*PROJECT REPORT*

*On*

**CAMPAIGN MANAGEMENT SYSTEM**

*Submitted by*

**DHYAN SHAH** **(IU1641050051)**

*In fulfillment for the award of the degree*

*Of*

*BACHELOR OF TECHNOLOGY*

*In*

*COMPUTER ENGINEERING*



**INSTITUTE OF TECHNOLOGY AND ENGINEERING**

**INDUS UNIVERSITY CAMPUS, RANCHARDA, VIA-THALTEJ**

**AHMEDABAD-382115, GUJARAT, INDIA,**

WEB: www.indusuni.ac.in

MAY 2020

**CANDIDATE’S DECLARATION**

I declare that final semester report entitled “**CAMPAIGN MANAGEMENT SYSTEM**” is my own work conducted under the supervision of the guide **Prof. NAISWITA PARMAR (internal guide)** and **VANDITA DHARIYAL (external guide).**

I further declare that to the best of my knowledge the report for B.Tech. Final semester does not contain part of the work which has been submitted for the award of B.Tech.

Degree either in this or any other university without proper citation.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Candidate’s Signature

DHYAN SHAH (IU1641050051)

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Guide: Prof. NAISWITA PARMAR

Assistant Professor

Department of Computer Engineering,

Indus Institute of Technology and Engineering

INDUS UNIVERSITY– Ahmedabad, Gujarat

**COMPANY CERTIFICATE**

**INDUS INSTITUTE OF TECHNOLOGY AND ENGINEERING**

**COMPUTER ENGINEERING**

**2018 -2019**



**CERTIFICATE**

**Date: 6th May 2020**

This is to certify that the project work entitled “**CAMPAIGN MANAGEMENT SYSTEM”** has been carried out by **DHYAN SHAH** under my guidance in the fulfillment of degree of Bachelor of Technology in **COMPUTER ENGINEERING (Final Year)** of Indus University, Ahmedabad during the academic year 2019 - 2020.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. NAISWITA PARMAR Dr. SEEMA MAHAJAN

Assistant Professor, Head of the Department,

Department of Computer Engineering, Department of Computer Engineering,

I.I.T.E, Indus University I.I.T.E, Indus University

Ahmedabad, Gujarat Ahmedabad, Gujarat

**ACKNOWLEDGEMENT**

Towards the successful completion of my B.Tech. In Computer Engineering final year project training in Gateway Techno labs, I feel greatly obliged to certain specials.

I am sincerely gratitude to Vandita Dhariyal (HR) of GATEWAY GROUP OF COMPANIES, Ahmedabad, who provided me this opportunity and necessary support during the development of the project. I am also grateful to all staff members of this company for providing all the technical assistance and helping in the solving problems arising during developments.

I have no other words to express my sincere thanks to all faculties of Indus University,

Ahmedabad for their kind co-operation and able guidance. Especially to Prof. Naiswita Parmar, who helped us in identifying project definition and without her the project could not be executed.

* Dhyan Shah

**TABLE OF CONTENT**

**Title Page No**

**COVER PAGE…………………………………………………………………… I**

**FIRST PAGE………………………………………………….............................. II**

**CANDIDATE DECLARATION………………………………………….......... III**

**COMPANY CERTIFICATE…………………………………………………… IV**

**COLLEGE CERTIFICATE……………………………………………………. V**

**ACKNOWLEDGEMENT……………………………………………………… Viii**

**CHAPTER 1 INTRODUCTION …………………………………………… 1**

1.1 Project Summary………………………………………………2

1.2 Project Purpose………………………………………………...2

1.3 Project Scope…………………………………………….......... **3**

1.4 Technology and Literature Overview………………………….3

**CHAPTER 2 PROJECT MANAGEMENT ………………………….......... 7**

2.1 Project Planning Objectives……………………………………8

2.2 Project Scheduling……………………………………………...9

**CHAPTER 3 SYSTEM RQUIREMENTS ………………………………… 12**

3.1 User Characteristics…………………………………………….13

3.2 Functional Requirement…………………………………...........13

3.3 Non Functional Requirement…………………………………..15

3.4 Hardware and Software Requirement……………………..........16

**CHAPTER 4 SYSTEM ANALYSIS ………………………………….......... 17**

4.1 Feasibility Study………………………………………………..18

4.2 Project Planning……….…………………………………..........19

**CHAPTER 5 SYSTEM DESIGN…………………………………………… 20**

5.1 Data flow Diagram……………………………………………21

5.2 Use-Case Diagram ………………………………………..........23

5.3 Activity Diagram ………………………………………………25

5.4 Sequence Diagram.…………………………………………….26

5.5 Data Dictionary………………………………………………26

**Title Page No**

5.6 Application Dictionary…………………………………..........29

**CHAPTER 6 TESTING …………………………………………………….. 31**

6.1 Black-Box Testing……………………………………………...32

6.2 White-Box Testing …………………………………………….33

6.3 Test Cases …………………………………………………….34

**CHAPTER 7 SCREEN SHOTS …………………………………………… 36**

**CHAPTER 8 CONCLUSION & FUTURE EXTENSION……..………... 46**

**APPENDICES………………………………………………………………… 48**

**REFERENCES & EXPERINCE……………………………………….......... 50**

**ABSTRACT**

Here CMS means ‘Campaign Management System’. So campaign management system is all about managing campaign where marketing or management team from specific organization can add and manage campaign. Our system will send that specific campaign to customers via mail and then track customer responses and help organization to decide whether campaign was successful or not and represent it in chart format for marketing team and will generate report for management team .As per admin perspective admin can add customer from spreadsheet to database which can be used to send campaigns and also he can modify campaign as needed and admin only will be adding organizations in system as well as users. The organization teams also can add customers as per required needs via spreadsheet to database and can also select that to which customer to send campaign and to which to not. So mainly campaign management system is a bridge between organization and their customers

**COMPANY PROFILE**

The Gateway Corp, a privately held global enterprise, is headquartered in India and The Netherlands. The Gateway Corp is the principal holding company and promoter of all Gateway Group Companies, comprising of 20 independent operating companies across 16 countries.

Over the last 22 years of Technology excellence, the Group has etched an enviable record of 80% client retention, in some of the most complex, non-English speaking markets, where we ensure that technology remains at the forefront of our clients’ business to achieve strategic differential advantage over their competition.

Our Vision 2023 is built on this foundation. The aggressive strategy for organic and inorganic growth entails – development of disruptive solutions, strategic initiatives & partnership and new markets penetration.

**COMPANY OVERVIEW**

Our group companies are seeded with a principle to achieve excellence through innovation. Spanning across a plethora of evolving technologies, our focus is to provide intelligent and insight driven information technology solutions, through a wide range of technology enabled services and developing product based platforms for enterprises in various industries. We are powered with an experience of delivering industry based technology solutions and IT enabled services for over two decades. We are specialized in our skills, working in 8 major disruptive industries, understanding their business functional lifecycles. Our comprehensive in-house capabilities and strategic international presence, enables us to make information technology more accessible, affordable and relevant, even to the most demanding customers in the farthest regions of the world including 30 of the European, American and Asian markets.

Our globally integrated infrastructure provides our group companies with a fully operational platform, localized businesses touch points and global development centers. This helps us to achieve a unique convergence of cost advantage, faster go to market, engagement flexibility and technology excellence. This institutes each of the group companies with a perfect blend of agility with quality, allowing us to map the regional technology trends with global industry expertise.

**WHAT COMPANY DO**

The inherent DNA of each Gateway Group Company is to deliver tangible and measurable business value to our global clients. We have ingrained it across all our strategic initiatives and brands. The guiding values and principles are sustained across all our enterprises by building an environment that integrates cutting edge technologies, scalable infrastructure and people culture across our global offices.

Our conviction to succeed, drives us to explore and operate in the most diverse markets, understanding their culture, languages and lifestyle, localizing teams in various nationalities. Our convergence in such challenging diversities is what constitutes us as one global family called

‘Gateway Group’.

**OUR MOTIVATION**

Gateway Group’s entrepreneurial genome is its key differentiation. Our customers globally have experienced the founding principles of Integrity, Ethics, and Commitment that has inspired the group to outperform challenges over the past two decades. On an equitable landscape of skills, expertise and track record and a strong belief in our value system, empowers the group, in all its ventures and brands to succeed.

**OUR VISION**

Consistently deliver ingenious, comprehensive, efficient, cost-effective Business solutions through IT for global market by creating a professionally stimulating and happy environment for our team where professionals thrive alongside the growth of our company.

**OUR MISSION**

Skilfully applying the most sought-after, fast-growing pertinent technology to yield total solutions in the most critical business areas.

**TABLES**

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure Number** | **Title** | **Page Number** |
| Fig 1.1 | Architectural Design | 8 |
| Figure 2.2.3 | TimeLine Chart | 9 |
| Figure 5.1.1 | Context Level (Level 0) | 26 |
| Figure 5.1.2 | Context Level (Level 1) | 27 |
| Figure 5.1.3 | Context Level (Level 2) | 28 |
| Figure 5.2 | Use Case Diagram | 29 |
| Figure 5.3 | Activity Diagram | 30 |
| Figure 5.4 | Sequence Diagram | 31 |
| Figure 5.6.1 | Navigation for Admin | 34 |
| Figure 5.6.2 | Navigation for User | 35 |
| Figure 6.1 | Black-Box Testing | 37 |
| Figure 6.2 | White-Box Testing | 38 |
| Figure 7.1.1 | Dash Board | 42 |
| Figure 7.1.2 | Dash Board | 42 |
| Figure 7.1.3 | Dash Board | 43 |
| Figure 7.1.4 | Dash Board | 43 |
| Figure 7.1.5 | Dash Board | 44 |
| Figure 7.1.6 | Dash Board | 44 |
| Figure 7.2 | Login Page | 45 |
| Figure 7.3 | Registration Page | 45 |
| Figure 7.4 | Forgot Password Page | 46 |
| Figure 7.5 | Add Campaign | 46 |
| Figure 7.6 | Add Campaign -> Email Page | 47 |
| Figure 7.7 | Email Template Master | 47 |
| Figure 7.8 | Email Template Editor | 48 |
| Figure 7.9 | Upload Customer Data | 48 |
| Figure 7.10 | Quick Campaign | 49 |
| Figure 7.11 | Campaign List | 49 |
| Figure 7.12 | Campaign List (On Button Click) | 50 |
| Figure 7.13 | Campaign Dashboard | 50 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| **Table Number** | **Title** | **Page Number** |
| Table 5.5.1 | Campaign Table | 32 |
| Table 5.5.2 | Campaign Status Table | 32 |
| Table 5.5.3 | Customers Table | 32 |
| Table 5.5.4 | Marketing Strategy Table | 32 |
| Table 5.5.5 | Marketing Types Table | 32 |
| Table 5.5.6 | User Table | 33 |
| Table 5.5.7 | Organization Table | 33 |
| Table 5.5.8 | Customer Campaign Table | 33 |
| Table 5.5.9 | Responses Table | 34 |
| Table 6.3 | Test Cases | 39 |

**ABBREVIATIONS**

Abbreviations used throughout this whole document for Survey Application are:

|  |  |
| --- | --- |
| **ABBREVIATION** | **FULL FORM** |
| IDE | INTEGRATED DEVELOPMENT ENVIRONMENT |
| UI | USER INTERFACE |
| JSON | JAVA SCRIPT OBJECT NOTATION |
| HTML | HYPER TEXT MARK UP LANGUAGE |
| CSS | CASCADING STYLE SHEET |
| DOM | DOCUMENT OBJECT MODEL |
| URL | UNIFORM RESOURCE LOCATOR |
| API | APPLICATION PROGRAMMING INTERFACE |
| GUI | GRAPHICAL USER INTERFACE |
| OS | OPERATING SYSTEM |

# CHAPTER 1 INTRODUCTION

* **PROJECT SUMMARY**
* **PROJECT PURPOSE**
* **PROJECT SCOPE**
* **TECHNOLOGY AND**

**LITERATURE OVERVIEW**

## 1.1 PROJECT SUMMARY

Campaign Management System is web application that manages campaigns and its customers and provides reports and charts in user readable format.

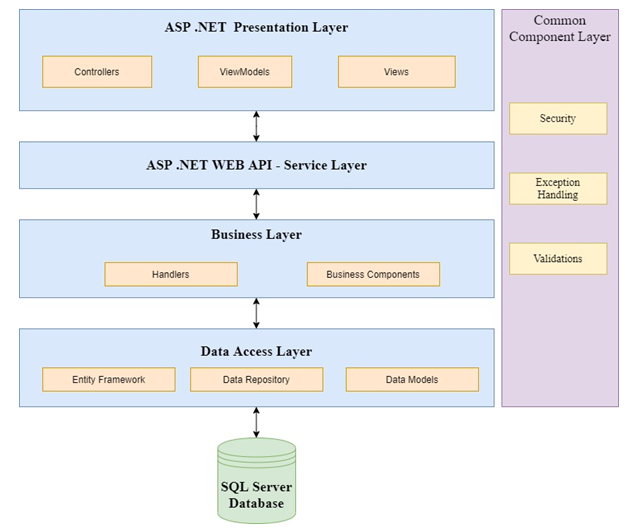


Fig 1.1 Architectural Design

## 1.2 PROJECT PURPOSE

This system is mainly used by organizations who want to launch any campaign. This system can help them managing the campaign and its respective customers in easy way.

Using this system organization can also check the profit and who is interested in their campaign just by few clicks.

## 1.3 PROJECT SCOPE

The scope of the CMS system is to manage huge campaign and its huge customers. It helps the organization to manage campaign by just adding it in system then they just have to observe where the campaign is going to loss or to profit. The organization can check all its previous campaign and check which strategy was used and how much profit was gained by that campaign

## 1.4TECHNOLOGY AND LITERATURE OVERVIEW

**1.4.1 Tools And Libraries**

* **.NET**

.NET is a free, cross-platform, open source developer platform for building many different types of applications. with .Net, you can use multiple languages, editors, and libraries to build for web, mobile, desktop, gaming, and IoT.

* **MVC**

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.

* **SQL Server**

The sql server is relational database management system, or RDBMS, developed and marketed by Microsoft.

Similar to other RDBMS software, SQL Server is built on top of SQL, a standard programming language for interacting with the relational databases. SQL server is tied to Transact-SQL, or T-SQL, the Microsoft’s implementation of SQL that adds a set of proprietary programming constructs.

SQL Server works exclusively on Windows environment for more than 20 years. In 2016, Microsoft made it available on Linux. SQL Server 2017 became generally available in October 2016 that ran on both Windows and Linux.

* **C#**

 C# is a general-purpose, modern and object-oriented programming language pronounced as **“C sharp”**. It was developed by Microsoft led by Anders Hejlsberg and his team within the .Net initiative and was approved by the European Computer Manufacturers Association (ECMA) and International Standards Organization (ISO). C# is among the languages for Common Language Infrastructure and the current version of C# is version 7.2. C# is a lot similar to Java syntactically and is easy for the users who have knowledge of C, C++ or Java.

* **HTML And CSS**

HTML (the Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML provides the structure of the page, CSS the (visual and aural) layout, for a variety of devices. Along with graphics and scripting, HTML and CSS are the basis of building Web pages and Web Applications.

* **LINQ**

LINQ (Language Integrated Query) is uniform query syntax in C# and VB.NET to retrieve data from different sources and formats. It is integrated in C# or VB, thereby eliminating the mismatch between programming languages and databases, as well as providing a single querying interface for different types of data sources.

* **BOOTSTRAP**

Bootstrap is the most popular CSS Framework for developing responsive and mobile first websites.

**1.4.2 Testing**

**1.4.2.1 Testing Plan**

The testing is a technique that is going to be used in the project is unit testing and system testing.

**1.4.2.2 Testing Strategy**

Unit Testing tests a unit of code after coding of that unit is completed. System testing ensures that the system meets its stated design specifications. Acceptance testing is testing by users to ascertain whether the system developed is a correct implementation of the software requirements specification. Testing is carried out in such a hierarchical manner that each component is correct and the assembly/combination of components is correct. Merely testing a whole system at the end would most likely throw up errors in components that would be very costly to trace and fix.

So we will be performing both unit testing to test every component is working as desired or not and then we will perform system testing where we will check that system meets desired requirements or not.

# 

# CHAPTER 2 PROJECT MANAGEMENT

* **PROJECT PLANNING**

**OBJECTIVES**

* **PROJECT SCHEDULING**

## 2.1 PROJECT PLANNING OBJECTIVES

Project planning is an organized and integrated management process, which focuses on activities required for successful completion of the project.

Objectives of Project Planning:

* It defines the roles and responsibilities of the project management team members.
* It ensures that the project management team works according to the business objectives.
* It checks the feasibility of the schedule and user requirements.
* It determines project constraints.

The Project development was planned and divided into various activities like:

* Initially, requirement gathering and research work.
* Based upon the research, learning and understanding of tools and libraries required for development.
* After understanding, start of actual implementation of the project.

### 2.1.1 Project Development Approach

The activities followed for this project is listed below:

* Requirement Gathering
* Analysis and Design
* Organizing and distributing work
* Back end code development
* UI development
* Integration of UI and backend
* Process model used: Incremental Process model

**2.1.2 Module Specification**

Main modules of system are:

* Login system for admin as well organization users
* Campaign with basic campaign details, schedule, financial and activity details.
* Data Import screen that will allow to import data from spreadsheet.
* Quick Campaign to send emails to bulk customers.
* Campaign Response Screen to track campaign response.
* Generate User-friendly Reports

**2.1.3 Coding Standards**

To develop reliable and maintainable applications, you must follow coding standards and best practices. We’ve followed C# coding standards from Microsoft for campaign management systems development.

## 2.2 PROJECT SCHEDULING

### 2.2.1 Basic Principle

Software project scheduling can be defined as an activity that distributes the estimated effort across the planned project duration by allocating the effort to specific software engineering tasks. Simply one can say that project schedule is a tool which communicates:

1. What works has to be performed.
2. Who will perform the work.
3. Time duration during which the works need to be completed.

There are seven principles of software project scheduling:

**Compartmentalization:**

A given software project is compartmentalized into a number of manageable activities.

**Interdependency:**

Certain tasks occur in sequence whereas other tasks occur in parallel. Therefore tasks which occur in sequence has to be performed in a sequential order since the output of one task will be the input of next task. Other tasks can occur independently.

**Time Allocation:**

Each and every task has to be assigned a specific time period.

**Effort Validation:**

Every project is assigned a software team.The project manager has to make sure that the effort allocated should not be more than the number of people available to do the work.

**Defined Responsibilities:**

Each of the scheduled task is assigned to a specific member of the software team.

**Defined Outcomes:**

Work product is the outcome of the software project.

**Defined milestones:**

Every task is associated with a milestone. A milestone is an action or an event marking a significant change in development process.

### 2.2.2 Compartmentalization

The development of the project is divided into the following set of activities/tasks:

1. Implement code for login system with encryption and authorization.
2. Implement code for crud operations for campaign.
3. Implement code for uploading spreadsheet and storing it in database
4. Implement code for adding quick campaign and sending mail to all its customers automatically
5. Implement forgot password functionality with send mail.
6. Implement code for sending welcome mail at start of campaign and response mail at end of campaign automatically
7. Implement response tracking code using guide and storing it in database
8. Implemented code for checking response in graph format and view its report
9. Implement code for checking customers is interested in which type of campaigns

### 2.2.3 TimeLine Chart

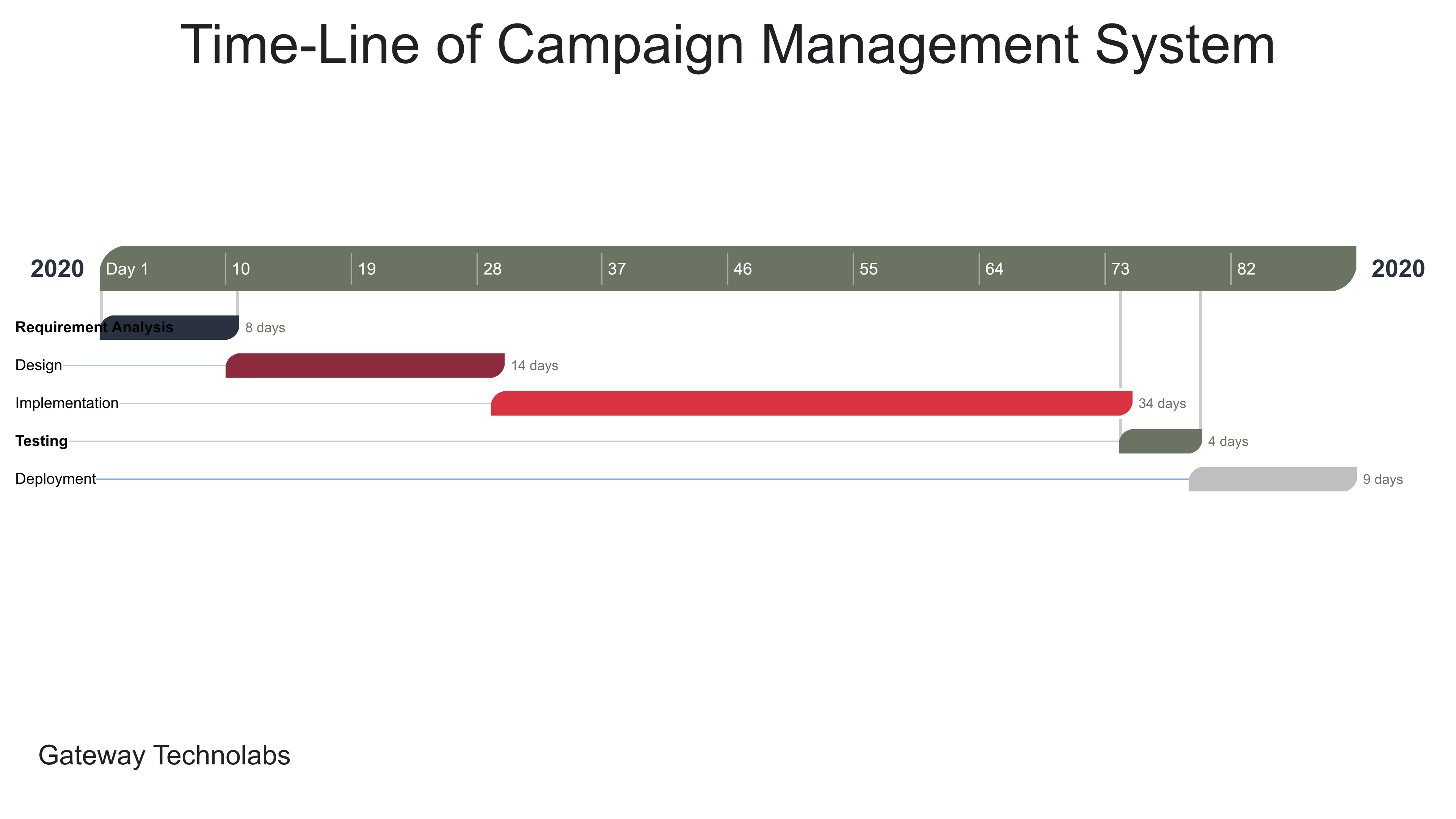


Fig 2.2.3 TimeLine Chart

# CHAPTER 3

**SYSTEM REQUIREMENTS**

* **USER CHARACTERISTICS**
* **FUNCTIONAL REQUIREMENT**
* **NON FUNCTIONAL**

**REQUIREMENT**

* **HARDWARE AND SOFTWARE**

**REQUIREMENT**

## 3.1 USER CHARACHTERISTICS

**User:**

User can use CMS for management and prediction of the growth or downfall of their campaigns.

## 3.2 FUNCTIONAL REQUIREMENT

Functional requirements define the internal working of the software: that is, the calculations, technical details, data manipulation and processing and other specific functionality that show how the use cases are to be satisfied.

The functional requirements of the beat are mentioned as follows:

**R1: Login As Admin/Organization User**

**Input:** Email Id And Password.

**Output:** Dashboard for Admin/Organization User.

**Description:** check for email id and password in database and role for found entry and providing dashboard as role

.

**R2: Add New User**

**Input:** User Details.

**Output:** User Add Success or failed notification.

**Description:** User details will be added in database.

**R3: Retrieve Password**

**Input:** Email ID.

**Output:** random generated password.

**Description:** check in database for email id and send random generated password to given email id.

**R4: Add New Campaign**

**Input:** Campaign Details.

**Output:** Campaign List with added campaign.

**Description:** Add campaign detail in database.

**R5: Edit Campaign**

**Input:** Edited Campaign Details.

**Output:**  Campaign List With Edited Campaign.

**Description:** Edit campaign detail in campaign database

**R6: Delete Campaign.**

**Input:** Campaign ID.

**Output:** Campaign list without deleted campaign.

**Description:** delete campaign with given campaign id from database.

**R7: Check Response.**

**Input:** Campaign Id.

**Output:** Response Got from User for given campaign.

**Description:** systemwill .check for response for campaign id from database

**R8: Generate Report.**

**Input:** Campaign Id.

**Output:** Generated Report.

**Description:** system will generate report for given campaign id.

**R9: Add Quick Campaign.**

**Input:** Quick Campaign Detail.

**Output:** Send Mail to all customers.

**Description:** quick campaign will be added in database and mail will be sent to all its customers.

**R10: Upload Spreadsheet.**

**Input:** spreadsheet with customer email list.

**Output:** customers added to database.

**Description:** all the customers from spreadsheet will be added to database

**R11: Send Automatic Mail.**

**Input:** Email List.

**Output:** Welcome/Response Mail Will be sent.

**Description:** mail will be sent to all customers automatically.

## 3.3 NON FUNCTIONAL REQUIREMENT

**Accuracy**

As we were developing the system, we must make the system that is very accurate in its functions. All the functions should keep working properly, keep getting perfect input, process accurately and produce the perfect output. Accuracy is the most important non-functional characteristic or requirement of the system.

**Reliability**

Error handling mechanism must be robust to avoid failure of operation and in case of failure the system reports it to the admin without any due harm.

**Performance and Automation**

Campaign management system is very fast as we are using dependency injection and we are automating response tracking and mail sending without user or admin interaction

**Consistency**

In system data must should be consistent

**Scalability**

System must should scalable and can be easily scaled horizontally or vertically as in our system we are using repository pattern and layered architecture scalability can be achieved

## 3.4 HARDWARE AND SOFTWARE REQUIREMENT

### 3.4.1 Hardware Requirement

* Processor(CPU) with 2GHz frequency or above
* A minimum of 4 GB RAM
* Internet connection with a speed of 4Mbps or higher

### 3.4.2 Software Requirement

* Latest version of Visual Studio
* SSMS (SQL Server Management Studio)
* Operating System : Windows 7/8/8.1/10

11

# CHAPTER 4 SYSTEM ANALYSIS

* **FEASIBILITY STUDY**
* **PROJECT PLANNING**

## 4.1 FEASIBILITY STUDY

Once scope has been identified, it is reasonable to ask whether we can build software that meets this scope. Is this project feasible?

The feasibility of software can be tested in four dimensions:

**4.1.1 Technical Feasibility**

What we have planned to implement is technically feasible. Do we have sufficient knowledge or technology to make it a reality? And the answer is yes we are having knowledge of technologies which are required for making this system working, we can easily design frontend using html, css and bootstrap and for backend we can use .net, mvc, c# and sql for database.

**4.1.2 Time Schedule Feasibility**

We planned keeping all the development phase in our mind so our system can be ready in time without any error, after keeping this in mind we planned our deadline one week before actual submission date so we can complete system without error and we found out that we will require minimum 2 months to implementation of the complete project with all the features implemented. This also includes the testing and debugging phase.

**4.1.3 Operational Feasibility**

How the project will work and who will use it, all such concerns arise in this phase. As by using this system, campaigns and customers should be managed efficiently without any data loss, cohesion and errors

**4.1.4 Implementation Feasibility**

As we are using mvc framework for implementing this system we can use its different features in implementing this project like model, view and controller as well as we can use sql server for database related needs

## 4.2 PROJECT PLANNING

In the development of this project, we will first check to see if our project is feasible functionally, technically and economically. Then we collect the requirements. Hence, we gather all the requirements which we need to develop our system. Then, after thoroughly understanding the requirements, we will start development.

**Project Development Approach**

The Agile Model is used for project development. We have selected Agile Model because of its beneficial speed without affecting the quality of product and agile makes the team so much more productive.

**Agile Model**

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing. At the end of the iteration a working product is displayed to the customer and important stakeholders.

# CHAPTER 5 SYSTEM DESIGN

* **DATA FLOW DIAGRAM**
* **USE CASE DIAGRAM**
* **ACTIVITY DIAGRAM**
* **SEQUENCE DIAGRAM**
* **DATA DICTIONARY**
* **APPLICATION NAVIGATION**

## 5.1 DATA FLOW DIAGRAM

**5.1.1 Context Level (Level 0)**

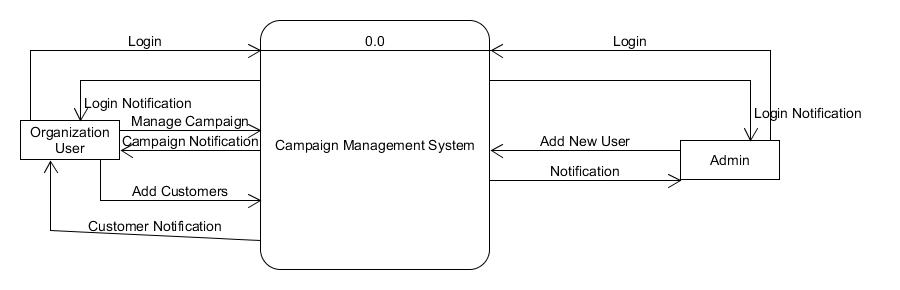


Fig 5.1.1 Context Level (Level 0)

**5.1.2 Context Level (Level 1)**

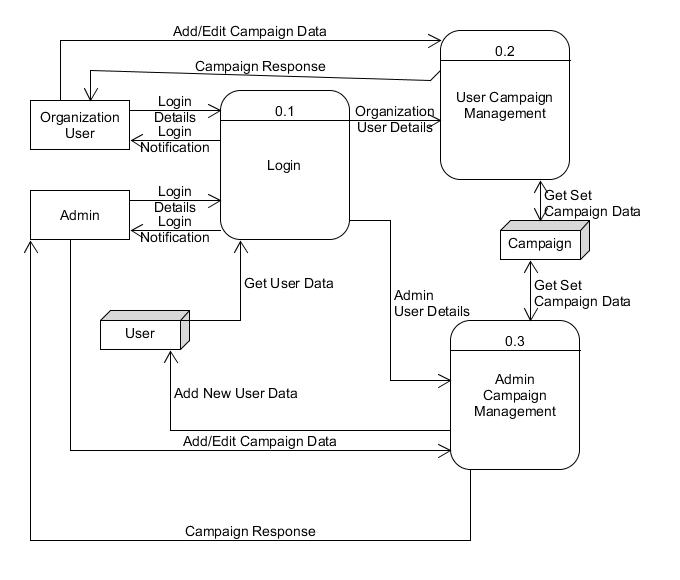


Fig 5.1.2 Context Level (Level 1)

**5.1.3 Context Level (Level 2)**

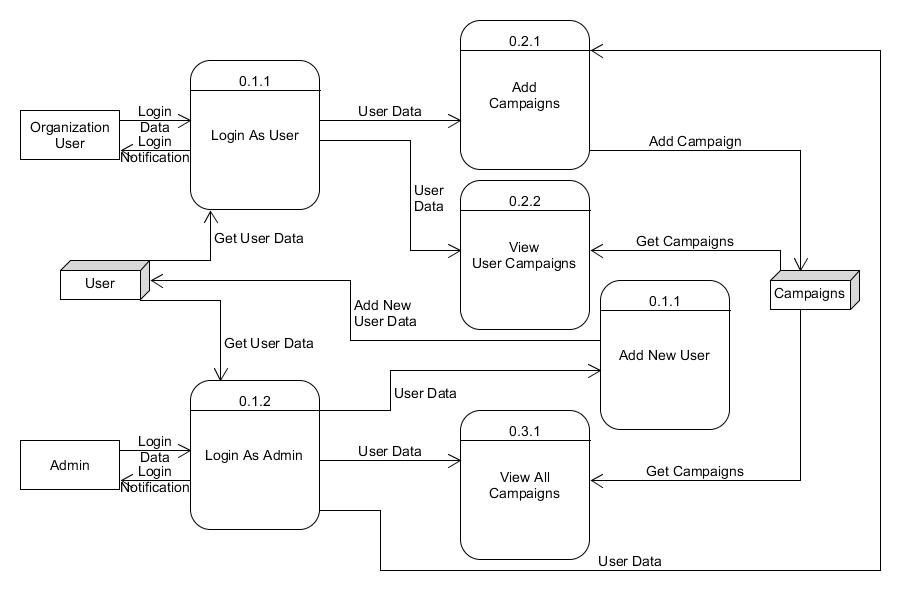


Fig 5.1.3 Context Level (Level 2)

## 5.2 USE CASE DIAGRAM

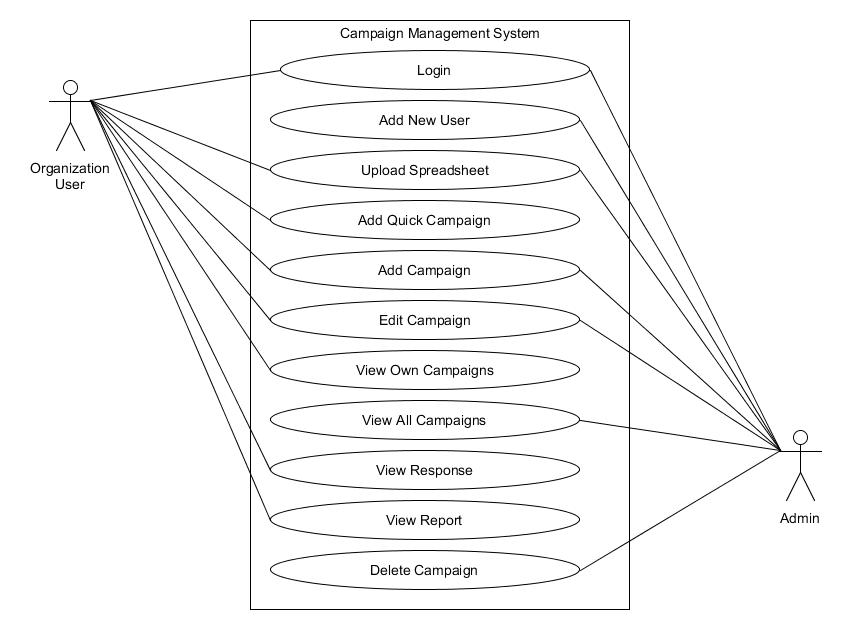
****

Fig 5.2 Use Case Diagram

## 5.3 ACTIVITY DIAGRAM

## Activity

Fig 5.3 Activity Diagram

## 5.4 SEQUENCE DIAGRAM

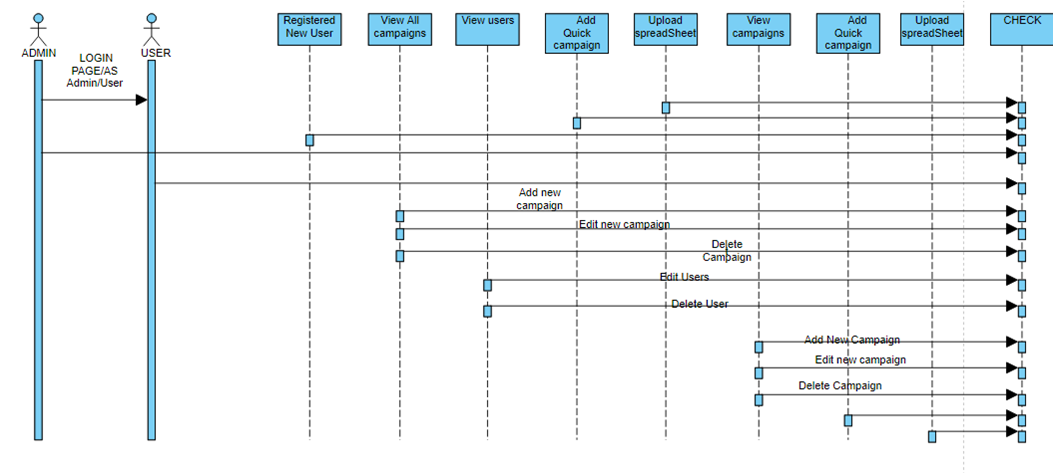
****

Fig 5.4 Sequence Diagram

## 5.5 DATA DICTIONARY

1. **Campaign**

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Column Name** | **Datatype** |
| 1 | CampaignId | Int |
| 2 | CampaignName | Varchar |
| 3 | CampaignOwner | Varchar |
| 4 | Start\_Date | Datetime |
| 5 | End\_Date | Datetime |
| 6 | CampaignBudget | Decimal |
| 7 | ExpectedRevenue | Decimal |
| 8 | CreatedOn | Datetime |
| 9 | ModifiedOn | Datetime |
| 10 | CreatedBy | Int |
| 11 | CampaignStatusId | Int |
| 12 | MarketingId | Int |
| 13 | MarketingStrategyId | Int |

Fig 5.5.1 Campaign Table

1. **CampaignStatus**

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Column Name** | **Datatype** |
| 1 | CampaignStatusId | Int |
| 2 | Status | Varchar |

Fig 5.5.2 Campaign Status Table

1. **Customers**.

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Column Name** | **Datatype** |
| 1 | CustomerId | Int |
| 2 | CustomerName | Varchar |
| 3 | Email | Varchar |

Fig 5.5.3 Customers Table

1. **MarketingStrategy**

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Column Name** | **Datatype** |
| 1 | MarketingStrategyId | Int |
| 2 | StrategyName | Varchar |

Fig 5.5.4 Marketing Strategy Table

1. **MarketingTypes**

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Column Name** | **Datatype** |
| 1 | MarketingTypeId | Int |
| 2 | MarketingTypeName | Varchar |

Fig 5.5.5 Marketing Types Table

1. **User**

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Column Name** | **Datatype** |
| 1 | UId | Int |
| 2 | Email | Varchar |
| 3 | Password | Varchar |
| 4 | Role | Varchar |
| 5 | FirstName | Varchar |
| 6 | LastName | Varchar |
| 7 | OrganizationId | Int |

Fig 5.5.6 User Table

1. **Organizations**

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Column Name** | **Datatype** |
| 1 | OrganizationId | Int |
| 2 | OrganizationName | Varchar |

Fig 5.5.7 Organization Table

1. **CustomerCampaign**

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Column Name** | **Datatype** |
| 1 | CustomerCampaignId | Int |
| 2 | CampaignId | Int |
| 3 | CustomerId | Int |

Fig 5.5.8 Customer Campaign Table

1. **Responses**

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Column Name** | **Datatype** |
| 1 | ResponseId | Int |
| 2 | CampaignId | Int |
| 3 | Positive | Int |
| 4 | Negative | Int |
| 5 | Neutral | Int |

Fig 5.5.9 Responses Table

**5.6 APPLICATION NAVIGATION**

**5.6.1 NAVIGATION FOR ADMIN**

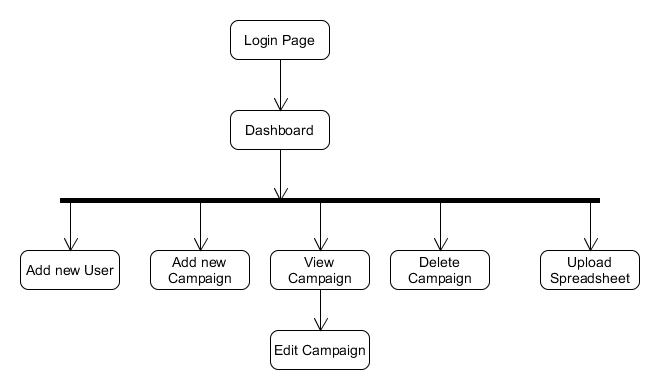
****

Fig 5.6.1 Navigation for Admin

**5.6.2 NAVIGATION FOR USER**

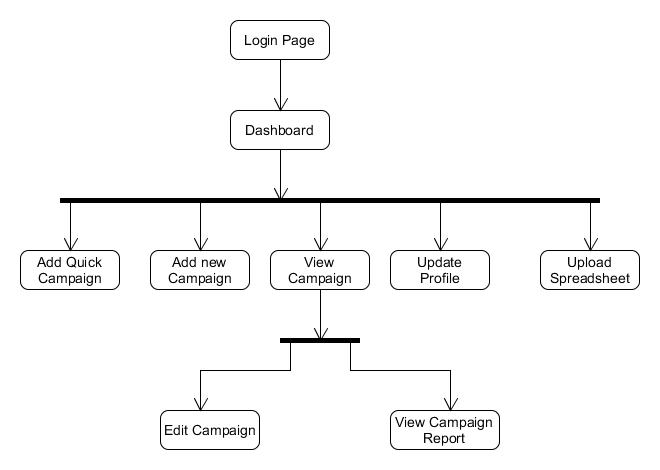
****

Fig 5.6.2 Navigation for User

# CHAPTER 6 TESTING

* **BLACK-BOX TESTING**
* **WHITE-BOX TESTING**
* **TESTCASES**

## 6.1 BLACK- BOX TESTING

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

This method of testing is named as black box because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

* Incorrect or missing functions
* Interface errors
* Errors in data structures or external database access
* Behaviour or performance errors
* Initialization and termination errors

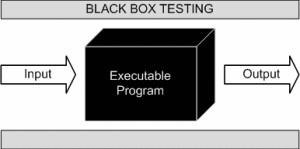
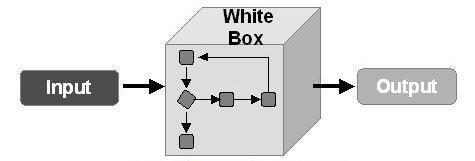


Fig 6.1 Black-Box Testing

## 6.2 WHITE- BOX TESTING

White-box testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called glass testing or open-box testing. In order to perform white-box testing on an application, a tester needs to know the internal workings of the code.

This method of testing is named as white box because the software program, in the eyes of the tester, is like a white/transparent box; inside which one clearly sees. The tester chooses inputs to exercise paths through the code and determines the appropriate output.



6.2 White-Box Testing

## 6.3 TEST CASES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **Test Case** | **Expected Output** | **Actual Output** | **Test Case Result** |
| 1 | For Login as admin provide correct admin credentials. | System should return admin dashboard | Admin dashboard | Pass |
| 2 | For Login as Organization User Provide correct credentials | System should return User dashboard | User Dashboard | Pass |
| 3 | For Adding New Campaign details are provided | Campaign list should be returned and stored in database | Campaign list with new added campaign and stored in database | Pass |
| 4 | For Deleting campaign  Delete campaign button is clicked | Campaign should be removed from database and list should be returned. | Campaign list is returned and campaign is removed from database | Pass |
| 5 | Add New User:  User Details are provided | User details should be stored in database. | User details are stored in database | Pass |
| 6 | Check customer Response for Campaign | System should return Customer response screen for campaign | Customer response screen for campaign | Pass |
| 7 | Upload spreadsheet | Customer in spreadsheet should be stored in database and list should be returned | Customers are stored in database and list is returned | Pass |
| 8 | Forgot password  Email address is provided | Random generated password should be mailed to given mail address | Random generated password is mailed | Pass |
| 9 | Send Welcome Mail | Welcome mail should sent to all customers on start date of campaign | Welcome mail is sent to all customers on start date | Pass |
| 10 | Send Feedback Mail | Feedback mail should be sent to all customers at end date of campaign | Feedback mail is sent to all customers on end date of campaign | Pass |

Fig 6.3 Test Cases

# CHAPTER 7 Screen Shots

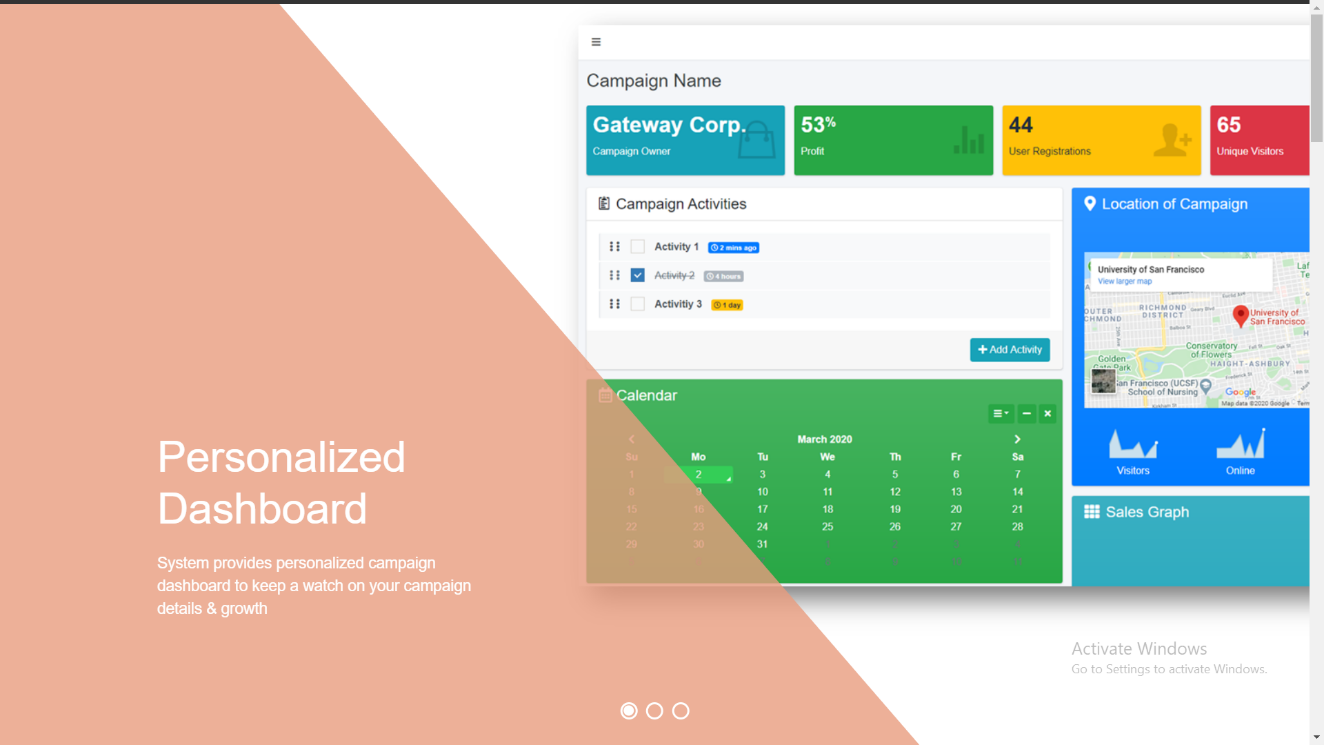


Fig 7.1.1 Dash Board

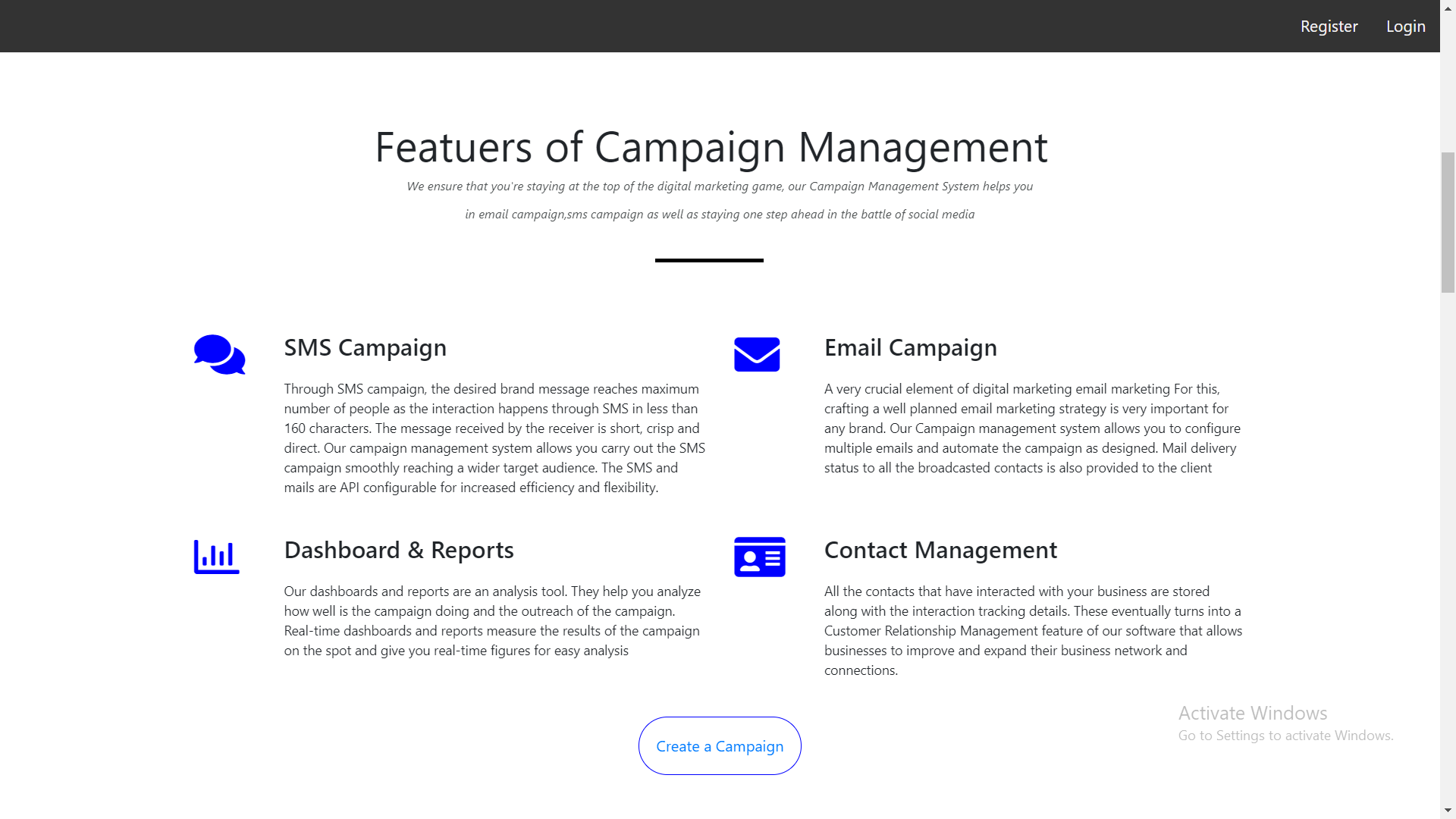


Fig 7.1.2 Dash Board



Fig 7.1.3 Dash Board

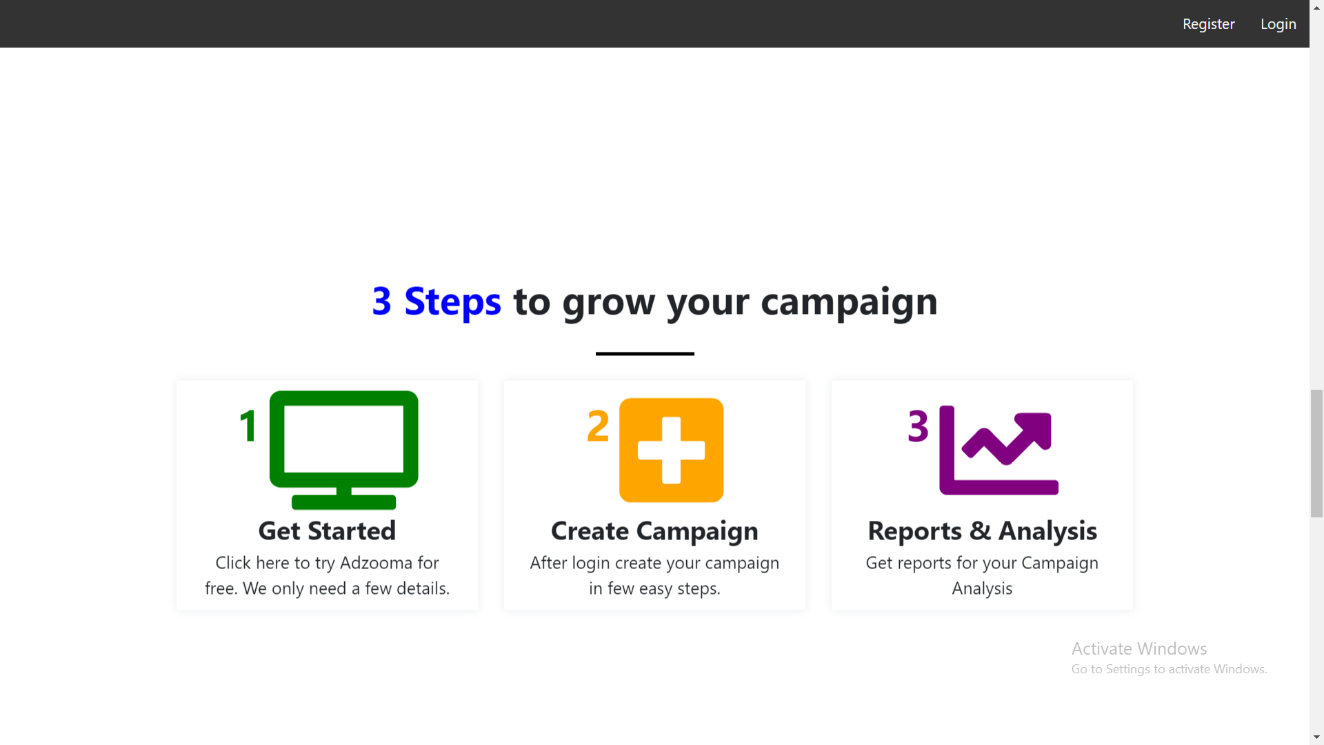


Fig 7.1.4 Dash Board

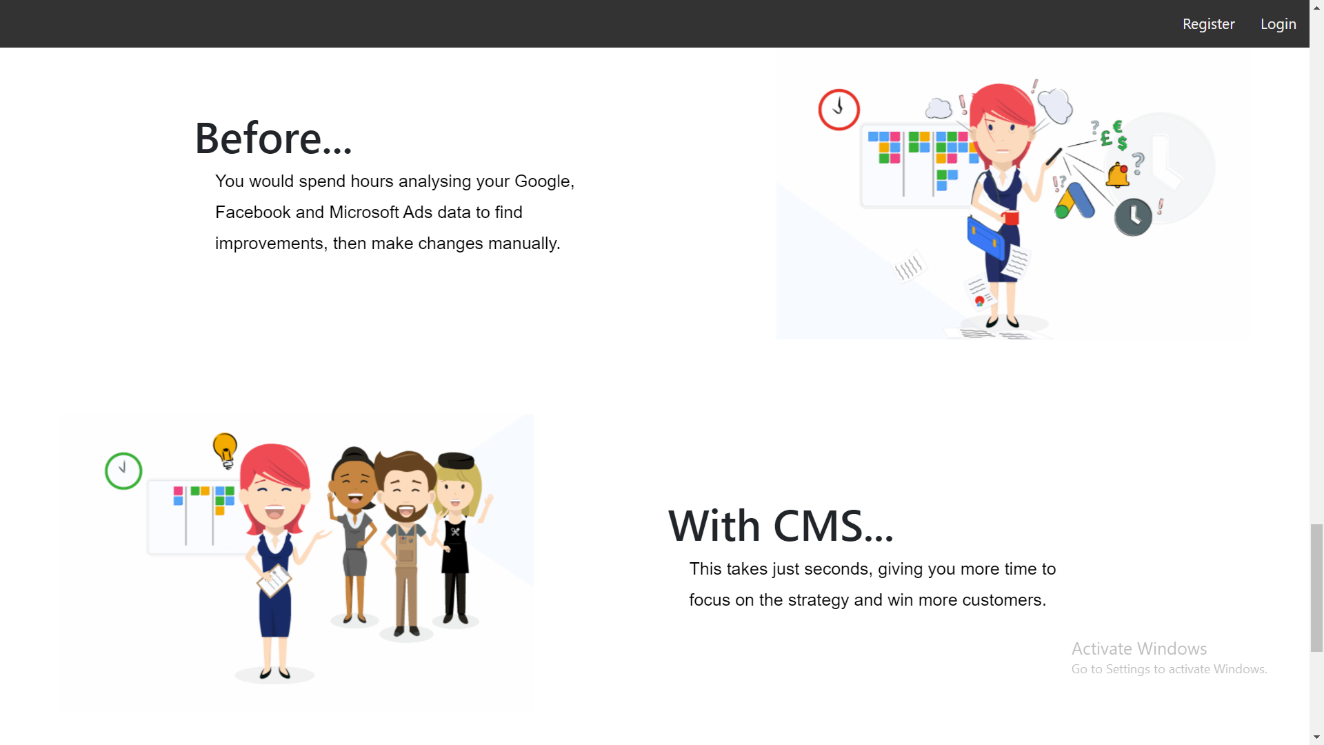


Fig 7.1.5 Dash Board



Fig 7.1.6 Dash Board

## loginpage

Fig 7.2 Login Page

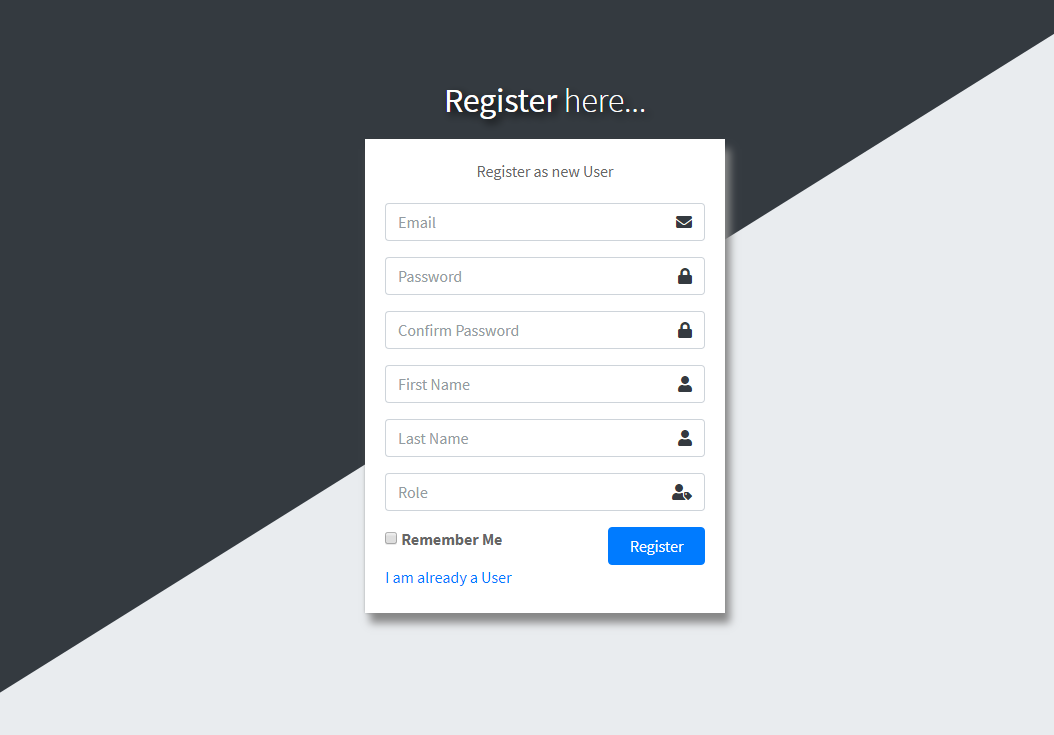
****

Fig 7.3 Registration Page

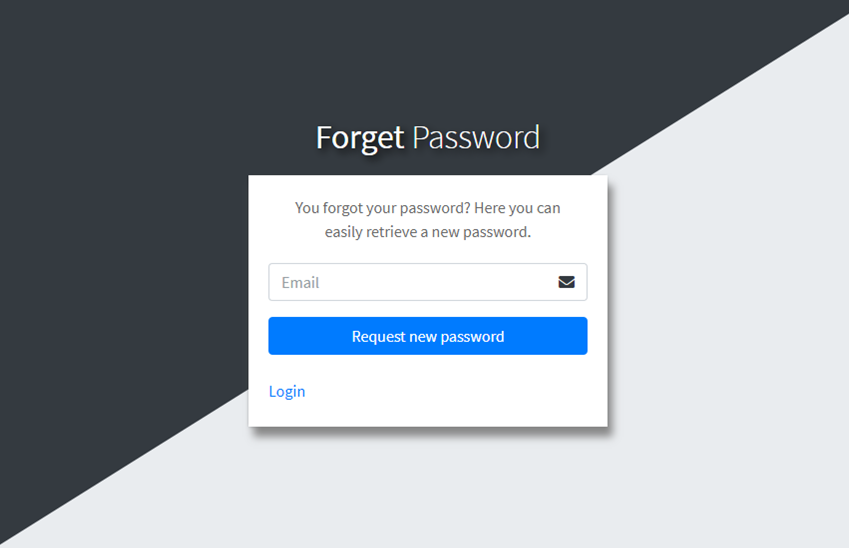
****

Fig 7.4 Forgot Password Page

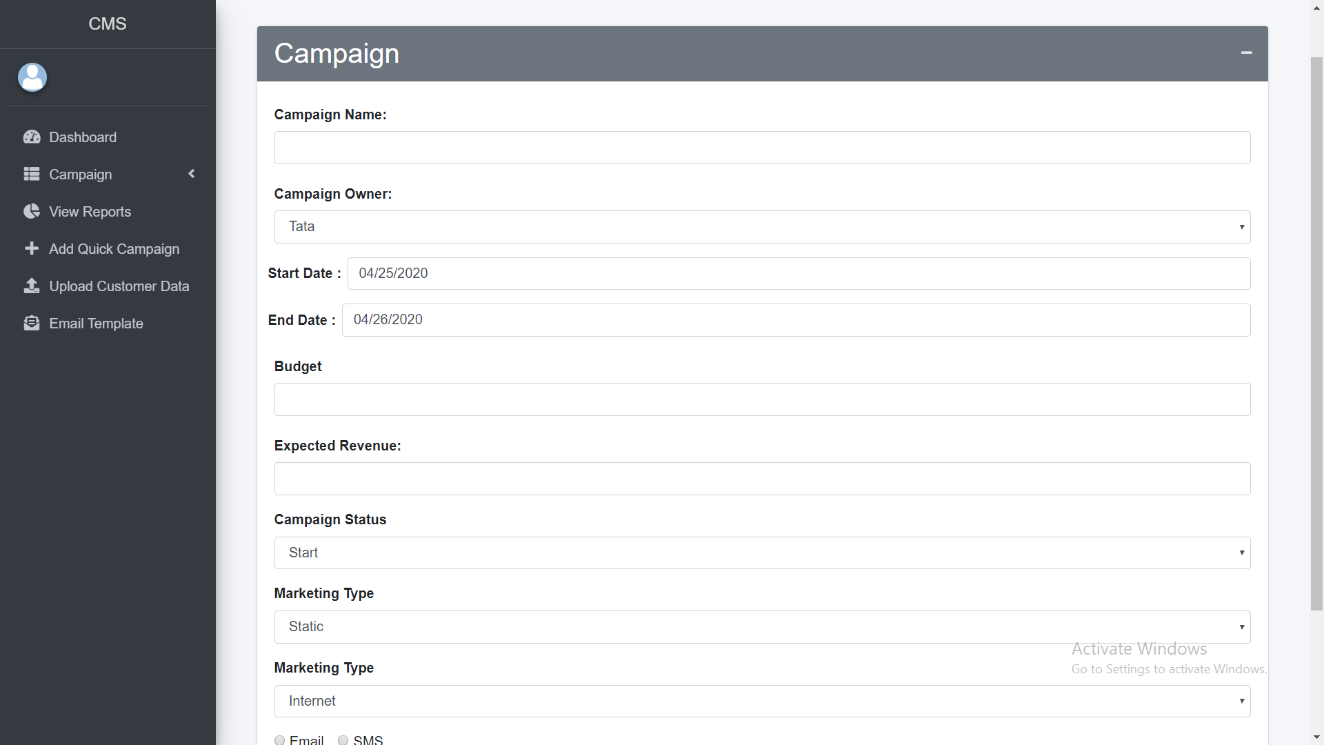


Fig 7.5 Add Campaign

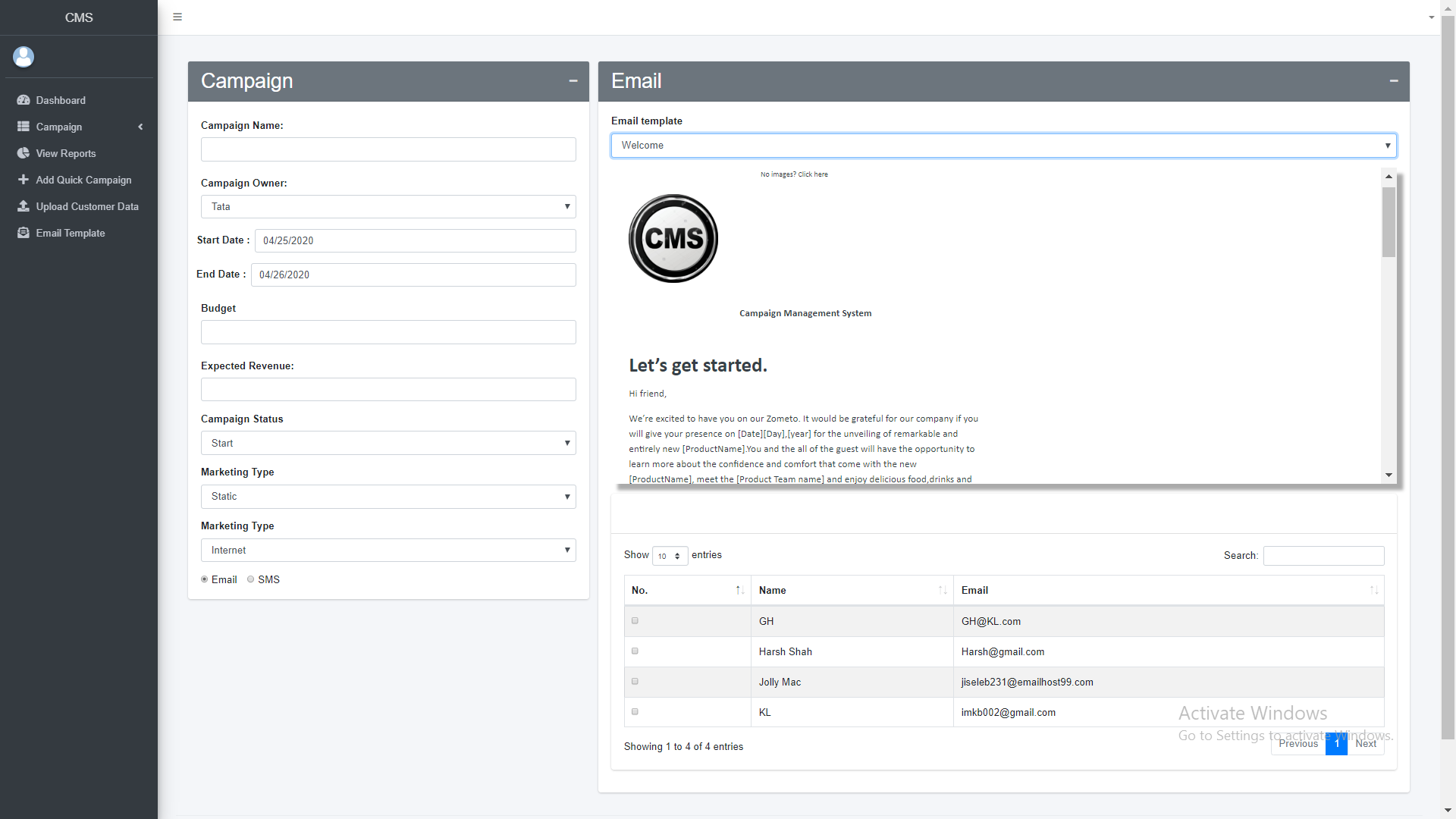


Fig 7.6 Add Campaign -> Email Page

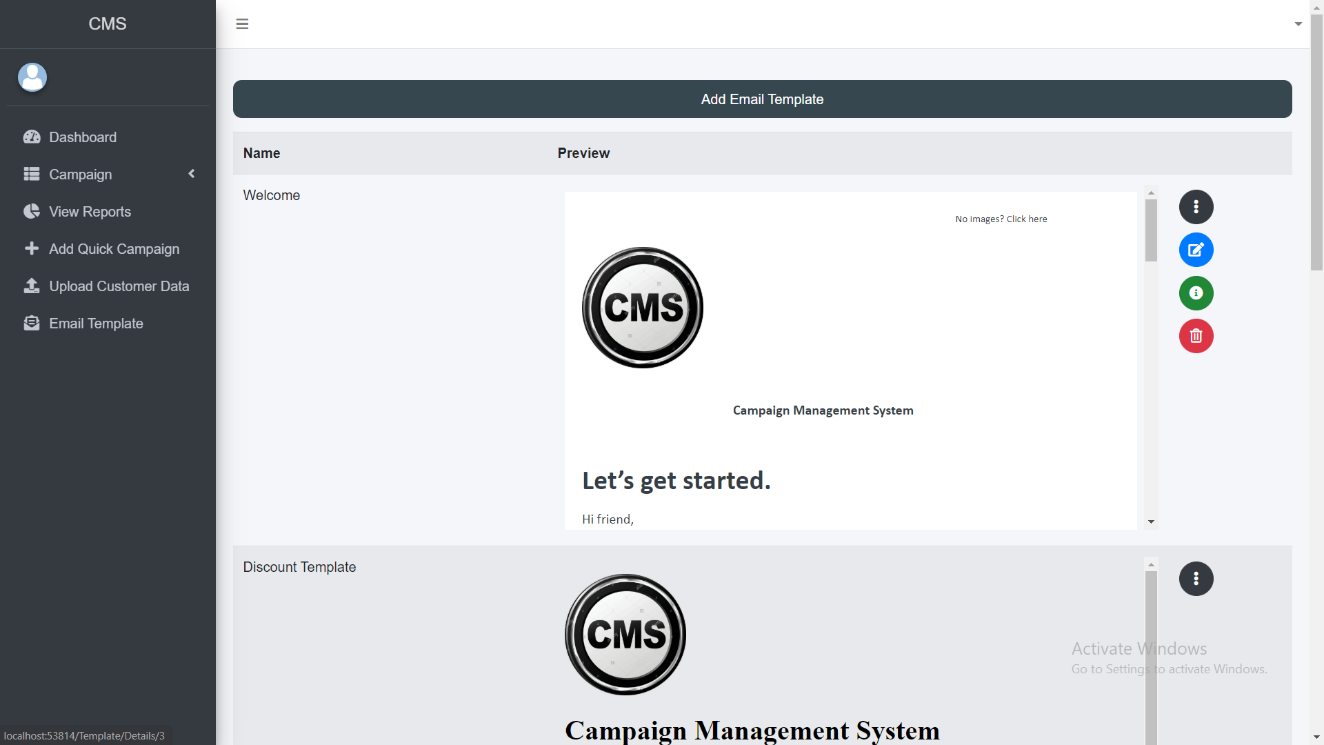


Fig 7.7 Email Template Master



Fig 7.8 Email Template Editor

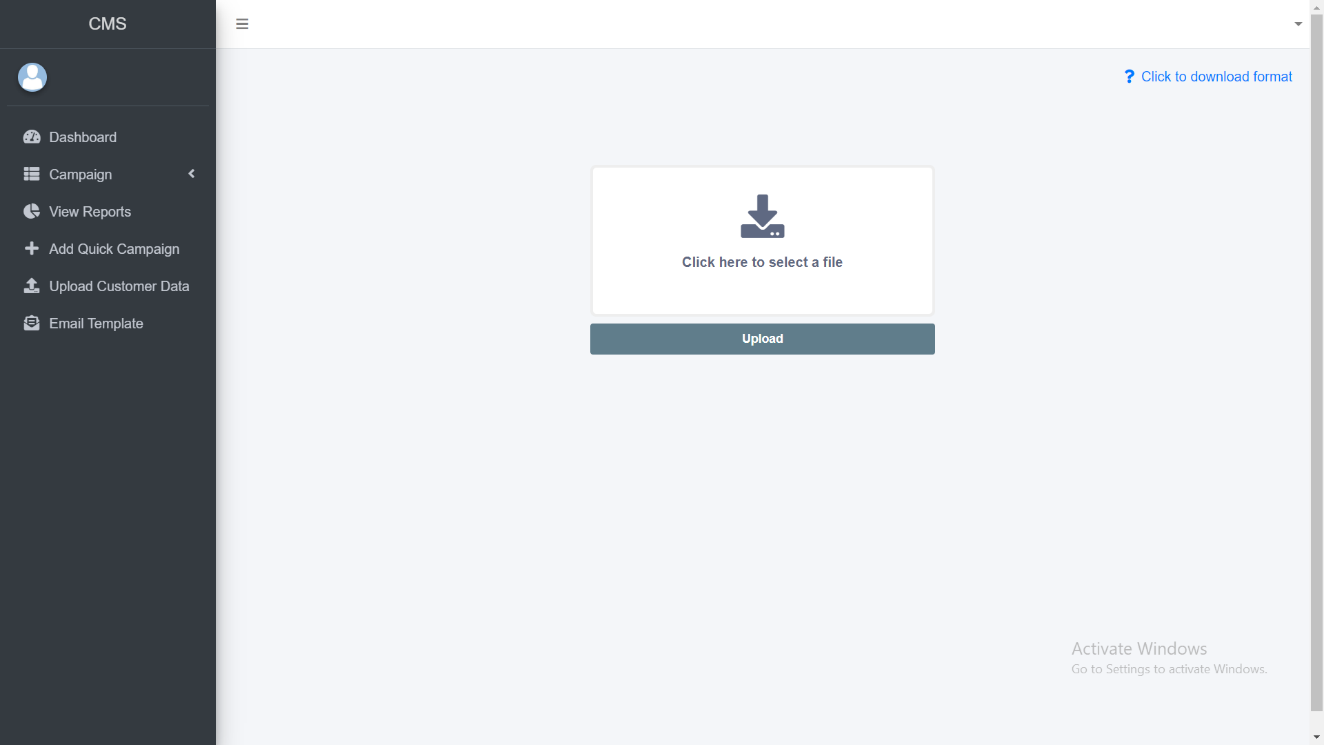


Fig 7.9 Upload Customer Data

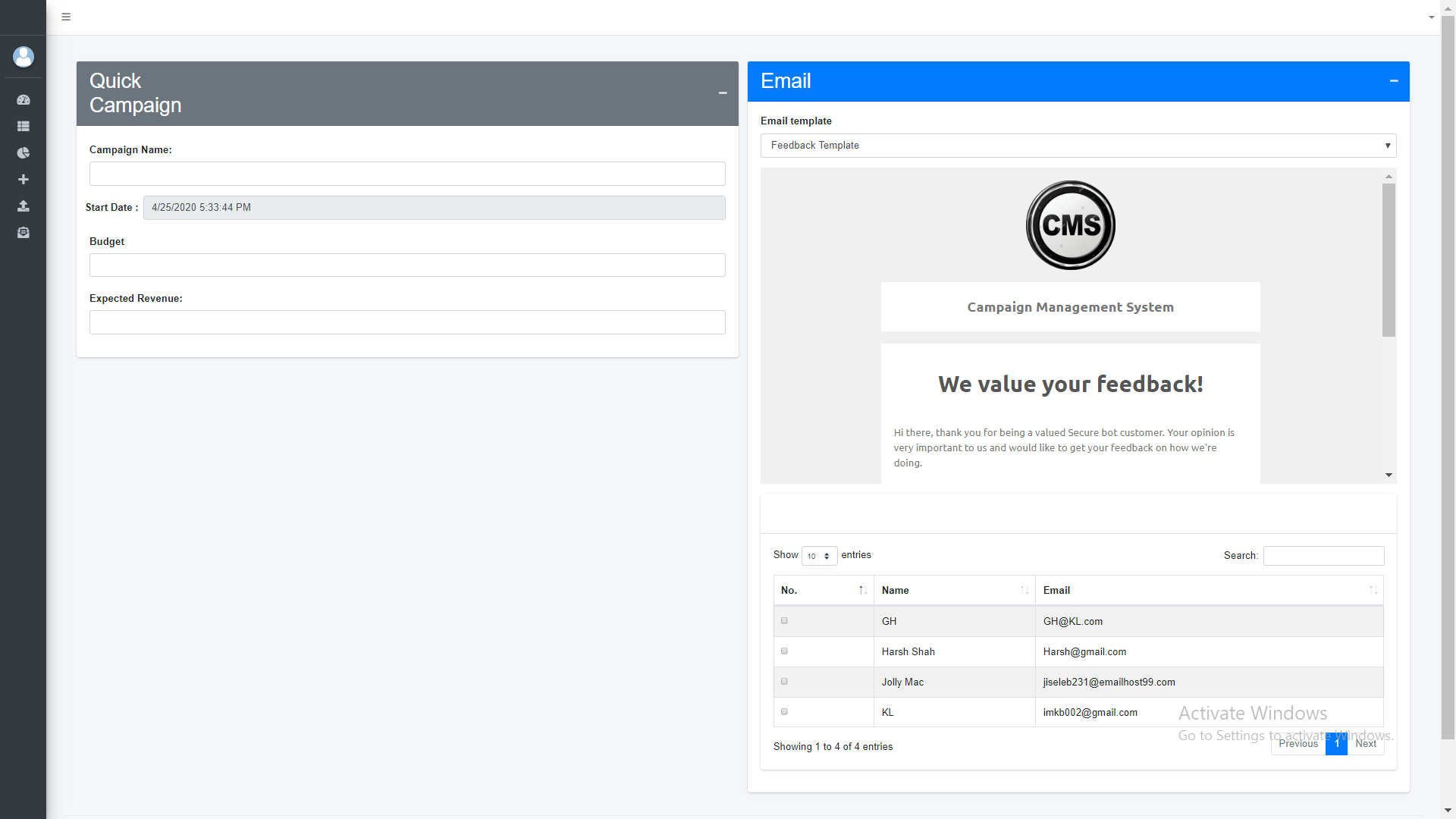


Fig 7.10 Quick Campaign

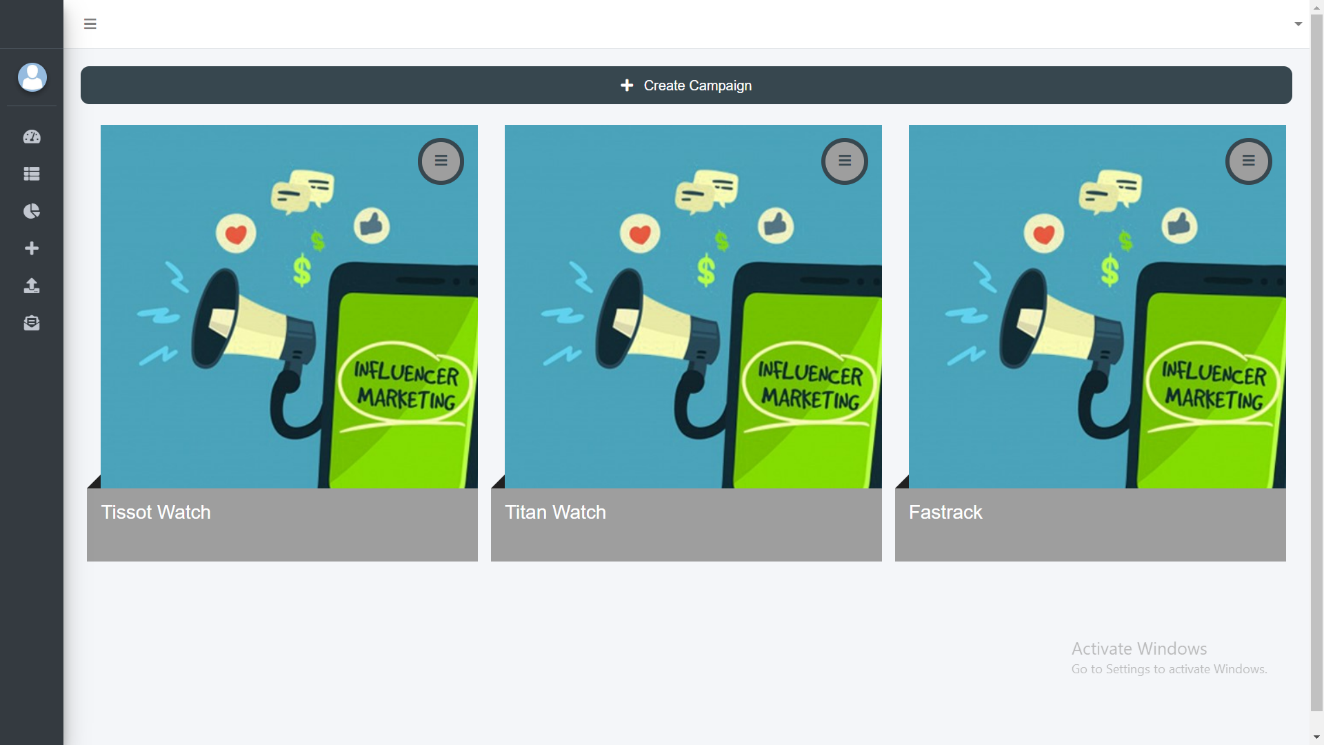


Fig 7.11 Campaign List

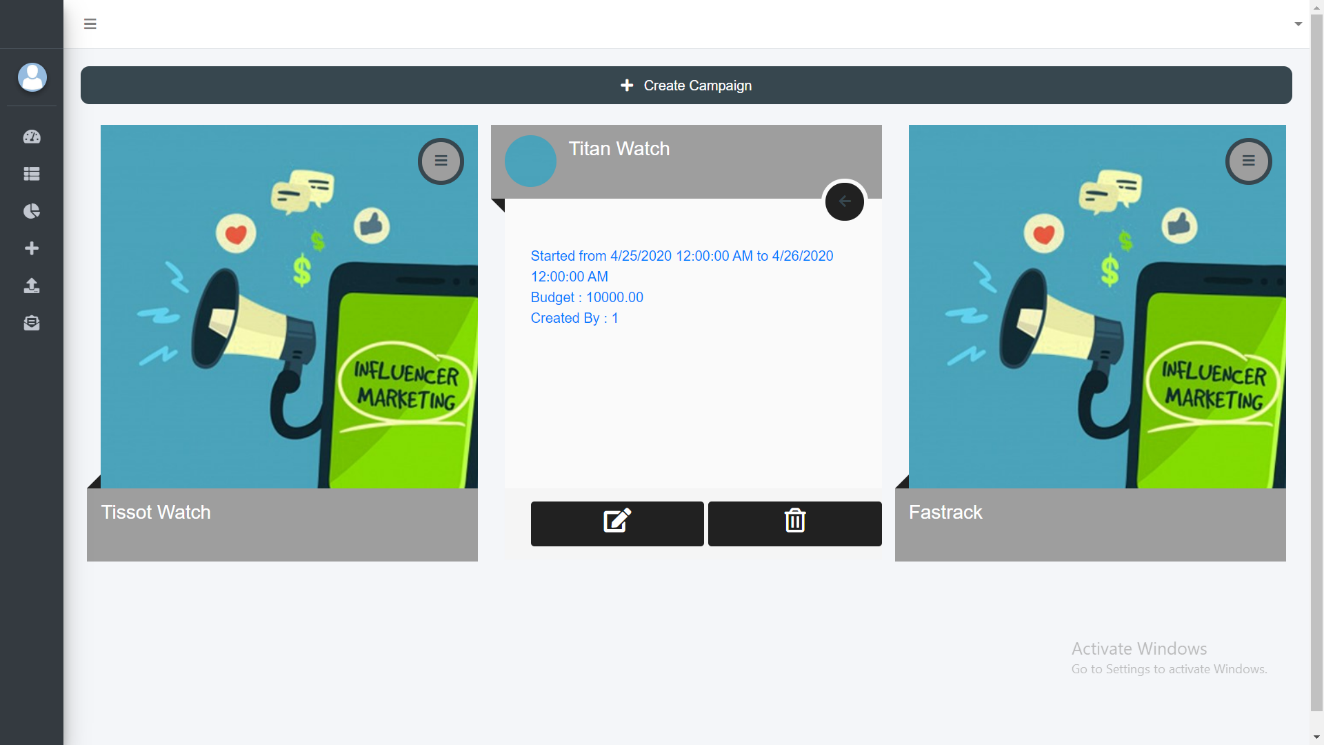


Fig 7.12 Campaign List (On Button Click)

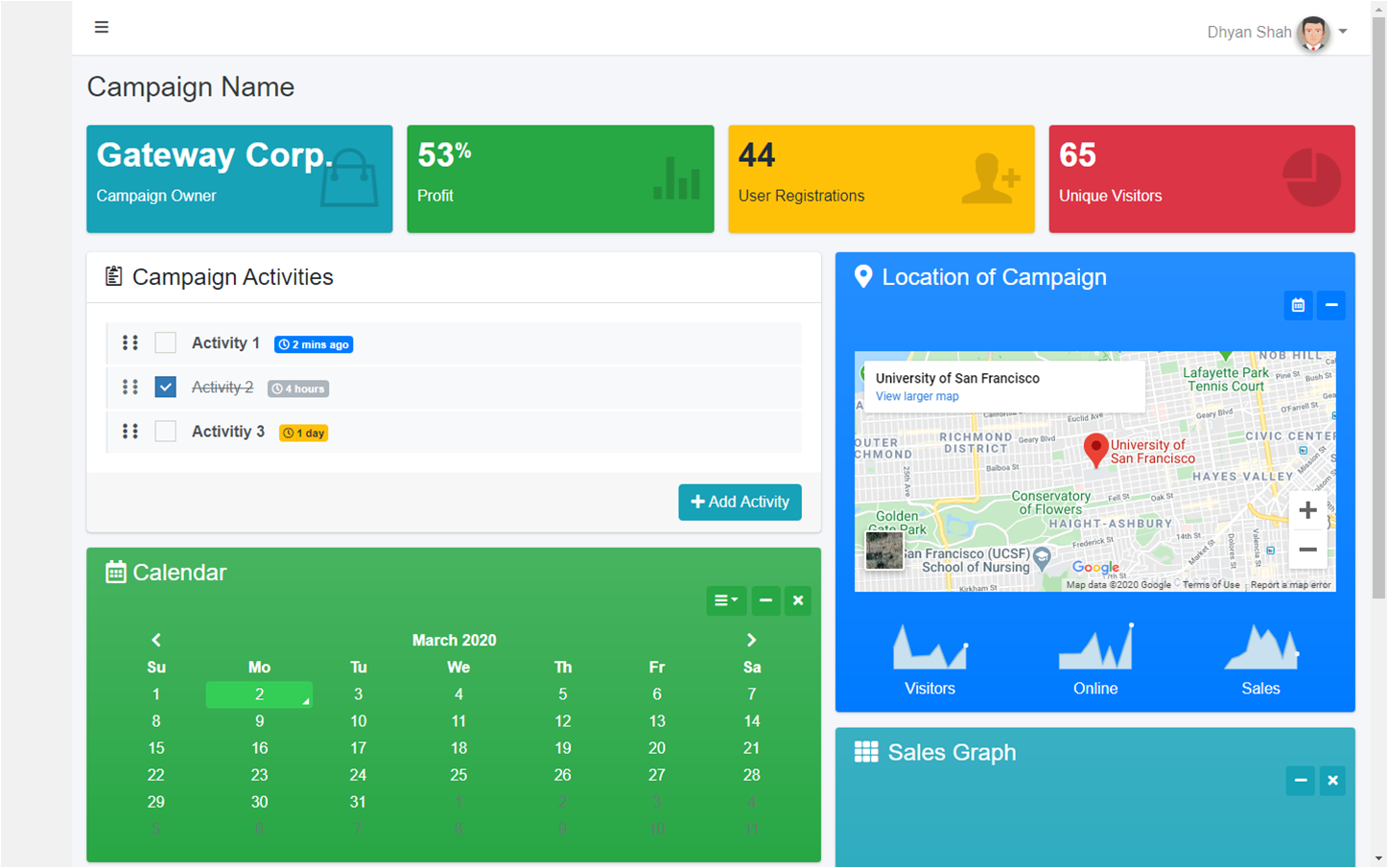
****

Fig 7.13 Campaign Dashboard

# CHAPTER 8 CONCLUSION & FUTURE EXTENSION

# ­

# ­

## 8.1 CONCLUSION

* Campaign management system eases the work of organizations or individuals who want to launch campaign and want to manage that campaign easily.
* Campaign management is main feature of this system.
* We are using sql server so it is easy to manage database for this system.
* Mvc is most used web development technology so it is providing much functionality which are implemented in this system.
* The response tracking in this system is very efficient as we are using guide for tracking which customer is giving what response for which campaign so it is easy to track response from user.
* We are using smtp for sending mails and it is very fast.
* We can easily extend or add new functionality in our system as we are using layered architecture in our system.
* Our system works very fast as we are using repository patterns into it.

## 8.2 FUTURE EXTENSIONS

* Currently we are using only email to send campaigns but in future we will try to use sms and social media for sending campaigns.
* Currently we are using only buttons and radio buttons for response in future we will use textbox for getting feedback.
* Currently we are not able to predict campaign response in future we will predict what response will campaign get using previous campaigns responses
* In future we are planning to extend our system to be able to automate responses and send it to organization via email

within a test scenario and creating multiple such senarios, creating more generalized report of test result (pass or fail) using charts.

36

# APPENDICES

**A**

Abbreviations……………………………………………………………………….. viii

Abstract……………………………………………………………………………... iv

Activity Diagram…………………………………………………………………… 33

API………………………………………………………………………………… 5

**B**

Basic Principle……………………………………………………………………… 7

Black Box Testing………………………………………………………………….. 24

**C**

Class Diagram……………………………………………………………………… 31

Company Profile………………………………………………………….………… V

Compartmentalization………………………………………………………………. 8

Conclusion………………………………………………………………………….. 37

**D**

Data Flow Diagram…………………………………………………………………. 34

**E**

Economical Feasibility…………………………………………………………….. 15

**F**

Feasibility Study…………………………………………………………………… 14

Features of New System…………………………………………………………… 15

Framework………………………………………………………………………… 3

Functional Requirement…………………………………………………………… 10

Future Enhancement………………………………………………………………. 36

**H**

Hardware Requirement……………………………………………………………. 11

**I**

IDE………………………………………………………………………………… 4

**L**

Language…………………………………………………………………………… 5

Limitation…………………………………………………………………………... 36

List of Figures………………………………………………………………………. Vi

List of Tables……………………………………………………………………….. Vii

**N**

Non Functional Requirement……………………………………………………….. 10

**O**

Operational Feasibility……………………………………………………………… 15

**P**

Problems in Current System………………………………………………………... 12

Process Model………………………………………………………………………. 13

Project Development Approach ……………………………………………………. 7

Project Planning Objectives………………………………………………………… 6

Project Purpose……………………………………………………………………... 1

APPENDICES

Project Scheduling………………………………………………………………….. 7

Project Scope……………………………………………………………………….. 1

Project Summary…………………………………………………………………… 1

**R**

Requirement of New System……………………………………………………….. 12

**S**

Schedule Feasibility………………………………………………………………… 15

Software Requirement……………………………………………………………… 11

Software Scope……………………………………………………………………... 6

Study of Current System…………………………………………………………… 12

**T**

Table of Content…………………………………………………………………… Ii

Technical Feasibility………………………………………………………………... 14

Technology and Literature Overview………………………………………………. 2

Test Cases………………………………………………………………………….. 26

TimeLine Chart…………………………………………………………………….. 9

Tools and Libraries…………………………………………………………………. 2

**U**

Use Case Diagram………………………………………………………………….. 32

User Characteristics………………………………………………………………… 10

**W**

White Box Testing………………………………………………………………….. 25

# REFERENCES

**WEBSITES:**

[1]. <https://www.c-sharpcorner.com/>

[2]. <https://www.infragistics.com/community/blogs/b/dhananjay_kumar/posts/how-to-implement-the-repository-pattern-in-asp-net-mvc-application>

[3]. <https://stackoverflow.com/>

[4]. <https://www.tutorialspoint.com/index.htm>l

[5]. **C# Docs :** <https://docs.microsoft.com/en-us/dotnet/csharp/>

[6]. **SQL Server Docs:** <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15>

[7]. **MVC Docs :** <https://dotnet.microsoft.com/apps/aspnet/mvc>

[8]. **HTML Docs :** <https://devdocs.io/html/>

**EXPERIENCE**

My experience at “Gateway group of companies” was wonderful as we were primarily trained with below technologies and platforms

* TFS (GIT)
* ASP.NET MVC
* WEB API
* UNIT TESTING
* AZURE
* C#
* CRUD OPERATIONS

I also got to learn project architecture with dependency injection, how to deploy system using Azure, how to publish website on IIS server.

The work culture at Gateway is one of its kind, they believe in work hard, Play hard, along with pushing our limits to work efficiently and effectively.

I have also got a job in “Gateway group of companies” through the campus placement.