

# MAD 1 PROJECT REPORT

## STUDENTS DETAILS:-

NAME : YUVRAJ GOSAIN

ROLLNO. 22F3001690

PROGRAM: BS IN DATA SCIENCE & APPLICATION

## PROJECT QUESTION STATEMENT:-

HOUSEHOLD SERVICES APPLICATION

### How I approached the problem statement;

I started by creating an ER diagram, which laid a solid foundation for my project and made the development process smooth and organized. Gradually, I implemented all the required functionalities as specified for the application

- **Frameworks and libraries used;**

- **Standard Python Libraries:**

**1) datetime** - For handling date and time-related operations.

**2) enum** - For defining enumerations (custom data types with named values).

**3) os** - For interacting with the operating system (e.g., file handling, paths).

**4) re** - For working with regular expressions.

**5) Flask Libraries:**

**6. flask:**

- **Flask** - The main framework for building web applications.
- **flash** - For sending one-time messages to the user (e.g., success or error notifications).
- **redirect** - For redirecting users to a different URL.
- **request** - For handling incoming requests (e.g., form data, query parameters).
- **send\_from\_directory** - For serving files from a directory.

- **url\_for** - For building URLs dynamically.
- **render\_template** - For rendering HTML templates.
- **session** - For handling session data.

## **7.flask\_sqlalchemy:**

- **SQLAlchemy** - An ORM (Object-Relational Mapper) for interacting with databases.

## **8.flask\_migrate:**

- **Migrate** - For database migrations (schema changes)

## **9.flask\_login:**

- **LoginManager** - For managing user authentication.
- **UserMixin** - A helper class for managing user sessions.

- **current\_user** - To access the currently logged-in user.
- **login\_required** - A decorator to restrict access to authenticated users.
- **login\_user** - To log in a user programmatically.
- **logout\_user** - To log out the current user.

- **SQLAlchemy Libraries:**

## **10.sqlalchemy:**

- **Column, ForeignKey, Integer** - For defining database table schemas.
- **or\_, func** - For performing complex queries and using SQL functions.
- **exc.SQLAlchemyError** - For handling SQLAlchemy-related exceptions.

- **Werkzeug Libraries:**

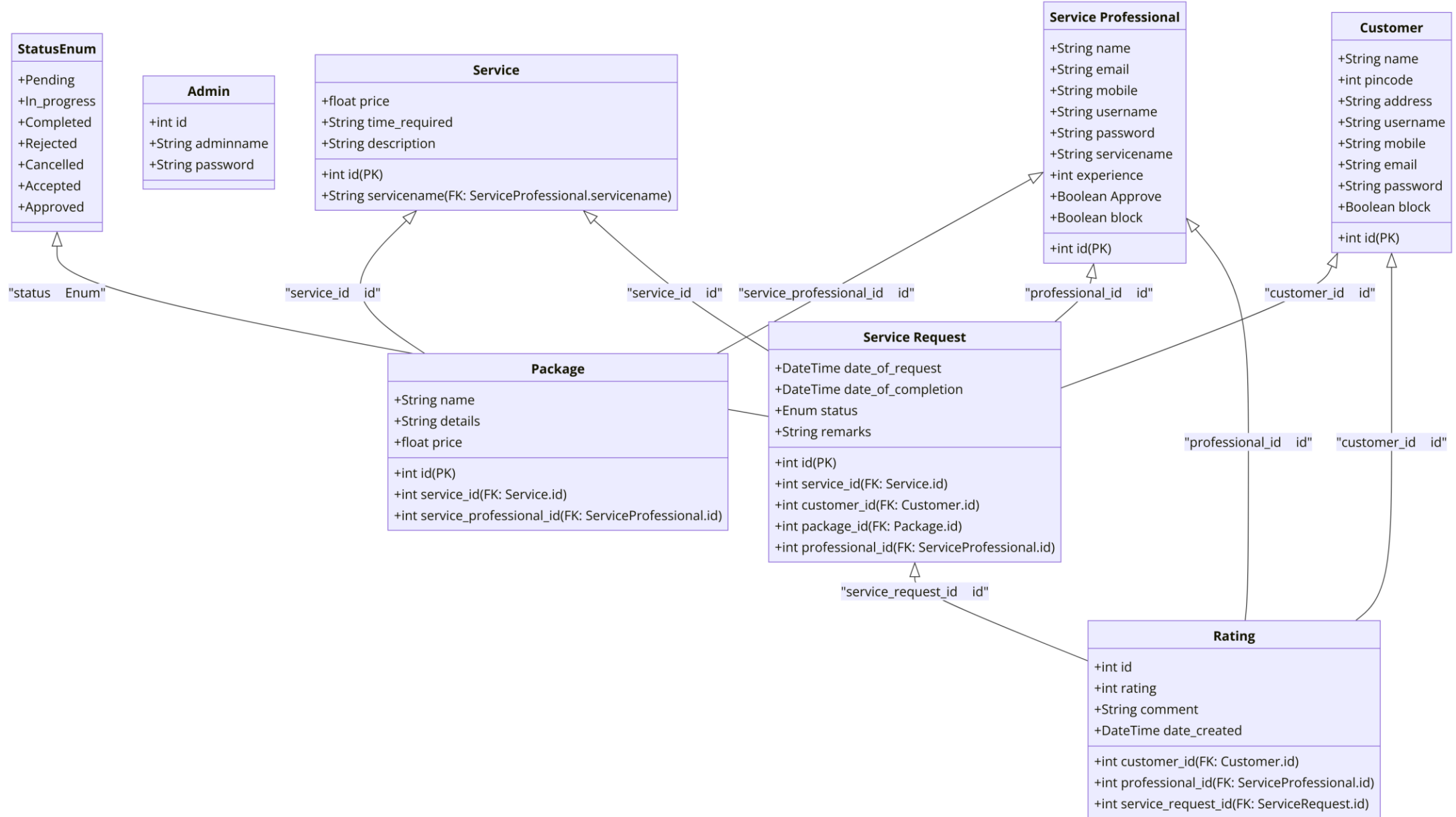
## **11.werkzeug.security:**

- **check\_password\_hash, generate\_password\_hash** - For securely hashing and verifying passwords.

## **12.werkzeug.utils:**

- **secure\_filename** - For safely handling uploaded file names.

# ER DIAGRAM



# APPLICATION WORKFLOW

