# CHAPTER 1

# INTRODUCTION

## 1.1 ABOUT THE PROJECT

# CHAPTER 2

# SYSTEM ANALYSIS

## 2.1 INTRODUCTION

## 2.2 EXISTING SYSTEM

### 2.2.1 DRAWBACKS

•

## 2.3 PROPOSED SYSTEM

**2.3.1 ADVANTAGES OF PROPOSED SYSTEM**

## 2.4 MODULE DESCRIPTION

* **Sign-up and log in:** The user has to sign in with the name, email ID, contact number, and address.
* **Admin Module**
* **User Module**

## 2.5 FEASIBILITY STUDY

A feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, time and effort spent on it. Feasibility study lets the developer to foresee the future of the project and its usefulness. There are three aspects in the feasibility study portion of the preliminary investigation:

* Technical Feasibility
* Economic Feasibility
* Operational Feasibility

The proposed system must be evaluated from the technical point of view first, and if technically feasible their impact on the organization must be assessed. If compatible, operational system can be devised. Then they must be tested for economic feasibility.

### 2.5.1 TECHNICAL FEASIBILITY

The technical feasibility study is a study of function, performances and constraints and improve the ability to create an acceptable system. Technical feasibility is frequently the most difficult are to achieve at the stage of product engineering process. The system must be evaluated from technical viewpoint first. The assessment of this feasibility must be based on the outline design of the system requirements in the terms of inputs, outputs program procedure and staffs. The project (Project Name) is said to be technically feasible. Technical feasibility centres on the existing computer systems and extend to which it can support the proposed system. This involves financial consideration to technical enhancements.

### 2.5.2 ECONOMICAL FEASIBILITY

Economic analysis is the most frequently used method for evaluating the effectiveness of the proposed system. It evaluates whether the system benefits greater than cost. The proposed (Project Name) is an effective one since the benefits of the software outweigh the cost incurred in installing it. It can be developed under optimal expenses with the available hardware and software.

### 2.5.3 OPERATIONAL FEASIBILITY

The purpose of the operational feasibility study is to determine the whether the new system “(Project Name)” will be used if it is developed and installed. And whether there will be resistance from users that will undermine the possible application benefit. The first challenge was whether the system meets the organizational requirements. This is checked by the system requirement collected from the different datasets of different users posts and the operational feasibility proved that the system is capable to meet its functional requirements. The developed system is completely driven and user friendly.“(Project Name)” is a best solution for detecting derogatory / provocative / sensitive messages or posts. In Operational feasibility is a measure of how proposed system solves the existing system problem and how it satisfies requirements identified in the requirement analysis phase of system development.

### 2.5.4 BEHAVIOURAL FEASIBILY

Proposed projects are beneficial only if they can be turned into information systems that will meet the operating requirements of the organization. This test of feasibility asks if the system will work when it is developed satisfies all the operational conditions. It was the most difficult task for me, but met efficiently. As this package is found to be feasible technically, economically and functionally, the system is judged feasible. Viewing the collected information, recommendation and justification, conclusions is made of the proposed system. Hence decision is taken to go on with the project.

### 2.5.5 LEGAL FEASIBILITY

A determination of any infringement, violation, or liability that could result from development of the system. Legal feasibility encompasses a broad range of concerns that include contracts, liability, infringement, and myriad other traps frequently unknown to technical staff. It determines whether the proposed system conflicts with legal requirements, eg., a data processing system must comply with the local data protection regulations and if the proposed venture is acceptable in accordance to the low of the land.

# CHAPTER 3

**SYSTEM REQUIRMENTS AND SPECIFICATIONS**

## 3.1 INTRODUCTION

Requirement analysis involves studying the current system to find out how it works and where improvements could be made. A clear idea about the existing system is a must for making improvements where it is required. Proper planning and collection of data serves the purpose. The popularity of this document is to describe all the requirements for the popularity of the website for intelligent business analysis and planning software for transport department. This website gives all the information for a person. In our busy life, we cannot go to each department or offices. This problem will manage this site. It will help to avoid in correction and missing data.

## 3.2 SYSTEM REQUIREMENT

System requirements are all of the requirements at the system level that describe the functions which the system as a whole should fulfill to satisfy the stakeholder needs and requirements, and is expressed in an appropriate combination of textual statements, views, and non-functional requirements; the latter expressing the levels of safety, security, reliability, etc., that will be necessary.

System requirements play major roles in systems engineering, as they:

* + - Form the basis of system architecture and design activities.
    - Form the basis of system integration and verification activities.
    - Act as reference for validation and stakeholder acceptance.
    - Provide a means of communication between the various technical staff that interact throughout the project.

## HARDWARE REQUIREMENTS

* CPU - I3 Processors
* Hard Disk Space - 250 Gb
* Display - 15‖ Color Monitor
* Main Memory - 8 Gb
* Keyboard - 104 Keys
* Clock-Speed - 2.6 Ghz
* Monitor - 15 ″ Svga Color

## SOFTWARE REQUIREMENTS

* Operating System - Windows 10
* Backend - Python, PHP, MySQL
* Front End - HTML, CSS
* Server - Apache
* Software Used - Python Anaconda, XAMPP

**3.3 LANGUAGE DESCRIPTION**

**3.3.1 FRONT END**

* **HTML**

The HyperText Markup Language, or HTML is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language) for documents designed to be displayed in a [web browser](https://en.wikipedia.org/wiki/Web_browser). It can be assisted by technologies such as [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) and [scripting languages](https://en.wikipedia.org/wiki/Scripting_language) such as [JavaScript](https://en.wikipedia.org/wiki/JavaScript" \o "JavaScript).The first publicly available description of HTML was a document called HTML tags, first mentioned on the internet by Tim Berners-Lee in late 1991.

[Web browsers](https://en.wikipedia.org/wiki/Web_browser) receive HTML documents from a [web server](https://en.wikipedia.org/wiki/Web_server) or from local storage and [render](https://en.wikipedia.org/wiki/Browser_engine) the documents into multimedia web pages. HTML describes the structure of a [web page](https://en.wikipedia.org/wiki/Web_page) [semantically](https://en.wikipedia.org/wiki/Semantic_Web) and originally included cues for the appearance of the document.

[HTML elements](https://en.wikipedia.org/wiki/HTML_element) are the building blocks of HTML pages. With HTML constructs, [images](https://en.wikipedia.org/wiki/HTML_element#Images_and_objects) and other objects such as [interactive forms](https://en.wikipedia.org/wiki/Fieldset) may be embedded into the rendered page. HTML provides a means to create [structured documents](https://en.wikipedia.org/wiki/Structured_document) by denoting structural [semantics](https://en.wikipedia.org/wiki/Semantics) for text such as headings, paragraphs, lists, [links](https://en.wikipedia.org/wiki/Hyperlink), quotes and other items. HTML elements are delineated by *tags*, written using [angle brackets](https://en.wikipedia.org/wiki/Bracket#Angle_brackets). Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

* **CSS**

[CSS](https://www.w3.org/Style/CSS/)(Cascading Style Sheets) is the language for describing the presentation of Web pages, including colours, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. This is referred to as the separation of structure (or: content) from presentation.

* **XAMPP**

XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local web server. It was developed by the Apache Friends, and its native source code can be revised or modified by the audience. It consists of Apache HTTP Server, MariaDB, and interpreter for the different programming languages like PHP and Perl. It is available in 11 languages and supported by different platforms such as the IA-32 package of Windows & x64 package of macOS and Linux. XAMPP is an abbreviation where X stands for Cross-Platform, A stands for Apache, M stands for MYSQL, and the Ps stand for PHP and Perl, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command line executables along with modules such as Apache server, MariaDB, PHP, and Perl.

XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, Perl is a programming language used for web development, PHP is a backend scripting language, and MariaDB is the most vividly used database developed by MySQL.

**3.3.2 BACK END**

* **PYTHON-ANACONDA**

**Python** is an [interpreted](https://en.m.wikipedia.org/wiki/Interpreted_language)[,](https://en.m.wikipedia.org/wiki/High-level_programming_language) [high-level](https://en.m.wikipedia.org/wiki/High-level_programming_language) [a](https://en.m.wikipedia.org/wiki/High-level_programming_language)nd [general-purpose](https://en.m.wikipedia.org/wiki/General-purpose_programming_language) [programming](https://en.m.wikipedia.org/wiki/General-purpose_programming_language) [language.](https://en.m.wikipedia.org/wiki/General-purpose_programming_language) Created by [Guidovan](https://en.m.wikipedia.org/wiki/Guido_van_Rossum) [Rossum](https://en.m.wikipedia.org/wiki/Guido_van_Rossum) [a](https://en.m.wikipedia.org/wiki/Guido_van_Rossum)nd first released in 1991, Python's design philosophy emphasizes [code](https://en.m.wikipedia.org/wiki/Code_readability) [readability](https://en.m.wikipedia.org/wiki/Code_readability) [w](https://en.m.wikipedia.org/wiki/Code_readability)ith its notable use o[f](https://en.m.wikipedia.org/wiki/Off-side_rule) [significant](https://en.m.wikipedia.org/wiki/Off-side_rule) [white space.](https://en.m.wikipedia.org/wiki/Off-side_rule)Its [language](https://en.m.wikipedia.org/wiki/Language_construct) [constructs](https://en.m.wikipedia.org/wiki/Language_construct) [a](https://en.m.wikipedia.org/wiki/Language_construct)nd [object-oriented](https://en.m.wikipedia.org/wiki/Object-oriented_programming) [a](https://en.m.wikipedia.org/wiki/Object-oriented_programming)pproach aim to help [programmers](https://en.m.wikipedia.org/wiki/Programmers) [w](https://en.m.wikipedia.org/wiki/Programmers)rite clear, logical code for small and large-scale projects. Python is [dynamically](https://en.m.wikipedia.org/wiki/Dynamic_programming_language) [typed](https://en.m.wikipedia.org/wiki/Dynamic_programming_language) [a](https://en.m.wikipedia.org/wiki/Dynamic_programming_language)nd [garbage-collected.](https://en.m.wikipedia.org/wiki/Garbage_collection_(computer_science)) It supports multiple programming ,paradigms ,including structured [(](https://en.m.wikipedia.org/wiki/Structured_programming)particularly, [procedural)](https://en.m.wikipedia.org/wiki/Procedural_programming)[,object-oriented ,](https://en.m.wikipedia.org/wiki/Object-oriented_programming)and functional [programming.](https://en.m.wikipedia.org/wiki/Functional_programming) Python is often described as a "batteries included" language due to its comprehensive [standard](https://en.m.wikipedia.org/wiki/Standard_library) [library.](https://en.m.wikipedia.org/wiki/Standard_library) Python [interpreters](https://en.m.wikipedia.org/wiki/Interpreter_(computing)) [a](https://en.m.wikipedia.org/wiki/Interpreter_(computing))re available for many [operating](https://en.m.wikipedia.org/wiki/Operating_system) [systems.](https://en.m.wikipedia.org/wiki/Operating_system) A global community of programmers develops and maintains [Python,](https://en.m.wikipedia.org/wiki/CPython)[a](https://en.m.wikipedia.org/wiki/Free_and_open-source_software) [free](https://en.m.wikipedia.org/wiki/Free_and_open-source_software) [and](https://en.m.wikipedia.org/wiki/Free_and_open-source_software) [open-source](https://en.m.wikipedia.org/wiki/Free_and_open-source_software) [reference](https://en.m.wikipedia.org/wiki/Reference_implementation) [implementation.](https://en.m.wikipedia.org/wiki/Reference_implementation)A non-profit organization, the [Python](https://en.m.wikipedia.org/wiki/Python_Software_Foundation) [Software](https://en.m.wikipedia.org/wiki/Python_Software_Foundation) [Foundation,](https://en.m.wikipedia.org/wiki/Python_Software_Foundation)manages and directs resources for Python and Python development.

Python 2.0 was released on October 16, 2000, with many major new features, including a cycle-detecting [garbage](https://en.m.wikipedia.org/wiki/Garbage_collection_(computer_science)) [collector(](https://en.m.wikipedia.org/wiki/Garbage_collection_(computer_science))in addition to [reference](https://en.m.wikipedia.org/wiki/Reference_counting) [counting)](https://en.m.wikipedia.org/wiki/Reference_counting) for [memory](https://en.m.wikipedia.org/wiki/Memory_management) [management](https://en.m.wikipedia.org/wiki/Memory_management) [a](https://en.m.wikipedia.org/wiki/Memory_management)nd support for [Unicode .](https://en.m.wikipedia.org/wiki/Unicode)However, the most important change was to the development process itself, with a shift to a more transparent and community-backed process.Python 3.0, a major, backwards-incompatible release, was released on December 3, 2008 after a long period of testing. Many of its major features have also been[backported](https://en.m.wikipedia.org/wiki/Backport)to the backwards-compatible, while by now unsupported, Python 2.6 and 2.7. The Python 2 language was officially discontinued I n 2020 (first planned for 2015), and "Python 2.7.18 is the last Python 2.7 release and therefore the last Python 2 release[.](https://en.m.wikipedia.org/wiki/Monty_Python%27s_Flying_Circus)

**Anaconda** is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012.[8] As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator,[12] as a graphical alternative to the command-line interface (CLI).

Before version 20.3, when pip installed a package, it automatically installed any dependent Python packages without checking if these conflict with previously installed packages. It would install a package and any of its dependencies regardless of the state of the existing installation. Because of this, a user with a working installation of, for example, TensorFlow, could find that it stopped working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by TensorFlow. In some cases, the package would appear to work but produce different results in detail. While pip has since implemented consistentdependency resolution, this difference accounts for a historical differentiation of the anaconda package manager.

* **Django**

Django is a high-level, open-source web framework written in Python that enables the rapid development of secure and maintainable web applications. It was designed to help developers take applications from concept to completion as quickly as possible, emphasizing reusability, less code, and the "don't repeat yourself" (DRY) principle.

* **MYSQL**

MySQL is the world's most popular open-source database. With its proven performance, reliability and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, Yahoo! and many more. Oracle drives MySQL innovation, delivering new

capabilities to power next generation web, cloud, mobile and embedded applications. MySQL AB was a software company that was founded in 1995. It was acquired by Sun Microsystems in 2008; Sun was in turn acquired by Oracle Corporation in 2010. MySQL AB is the creator of MySQL, a relational database management system, as well as related products such as MySQL Cluster. The company was dually headquartered in Uppsala, Sweden and Cupertino, California with offices in other countries (Paris,

Munich, Dublin, Milan, and Tokyo).With around 400 employees in 25 countries, MySQL AB was one of the largest open-source companies worldwide. Around 70% of the employees worked for MySQL from their home offices.

MySQL is an open-source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX, and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web-based applications and online publishing and is an important component of an open-source enterprise stack called LAMP. LAMP is a Web development platform that uses Linux as the operating system, Apache as the Web server, and MySQL as the relational database management system and PHP as the object-oriented scripting language. (Sometimes Perl or Python is used instead of PHP). MySQL, which was originally conceived by the Swedish company MySQL AB, was acquired by Oracle in 2008. Developers can still use MySQL under the GNU General Public License (GPL), but enterprises must obtain a commercial license from Oracle.

# CHAPTER 4

# SYSTEM DESIGN

## 4.1 INTRODUCTION

The logical design describes structure and characteristics of features, like the outputs, inputs, databases and procedures. The physical construction which follows the logical design produces actual program software files and the working system. System design sits at the technical kernel of the software engineering and is applied regardless of the software process model that is used. Beginning once software requirements have been analyzed and specified software design is the first technical activity that is used to build and verify the software. Each activity (designing, coding and testing) transform information in a manner that ultimately results in validated computer software. System design is the process of developing specifications for a candidate system that meet the criteria established in the system analysis. Major step in system design is the preparation of the input forms and the output reports in a form applicable to the user. The main objective of the system design is to use the package easily by any computer operator. System Design is the creative act of invention, developing new inputs, a database, offline files, method, procedures and output for processing business to meet an organization objective. System design builds information gathered during the system analysis. In design an efficient and effective system is of great importance to consider the human factor and equipment that these will require to use. System analyst must evaluate the capabilities and limitations of the personal and corresponding factors of the equipment itself.

## 4.2 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

### 4.2.1 FUNCTIONAL REQUIREMENTS

Functional requirement explains what has to be done by identifying the necessary task, action or activity that must be accomplished. Functional requirement analysis will be used as the top level functions for functional analysis. Some of the requirements that resembles to our project are listed below.

Interface**:** The Graphical User Interface that communicates with the user with ease.

Network Access**:** Provide accessibility to application through Wi-Fi or cellular network.

Storage: MySQL, and Firebase that store the information to be displayed to the user.

Easy integration**:** The application should be easy to integrate with the existing system.

### 4.2.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are requirements that specify criteria that can be used to judge the operation of a system rather than specific behaviors.

### Reliability

It is required that the system should be available all the time. This can be achieved by hosting the system in a reliable server. Also the system is built using Java, and this adds more confidence to the system.

### Performance

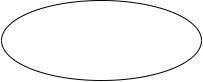
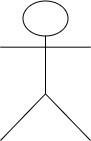
The application would be used by numerous customers throughout the country. So it was required that the system should take minimum time to produce output.The main concern was the time taken to query database system to extract the relevant names and calculate the similarity scores. This time depend upon the type of internet service used. Overall the response time for map loading and data retrieve from database has been improved to some extends.

### Accuracy

The application is purposed to be real time, so it is required that the high accuracy is maintained. This is maintained by the in-built Google API's and open source frameworks of Apache that supplies correct data and classes to give a precise coding option.

## 4.3 USE CASE DIAGRAM

Use cases help to determine the functionality and features of the software from user‘s perspective. A use case describes how a user interacts with the system by defining the steps required to accomplish a specific goal. Variations in the sequence of steps describe various scenarios. In the diagram the stick figure represents an actor that is associated with one category of user. In the use-case diagram the use cases are displayed as ovals. A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.



Actor Use Case

The actors are connected by lines to the use cases that they carry out. The use cases are placed in a rectangle but the actors are not .This rectangle is a visual remainder of the system boundaries and that the actors are outside the system. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view. In brief, the purposes of use case diagrams can be said to be as follows –

* Used to gather the requirements of a system.
* Used to get an outside view of a system.
* Identify the external and internal factors influencing the system.
* Show the interaction among the requirements is actors.

**4.3.1 Use Case Diagram**

## 4.4 DATA FLOW DIAGRAM

Data Flow Diagram (DFD) representing a system at any level of detail with a graphic network of symbols showing data flows, data stores, data processes, and data sources. The purpose of DFD is to provide a semantic bridge between users and system developers. The diagram is the basis of structured system analysis. A level 0 DFD, also called a fundamental system model or a context model represents the entire software elements as a single bubble with input and output indicated by incoming and outgoing arrows respectively. Additional process and information flow parts are represented in the next level i.e., Level 1 DFD. Each of the processes represented at Level 1 are sub functions of overall system depicted in the context model. Any processes, which are complex in Level 1, will be further represented into sub functions in the next level, i.e., in level 2.Data flow diagrams illustrate how data is processed by a system in terms of inputs, and outputs. Represent major components or functions with Circles. Actions for input by a user or a system go in Rectangular Boxes. Databases are represented by Parallel lines enclosing a phrase corner.

For each data flow, at least one of the endpoints (source and / or destination) must exist in a process. The refined representation of a process can be done in another data- flow diagram, which subdivides this process into sub-processes. The data-flow diagram is part of the structured-analysis modelling tools. When using UML, the activity diagram typically takes over the role of the data-flow diagram. A special form of data-flow plan is a site-oriented data- flow plan. Data-flow diagrams can be regarded as inverted Petri nets, because places in such networks correspond to the semantics of data memories. Analogously, the semantics of transitions from Petri nets and data flows and functions from data-flow diagrams should be considered equivalent. Rules for Data flow diagrams are:

* External Entities

External entities are objects outside the system, with which the system communicates. External entities are sources and destinations of the system‗s inputs and outputs

* Processes

When naming process, avoid glossing over them, without really understanding their role. Indications that this has been done are the use of vague terms in the descriptive title area like‘ process or Data Flows update.

* Data Flows

Double-headed arrows can be used (to show two-way flows) on all but bottom level diagrams. Furthermore, in common with most of the other symbols used, a data flow at a particular level of a diagram may be decomposed to multiple data flows at lower levels. Flow should only transmit one type of information (material). The arrow shows the flow direction (it can also be bi-directional if the information to/from the entity is logically dependent - e.g. question and answer). Flows link processes, warehouses and terminators.

* Data Store

Data store represent stores of data within the systems and are represented by open rectangle. Data Flows represent the movements of data between other components and are shown by arrows.

Rectangular box defined a source of destination of the system.

A circle stands for processes that convert data into information

Data Flow

Open Rectangle, which shows a data stores.

### 4.4.1 LEVEL 0

**4.4.2 LEVEL 1**

### 4.4.3 LEVEL 1.1

**4.4.4 LEVEL 1.2**

## 4.5 INPUT DESIGN

Input design is the process of converting the user-originated inputs to a computer-based format. The design for handling input specifies how data are accepted for computer processing. Input design is a part of overall system design that needs careful attention and if includes specifying the means by which actions are taken. A system user interacting through a workstation must be able to tell the system whether to accept input produce a report or end processing. The collection of input data is considered to be the most extensive part of the system design. Since the inputs have to be planned in such a manner so as to get the relevant information extreme care is taken to obtain the information. If the data going into the system is incorrect then the processing and outputs will magnify these errors. The major activities carried out are,

* Collection of needed data from the source
* Conversion of data into computer accepted
* Verification of converted data

## 4.6 OUTPUT DESIGN

The output design has been done so that the results of processing should be communicated to the user. Effective output design will improve the clarity and performance of outputs. Output is the main reason for developing the system and the basis on which they will evaluate the usefulness of the application. Output design phase of the system is concerned with the convergence of information to the end user-friendly manner. The output design should be efficient, intelligible so that system relationship with the end user is improved and they‘re by enhancing the process of decision-making. Because useful output is essential to ensuring the use and acceptance of the information system, there are six objectives that the systems analyst tries to attain when designing output:

* Designing output to serve the intended purpose.
* Designing output to fit the user.
* Delivering the appropriate quantity of output.

## 4.7 DATABASE DESIGN

The database design is a logical development in the methods used by the computers to access and manipulate data stored in the various parts of the computer systems. Database is defined as an integrated collection of data. The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. The main objectives of databases are data integrity and data independence. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. The database serves as the repository of data, so a well-designed database can lead to a better program structure and reduce procedural complexity. In a database environment, common data are available and used by several users. The main objectives covered in database design are:

### Controlled redundancy

* **Data independence**

### Accuracy and integrity

* **Privacy and security**
* **Performance**

## 4.8 TABLE DESIGN

This is one of the major tasks is designing the database. It is important to realize that the design of the system is totally inter-related and so table design of the system is totally inter-related and so table design cannot really be considered in isolation from inputs, outputs, procedures, codes and security requirements. In this project, the database has to maintain all the information about the username, password, tolerance, picture sequence, sound clips and click points etc.

## 4.9 DATABASE TABLE

Table No: 4.9.1

Table Name:

Table Description:

Table No: 4.9.2

Table Name:

Table Description:

## 4.10 DATASET

# CHAPTER 5 SYSTEM CODING

## 5.1 INTRODUCTION

The coding step is a process that transforms design into a programming language. It translates a detail design representation of software into a programming language realization. The translation process continues when a compiler accepts source code as input and produces machine-dependent object code as output. Quality is an important goal during coding. The quality of source code can be improved by the use of structured coding techniques; good coding style and readable, consistent code format. During coding, some coding standards are to be followed. This has two purposes; reducing the chance of making it easier for some time to modify the code later on. Coding phase affects both testing and maintenance profoundly. The ―Budget Planner‖ uses Python as the programming language for coding. Coding methodology refers to a set of well- documented procedures and guidelines used in the analysis, design, and implementation of programs. Coding methodology includes a diagrammatic notation for documenting the results of the procedure. It also includes an objective set (ideally quantified) of criteria for determining whether the results of the procedure are of the desired quality.

## 5.2 CODING STANDARDS AND GUIDELINES

Good software development organizations usually develop their own coding standards and guidelines depending on what best suits their organization and the type of products they develop. The following are some representative coding standards:-Rules for limiting the use of global: These rules list what types of data can be declared global and what cannot. Contents of the headers preceding codes for different modules:

The information contained in the headers of different modules should be standard for the organization. The exact format in which the header information is organized in the header can also be specified.

The following are some standard header data:

* + - Name of the module.
    - Date on which the module was created. Author‘s name.
    - Modification history.
    - Synopsis of the module.
    - Different functions supported, along with their input/output parameters.Global variables accessed/modified by the module.

Naming conventions for global variables, local variables and constant identifiers: A possible naming convention can be that global variable names always capital letter, local variable names are made of small letters, and constant names are always capital letters. Error returns conventions and exception handling mechanisms: The way error conditions are reported by different functions in a program are handled should be standard within an organization. For example, different functions while encountering an error condition should either return a 0 or 1 consistently. The following are some representative coding guidelines recommended by many software development organizations. Do not use a coding style that too clever or too difficult to understand: Code should be easy to understand. Many inexperienced engineersactuallytake pride in writing cryptic and incomprehensible code. Clever coding can obscure meaning of the code and hamper understanding. It also makes maintenance difficult. Do not use an identifier for multiple purposes: Programmers often use the same identifier to denote several temporary entities. However several things wrong with this approach and hence should be avoided. Some of the problems caused by use of variables for multiple purposed as follows: Contents of the headers preceding codes for different modules:

The information contained in the headers of different modules should be standard for the organization. The exact format in which the header information is organized in the header can also be specified. Each variable should be given a descriptive name indicating its purpose. Use of variable for multiple purposes usually makes future enhancements more difficult. Avoid obscure side effects: The side effects of a function call include modification of parameters passed by reference.

## 5.3 SAMPLE CODE Testing code

## Training Code

# CHAPTER 6

# SYSTEM TESTING

## 6.1 INTRODUCTION

In any software development, testing is a process to show the correctness of the program and it meets the design specifications. Testing is needed to prove correctness, to show completeness, to improve the quality of the software and to provide the maintenance aid. Some testing standards are therefore necessary to ensure completeness of testing, improve the quality of the software, and reduce the testing costs and to reduce study needs and operation time. Testing software extends throughout the coding phase and it represents the ultimate review of configurations, design and coding. A series of test cases are created that are intended to demolish the software that has been built. Testing involves a series of operation of a system of application under controlled conditions and subsequently evaluating the result. The controlled condition should include both normal and abnormal conditions. It is planned and monitor for each testing level (e.g., unit, integration, system and acceptance). The various testing performed are unit testing, integration testing, validation testing, output testing and system testing.

## 6.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be tested within the test plan.

The levels of testing include:

### 6.2.1 UNIT TESTING

The primary goal of unit testing is to take the smallest piece of testable software in the application, isolate it from the remainder of the code, and determine whether it behaves exactly as you expect. Each unit is tested separately before integrating them into modules to test the interfaces between the modules. Unit testing has proven its value in that a large percentage of defects are identified during its use. The most common approach to unit testing requires drivers and stubs to be written. The driver simulates a calling unit and the stub simulates a called unit. The investment of developer time in this activity sometimes results in demoting unit testing to a lower level of priority and that is almost always a mistake. Even though the drivers and stubs cost time and money, unit testing provides some undeniable advantages. It allows for automation of the testing process, reduces difficulties of discovering errors contained in more complex pieces of the application, and test coverage is often enhanced because attention is given to each unit. Unit testing focuses first on modules, independently of one another, to locate errors. This testing includes entering data and ascertaining if the value matches to the type and size. The various controls are tested to ensure that each performs its action as required. The main modules of the project such as admin, stand, conductor, checker, virtual bank are tested separately. Also each user is login and tested the sub modules.

### 6.2.2 DATA VALIDATION TESTING

Data validation is the process of testing the accuracy of data. A set of rule we can apply to a control to specify the type and range of data that can enter. It can be used to display error alert when users enter incorrect values in to a form. Now performing validation testing in system Centralized Social Welfare by undergoing validation for each tools and in a manner that can be reasonably accepted, by the user. At the culmination of integration testing, the software was completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of the validation succeeded when the software function in a software validationtestingbegan.In validation testing the entered data validated for correct format, and correct order.

### 6.2.3 OUTPUT TESTING

After performing validation test, the next phase is output test of the system, since no system could be useful if it does not produce desired output in the desired format. The output format was considered in two ways: one is on the screen and the other as a printed form. The main output of the project is that the admin gets the graph of loss and profit of bus extracted from the reviews provided by the stand and the output is successfully tested.

### 6.2.4 BLACK-BOX TESTING

Black box testing is defined as a testing technique in which functionality of the Application Under Test (AUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on software requirements and specifications. In Black-Box Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program. It is a way of software testing in which the internal structure or the program or the code is hidden and nothing is known about it. It is mostly done by software testers. No knowledge of implementation is needed. It can be referred as outer or external software testing. It is functional test of the software. This testing can be initiated on the basis of requirement specifications document. No knowledge of programming is required. It is the behavior testing of the software. It is applicable to the higher levels of testing of software. It is also called closed testing. It is least time consuming. It is not suitable or preferred for algorithm testing. It Can be done by trial and error ways and methods.

### 6.2.5 WHITE-BOX TESTING

White-box testing is a testing technique which evaluates the code and the internal structure of a program. White box testing involves looking at the structure of the code. When you know the internal structure of a product, tests can be conducted to ensure that the internal operations performed according to the specification. And all internal components have been adequately exercised. It is a way of testing the software in which the tester has knowledge about the internal structure r the code or the program ofthe

Software. It is mostly done by software developers. Knowledge ofimplementation is required. It is the inner or the internal software testing. It is structural test of the software. This type of testing of software is started after detail design document. It is mandatory to have knowledge of programming. It is the logic testing of the software. It is generally applicable to the lower levels of software testing. It is also called as clear box testing. It is most time consuming. It is suitable for algorithm testing. Data domains along with inner or internal boundaries can be better tested.

## 6.3 TEST CASE

Test case is a set of conditions or variables under which will a tester will determine whether a system under test satisfies requirements correctly. The process of developing test cases can also find problems in the requirements or design of an application. In order to fully test that all the requirements of an application are met, there must be at least two test cases for each requirement: one positive test and one negative test. If a requirement has sub- requirements, each sub-requirement must have at least two test cases. Keeping track of the link between the requirement and the test is frequently done using a traceability matrix. Written test cases should include a description of the functionality to be tested, and the preparation required to ensure that the test can be conducted. A formal written test-case is characterized by a known input and by an expected output, which is worked out before the test is executed. The known input should test a precondition and the expected output should test a post condition.

|  |  |  |
| --- | --- | --- |
| **Module Name** | **Test Case** | **Expected Result** |
|  |  |  |

Table 6.3.1 Test Case

# CHAPTER 7 IMPLEMENTATION AND MAINTENANCE

## 7.1 IMPLEMENTATION OF THE PROJECT

The implementation is the final stage and it is an important phase. It involves the individual programming; system testing, user training and the operational running of developed proposed system that constitutes the application subsystems. One major task of preparing for implementation is education of users, which should really have been taken place much earlier in the project when they were being involved in the investigation and the design. The implementation phase of the software development is concerned with translating design specifications into source code. The user tests the developed system and changes are made according to their needs. Our system has been successfully implemented. Before implementation several tests have been conducted to ensure that no errors are encounter during the operation. The implementation phase ends with an evaluation of the system after placing it into operation for a period of time. Implementation is the third phase of the system process. In order to achieve the objectives and the expected performance the system has been developed in a highly interactive and user-friendly manner.

## 7.2 SYSTEM MAINTENANCE

The launching of the system, even though it is fully correct and complete is not the end of the matter. The system should be given proper security and maintenance in order to keep them efficient and up-to-date. The system security is for protection against fraud and disaster. To avoid unauthorized access, password protection is highly recommended while running this new system. The password as to be maintained directly and files have to be kept very confidential. Finally systems and programs that have been successfully implemented are usually subjected to continuous change. The system should made modification and improvement to meet changing conditions. The first class maintenance requires the cooperation of the people served by the system or programs and that Reasonable for maintaining it. An integral part of software is the maintenance

One, which requires an accurate maintenance plan to be prepared during the software development

### 7.2.1 MAINTENANCE ISSUES

Maintenance ease with which a program can be corrected if any error is encountered, adapted if its environment changes or enhanced if the customer decides a change in requirements. The software is characterized by the following activities. In this project considerable amount of time is spent in maintenance and monitoring.

* Corrective Maintenance

Corrective maintenance is to uncover the errors still exist after testing. During this maintenance work the user is asked to work on the system and if any error is reported.

* Adaptive Maintenance

The adaptive maintenance is needed if the platform or the environment of the project to be change. For the project the language takes care of all of these things.

* Perfective Maintenance

The third maintenance active is the perfective maintenance. The recommendation of new capabilities and modification of existing function and general enhancement are received from the user and proposed future enhancement.

## 7.3 SOFTWARE MAINTENANCE

Maintenance involves the software industry captive, typing up system resources. It means restoring something to its original condition. Maintenance involves a wide range of activities including correcting, coding, and design errors, updating documentation and test data, and upgrading user support. Maintenance was done after the success implementation. Maintenance is continued till the product is reengineered or deployed to another platform

# CHAPTER 8

**CONCLUSION AND FUTURE ENHANCEMENT**

## 8.1 CONCLUSION

## 8.2 FUTURE ENHANCEMENT

**REFERENCES**

## APPENDIX

**SCREENSHOT**