UCS1611 - Internet Programming Lab
MINI PROJECT REPORT
TRAVELER ITINERARY MANAGEMENT SYSTEM

K Sowmya 205001110
Sowmya G 205001111
Subhashree M 205001113

1. Detailed problem statement

Planning a vacation is probably one of the most tedious tasks. A detailed plan for a journey is very essential for a smooth and hassle free trip that one can enjoy breaking away from their usual monotonous lives. By planning a good itinerary, one can manage their time, expenditures, prioritize what they want and ensure they don't forget the essentials.

By creating a one-stop website that caters to all the needs for planning an itinerary- from listing appropriate details of to and fro journey and accommodation to suggesting places to visit, we aim to help our customers plan an enjoyable trip.

We live in a busy world, where people toil everyday, working to make their ends meet. A well deserved vacation is much needed for people these days to take a break from their excruciating routine. In this vexed state of mind, it's difficult for someone to come up with a proper plan to have a relaxed and memorable trip. By assisting people who plan for a vacation, we can reduce their anxiety and stress and help them prepare themselves better for any situation.

Our Traveler Itinerary Management System aims to design and manage an itinerary for a traveler.
By getting the details of the traveler’s details (Name, Age, etc), destination and time and duration of stay, the system comes up with an efficient plan by providing insights of whatever that is required for travel.
Based on the specifications given by the user and the database of details of different cities and places that is fed in prior, the system gives details about the
-> Available transit (mode of transport, timings, cost etc)
-> Accommodation (alongwith facilities available)

-> Places to visit (categorized on the type - Historical, Cafes/Restaurants, Recreational, Adventure, Scenic, etc)
-> Provide an estimated budget of the expenses at different stages of the journey
-> Prepare a to-do list and keep track of list of items packed and bought

With the help of this system, we hope our users will be able to plan their trip meticulously and go with a free mind to enjoy their vacation.

2. Software requirement (functional, non-functional) specification for your project.

a. Users (Actors) and Characteristics

Travelers: Travelers are individuals who use the TIMS to book and manage their travel arrangements. They may include employees, clients, or any other individual who needs to travel for leisure or business. The travelers may have little to no knowledge of technical expertise.

Admin: Admins are the individuals who run and maintain the software. They are responsible for creating and managing travelers profiles and their information, recommending an itinerary and assisting the travelers etc. The admins must have technical expertise to run the TIMS.

b. Functional Requirements

User management

The system must create and manage user accounts.

Traveler management

The system must allow travelers to create and manage traveler profiles, including personal information and other relevant information.

Itinerary management

The system must allow admins to create and manage travel itineraries, including transit, hotels, restaurants and other travel arrangements.

Travel Details management

The system must allow travelers to view and edit the recommended travel details based on their preferences and view a detailed summary.

Security

The system must be secure and protect the privacy of traveler data and access control for user accounts.

Compatibility

The system must be compatible with different operating systems and web browsers.

c. **Sub-Functionalities**

Traveler Profile Management

The system should aid in creating and managing traveler profiles along with their personal information, contact details and other relevant information. Also provide access to authorized admin to view and edit profiles as needed.

Expense Management

The traveler can enter a budget and update each time a payment is made, which will then be compared with the budget to keep track of the expenses.

Checklist management

The traveler can have their own to-do list or checklist to keep track of items to buy and pack. They may also check the places they have visited.

d. Non-functional Requirements

Usability: The system must be easy to use and intuitive, with clear and concise user interfaces.

Performance: The system must be fast and responsive, with minimal delays or downtime.

Scalability: The system must be able to handle a large volume of users and data, and be able to scale up or down as needed.

Reliability: The system must be reliable and able to recover from system failures or errors.

Maintainability: The system must be easy to maintain and upgrade, with clear documentation and a well-structured codebase.

e. Operating Environment

The system will require the use of front-end web development to develop the user interfaces. The System must maintain a centralized repository of all travel related information for a destination to fetch the traveler's details and also make use of a database of all cities with popular tourist attractions to choose from.

f. Hardware Requirements

Any OS that supports high speed RAM (4GB or more) and large storage (100GB or more)

Web Browser such as Google Chrome , Mozilla Firefox or Microsoft Edge

g. Technologies

We will need to use HTML, CSS, JavaScript to build the TIMS and SQL server to maintain a database

h. Software Quality Attributes

Portability: The system should be portable in future, available for use anytime.

Scalability: The system must be able to handle a large volume of users and data, and be able to scale up or down as needed.

Availability: The system must be available 24*7

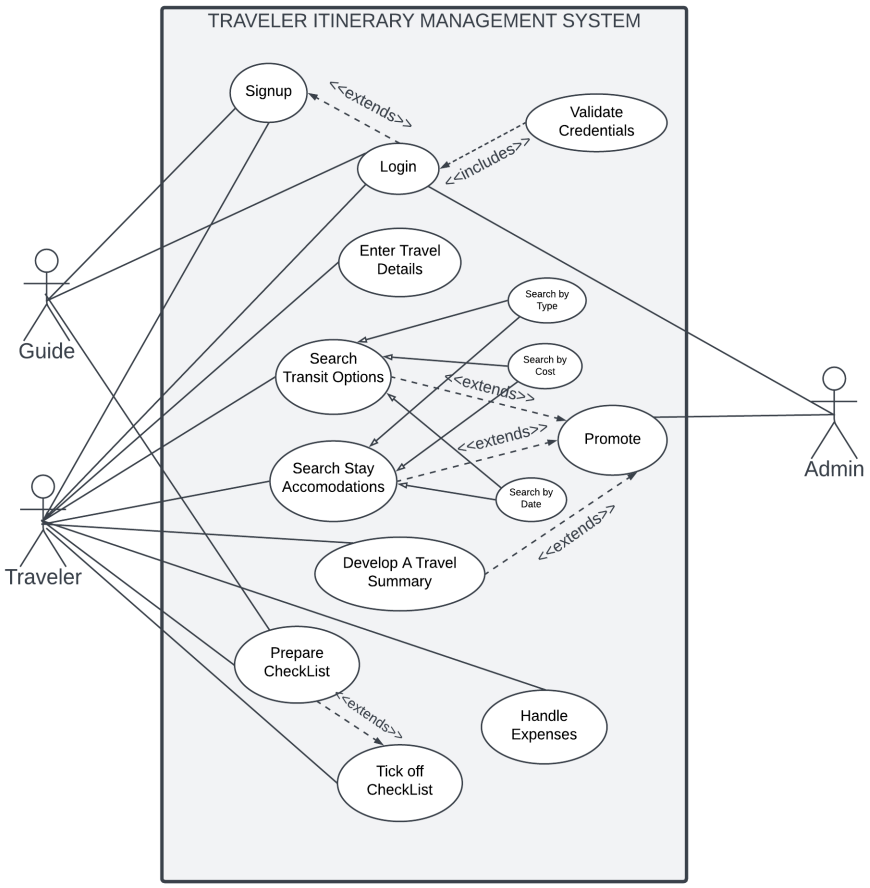
Reliability: The system must be reliable and able to recover from system failures or errors.

Maintainability: The system must be easy to maintain and upgrade, with clear documentation and a well-structured codebase.

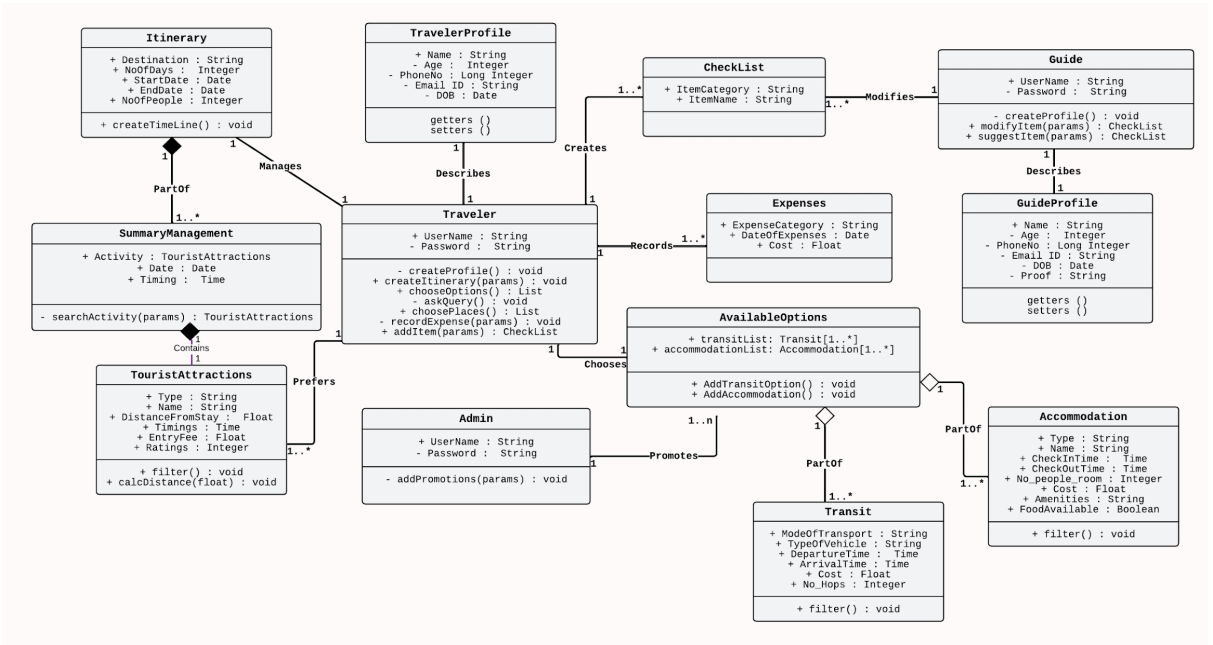
Standards Compliance: The system must be compliant with latest standards.

3. System Design

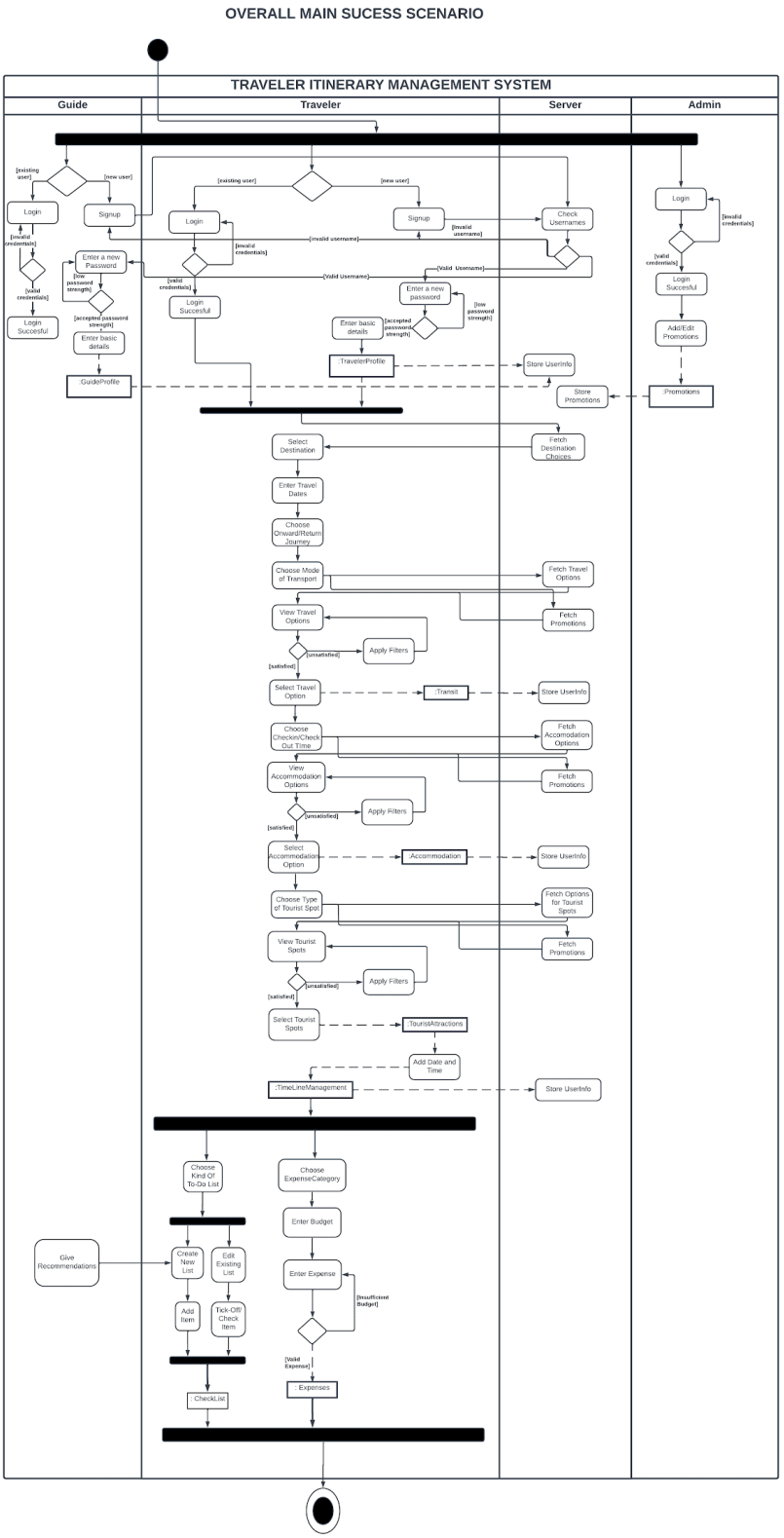
Use Case Diagram



Class Diagram



Activity Diagram



4. Functionality used for each module

1) Signup

Scope: Traveler Itinerary Management Application

Level: User Goal

Primary Actor: Traveler, Guide

Stakeholders: Traveler

Preconditions: Traveler/Guide is identified and Authenticated

Success Guarantee (Postconditions): User's Account is verified and Created

Main Success Scenario (Basic Flow):

Traveler/Guide will create a username. System validates the username.
Traveler/Guide will create a password. System will check for the password's strength.
Traveler/Guide will give basic information such as name, age, mobile number, email ID to finish setting up the account.

Extensions (Alternate Flows):

1. Invalid Username: User attempts to create an existing username. User is prompted for a new username, until distinct username chosen
2. Weak Password: User creates a password that does not satisfy the minimum requirements such as minimum length, no numerals, no uppercase, no special symbols
3. Mandatory Fields Empty: User does not fill mandatory information such as Mobile No etc while setting up the account.

In all the above scenarios, System will not let the user create an account.

2) Login

Scope: Traveler Itinerary Management Application

Level: User Goal

Primary Actor: Traveler, Guide, Admin

Stakeholders: Traveler, Guide

Preconditions: Traveler/Guide has an already existing account

Success Guarantee (Postconditions): User can avail the services provided by the website

Main Success Scenario (Basic Flow):

User can login to the website as a traveler or guide or admin
System will check and verify the username and password.
Traveler or guide or admin will be logged into the system through their account.

Extensions (Alternate Flows):

1. Username does not exist: User is prompted for a valid username.
2. Incorrect Password: User is prompted to enter the correct password.

In all the above scenarios, System will not let the user avail the services of the website

3) Enter Travel Details

Scope: Traveler Itinerary Management Application

Level: User Goal

Primary Actor: Traveler

Stakeholders: Traveler

Preconditions: Traveler has logged into the system

Success Guarantee (Postconditions): Traveler can start planning their itinerary and travel requirements

Main Success Scenario (Basic Flow):

Traveler must choose the destination place from the list of given cities
Traveler must give details of time (to and from place) and duration of stay
Traveler must also give information about number of fellow companions and their details

Extensions (Alternate Flows):

1. Unavailable City: Traveler searches a city not available in the system. System prompts traveler to choose from listed cities.
2. Wrong duration: Traveler provides return date that is lesser than start date. System prompts traveler to enter correct duration dates.

4) Search Transit Options

Scope: Traveler Itinerary Management Application

Level: User Goal

Primary Actor: Traveler

Stakeholders: Traveler, Travel Agencies

Preconditions: Traveler has logged onto the system

Success Guarantee (Postconditions): Traveler can book their transit arrangements with the corresponding transport services

Main Success Scenario (Basic Flow):

Traveler must choose either onwards or return journey
System will provide the list of available transit options
Traveler can apply the following filters:
- Mode of Transportation , timings, cost, no of hops etc
and choose from the filtered list based on their needs.

Extensions (Alternate Flows):

1. Traveler provides incorrect location for onward or return journey. System will prompt user to enter correct location.
2. Traveler is not satisfied with the list given by the system (even after filtering).

5) Search Stay Accommodations

Scope: Traveler Itinerary Management Application

Level: User Goal

Primary Actor: Traveler

Stakeholders: Traveler, Hotel Industry

Preconditions: Traveler has logged into the system

Success Guarantee (Postconditions): Traveler can book their stay accommodation with the required company

Main Success Scenario (Basic Flow):

System provides a list of stay accommodations like hotels, B&Bs, hostels, villas, etc.
Travelers can filter the options based on type of stay (resorts/ hotels/ hostels etc), cost, amenities, food provision, number of rooms, number of people per room and reviews along with check-in check-out time.
System will filter and generate a new list of options. Travelers can select from the list based on their needs.

Extensions (Alternate Flows):

System provides a list of stay accommodations, but traveler is not satisfied with given recommendations (even after filtering)

6) Develop a Timeline

Scope: Traveler Itinerary Management Application

Level: User Goal

Primary Actor: Traveler

Stakeholders: Traveler

Preconditions: Traveler has logged onto the system

Success Guarantee (Postconditions): Traveler's itinerary is planned and can be modified at any time

Main Success Scenario (Basic Flow):

System will give a list of available events and tourist spots.
Travelers can filter the list on the basis of type (historical, restaurants/cafes, scenic, recreational, adventurous etc) along with details of distance from place of stay, timings, entry fee, ratings etc.
Travelers can choose from the list based on their needs.
System will help the traveler recommend an itinerary based on the traveler's choice.

Extensions (Alternate Flows):

Traveler is not satisfied with the itinerary given by the system

7) Use Case 7: Prepare CheckList

Scope: Traveler Itinerary Management Application

Level: User Goal

Primary Actor: Traveler, Guide

Stakeholders: Traveler

Preconditions: Traveler has logged onto the system

Success Guarantee (Postconditions): Travelers can prepare themselves better for the vacation.

Main Success Scenario (Basic Flow):

Traveler chooses the kind of to-do list option: list of items to pack, list of souvenirs to buy, list of places to visit, restaurants/cafes to eat at or activities to do, etc.
Traveler can record items to a new list and even check mark/tick off activities done from the list.
Guides can also recommend new lists which may be beneficial to travelers.

8) Handle Expenses

Scope: Traveler Itinerary Management Application

Level: User Goal

Primary Actor: Traveler

Stakeholders: Traveler

Preconditions: Traveler has logged onto the system

Success Guarantee (Postconditions): Traveler can keep track of the expenses covered in the trip.

Main Success Scenario (Basic Flow):

Traveler enters a budget for the trip.
Traveler can then keep track of expenses at every level by making entries into the system.
Traveler can also modify their budget at any given time.

Extensions (Alternate Flows):

Traveler enters a very low irrational budget. System will prompt to enter a higher budget for basic accommodations.

9) Promote

Scope: Traveler Itinerary Management Application

Level: Sub Function

Primary Actor: Admin

Stakeholders: Traveler, Travel Agencies, Transport Services, Hotel Industry etc

Preconditions: Admin has logged onto the system

Success Guarantee (Postconditions): Admin can convey information regarding promotions which the affects the ordering of booking options for travelers

Main Success Scenario (Basic Flow):

Admins may advertise promotions or discounts available for transit arrangements, stay accommodations when the website is featured

5. Logic used for each module

i. Login Portal

The traveler or guide can sign up or login into the system. He/She can enter personal information with the traveler entering additional details regarding a new trip while the guide enters details regarding proof of being a guide.

The admin can only login to the system.

ii. Itinerary Portal

- Transit

  The System will provide transit recommendations filtered on the basis of mode of transport, timings, cost, no of hops etc. Travelers can select from the list for to and fro journey.

- Accommodation

  The System will provide a list of stay accommodations available filtered based on type of stay (resorts/ hotels/ hostels etc), cost, amenities, food provision, number of rooms and no of people per room along with check-in check-out time Travelers can select from the list based on their needs.

- Tourist Attractions

  System will give a list of available events and tourist spots filtered on the basis of its type (historical, restaurants/cafes, scenic, recreational, adventurous etc) along with details of distance from place of stay, timings, entry fee, ratings etc. Based on a timeline, traveler can choose places to visit from the list of recommendations on tourist spots, events and restaurants/cafes.

iii. Checklist Portal

Travelers can record items and essentials packed and bought for the trip with the help of the to-do list recommended by the system.

Travelers can also check the places they have covered during the vacation.

iv. **Budget and Expenses Portal**

Travelers can plan a budget for a category in the trip (like food) and record the expenses spent in that category. This helps the traveler check his expenses during the trip and the system also displays a warning when he/she exceeds the budget.

6. Usage of new tools for better functionalities

For our Traveler Itinerary Management System, we used HTML, CSS and JavaScript for the frontend.
For the backend, we made extensive use of Java Servlets to couple the front end and database. We employed servlets to fetch, insert, update and delete entries from the table.
We used Ajax to improve the front end of the website and display the modified details in the page. We used MySQL for our database.

7. Features or techniques that were not used in your project and justify why it was not used. Suggest how it can be used in the future work

We haven't implemented Google API as it was not compatible with the version of JAVA that was used to run the servlets. It can be used in the future work, to display and pin the selected tourist spots or places in a map-view

8. Best Practices Used

1. Naming convention of html, css and servlet files (java) are intuitive and self explanatory
2. Usage of indentation to make the code readable
3. Usage of external css to style common elements across different pages
4. Usage of script files to define common functions used across different pages
5. Usage of ajax to display updated details in the same web page
6. Normalizing the relations used in our database to avoid conflicts