DESIGN OF A TRAVEL ITENERARY PLANNING AND RESERVATION SYSTEM

By

Yashkumar Paneliya (22CS60R70)

Shailesh Chaudhary (22CS60R37)

Submitted to

Prof. Pallab Dasgupta

Prof. Sourangshu Bhattacharya



COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

**Introduction:**

* Travel itinerary and reservation system is a portal or application where a user can provide several places that he/she want to visit.
* In addition to that several constraints like arrival date at a particular place, number of days to stay at a particular place, mode of transportation, etc.
* On the basis of these inputs, the system generates an itinerary plan along with status of tickets for suggested travel mode between consecutive places.
* Once the plan is generated, user is given an option to confirm their booking and after successful payment, the bookings are done and stored in user’s profile.

**Features:**

* Authentication and registration
* Itinerary planning
* Payment

**Use cases:**

* Use Cases describe the sequence of events that an actor take place using the system in order to complete a process.
* Use case identifies the actors and the processes they perform.
* In the following table all the use cases that may be included in this project is described.

|  |  |
| --- | --- |
| **Use case** | **Actor** |
| Registration | Traveller (User) |
| Login |  |
| Input places |  |
| Itinerary generation |  |
| Confirm bookings |  |
| View bookings |  |
| Payment |  |

**Use case: Registration**

* Actor: Traveller
* Description:
  + Traveller first has to register to use the feature of itinerary planning in the system. Registration requires email, name and a password from the user.
  + An HTTP request is then sent to server with all these data and server checks whether this user already exists or not, if no then registers the user and store the data into database and user is redirected to the home page of system.

**Use case: Login**

* Actor: Traveller
* Description:
  + If user has already registered to the system, then he/she can directly login to the system.
  + Login requires email and password
  + An HTTP request is sent to server for verification of credentials, if verified then user is redirected to home page and its username is stored in the local storage of the end-user machine for further interaction with server.

**Use case: Input places**

* Actor: Traveller
* Description:
  + User has choices to input three different places, their arrival dates, number of days to stay and mode of transportation.
  + Currently the choice of places is fixed.
  + Transportation modes can be flight, bus or train.
  + Several constraints are also imposed, places and dates should be different, difference between dates should not be smaller than number of days of stay between two places, etc.

**Use case: Itinerary Planning**

* Actor: Traveller
* Description:
  + Once the input is provided, the data is sent to server as a POST request.
  + Server processes the data and generates the itinerary plan along with ticket costs, status of availability of tickets and order of places to visit.
  + Whole plan is sent to client and an option to confirm booking is provided.

**Use case: Confirm booking**

* Actor: Traveller
* Description:
  + An HTTP request is sent to server to confirm the booking.
  + Server then generates a unique ID for the plan and stores it into user’s data and also the payment data is stored in separate data file of admin analysis.
  + After successful confirmation, user can view their bookings on system.

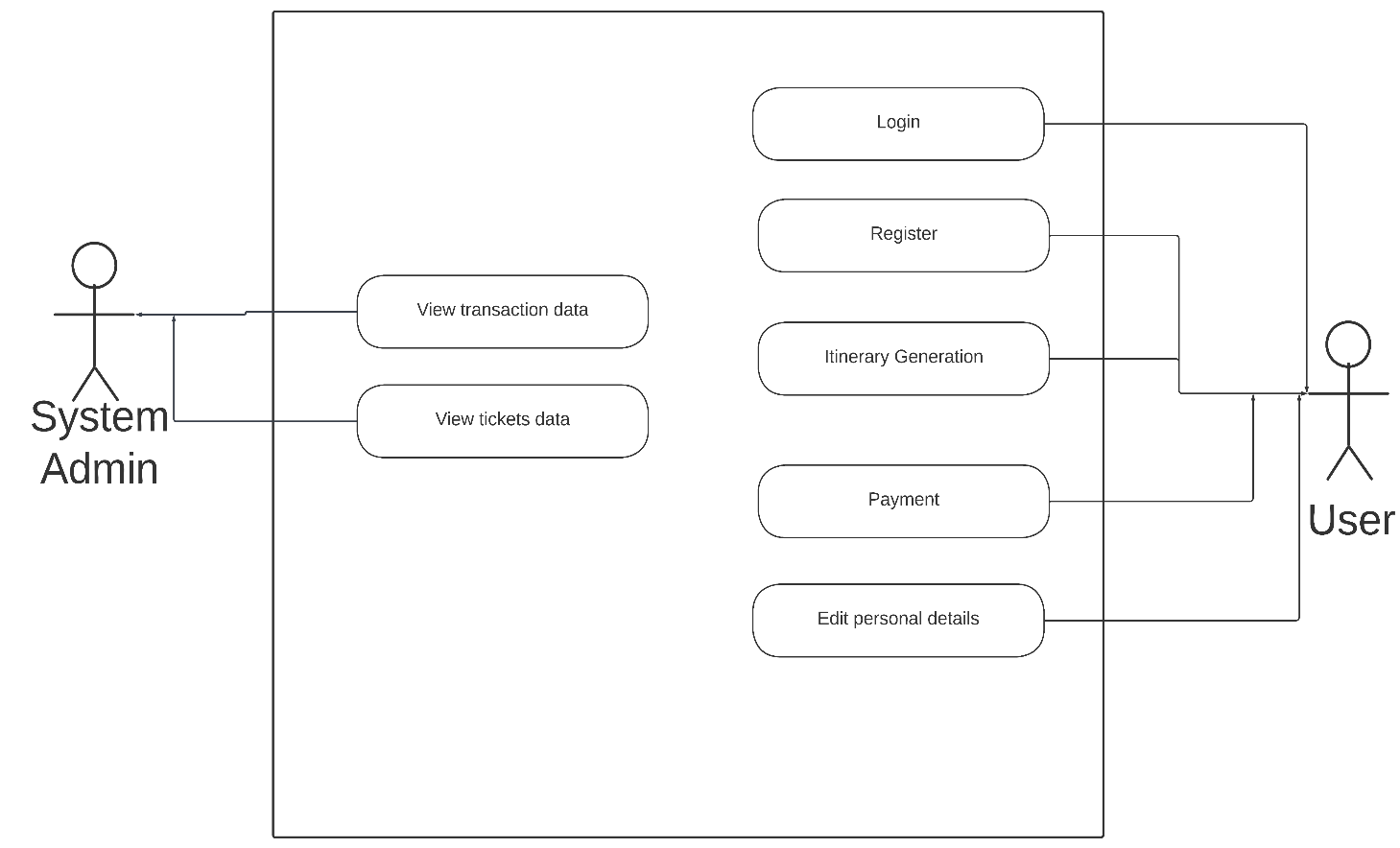
**Use case: Payment**

* Actor: Traveller
* Description:
  + Before confirming the bookings, user has to pay the total cost of tickets.
  + On successful payment, bookings get confirmed and database entries are modified (for tickets).

**Use case: View booking**

* Actor: Traveller
* Description:
  + A user can view his/her all bookings in the profile section in system.
  + Bookings are stored in separate files for each user and fetched according to logged in user.

**Diagram:**



**Languages used:**

**Frontend:** HTML, Javascript

**Style:** CSS

**Backend:** Python

**External libraries used:**

* Pandas (For CRUD operations on datasheets)