SamanyaSeva : Client Management Assure

Submitted for partial fulfilment of the Degree
of
Bachelor of Technology
(Computer Science)

Submitted By:
Amisha Budhiraja (1410808)
Harmanjot Kaur (1410838)
Deepti Sharma (1410821)
D4CSE-A1

Submitted To:
Sumeet Kaur Sehra
Project Supervisor
CSE Department

Department of Computer Science & Engineering
Guru Nanak Dev Engineering College
Ludhiana 141006

Abstract

SamanyaSeva is basically an E-Commerce cum CRM (Customer relationship management) Django Web Application. The basic idea behind it was to make a generalized E-commerce software that could be implemented in Companies as well as hospitals, bakery and other shops.

This project is basically a combination of people, processes and technology that seeks to understand a company's customers. It is an integrated approach to managing relationships by focusing on customer retention and relationship development.

CRM revolves around people, process and technology. SamanyaSeva has been developed using the latest technologies. You don't have to be a techie to use this software. All you need to do is sip on your coffee and maintain the record.

Some of the important features of this application are that it provides different powers for specific roles like customer, admin etc. Also, there are different modules for every task such as Bills, Catalog , Vouchers etc. Moreover, anyone can use this service and need not have dependencies installed on their systems and can use this service remotely.

Also, this project is completely open source and the entire code is available to the user as and when required. There is Complete developer's Documentation as well as User manual alongwith it that helps using it a lot easier.

Contents

List of Figures

List of Tables

Introduction

1.1 Introduction to Project

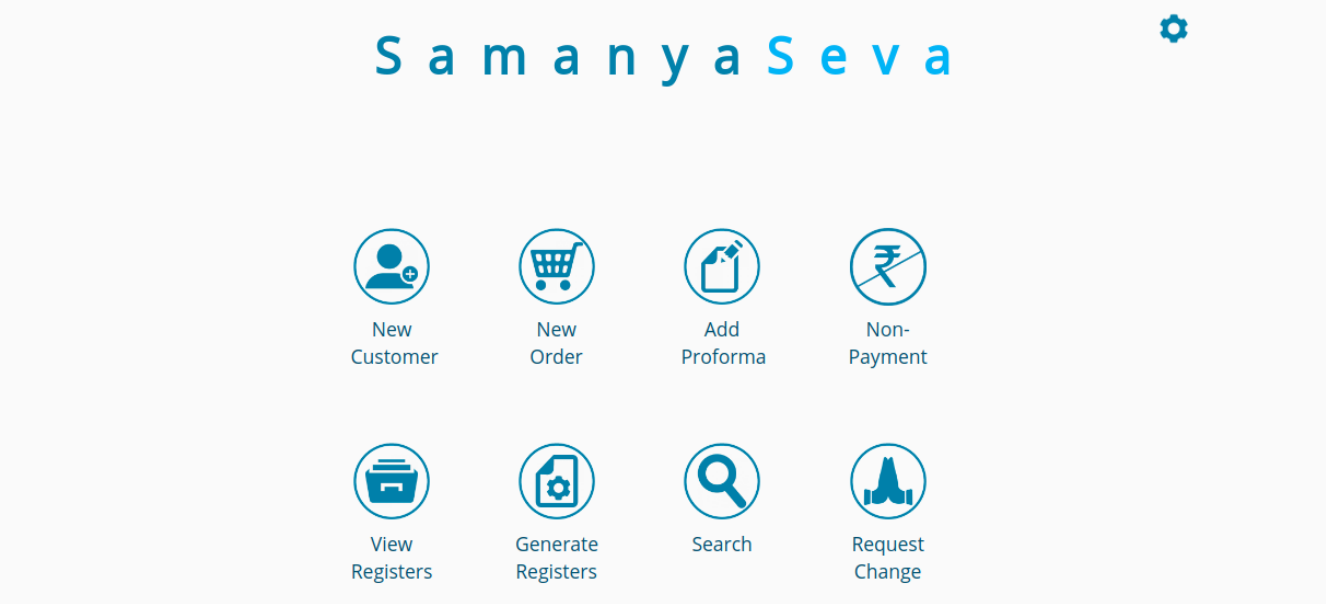


Figure 1.1: Client Management Assure

1.1.1 What is SamanyaSeva?

SamanyaSeva is an integrated software solution offered which can be used to support the seamless integration of information that flows through an organization. The basic idea was to make a generalized E-commerce software. It is provided as a package comprising different modules, such as Bills, Voucher, Catalog, Customer information etc.
Mainly, a SamanySeva software is a customer relationship management tool that most businesses use to manage and analyze customer interactions and data throughout the customer lifecycle, with the goal of improving business relationships with customers, assisting in customer retention and driving sales growth.

It is basically an E-Commerce cum CRM (Customer relationship management) Django Web Application. The basic idea behind it was to make a generalized E-commerce software that could be implemented in Companies as well as hospitals, bakery and other shops.

This project is basically a combination of people, processes and technology that seeks to understand a company's customers. It is an integrated approach to managing relationships by focusing on customer retention and relationship development. In code, there are several modules in SamanyaSeva including Catalog, Reports, Bills, Prints, Program Letter, Voucher.

CRM revolves around people, process and technology. SamanyaSeva has been developed using the latest technologies. Earning profit is the principal and final goal of any commercial activity. Being a catalyst for interplay with clients, CRM aids the earning power enhancement.

1.1.2 Who uses SamanyaSeva

Customer Relation Management systems are widely used by engineering teams and scientists, in both industry and academia for different purposes. Its even been used by business maintenance teams and various authors, for attract- ing the crowd and for promoting their work. These can be used in various sectors:

- Business Sector
- Education Sector
- IT firms
- Organizations

1.1.3 Why SamanyaSeva?

The reason why we thought of creating a CRM system of our own because we wanted to get a very specific idea of what it should do, and how it should work. These systems are a necessity nowadays for business areas. SamanyaSeva can provide service in many fields and can maintain the record with an ease. It enhances the productivity of business. Some of it's benefits:

- Improved Information arrangement
- Enhance Productivity
- Improved Customer Support
- Task Automation
- Team Work Efficiency
- Improved Reporting & Analysis

1.2 Project Category

Categories and category groups are organized by types. These types correspond to the transaction types that are available in Project management and accounting.
SamanyaSeva comes under the category of Application or System Development. The project mainly falls under the domain of Industry Oriented.

1.3 Objective of Project

SamanyaSeva is a CRM cum ERP Application and the main objectives of this project is to :

1. Enhancing the operational efficiency of business resources.
2. To make an open source project in which people can contribute and learn.
3. To follow modular approach for easy configuration.

1.4 Problem Formulation

As the market is rising rapidly, so is the demand and technology. SamanyaSeva will provide a platform to business oriented marketers, Education Sectors etc. to store their important data at a single place which provides alot of benefits. The USP of this project will be it's modular approach and catalogue.

1.5 Need and Significance

- Better customer satisfaction.
- Improved supplier Performance.
- Improved Resource Utility.

1.6 The Existing System

There are few existing systems which do the task like Busy, SAP or other softwares but they don't have few features which are here in this system. Migration of existing data to the new ERP systems is difficult (or impossible) to achieve. Integrating ERP systems with other stand alone software systems is equally difficult (if possible). These activities may consume a lot of time, money & resources. Moreover, these system were not open source and free web based software. SamanyaSeva uses Django which makes it easier to migrate from one type of database to other. All exiting systems suffers from at least one of the following:

1.6.1 Limitations of previous system

- No Modular Approach.
- No Independent modules such as Catalog.
- Can't be easily configured for different purposes.

1.7 Technologies Used

1. Web Development languages (JS, HTML, CSS)
2. Django (Python Framework)
3. Bootstrap
4. MySQL
5. Apache
6. Hovercraft
7. LaTeX
8. Git

1.8 Unique Features of Proposed System

- Modular Approach
- Proper Catalog
- Easy Configuration
- Independent Modules

Requirement Analysis and System Specification

2.1 Feasibility Study

This study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness.
A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Carrying out a feasibility study involves information assessment, information collection and report writing. The information assessment phase identifies the information that is required to answer the three questions set out above.
Once the information has been identified, you should question information sources to discover the answers to these questions Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

A feasibility study is designed to provide an overview of the primary issues related to a business idea. The purpose is to identify any make or break issues that would prevent your business from being successful in the marketplace. In other words, a feasibility study determines whether the business idea makes sense. A thorough feasibility analysis provides a lot of information necessary for the business plan. For example, a good market analysis is necessary in order to determine the project's feasibility. This information provides the basis for the market section of the business plan.

The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards.
Objectives of feasibility study are listed below:

- To analyze whether the software will meet organizational requirements.
- To determine whether the software can be implemented using the current technology and within the specified budget and schedule.
- To determine whether the software can be integrated with other existing software.

2.2 Types of Feasibility

2.2.1 Technical Feasibility

Technical feasibility is one of the first studies that must be conducted after the project has been identified. In large engineering projects consulting agencies that have large staffs of engineers and technicians conduct technical studies dealing with the projects. In individual agricultural projects financed by local agricultural credit corporations, the technical staff composed of specialized agricultural engineers, irrigation and construction engineers, and other technicians are responsible for conducting such feasibility studies.
The Technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system. This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project. When writing a feasibility report, the following should be taken to consideration:

- A brief description of the business to assess more possible factors which could affect the study.
- The part of the business being examined.
- The human and economic factor.
- The possible solutions to the problem.

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed. Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements. A Technical feasibility also performs the following tasks.

- Analyzes the technical skills and capabilities of the software development team members.
- Determines whether the relevant technology is stable and established.
- Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?

- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using PHP the project is technically feasible for development.

SamanyaSeva is technically feasible as it is built up using various open source technologies and can run on any platform.

2.2.2 Economic Feasibility

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/ benefits analysis.

Economic feasibility is the cost and logistical outlook for a business project or endeavor. Prior to embarking on a new venture, most businesses conduct an economic feasibility study, which is a study that analyzes data to determine whether the cost of the prospective new venture will ultimately be profitable to the company. Economic feasibility is sometimes determined within an organization, while other times companies hire an external company that specializes in conducting economic feasibility studies for them. The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require. Economic feasibility determines whether the required software is capable of generating financial gains for an organization. In addition, it is necessary to consider the benefits that can be achieved by developing the software. Software is said to be economically feasible if it focuses on the issues listed below.

- Cost incurred on software development to produce long-term gains for an organization.
- Cost required to conduct full software investigation (such as requirements elicitation and requirements analysis).
- Cost of hardware, software, development team, and training.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Economic analysis is the most frequently used method to determine the cost/benefit factor for evalu- ating the effectiveness of a new system. In this analysis we determine whether the benefit is gained according to the cost invested to develop the project or not. If benefits outweigh costs, only then the decision is made to design and implement the system. It is important to identify cost and benefit factors, which can be categorized as follows:

- Development Cost
- Operation Cost

This System is Economically feasible with 0 Development and Operating Charges as it is developed in Qt Framework and Octave which is open source technology and is available free of cost on the internet.

Since the system will be developed as part of project work, there will be no manual cost to spend for the proposed system. Also all the resources are already available, it gives an indication that the application is economically possible for development.

2.2.3 Safety Feasibility

It refers to an analysis of whether a project is capable of being implemented and operated safely with minimum adverse effects on the environment.

2.2.4 Operational Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date,Âăcorporate culture, and existing business processes.

Operational feasibility is a measure of how well a project solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. All the operations performed in the software are very quick and satisfy all the requirements.

2.2.5 Legal Feasibility

It investigates if the proposed system conflicts with legal requirements like data protection acts or social media laws.

2.2.6 Technological Feasibility

Technological feasibility is carried out to determine whether the project has the capability, in terms of software, hardware, personnel to handle and fulfill the user requirements. The assessment is based on an outline design of system requirements in terms of Input, Processes, Output and Procedures. Automated Building Drawings is technically feasible as it is built up using various open source technologies and it can run on any platform.

2.2.7 Behavioral Feasibility

Behavioral feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. It is a measure of how well the solution of problems or a specific alternative solution will work in the organization. It is also measure of how people feel about the system. If the system is not easy to operate, than operational process would be difficult. The operator of the system should be given proper training. The system should be made such that the user can interface the system without any problem.

To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters such as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviors are to be realized. A system design and development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters.
A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phasesThis feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks.

- Determines whether the problems anticipated in user requirements are of high priority.
- Determines whether the solution suggested by the software development team is acceptable.
- Analyzes whether users will adapt to a new software.
- Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.

This includes the following questions:

- Is there sufficient support for the users?

- Will the proposed system cause harm?
- The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.3 Software Requirement Specification Document

It is a basically a document or contract that's signed between the system users and system developers. A Software Requirements Specification (SRS) is a document that describes the nature of a project, software or application. In simple words, SRS document is a manual of a project provided it is prepared before you kick-start a project/application. This document is also known by the names SRS report, software document.

2.3.1 Data Requirements

- User Data
- Client Information
- Department Information

2.3.2 Dependability Requirements

- Python and Django
- Web Server like Apache
- Mysql Database

2.3.3 Look and feel Requirements

- Front-end: HTML, CSS, Bootstrap, Django
- Back-end: Django, MySQL

2.3.4 Functional Requirements

- **Specific Requirements**: This phase covers the whole requirements for the system. After understanding the system we need the input data to the system then we watch the output and determine whether the output from the system is according to our requirements or not. So what we have to input and then what we'll get as output is given in this phase. This phase also describe the software and non-function requirements of the system.
- **Input Requirements of the System**

1. Addition of Users.
2. Content in different modules.

- **Output Requirements of the System**

1. Final output after addition of content in web module.
2. Generates Bills, amounts etc.

- **Special User Requirements**

1. It can save data of every user.
2. It can take bulk input.

- **Software Requirements**

1. Framework: Django
2. Package: Python
3. Server: Apache2
4. Database: Mysql
5. Operating System: Cross Platform

2.3.5 Non functional requirements

1. Scalability: System can handle a number of users. For e.g., handling around hundreds of users at the same time.
2. Usability: Simple user interfaces that a layman can understand.
3. Speed: Processing input can be done in reasonable time.

2.3.6 Security requirements

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

2.4 Validation

Validations are applied on every field in this project.
Data Validation in computer science, ensuring that data inserted into an application satisfies defined formats and other input criteria Few validation areas are:

- GST Number
- Date & Email

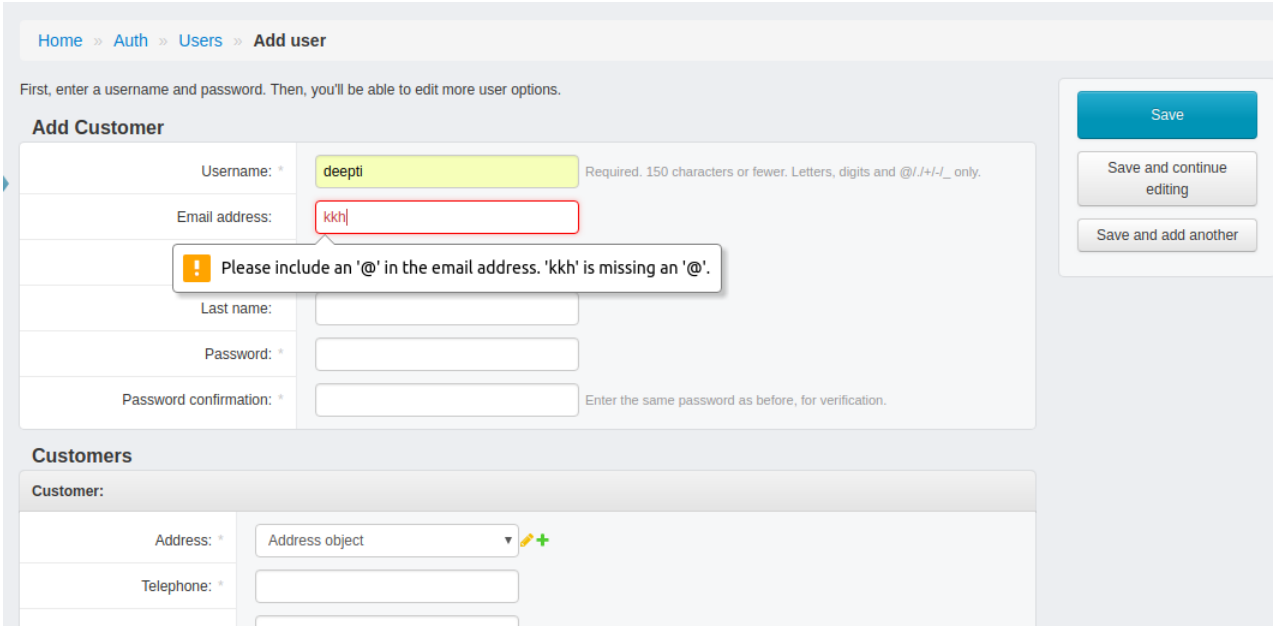


Figure 2.1: Email Validation

- Telephone
- Admin Access over Bills only
- Mandatory Fields like Group name for inserting data

Figure 2.2: Mandatory Field Validation

2.5 Expected Hurdles

This system is made with a basic idea of modular approach. Still, it needs some enhancement (which every project needs at a certain stage).
Cores of SamanyaSeva are Bills, Print and Catalog. But with huge amount of usage and data, oter fields like Report, Program Letter etc. can communicate with the cores of Librehatti which needed to be improved in future.

2.6 SDLC Model to be used

SamanyaSeva basically has modular approach. ALmost every module is independent of each other.
Hence, this project is based on Agile Model. Agile Model is the combination of Iterative and Incremental models. Agile Methods break the product into small incremental builds. These builds are provided in iterations.
In this incremental model, the whole requirement is divided into various builds. During each iteration, the development module goes through the requirements, design, implementation and testing phases. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is ready as per the requirement.

Agile model focus on process adaptability and customer satisfaction by rapid delivery of working software product. Every iteration involves cross functional teams working simultaneously on various areas like:

- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing
- Acceptance Testing

System Design

3.1 Design Approach

Modular approach has been used for designing this application. Every Module is independent of each other.

3.2 Detail Designing using Tools

3.2.1 DFDs

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs of CMS is as following-:

1. Data flow LEVEL 0 fig 3.1
2. Data flow LEVEL 1 fig 3.2
3. Data flow LEVEL 2 fig 3.3

A level 0 data flow diagram (DFD), also known as a context diagram, shows a data system as a whole and emphasizes the way it interacts with external entities. This DFD level 0 example shows how such a system might function within a typical retail business.

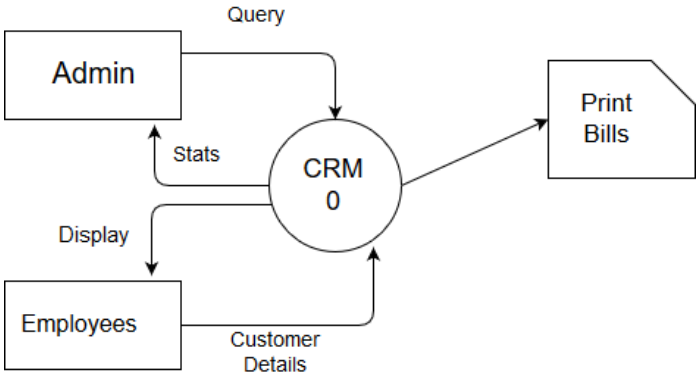


Figure 3.1: Data flow LEVEL 0

DFD level 1 breaks down the main processes into subprocesses that can then be analyzed and improved on a more intimate level.

Now the Basic Data Flow:-

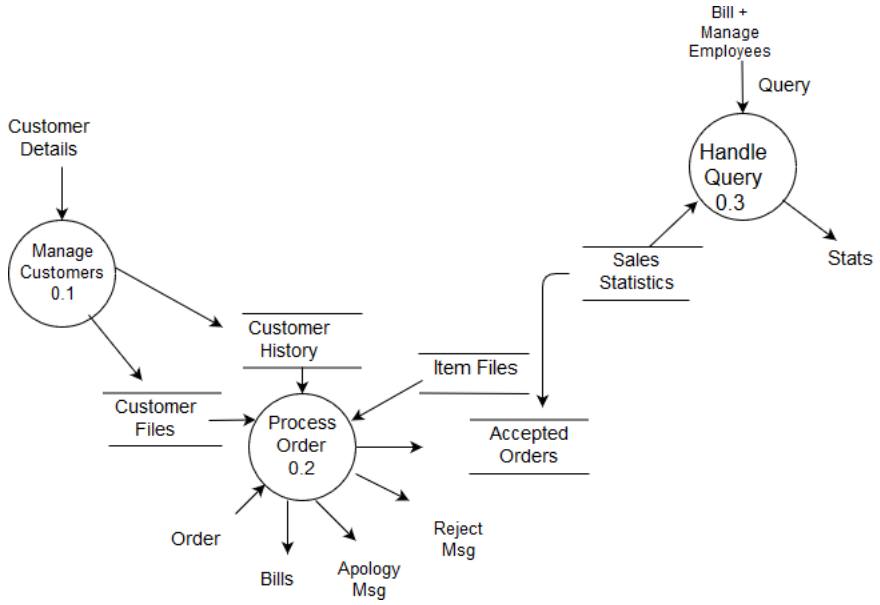


Figure 3.2: Data Flow LEVEL 1

A level 2 data flow diagram (DFD) offers a more detailed look at the processes that make up a system than a level 1 DFD does.

Further flow:-

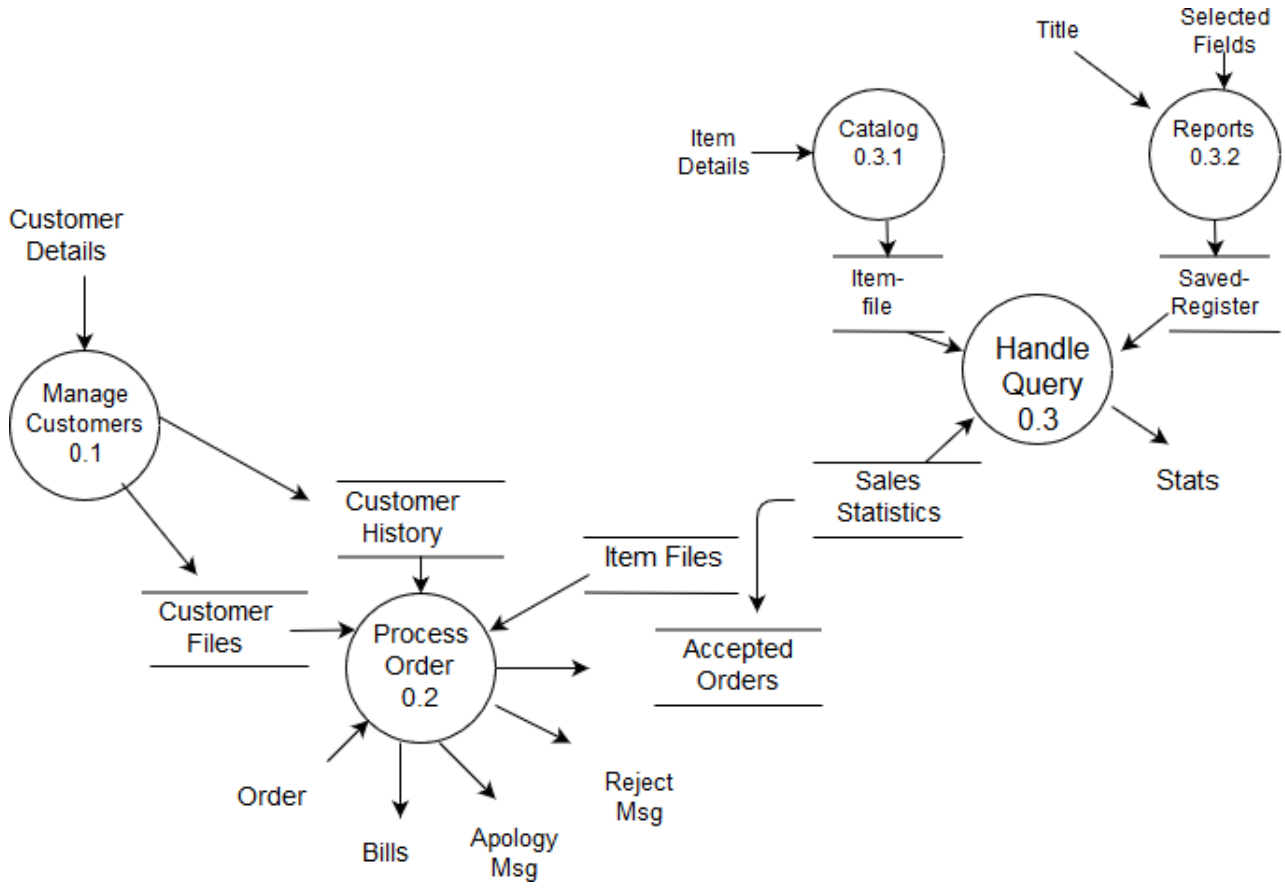


Figure 3.3: Data Flow LEVEL 2

3.2.2 Flowchart

A flowchart is a type of diagram that represents an algorithm, work flow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. Flowcharts are used in designing and documenting simple processes or programs. Like other types of diagrams, they help visualize what is going on and thereby help understand a process, and perhaps also find flaws, bottlenecks, and other less-obvious features within it. There are many different types of flowcharts, and each type has its own repertoire of boxes and notational conventions. The two most common types of boxes in a flowchart are:

1. A processing step, usually called activity, and denoted as a rectangular box.
2. A decision, usually denoted as a diamond.
3. The terminator symbol marks the starting or ending point of the system. It usually contains the word "Start" or "End".
4. A printed document or report is represented by a Document Symbol.

Following is flowchart of system showing flow of control and Data in the software-:

3.3 UI Flow Diagram

User interface-flow diagrams - also called storyboards, interface-flow diagrams, windows navigation diagrams, and context-navigation maps - enable you to model the high-level relationships between major user interface elements and thereby ask fundamental usability questions. UI Flow diagram tells how user will perceive different interface on click of different buttons or trigger. The rectangular blocks represent entities and Arrow represents change of view from one to another on the bases of button clicked mentioned near arrow.

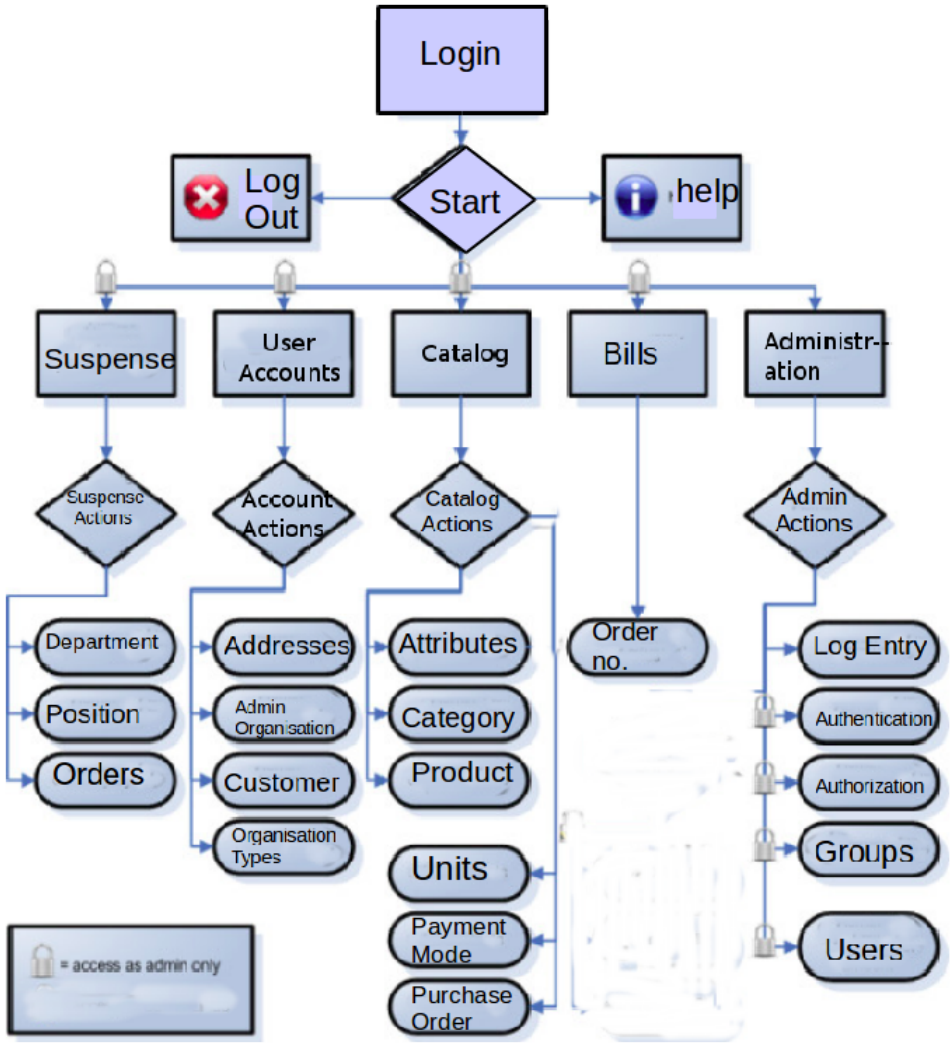


Figure 3.4: Flow diagram

3.4 Database Design

3.4.1 ER Diagram

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how âĂIJentitiesâĂİ such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research.

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

3.5 Methodology

This includes the steps to be followed to achieve the objective of the project during the project development.

	Name	Duration	Start	Finish
	Planning	7 days	**4/1/18 8:00 AM**	**12/1/18 5:00 PM**
	Aim	3 days	4/1/18 8:00 AM	8/1/18 5:00 PM
	Discussion about field	4 days	9/1/18 8:00 AM	12/1/18 5:00 PM
	Requirement Analy	8 days	**12/1/18 8:00 AM**	**23/1/18 5:00 PM**
	functional requiremnt	4 days	12/1/18 8:00 AM	17/1/18 5:00 PM
	non-functional require	4 days	18/1/18 8:00 AM	23/1/18 5:00 PM
	Data Gathering	3 days	**23/1/18 8:00 AM**	**25/1/18 5:00 PM**
	Fields functions	3 days	23/1/18 8:00 AM	25/1/18 5:00 PM
	Study	5 days	**25/1/18 8:00 AM**	**31/1/18 5:00 PM**
	Existing Systems	3 days	25/1/18 8:00 AM	29/1/18 5:00 PM
	Functionalities	3 days	29/1/18 8:00 AM	31/1/18 5:00 PM
	Implementation	18 days	**29/1/18 8:00 AM**	**21/2/18 5:00 PM**
	Front-End	10 days	29/1/18 8:00 AM	9/2/18 5:00 PM
	Database	6 days	9/2/18 8:00 AM	16/2/18 5:00 PM
	User Interface	3 days	26/2/18 8:00 AM	28/2/18 5:00 PM
	Progress	40 days	1/3/18 8:00 AM	25/4/18 5:00 PM

Figure 3.5: Plan

Implementation, Testing, and Maintenance

4.1 Introduction to Languages, IDE's, Tools and Technologies

4.1.1 Introduction to Languages

Front End languages are language that are used to give better user experince and user interface. These mainly include HTML, CSS, JS. Some Frameworks like Bootstrap, Django are also used with these basic languages. Database used with this project is MySQL.

4.1.1.1 HTML

Figure 4.1: HTML5 logo

HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Along with CSS, and HTML is a cornerstone technology, used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.
Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.
HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
```

```
    <p>Hello world!</p>
  </body>
</html>
```

4.1.1.2 CSS

Figure 4.2: CSS logo

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language.Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media.
Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.
CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts.
This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content, such as semantically insignificant tables that were widely used to format pages before consistent CSS rendering was available in all major browsers.
CSS makes it possible to separate presentation instructions from the HTML content in a separate file or style section of the HTML file. For each matching HTML element, it provides a list of formatting instructions

```
p {
    color: red;
    text-align: center;
}
```

Figure 4.3: Javascript logo

4.1.1.3 Javascript

JavaScript is a high-level, dynamic, untyped, and interpreted programming language. It has been standardized in the ECMAScript language specification. Alongside HTML and CSS, it is one of the three essential technologies of World Wide Web content production; the majority of websites employ it and it is supported by all modern web browsers without plug-ins. JavaScript is prototype-based with first-class functions, making it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. It has an API for working with text, arrays, dates and regular expressions, but does not include any I/O, such as networking, storage or graphics facilities, relying for these upon the host environment in which it is embedded.

4.1.1.4 jQuery

Figure 4.4: jQuery logo

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It is free, open-source software using the permissive MIT License. The purpose of jQuery is to make it much easier to use JavaScript on a website. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

Figure 4.5: Djangologo

4.1.2 Introduction to Frameworks

4.1.2.1 Django

Django is a free and open-source web framework, written in Python, which follows the model-view-template architectural pattern. It is maintained by the Django Software Foundation, an independent organization. It is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.

4.1.2.2 Bootstrap

Figure 4.6: BootStrap logo

Bootstrap is a free and open-source collection of tools for creating websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.
It aims to ease the development of dynamic websites and web applications.
Bootstrap is a front end framework, that is, an interface for the user, unlike the server-side code which resides on the "back end" or server.

4.1.3 Tools & Technologies

4.1.3.1 Doxygen

Doxygen is a documentation generator, a tool for writing software reference documentation. The documentation is written within code, and is thus relatively easy to keep up to date. Doxygen can cross reference documentation and code, so that the reader of a document can easily refer to the actual code.

Figure 4.7: Doxygen logo

Features

Doxygen is a tool to create a documentation for your program/project written in the languages like C, C++, Java, python and so on. It reads the well formatted and special doxygen comments to create the required documentation. This documentation is very important for the new developers who want to help in the development of the project. Documentation is one of the main pillar of an open-source project.

- Requires very little overhead from the writer of the documentation. Plain text will do, Markdown is support, and for more fancy or structured output HTML tags and/or some of doxygen's special commands can be used.
- Cross platform: Works on Windows and many Unix flavors (including Linux and Mac OS X).
- Comes with a GUI frontend (Doxywizard) to ease editing the options and run doxygen. The GUI is available on Windows, Linux, and Mac OS X.
- Automatically generates class and collaboration diagrams in HTML (as clickable image maps) and LaTeX (as Encapsulated PostScript images).
- Allows grouping of entities in modules and creating a hierarchy of modules.
- Doxygen can generate a layout which you can use and edit to change the layout of each page.
- Can cope with large projects easily.

Installation

Doxygen can be installed using following commands:

$ git clone https://github.com/doxygen/doxygen.git

$ cd doxygen

$ ./configure

$ make

This will install Doxygen on your pc or laptop.
You can create the documentation using the graphical user interface (GUI) or console mode.

While writing the comments we have to follow a pattern with the tags i.e. before every tag we should have something special so that Doxygen can understand what are we creating. Actually Doxygen read these tags and place them at special location in the generated output. So, we have to specify them explicitly.

4.1.4 LaTeX

Figure 4.8: LaTeX

LaTeXis a document markup language and document preparation system for the TeX typesetting program. Within the typesetting system, its name is styled as LaTeX.

To run LATEX on your own computer, you need to use a latex distribution. A distribution includes a latex program and (typically) several thousand packages.

- On Windows: MikTEX or TEXLive
- On Linux: TEXLive
- On Mac: MacTEX

Apart from the pdf file; lat.aux, lat.log, lat.pdf files are created by default.

- AUX is a data file format used by Latex AUX is a data file format used by LaTeX. LaTeX is a macro package which uses TeX typesetting language in its documents. AUX files contain information used for cross-referencing, and is also used to transport information from one compiler run to the next.
- Some of the compilers are pdftex, Xelatex, Lualatex etc.
- A log file is usually a flat text file that contains a list of events that happend when a program was running, with one event on each line. Often times errors are recorded in log files.
- .pdf: The common output format for your document. Created by pdflatex/ xelatex

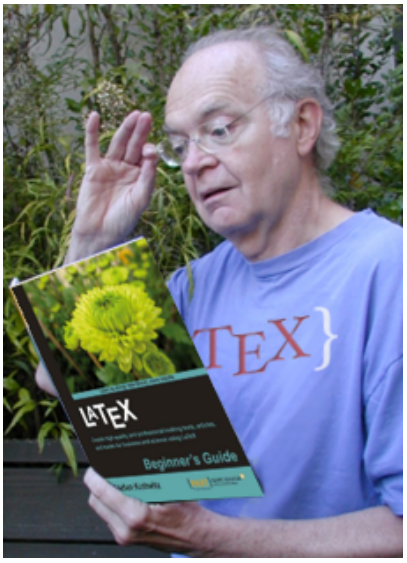


Figure 4.9: Donald Knuth, Inventor Of TEX typesetting system

LATEXwas first developed in 1985 by Leslie Lamport. In preparing a LATEX document, the author specifies the logical structure using familiar concepts such as chapter, section, table, figure, etc., and lets the LATEX system worry about the presentation of these structures. It therefore encourages the separation of layout from content while still allowing manual typesetting adjustments where needed.

4.1.4.1 MySQL

Figure 4.10: MySQL Database

MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.
SQL is a standard language for storing, manipulating and retrieving data in databases.

4.1.4.2 HoverCraft

Hovercraft's power comes from the combination of reStructuredText's convenience with the cool of impress.js, together with a flexible and powerful solution to position the slides.

It creates wonderful presentations by providing an ease with the effects like Rotation, Zoom etc.

4.1.4.3 Git & GitHub

Figure 4.11: Github Logo

GitHub is a Git repository web-based hosting service which offers all of the functionality of Git as well as adding many of its own features. Unlike Git which is strictly a command-line tool, Github provides a web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as wikis, task management, and bug tracking and feature requests for every project.

Installation

Installation of git is a very easy process. The current git version is: 2.0.4. Type the commands in the terminal:

$ sudo apt-get update

$ sudo apt-get install git

This will install the git on your pc or laptop.

Various Git Commands

Git is the open source distributed version control system that facilitates GitHub activities on your laptop or desktop. The commonly used Git command line instructions are:-
For starting a new repository or obtaining from an exiting URL

$ git init [project-name]
Creates a new local repository with the specified name

$ git clone [url]
Downloads a project and its entire version history

4.2 Project Scheduling

The project schedule is the tool that communicates what work needs to be performed, which resources of the organization will perform the work and the timeframes in which that work needs to be performed. The project schedululing reflects all of the work associated with delivering the project on time.

4.2.1 Gantt Chart

A Gantt chart is a type of bar chart that illustrates a project schedule. This chart lists the tasks to be performed on the vertical axis, and time intervals on the horizontal axis.
A Gantt chart is a graphical depiction of a project schedule. A Gantt chart is a type of bar chart that shows the start and finish dates of several elements of a project that include resources, milestones, tasks and dependencies.

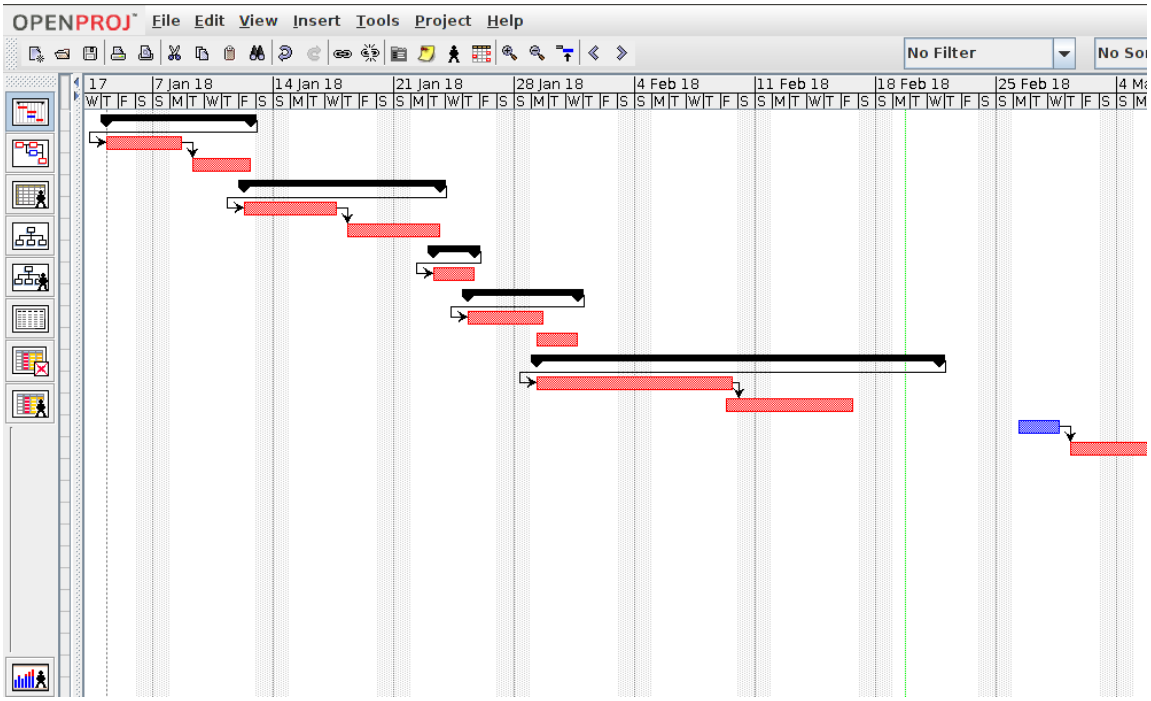


Figure 4.12: Gantt Chart

Gantt charts are most commonly used for tracking project schedules. For this it is useful to be able to show additional information about the various tasks or phases of the project, for example how the tasks Gantt charts may be simple versions created on graph paper or more complex automated versions created using project management applications.

4.2.2 Network Diagram

A Network Diagram is a visual representation of a project's schedule. A network diagram in project management is useful for planning and tracking the project from beginning to finish. It represents a project's critical path as well as the scope for the project.

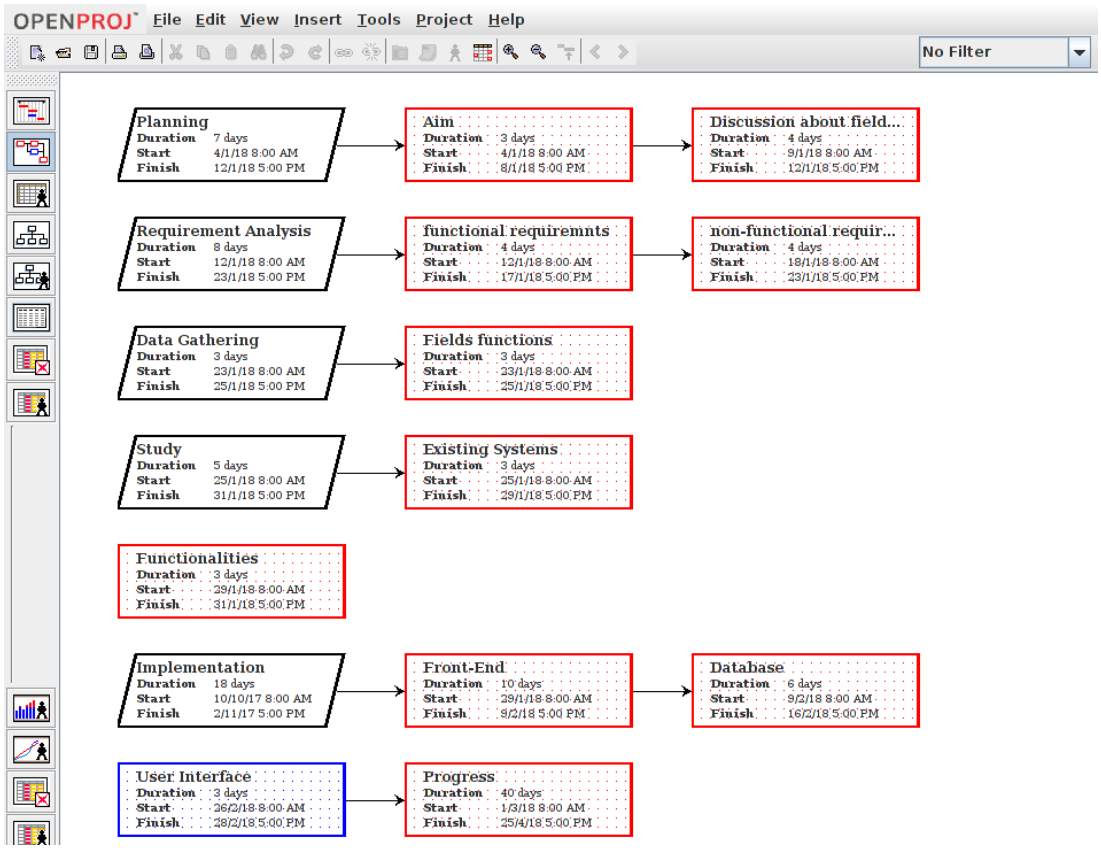


Figure 4.13: Network Chart

4.3 Testing

Testing a program consists of providing the program with a set of test inputs (or test cases) and observing if the program behaves as expected. If the program fails to behave as expected, then the conditions under which failure occurs are noted for later debugging and correction.

This software had been taken through rigorous test to fully found potential causes of error and system failure and full focus have been given to cover all possible exceptions that can occur and cause failure of the software.

As this software is based on intensive background process it have been taken care that if correct input and email address are given then processing of user job can even continue or a least automatically restart even after server shuts down or even crash.

Overview of SamanyaSeva	
Module	Working
UserAccounts	Users Data
Bills	Billing Module
Catalog	A basic menu
Reports	Maintained Registers
Suspense	Payment
Administration	Admin Power

Table 4.1: Modules

Validations	
Validation	Error
Email	With the use of @
GST Number	Alpha-Numeric Value
Telephone	Numeric Only
Date Format	Can't add in other formats
Mandatory Fields	Can't add data without defining Group name
Admin Power	Can't access Bills module without Admin rights

Table 4.2: Validations

Results and Discussions

5.1 User Interface Representation

UI design of our project is really simple which can be easily understood by a layman.

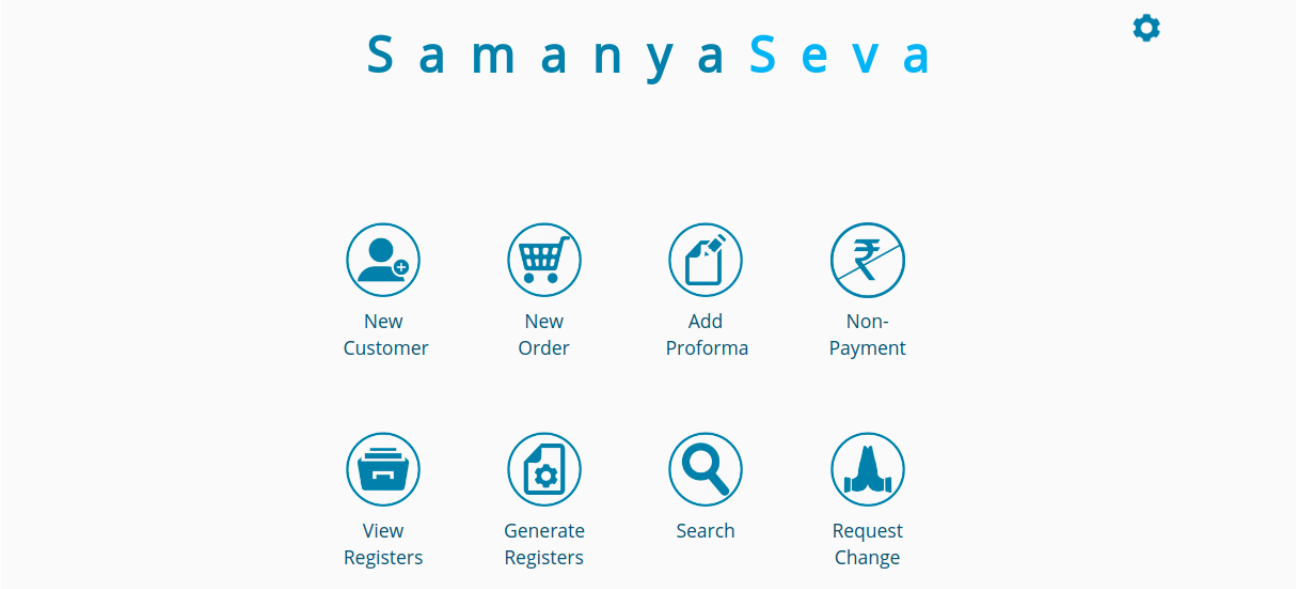


Figure 5.1: UI

5.2 Brief Description of Various Modules of the system

5.2.1 UserAccounts

In this module, admin can check user data which were added while adding a new customer/user. Addition of users contains many fields like First and last name, Telephone, GST Number, Username and Password.

5.2.2 Bills

User can add a particular payment mode, as per user convinience.

5.2.3 Suspense

One can add the record department wise and can have a look over that.

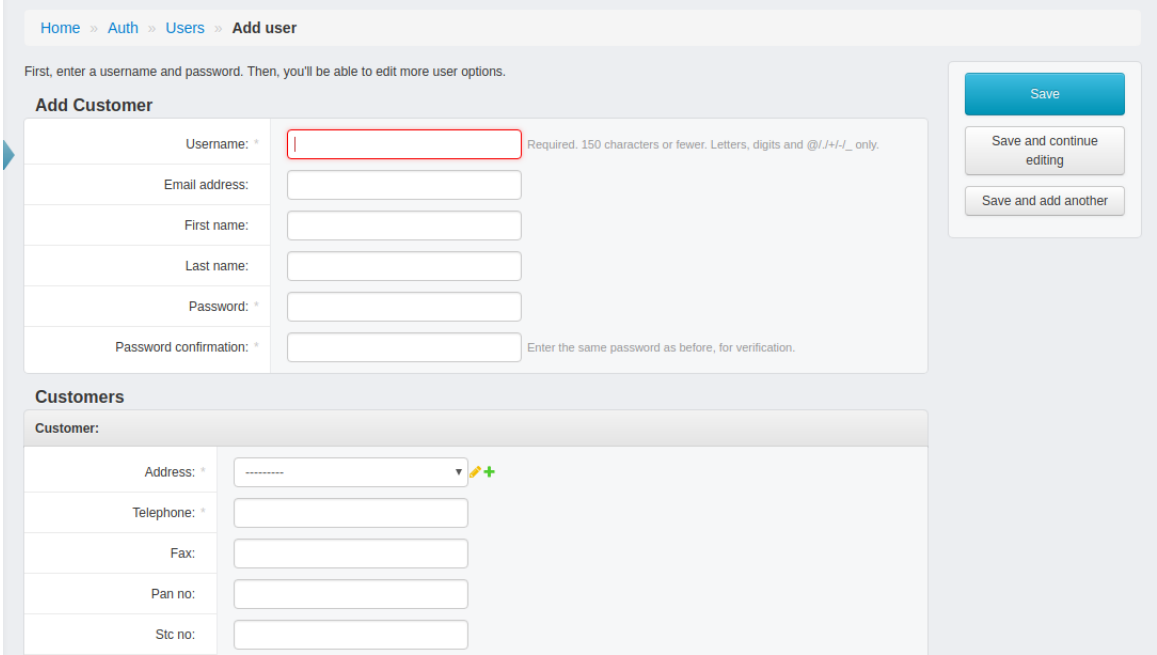


Figure 5.2: User Interface

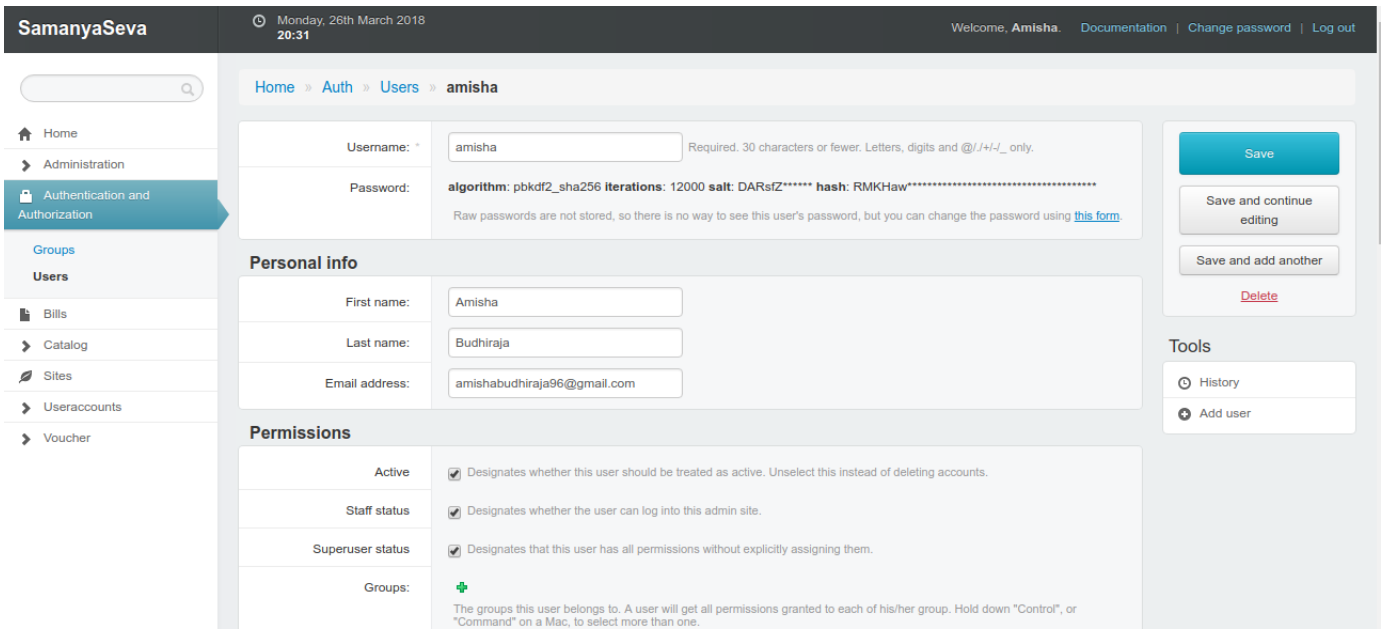


Figure 5.3: UserAccounts

5.2.4 Reports

Here, Give a unique name to your report and have a excel type ready to use report in return. There are many fields provided for user's ease such as:

- First name
- Last name

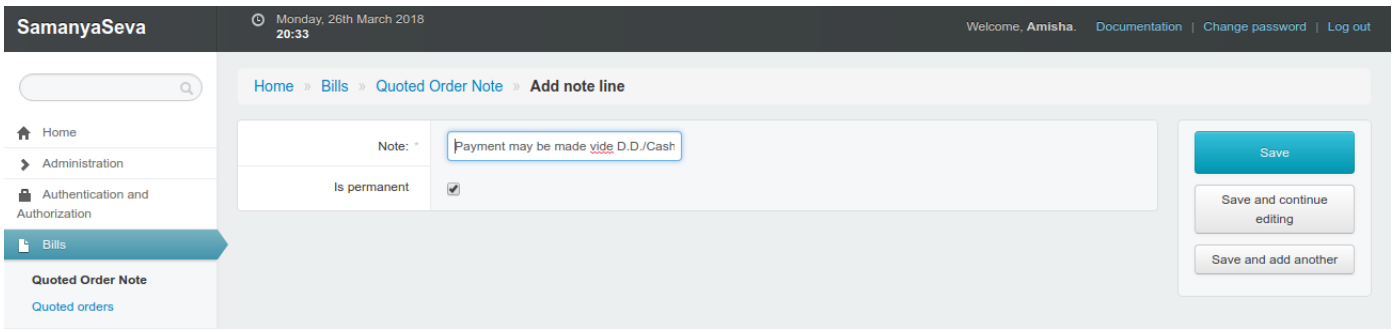


Figure 5.4: Bills

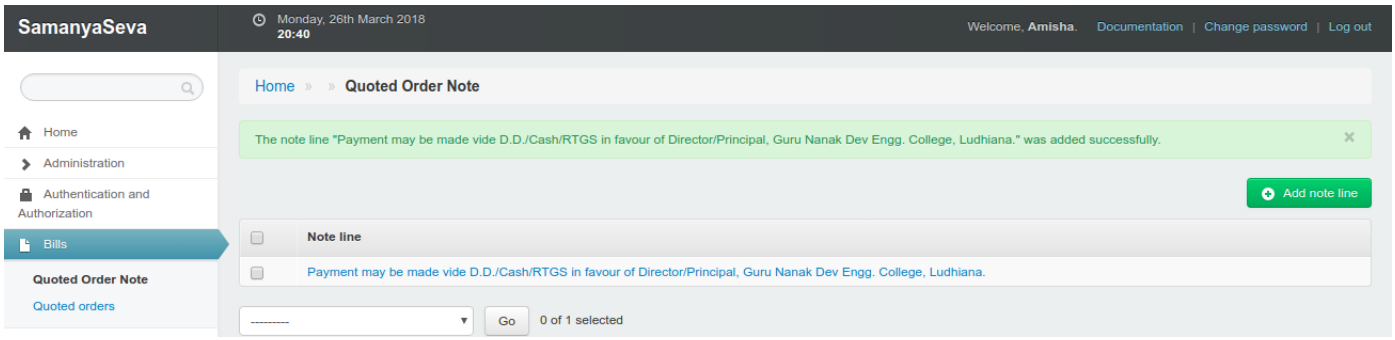


Figure 5.5: Billing Module

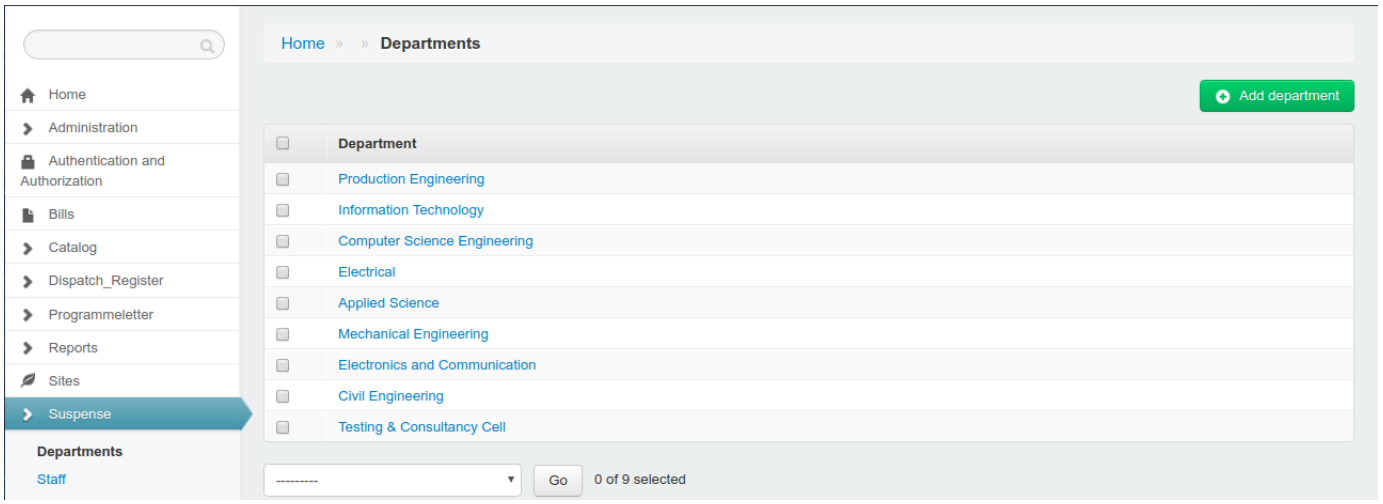


Figure 5.6: Departments

- Title
- Street Address
- District
- Province

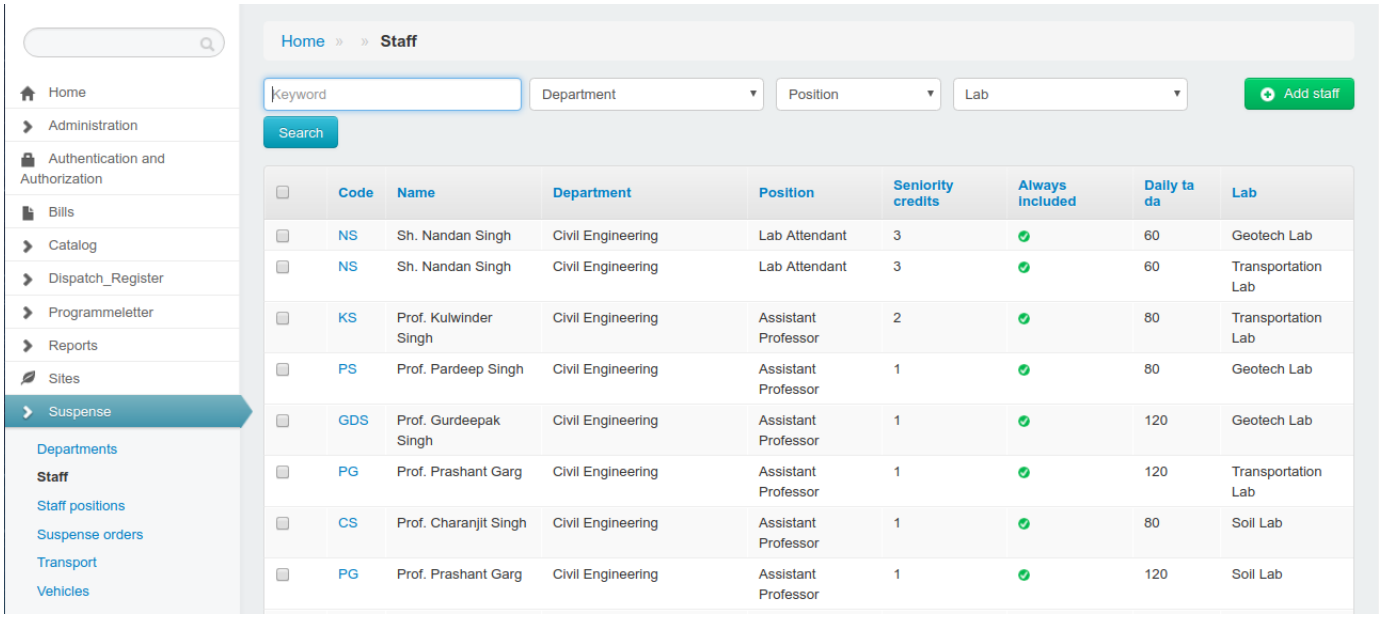


Figure 5.7: Suspense Module

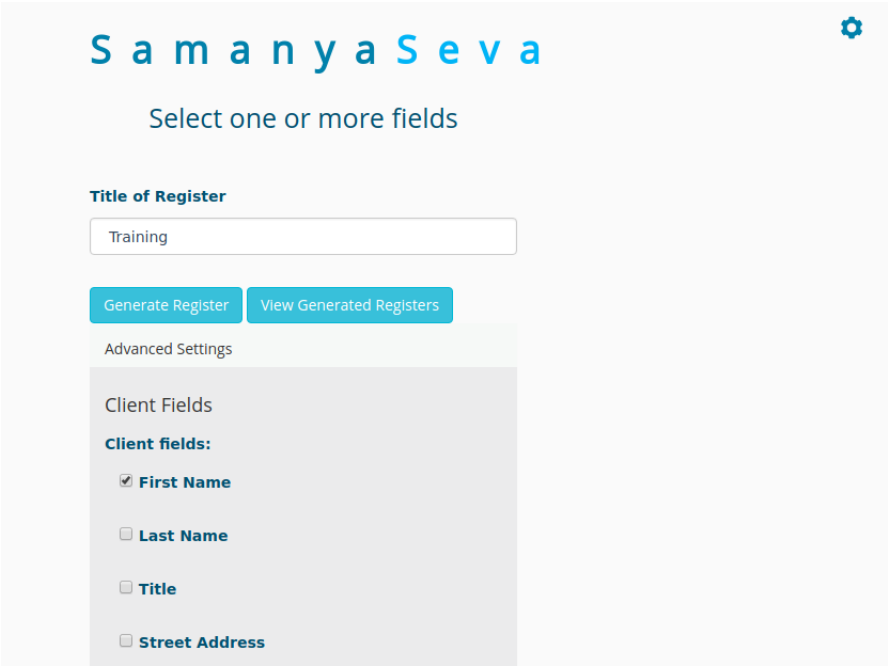


Figure 5.8: Generate Report

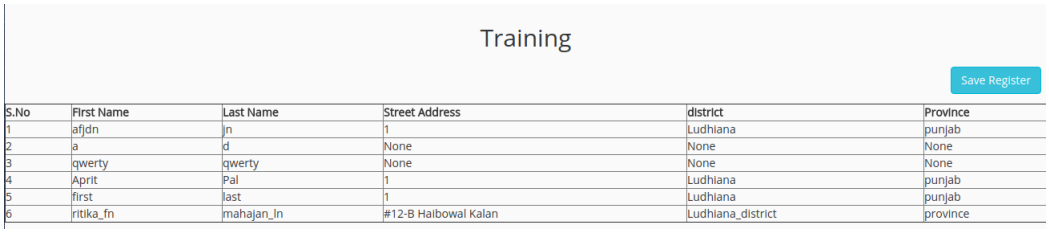


Figure 5.9: Generated Report

5.2.5 Generate Registers

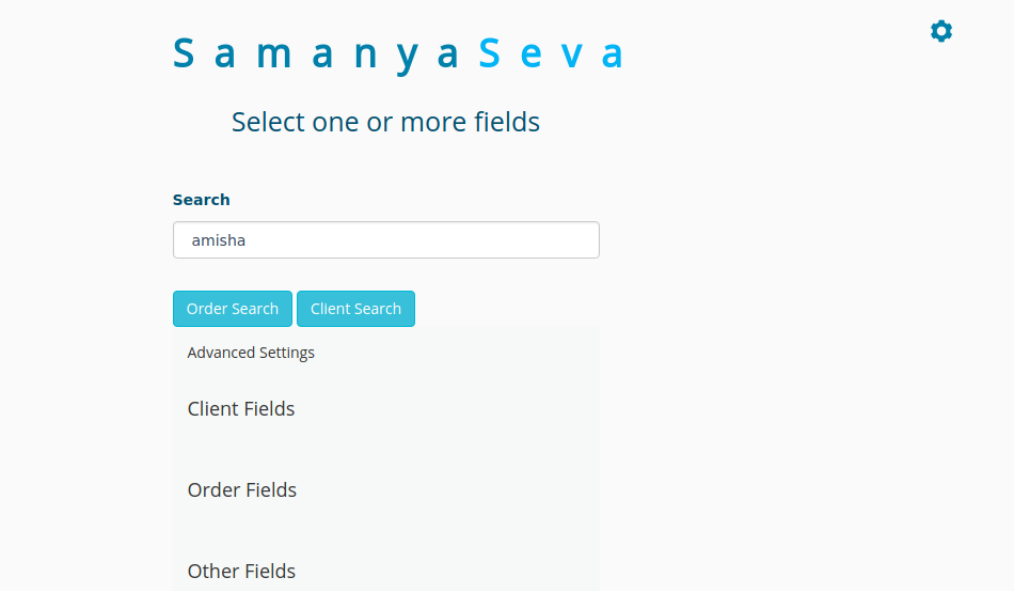


Figure 5.10: Search Client

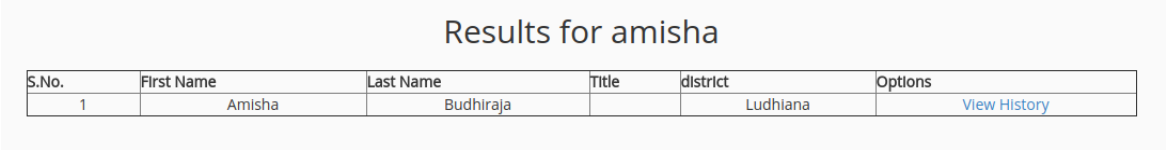
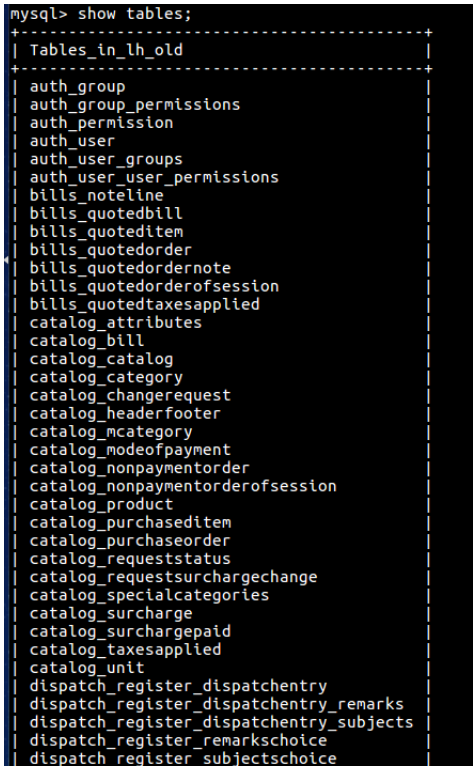


Figure 5.11: Register

5.3 Snapshots of Database Tables with brief description

This depicts almost all the tables been created for this project. There are different tables made for different modules.

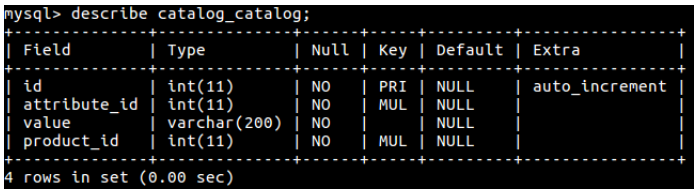
A proper naming structure with standards have been used for easy understanding about the working of this system. There are different tables shown below like for Bills module, User Module etc.



```
mysql> show tables;
+-----------------------------------------+
| Tables_in_lh_old                        |
+-----------------------------------------+
| auth_group                              |
| auth_group_permissions                  |
| auth_permission                         |
| auth_user                               |
| auth_user_groups                        |
| auth_user_user_permissions              |
| bills_noteline                          |
| bills_quotedbill                        |
| bills_quoteditem                        |
| bills_quotedorder                       |
| bills_quotedordernote                   |
| bills_quotedorderofsession              |
| bills_quotedtaxesapplied                |
| catalog_attributes                      |
| catalog_bill                            |
| catalog_catalog                         |
| catalog_category                        |
| catalog_changerequest                   |
| catalog_headerfooter                    |
| catalog_mcategory                       |
| catalog_modeofpayment                   |
| catalog_nonpaymentorder                 |
| catalog_nonpaymentorderofsession        |
| catalog_product                         |
| catalog_purchaseditem                   |
| catalog_purchaseorder                   |
| catalog_requeststatus                   |
| catalog_requestsurchargechange          |
| catalog_specialcategories               |
| catalog_surcharge                       |
| catalog_surchargepaid                   |
| catalog_taxesapplied                    |
| catalog_unit                            |
| dispatch_register_dispatchentry         |
| dispatch_register_dispatchentry_remarks |
| dispatch_register_dispatchentry_subjects |
| dispatch_register_remarkschoice         |
| dispatch_register_subjectschoice        |
```

Figure 5.12: List of Tables

Below is an important module which distinguishes our project from many i.e Catalog. It's basically a menu driven module.



```
mysql> describe catalog_catalog;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| id           | int(11)      | NO   | PRI | NULL    | auto_increment |
| attribute_id | int(11)      | NO   | MUL | NULL    |                |
| value        | varchar(200) | NO   |     | NULL    |                |
| product_id   | int(11)      | NO   | MUL | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

Figure 5.13: Catalog

It's table of exisiting users with fields like First and last name, username etc.

```
mysql> select * from auth_user;
+----+------------------------------------------------------------------------------------------------+--------------------+-------------+-------------+-----
------+-----------+----------------------------+----------+-----------+--------------------+
| id | password                                                                                       | last_login         | is_superuser | username    | firs
t_name | last_name  | email                      | is_staff | is_active | date_joined        |
+----+------------------------------------------------------------------------------------------------+--------------------+-------------+-------------+-----
------+-----------+----------------------------+----------+-----------+--------------------+
|  1 | pbkdf2_sha256$12000$DARsfZX79q74$RMKHawIEUIjdJDnM/EeSgWWOfUePEw742uyIO+fQU6k= | 2018-03-13 08:50:19 |            1 | amisha      | Amis
ha     | Budhiraja  | amishabudhiraja96@gmail.com |        1 |         1 | 2017-11-16 12:25:25 |
|  2 | pbkdf2_sha256$12000$0IU2L8WMxLxR$8xcFkMSo4H/SbAiDmWR6uHGm4ogNAjghwvE4EukCqGE= | 2017-11-16 14:24:49 |            0 | hello       | afjd
n      | jn         | jfenv@gmail.com            |        0 |         1 | 2017-11-16 14:24:49 |
|  3 | pbkdf2_sha256$12000$0WKAmKpbwqDo$xmlmVOtv+IPjfJUBbhcvkmZFBad1ahlvXbP0X4IYNgE= | 2017-11-17 11:19:37 |            0 | sir         | a
      | d          | a@gmail.com                |        0 |         1 | 2017-11-17 11:19:37 |
|  4 | pbkdf2_sha256$12000$TmjNTVaVEPMK$EszDiuvJRCNXuSo14MxdWU8+AaPqiAqz6QA/L/jEWQ0= | 2017-11-17 11:21:03 |            0 | q           | qwer
ty     | qwerty     | qwerty@gmail.com           |        0 |         1 | 2017-11-17 11:21:03 |
|  5 | pbkdf2_sha256$12000$pFCfkzE8xTsS$mRUSA4FTefZULoxZzIhkXH0QFqJwO3oREet+fnCVw/E= | 2017-11-17 11:30:29 |            0 | sub         | Apri
t      | Pal        | ttc@gmail.com              |        0 |         1 | 2017-11-17 11:30:29 |
|  6 | pbkdf2_sha256$12000$WTKm8TTyFZEv$uTUrGZNPGV4QlzH0ET1acoBTxkep3sYd53G2Um0y7ao= | 2017-11-29 08:20:34 |            0 | try_username | firs
t      | last       | try@gmail.com              |        0 |         1 | 2017-11-29 08:20:34 |
|  7 | pbkdf2_sha256$12000$6e87ZyXjQ8My$zP9cE29EC+YIDu5hgtTuVabgPCHN16Ukj0MqxlTX2ro= | 2017-12-27 18:45:05 |            0 | ritika_un   | riti
ka_fn  | mahajan_ln | ritikimahajan9@gmail.com   |        0 |         1 | 2017-12-27 18:45:05 |
+----+------------------------------------------------------------------------------------------------+--------------------+-------------+-------------+-----
------+-----------+----------------------------+----------+-----------+--------------------+
7 rows in set (0.00 sec)
```

Figure 5.14: Auth User

5.4 BackEnd Representation

```
mysql> describe catalog_catalog;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| id           | int(11)      | NO   | PRI | NULL    | auto_increment |
| attribute_id | int(11)      | NO   | MUL | NULL    |                |
| value        | varchar(200) | NO   |     | NULL    |                |
| product_id   | int(11)      | NO   | MUL | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

Figure 5.15: Catalog

Known for it's performance, reliability and ease: 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, and all five of the top five websites. A no. of tables has been made for each module separately and the database used for the same is MySQL.

5.5

Conclusion and Future Scope

6.1 Future Scope

It's made by considering professional used terms, so that it can be easily set into the market. All the modules are designed independently. SamanyaSeva can be used in organisations, IT firms, Education and business sectors for maintaining records and other important work like Bills etc.

6.2 Conclusion

SamanyaSeva is an open source altenative to official record maintainance systems. A server having this project can be beneficial as it can used for maintainance purpose in colleges and also for work in Organisations. It can save the data of thousand users at a time. There are different modules for different tasks in this, like Billing module, Catalog, Vouchers, Suspense etc.

6.3 Future Scope

It's made by considering professional used terms, so that it can be easily set into the market. All the modules are designed independently. SamanyaSeva can be used in organisations, IT firms, Education and business sectors for maintaining records and other important work like Bills etc.

6.4 Conclusion

SamanyaSeva is an open source altenative to official record maintainance systems. A server having this project can be beneficial as it can used for maintainance purpose in colleges and also for work in Organisations. It can save the data of thousand users at a time. There are different modules for different tasks in this, like Billing module, Catalog, Vouchers, Suspense etc.

Bibliography

[1] Stack Overflow, https://stackoverflow.com/questions/8157511/designig-an-erp-system

[2] LaTeX Beginner's Guide By Stefan Kottwitz

[3] Django, https://www.djangoproject.com/start/

[4] Source Code, https://github.com/amisha2016/samanyaseva

[5] Doxygen, www.doxygen.org