Design Specification On

HOSTEL MANAGEMENT SYSTEM

SARANCE SHRESTHA

00162792

Computing Project

Level 5 Diploma in Computing

Softwarica College of IT & E-Commerce

Kathmandu, Nepal

24th, may 2017

Contents

[4. Design 3](#_Toc483394177)

[4.1 Introduction 3](#_Toc483394178)

[4.2 Structural Model: 4](#_Toc483394179)

[4.2.1 Class Diagram: 4](#_Toc483394180)

[4.3 Behavioral Model: 4](#_Toc483394181)

[4.3.1 Activity Diagram: 4](#_Toc483394182)

[4.3.2 Sequence Diagram: 6](#_Toc483394183)

[4.4 Database Design 12](#_Toc483394184)

[4.4.1 Entity-Relationship Diagram: 12](#_Toc483394185)

[Conclusion: 14](#_Toc483394186)

# 4. Design

# 4.1 Introduction

The planning of software development, which shows the architectures and interfaces part of a system that helps the client in order to achieve their noted requirement is defined as planning. It is usually done before the coding phase of the system development in diagrammatic form followed by UML diagrams. It helps to clarifies upon the overall and inclusive design of the system and overall aspect of the system. Since, every modules of system is broke down into the small-small manageable chunk module that helps to make system much efficient and reliable. It generally focuses on the user acceptance design such as interface design and attractive GUI interface, easy to use and much effective. It provides the overall blueprint of all the requirement of the developing system that may be in interactive form or in diagrammatic form, which build the effectiveness of system.

Unified Modelling Language (UML), it is the standard notation modelling language, which is used for general purpose to visualize the design part of the system in a standard way and to provide modelling information to the client. Small diagrams of the system can be easily drawn by hand but as per the system becomes bigger and intricate UML provides the ease of maintenance over the system. It is totally used to build the diagrammatic part of the design process, which can be used to design the system design as per the requirement of the client and make them easy to understand with the vivid behavior and function of the system.

There are many kinds of tools that can be used to design the UML model for the design such as Visio, star UML, Visual Paradigm. Among them, I have used Visual Paradigm for developing the diagrammatic design of the system. Due to its easiness and user-friendly platform it helps in building the UML diagrams more easily, effectively and in understandable way.

There are various kinds of UML notation diagrams, which are discussed below with the related diagrams within it. The three approaches of design is discussed below, they are:

- Structural Model

- Behavioral Model

- Database Design

# 

# 4.2 Structural Model:

Structural Model defines the whole infrastructures aspect of the system. It generally describes the static model of the system rather than the behavioral model. This model visualizes the static structure, attributes, operations and their relationships. It mainly interpret the relationships of system modules but does not define the interactions handled by the system.

## 4.2.1 Class Diagram:

Class diagrams is static model of the system. It provides the logical structure of the system, classes, attribute of each classes, operations of each classes and the relationships between the classes. It does not shows the interactions between the classes.

The class diagram for the system is discussed below:

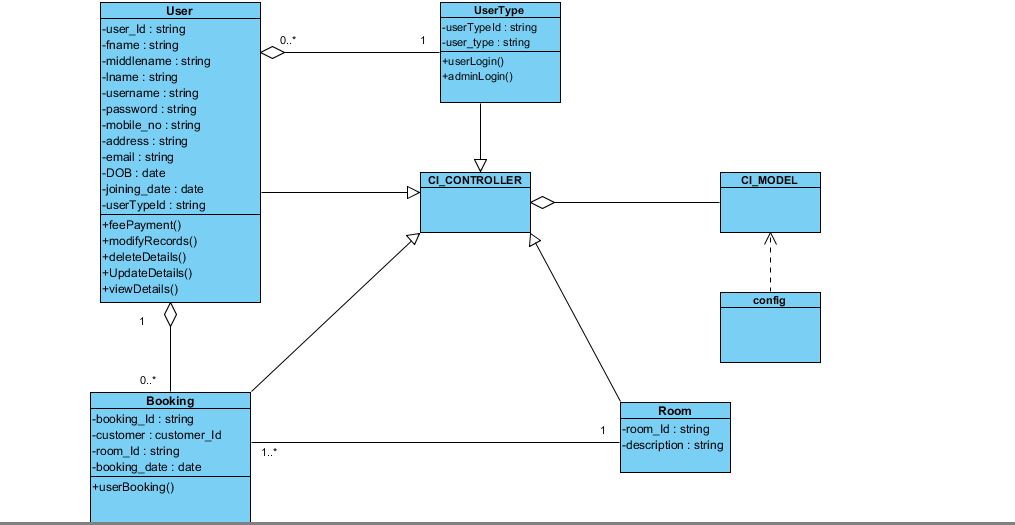


Figure 1 : Class diagram

In the class diagram, userType and user relates with the aggregation relationship because if the userType class gets destructed then another whole user class will not get destructed but its some objects only gets eliminated, in the same way aggregation relation exist between the booking , user and also between CI\_Controller and CI\_Model class.. Booking and Room are linked by a junction class booking\_room where there exist association relation because both of the class are independent of each other and doesn’t relate to each other. Config class i.e. of database class is use by the CI\_Model class, The necessary parts of the database are linked by that class which shows generalization relation in them. Finally CI\_Controller class shows generalization relation with user, userType, booking, booking\_room and room because all those classes extends the controller class in them.

# 4.3 Behavioral Model:

Behavioral Model defines the interactions and behaviors of the system. It exhibit the parts of actions contained by the system and need to be performed. It provides the detail information about the data management and processing of the system. It also shows the dynamic behaviors of the system defined in the action format.

There exist many kinds of behavioral diagrams; use case diagrams, activity diagrams, sequence diagrams, collaboration diagrams and so on.

Among the various diagrams I choosed activity and sequence diagram where Activity diagram, UML diagram is much simple to illustrate the activity or the workflow of a use case. It simply illustrate the flow of the activity, one or more than one activities that can be flown parallel, the kinds of decision that may appear and to derive the meaningful decision if any alternative workflow seems to be appear. It is widely used as to predict or clarify the workflow of the complex structures. Sequence diagrams is a logical structure comprising many elements such as lifeline, messages, focus of control, interaction frame, which helps to make the design more simpler and easy to understand. Many of the messages are flowed to another control, return message and recursive message, which makes the client easier to understand about the system and their requirement.

## 4.3.1 Activity Diagram:

It is behavioral diagram, considered as the workflow diagrams that show the flow of the activity or the action in the limited part of the system. It shows the workflow of the activity from the starting point to the end point explaining the decisions, activities, process and progression of the activity. They also show the logic of operations of the activities shown on class diagrams.

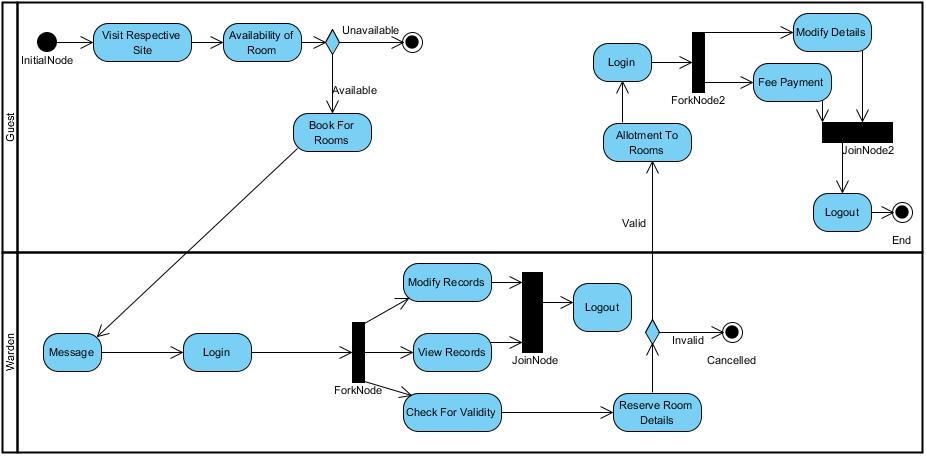


Figure 3: Activity diagram

The above diagram visualizes the flow of the activity from one activity to another activity along with decision parts. As shown in diagram, firstly guest visits the respective site of the organization to ensure there is room for booking. They check for the availability of the room then. If they find any vacant room then they book for the room providing their details of information. After then, the admin or head of the organization check out booking message. They login in into the system and check for the validity for the reserve room details. If the room is booked for the customer then they are allotted to the hostel rooms but if the reservation details is inappropriate then reservation is cancelled. Here admin is allowed to modify the records or information of the members if any of the member of the organization gets out of the organization after certain span of time, also they are able to view the every information of the member that are allotted to the organization. After the customer is allotted to the organization they are allowed to modify their own information and provide payment to the organization if only they login to the system. In the above diagram fork node is used which show the multiple activity that comes out after the single activity is done and join node show the result of only a activity that comes after the processing of the multiple parallel activities.

## 4.3.2 Sequence Diagram:

It is a behavioral UML diagram, which show the interactions and dynamic relations of the different modules of the system. Generally, it indicates how interactions happens in the order of time. This shows the various methods and their parameters are invoked into a system and shows the scope of the objects. Through this diagram, client may get the clear vision of methods on classes and what data they might expect on return. The kind of elements which is use to define the whole system, the messages are objects, lifeline, messages, focus of control, interaction frame.

The sequence diagram that represent the whole system is divided in the chunk part and is shown below:

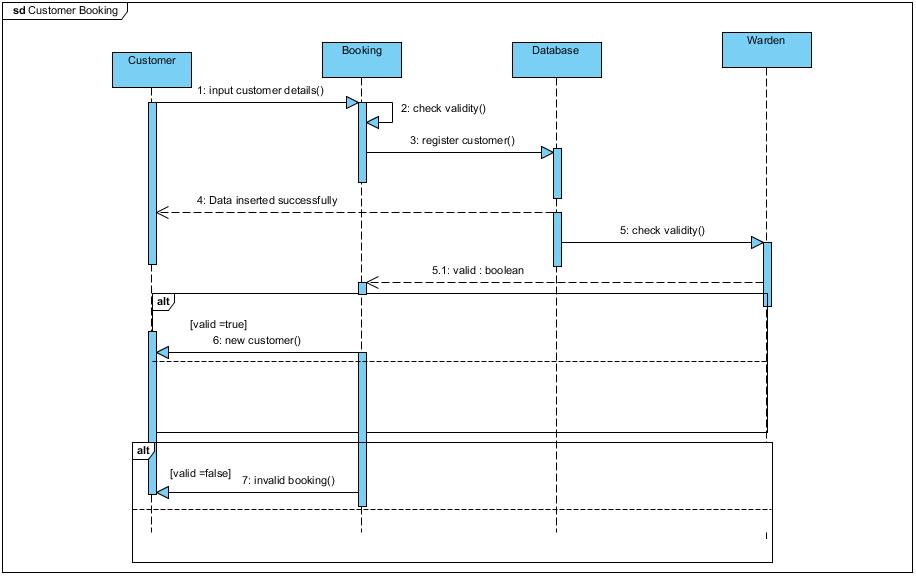


Figure 4: Customer Booking

In this diagram, customer provides their detail information during booking session. Then, the of validation is performed on their details. The registered details information are stored in database. Then warden performs the reservation validation using the Boolean value, if the reservation seems to be valid i.e. if the reservation details are true then the customer are allotted to the organization, if not reservation will be cancelled.

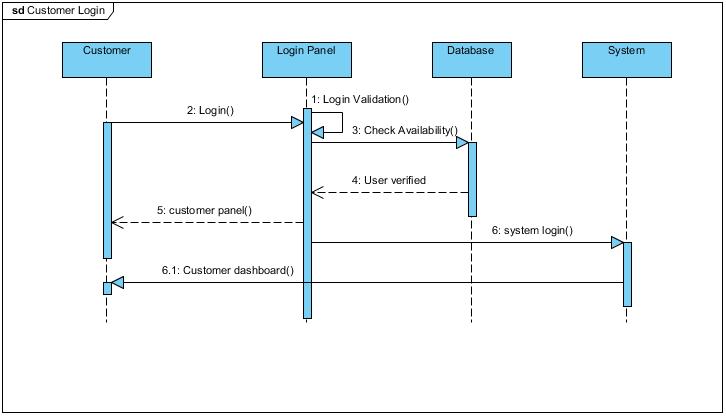


Figure 5: Customer login

In this diagram, customer requests for login access to their account by providing their username and password. A simple validation is done for their login and if the login get validated after checking from the database details then they gets access to the system also permitting the access to the customer dashboard.

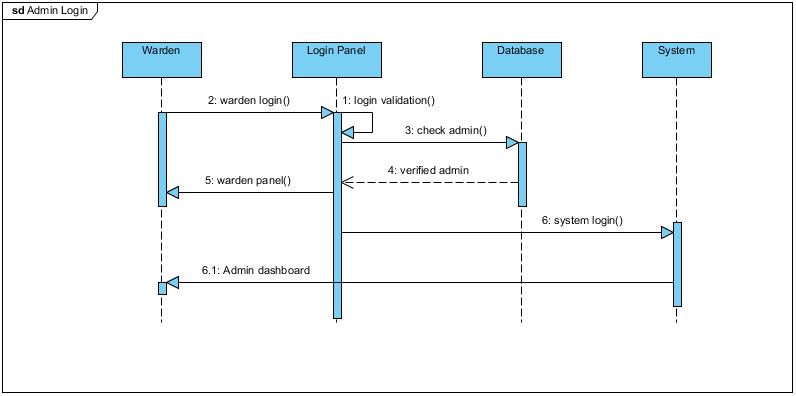


Figure 6: Admin login

In this diagram, admin performs the login process to get access to the system. A simple validation is done in this process, after the validation seems to be valid then the return message flows back to the login panel and then after login they get access to the system and system flows the message to go to the admin dashboard.

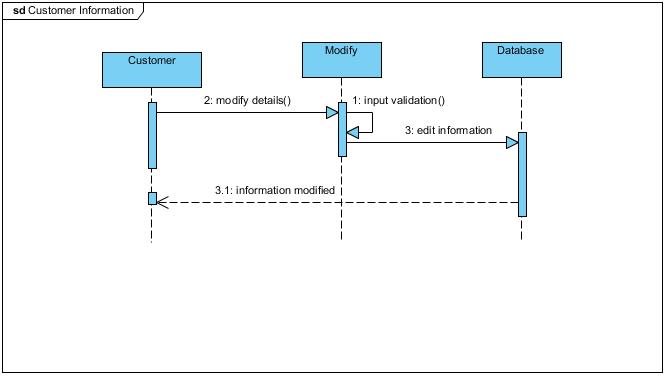


Figure 7: Modify information

In this diagram, customer provides the message to modify the detail information of them to the system and check for the input validation. If the validation is appropriate then it goes to the database to perform the modification status. Once the modification is done the return message ‘information modified’ is return to the customer.

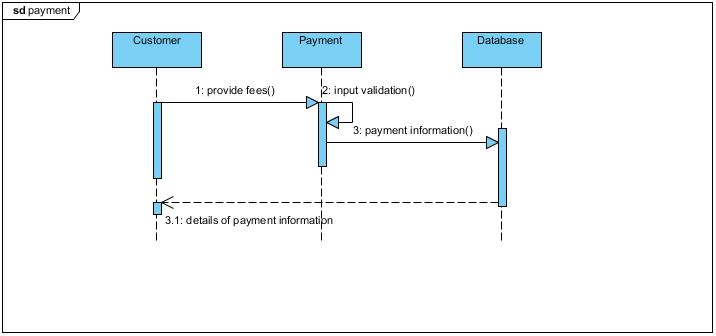


Figure 8: Fee payment

In this diagram, customer request for the payment dashboard, there the validation is checked for the customer’s details and the payment of fees is stored into the database. After then, a return message is send back to the customer ‘details of payment information’.

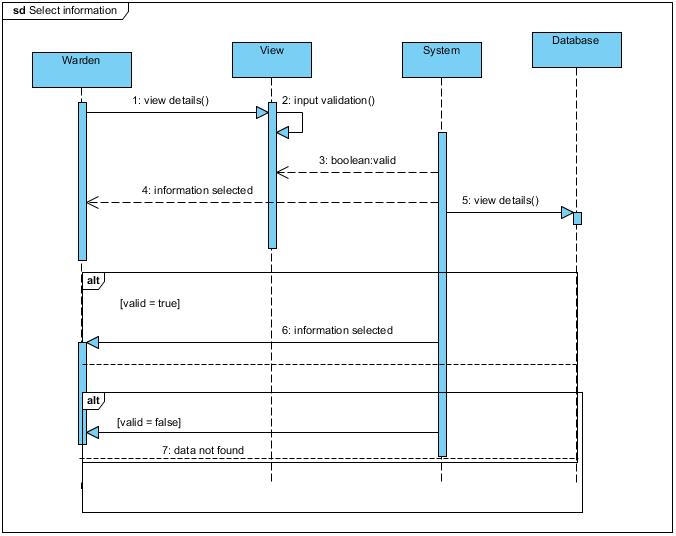


Figure 9: View member information

In this diagram, warden sends the view details() method to the system through view panel. The system comprises the details and checks in to the database. If the information exist in the database then it is view to the warden. Nevertheless, if the input data does not match to the recorded data then warden get the return message as ‘data not found’.

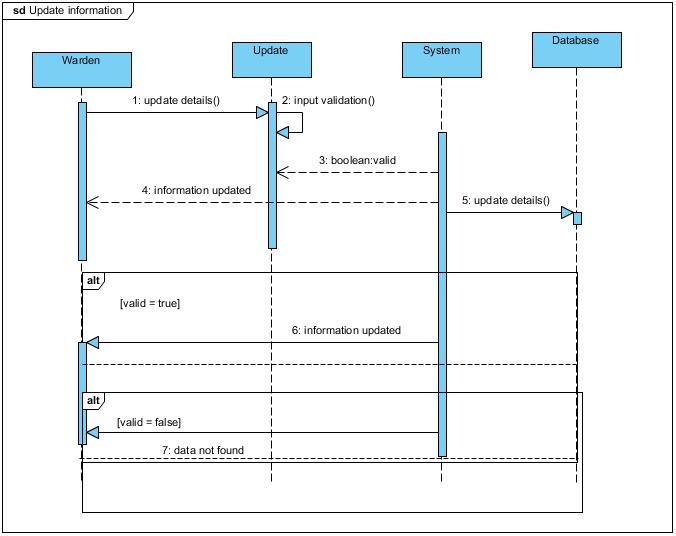


Figure 10: Update member information

In this diagram, warden or admin of the organization request system for updating any details of the member stored in database. In update panel the input validation is checked, and the method update details() flows to the database. If the record is found in the database then the system replies the message of information updated according to input validation. But if the input validation is not valid and the record is not found in database then it returns the message as ‘data not found’.

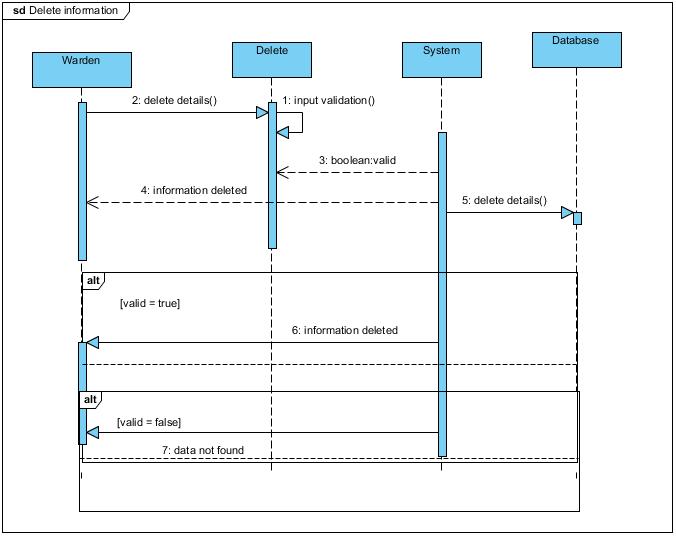


Figure 11: Delete member information

In this diagram, warden request to the system for deleting the records of some of the out casted member through the system. The input validation is done in the delete panel and Boolean value is given to the system to check the information in the database. The methods delete details() flows to the database and the delete action is performed if only, the input data exist in the database but if the input data is incorrect then the message ‘data not found’ flows back to the warden and urge for the process again with correct input of data.

# 4.4 Database Design

It visualizes the logical structure of a database, which contains the various constraints, relationships, Indexes, routines that defines the stored records and the appropriate way to access to those stored data. For this project, MySQL is use as the database. Database design mainly target on creating the database of fact data after all the requirements are gathered. It incorporates the conceptual, logical and physical design, where conceptual design contains only the initial concept of entities, relationship. Whereas, logical design contains the attributes for the entities, relationships between the entities, the normalization part and finally physical design incorporates the creation of table after the identification of entities and attributes, constraints, referential integrity, routines and security measures.

## 4.4.1 Entity-Relationship Diagram:

Entity-Relationship diagram shows the diagrammatic relationship among the various entities. This diagram represents the various entities and their respective attribute along with the relationship among the entities. There relationships can be classified into three part:

-One-to-One relationships

-One-to-Many relationships

-Many-to-Many relationships

The design of the database of the system is described below along with the E-R diagram:

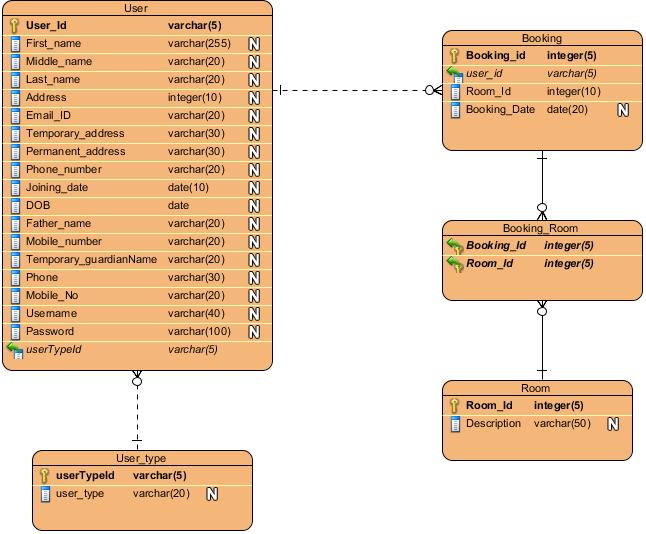


Figure 12 : E-R diagram

In the diagram, we can see number of entities are used to design the database model. The entities such as customer\_details, booking, warden, room, hostel, payment are used as to design the model. Each entities contains its own set of multiple attributes along with their own kind of relationship. Such as the relationship between the user and the booking is one-to-many. Since a user can perform many booking for the rooms of the hostel, but the booking done by that user cannot be performed by other users. So, there exist one-to-many relationships. Similarly, there exist one-to-many relationship from userType to user entities. since, there may be many users who gets login to the system such as user may be customer or the system admin. Therefore, userType shows one-to-many relation with user, where users may be different. Room and booking contains many-to-many relationship since a booking can be done for many rooms and many booking can be done for a single room. So, that’s why there exist many-to-may relationship between them.

So, in this way various relationship is held among the various entities listed to design the database model.

# Conclusion:

Hence, the design for the system is completed comprising different level of UML notations and diagrams. It shows the architectural and interface part of the system followed by different diagrammatical representation. It sums up to the different UML notational diagrams such as structural, behavioral and database diagrams. The sub part of those diagrams were also drawn to show the logical and interactional level of the design part which might help in better understanding of the system and might directly help clients for meeting there listed set of requirements. All the UML diagrams were constructed using **Visual Paradigm** tool.