this module a tourist can pick a visiting spot based on the category he selects. If he wants to visit a museum nearby, he/she will be able to access the category system when under the museum category he/ she will be able to pick the museum according to their choice.

<u>Category of users</u>

1. End users: Tourist
2. Backend user: Admin

<u>User analysis</u>

1. Tourist: Tourist can pick a category and then select the spot he/she wants to visit.
2. Admin: After a tourist has picked a particular spot to visit, the admin will have to authenticate it and update the database.

Requirement Specifications :

1. **Functional**

- Tourist can select a category according to their choice and pick a place to visit.
- Tourist can access the reviews posted by the people who have already been to that place before.
- On finalizing the visiting spot, the tourist should be able to use the select button.
- Multiple tourist should be able to pick spots at the same time.
- Tourist should be warned if the place they select is closed or under maintenance.
- The module should be compatible enough to check if there are any other spots similar to what the tourist is looking for that has better reviews or is more famous than the current selection.

2. **Non-Functional**

- The system should be user friendly.
- The system should allow multiple users to access it at the same time.
- Error messages should pop whenever some issue is there.
- The module should auto save the spot picked by the tourist for them to access it later if required.

System and Data Models

1. System models

1. **System:** Displays the desired categories for the tourist to select from.
2. **Database:** It will update the database with the user's selection of the spots that he/she has selected and sends the confirmation to the tourist via the main server.
3. **Interface:** System.
4. **Interaction:** System can interact with the database.

2. User Interaction

1. **Registration Page:** New Users will have to fill out a general form to register. They must also choose a suitable Username and Password to login to the Home Page.
2. **Login Page:** Users have a username and password through which the User can login to the application. Administrative users have a different login username and password so as to differentiate the normal User from an Admin.
3. **Home Page:** All Users have a Home Page where there are quick action icons to access their account and their search history. There is also a log out button at all pages to redirect the User to the Login Page.
4. **Location Page:** Once the Users search for a particular place, the application is redirected to the page which shows the map of the place desired by the User and reviews of the place by fellow tourists.

3. General Flow of Information

1. Launch the application to get the Login Page.
2. Enter the User credentials and click login to enter the Home Page.
3. Home Page has multiple features with pre-defined tourist's location which the Users can select.
4. Once the User selects any place, the map of the place is shown along with reviews.
5. User can view any other page by going back to the Home Page or can logout with the click of the Logout button.
6. All these pages will be designed in JSP and/or JS. The User credentials and other necessary details would be stored in Serializable files.

4. Storage and Retrieval

1. Serializable files would be created and placed in any path desired by the Admin.
2. Every time a new user is registered from the Registration Page, the details are serialized automatically.
3. When the same User tries to login to the Home Page, this serialized data is retrieved and checked if the User exists or not. Appropriate error handling would also be created.

Architectural diagram

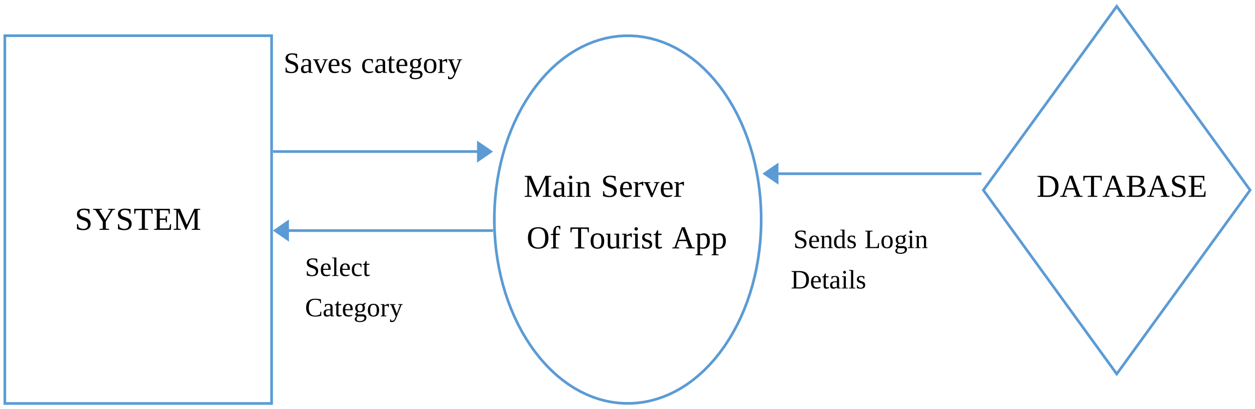


Figure 7: Architectural Diagram for Category Selection

DATA MODEL :

Flow of Information

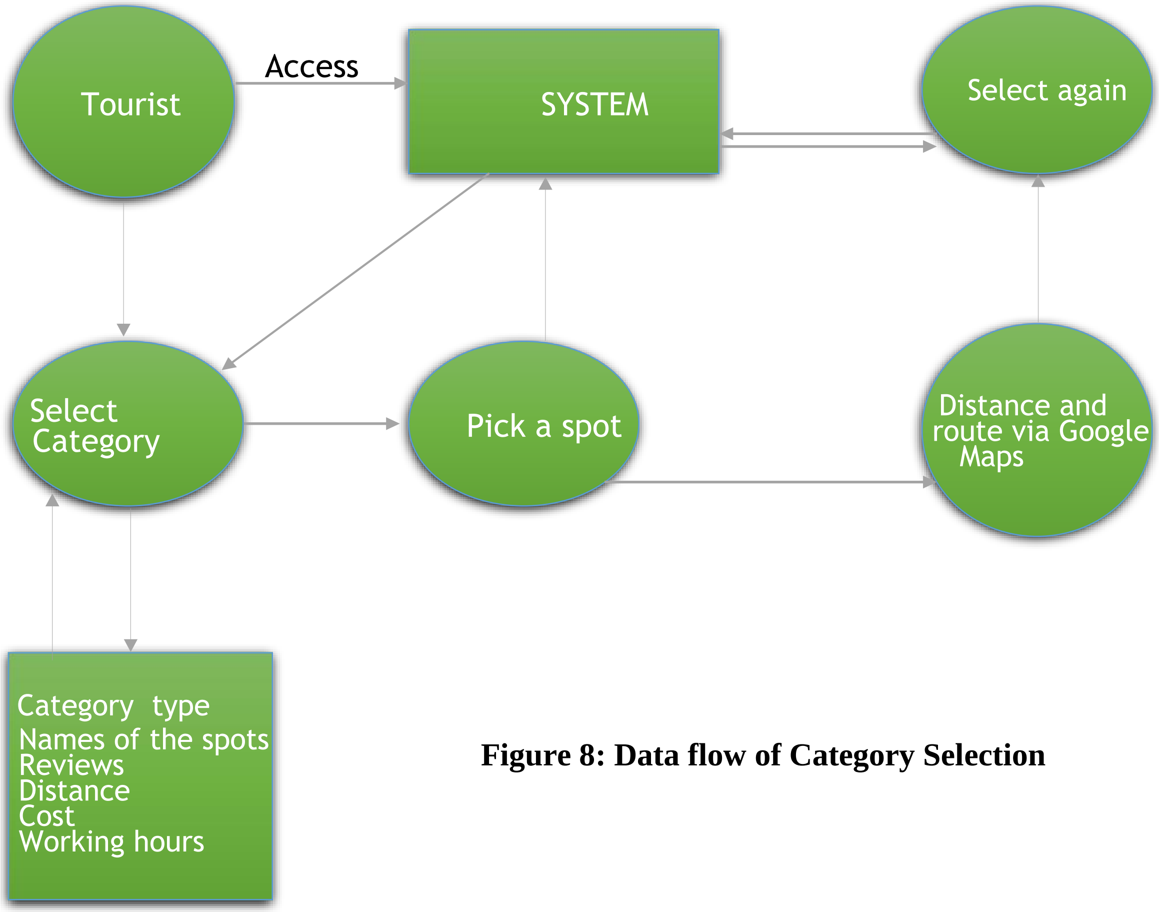


Figure 8: Data flow of Category Selection

Storage and Retrieval

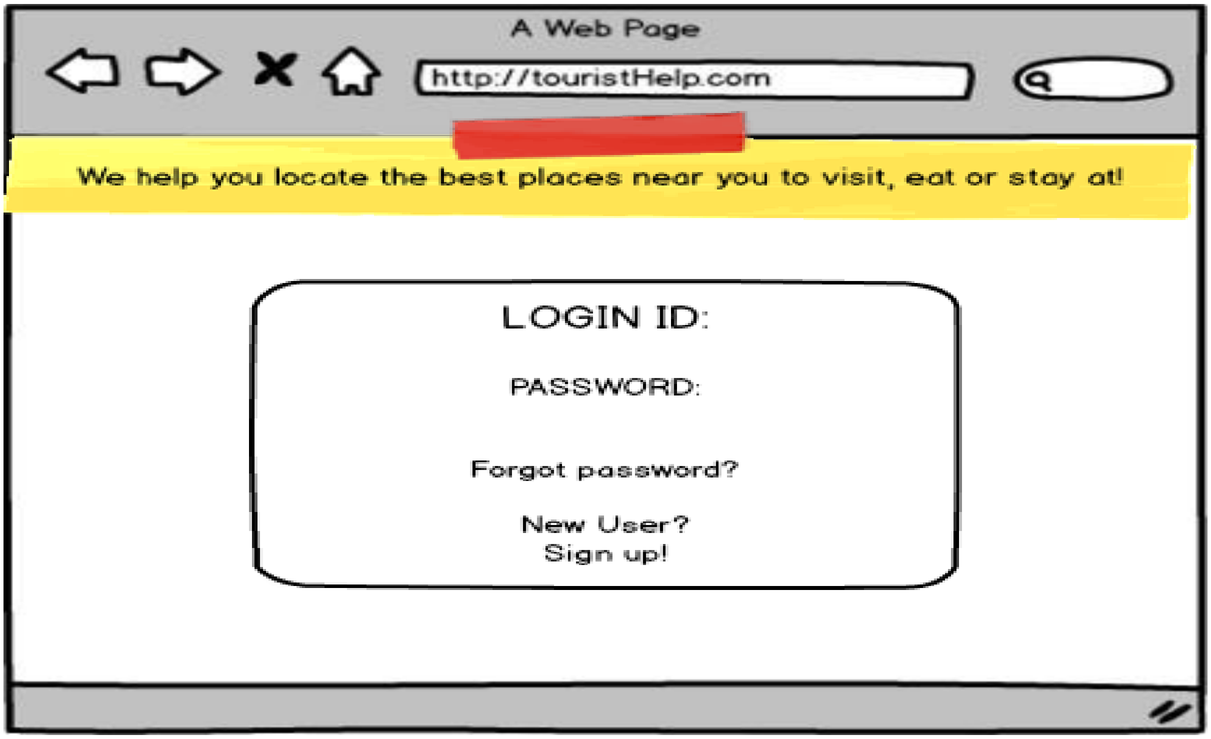
We use a database system to store all data for the Category Selection Page.
Following are the attributes stored per user category within the database:

USER CATEGORY	ATTRIBUTES STORED IN DATABASE
Tourist	Login ID Password Category selection Place selection
Admin	Login ID Password Approve category selection Approve place selection

DESIRED PROTOTYPE:

The following images show the different ways in which this tourist application which is a web based application works.

1. Home Page



2. Category selection page

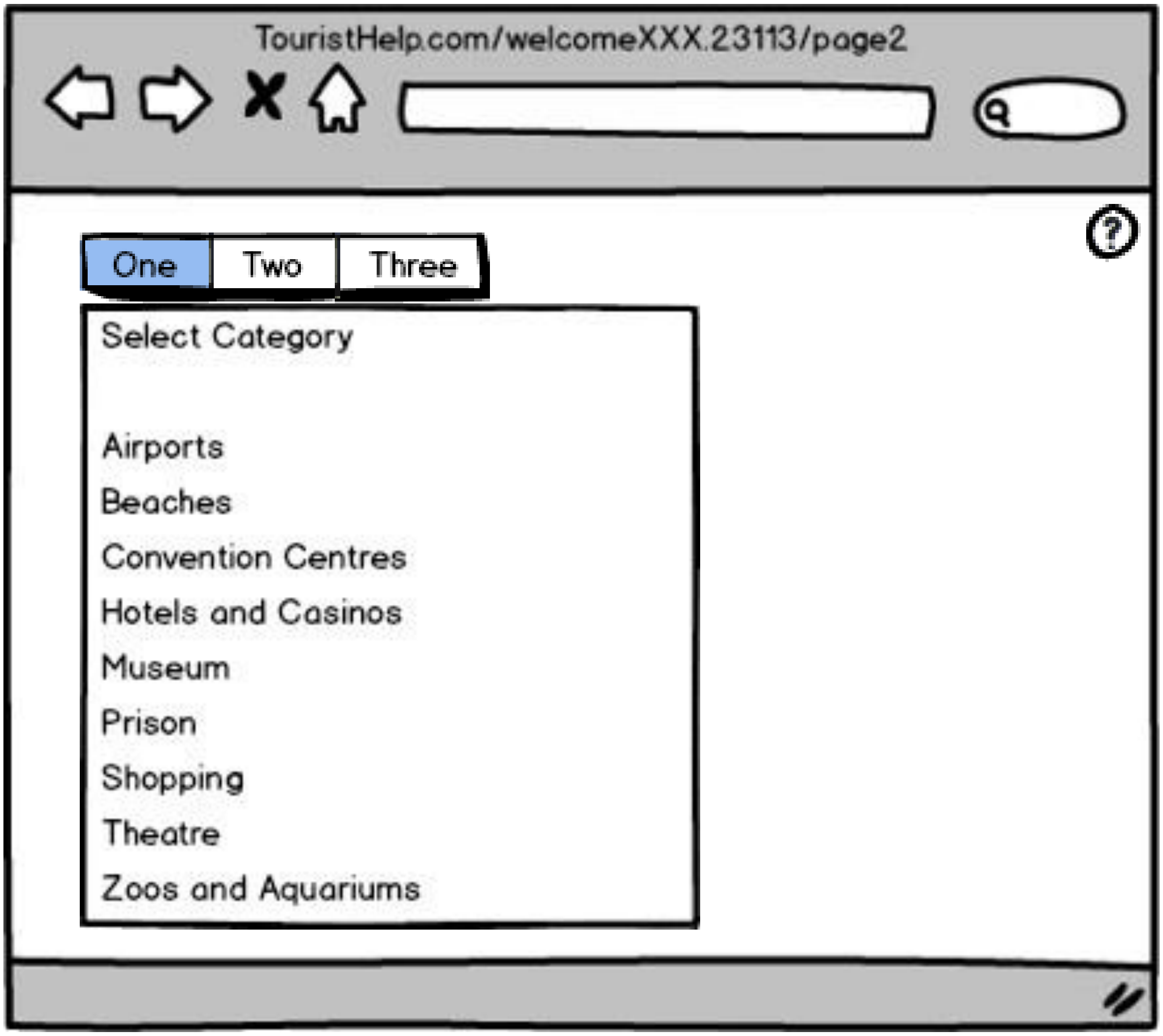


Figure 10: Selection Page

Test Cases:

Project Name: **Tourist Assist Application**	
Test Case	
Test case ID: TA001	**Test Designed by**: Chaitanya
Test Priority: Med	**Test Design Date**: 03/07/2016
Module Name: Login module	**Test Executed by:** Chaitanya
Test title: Verify user login details	**Test Execution Date:** 03/07/2016
Description: Check whether the username exists and if it exists verify the password	
Pre-Conditions: Username and password must exist in the database	

Step	**Test Steps**	**Test Date**	**Expected Results**	**Actual Results**	**Status**	**Notes**
1	Check whether the username exists in the database	03/07/2016	Username exists in the database	Username exists	Pass	
2	Check if the password matches with the username	03/07/2016	Password matches with username	Password does not match	Fail	

Project Name: **Tourist Assist Application**	
Test Case	
Test case ID: TA002	**Test Designed by**: Chaitanya
Test Priority: High	**Test Design Date**: 03/07/2016
Module Name: Category module	**Test Executed by:** Chaitanya
Test title: Selecting the visiting spot	**Test Execution Date:** 03/07/2016
Description: Tourist must be able to select a place to visit from a list of available categories	

Pre-Conditions: None						
Step	**Test Steps**	**Test Date**	**Expected Results**	**Actual Results**	**Status**	**Notes**
1	Select a category from list of categories	03/07/2016	Show places to visit in category	Places to visit in category	Pass	
2	Select place to visit in category	03/07/2016	Tourist must be able to use the select button	Tourist is able to use the select button	Pass	