**Contents**

[**Chapter 1: Introduction** 2](#_Toc4879029)

[**1.1** **Introduction** 2](#_Toc4879030)

[**1.2** **Background of the project** 2](#_Toc4879031)

[**1.3** **Problem statement** 2](#_Toc4879032)

[**1.4 Description of the project** 2](#_Toc4879033)

[**1.4.1 Features of the project** 2](#_Toc4879034)

[**1.5 Overview of the project** 2](#_Toc4879035)

[**Chapter 2: Scope of the project** 2](#_Toc4879036)

[**2.1 Scope** 2](#_Toc4879037)

[**2.2 Limitation** 2](#_Toc4879038)

[**2.3 Aim** 2](#_Toc4879039)

[**2.4 Objectives** 2](#_Toc4879040)

[**2.5 Overview of the scope** 3](#_Toc4879041)

[**Chapter 3: Development methodology** 3](#_Toc4879042)

[**3.1 Description of the methodology chosen** 3](#_Toc4879043)

[**3.2 Design pattern** 4](#_Toc4879044)

[**3.3 Architecture** 4](#_Toc4879045)

[**Chapter 4: Project planning** 5](#_Toc4879046)

[**Chapter 5: Risk management** 5](#_Toc4879047)

[**Chapter 6: Configuration management** 5](#_Toc4879048)

[**Chapter 7: Conclusion** 5](#_Toc4879049)

[**Chapter 8: Reference and Bibliography** 5](#_Toc4879050)

# **Chapter 1: Introduction**

## **Introduction**

Bus ticket booking system is a web-based application where people can book seats easily with less effort and time. This system is built for managing and computerizing the traditional database, ticket booking and travel made. It maintains all the customer details, bus details and booking (reservation) details.

## **Background of the project**

## **Problem statement**

At the present time, the system used in the counter is an internal system which is manually used in the selling the tickets. The main problem with the current system is customer have to go to the counter and ask for the tickets which is very time-consuming. Sometimes, they have to queue up for long time and have to pay cash while buying tickets.

## **1.4 Description of the project**

### **1.4.1 Features of the project**

The main features of Bus ticket booking system are as follows:

* Online booking of ticket
* Online payment

## **1.5 Overview of the project**

# **Chapter 2: Scope of the project**

## **2.1 Scope**

The scope of this system is people can book/reserve seats online. They don't have to go to counter in order to buy tickets and can do online payment instead of cash. Booking can be done anytime, from anywhere.

## **2.2 Limitation**

The limitation of this system is there might be problem while cancelling the reserved seats. This function is only available to the company staffs. So, for cancelling the ticket, customers have to communicate with the staffs. Currently, this service is available only in computer not in the smartphones.

## **2.3 Aim**

* To design the web-based application for booking the bus tickets.
* To design system which is user-friendly, time and effort saving.

## **2.4 Objectives**

* Observation techniques is performed during analysis phase to get better view of what people want.
* Scheduling is done in order to finish project in time.
* To make it user-friendly, good and simple GUI is designed.
* Testing is performed to find out bugs in the system.
* Documentation of the project is done for the future reference.

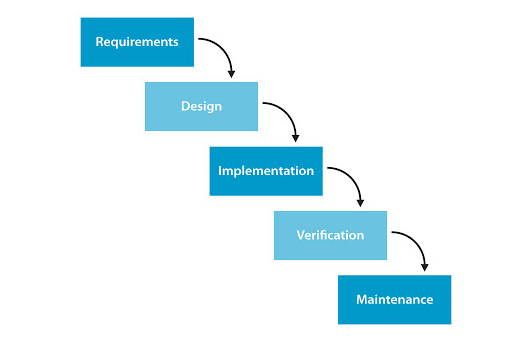
## **2.5 Overview of the scope**

# **Chapter 3: Development methodology**

## **3.1 Description of the methodology chosen**

Among the many SDLC methodology, I have decided to use Waterfall model. It is a linear, sequential and traditional approach to the software development life cycle. It consists of different stages and each stage are performed serially.

**Stages of the waterfall model**:



I have chosen Waterfall model because of the following reasons:

* It is simple and easy to understand and use.
* Each stage is performed serially. So, no delay in development of the system.
* It facilities in scheduling.
* It is easy to arrange the tasks.
* Process and results are well documented.

## **3.2 Design pattern**

I will use MVC pattern for code architecture. It stands for "Model-View-Controller." It is a software architecture pattern, commonly used to implement user interfaces Below is a description of each aspect of MVC:

* **Model** - Model represents an object. It can also have logic to update controller if its data changes.
* **View** - View represents the visualization of the data that model contains.
* **Controller** - Controller acts on both model and view. It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate.

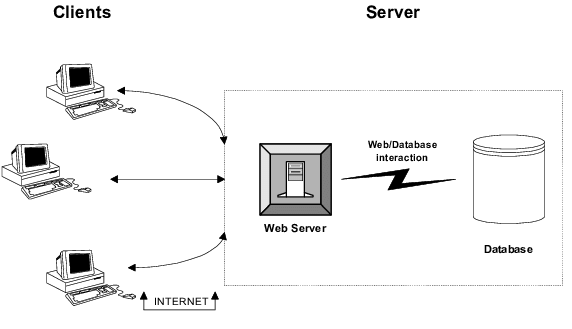


The reasons behind choosing MVC design pattern are as follows:

* Save time and effective use of resources
* Rapid application development
* Facilitates the multiple view
* The Modification Never Affects The Entire Mode

## **3.3 Architecture**

I will use client-server architecture for this project. In this architecture, Clients request services from the server and the server provides relevant services to those clients.



The advantages of this architecture are as follows:

* All the data are stored in the central location.
* Users can access shared data which is centrally controlled.
* Backups and network security is controlled centrally.

# **Chapter 4: Project planning**

# **Chapter 5: Risk management**

# **Chapter 6: Configuration management**

# **Chapter 7: Conclusion**

# **Chapter 8: Reference and Bibliography**