**Software Engineering Project**

**Phase III: Software design and modeling**

***Group Members: Favio Kushta, Livia Kadisi, Eriselda Deda, Arjola Sinaj, Viktoria Peshku***

Overview:

In this phase, we will design the software architecture and model the system to ensure that it meets the user requirements and application specifications outlined in the previous phase. The design and modeling will be documented and presented for review and feedback.

Tasks:

1. Software Architecture Design:

We will design the software architecture for the bus reservation app using UML diagram such as use case diagram. The architecture will define the components of the system, their interactions, and how they are organized.

The Bus Reservation App will have a three-tier architecture consisting of a presentation layer, business logic layer, and data storage layer. Each layer is responsible for a specific set of tasks and communicates with the other layers to achieve the overall functionality of the application.

Presentation Layer:

The presentation layer is the user interface of the Bus Reservation App. It is responsible for handling user interactions and displaying information to the user. This layer includes the mobile application, web application, or any other platform used by the user to access the system.

Business Logic Layer:

The business logic layer handles the core functionality of the Bus Reservation App. This layer is responsible for processing user requests, managing data, and coordinating the activities of the system components. It includes the following components:

User Interface Manager: The User Interface Manager is responsible for managing user input and output, and processing user requests.

Route and Schedule Manager: The Route and Schedule Manager is responsible for managing the bus routes and schedules. It receives and processes user requests related to the selection of bus routes and seats.

Payment Manager: The Payment Manager is responsible for handling payment transactions related to bus seat reservations.

Reservation Manager: The Reservation Manager is responsible for managing user reservations, processing cancellations, and updating the seat availability.

Data Storage Layer:

The data storage layer is responsible for storing and managing the data used by the Bus Reservation App. It includes a database or any other form of data storage that is used to store information such as user profiles, bus schedules, and seat availability.

The following diagram illustrates the interactions between the different components of the Bus Reservation App:

+----------------------------------------------------+

| Presentation Layer |

+----------------------------------------------------+

|

|

+--------------------------------------+

| Business Logic Layer |

+--------------------------------------+

|

|

+-------------------+ +-------------------------+

| Data Storage Layer| | External Payment Gateway |

+-------------------+ +-------------------------+

In summary, the presentation layer communicates with the business logic layer to process user requests and display information to the user. The business logic layer communicates with the data storage layer to retrieve and update data, and with the external payment gateway to process payment transactions.

Bus Reservation App

Use Cases

+------------+

| View Routes |

+------------+

|

|

v

+---------------------------+

| Select Route and Bus Seats |

+---------------------------+

|

|

v

+---------------------------+

| Pay for Bus Seat Reservation|

+---------------------------+

|

|

v

+---------------------------+

| Cancel Bus Seat Reservation|

+---------------------------+

Explanation:

The use case diagram shows the different actions that the user can perform within the bus reservation app.

The app has four main use cases:

View Routes: The user can view the available bus routes.

Select Route and Bus Seats: The user can select a route and the number of seats they want to reserve.

Pay for Bus Seat Reservation: The user can pay for the reserved seats.

Cancel Bus Seat Reservation: The user can cancel the reserved seats.

Each use case represents a specific task that the user can perform within the app.

The arrows between the use cases show the flow of the user's interaction with the app.

1. User Interface Design:

We will design the user interface of the app using wireframes, prototypes, and mockups. This will help us to visualize the user experience and ensure that the app is user-friendly and easy to navigate.

Landing Page: The landing page should be simple and intuitive, providing users with a clear idea of what the app is for and how to use it. It should include a search bar for finding available buses and a list of popular destinations.

Bus Schedule Page: This page should display a list of available buses, their schedules, and seat availability. Users should be able to filter and sort the results by various criteria, such as departure time, arrival time, and price.

Reservation Page: This page should allow users to select a bus and reserve a seat. Users should be able to select their preferred seat and choose whether they want a one-way or round-trip ticket. The page should also display the total cost of the reservation and allow users to make payments.

Reservation Management Page: This page should allow users to view and manage their reservations. Users should be able to view their reservation details, cancel their reservation, and view their payment history.

Account Settings Page: This page should allow users to manage their account settings, including their personal information, payment methods, and notification preferences.

Feedback Page: This page should allow users to provide feedback and suggestions about the app. Users should be able to submit their feedback and view responses from the app developers.

1. Database Design:

We will design the database schema for the app using entity-relationship diagrams. This will define the data models, relationships, and constraints for the system.

+---------------+

| USER |

+---------------+

| -user\_id |

| -username |

| -password |

| -email |

| -phone |

+---------------+

|

|

+---------------+

| RESERVATION |

+---------------+

| -reservation\_id |

| -user\_id |

| -bus\_id |

| -date |

| -status |

+---------------+

|

|

+---------------+

| BUS |

+---------------+

| -bus\_id |

| -bus\_number |

| -driver |

| -route |

| -capacity |

+---------------+

The database schema consists of three main entities: USER, RESERVATION, and BUS. The USER entity stores information about the users of the app, including their user ID, username, password, email, and phone number. The RESERVATION entity stores information about each reservation made by the users, including the reservation ID, user ID, bus ID, date, and status. The BUS entity stores information about each bus in the system, including the bus ID, bus number, driver name, route, and seating capacity.

The USER entity is connected to the RESERVATION entity through the user ID attribute, which is a foreign key in the RESERVATION entity. This indicates that each reservation is made by a specific user. Similarly, the BUS entity is connected to the RESERVATION entity through the bus ID attribute, which is a foreign key in the RESERVATION entity. This indicates that each reservation is made for a specific bus.

1. Documentation:

We will document the design and modeling process, including UML diagrams, wireframes, prototypes, and entity-relationship diagrams. We will also include explanations and justifications for the design decisions made on the final version.

1. Review and Feedback:

The design and modeling documentation will be reviewed by the project team and any feedback will be incorporated into the final design. The final design will then be presented to the client for approval before proceeding to the next phase of the project.

As students working on a university project, it is important to keep in mind that the project should be feasible given the time and resources available.