**Chapter 2**

**Software Requirement and Specification**

**2.1 Introduction:**

A software requirement specification (SRS) is a comprehensive description on the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform.

An SRS minimises the time and effort required by developer to achieve desired goals and also minimise the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-world situation.

**2.1.1 Purpose:**

The purpose of this document is to specify each requirement of the system in detail, while helping the client object a better understanding of the project. The document describes the project target audience and its user interface, hardware and software requirements.

**2.1.2 Scope:**

The document specifies requirements of the system. The validation of the process is done through software requirement specification. If there are any changes in the software in future, the changes are done through software requirement specification. First, the developer is responsible in case any change is seen in near future. The clarification must be changed in the recording of the system requirement specification.

* + 1. **FEASIBILITY STUDY**
* Economic Feasibility:
  + As existing system is manual, where data may not accurate, up to date, and available on time. But proposed system will be computerized, so we can overcome all limitations of existing system.
  + This system will reduce the paperwork. And quality of data will be improved.
* Technical Feasibility:
  + It is technically feasible, since the whole system is designed into the latest technologies like HTML, CSS, JavaScript, Python and My SQL Server which are the most recent technologies to develop web-based systems and design databases.
  + It uses the latest hardware technologies like Intel Pentium 4 or above Systems so easy to operate.
* Operational Feasibility:
  + It is operational feasible, since the system is providing a interactive user interface to the operator/end user, so he/she feel very easy to work onto it. Response to operator/end user is very fast and very good.
  + Since, as we mentioned above that it requires much less amount of cost, it uses computer work so it is very fast to operate and it is very easy for user to work on it.

**2.1.4 Definition, Acronyms and Abbreviation:**

**Admin:**

Admin is the administrator who is responsible for view, verify and approve or reject the student request and also responsible for view and issue the bus pass.

**Student:**

User of Online Student Bus Pass Application

HTML: Hypertext Markup Language

CSS: Cascading Style Sheets

MySQL: My Structured Query Language

DFD: Data Flow Diagram

CFD: Context Flow Diagram

ER: Entity Relationship

SRS: Software Requirement Specification

GUI: Graphical User Interface

**2.1.5 Overview:**

Online Student Bus Pass Application is useful for Students who are facing problems with the current manual work of bus pass registration and renewal. The Students need to register by submitting their details through online …The administrator will verify the student details and if they are satisfied, they will approve bus pass. After making the payment by the student, administrator verify and issue the bus pass to Students.

**2.2 Languages and Tools:**

**2.2.1 Frontend:**

**HTML:**

HTML stands for **H**yper**t**ext **M**arkup **L**anguage, and it is the most widely used language to write Web Pages.

* **Hypertext** refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.
* As its name suggests, HTML is a **Markup Language** which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language.

**CSS:**

**C**ascading **S**tyle **S**heets, fondly referred to as **CSS**, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.  
CSS is easy to learn and understood, but it provides powerful control over the presentation of an HTML document.

**BOOTSTRAP:**

**Bootstrap** is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source) [CSS framework](https://en.wikipedia.org/wiki/CSS_framework) directed at responsive, [mobile-first](https://en.wikipedia.org/wiki/Responsive_web_design#Mobile_first,_unobtrusive_JavaScript,_and_progressive_enhancement) [front-end web development](https://en.wikipedia.org/wiki/Front-end_web_development).

It contains [CSS](https://en.wikipedia.org/wiki/CSS)-and(optionally) [JavaScript](https://en.wikipedia.org/wiki/JavaScript)-based design templates for [typography](https://en.wikipedia.org/wiki/Web_design#Typography), [forms](https://en.wikipedia.org/wiki/Form_(HTML)), [buttons](https://en.wikipedia.org/wiki/Button_(computing)#HTML), [navigation](https://en.wikipedia.org/wiki/Web_navigation#Local_website_navigation), and other interface components.

**JavaScript:**

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as **LiveScript,** but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **LiveScript**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

**2.2.2 Backend:**

**Python:**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

**Python** is an interpreted, object-oriented, high-level programming language with dynamic semantics. ... **Python's** simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. **Python** supports modules and packages, which encourages program modularity and code reuse.

**2.3 Overall description**

**2.3.1 Product Perspective:**

This project is intended to enhance and streamline the current manual bus pass system. It is improved version of system already implemented in the state of Haryana.

**2.3.2 Product Functions:**

* Issue an Electronic pass to all Students.
* Allow Students to pay for pass Online.
* Authenticate personal details of Students quickly while on the bus.

**2.3.3 User Characteristics:**

1. The user needs to have the basic computer knowledge to operate the system.
2. To use this application user should be the administrator. To login into the system, the user should be familiar with valid username and password. If the user invalid user name and password access denied.

The type of users and their characteristics:

* **Admin**

The admin has the has high value of authority compared to the students. He has an authority of creating his own password and he can add new admin to the online student bus pass application.

* **Student**

Here the Students has less authority compared to administrator. Student has authority of creating his own password and he can change his password at any time.

**2.3.4 General Constraints:**

1. User of the system must be comfortable with English language.
2. Developed system should run on type of operating system that supports HTML, CSS and JavaScript, the required front end and Python should be used as backend.

**2.3.5 Hardware Requirements:**

1. Any standard keyboard.
2. Any standard mouse.
3. Any standard CD-ROM.

**2.3.6 Assumptions and dependencies**

* User of this computer should have the computer knowledge. The user knows the English language as the interface will provided in English.
* Admin should be careful in handling or modifying data knowingly or unknowingly which will lead to inconsistency of the database.

**2.4 Specific Requirements:**

The specification requirement section describes all the details that the software developers need to know for designing and developing the system. This is the largest and most important part of the document. The specific requirement can be organized by specifying the external interface followed by functional requirements, design constraints and system attribute.

**2.4.1 External interface requirements:**

The interface should be simple and easy to understand and use. It should always be interactive interface. It should prompt the user and administrator for proper input criteria.

**2.4.1.1 User Interfaces:**

This application provides good graphical interface for the user and the Administrator can operate on the system, performing the request task.

* The user interface must be customizable by the administrator.
* All the modules provided with the software must fit into this graphical

user interface and accomplish to the standard defined.

* The design should be simple and all the different interfaces should follow a standard template.
* The user interface should be able to interact with the other users.
* Design of registration page, log in page is easy to understand for every user.

**2.4.1.2 Hardware Interfaces:**

1. RAM :4GB
2. Input device : Keyboard, Mouse
3. Output device : Monitor, printer
4. Storage :40GB
5. Processor : Intel Pentium 4 or above.

**2.4.1.3 Software Interfaces:**

1. Operating System : Windows XP and above
2. Fronted : HTML, CSS, JavaScript
3. Backend : Python

**2.4.2 Function requirement specifications**

The Functional Requirements Specification documents the operations and activities that a system must be able to perform.

Functional Requirements should include:

* Descriptions of data to be entered into the system
* Descriptions of operations performed by each screen
* Descriptions of work-flows performed by the system
* Descriptions of system reports or other outputs
* Who can enter the data into the system
* How the system meets applicable regulatory requirements

The Functional Requirements Specification is designed to be read by a general audience. Readers should understand the system, but no particular technical knowledge should be required to understand the document.

**2.4.2.1 Module Description:**

**2.4.2.1.1 Admin Module:**

|  |  |
| --- | --- |
| **Module Name** | **Login** |
| **Description** | This module allows the admin to access all his authority. |
| **Data Inputs** | 1. User Name 2. Password |
| **Processing** | Login Module allows admin to login the system using username and password. |
| **Output** | If login success displays related page otherwise display error message |

|  |  |
| --- | --- |
| **Module name** | **View** |
| **Description** | Administrator can view the student’s who are applied for bus pass also view the documents uploaded by the students, then verify and manage the students’ profile. |
| **Data Inputs** |  |
| **Processing** |  |

|  |  |
| --- | --- |
| **Module Name** | **Approve/ Reject** |
| **Description** | The admin views all the credentials of student and verify it, based on the verification conformation update the approve or reject status. |
| **Data Inputs** |  |
| **Processing** |  |

|  |  |
| --- | --- |
| **Module Name** |  |
| **Description** |  |
| **Data Inputs** |  |
| **Processing** |  |

|  |  |
| --- | --- |
| **Module Name** |  |
| **Description** |  |
| **Data Inputs** |  |
| **Processing** |  |

**2.4.2.1.2 Student Module:**

|  |  |
| --- | --- |
| **Module Name** | **Login** |
| **Description** | Here Student can login to the device based on the registered user name and password. |
| **Data Inputs** | 1. User Name 2. Password |
| **Processing** | 1. Student will enter his/her username and password to login and then apply or renew for bus pass 2. The login information is verified in table TblLogin. |

|  |  |
| --- | --- |
| **Module name** | **Register** |
| **Description** | This module allows the students to register. |
| **Data Inputs** | 1. Username 2. Email id 3. Phone Number 4. Password 5. Confirm Password |
| **Processing** | * Registered information will store in the table TblLogin. |

|  |  |
| --- | --- |
| **Module Name** |  |
| **Description** |  |
| **Data Inputs** |  |
| **Processing** |  |

|  |  |
| --- | --- |
| **Module Name** |  |
| **Description** |  |
| **Data Inputs** |  |
| **Processing** |  |

|  |  |
| --- | --- |
| **Module Name** |  |
| **Description** |  |
| **Data Inputs** |  |
| **Processing** |  |

|  |  |
| --- | --- |
| **Module Name** |  |
| **Description** |  |
| **Data Inputs** |  |
| **Processing** |  |

|  |  |
| --- | --- |
| **Module Name** |  |
| **Description** |  |
| **Data Inputs** |  |
| **Processing** |  |

**2.4.3 Performance Requirement:**

This part of SRS specifies the performance constraints on the software system. All the requirements relating to the performance characteristics of the system must be clearly specified.

There are two types of performance requirements

1. Static
2. Dynamic

Static Requirements:

Static Requirements are those that do not impose constraints on the execution characteristics of the system. These includes requirements like the number of terminals to be supported and the number of files that system has no produce and their sizes.

Dynamic Requirements:

Dynamic Requirements specifying constraints on the execution behaviour of the system this typically include response time and throughput constraints on the system.

* + 1. **Design Constraints:**

When the developer states design of the system there exist number of factors in client’s environment that may be restrict the choices. Such factors include standards that must be followed, resource limits, operating environment, reliability and security.

1. Security: only the authorised users are allowed to use this application.
2. Reliability: Reliability includes the factor that it gives the proper result. If the system does not find the required result then an appropriate notification id displayed in form of message in pop up.
3. Portability: System is adoptable to any computer and work more efficiently in windows XP and above operating systems.
4. Reusability: The system modules are reusable for the next development process. With the user of this system developer can enhance it with other improved or additional feature.
5. Extensibility: It allows an easy functionality and add interface that suits to current requirement of the user.

**2.4.5 Other Requirements:**

When the system in completely developed and submitted to the management, few sessions will be required to make the users of the system understand about the functionality of it and some time to adopt the system. After those sessions it is required that the members from the development team should spend some time in the system background for an agreed time period. That new time period used in identifying new bugs that could not be reached in the earlier phases of the development process.