Software Design Specification

for

Online Store-House Booking System

Prepared by

Al Musabbir - 1611158042

Md. Rafat Rahman Tushar - 1621450042

Raisa Mehjabin Azni - 1620216042

Monjur A-Elahi Tanmoy – 1620587042

Course: CSE327

Sec: 06

North South University

Summer 2019

Table of Contents

1. Purpose

2. Architectural Design

3. Architectural Pattern

4. ER Diagram

5. Database Schema

6. Use Case Diagram

7. Class Diagram

8. Sequence Diagrams

**Purpose**

Nowadays, the small businesses and start-ups are growing at a large scale. But the capacity, space is a major concern in this kind of businesses. They need to store their necessary products somewhere else. But going to store by store without knowing the capacity and other information regarding the store-house is a problem for the businessman. Moreover, many store owners can’t get enough customers by traditional methods. Our purpose is to create an app that can solve this problem by connecting the store owners and the businessmen. Basically, we are building this application focusing all kind of criteria regarding this scenario. We are also building this application on the basis of users’ requirements.

**Architectural design**

The software basically has 2 components. The frontend UI for both admins and end users, and the database that only admins will see and interact with.

**Architectural pattern**

The application will be developed in the MVC architectural style. There are three main components to it

- Model component, which manages the data

- View component, which determines how the data is presented

- Controller component, that manages the both model and view components.

The purpose of using MVC is to separate the user interface from the logic so that it is easier to handle and this makes it more flexible to alter data or its representation. Hence being an easy option for applications.

ER Diagram

User\_shipper

Sh\_guy\_vehicle\_id:varchar Sh\_car:varchar Vehicle\_status:varchar Fee\_per\_kilo:double Vehicle\_rating:double Vehicle\_owner\_rating:double Sh\_policies\_&\_licenses:varchar

Shipper

User\_bus

Bus\_id:varchar Bus\_name:varchar Bus\_owner:varchar Bus\_products:varchar Bus\_house\_no:int Bus\_road\_no:int Bus\_area/vill:varchar Bus\_po:varchar Bus\_pcode:int Bus\_thana:varchar Bus\_district:varchar Bus\_shipping\_vehicle:varchar Bus\_policies\_&\_licenses:varchar

Business

User\_store

Store\_id:varchar Store\_name:varchar Store\_owner:varchar Material\_types:varchar Capacity:int Available\_space:int Store\_house\_no:int Store\_Road\_no:int Store\_area/vill:varchar Store\_po:varchar Store\_pcode:int Store\_thana:varchar Store\_district:varchar Store\_shipping\_vehicle:varchar Fee\_per\_kilo\_shipping:double Fee\_per\_day\_per\_100\_product\_storing:double Subscription\_status:varchar Subscription\_expiry:DateTime Store\_rating:double Store\_owner\_rating:double Policies\_&\_licenses:varchar

Store

User

User\_id:varchar Password:varchar Email:varchar Phn:int Name:varchar Type:varchar

Admin\_id:varchar Admin\_pword:varchar

Admin

Shiper\_order

Shipping\_order

Sh\_or\_no:int Sh\_guy\_vehicle\_id:varchar Order\_no:varchar From\_whr:varchar To\_whr:varchar Departure:DateTime Arrive:DateTime Estimated\_arrival\_time:DateTime Reached?:varchar Paid?:varchar

Order\_ship

Order

Order\_no:int Store\_id:varchar Bus\_id:varchar Product\_type:varchar Product\_size/weight:varchar From\_when:DateTime To\_when:DateTime Quantity:double Sh\_from\_whr:varchar Sh\_to\_whr:varchar

Pay\_order

Str\_Bu\_Or

Stored\_item

Payment\_account

Store\_item\_id:varchar Bus\_id:varchar Store\_id:varchar Order\_no:int Product\_name:varchar Product\_size:varchar From\_date:DateTime To\_Date:DateTime Product\_status:varchar

Str\_itm\_order

Store\_id:varchar Bus\_id:varchar Sh\_guy\_vehicle\_id:varchar Payment\_method:varchar Account\_no:varchar

Database Schema

Admin(Admin\_id:varchar, Admin\_pword:varchar)

User(User\_id:varchar, Password:varchar, Email:varchar, Phn:int, Name:varchar, Type:varchar)

User\_bus(User\_id:varchar, Bus\_id:varchar)

Business(Bus\_id:varchar, Bus\_name:varchar, Bus\_owner:varchar, Bus\_products:varchar, Bus\_house\_no:int, Bus\_road\_no:int, Bus\_area/vill:varchar, Bus\_po:varchar, Bus\_pcode:int, Bus\_thana:varchar, Bus\_district:varchar, Bus\_shipping\_vehicle:varchar, Bus\_policies\_&\_licenses:varchar)

User\_store(User\_id:varchar, Store\_id:varchar)

Store(Store\_id:varchar, Store\_name:varchar, Store\_owner:varchar, Material\_types:varchar, Capacity:int, Available\_space:int, Store\_house\_no:int, Store\_Road\_no:int, Store\_area/vill:varchar, Store\_po:varchar, Store\_pcode:int, Store\_thana:varchar, Store\_district:varchar, Store\_shipping\_vehicle:varchar, Fee\_per\_kilo\_shipping:double, Fee\_per\_day\_per\_100\_product\_storing:double, Subscription\_status:varchar, Subscription\_expiry:DateTime, Store\_rating:double, Store\_owner\_rating:double, Policies\_&\_licenses:varchar)

User\_shipper(User\_id:varchar, Sh\_guy\_vehicle\_id:varchar)

Shipper(Sh\_guy\_vehicle\_id:varchar, Sh\_car:varchar, Vehicle\_status:varchar, Fee\_per\_kilo:double, Vehicle\_rating:double, Vehicle\_owner\_rating:double, Sh\_policies\_&\_licenses:varchar)

Order(Order\_no:int, Store\_id:varchar, Bus\_id:varchar, Product\_type:varchar, Product\_size/weight:varchar, From\_when:DateTime, To\_when:DateTime, Quantity:double, Sh\_from\_whr:varchar, Sh\_to\_whr:varchar)

Shipping\_order(Sh\_or\_no:int, Sh\_guy\_vehicle\_id:varchar, Order\_no:varchar, From\_whr:varchar, To\_whr:varchar, Departure:DateTime, Arrive:DateTime, Estimated\_arrival\_time:DateTime, Reached?:varchar, Paid?:varchar)

Payment\_account(Store\_id:varchar, Bus\_id:varchar, Sh\_guy\_vehicle\_id:varchar, Payment\_method:varchar, Account\_no:varchar)

Stored\_item(Store\_item\_id:varchar, Bus\_id:varchar, Store\_id:varchar, Order\_no:int, Product\_name:varchar, Product\_size:varchar, From\_date:DateTime, To\_Date:DateTime, Product\_status:varchar)

Pay\_order(Store\_id:varchar, Bus\_id:varchar, Sh\_guy\_vehicle\_id:varchar, Payment\_method:varchar, Account\_no:varchar)

Shipper\_order(Sh\_guy\_vehicle\_id:varchar, Sh\_or\_no:int)

Order\_ship(Sh\_or\_no:int, Order\_no:int)

Str\_Bu\_Or(Order\_no:int, Bus\_id:varchar, Store\_id:varchar)

Str\_itm\_order(Store\_item\_id:varchar, Order\_no:int)

Use Case Diagram:

Figure: Use Case Diagram

Str Owner

Admin

Shipper

Businessman

Admin Login use case scenario:

1. Input User ID and password.

2. Check user id and password (user authentication).

3. Save id and password.

4. Remember password.

5. Admin will have full control over every information. Admin can modify or delete any information.

Login use case scenario:

1. Input User ID and password.

2. Check user id and password (user authentication).

3. Save id and password.

4. Remember password.

Sign Up use case scenario:

1. Input user id, password, confirm password, name, address, date of birth.
2. Check user id unique or not.
3. Store new user account into the database.

Delete use case scenario:

1. The admin selects a ‘List User Accounts’ option.
2. The admin selects one user from the list, and then delete choose an option to delete that user account.
3. The system deletes the user the user account from the database.

Modify use case scenario:

1. Open user id, shipping status, Shipping guy status, name, address etc. for editing.
2. Add or reset the wanted field to change in desired way.
3. System modifies the database as given edit instruction.

Add-Store use case scenario:

1. Open ‘Add store’ option.
2. Add new store in the store list then enter.
3. The system adds new store in the database.

Store item use case scenario:

1. Open ‘Store item’ option.
2. Add new order to the store item task.
3. The system adds new order into the database.

Return item use case scenario:

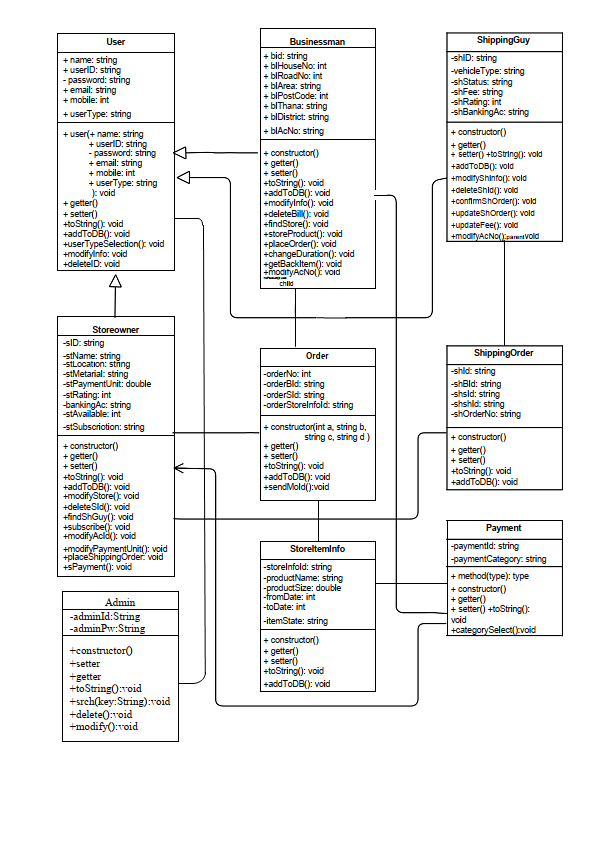
1. Open ‘Return Item’ option.
2. Add new return item task and provide necessary instruction.
3. The system adds new task on the returning item list.

Hire shipping guy use case scenario:

1. Open ‘Hire shipping guy’ option.
2. Check applicant’s details to match all conditions.
3. Add new name in shipping guy list.
4. The system will add name in the system.

Payment use case scenario:

1. Open ‘Payment’ use case scenario.
2. Insert payment amount for Store payment, Shipping payment, Subscription payment to pay or cut the fee.
3. The system implements the instruction and store the information on the database.

Class Diagram:

UML Class Diagram

Sequence Diagram:

