TRAVEL MANAGEMENT SYSTEM

By – Agrima Shukla (Batch 3)

Project overview:

This Travel Management project is a console project which is built to optimize the booking facility offered to customers. It offers functionalities to simplify booking of the packages offered by the site. Various functionalities are implemented to provide user-friendly access. This project is built in python and sqlite.

Technology Used:

- Python for Programming language.
- SQLite for database
- Logging for maintaining log file for errors.
- Shortuuid for unique ID.
- Datetime for storing dates.
- Hashlib for hashing password.
- Maskpass for masking password.
- Regex for input validation.
- Tabulate for pretty console.
- Visual Studio code for IDE.
- DrawIO for diagrams.

Modules:

1) Admin module:
    - One admin for the system
    - Authenticated when providing username and password.
    - It is responsible for handling package and itinerary module.
    - Perform many operations like adding and updating the modules.
2) Customer module
    - Multiple customers for the system.

- Register themselves.
- Authenticated when providing username and password.
- They can see packages offered by the system by providing preferences.
- Book packages after viewing packages.
- Cancel packages booked in the past.

3) Package module
    - Package details added by admin.
    - Viewed by user.
    - Input validation using regex when added by admin.
    - Can be updated by admin.
4) Itinerary module
    - Itinerary details added by admin.
    - Viewed by user.
    - Input validation using regex when added by admin.
    - Can be updated by admin.
5) Booking module
    - Customer can book packages by providing personal information and information needed for booking.
    - Booking can be viewed by customer.
    - Can be cancelled by customer.

Functional Requirements:

1) ADMIN
    I. Package module
        - Add package by entering details. Applied input validation for admin input.
        - Activate and deactivate packages by entering package id.
        - View all the packages.
        - Update package – can update multiple fields (package name, duration, limit, price and category)

    II. Itinerary module

- Add itinerary by entering details. Applied input validation for admin input.
- Show itinerary by entering itinerary id.
- Update itinerary – can update multiple fields (city, day and description)

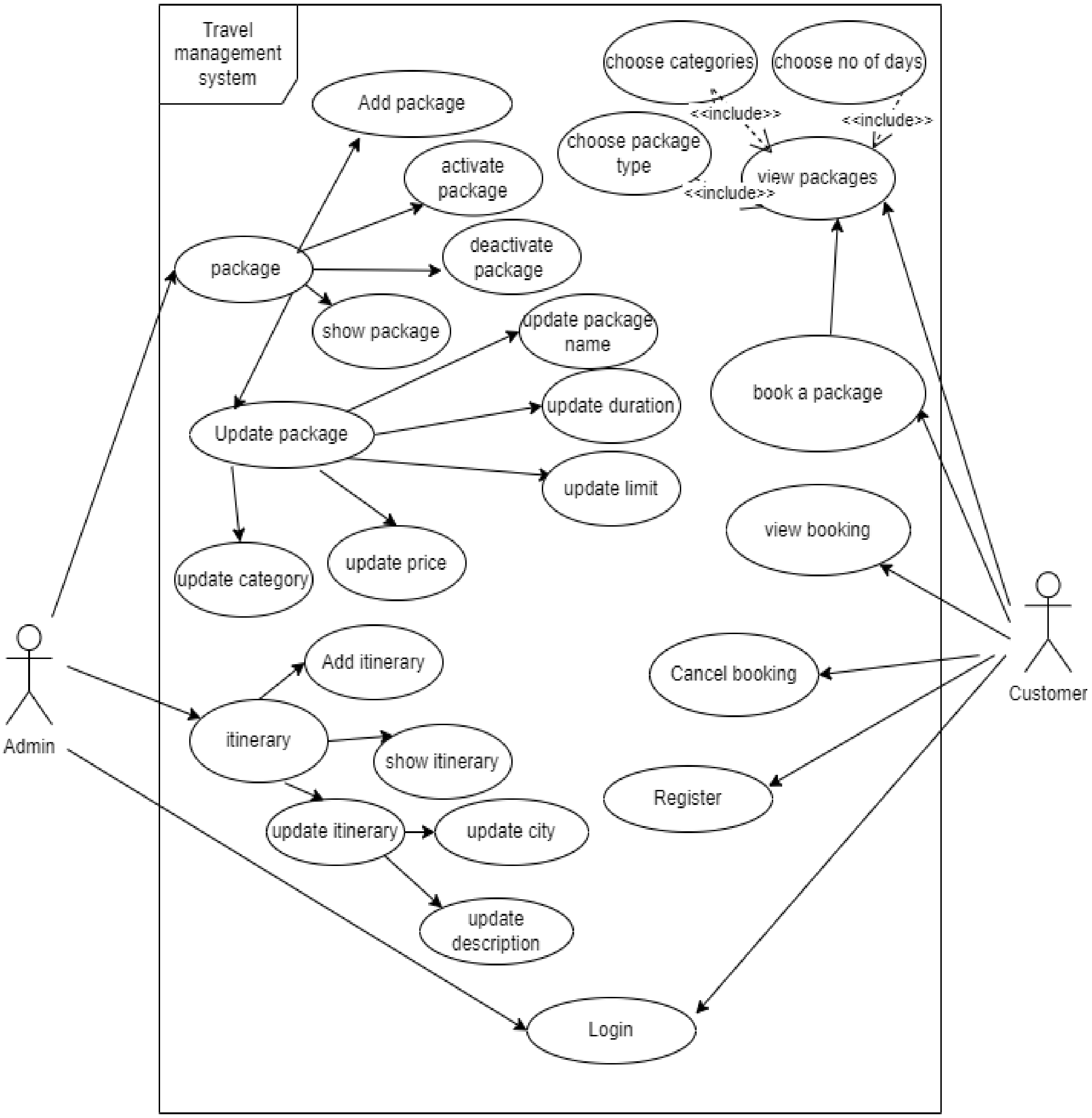
2) CUSTOMER
   I. View packages by choosing details (categories, no of days and package type)
   II. Book packages by entering details. Applied input validation for user input.
   III. View bookings.
   IV. Cancel bookings.

Nonfunctional requirements:

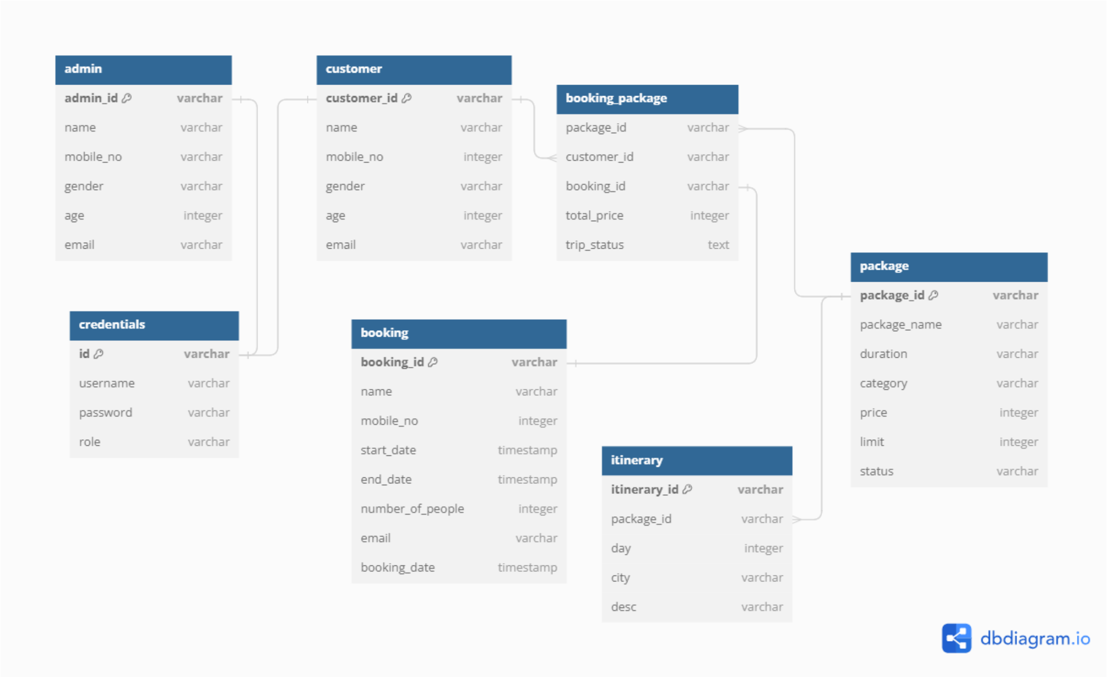
1. Security
   - Added hashing for password to maintain security.
   - Used data as parameters for query to avoid SQL injection attack.
2. Authentication
   - It is maintained by user credentials username and hashed password.
3. Validation
   - Validated every user input.
4. Logging
   - Added logging wherever necessary to log the information or errors.

Diagrams:

1) USE CASE



2) DATABASE DIAGRAM



admin	
admin_id	**varchar**
name	varchar
mobile_no	varchar
gender	varchar
age	integer
email	varchar

customer	
customer_id	**varchar**
name	varchar
mobile_no	integer
gender	varchar
age	integer
email	varchar

booking_package	
package_id	varchar
customer_id	varchar
booking_id	varchar
total_price	integer
trip_status	text

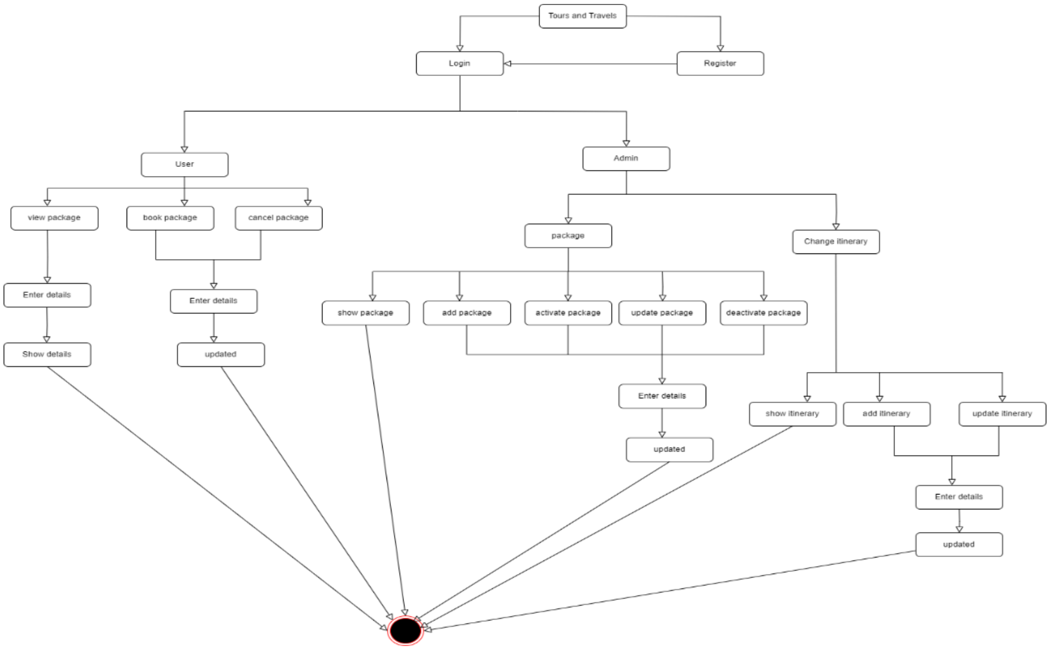
package	
package_id	**varchar**
package_name	varchar
duration	varchar
category	varchar
price	integer
limit	integer
status	varchar

credentials	
id	**varchar**
username	varchar
password	varchar
role	varchar

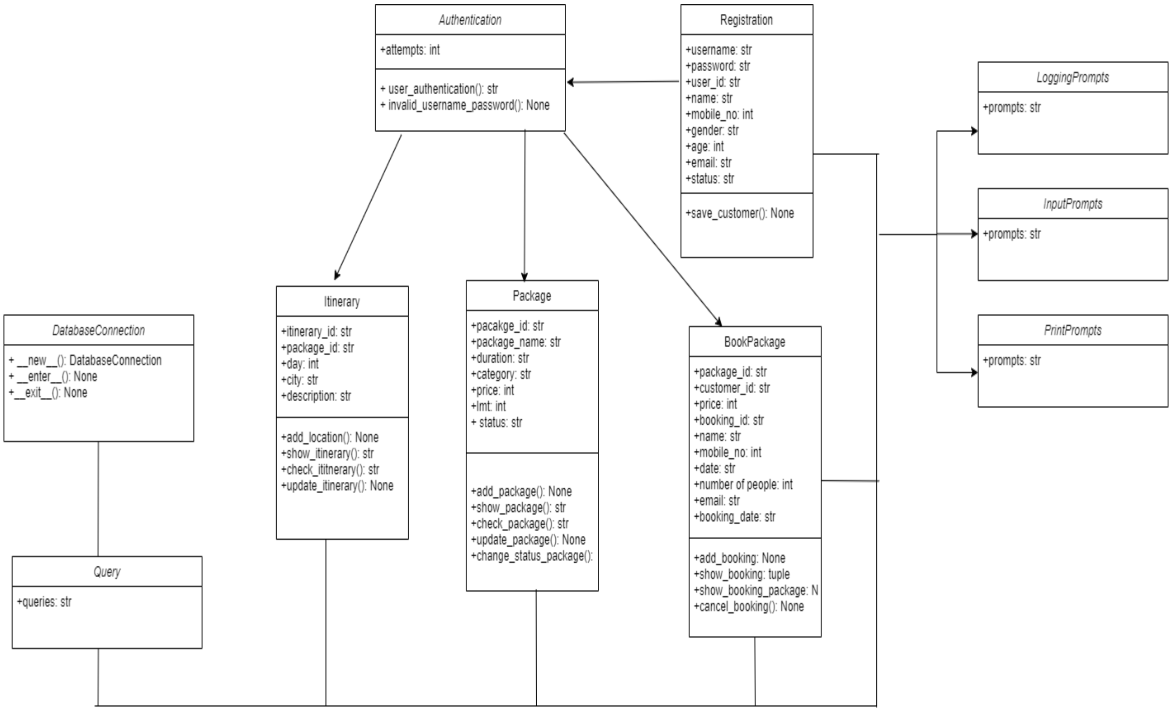
booking	
booking_id	**varchar**
name	varchar
mobile_no	integer
start_date	timestamp
end_date	timestamp
number_of_people	integer
email	varchar
booking_date	timestamp

itinerary	
itinerary_id	**varchar**
package_id	varchar
day	integer
city	varchar
desc	varchar

3) FLOWCHART



4) CLASS DIAGRAM


Authentication
+attempts: int
+ user_authentication(): str
+ invalid_username_password(): None
Registration
+username: str
+password: str
+user_id: str
+name: str
+mobile_no: int
+gender: str
+age: int
+email: str
+status: str
+save_customer(): None
LoggingPrompts
+prompts: str
InputPrompts
+prompts: str
PrintPrompts
+prompts: str
DatabaseConnection
+ __new__(): DatabaseConnection
+ __enter__(): None
+__exit__(): None
Itinerary
+itinerary_id: str
+package_id: str
+day: int
+city: str
+description: str
+add_location(): None
+show_itinerary(): str
+check_ititnerary(): str
+update_itinerary(): None
Package
+pacakge_id: str
+package_name: str
+duration: str
+category: str
+price: int
+lmt: int
+ status: str
+add_package(): None
+show_package(): str
+check_package(): str
+update_package(): None
+change_status_package():
BookPackage
+package_id: str
+customer_id: str
+price: int
+booking_id: str
+name: str
+mobile_no: int
+date: str
+number of people: int
+email: str
+booking_date: str
+add_booking: None
+show_booking: tuple
+show_booking_package: N
+cancel_booking(): None
Query
+queries: str