# Part1

## Introduction

## Background of the project

Bus Ticketing Management System (BTMS) is a web based software which focuses for the people who don’t have time to visit the bus ticket counter for ticket and it allows them to book ticket from home/office. For booking they can make a phone call or by using the software.

This system would also be benefit for each of the individual staffs of the counter which would increase the efficiency of the whole company. It would manage the information in a systematic and scientific way, also stores the data of the passengers for long term use. It would also address the needs of the company like their problem and their solutions.

## Problem background to the system

As the increasing number of the business it is the trouble for Bus Company to manage their daily routines and schedules. It is really difficult for them to manage the ticket counter as the counter is being over loaded. It is not the trouble for company only, but it is also the big problem for the traveler. Similarly it is very hard to take over all the problems by standing on the traditional system.

Hence, to manage all this trouble the system is being developed. And the system will addressed all our current troubles and makes us aware from possible worry.

## Current system

* It is the current system being used and can be replaced with a new system to addressed all problems.
* In this system, it is not possible to store data information for long period of time because it is based on paper which can be lost.
* It is really hard to analyze the data of the system.

## Drawbacks of current system

* It is time consuming system which also hampers the efficiency of business.
* Anybody can operate this which means data can be theft.
* It is not the scientific and systematic way, so lack of awareness about the technology is can be seen.

## Justification

* The system will improve all the possible functionality which the current system cannot deliver.
* The passengers can easily know their bus departure time, arrival time and ticket price from anywhere.
* It will be more advanced in terms of technology which will be useful for long period of time.
* It can display the counters telephone number and address with location.
* It will provide the users bus options with different offers.

## Merits of desired system

* Reduction of operation time.
* Passengers can choose the offered services by different company.
* It can identify the passenger’s details which control the fraudulent booking on ticket.
* It is well coordinated and centralized structure for service identification.
* It is easy to use and perform proof of concepts.

It is flexible as it allows the edit; update the system services as well as customers details.

# Aims

The vital aim of the project is to build a user interface for managing all the detailed information about the bus ticket in where all the details of the buses, routes, customers, staffs, etc can be stored in an organized way. All the details of the company can be regularly updated for the customers. It would helps in developing business process.

Some of the basic aims of the software:-

* To be a central portal where customers can enjoy the advantage of booking tickets of any bus from anywhere.
* To act as the means of easiest way for anyone to book any bus company ticket from this system.
* To empower the basic facilities.
* To involve more passengers in our system.
* Making the availability of bus ticket to the customer at low cost.

# Objectives

Above is the goal of the project which we have to achieve and to achieve those goals performing some actions. The bus ticketing management system will focus on its main objectives which would guide the company in its direction as well as making in decisions. Some of the major objectives are:-

* Analysis of bus management system to make it systematic.
* Providing the features of CRUD functions.
* All the manual work should be converted into computerized so that the load of employees should decrease.
* Proper training should be provided to the employee.
* Detail analysis of each and every individual problem.
* To meet all the requirements a best design should be done.
* Implementation and testing should be done properly.
* The project should be deployed in time.
* User friendly for booking and paying.
* For saving from thieving and crashing a safe and secure system should be developed.
* System should be easy to use and cost effective.
* It should be error free and any one can use it easily once they learn.

# Overview of the project design

As my project is based on web based whose main motive is to help the traveler to purchase ticket from their home or office without any stress. In this system passengers can be provided with different company’s bus as well as they have the option to select different price ticket. The system will able to store all types of passenger data and bus data. Similarly, we are providing the features to the passenger that they can choose the available seats own self. Both the passengers and admin have to login first before any task carry out. So they have to sign up each individual account with username and password for the security. The following points are the main

* Registration form from which new passengers can sing up and book ticket.
* Manage passengers information efficiently (CRUD passengers profile)
* Book ticket is the main feature which allows the passengers to book ticket for the reserve of the bus’s seat.
* Manage payment method as per the requirement.
* Allow to add bus and routes of different company.
* Helps to generate the different reports.

# Work breakdown structure

For developing the project in an easy and managed way all tasks are divided into small multiple portions on base of work breakdown structure.

It is the key for the project development that organizes each and every works into the team work into the different manageable sections. According to the Project Management Body of Knowledge (PMBOK) the WBS as a deliverable oriented hierarchical decomposition of the task which should be executed by the project team.

We use WBS because it helps to reduce complicated complex activities into the simple activities by changing complexity into simplicity. This would help the project manager to oversee all the complex activities with effectively which helps to grow the efficiency of software development process. All the task is clearly defined which are independent to each other.

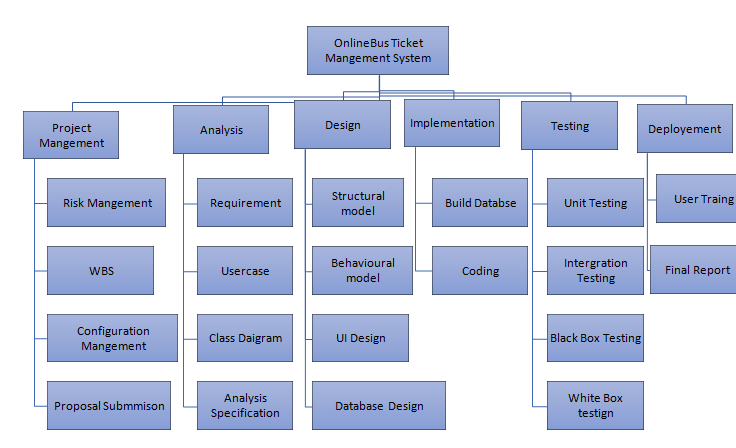


Figure Work break structure

# Time estimate/milestone

The milestones for this project are represented below in the table. We have figured out the some date with objectives and aim to reach all those date.

|  |  |
| --- | --- |
| **Milestones** | **Date** |
| **Project Management**  Risk Management  WBS  Configuration Management  Proposal Submission | 21/12/18-3/1/19  21/12/18-25/12/18  26/12/18-28/12/18  29/12/18-30/12/18  31/12/18-3/1/19 |
| **Analysis**  Requirement  Use Case  Architecture  Analysis specification | 4/1/19-28/1/19  4/1/19-10/1/19  11/1/19-16/1/19  17/1/19-22/1/19  23/1/19-28/1/19 |
| **Design**  Structural model  Behavioral model  UI Design  Database Design | 18/1/19-16/2/19  18/1/19-25/1/19  26/1/19-1/2/19  2/2/19-8/2/19  9/2/19-16/2/19 |
| **Implementation**  Building Database  Coding | 17/2/19-18/3/19  17/2/19-22/2/19  23/2/19-18/3/19 |
| **Testing**  Unit Testing  Integration Testing  Blackbox Testing  Whitebox Testing | 19/3/19-29/3/19  19/3/19-20/3/19  21/3/19-22/3/19  23/3/19-25/3/19  26/3/19-28/3/19 |
| **Deployment**  User Training  Final Report | 29/3/19-7/4/19  29/3/19-2/4/19  3/4/19-7/4/19 |

It is the tools used for the project management to focus on specific points along with a project timeline which planned for significant events and scheduled to occur at specific times. These events may be like project start date and end date. And milestones do not hamper in the project duration.

* **Project Management (14days)**

In this stage of milestone, I have successfully submitted my proposal for the project. Its task took 14 days all together. With proper planning and scoping the proposal was completed for the further milestone.

**Subtasks are:-**

* **Risk Management (5days)**
* **WBS Management (3 days)**
* **Configuration Management (2days)**
* **Proposal Submission (4days)**
* **Analysis (25days)**

It is the process of analyzing the gathered requirements to specify the vital requirements of the system. It is the predictable features of the software.

**Subtasks are:-**

* **Requirements (7days)**
* **Use case (6days)**
* **Architecture (6days)**
* **Analysis Specification (6days)**
* **Design (30 days)**

It is the core structure of the system which includes the design pattern like MVC design pattern etc. It is the representation of structural, behavioral and process of data carrying and the user interface design.

**Subtasks are:-**

* **Structural Model (8days)**
* **Behavioral Model (7days)**
* **UI design (7days)**
* **Database design (8days)**
* **Implementation (30days)**

In this step we give the last final system to the real world environment. It is the process of installing the finished product.

**Subtasks are:-**

* **Build Database (6days)**
* **Coding (24days)**
* **Testing (10 days)**

In this process testing is done within the team, its main motto is to find out the basic errors which can be solved in short period of time. It fixes all the initial trouble. It ensures the final system to be launch.

**Subtasks are:-**

* **Unit testing (2days)**
* **Integration testing (2days)**
* **Black box testing (3days)**
* **White box testing (3days)**
* **Deployment (10days)**

After the completion of the step of implementation people will first phase the problem of confusion, they may feel uneasy to run the system. So to solve this user training program can be conducted for 5 days and on the basis of training final report can be given.

**Subtasks are:-**

* **User training (5days)**
* **Final report (5days)**

# Scheduling

Scheduling can be defined as the process of listing of a project’s milestones, deliverables, and activities which intended start and finish dates. It is commonly used for the project planning and project portfolio management.

The development of project is started from the

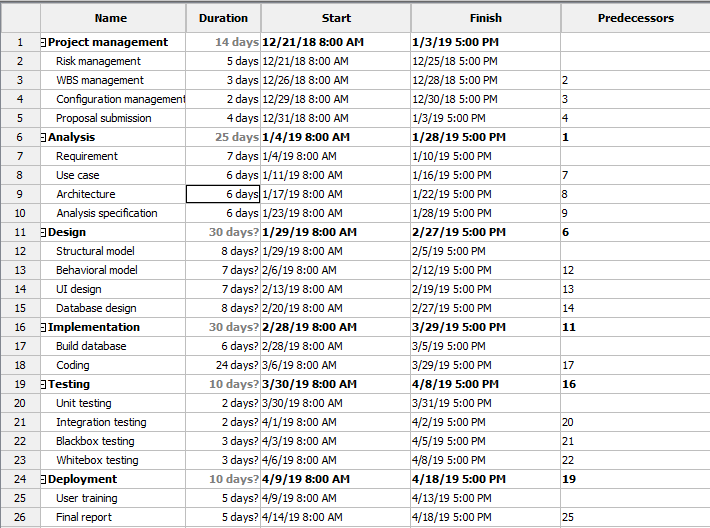


Figure scheduling

# Gantt chart

Similarly, A Gantt chart can be defined as the type of a bar chart which illustrates the project schedule with its start date and end date. In this chart vertical axis indicates the task to perform where as on the horizontal axis indicates the time intervals of the project. From this chart, it can be useful for planning, coordinating and track all the specific task of that project.

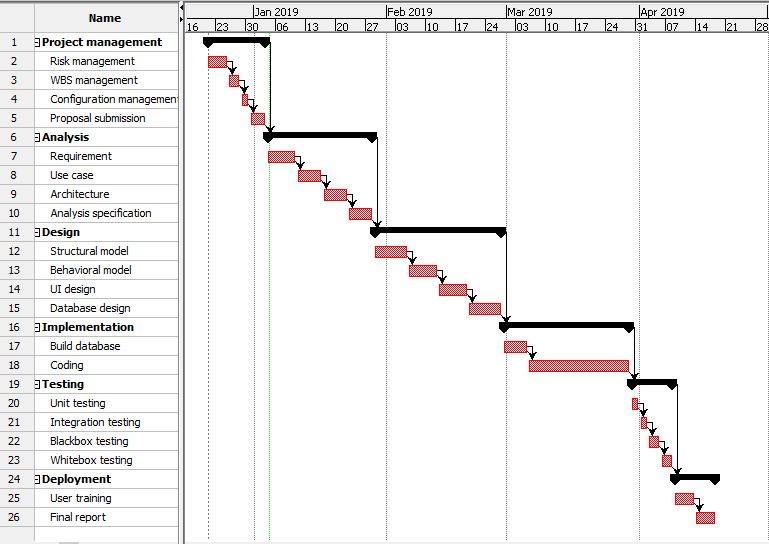


Figure Gantt chart

# Part2 Analysis

## Introduction

Analysis is the process of studying the nature of the complex topic and determining essentials features and their relations, which helps to divide that complex topic into smaller topics to gain better understanding. It will analyse the behavior of the system. For example: - robustness, correctness, safety, etc.

## Need for analysis

Analysis is the process of analyzing the task known as problem domain for the development of the conceptual model which can be used to complete the task in a systematic way. It also can help to describe the computer system that could be used to satisfy the set of the customer requirements. It is very important to carry out for the betterment of the system because it will help to collect and interpret the facts. In the same way to identify the problems and system’s objectives, it needs to be carried out.

## Object oriented analysis

It can be described as a technical approach for analyzing which can help to design the application and can build the whole system by the use of object-oriented programming. It is focus on capturing the structure and behavior of data systems into minor modules which combines the both data and process. It is mainly consists of three macro process which are called: - analysis, design and implementation. For the development of this BTMS, waterfall model is being used:-

**Water fall model:-**

To develop this software traditional way of systems development lifecycle (SDLC) waterfall model is going to be used. This model is the first process of model in the history of software development life cycle. It is too simple to understand and use. The waterfall model has the basic principle of completing each phase before the next phase begins. Finally, there is no overlapping between the phases.

The different stage of waterfall model with figure and description:-

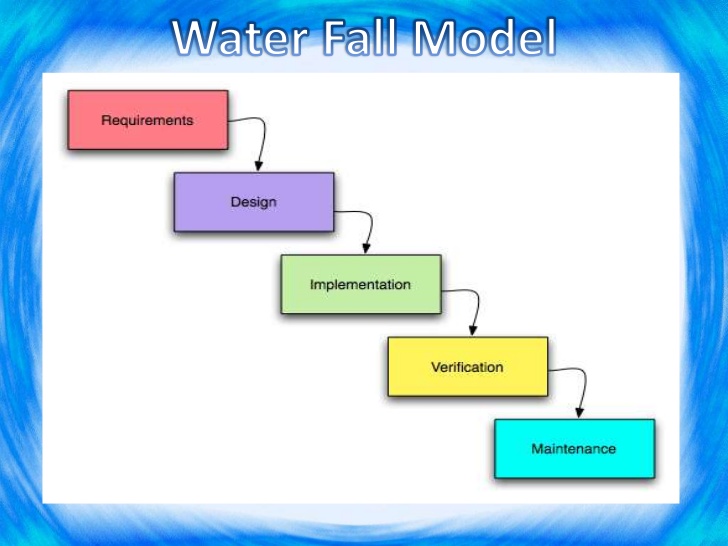


Figure water fall model

### *Figure 1- waterfall model*

* **Requirements:** The first step of waterfall model is to gather the requirements from the clients. From those requirements have to identify the vital one. And those vital documented requirements integrate our new systems purpose.
* **Design:** After requirement gathered and analysis our next phase is designing our system. In this phase we use various programming techniques to design both hardware as well as software to meet the objectives of the previous phase.
* **Coding:** It is one of the major steps to give the output of our design which is also our software. For this we will use various programming languages to implement the program as per the design that we already made at previous phase.
* **Testing:** In this stage we try to ensure that whether our output software works as it should be or the output from the software is correct or not, for this we tested the software for many times. This will ensure that whether the software meets the client requirements or not.
* **Implementation:** This is the final step of the waterfall model which is used to implement the software which was tested in previous step. In this process we give training about the software usability to the users. Also we see whether some errors going to occur or not.
* **Maintenance:** it is all about the evolution of the software. Before the software was not live the tester of the company cannot find out all the errors or bugs. But after software goes to the real environment the user may find out many problems which have to be addressed by the developer. So it is also a vital step of waterfall model.

All above steps of waterfall model should be following in a sequential manner. If you try to not follow the sequential manner then software development process cannot be done. It is easy to use and manage so we chose it for our project. The all process and results are well documented with systematic way.

**Some advantages of waterfall model are:-**

* It is suitable for small projects.
* Easy to manage.
* Requirements and analysis are well understood.
* Easy to understand.
* Milestones of software development are clear.
* Well documented.

**Some disadvantages of waterfall model are:-**

* It does not allow much revision or reflection.
* It is not suitable for large project and randomly changing requirement.
* High amount of risk and uncertainty.
* Poor model for long and on-going projects.
* Not a good model for complex object oriented projects.

**When to use it**

* When the requirements are well known, clear and fixed.
* If the project is short.
* If no ambiguous requirements.
* If the product define is stable.

## Requirements

It is the bridging gap between the user and developer where requirement analysis is done in the presence of all parties with a agreement. At the end of the software deployment the system must perform all the required needs of the clients for the successful of the system. It is also known as the final agreement between the developer and client.

As the requirements is consists of both the functional and non-functional requirements.

Following are the some of the requirements gathering techniques:-

* Interview
* Observation
* Questionnaire
* Brainstorming
* Survey

**Interview**

It is the best way for gathering the real requirements for the system and to know the actual needs of the client. In this technique conversation is done with the clients.

**Observation**

In this step, the requirement can be gathered by seeing and heard in the real environment where the system is going to be deployed.

**Questionnaire**

In this process of requirement’s study, user can be asked different questions related to their software. It is the best way to understand the deep requirements of the system.

**Brainstorming**

The group discussion to propose new idea but no judgments and no decisions can be getting. It is mainly focused to identify the main problems and helps to exchange the ideas.

**survey**

It is the process of collecting the information from people about how they think and act. It determines the specified people opinions. It is the type of collecting information which consists of both interview and questionnaire.

## Functional requirements

It is the operational activity of the system which is the compulsory requirements of the software without the software cannot be imagined. This is the statement of the software which shows how the system will reacts to the specific inputs with MoSCow prioritization.

Following are the functional requirements of BTMS:-

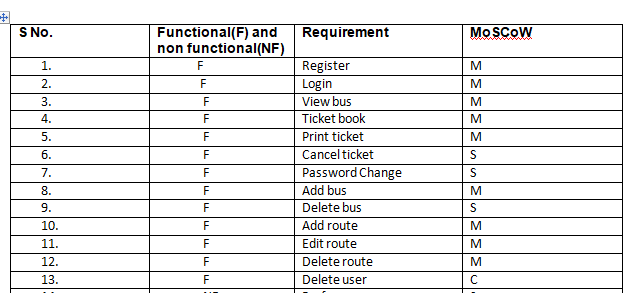


Figure Functional/non-functional/MoSCow table

## Non-functional requirements

It is the operation of the system which is not so compulsory for the system but needed. It also can be called as invisible features but it is required for the smooth running of the system. For example of non-functional requirements used in bus ticketing management system (BTMS) are:-

* Performance

It is the act of performing the task/operation of the system.

* Usability

It is the requirements which help to determine, is the system helping the user with its functions. Is it useful for the people? It identifies all this questions answers.

* Availability

It insures that system should ran in every working hours. It is the quality of being able to use when needed.

* Security

it is related with protecting the system’s attributes, data and all those private data of the system.

* Reliability

Every calculation performed needs to be accurate. This will definitely increase the system’s reliability.

## Prioritization

It is the process of dividing the gathered requirements of functional and non-functional in the order of priority. I ordered my collected tasks and identified the most important ones.

For the prioritization process to make it easy, I preferred for MoSCoW prioritization which can complete the task of prioritization in an efficient way.

MoSCoW stands for:-

**M=Must have**

In the case of every project, without this projects cannot be completed.

* Very necessary
* Without it project become incomplete and cannot be delivered.

**S=should have**

It is the something, if did then it will be 100% beneficial for the system.

* Important not necessary
* Without it system looks incomplete but still feasible.

**C=could have**

Project is tolerable if it is used as alternatively.

* Not so important
* Little influenced if it is left.

**W=would have**

These functions will not be planned in the system. But can be done if stakeholders ask about it.

* More threats can be generates but recompenses can e executed

|  |  |  |  |
| --- | --- | --- | --- |
| **S No.** | **Functional(F) and non functional(NF)** | **Requirement** | **MoSCoW** |
|  | F | Register | M |
|  | F | Login | M |
|  | F | View bus | M |
|  | F | Ticket book | M |
|  | F | Print ticket | M |
|  | F | Cancel ticket | S |
|  | F | Password Change | S |
|  | F | Add bus | M |
|  | F | Delete bus | S |
|  | F | Add route | M |
|  | F | Edit route | M |
|  | F | Delete route | M |
|  | F | Delete user | C |
|  | NF | Performance | S |
|  | NF | Usability | S |
|  | NF | Availability | M |
|  | NF | Security | M |
|  | NF | Relaibility | M |

**Merits of Prioritization**

* It identifies the system’s core requirements to meet the client needs.
* It saves the developer time because must have requirements were already fixed.
* It makes easier to the developer.
* It helps to deliver the software in time.

**Demerits of Prioritization**

* It can take a little bit time for classifying the requirements in different category of prioritization.
* Due to the completion of work in time, saves time for the developer, may be the reason for more free time and less re-work.
* Only urgent work can get attention which is not result good for other less important work.

## Natural Language Analysis

It is a way to process sentence in a natural language. It is a type of method to fetch candidate classes and operations for the system development process

In NLA, first we get the nouns as candidate classes and verbs as operations. And for my system it is shown only noun which is the final candidate class below:

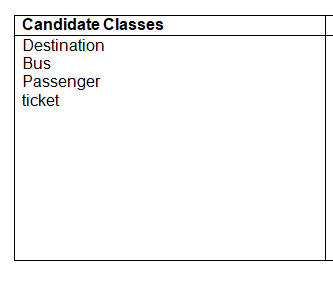


Figure 6 NLA candidate class list

## Initial class diagram

Class diagram is a type of static structure diagram that describes the structure of a system. And the initial class diagram is drawn during the period of analysis. This diagram is the foundation for the final class diagram.

The initial class diagram of BTMS is:-

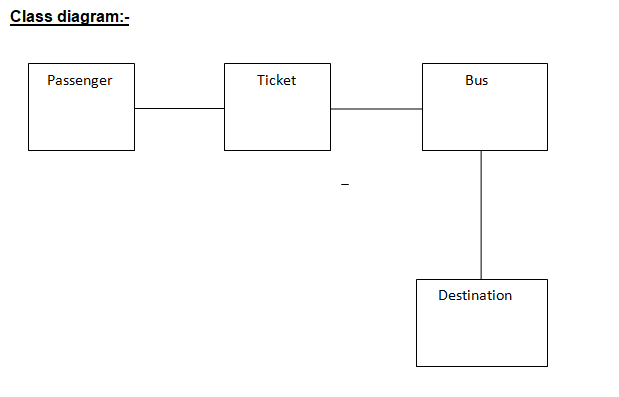


Figure 7 Initial class diagram

## Use cases

A use-case diagram is the simplest graphical representation of user’s interaction with the system to achieve particular goals of the system. In this actors can be both the human and the any external system. It is mainly focused to identify the system’s vital requirements.

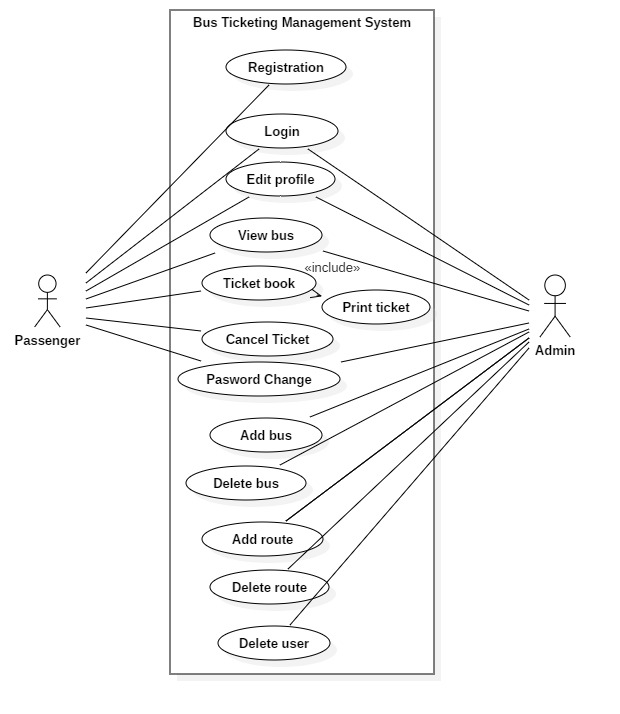


Figure 8 Use case diagram for bus ticketing system

Justification table for using use-case which indicates the actors task to be perform i.e. the every functions drawn in above use-case does the following:-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Title | Actor | Goal | User Story |
| 1. | Registration | Passenger | Sign up new account | -registers all the necessary details |
| 2. | Login | Admin/Passenger | Access to the account | -Fill with username and password.  -Click on login button |
| 3. | Edit profile | Admin/Passenger | Update users profile | -click on the profile  -change the data which is needed  -edit with new data  -click on the update button |
| 4. | View bus | Passenger | Shows the different company’s bus | -allow to choose appropriate bus |
| 5. | Ticket book | Passenger | Allow to reserve ticket | -after choosing the bus it allows to book ticket for travel |
| 6. | Print ticket | System | Printing the ticket | -on clicking print ticket by user system itself generate a ticket |
| 7. | Cancel ticket | Passenger | Cancelation of ticket | -user should click on the cancel button |
| 8. | Password change | Admin/Passenger | Updating password | -password of each individual can be changed |
| 9. | Add bus | Admin | Adding new bus and their company | -bus can be added to the system with its details |
| 10. | Delete bus | Admin | Removing existing bus from system | -bus can be removed from the system |
| 11. | Add route | Admin | Adding new route | -as necessary new route can be added |

## System Architecture

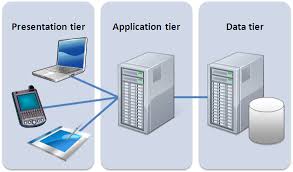
For the system architecture development of this project we are going to use three-tier architecture. It can be defined as the client-server architecture in which the functional process logic, computer data storage, user interface and data access are developed and can be maintained as independent modules on separate platforms. It is consist of three basic tiers which are presentation tier, business logic tier and data processing tier.

Figure -Three-tier architecture

**Presentation tier**: - It is the front end layer of the 3 tier system and consists of user interface from which the user input the request and it displays the request which is done. The user interface is graphical one accessible through a web browser or web based application. Its main function is to display the content and information useful for the end user.

**Application tier**: - This is the tier of full of business logical tier and which is the processor or indicates the central application server of the system.

**Data tier**: - It is a clustered database which is used in the network storage. If the primary server goes down it helps to maintain the smooth run of the system.

# Chapter3 Design

## Overview of design

Software designing is the set of primitive components in which agents create a specific artifact of the software to accomplish the goals. It usually involves for problem solving and gives the sight of the system. In my BTMS following four different design patterns are illustrated besides the documentations are:-

* Behavioral modeling
* Structural modeling
* Database modeling
* UI modeling

## Structural modeling

According to the software engineering structural design patterns can be defined as the

Design pattern that eases the designing by identifying a simple way to realize the relationship between the entities of the software.

Now, I’m going to discuss about the class diagram:-

## Final Class diagram

It is a static structure diagram based on the Unified Modeling Language (UML) that defines and illustrates the overview & structure of the system in terms of classes, attributes and method. It also shows the relationship between the objects.

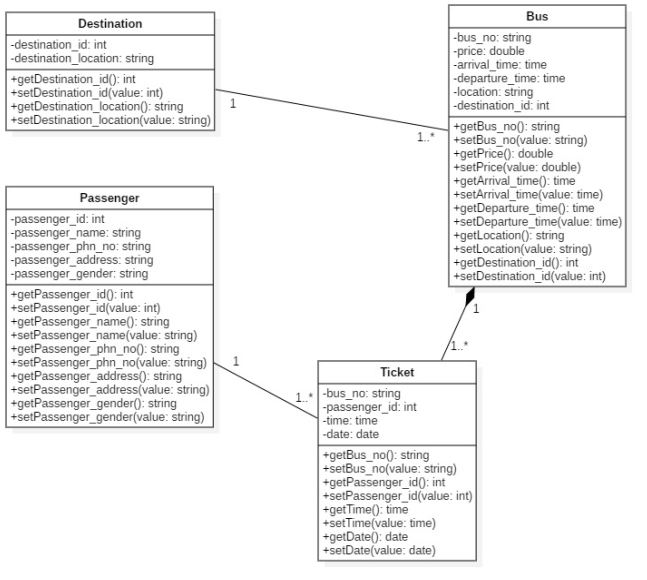
****

Figure 10 Final class diagram

## Sequence diagram

It is a diagram shows object interactions arranged in time sequence which depicts the objects and classes involved in the scenario and the message can be exchanged between the objects needed to carry out functionally.

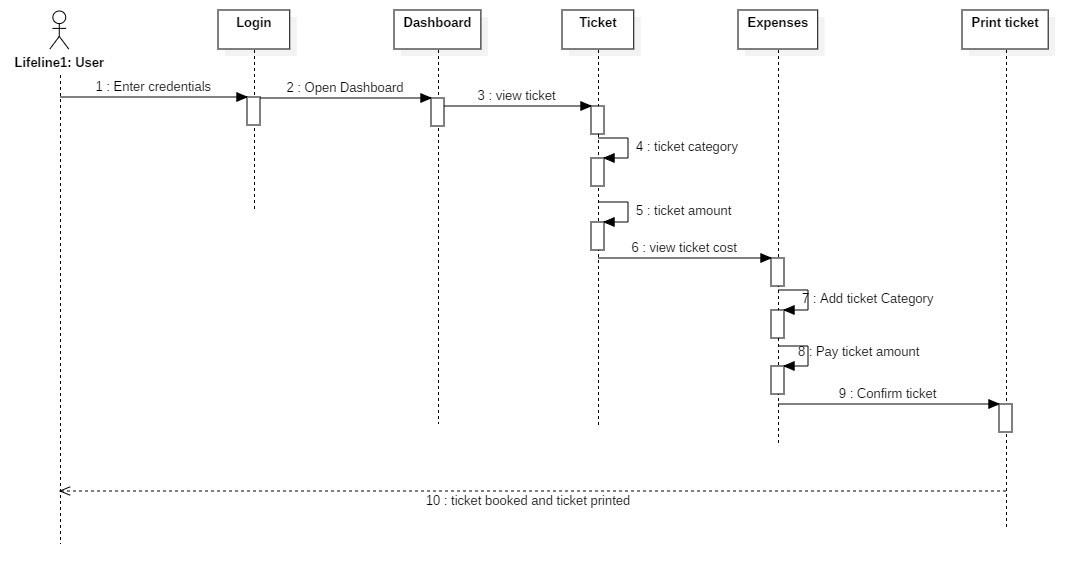


Figure sequence diagram

## Data Flow Diagram (DFD)

It is a kind of diagram in software engineering which is used to represent the flow of a data of process of a system; it also provides the information about the outputs and inputs of each entity and the process. It has no control flow, no decision rules and no loops. The data flow diagram for the BTMS: -

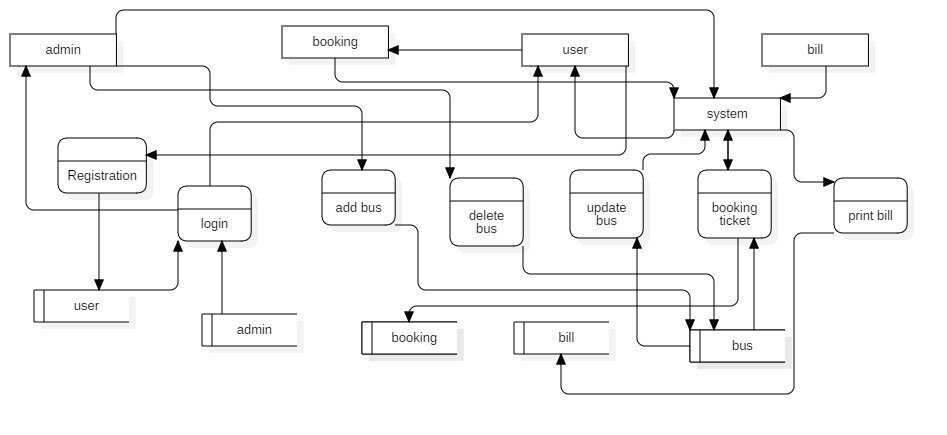


Figure Data flow diagram

## Justification for class diagram

By the use of class diagram, it provides us the detailed insight into the structure of our system. Secondly, it provides the quick overview of the system elements and their relationship. It also can be directly used with object oriented languages.

## Behavioral modeling

The design directly related with the shape and got impact with the people behavior is called behavioral design. It always increase the flexibility in the system for the enhancement of the system.

Activity diagram

An activity diagram can be described as the diagram in which it illustrates the use case functions into the detailed form; it is the process of execution of the set procedural activities of the system with its initial point and end point. On this diagram there can be used of swim lane to separate the operational values of the each actors which can be given in horizontal or vertical form.

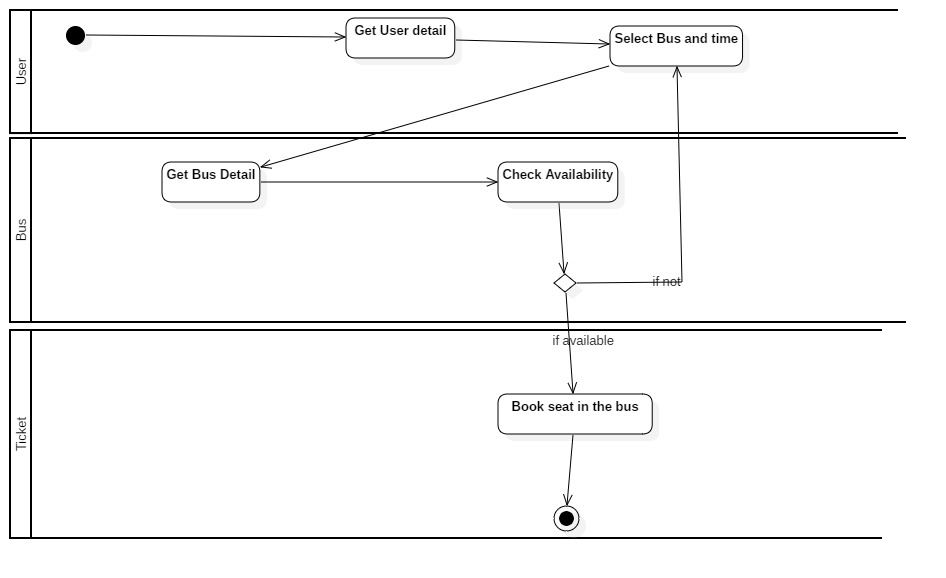


Figure :-Activity diagram for bus ticketing

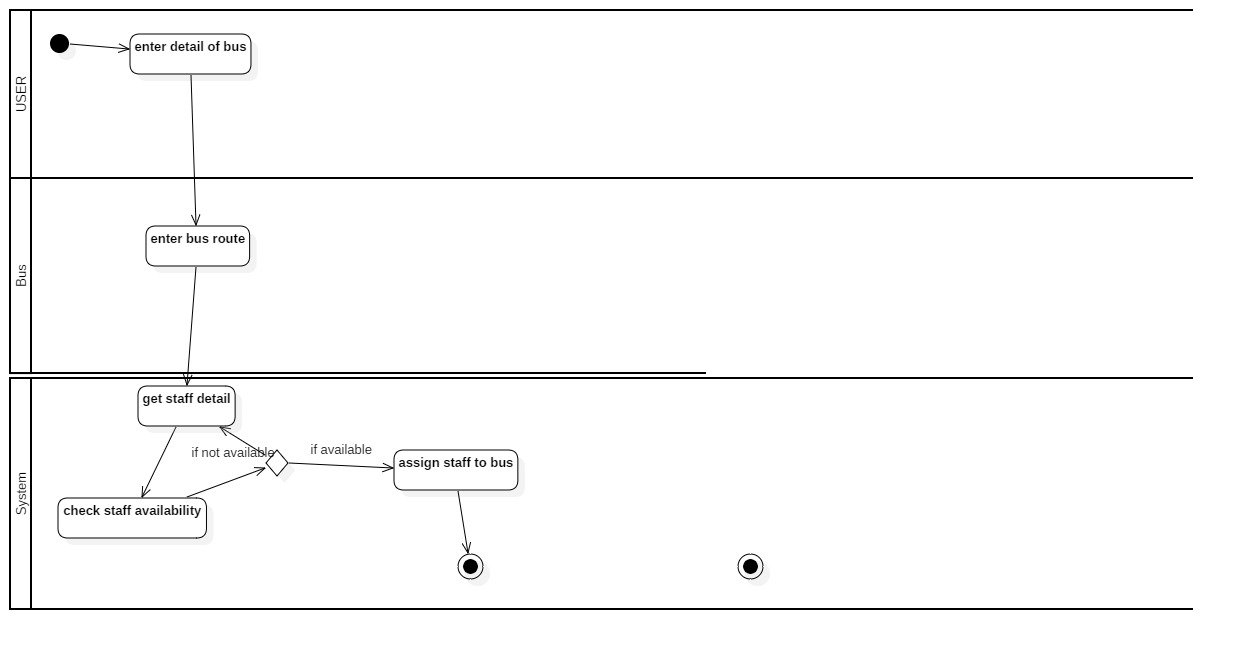


Figure 14 activity diagram for adding bus

Figure:- adding bus

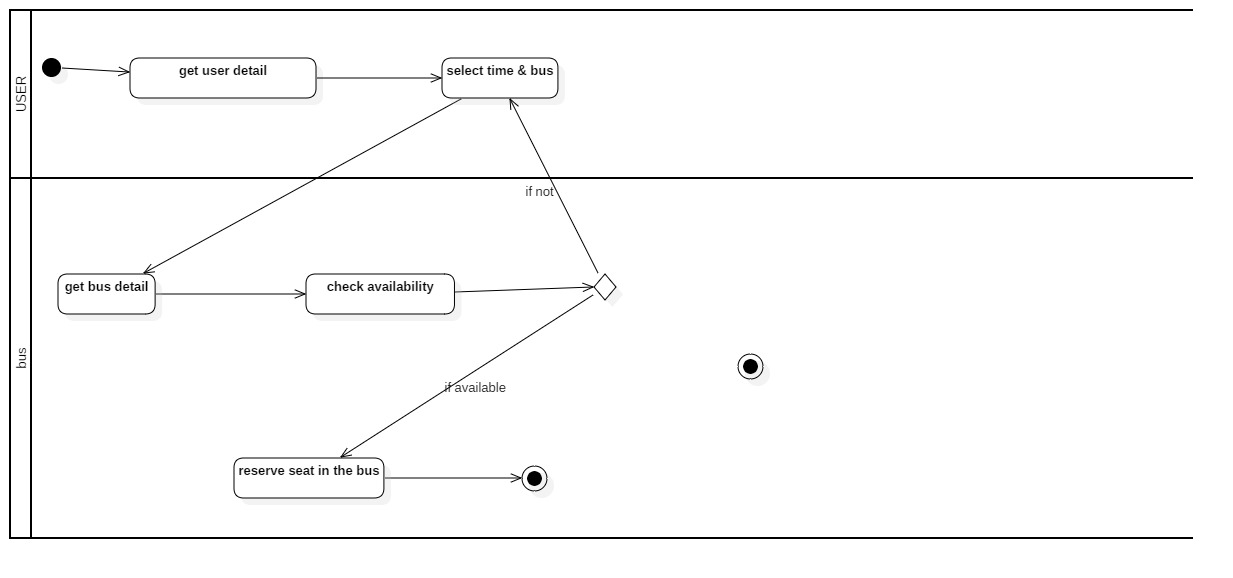


Figure 15 activity diagram for ticket booking

## Database modeling

It shows the data model in the system and also shows the data handled process. Similarly a poor database design can cause the system failure in the future. If a new type of data comes which may have relation with previous kind of data but the previous design is worst that the relation may not be maintained which makes very difficult to implement. So before developing any system, we have to be focused for the better development of the database design.

## ER diagram

It is stand for Entity-Relationship Diagram, which is a data modeling technique which is used to illustrate the system’s entities and their relationship with each other. It is also called as the conceptual and representational model of data used for representing the entity framework infrastructure. Similarly, it shows the relationship among the people, objects, places, concepts and events within the system. It uses different kind of graphical symbols to represent the relationship of (one to one, one too many, many to many).

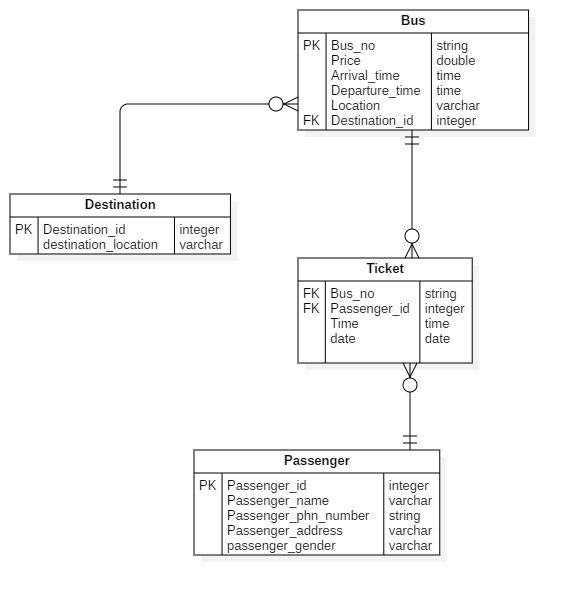


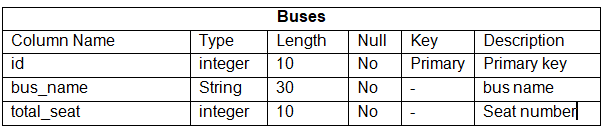
Figure ER diagram

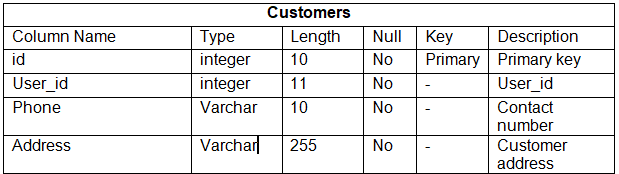
**Justification:-**

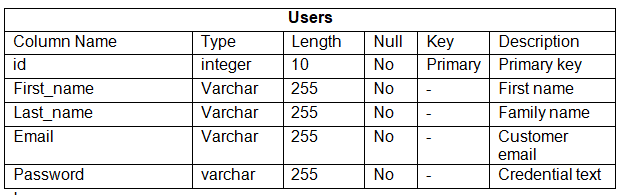
ER diagram helps me out to implements the database and tables which is needed for my system to operate its operation. Above my ER diagram is in normalization form so if there is any update in my system, i can be able to update and maintain my ER diagram.

## Data dictionary

By the IBM dictionary of computing, a data dictionary can be defined as the centralized collection of information about the data such as meaning, relationship to other data, origin, format, etc. In the case of our system it can be defined as a collection of tables with its metadata.







## UI modeling

User Interface Modeling (UI modeling) is a type of software development technique used by the computer application programmers which is the complex software components which play a vital role in the usability of the system.

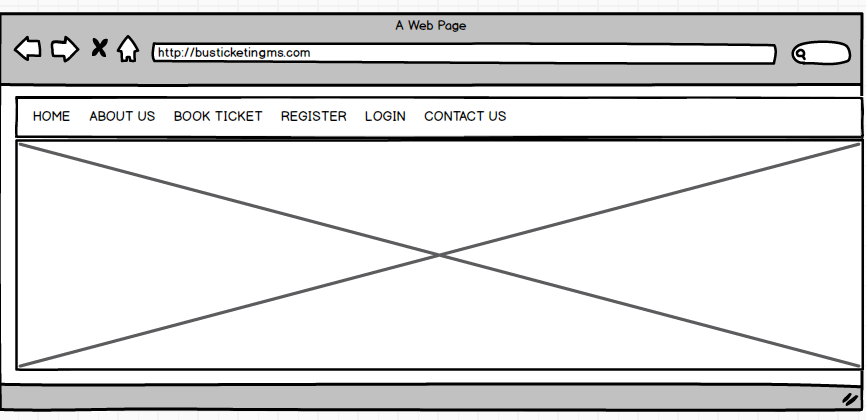


Figure home page

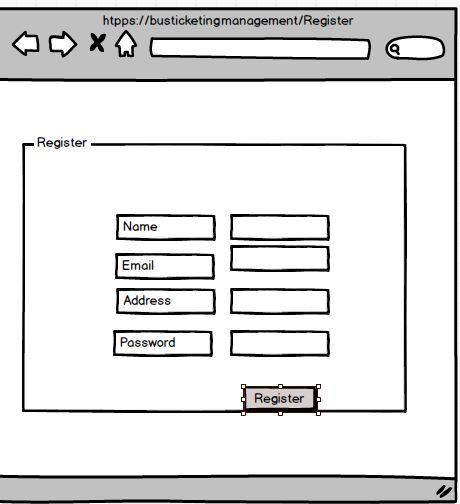


Figure register

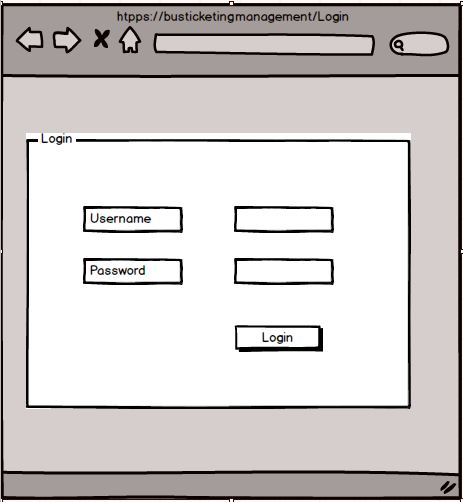


Figure Login

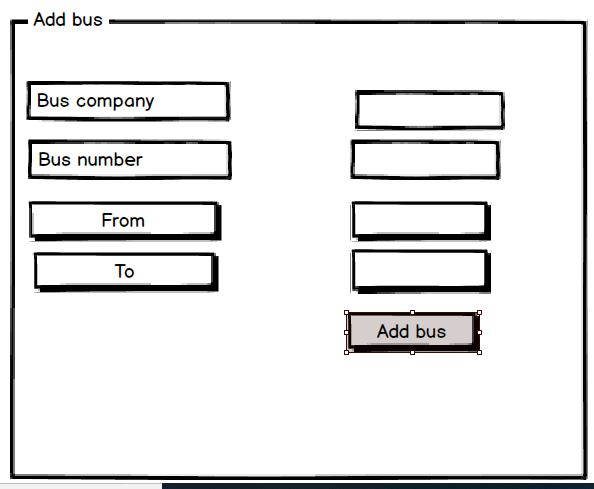


Figure Add bus

# Part 4

## Implementation

## Introduction

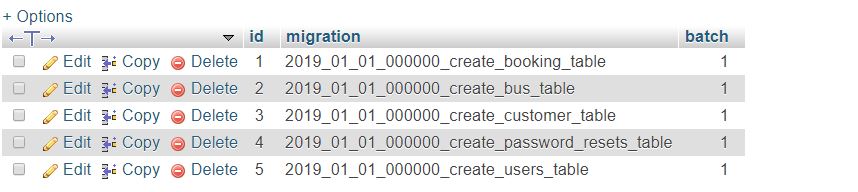
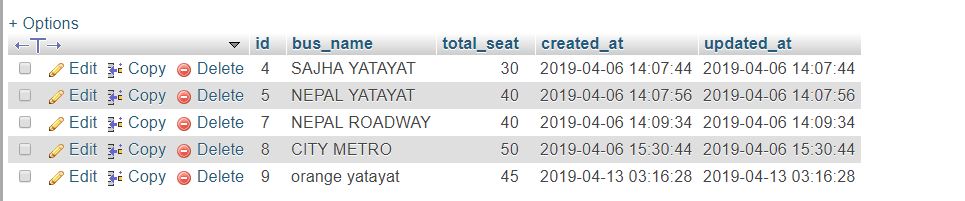
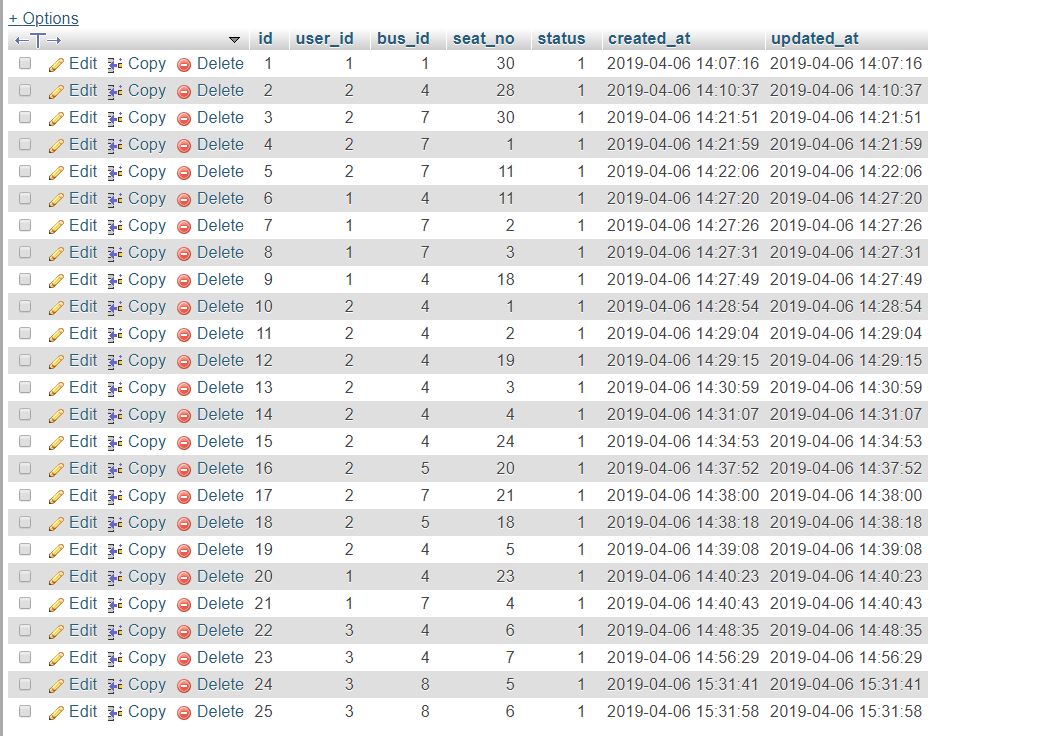
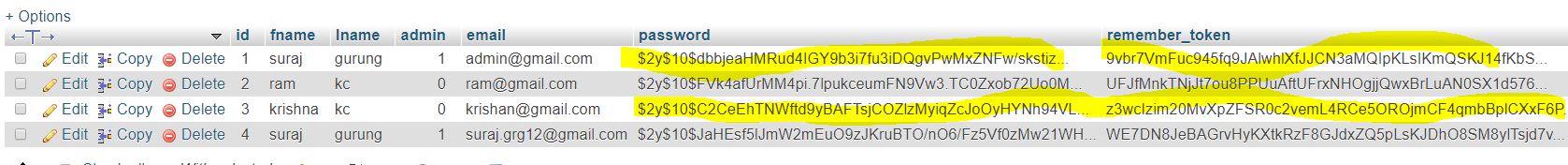
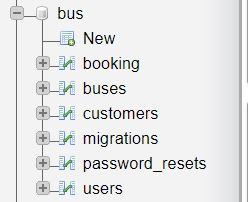
Implementation is regarded as one of the phase of the software development cycle, in this phase how the software going to be developed ,which programming language can be used, which developing platform such as sublime, Laravel, etc used, it fixed all this things.

## Technology stack used

* Front end : HTML, CSS
* Back end : My SQL database, PHP
* Framework : laravel

As above in the technology stack, we can clearly see it that for the front end designing of the system HTML and CSS is used, where as for the back end development of the system, PHP as the main programming language used. Similarly, for the database My SQL is used and for the development of system framework or as a development platform laravel is being used.



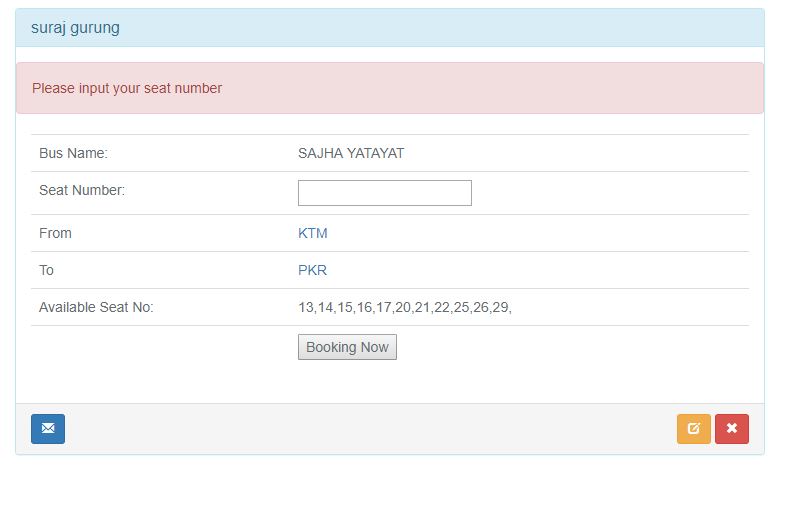
****

# Chapter5 Testing

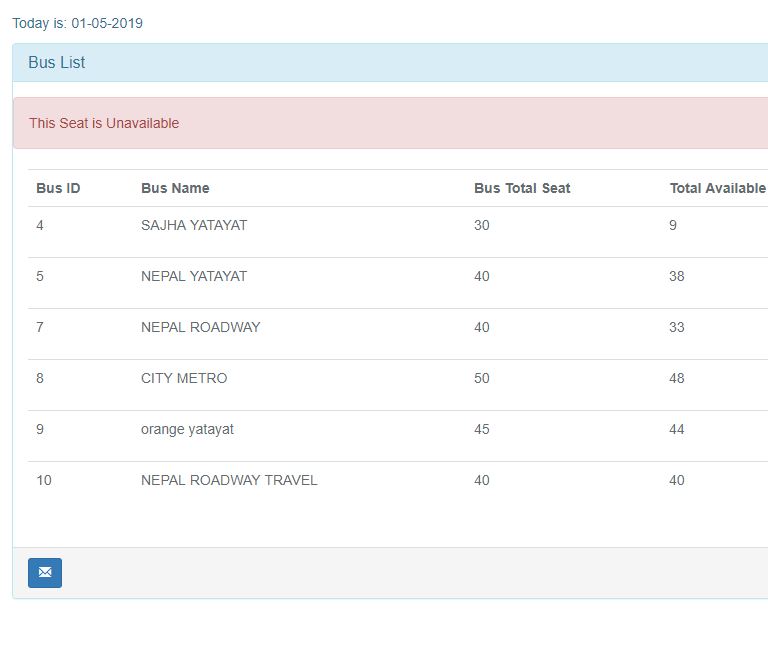
Testing is a method that can be used to test the design and structure of the code in a system which is not known to the tester and engineers. There are different type of testing techniques are in practice, among them black box testing and unit testing can be described in a detail way with it’s test cases and test logs:-

## Black box testing

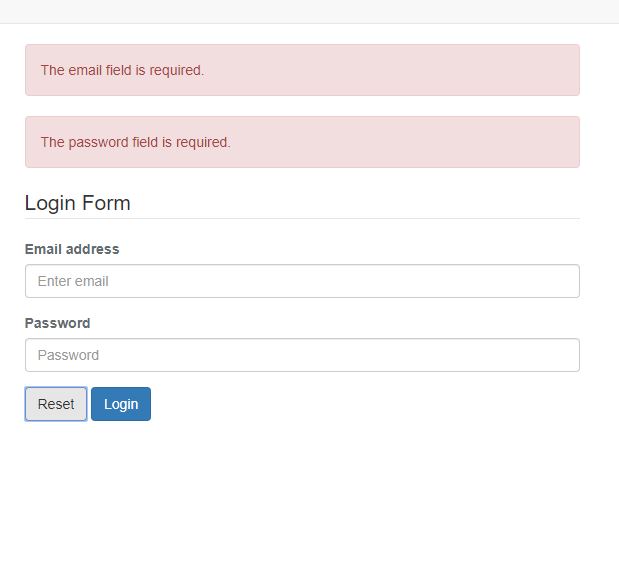
The process of testing the internal structure of a system, design and implementation of the item through passing a input to the system with a output, which is going to be tested is called black box testing. It is also known as behavioral testing. And the test of this testing can test both functional and non-functional requirements of the system.

****

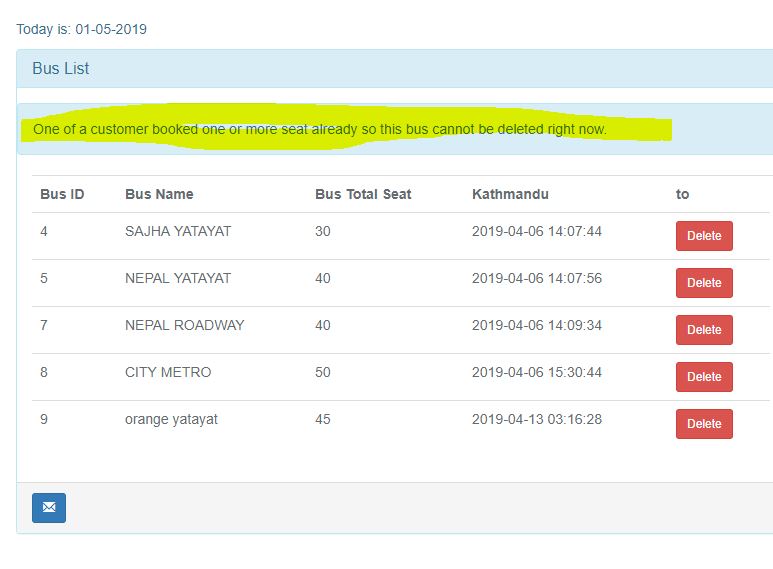
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Blackbox Test 1 Test Class: empty seat validation booking | | | | | |
| Data Source: Booking | | Objective: To check seat | | | |
| Test Case | Description | Tasks | Expected Result | Actual result | Conclusion |
| 1 | Entered without seat | Type wrong email format. | Error in seat | Validation error in seat | Pass |

****

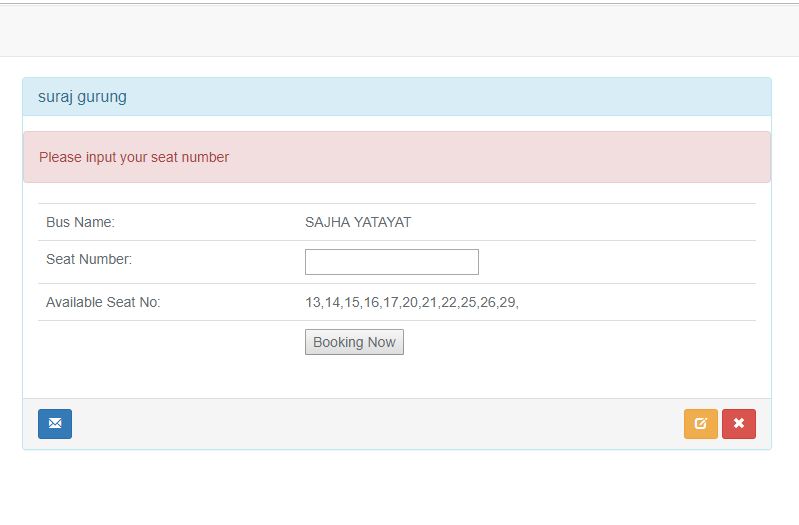
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Blackbox Test 2 Test Class: seat validation | | | | | |
| Data Source: Bus | | Objective: seat validation | | | |
| Test Case | Description | Tasks | Expected Result | Actual result | Conclusion |
| 2 | seat should be in correct format. | Type wrong seat | Error in seat form. | Validation error in seat | Pass |

****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Blackbox Test 3 Test Class: login Validation | | | | | |
| Data Source: User | | Objective: To check validation in login | | | |
| Test Case | Description | Tasks | Expected Result | Actual result | Conclusion |
| 3 | Email and password should not be empty | Not entered email and password | Password and email required | Validation error in email and password | Pass |

****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Blackbox Test 4 Test Class: bus deletion validation from list | | | | | |
| Data Source: bus | | Objective: To check validation in deleting ticket sold bus from list | | | |
| Test Case | Description | Tasks | Expected Result | Actual result | Conclusion |
| 4 | Ticket booked bus delete | Ticket sold bus want to delete | Error in deleting the bus | Please delete the seat not sold bus | Pass |

****

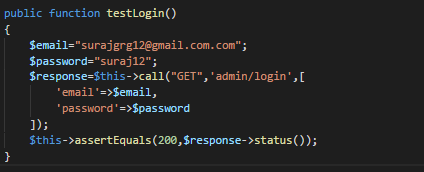
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Blackbox Test 5 Test Class: seat number validation | | | | | |
| Data Source: booking | | Objective: To check validation in booking seat number | | | |
| Test Case | Description | Tasks | Expected Result | Actual result | Conclusion |
| 5 | Without selecting seat number | .without selecting seat entered book now button | Error in booking seat | Please input seat number | Pass |

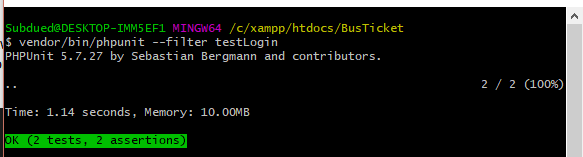
## Unit testing

Unit which means single or only one, so in terms of software term unit testing means software testing where each and individual units and components of the software tested.

1)

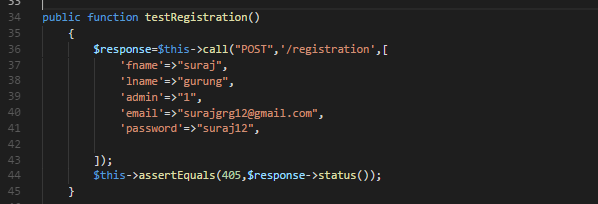
|  |  |
| --- | --- |
| Test Case | Whitebox test 1 |
| Purpose of test case | testlogin |
| Test data | [Surajgrg12@gamil.com](mailto:Surajgrg12@gamil.com), suraj12 |
| Class Name | User |
| Function name | Testlogin |
| Expected Result | Successful login |
| Actual Result | User successfully login |
| Outcome as expected | Yes |

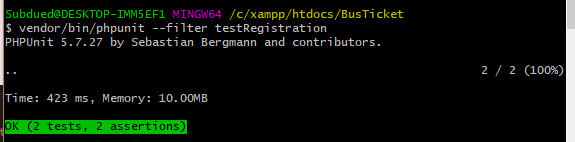
****

****

2)

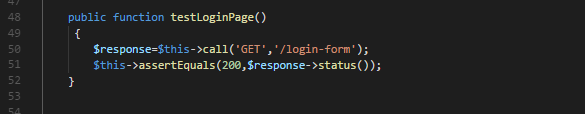
|  |  |
| --- | --- |
| Test Case | Whitebox test 2 |
| Purpose of test case | For checking registration |
| Test data | Suraj, guurung, 1, [surajgrg12@gmail.com](mailto:surajgrg12@gmail.com), suraj12 |
| Class Name | User |
| Function name | testRegistration |
| Expected Result | Successful registration |
| Actual Result | User successfully registered |
| Outcome as expected | Yes |

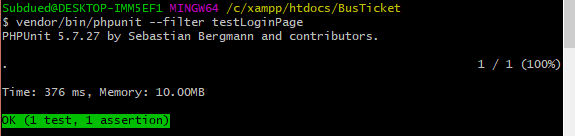




3)

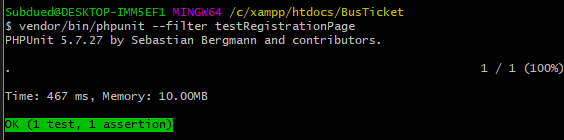
|  |  |
| --- | --- |
| Test Case | Whitebox test 3 |
| Purpose of test case | For checking loginpage |
| Test data | Get, login-form |
| Class Name | Usercontroller |
| Function name | Testlogin |
| Expected Result | Welcome dashboard |
| Actual Result | Welcome dashboard |
| Outcome as expected | Yes |





4)

|  |  |
| --- | --- |
| Test Case | Whitebox test 4 |
| Purpose of test case | For checking registration page |
| Test data | userdata |
| Class Name | UserRegistration |
| Function name | testRegistration |
| Expected Result | Successful registration |
| Actual Result | User successfully registered |
| Outcome as expected | Yes |



# Chapter6 Other project issues

## Risk management

It is the process of identifying, evaluating, and prioritizing the possible risk or present risk and to minimize, to monitor and to control the impact of that risk is called risk management.

Risk Likelihood values are shown as follows

|  |  |
| --- | --- |
| Likelihood | Value |
| Low | 1 |
| Medium | 2 |
| High | 3 |

Risk Consequence values are shown below

|  |  |
| --- | --- |
| Consequence | Value |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very High | 5 |

**Calculation of impact**

Here we have calculated all the impact that our software could encounter and effect. So we have calculated all possible the risk with the following method:-

Impact = likelihood \* consequences

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Likelihood** | **Consequences** | **Impact** | **Solution** |
| Changing Requirement | 3 | 5 | 15 | All requirements should be fixed initially. |
| Malicious | 2 | 2 | 4 | Up to date the antivirus and scan everyday if possible. |
| Hard Disk Crash | 2 | 4 | 8 | Data Backup every week. |
| Requirement Failure | 3 | 3 | 9 | Planning well and analysis intelligently. |
| Data Security | 3 | 4 | 12 | Identifying the users. |
| Phishing | 2 | 3 | 6 | Stop the unauthorized IP to the network. |
| Server failure | 3 | 5 | 15 | Weekly backup and regular maintenance of the devices. |
| Data modification | 2 | 3 | 6 | Firewall rules Access control must be implemented. |
| Spyware | 2 | 3 | 6 | Security mechanism must be implementing. |
| Spam | 1 | 2 | 2 | Traffic filtering and blocking the unnecessary entry. |

Figure 7: Table showing Risk Management

## Configuration Management

It is a system in which the process of engineering occurs for establishing and maintaining consistency of a product’s performance, functional, and physical attributes with its all requirements and design. Or it is also a task of controlling and tracking the changes in the software.

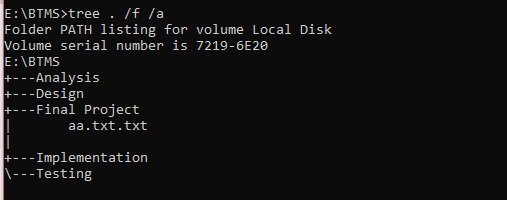
****

Figure configuration management

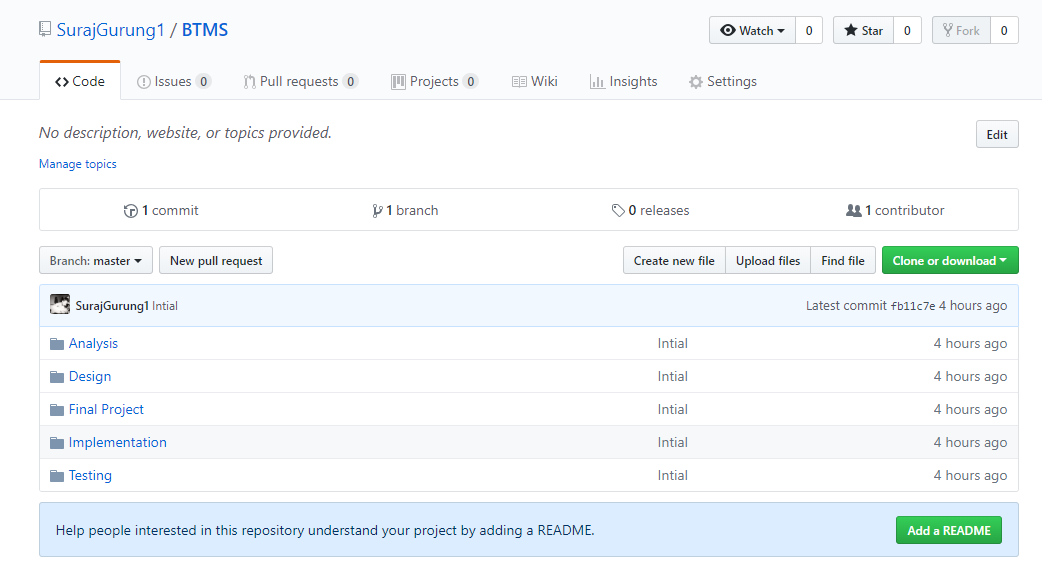
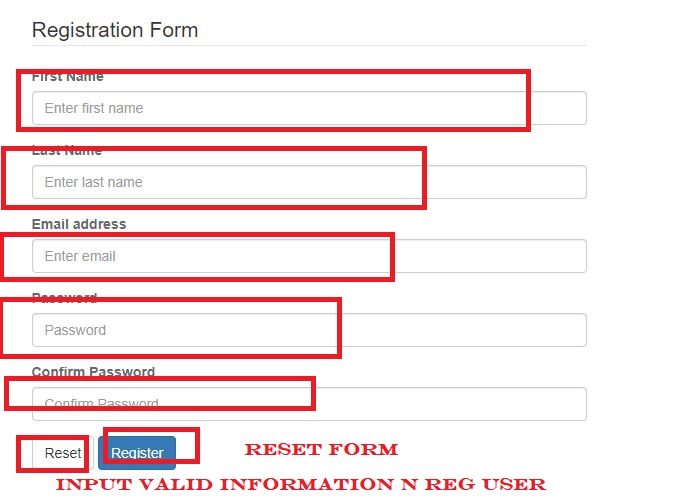


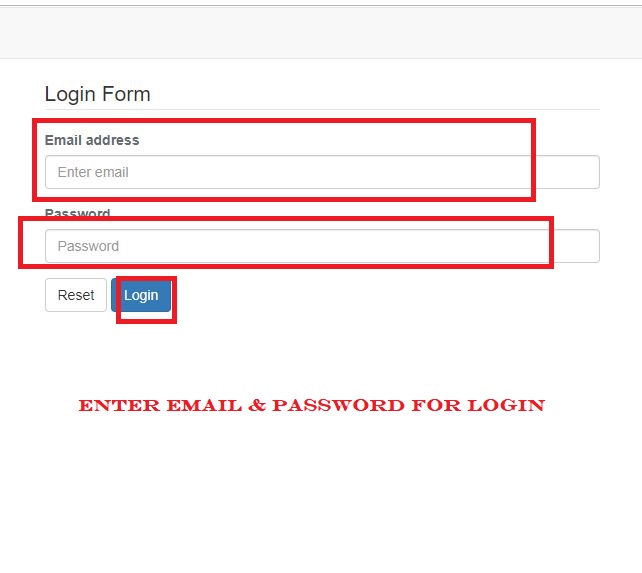
Figure Github folder

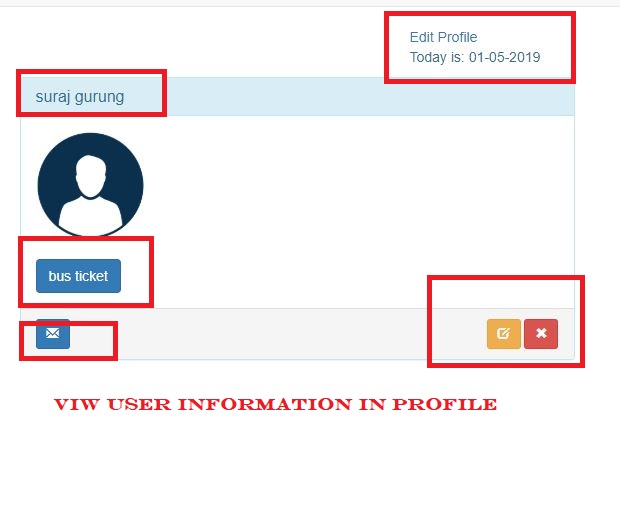
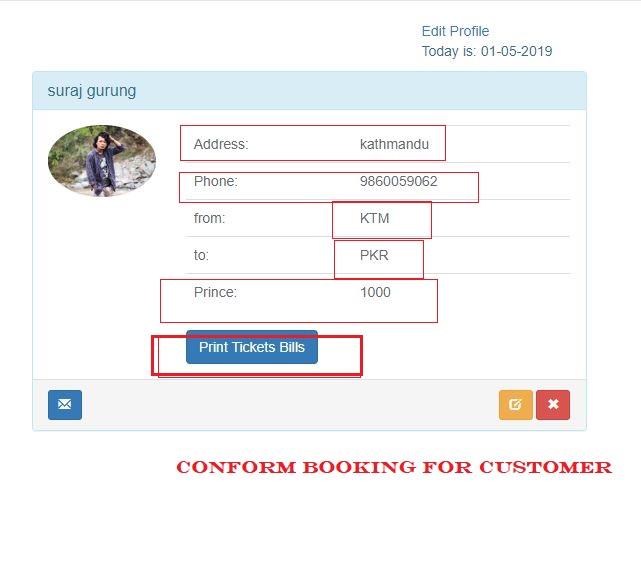
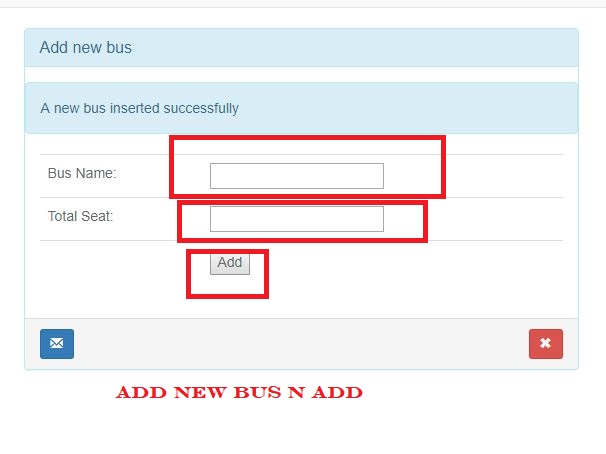
## Future work

This system is a efficient system for its goals however there are many areas where have to improve for the best results in bus ticketing management system. The things that’s going to be done in future like hosting this site with a domain registration so users can access this site via internet. In future with investment this system can be upgraded towards modern form. It will achieve company goals. This system can be expanded in large area. Any individual can have this system for booking ticket. By the help of financial institution payment through web site can be done. Advance reservation of seat can be done. it will have a linking application for this software. From the application users can download from internet and can access all the operation of this system. Data base size can be upgraded to it’s maximum size.

## User manualwelcome page - Copy.JPG







## Limitation

After the development of this project we have suffer with many logical errors as well as many expected things can not achieved from us. We thought to give online payment system but somehow we are unable to implement this time. At the time of analysis i want to give system at home or office, unfortunately due to no web hosting in domain registration. Users are unable to access the site. So this are the major limitation for this system.

# Chapter 7

## Conclusion

The project has been completed as per scheduled. In this stages of software development firstly, I introduced my system aims and objectives. Secondly, I scheduled my all task by allocating the enough time for each task to accomplish its aims and objectives. After this, analysis was started. On this, I made a rich picture from which it illustrates all the operations of the operators as well as use cases were drawn, requirements specification done, prioritization to the requirements, architecture of the system and initial class diagram was made. Then I started my design process, I made final class diagram, activity diagram, sequence diagram, database design base ER diagram was drawn and data dictionary listed. As soon as possible, after completing my design phase I started coding and implementation of the system was completed. I have detail described about the information related with implementation on this doc. Then I started my testing like black box testing and white box testing on the top of the unit testing. I checked all the quality of the software to ensure whether the system meets its requirements or not. It accomplish its a lot of goals which is the positive vibes for me. Also I documented all the results in this document. Finally, my Bus Ticketing Management System was successfully executed,

# Chapter 8

# References

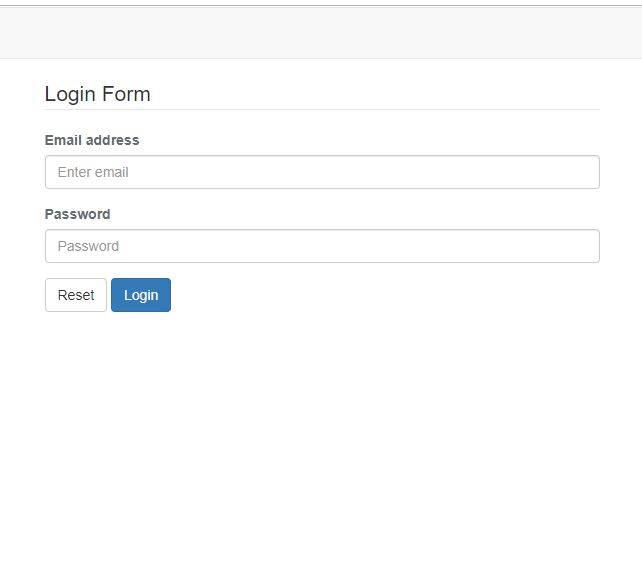
(2019, 02 16). Retrieved from <https://laravel.com/docs/4.2/introduction>

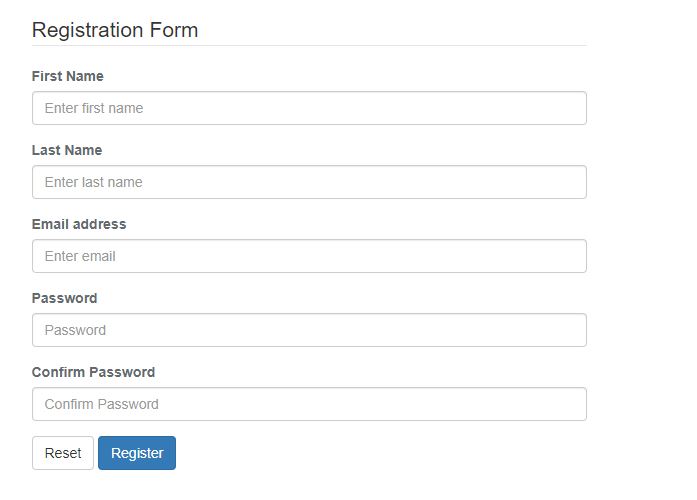
(2019, 02 22). Retrieved from <https://www.techopedia.com/definition/18822/design-pattern>

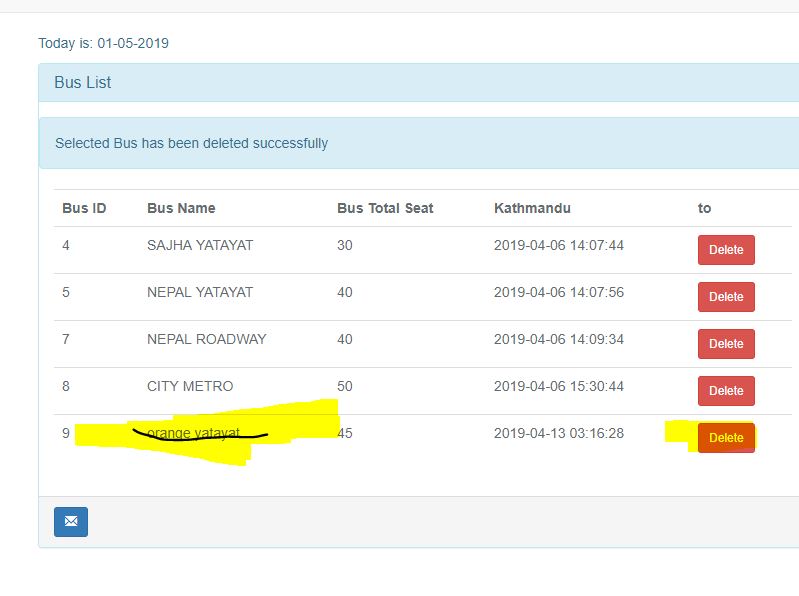
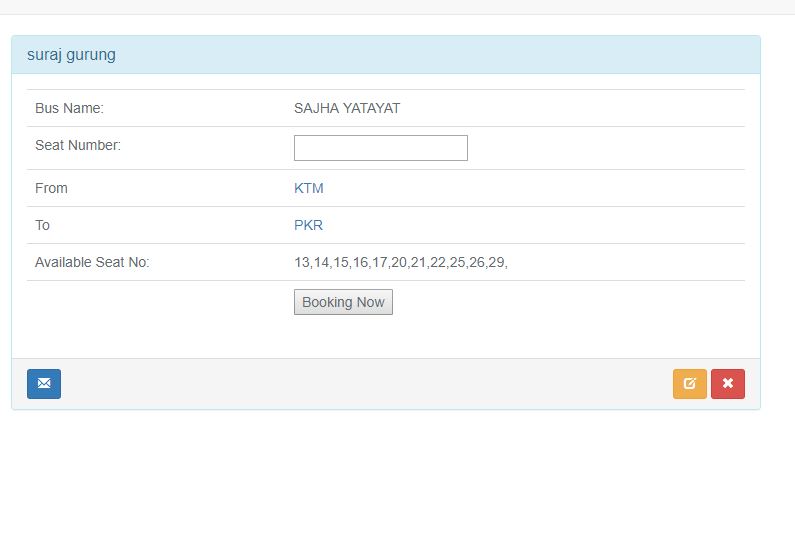
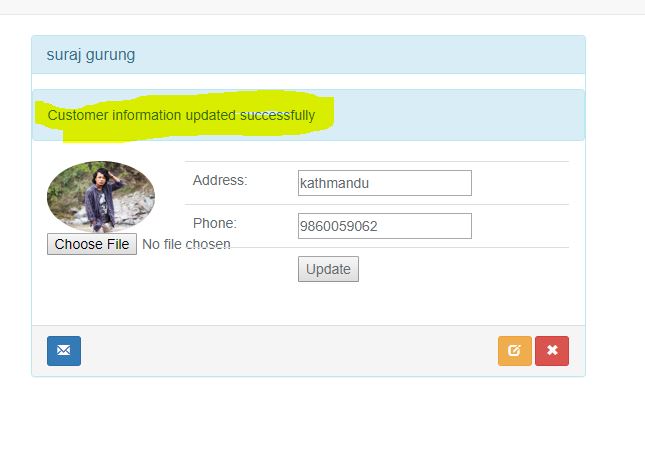
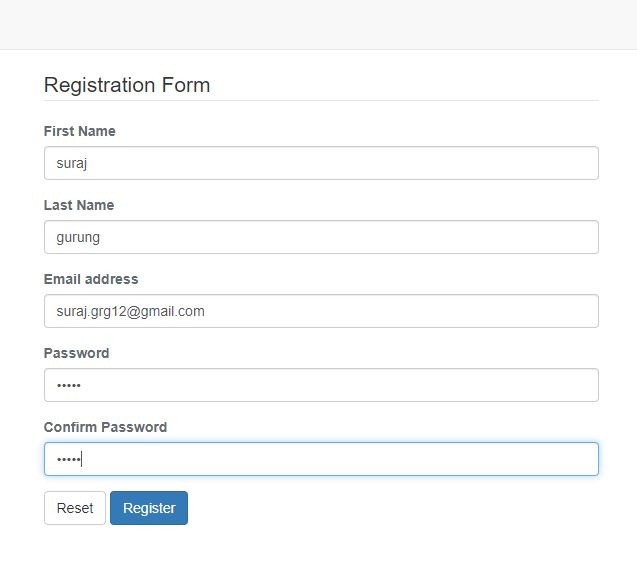
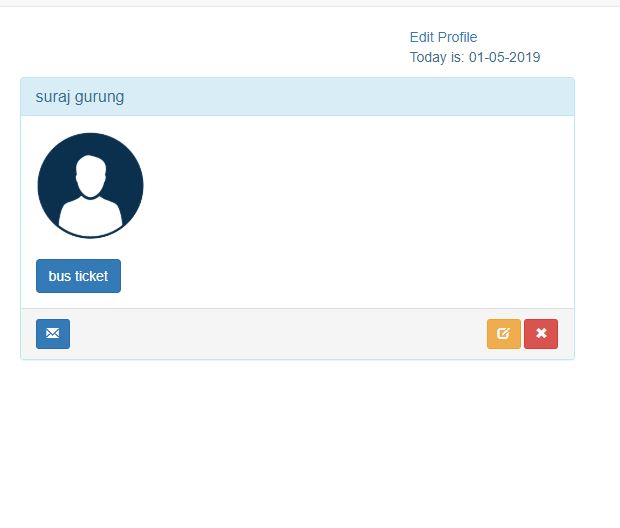
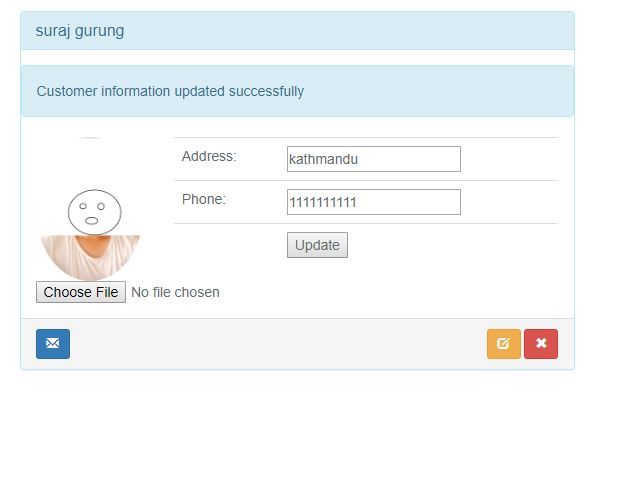
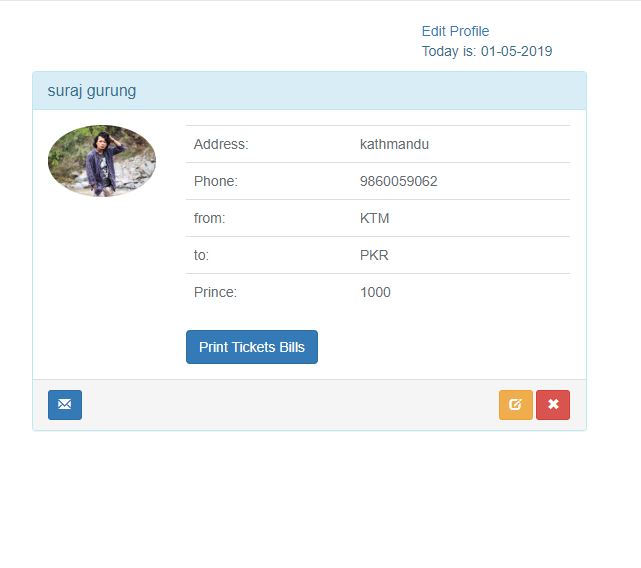
(2019, 02 25). Retrieved from <https://www.smartdraw.com/entity-relationship-diagram/>

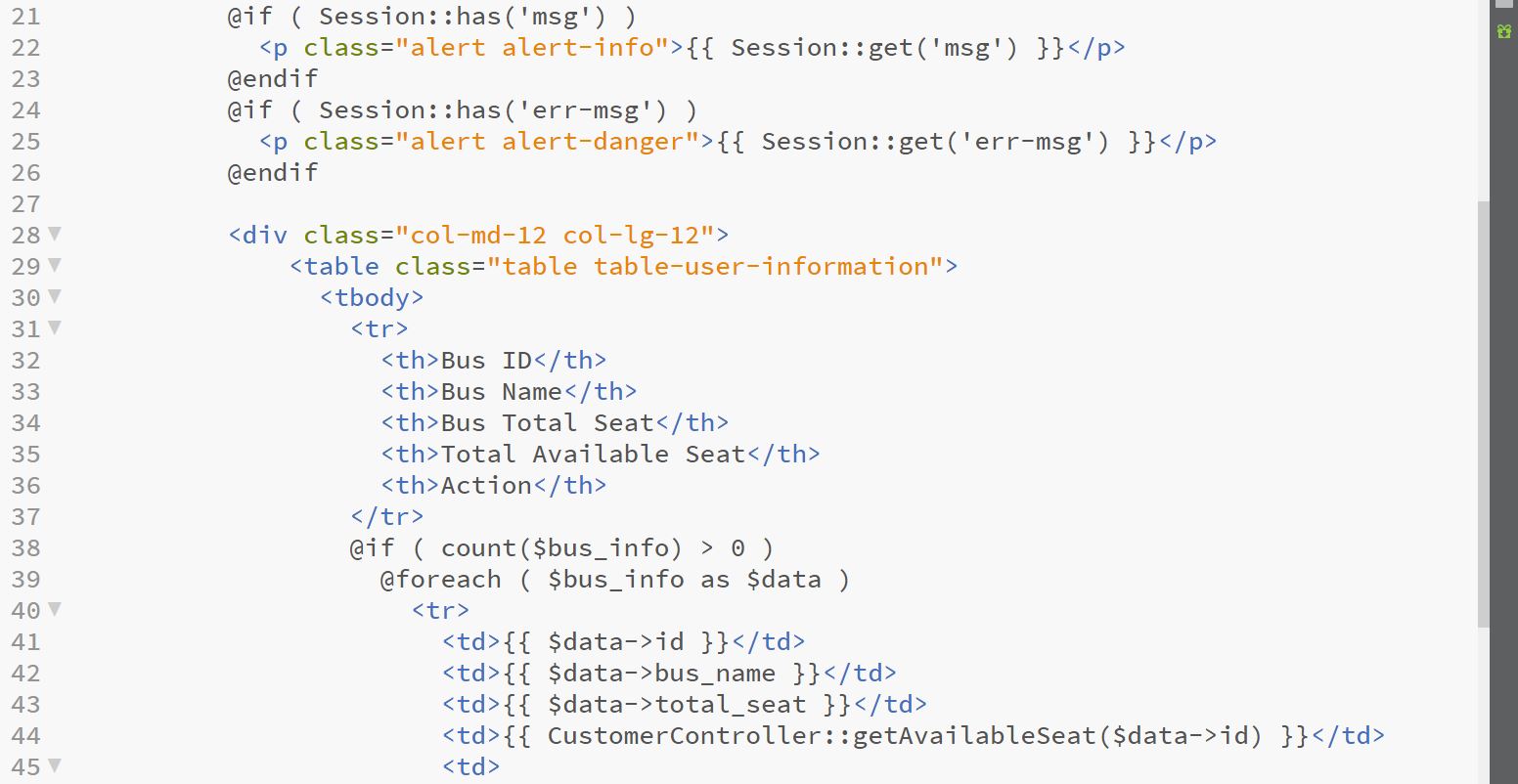
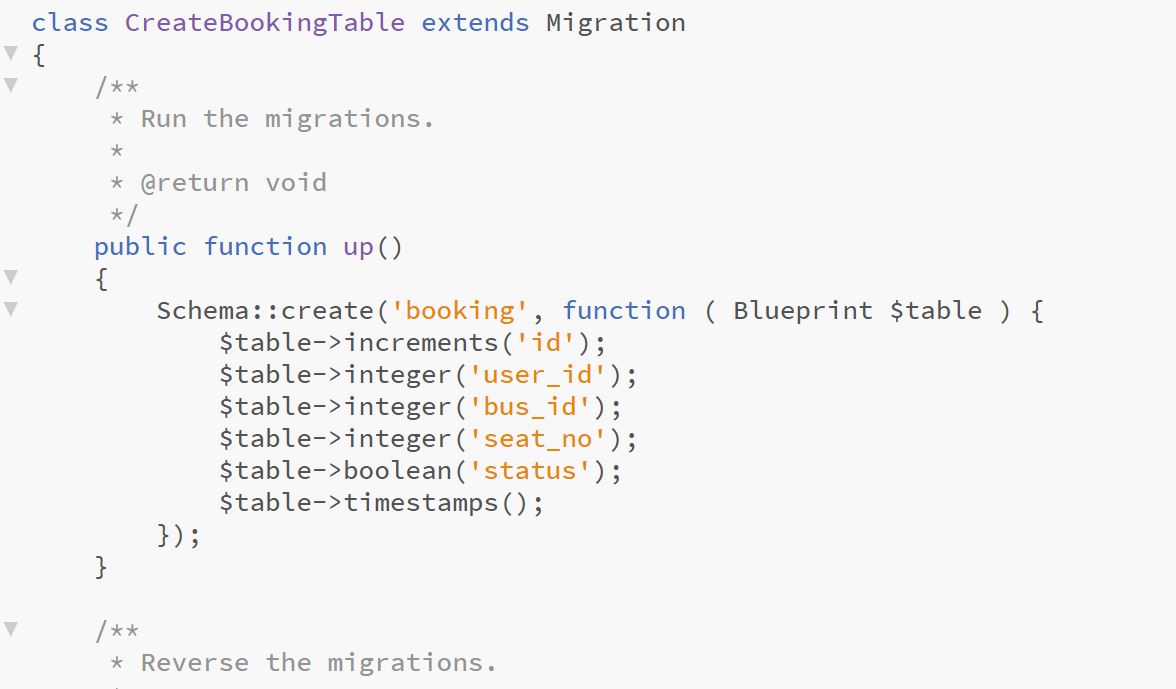
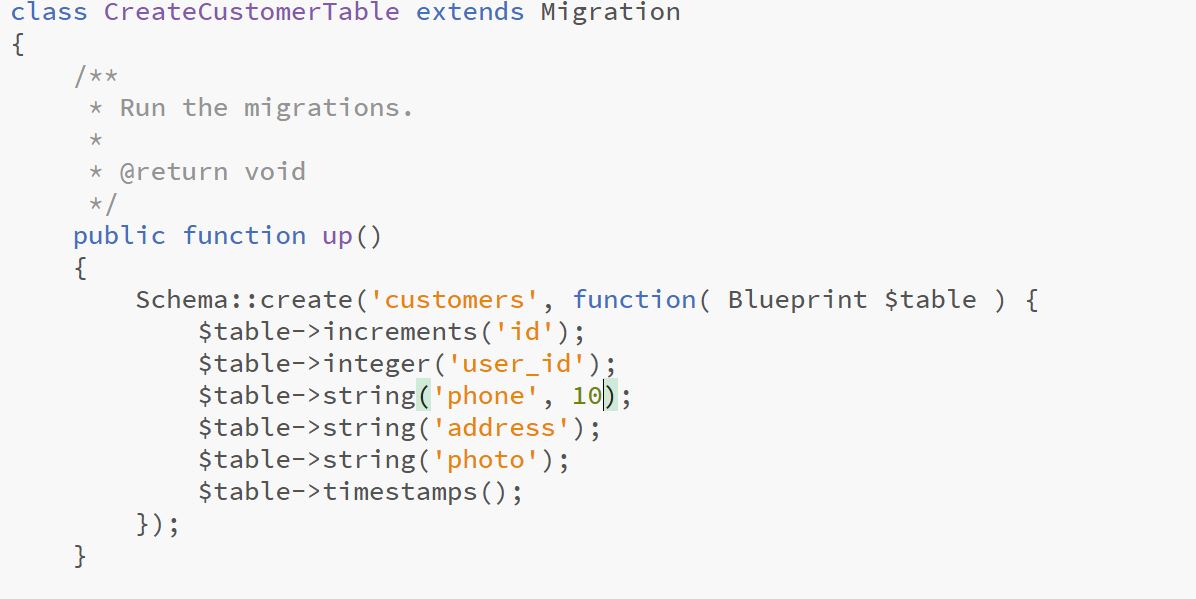
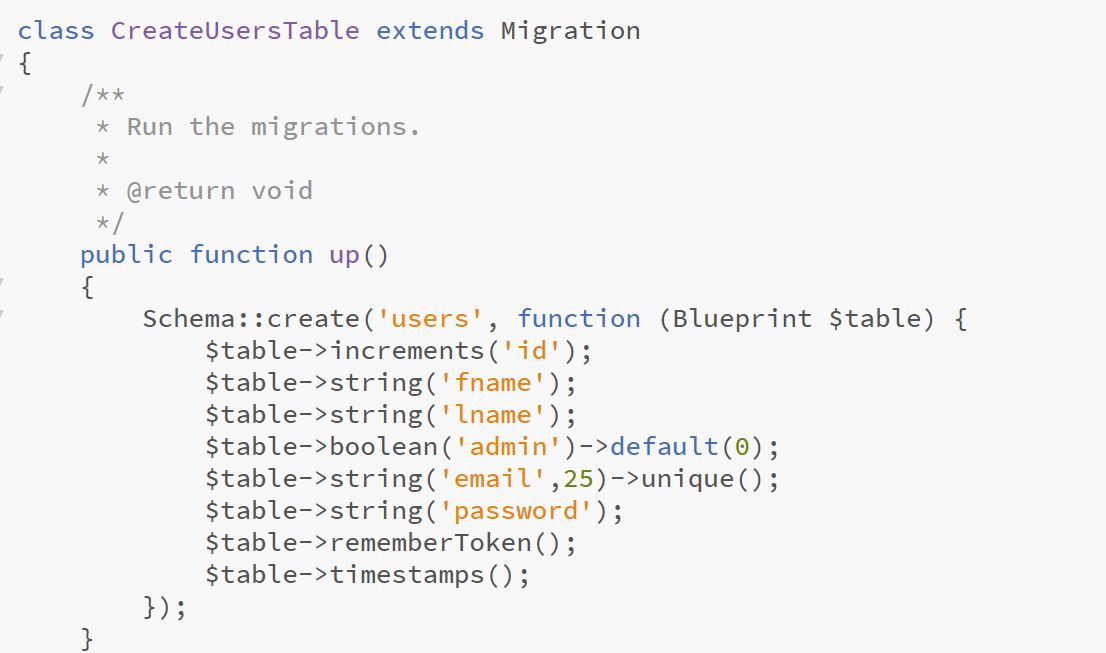
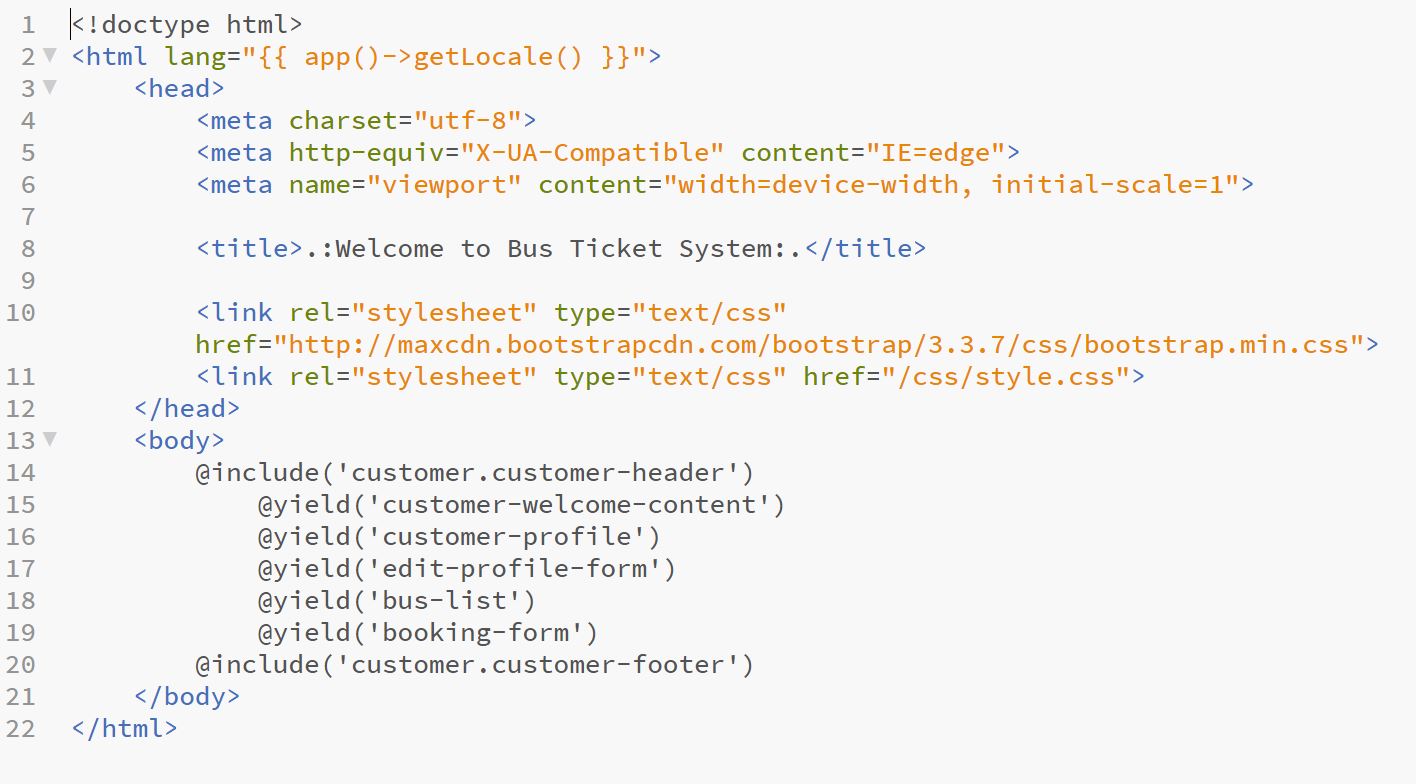
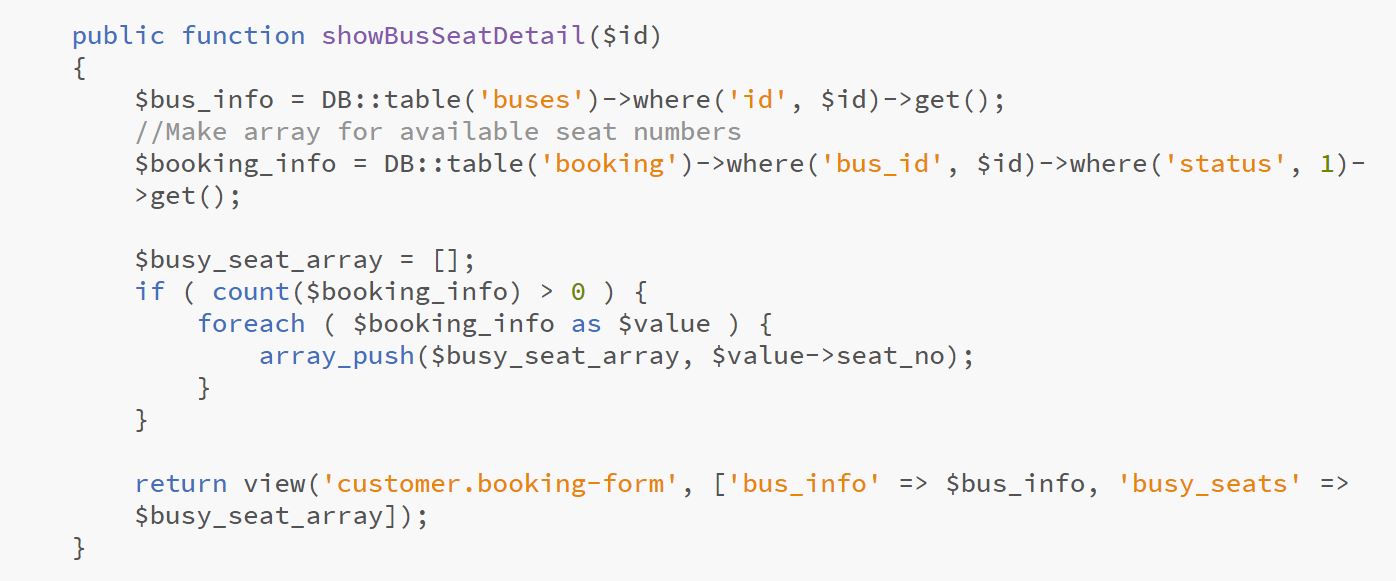
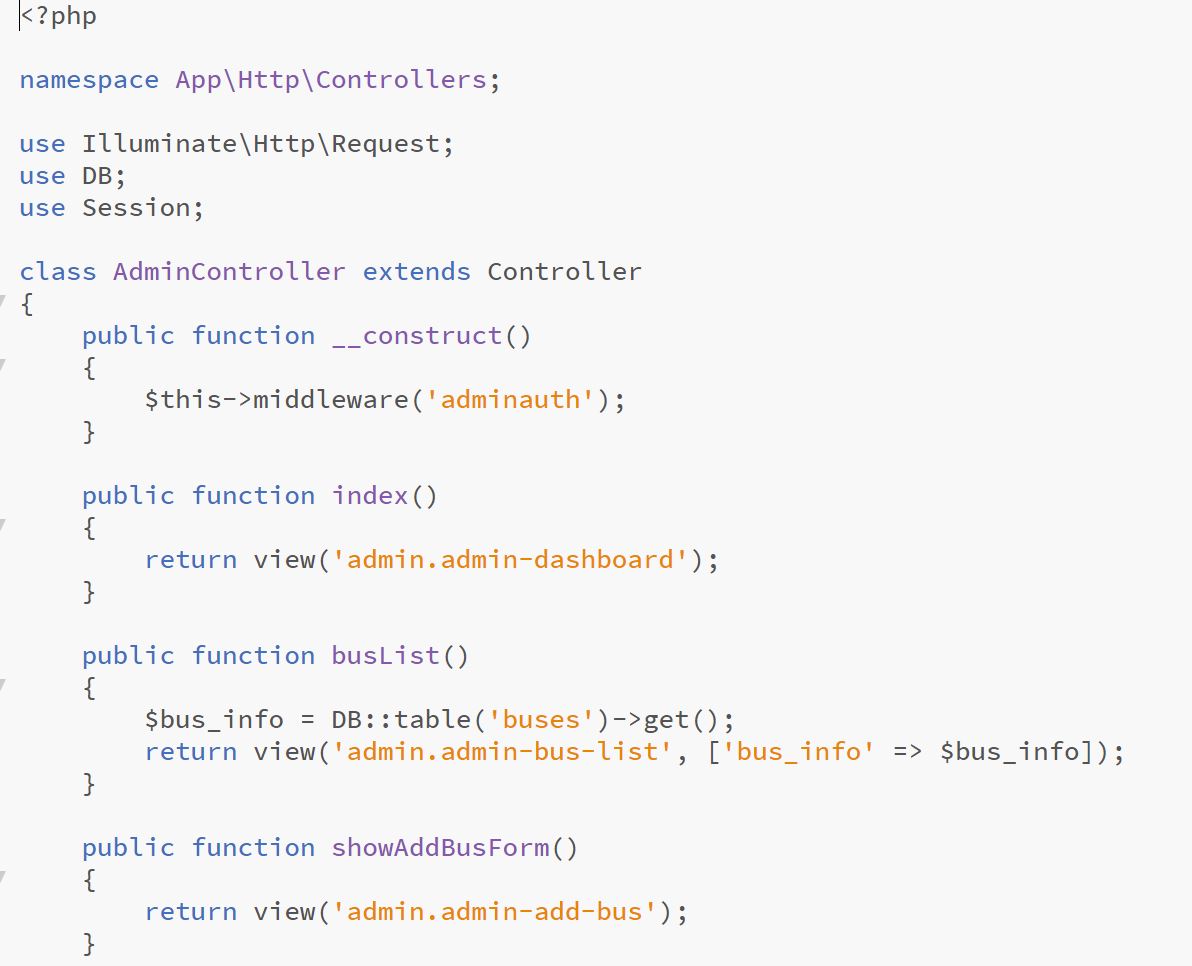
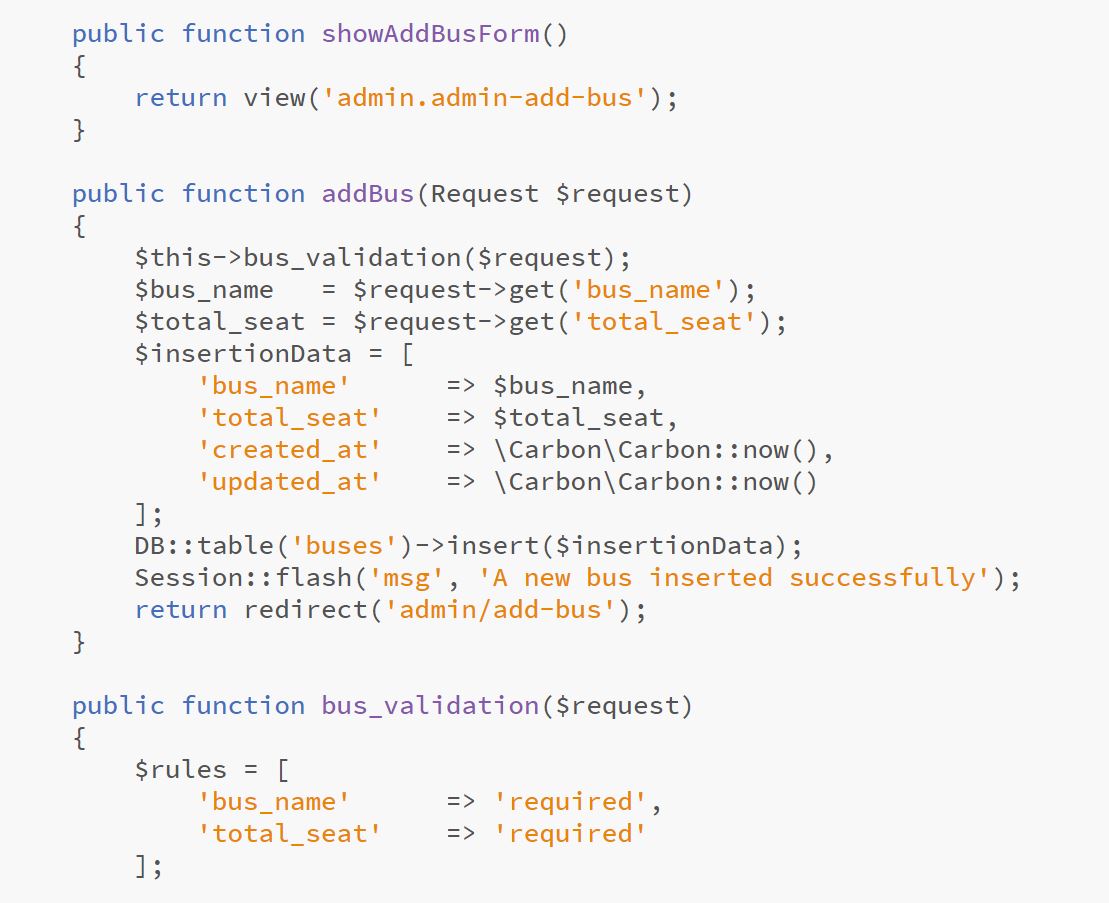
(2019, 02 26). Retrieved from https://www.lucidchart.com/pages/landing/activity-diagram

# Chapter 9 Appendix (CODE + UI screenshot)welcome page.JPG









# Reference