Introduction

This paper is going to include the architectural description of the system. in addition to this it will have the description and decomposition of the modules.

This paper is all about the architectural and structural view of the system. This is very important to the software development because it will have a brief description of the system that is going to be built and also it will have the detailed knowledge on the modules that are going to be developed.

Some of the topics covered here are going to be directly converted to the implementation phase or the concepts are going to be converted to code.

The limitation of this documentation is, as said in the SRS documentation the concept of design we are using to develop this system is agile so, it is going to be very dynamic and changeable. In addition to this while development some changes may be done without the guide of this document so the listed configuration of the system down below my change through time.

To conclude, this documentation says to the developer how the system will work. It even tells how the modules are classified and how they work.

Architecture overview

The architectural style used here is going to be the most commonly used and most understandable way of implementing currently emerging systems, which is the MVC architecture.

Which this model is generally all about division of the work of classes. It has three components: -

* The view: It is the component that is responsible for displaying the information of the system. in our system, the web uses html and partly php and in the android system it uses XML. It is used for interaction between the user and system. or it is generally called us front end development.
* The Controller: This component is the component where the data interaction is available between the view(front end), and the database. This component accepts data from the users manages it and sends it to the model, the model analyses the data in to query and does the CRUD operation and then sends back to the controller and this controller sends the organized data to the view. In our web system this is done using PHP, in the android system it uses PHP and java.
* The Model: This component is the part where there is interaction between the data that came from controller and the database. It changes the data that came into a query for the sake of data manipulation from database. In our web system this is done using PHP, in the android system it uses PHP and java.

The Component models

This particular section deals with the component model which aspires to explain the subsystem decomposition in detail. And it describes the components of the system. These are the few components.

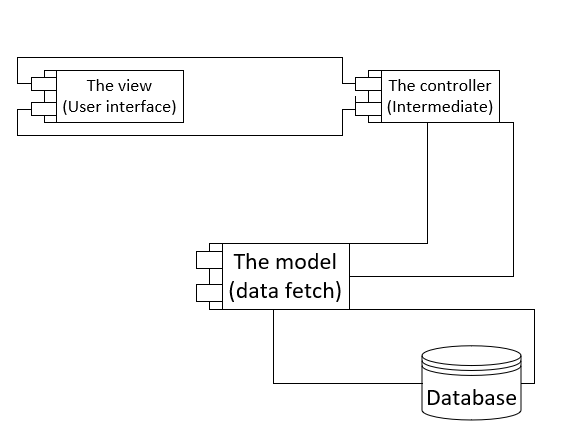


Figure 1.1 Component diagram for the MVC architecture

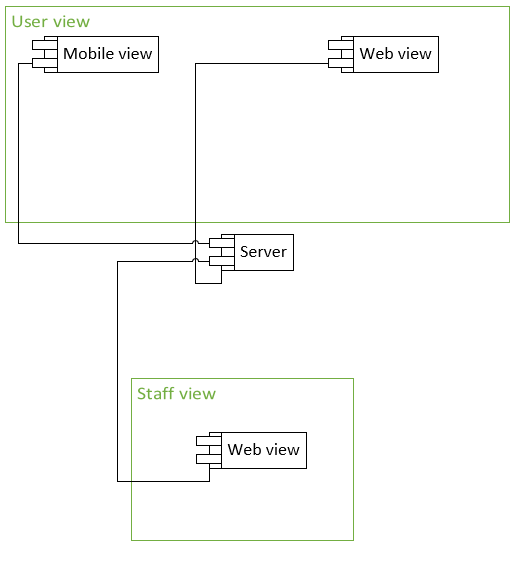


Figure 1.2 component diagram for user and staff members interaction with the system

The deployment models

In the deployment aspect we are thinking of using the 3-tier architecture. Because we need our system to get no bottle necked and crowded. Instead of describing it, the below diagram will tell more on how it works.

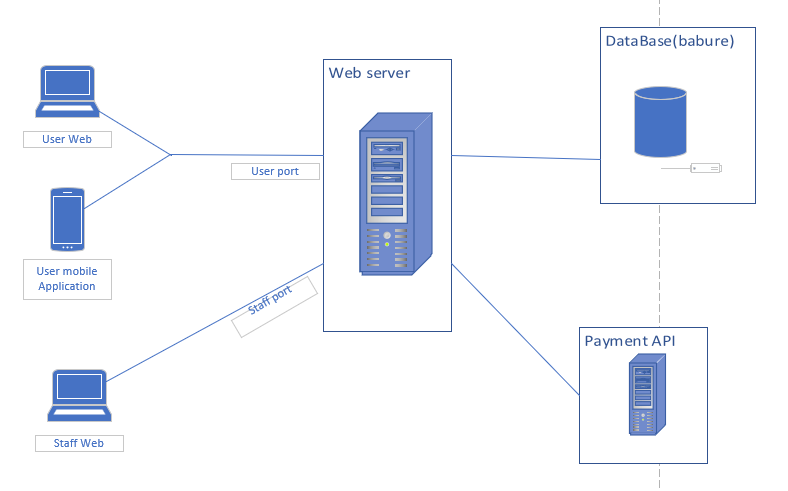


Figure 1.1 deployment diagram

Module Decomposition

This is a concept that deals with the division of labor or in other words the distribution of load in to sub-systems or components. This is very helpful to manage and work with, especially for testing and implementation. Because the decomposition makes it easier because a given component will only have one specific task, that makes it easily manageable.

In this part we will see each modules description and internal working of the decomposed systems. Like the data that will accept and the information that give or what it does with the given input data.

In our system there are a lot of modules, so we used a technique to decompose the system. and that system is we created components and put the features that are similar together for easy management.

Some of the components are listed below:-

* Login
* Reserve Ticket
* Book a ticket
* Create account
* View travel information
* Change reservation
* Cancel reservation
* Pay for booked tickets

These are the major components that we think are used to make the system full and work in a well-organized manner.

**The Login component**

This component as the name indicates is responsible for managing the activities of personals that access the system.

This component is used to manage staff members activities and give them privileges that are needed by them to do their work.

Also, this sub system is very important and must be well implemented because it our main line of defense from intruders. No user will have access to the staff portal if they don’t have access given to them by the admin.

It uses normal username and password combination to identify the personal who is accessing the system. It will also provide id numbers for the next page because the system has to record that is done and what is being done.

There is the usage of session for two purposes. One, for the sake of security. Because no one is allowed to enter with out the insertion of the username and password. And this session security helps us block any intruder with no privileges and there is no transition from one page to another without the session managing it especially from the home page of the staff portal to the login page and vice versa. Second, this session is used to identify who is accessing the system. it is mainly used in report generation and log reports.

The input parameters for this module or components are username (string datatype) and password (string datatype). The personal that access this component should insert this data’s and get authenticated. These parameters go through the normal architecture style meaning the view is responsible to display the form, the controller fetch’s the data from the user interface then sends it to the model, then the model uses the data and generates the query and fetch’s data from the database. Whilst checking it also validates if the user is authenticated and sends the user id to pass to the session.

This module uses the query operation select from the “staff\_account” table with the where condition that checks if the username and password are equal to the inserted data above.

The testing mechanism we used are manual testing but if we get time and finish our projects on time we might use php unit testing for this module. But our major testing mechanism is inserting data manually and checking if they will pass or not. The other testing mechanism is we insert the query directly to our MySQL DB and test it and verify the result.

**Reserving a ticket component**

This sub system deals with the major functionalities of the system. As the name indicates it is all about reserving a ticket.

This component can be both done by users and staff members. Where users reserve their own ticket using a user portal (web or mobile app). But while doing this some of the inserted data’s and some of the mechanisms might differ.

This component will require the login component. Meaning the staff members must login before managing user’s reservations.

As, this is the core component of the system its functionalities are implemented in a very serious and well-established manner. Other components will interact with it in different aspects and other components will use its output for their input mechanism.

To explain how it works. It starts by making the reservoir insert arrival date (date datatype), Arrival city (string datatype), departure city (string datatype). Then after insertion the system checks the seats left in each class levels (Hard sleeper, soft sleeper, normal) and displays the available seats to the user. Then after those are inserted, the user information is inserted by the reservoir. While that insertion of data is done it is only left with the payment mechanism. If the user is paying using online mechanisms, he /she will insert payment information in the API that provides the system will payment options. Or if the staff is managing the ticket and the user is paying using direct payment the staff accepts the cash and confirms payment mechanism.

When the staff members are managing the ticket for users, there id will be stored. Because when later use in report generation I will be needed. And if the user themselves were managing their own ticket, “user” will be stored in the reservoir attribute.

After each action are completed this module will produce ticket number also the ticket itself. This ticket will include the user’s information, seat number, class, ticket number. This are helpful later when the users are boarding their train. And also, the ticket number is also necessary in other components described below.

This module is going to use three tables “reservation” table, “user\_info” table and payment table for inserting data. And also, it uses the “route” table to get route id that is going to be used to store the routing information. This table is predefined.

Testing mechanism for this module is the same as above. We will try to test it using manually by inserting different types of data’s which can be illegible and false. And also, we will test the queries using our DB provider toc check the queries are well formed.

**Create account component**

This component deals with account creation of staff members. It gives staff members access to use the system.

This component is done by the actor admin. It is both initiated and managed by this actor only no other actors are allowed to get privileges. This is because if one actor gets this action’s they can create accounts and forge different data. There is a security risk there.

So, when we directly go to how it works. The admin wants to give privilege to a new employee. The admin opens the system to create account. Then the admin accepts the user information like full name (string datatype), email (string datatype), phone number (string data type), then inserts it into the system to create the account. And the system creates username and password form those given inputs. These all inserted and created systems will be stored in the DB.

The username and password of the staff will be sent to the staff’s email and will get its access to manipulate the system.

Testing for this component is also manual. We insert data into the system and the system will respond to those inserted data’s and we verify the output of the system. And also, we check the email sender if it is working properly or not. And us the above components we test the queries to check their correctness and efficiency.

Whilst creating this module we might face challenges like the email sender module might not work, or it needs an internet connection.

**Pay booked ticket**

This component is done by the staff members for the user that couldn’t reserve the ticket because don’t have the online payment mechanisms.

This sub system need a pre requirements like the user before coming to the staffs counter must book a ticket and the staff must login as well to do the needed action.

The staff members accept the booking reference of the user and checks if the reference number is correct or not. If it is correct, he/she will be forwarded to payment option. And the payment option will be cash only because it is direct payment to the staff’s counter.

After this is done the booked status will turn in to reserved and payment status will be active. And the user can travel to his destination city.

Testing of this component can be done using the previous means like entering data to this component and checking its results.