Table of Contents

[**Chapter-2: Analysis.** 2](#_Toc15578530)

[1.1: Analysis Introduction: 2](#_Toc15578531)

[1.2: Analysis Methodology: 2](#_Toc15578532)

[1.3: Feasibility Study: 4](#_Toc15578533)

[1.4: Information Gathering System. 5](#_Toc15578534)

[1.5: MoSCoW Prioritization: 6](#_Toc15578535)

[1.6: Software Requirement Specification (SRS): 6](#_Toc15578536)

[1.6.1: Functional Requirements:- 6](#_Toc15578537)

[1.6.2: Non- Functional Requirements:- 7](#_Toc15578538)

[1.6.3: Interface Requirement: 8](#_Toc15578539)

[1.7: System Architecture: 9](#_Toc15578540)

[1.7.1: Natural Language Analysis (NLA): 10](#_Toc15578541)

[1.7.2: Class Diagram: 11](#_Toc15578542)

[1.7.3: Use Case Diagram: 12](#_Toc15578543)

# **Chapter-2: Analysis.**

## 1.1: Analysis Introduction:

Analysis is the method of breakdown of the big complicated topic or substance into tiny parts to solve the problem easily and better understanding of the big complicated topic. It helps to use the system to be developed. It is used to gather all the requirements that would be beneficial for the development of the Bus Ticket Management System. It will help to achieve inter compatibility and unity of the sub system. Analysis play vital role in the failure and success of any project. If analysis is done properly then the ultimately the project will met its goal.

I have used the MoSCoW Prioritization for the filtration of Functional and Non- Functional requirement that I have analysed in the Analysis Phase of the project.

## 1.2: Analysis Methodology:

Analysis Methodology refers to the steps taken when collecting information, analysing information and the documenting the requirements. It offers a structure to follow when working on the project and also makes the analysis and design more manageable.

There are the several methodologies available for the analysis and development of the software that have their own prospective. Some of them are briefly described below:-

* **Hard Systems Methodology: -** It focuses on the technical requirements of the system. It is highly structured approach to the analysis and it follows a logical sequence of steps and obey the rules, guidelines and standards.
* **Soft Systems Methodology: -** It focuses on people rather than technical aspect. It is focused on the user interaction and user requirements.
* **Combined Methodology: -** It focuses on the both the combination of technical and user requirements. It offers the flexibility in the approach.
* **Organization-Oriented Methodology: -** It focuses on the aims, features and technical aspects of an organisation. It ensures through planning and scheduling.
* **People-Oriented Methodology: -** It refers to the soft approach to the analysis such as SSM and it focuses on the user interaction and user requirements.
* **Process-Oriented Methodology: -** It is hard approach to analysis and highly-structured, step-by-step methodology which follows the strict rules and guidelines, as defined in SDLC. It focuses on the flow, use and transformation of data.
* **Object-Oriented Methodology: -** It focuses on the relationships between each aspect of the system.

In this project, I will use the hard approach methodology. After analysing all the above mentioned methodology, Hard System Methodology suits to my project as it follows a logical sequence of steps and strictly obey the rules, guidelines and standards.It uses the three views of the system and they are process view, data view and event view. It also examines and determines that the project is technically, financially and socially feasible or not. This methodology is also suitable for small project like ours. That’s why I think using Hard System Methodology is the right choice for my project.

The advantages of using Hard System Methodology in our project are as follows:-

* It ensures through planning and scheduling that will ultimately save our cost and time.

* It strictly follows the waterfall model which is easy to understand and easy to use it.
* It strictly obey the rules, guidelines and standards of the company.

There are also the disadvantages of using Hard System Methodology in our project and they are listed below:-

* There might be lack of flexibility, e.g. if the requirements are not specified correctly or changed later in project, and it may not be possible to return to previous stage.

* There might be limited user involvement as this method is focused on technical requirements.
* The Project might often take longer to deliver than other methods that allows stages in project to be repeated.

Data Flow Diagram (DFD) is the techniques that can used in Hard System Methodology. It is briefly described below:-

**Data Flow Diagram (DFD): -** Data Flow Diagram is method that is used for the graphical representation of the flow of the data in a business information system. DFD represent the process which are involved in a system to transfer data from input to the file storage and report generation.

DFD visual representation makes the good communication between system designer and user. It also illustrate the boundary of the system. It is simple to construct the DFD and easy to understand. DFD is used to broke down complex system into smaller diagram i.e., sun-processes.

The DFD for my Project is shown below:-

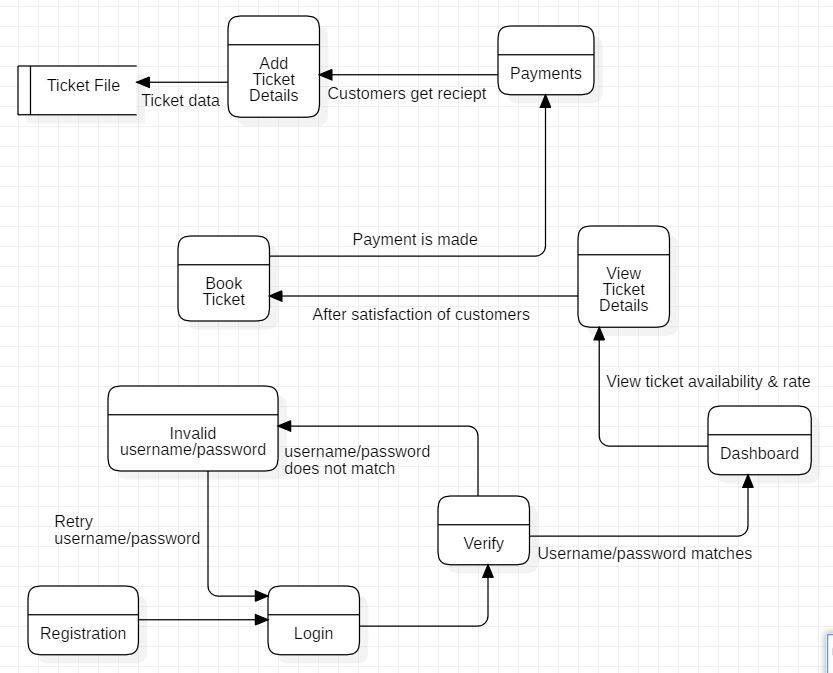


Figure 1: DFD for Bus Ticket Management System

## 1.3: Feasibility Study:

Feasibility study is the initial design stage of any project, that brings together the elements of the knowledge which states that the project is possible or not. It includes an estimation of the level of expertise required for a project and who will provide it. It takes all the project relevant factors like economic, technical, legal, and scheduling. Feasibility study provides a company management with critical information that can prevent the company from entering blindly into risky businesses.

The types of the feasibility study that could be done in our project are described bellows:-

1. **Technical Feasibility: -** It states that the whether the company have enough technological resources to undertake the project. It also states that if the processes and procedures are helpful to project success.

1. **Schedule Feasibility: -** It states that the whether the company currently have the time resources to undertake the project. It also estimate that will the project be completed in the available time.
2. **Economic Feasibility: -** It states that the whether the financial resources of company are enough to complete the project. It is commonly known as the cost benefit analysis.
3. **Operational Feasibility: -** It states that how well your company will be able to solve problems and take advantages of the opportunities that are presented during the course of the project.
4. **Legal/Ethical Feasibility: -** It states that legal implication of the project. We have to make sure that our project undertaken will meet all legal and ethical requirements before starting of project.

## 1.4: Information Gathering System.

Information gathering is the method of collection of the all the related information that could be useful for the project. The purpose of information gathering is to enhance the planning of our project work to become more fully inclusive. It helps to make good communication between designer and user. It help to make the system more usability and user friendly.

There are several methods for the information gathering and they are briefly described below: -

1. **Interview: -** An interview is a conversation where question are asked and answer are given. It is a one-to-one conversation between an interviewer and interviewee. Interview between developer and user is very important for the development of good and successful software.

1. **Questionnaire: -** It provides the most speedy and simple technique for gathering information about group of individuals scattered in a wide and extended field. A properly produced and administered questionnaire may serve as a most appropriate and useful information gathering device.
2. **Observation: -** It is most significant and common technique for data and information collection. Observation is the method in which one or more persons observe what is occurring in some real life situation. It dals with external behavioural of persons in appropriate situation.

## 1.5: MoSCoW Prioritization:

MoSCoW Prioritization is process of setting requirement by order of priority. The most vital requirement need to be met first for a greater chance of success. The MoSCoW is an acronym made up of the first letters. The two Os don’t have any meaning themselves they are added to make word ‘Moscow’ readable. The M stands for ‘Must haves’, S for ‘Should haves’, C for ‘Could haves’ and W for ‘Would haves’ or ‘Won’t haves’.

**Must haves: -** They are necessity for a workable product and there is no alternative. The ‘Must haves’ are essential. Without meeting these requirements, the project fails and product won’t be use-able.

**Should haves: -** They are additional and much desired requirements that have a high priority, but not essential for usable end product. The product will be usable even if these requirements aren’t met.

**Could haves: -** These requirements can be considered if there’s time left. If not, it’s no problem and will not have negative impact on final result. The ‘Could haves’ have a lower priority than the ‘Should haves’.

**Won’t haves (and would haves): -** These ae about wishes for the future that are often impossible to realise or cost a lot of time. If it’s simply not possible, it’s best not to waste any energy on it.

## 1.6: Software Requirement Specification (SRS):

A Software Requirement Specification (SRS) is a document which states that what the software will do and how it will be expected to perform. It includes a purpose, an overall description and specific requirements. It is used to provide vital information to multiple teams for quality assurance, operation and maintenance. It defines how the software will interact when connected to other software. There are two types software requirements and they are functional requirements and non-functional requirements. It will states that what the Bus Ticket Management System will do and how it will be expected to perform.

### 1.6.1: Functional Requirements:-

Functional Requirements is defined as the functionality of the system. It will states the functionality in the Bus Ticket Management System and what it will supposed to do. In this project we will perform several functional requirements.

The Functional Requirements for our project Bus Ticket Management System are stated below in MoSCoW prioritization table:-

**Functional Requirements with MoSCoW Prioritization:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FID** | **Requirements** | **MoSCoW** | **Description** | **Dependencies** |
| F1 | User Registration | Must have | The user must have to registered in order to access to the software | Independent |
| F2 | Login | Must have | Login is need to verify and validate the users. | F1 |
| F3 | Insert Bus | Must have | New Bus are to be inserted by Admin only to provide the available tickets for passengers. | F2 |
| F4 | Update Bus | Must have | In case of the unavailable of the reserved bus, it should be replaced by new one by admin | F3 |
| F5 | Delete Bus | Should have | Admin should be able to delete the unavailable bus. | F3 |
| F6 | CRUD function | Should have | User should be able to insert, update, delete and view their data and information. | F2 |
| F7 | Dashboard | Should have | There should be dashboard available after the user authentication is verified. | Independent |
| F8 | View Ticket Rate | Could have | The user should be able to see the rate of the ticket of their destination. | F1 |
| F9 | Generate receipt of ticket | Must have | The user are provided with ticket receipt after they have booked the ticket. | F2,F7 |
| F10 | Forget Password | Could have | User sometimes forget their passwords and in such case forget password would be useful. It will increase user satisfaction | F6 |

### 1.6.2: Non- Functional Requirements:-

Non- Functional requirements states the system attributes such as security, performance, maintainability, reliability, scalability and usability. It is also known as system qualities. They ensures the usability and effectiveness of entire system. The proper definition and implementation of non-functional requirements is critical.

The Non- Functional Requirements for our project Bus Ticket Management System are stated below in MoSCoW prioritization table:-

**Non- Functional Requirements with MoSCoW Prioritization:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **NFID** | **Requirements** | **MoSCoW** | **Description** |
| NF1 | Performance | Should have | The system should be quick and no buffering that will increase the performance of the software. There are various ways to increase the performance. |
| NF2 | Availability | Must have | The system should be available when the user want and system should be available regularly. |
| NF3 | Security | Must have | The user account, data and action performed should be provide a security and ensure them that their account, and action are safely stored. |
| NF4 | Reliability | Should have | The system used by user should be reliable after performing the any activities for many times. |
| NF5 | Maintainability | Could have | The system should be maintained in time and should be easy for maintenance. System should work properly after maintenance. |
| NF6 | Reusability | Could have | The code should be reusable and has to enhance the better performance. It should reduce redundancy. |
| NF7 | Supportability | Should have | The system should be technically supportive to the personnel to install, configure and monitor the system. |
| NF8 | Data Integrity | Could have | The system should have to maintain and assure the accuracy and consistency of data flow into the system. |
| NF9 | Safety | Could have | Human are error prone and user have to use system safely and try to limit the error. |

### 1.6.3: Interface Requirement:

1. **User Interface: -** The user interface is everything designed into an information device with which a person may interact. Any administration can operate the system.
2. **Hardware Interface: -**

* Operation system i.e. window.
* RAM:256 MB
* Hard-Disk: 30 GB
* Processor: Pentium® Dual-core CPU.

1. **Software Interface: -**

* Visual Studio 2017
* My SQL Database Server 2014
* C# Language

## 1.7: System Architecture:

The system architecture that I will using in Bus Ticket Management System is 3-tier architecture. A 3-tier architecture is a type of software architecture which is composed of three “tiers” or “layers” of logical computing. They are often used in applications as specific type of client- server system.

The three tier of 3-tier architecture are descried briefly below:-

* **Presentation Tier: -** The presentation tier is the front end layer in the 3-tier system and consists of the user interface. This tier communicates with other tiers by sending results to the database.
* **Application Tier: -** The application tier contains the functional business logic which drives an applications more capabilities. It controls application functionality by performing detailed processing.
* **Data Tier: -** The data tier comprises of database/data storage system and data access layer. Examples are MySQL, Oracle, and Microsoft SQL Server etc. Data in this tier is kept independent of application servers or business logic.

The figure of 3-tier architecture is shown in below:-

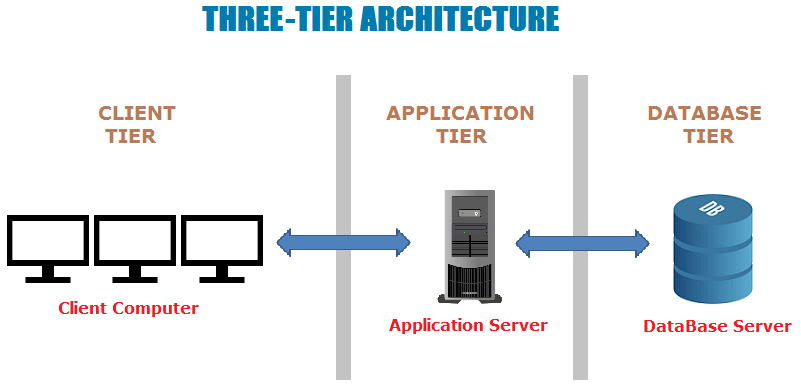


Figure 2: System Architecture

### 1.7.1: Natural Language Analysis (NLA):

Bus Ticket Management System is one of the online projects where customers can book their tickets by logging into the software. The customers could register their account easily. After login to the system, customers visit dashboard where they could view the available buses and their destination with available tickets. The system should be made in such a way that customers could get a receipt after booking and payments. Similarly, they could also edit their details.

Likewise, the admin could also insert new buses and could remove or replace those buses which are not in use. The admin should update the details about the system and the daily news.

|  |  |
| --- | --- |
| **Nouns** | **Verbs** |
| Bus, Ticket, System, projects, customers, software, account, dashboard, receipt, payments, admin, news, book. | Is, can, login, register, visit, view, destination, get, made, edit, insert, remove, replace, update. |

### 1.7.2: Class Diagram:

A class diagram is the kind of diagram and part of a UML (Unified Modelling Language) which provides and defines the overview and structure of a system in terms of classes, attributes and methods, and the relationships between different classes. The class diagram contains three compartments. The first compartment contain class, middle compartment contain attributes and last compartment contains operation. There are many class in my class diagram that are obtained from NLA and they are Login, Admin, Customers, Buses etc.

The initial class diagram for Bus Ticket Management system is shown in figure below:-

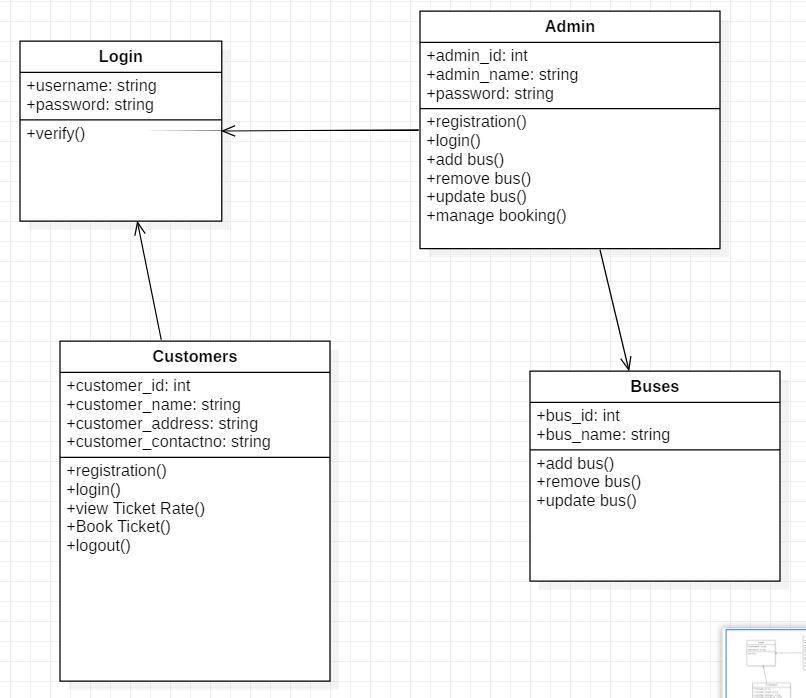


Figure 3: Initial Class Diagram for Bus Ticket Management System

### 1.7.3: Use Case Diagram:

Use Case Diagrams describe what a system does from the viewpoint of an external observer. Its emphasis is on what a system does rather than how. It is known as dynamic diagram in UML. It includes the actors, use case and the system. It deals with the interaction of user to the system that represents the relationship between the user and the different use cases used in the system.

The Use Case Diagram of the Bus Ticket Management system that will show the function of our system is as follows: -

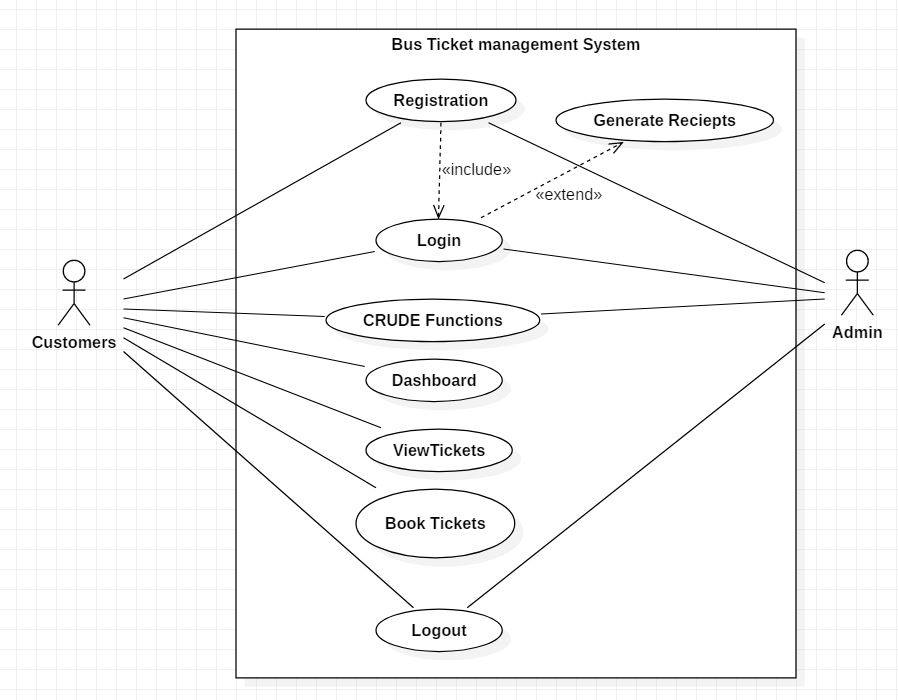


Figure 4: Use case diagram for Bus Ticket Management System