Software Engineering Project Increment #1

Customer Communication Management System

Chigullapally Akhil Gupta,

Hemanth Sai Mandava,

Alexis Blackwell,

Kasi Gowthami

Group #3

# Table of Contents

[Table of Contents 2](#_Toc132552166)

[I. Project Details 3](#_Toc132552167)

[*A. Module Descriptions* 3](#_Toc132552168)

[*i. Increment #1* 3](#_Toc132552169)

[*ii. Increment #2* 3](#_Toc132552170)

[*B. Roles and Responsibilities* 3](#_Toc132552171)

[II. Goals and Objectives 4](#_Toc132552172)

[*A.* *Motivation* 4](#_Toc132552173)

[*B. Significance: Solution* 4](#_Toc132552174)

[*C. Objectives* 5](#_Toc132552175)

[*D. Features* 5](#_Toc132552176)

[III. Background Work 6](#_Toc132552177)

[IV. Methodology 7](#_Toc132552178)

[*A.* *Kanban Model* 7](#_Toc132552179)

[*B.* *Dataset:* 10](#_Toc132552180)

[V. Project Management 10](#_Toc132552181)

[*A.* *Implementation Status Report* 10](#_Toc132552182)

[VI. Challenges 13](#_Toc132552183)

[VII. List of Tools 14](#_Toc132552184)

[A. *Hardware Specification* 14](#_Toc132552185)

[*B.* *Software Specification* 14](#_Toc132552186)

[VIII. Results 15](#_Toc132552187)

[IX. Analysis 20](#_Toc132552188)

[*A.* *Activity Diagram* 20](#_Toc132552189)

[*B.* *Use Case Diagram* 21](#_Toc132552190)

[X. Implementation 23](#_Toc132552191)

[A. *Implementation of Project with the Help of the In Class Lectures* 30](#_Toc132552192)

[XI. Discussion 33](#_Toc132552193)

[XII. Conclusion 34](#_Toc132552194)

[*A.* *Current State* 34](#_Toc132552195)

[*B.* *Future Enhancement* 34](#_Toc132552196)

[Appendix A 36](#_Toc132552197)

[Bibliography 36](#_Toc132552198)

[GitHub Link 36](#_Toc132552199)

# Project Details

## *A. Module Descriptions*

In this section, the general module descriptions are given to describe which team member worked on which piece of the project. Each module aims to implement a certain feature for the contact management program. The features implemented by module are described in the following sections.

### *i. Increment #1*

* Module 1: Add Customer Information
* Module 2: Display Customer Information
* Module 3: Search Customer Information
* Module 4: E - Mail checking, Clear and Exit

### *ii. Increment #2*

* Update Customer Information
* Delete Customer Information

## *B. Roles and Responsibilities*

* Chigullapally Akhil Gupta (Created front end using Tkinter, documentation and presentation)
* Hemanth Sai Mandava, (Created modules to add customer details module, display customer details, documentation, and presentation)
* Alexis Blackwell, (Creating login page, Backend development using Mysql and connecting to python using pymysql, documentation and presentation)
* Kasi Gowthami (Created search Module, email checking module, and module to clear and exit, documentation and presentation)

# Goals and Objectives

The Goals and Objectives section was developed mostly during the Project Proposal phase where Project Proposal #1 and Project Proposal #2 were written and turned in for feedback. This section mirrors the project proposal deliverable [1].

## *Motivation*

As we are students of software engineering, the requirements analysis is the very Important step to develop efficient software the way the user wants it to be. So, the requirement gathering can be successful by communicating well by using contact management systems.

So communication is the most important aspect. And in the present world we have many online applications for communication like teams, but there is no simple GUI which allows us to create contacts and manage the contacts and save them to our database. Mobile applications and software do exist with which we can store in our cloud, but to simply store the contacts in our systems we do not have a very good customer interactive contact management system. And also, a perfect system app exists through which we can find the details of the contacts we create.

By using this Application customer contact management system companies can get closer with the customers and get leads then growth in their revenue, they can reach the customers through Emails, messages, social media and more, also they can concentrate on particular targeted areas like considering age, gender, and place.

## *B. Significance: Solution*

The main significance for selecting this project is the fact that businesses rely heavily on (CRM). Good CRM focuses on building a customer profile, saving repeat customer information, and providing techniques and software that can help different departments collaborate to meet customer needs [9]. Contact management systems are one part of CRM that focuses on building a customer profile by taking customer information such a name, email addresses, or phone number and saving that information in an easy to access, secure location. With this information a business can work on building a loyal customer base as well as identifying customer demographics for product offers and promotions. This benefits the business as managing customer information can lead to increased revenue through sales to repeat customers.

The reason that this project was chosen was due to the importance of gathering and storing information for analysis. While gathering and storing customer information may seem menial, it can have a profound effect for a business. Contact management systems are a part of customer relationship management (CRM). One research paper found that, “In summary, the case studies reveal that organizations often do not make good use of their CRM systems’ capabilities to obtain knowledge from their customers. Customers are a very valuable knowledge resource for organizations” [2, p. 39]. Essentially through the use of CRM and by extension contact management systems, businesses could obtain feedback from their customers to improve products, advertisements, or business processes. This project is interesting as it is essentially developing software that is critical for a company to function as well as storing and organizing data that can be used in other fields like data analytics or machine learning.

Other notable CRM projects could include the contact management software that is used by Salesforce which features chatbot integration [7]. Another contact management software that is relevant to the proposed project could be the software offered by Hubspot. This software provides the added benefit of lead generation [6]. The main difference between the software that is presented by these companies and the software that we aim to develop is the fact that this software solution would focus on creating a simple GUI that focuses on saving personal information and to a database.

## *C. Objectives*

The main objective of this project is to implement a solution and simplify the communication of software products which makes it very easy for the management and the coordination of different contacts, which is very difficult without software support.

And also, we can very effectively manage the contacts with the contact management solution software. With this application the communication barrier between the customers and the business owners is completely eradicated. Productivity is also another concern. A firm could completely see peaks in their business when the customers are well managed and satisfied. And also there is no fear of data loss, data accuracy. The ultimate objective of the contact management application is to make the updating process simple and accurate. To conclude the ultimate objective of this project is to enhance the customer relationship with proper communication and collaboration.

## *D. Features*

As we know the features of the software product is nothing but the specification or the functionality that is provided to the end users, by our software. So by ensuring all the functionality and specifications are met with the required modules we can ensure all the features are provided in our application.

Our application has features which help to add a new contact. And the details such as the First Name, Last Name, Gender, Age, Address, Contact and further more details could be added. We aim to make individual profiles for each person with which many customized queries and searches based on the need can be done. The dynamic control is the most important feature of the application. We can also delete unwanted numbers or contacts as we wish. The last but the most important feature is the possibility of the exporting of files so that a backup can be easily made [2].

The features that we aim to implement are:

* A GUI where the user can create, save, and delete customer contact information
* A method for creating a customer profile with new contact information [First Name, Last Name, Gender. Age, email address, Address, Contact, location, image ]
* A method for saving customer profiles to a database for later use.
* A method for deleting customer profiles.
* A method for updating a customer Profiles.
* A Login Page for employee to login to access software
  + Lockout feature if too many wrong attempts
* A Lookup Feature where the user types a name and application will search through the saved contacts to retrieve information.

# III. Background Work

*Table 1: Background Work*

|  |  |
| --- | --- |
| **Title:** | Role of Customer Relationship Management |
| **Aim** | Here the main goal is maintaining the relationships with customers. We need to understand the customer what their requirements are, what type of products and services are wanted. Collecting the details of the customer or stakeholder. Some examples:   * How are they purchasing what they want? * Recording the usage of social network sites * Usage of apps like Facebook, Instagram, WhatsApp.   These activities or information stored in the database, and it tells how much information we were used in the past, present and future. |
| **Background of project** | In the early period between 1990’s and 2000’s they setback the relationships of customers or stakeholders. In the past they collected data for marketing and sales purposes. To know how many customers buy the products or services? Based on that we need to collect the data. For that purpose, a customer management system is formed. Everything works systematically, we can easily come to know what the customer is buying and easily stored. Now databases are created here and everywhere by using advanced technologies like:   * Artificial intelligence * Machine learning * Cybersecurity.   Moreover, it will record the details of customers or stakeholders. Here the customer is nothing but whether it is an employee, businesses persons or all categories of people details are stored in the database system. |
| **Benefits** | Benefits of the project background are:   * Main important thing is data privacy. Each customer detail in the database is very protective. * There will be a strong tie between customers and the management system. The more known about your customer, the better the business can tailor their products and services to the customer.. * You can stay up to date with the customers' data. * The data is centralized. * Higher communications will automatically lead to increased profit from sales. * Services are good * Very effective to use * Highly sophisticated CRM.   These are the benefits of the customer management system. |
| **Objectives** | Here are some plans or objectives to achieve goal:   * The main objective is an efficient plan to boost the customer’s satisfaction. That means we can give too much attention to customer details. Based on that we can build personalized features to the product or service. * Focus on current customers. |
| **Scope of the Project** | It depends upon the organization, industry, size and complexity of the data as well as the focus on customer service. Enter the data very appropriately without losing data. We can add the details and retrieve the data whenever we want. |

# IV. Methodology

In Methodology we essentially talk about what we have done so far, how we implemented this project and how we chose to collect the data, by using the Kanban model.

## *Kanban Model*

This is an Agile project management tool designed to visualize the work. This can help both the models in Agile and DevOps.

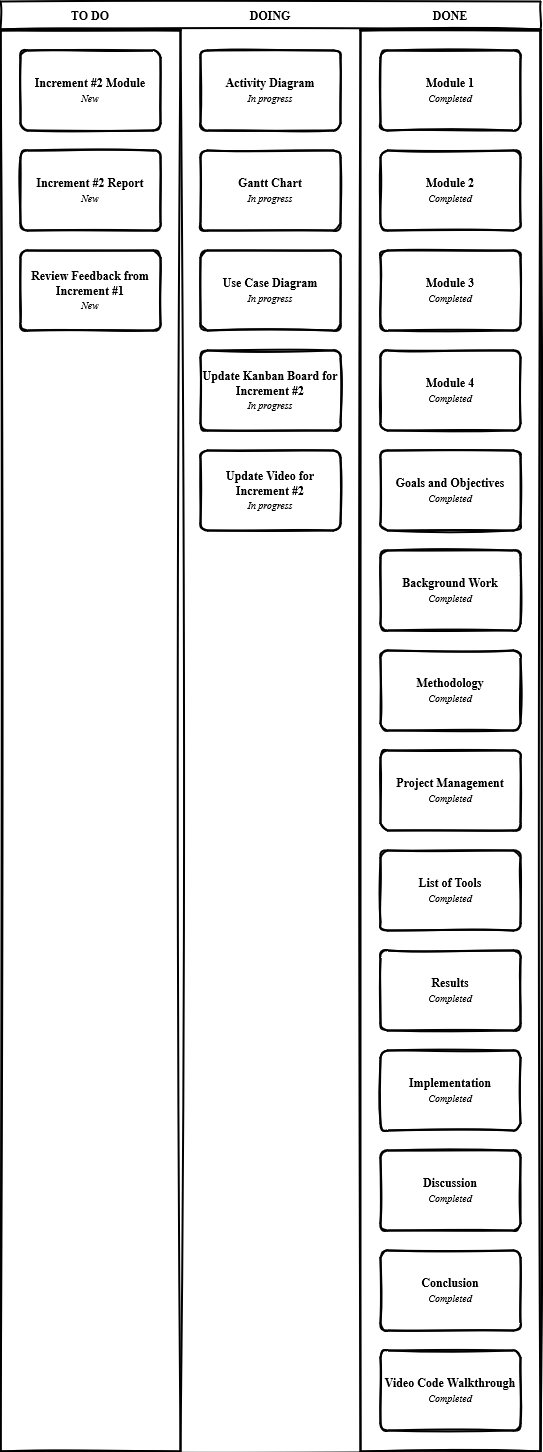
According to Figure 1, we have created a Kanban method for our project. Visualize workflow using a Kanban board. In this model framework contains Backlogs, Planning, Process, Developed, Tested and completed. The user story’s are are created and this is mainly concentrated on change management and service delivery, this is integrated into software based systems.

As per the below Figure 1, you can see how we have split and assigned story’s to complete our project. In order to do this, we mention the work that needs to be worked on for the Increment #2 module, report of Increment #2, and work on the feedback taken from the Increment #1. Then we have to work on the changes that need to be completed.

In the “Doing” module you can see the workflow of our project which we are currently working on. We have to work on Activity Diagram, Gantt Chart, Use Case diagram, Kanban model has updated for the next Increment #2, and video for the Increment #2.

In “Done” module, we have mentioned all the work we have already done for the project, so we have worked on different modules in the program like what methods to choose what kind of modules have to implement and Adding the customer contact, deleting the customer contact from the database and searching customer contact by their name, what to back end work and did multiple research about the project taken references from the previous contact management system and also we worked on Activity Diagram, Gantt Chart, Use Case diagram, Kanban model for the Increment #1.

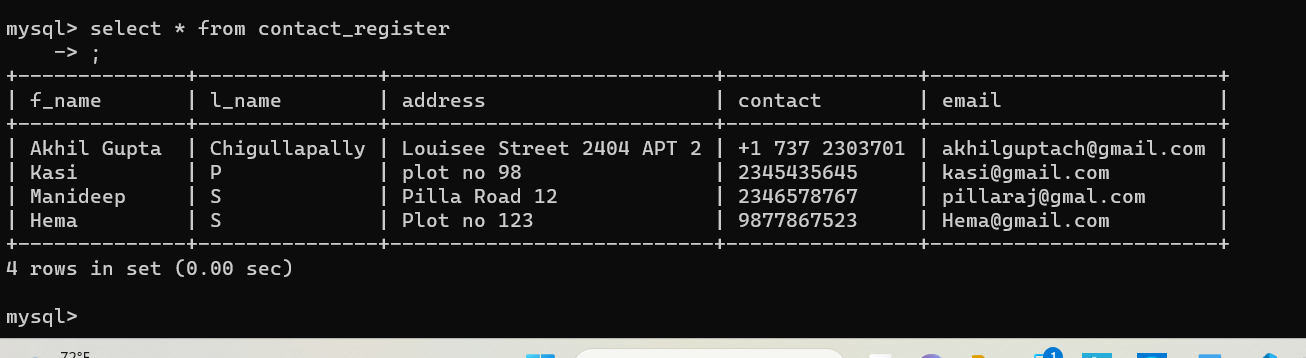
By using this Kanban model our work was assigned according to the deadline of the project, we assigned each and every small task in this Kanban board once the task is assigned the tasks will be managed and maintained..



*Figure 1: Kanban Board*

## *Dataset:*

In the Customer contact management system Dataset is important so we mention the data set of our project as an example as shown below. Once the user adds the customer contact details the data will be saved in this data set. As you can see in the Dataset below we have mentioned first name, last name, address, contact details of the customer and email. Even the customer data can be deleted from the application which we have done and also the users can search the data from the dataset through the application. Figure 2 shows an example of the Dataset.

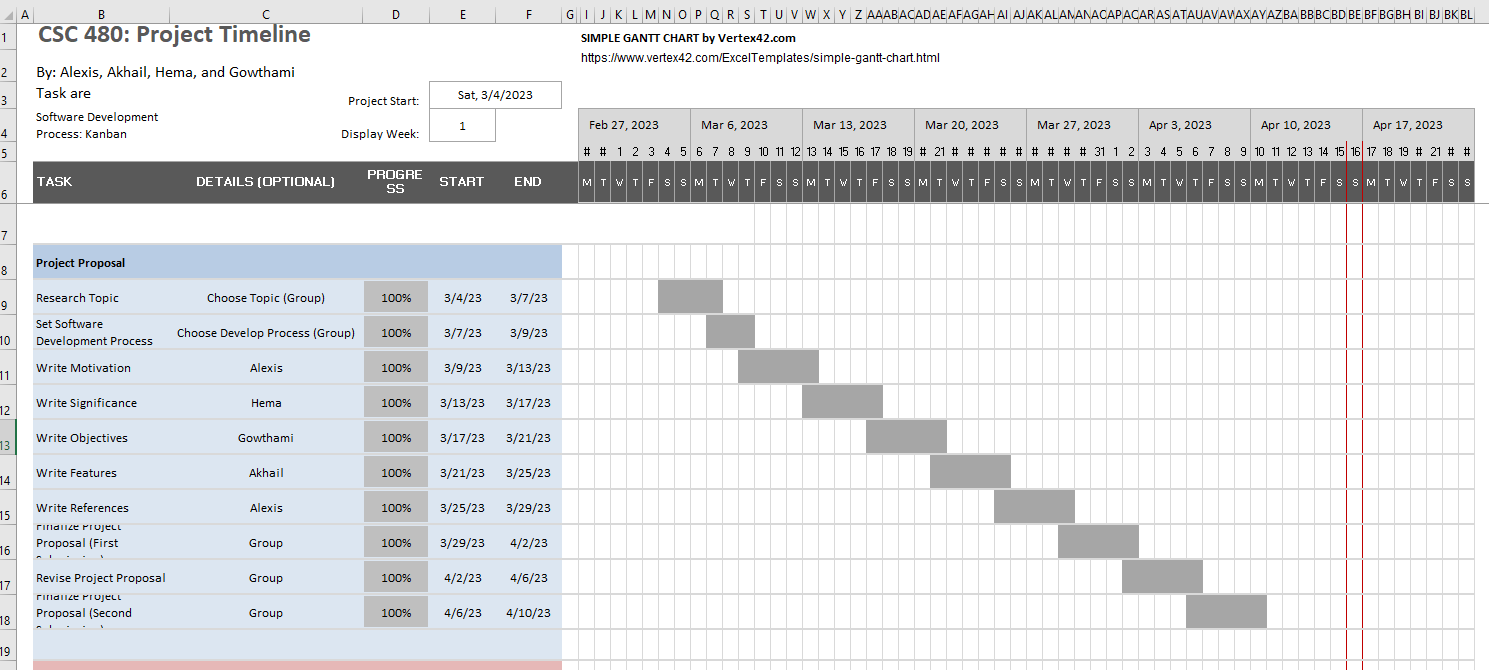


*Figure 2: Dataset Example*

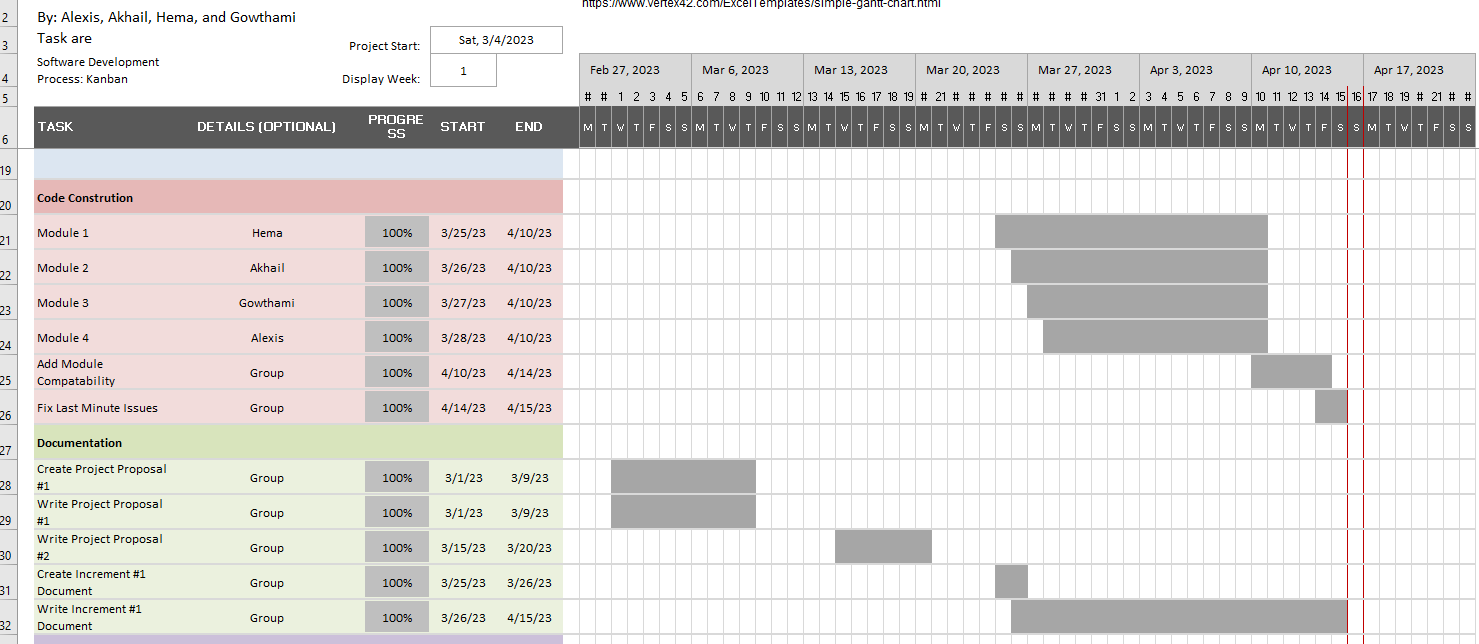
# V. Project Management

## *Implementation Status Report*

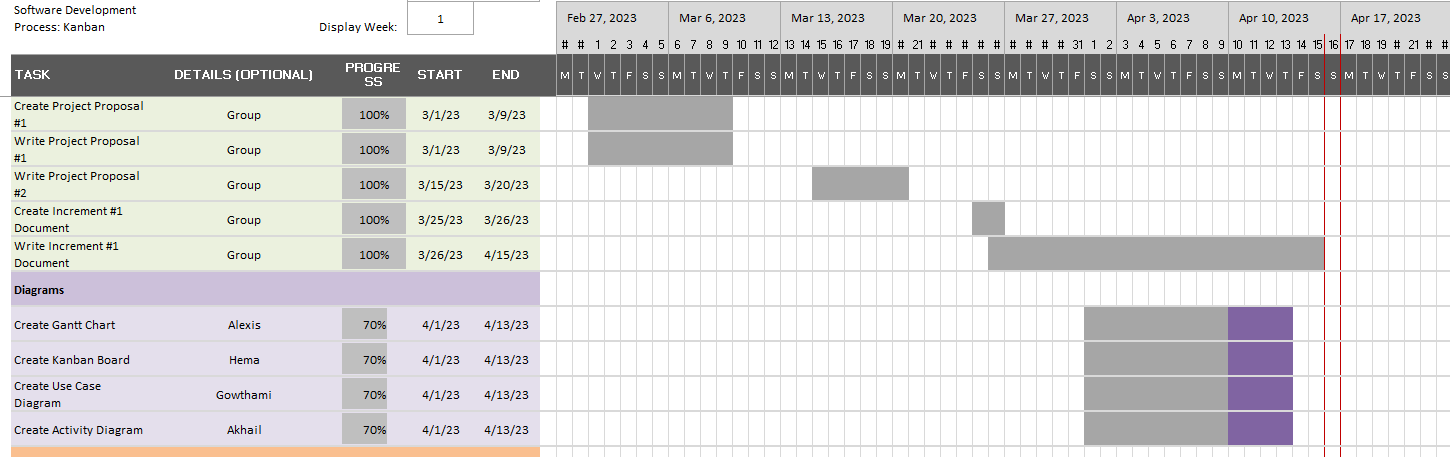
The work that has been completed has been in accordance with the requirements for this project being 70% completed with a future 30% to be completed at a later time. With this in mind a majority of the features have been implemented for Project Increment #1. The primary features that have been included are the implementation of Increment #1. These features are mentioned in an earlier part of the report in the Goals and Objectives Section.

Overall, the current state of the project is displayed through the use of a Gantt Chart. This Gantt chart displays what tasks need to be completed as well as details about that task, percent completed for that task, and a start date and end date for that task. The Gantt Chart was decided upon after considering the benefits of being able to visually see an overview of tasks completed, in progress, and which tasks have not yet started. This was used in conjunction with the Kanban Board to keep track of the software development progress. Figure 3 shows the Gantt Chart for the Project Proposal stage where the project proposal was drafted and written.

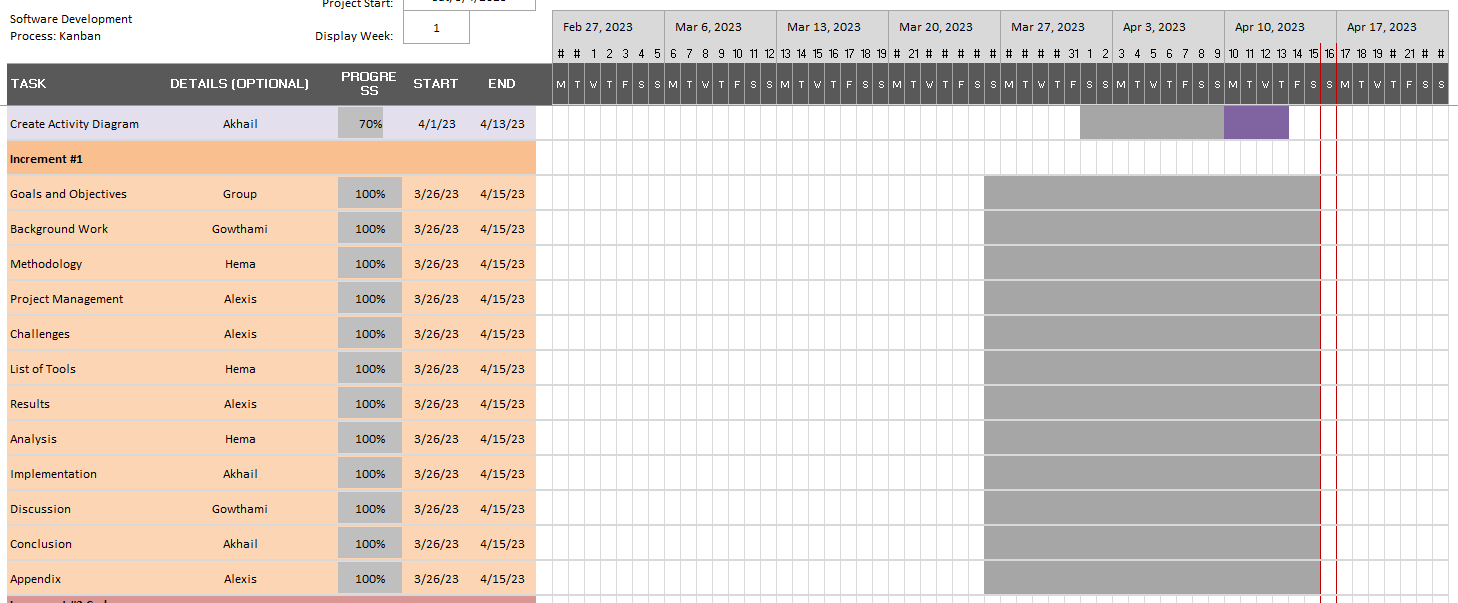
*Figure 3: Project Timeline*

Figure 4 shows the next step of the software development process where the code was constructed and relevant documentation written. The code construction was broken down into module where then each module was assigned to a group member. This is shown in the Gantt chart as well as when each module was started and then finished.

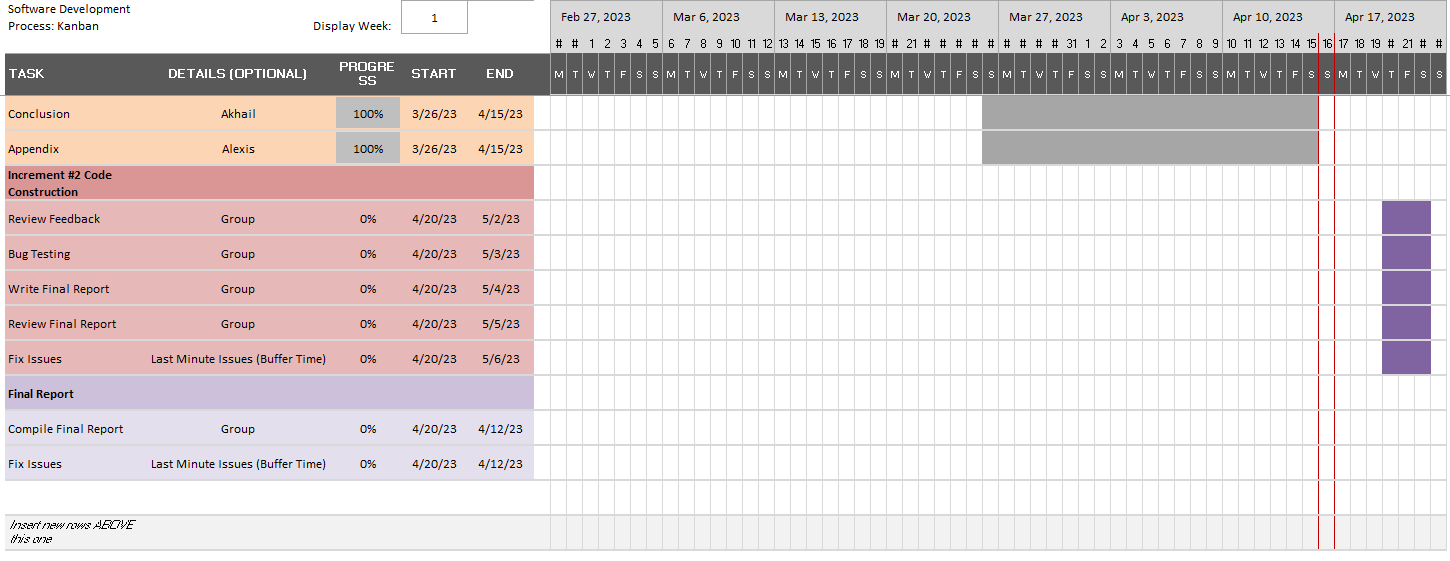
*Figure 4: Code Construction and Documentation*

Figure 5 shows the diagram creation phase when the diagrams that were created and finalized for Increment #1 occurred. These diagrams are not considered 100% complete yet as the diagrams such as the Gantt Chart and Kanban board will need to be updated for Increment #2. Instead, these diagrams as considered 100% percent complete for Increment #1 but only 70% complete for the project overall. 

*Figure 5: Diagram Creation*

Figure 6 shows the writing phase where the Increment #1 document was written based on the topic. This was done mostly at the same time in order to create an initial draft that the team could review. The goal was to assign a topic or section of the paper to each team member. Topics were assigned to team members who felt most comfortable writing about that topic. 

*Figure 6: Increment #1 Writing*

Lastly, at some point Increment #2 will need to be developed using the feedback given from Increment #1. This is tentatively planned out for the next upcoming weeks, but may be subject to change. This part of the project will include reviewing feedback, drafting the Increment #2 report, making any necessary changes to Increment #1, and adding the other 30% of the planned features for Increment #2.

*Figure 7: Increment #2*

As a whole this project has implemented the features that was stated in the project proposal and plans to include the update customer information and delete customer information features in the upcoming Increment #2. This would allow our project to roughly be 70% complete for Increment #1 and then 100% complete for Increment #2.

# VI. Challenges

The main challenges that occurred for this project were primarily centered around team organization and communication. These challenges mirrored some of the topics that were covered in class discussion as well as discussed in the class textbook [5]. Each of these challenges needed to be solved in order to move forward in the software development process.

The primary challenge faced during this project was the implementation of the Kanban Board methodology used to guide the software development. While other software development processes adhere to a more rigid approach where the development follows a set path, the Kanban methodology follows a more flexible approach. This flexibility can be a positive when working on a project where changes can occur and need to be tracked and managed [5]. Such changes occurred frequently during the development of this software program as different perspectives and design ideas were considered during development. However, part of the challenge comes from identifying the change, making appropriate steps to implement that change, and tracking that change so that all team members are on board. The Kanban Board worked well as details changed in the project, however it was a challenge to make sure to document those changes and include those changes into the project.

Another major challenge that occurred during this project was communication. This challenge was brought on by the fact that the group members had different schedules and other classes to take into account. Meetings in person had to be planned and communication in the group had to occur either through text or through annotations in the group Google document. While this was a means to communicate any changes or updates to the project, it was not as productive as the meetings that occurred in person. The results of those meetings would often lead to better communication and planning in the group that the group text or group calls. This is mirrored in the class textbook and discussion as face to face meetings are described as being the “better” form of communication as opposed to email or text [5]. Overall, the communication problem was mitigated by setting assigned meetings to go over milestones and progress as well as having a list of topics that each meeting would cover. In this way the problem of communication was managed.

Another challenge was the missing stakeholders that would normally be present in a software development process. This challenge was brought on by the fact that most software development projects include a client who is the one ordering the software for their business. Without a customer, several steps of the software engineering process needed to be assumed to continue during this project. For example, requirements engineering, which is where the stakeholders and the software development team meet in order to understand the project and refine the requirements provided by the customer, was not possible to perform [5]. Instead, the team met several times and made assumptions in order to understand the requirements for this project better. This was done in order to try and replicate what would occur during requirements engineering where the group would try to understand usage stories as well as clear up any ambiguity about the requirements from the client. The way that this was mitigated was through the use of a mediator. A mediator is recommended for handling negotiation between the client and the software development team [5]. In this case a mediator was useful for facilitating compromise when two team members disagreed about a certain feature or function. This allowed the team as a whole to agree on a set of requirements as well as clear up any vagueness about the requirements.

Also, it was difficult to get feedback from the client since there was no client present. A major part of the software development process is getting feedback from the client to incorporate into the project. The Kanban methodology as well as Agile and Scrum rely heavily on client feedback [5]. Without client feedback, it could be a challenge at times if the implementation of a feature or function was of acceptable standard. The solution was to rely on the other team members for their feedback where the team member would act as a pseudo client in order to give feedback on the implemented feature.

# VII. List of Tools

## *Hardware Specification*

Intel Pentium Processor

Speed : 1GHz to 2GHz

RAM : 512MB to 1GB

Hard Disk : 4GB to 30GB

## *Software Specification*

Language : Microsoft Visual Basic, Python, Visual Studio, Pycharm

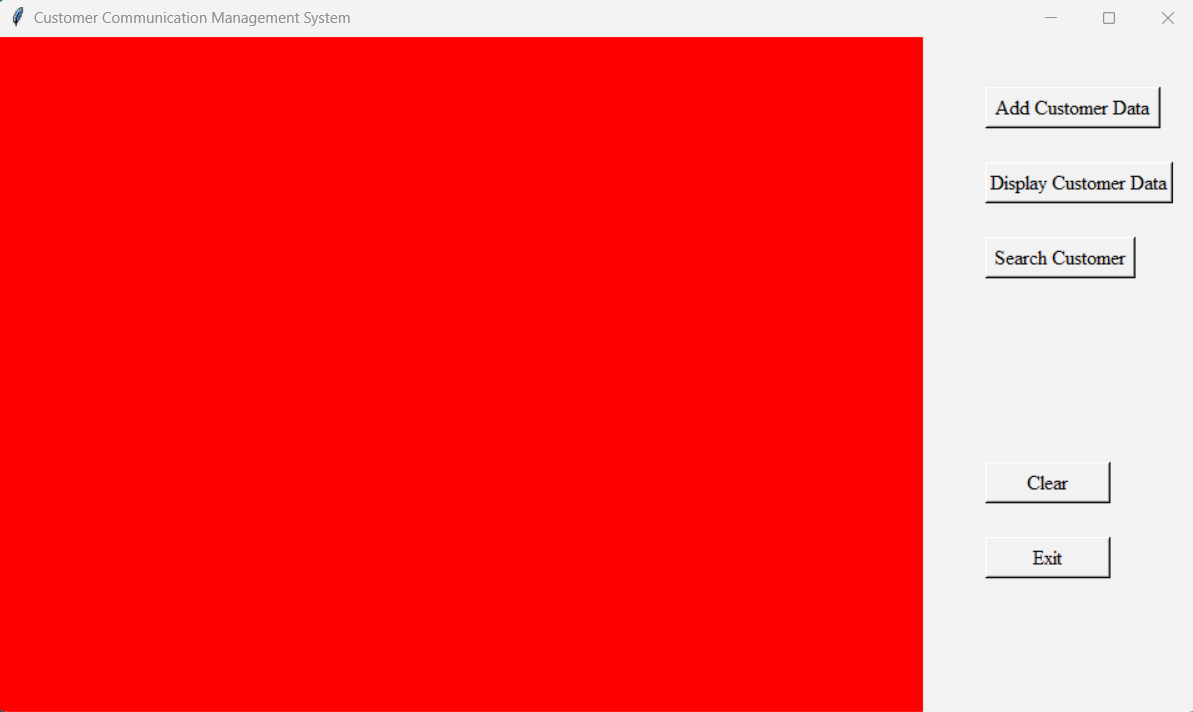
Database : MySQL

Operating System : Windows XP

To create this Contact management system application we have used multiple software, we used Python programming language to build this GUI application. By using tkinter, functools. We used MySQL for background to store the contact details and login credentials for this application.

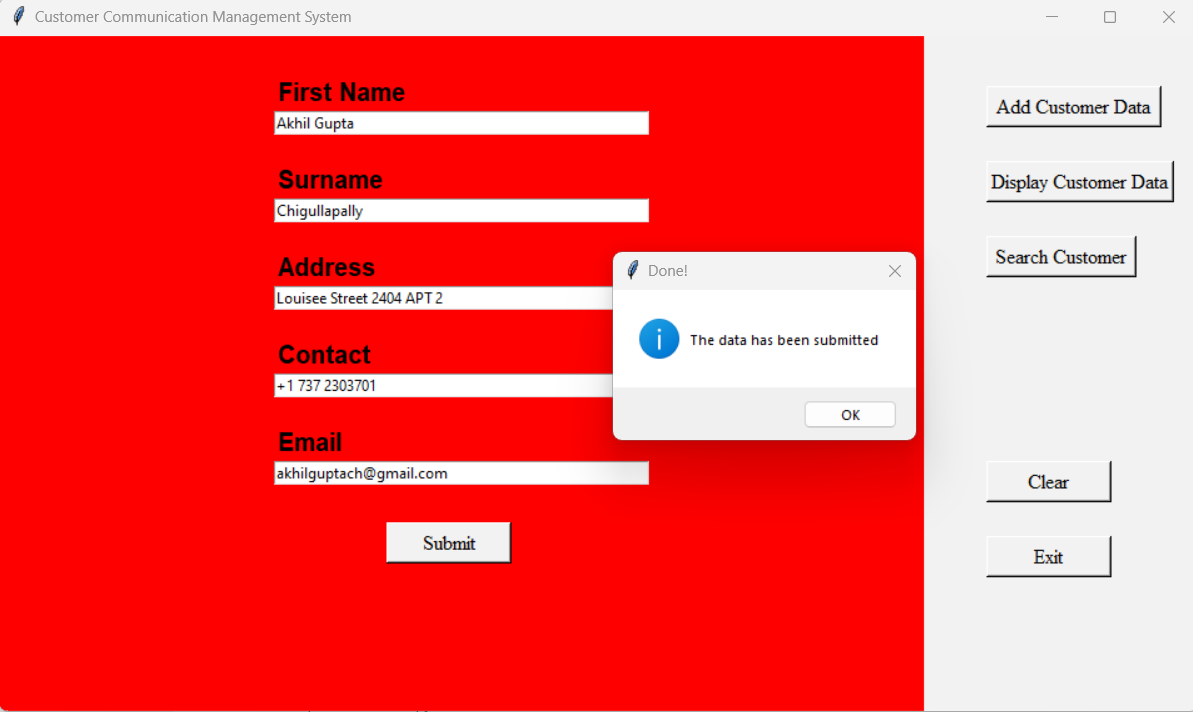
# VIII. Results

Here in the Results section, the goal will be to discuss the project for the contact management system.



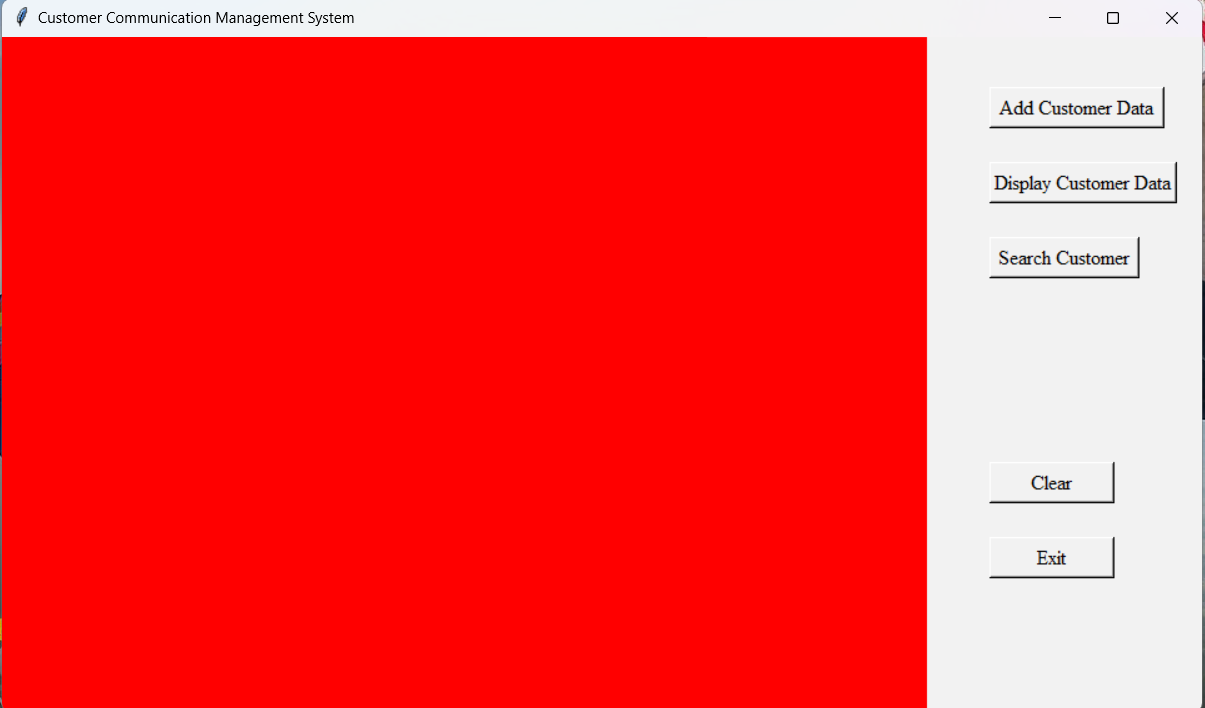
*Fig 8: Main Page*

In Figure 8, the main goal of our project is to add customer data and display the data which is existing in our database and search the data from the database and if your task is done you can exit from the page or else you can continue by choosing other modules.



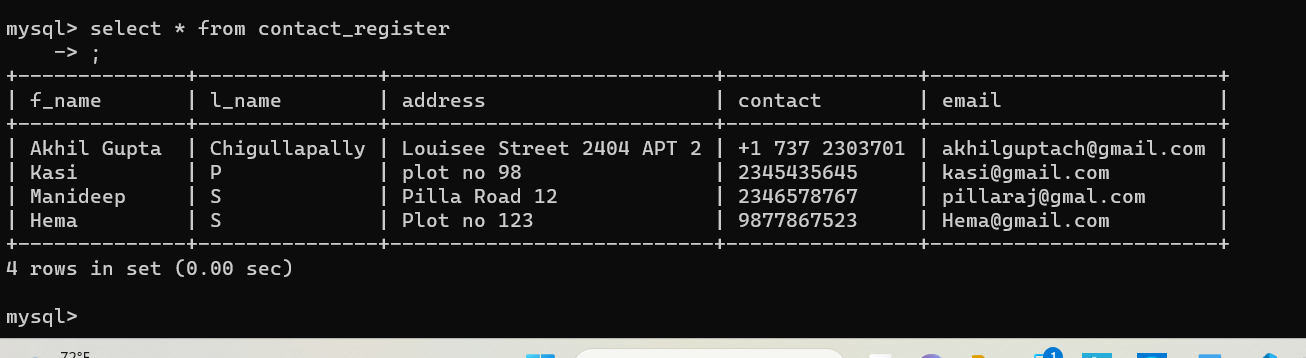
*Figure 9: Creating or Adding Customer Data.*

Figure 9 shows all about how we are creating and adding contact information to the database. As you can see we gave input details like name: Akhil Gupta, surname as : Chigullapally, then address as : louisee street, finally we can add contact as well as email and then submit the data by clicking on submit button . So here we will see the data has been submitted successfully by getting a pop up message. Once you get this pop up message you can have a clarity that data is stored in the database.



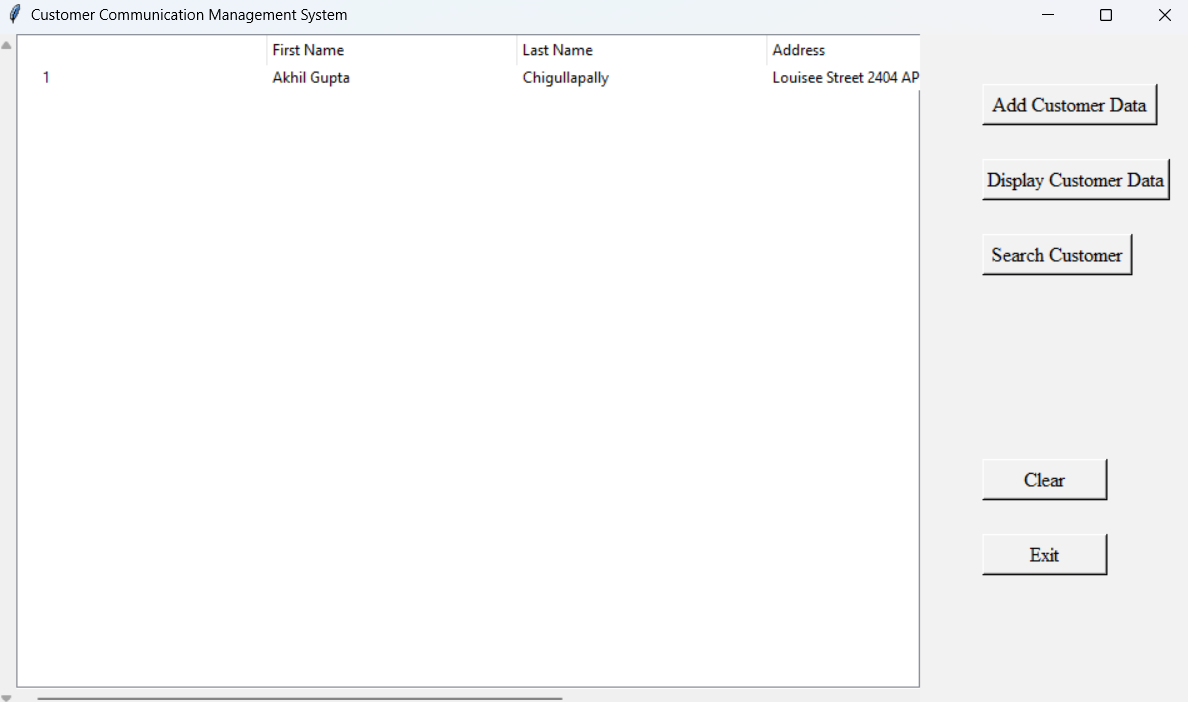
*Figure 10: Main Page*

In Figure 10, once we have submitted the data we will get to this main page, if user wanted to continue to add any contact they can do directly by clicking the Add customer data or else they can exit from the application.



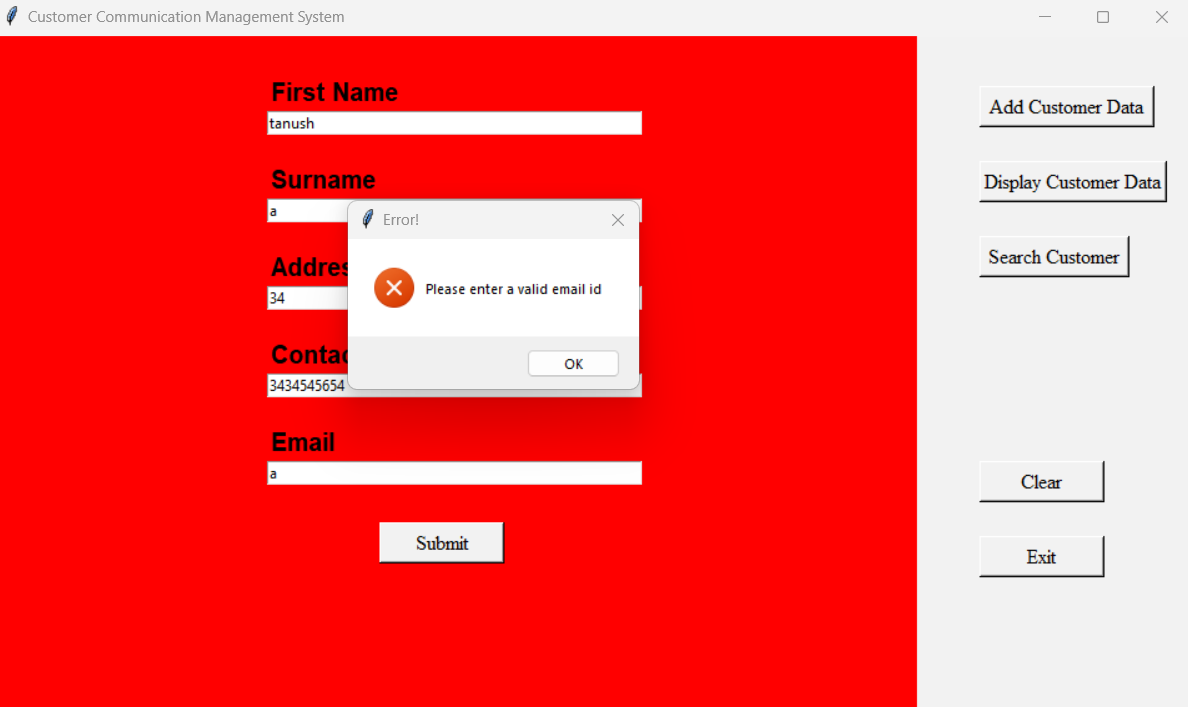
*Figure 11: Database of the Customer Data.*

In Figure 11, the data which we added will directly save in the database and, we entered the details in database MySQL. By giving command select \* from contact\_register we can fetch the data form the database.



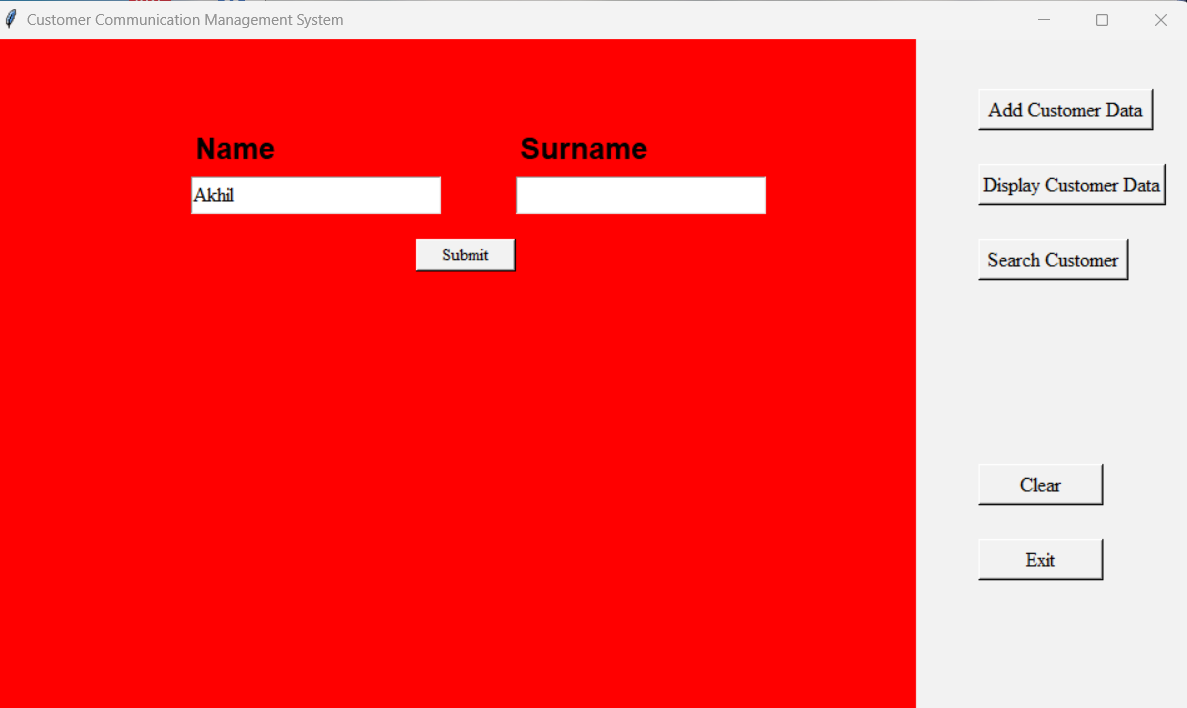
*Figure 12: Displaying the Customer Data*

Figure12, once we added customer data users can also view the data which is stored in the database directly in the application itself this feature makes access to the data easier.



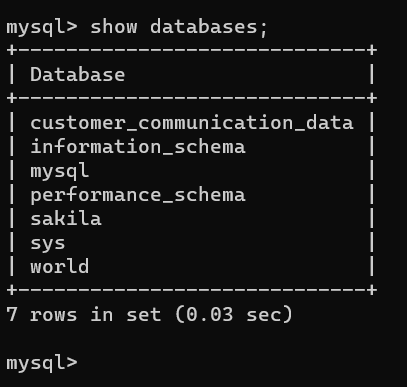
*Figure 13: Error Alert*

In Figure 13, you can see that if a user is given any invalid data into the application then immediately it will show the popup message that please enter the valid email address or phone number or if name of the customer.



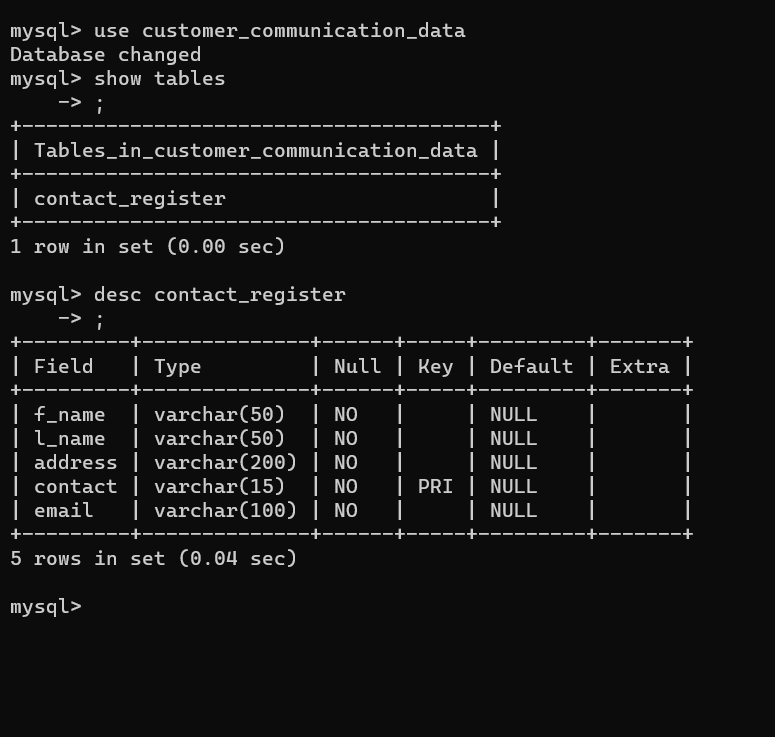
*Figure 14: Search the Customer Data*

In Figure 14, users are able to search the customer details. To search the customer details first the user needs to click on Search customer then you will get the page with name and surname. Then users need to enter the name and surname of the customer that they are searching for then they will get the customer email address, phone number and address of the customer.



*Figure 15: Database.*

The above Figure 15 is a database that we have connected to our application to store the data by using MYSQL. Figure 16 below shows the MySQL database table including the name of the inputs and the type of inputs that are accepted.



*Fig 16: Table Descending*

# IX. Analysis

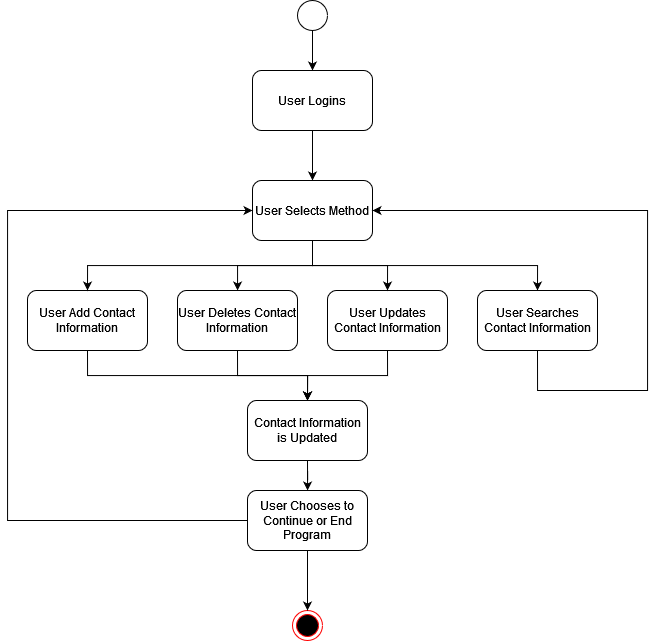
As we are students of software engineering, we have to analyze each and every phase of the project and the responsibilities that include for the Contact management system project are communication, requirements analysis, risk analysis, design model, program construction, testing, and support. So, analyzing is the very important step to develop efficient software the way the user wants it to be. So, we gathered and understood the problem then analyzed the importance of CMS.

We designed a couple of Analysis models for this project which gives better understanding about what we really require to project. In this we include solid functions for the designing tasks. Elements of analysis model, implemented Diagram to have a clear understanding of the project.

## *Activity Diagram*

To illustrate the use of an activity diagram and useful for the visualization and to document a software, this is a behavioral diagram in UML diagram to describe the active aspects and provide the clear idea of one activity to another activity of the system. This is an advanced version of flowchart.

The below Figure 17 shows the activity diagram for contact management software. First it starts with user login once user gives correct credentials then user can goto main page then user can select one method. User might create contact by adding new contact information that used to store in database, the next module is to delete existing contact information from the database, even users can also update the contact information also they can search any contact by using the name or email address or last name easily and user can continue the process or they can exit from the application.



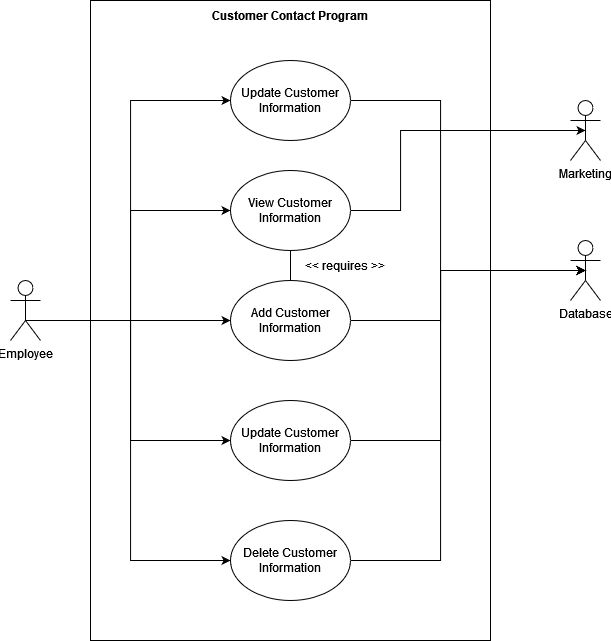
*Figure 17: Activity Diagram for the Contact Management Software*

## *Use Case Diagram*

The function of the Use Case diagram is to summarize the details of system users and create a professional diagram. The efficient Use Case diagram helps to discuss the scenarios in which an application interacts with the developers, organization or system and to achieve goals of an application and scope of the project.

In the Use Case diagram, all the details are used to demonstrate the different ways to interact with a system or users. To build interactions with the system use a set of specific symbols and connectors to connect specifications.

In this Customer contact program, Employees have access to customers' contact details. They can update the customer details, view the customer details and add the customer details. Then from there the marketing team can use those contact details to do marketing to those customers according to the products of their organization. Figure 18 shows the Use Case diagram created for this software product.



*Figure 18: The Use Case Diagram for the Contact Management Software*

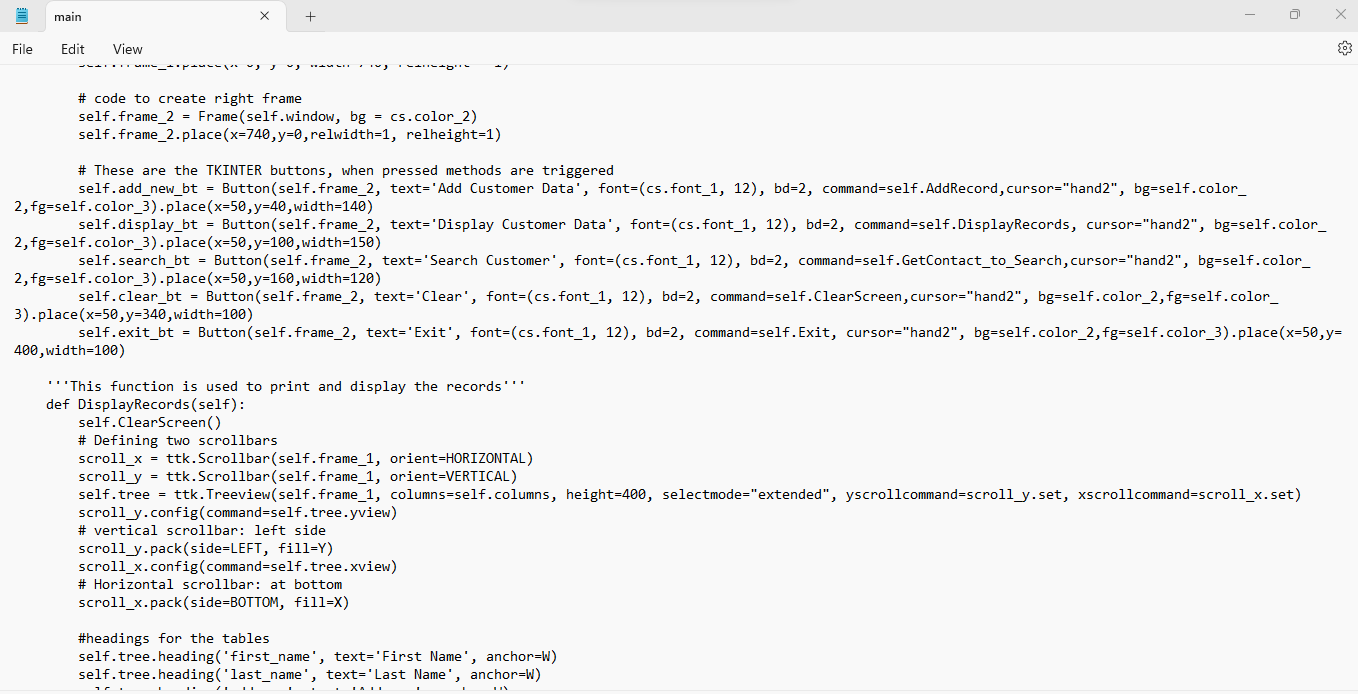
# 

# X. Implementation



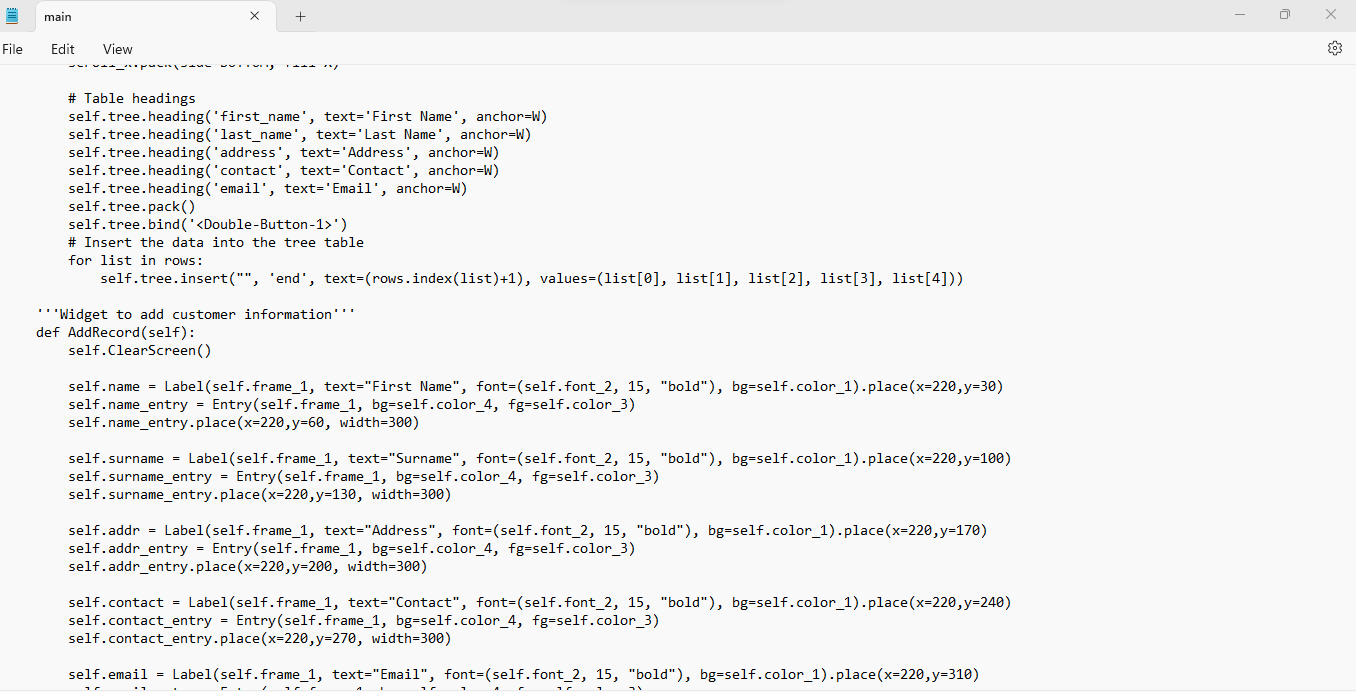
*Figure 19: Code for the Main Page*

In Figure19, here we have imported partial from functools module it will pass arguments to the functions and also passes from one another. Similarly, we have imported \* from the tkinter module for GUI. We have imported pymysql, imported messages from tkinter to get pop-up notifications, ttk considers widgets, imported program\_customization, represented as a ‘cs’. Here name itself customization settings, imported credentials represented as a cr for comparing the login credentials. Imported verify\_email represented as vl to verifying emails and get popup notifications. Class management is for management of classes in python. Here defined function def\_init\_ (self, root) here root took it as a graphical user interface. Here self.window assigned to root creates an instance. Title is given to the customer communication management system. The self-window background is set to white. Here we are fixing the values of the color as well as fonts and columns; these variables belong to program customization. Here all the connection attributes are taken from the credentials.py program from the local host, MySQL server, password and database name is used for connection.



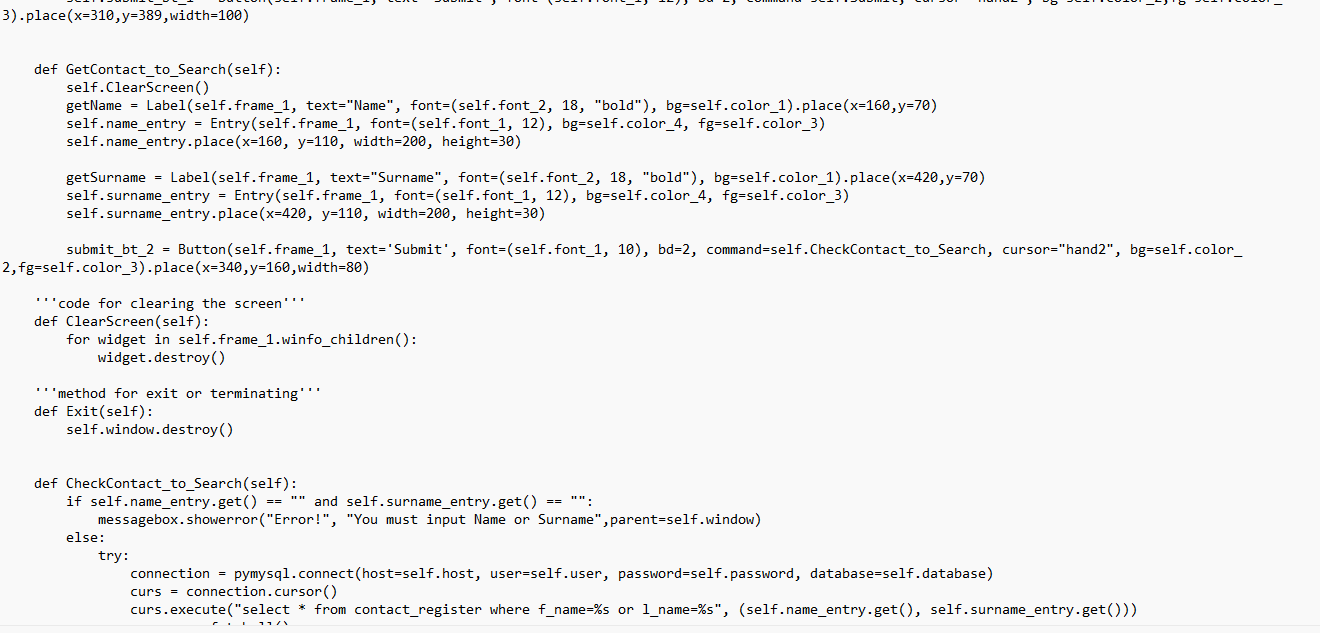
*Figure 20: Left Frame Creation*

In Figure 20 we are creating the left frame with the width of 740. Here we are given two self-frames like one frame for background and another self-frame to be placed for the given values like x-axis, y-axis, width and height. Code the create the right frame. Here we are given buttons, when they press it directly leads to another page. These buttons add customer data, display customer data, search customer, clear and exit. Secondly remaining things will print and display the data by defining functions like def DisplayRecords(self). Here the self-clearing function will clear things automatically. Here we define two scroll bars for horizontal and vertical by defining or giving a self-tree. It includes columns, height, select mode extended. Vertical scroll bar at left side, horizontal scroll bar at bottom.



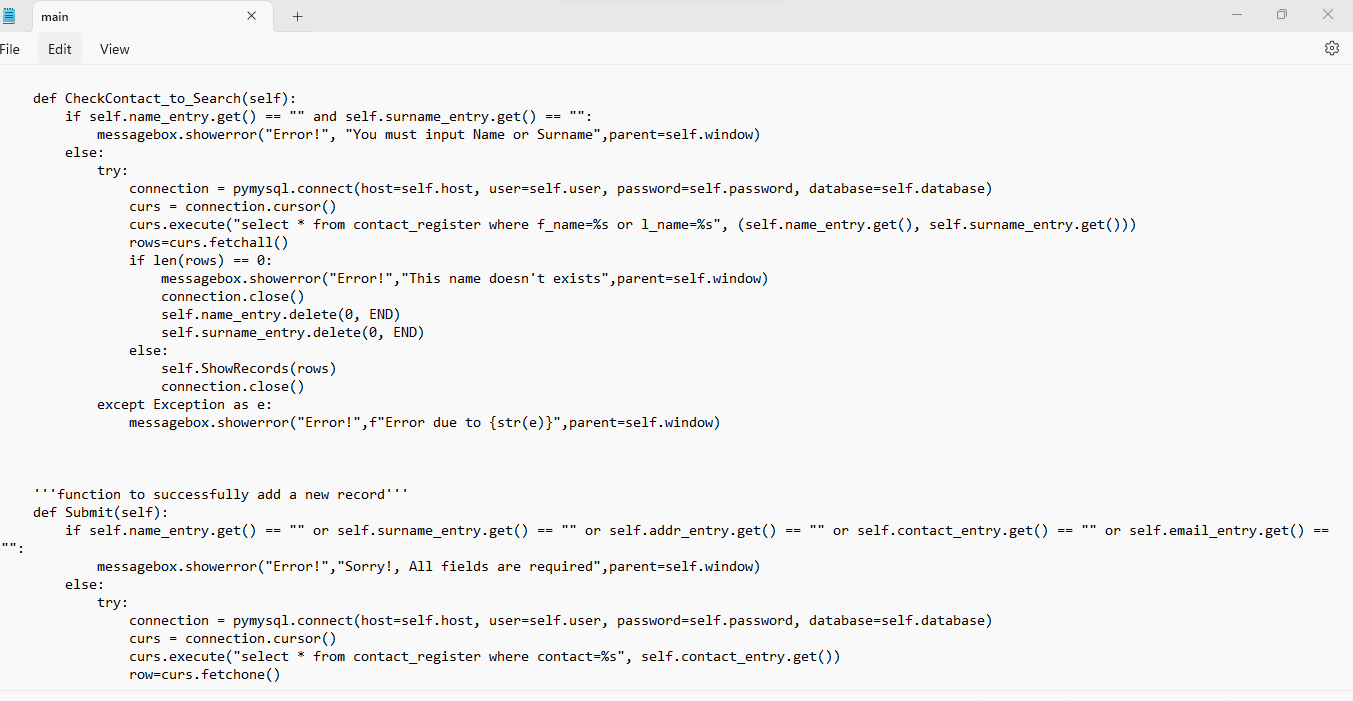
*Figure 21: Table Headings Creation*

In Figure 21, similarly for headings by using function self-tree, here headings include first name, last name, address, contact and email. Here connection is given which is connected to MySQL database for selfhost, self-user, self. Password, self. Database. It will execute all SQL queries then fetch the list of contacts from the contact register. It is very easy to fetch all contacts from MySQL database. If rows assigned to none the database will show empty, there is no data to show. So, automatically the connection closed and clearing the screen as well otherwise connection will be closed. Otherwise, it will show error by giving pop-up window. Here the def Show Record function is used for showing record from MySQL database. It matches the criteria, and it will display the subset of data.



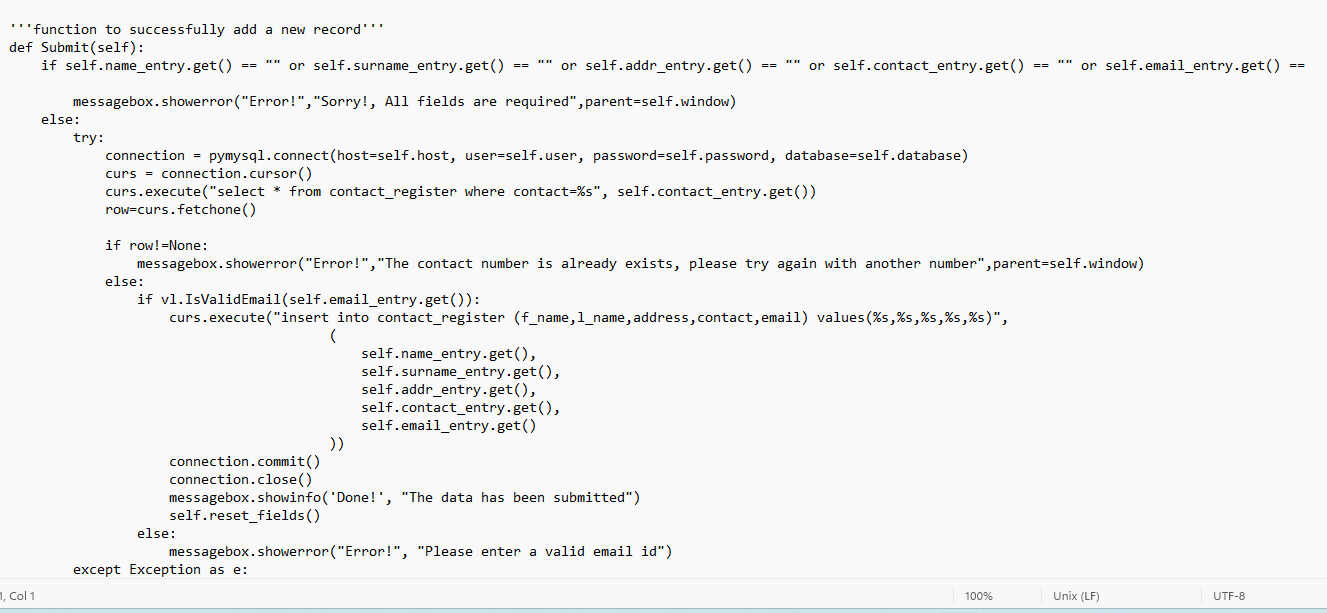
*Figure 22: Table Headings*

In Figure 22 the table headings are defined by using a function called self-tree for first\_ name, last\_name, address, contact, email. Insert the data into the tree table by using for loop. Similarly, widgets are added to the customer information by defining function called def add record(self). After that it will clear the screen. For every variable of self like name, surname, address, contact, email, we created widgets by defining labels. For every widget self-background adjusted as well as we created a place by giving x-axis, y-axis, and width.



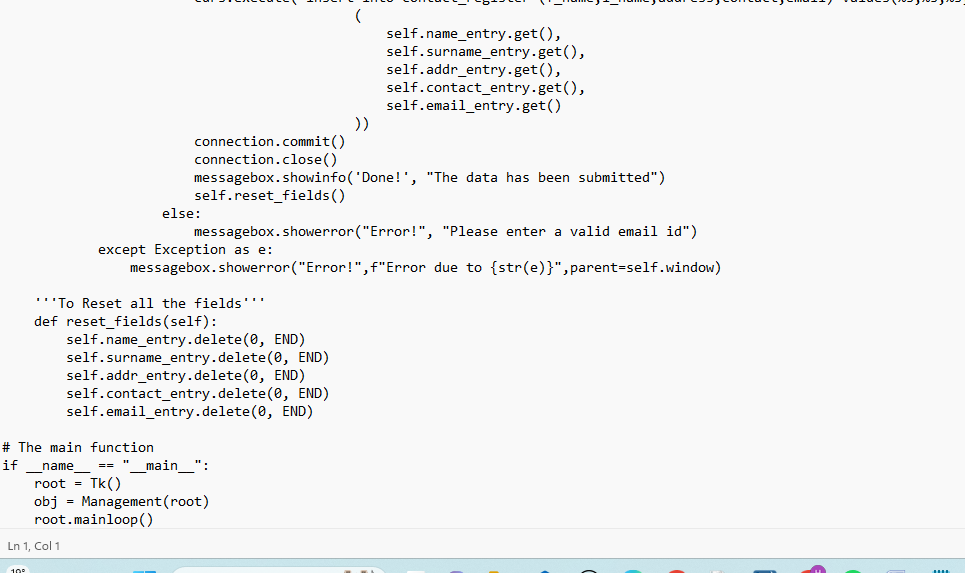
*Figure 23: ChechContact\_to\_Search() and Submit() Functions*

In Figure 23, to get name here we are using get contact\_to\_search(self) function is used as well as get name function is used for that we defined label, entry, place for name function. Similarly, the get surname function is used for that we defined label, entry, place for name function. Code for clearing the screen we are using clear screen function by defining it and then destroy the widget Secondly exit by defining exit(self) and destroy the window



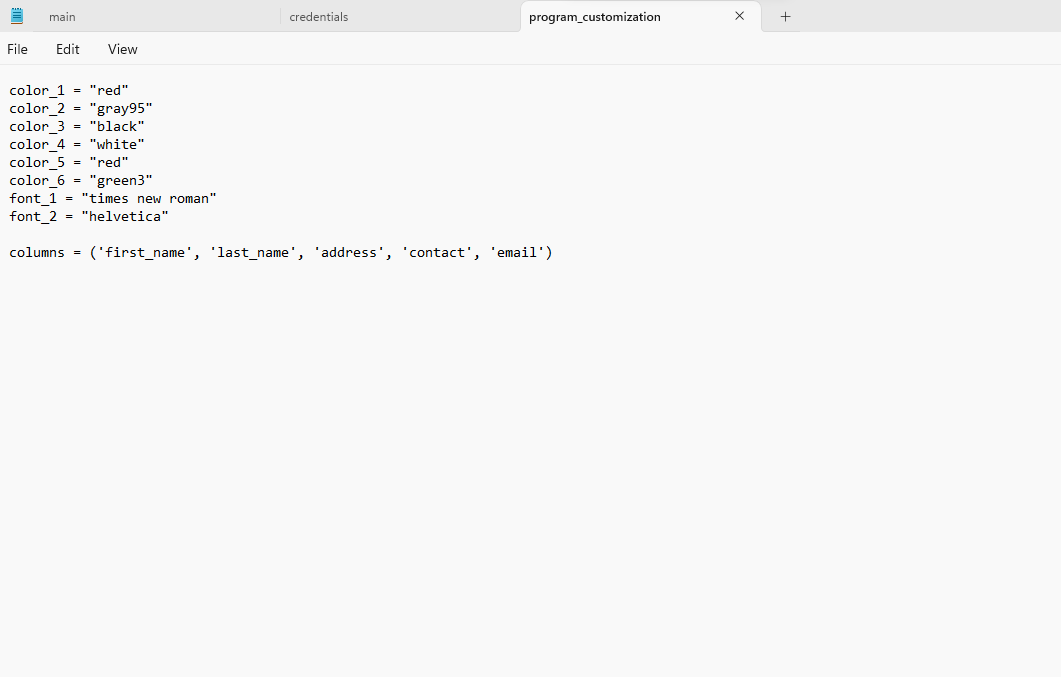
*Figure 24: Submit() Function*

In Figure 24 here an if else method is used as well the MySQL Connector is given to connect the database.



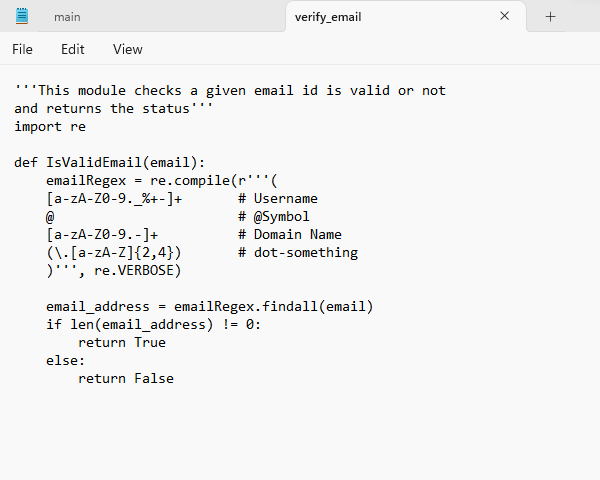
*Figure 25: reset\_fields() Function*

In Figure 25 the function to successfully add a new record is implemented. By defining a function called submit(self) to reset all the fields def reset\_field(self) to delete for name, surname, address, contact, email entry. And the main function is called.



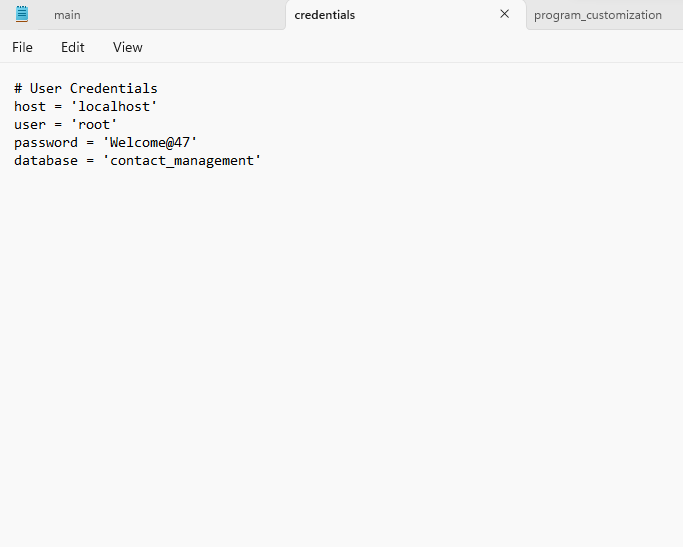
*Figure 26: Program Customization*

In Figure 26 here we can see variables like colors, fonts and columns like color\_1 is assigned to red, color\_2 is assigned to gray95, color\_3 is assigned to black, color\_4 is assigned to white, color\_5 is assigned to red, color\_6 is assigned to green3 as well as font\_1 is assigned to times new roman, font\_2 is assigned to Helvetica. Columns are assigned to first\_name, last\_name, address, contact, email.



*Figure 27: isValidEmail() Function*

Figure 27 is where this module checks whether this email is valid or not as well as it shows the status whether it is returned or not. Here the definition function is given like def IsValidEmail(email). This function returns the value whether it is Boolean or not. That means it will return either true or false. Here I gave the username, symbol, domain name, dot-something.



*Figure 28: User Credentials*

In Figure 28 straightly we are making a point these are user credentials which are connected to the MySQL database. Here the host is assigned to the local host that means the host is running by connecting to a MySQL database on the same platform. Generally, the user is assigned to root which is helpful for installing MySQL. Thirdly, a password is assigned to keen, which is helpful to authenticate the user values whereas, keen is used for privacy protection to keep in private for user details, mails, name, mobile number etc. MySQL database is finally used for creating tables that means entering details will be created in the form of tables. So, we can store the information. To retrieve we can select the customer details that are given that details will display in the database where it is stored.

## *Implementation of Project with the Help of the In Class Lectures*

Previous sections you would have noticed the details on the implementation of our project. Now we are going to say how our classes and syllabus of the different chapters taught in class had helped us in the successful implementation of our project.

So implementation is one of the most important phases of any software development process. Here the process is nothing but the steps of activities needed to develop a software product which is efficient and effective. So first with the help of the tutorial or the classes we were able to understand what the process is, and how we can give you an efficient and effective process based on the type and nature of the project we are going to deal with.

So in Chapter 1 we have learned about exactly what a software is [5]. What the different types of softwares are, and how they can be used in different scenarios by different users. Here we have also learned the importance of the documentation with which we wanted to ensure that our project is properly managed and well represented by the documentation. Here we have learned about the different types of softwares that exist such as the application software, system software, Web applications, mobile applications. With the exposure to the knowledge on the different kinds of software we were able to decide which type of software we have to implement for solving our problem. So here for managing the customer details we have decided to create a web application with proper GUI by using python, as it is very easy. And also proper modular structure can be provided with the help of python programming language.

Here in this chapter we have also learned about the process framework activities such as the communication, planning, modeling, construction and deployment. So by understanding these activities the way they work and the procedure or guidelines to follow to make sure to successfully finish these activities we can develop a good software work product. Here communication is the phase where the project team members get in touch with the project stakeholders such as the customer, end user and project manager, and the main task here is to gather all the requirements of the project by performing a requirement analysis on the problems of the task to be done. So here in this project all the team members had a meeting together to discuss the project and the requirements we needed for the successful implementation of the project. We have analyzed our requirements perfectly by knowing the type of tools required, analyzing the software and hardware requirements and the main task is to analyze the different features the client or the customer wanted. So we have properly conducted the requirement analysis.

Now as all the requirements are ready now we had to plan for the project implementation. Here we have created the blueprints and the paper prototypes for our project. We have properly understood the umbrella activities by which we have established proper milestones and targets. We have analyzed proper modular structure and have planned for the module increments. Then after proper planning and estimating we have modeled our project. We have constructed the proper unified modeling language diagrams to make sure all the core components and features are well established. And all the design structure and the functionalities are well understood now we have moved on to the construction phase. Here we have learned about the importance of the coding and testing activities which take place in the project. Here the coding is the main part in which we have to properly implement all the working models by using the plans, blueprints and the requirement features which we have analyzed. We have implemented the code in python and by using different libraries and tools we were successful in the coding part. The main reason for the successful implementation is that we have properly allocated different tasks to different team members like we have divided our project into different modules and made sure that all the modules are divided equally among the team members.

Once the implementation part is done we have to test our project and we have understood the need for testing through the process activity. And at last for the deployment our task was to make sure that there are no errors and malfunctions and then we were ready to submit our project. Now the user can easily use the python application with very simple and easy steps. And also in this first chapter we have learned about the seven hookers principles. Which helped us to focus and achieve our objective. Here the first principle says that there is a reason for every project to exist. So here when we are implementing the project there is a specific goal and objective for every project. And with the help of these rules we were able to understand and focus on the plan of our project. And also this principle makes sure we keep our work simple and stupid. So it is always important that the design of our working product is very simple and is not very hard so that naive users cannot use it. So we have ensured that the design is very simple so that it can be used by everyone in an easy and effective manner.

From Chapter 3 we learned about agility. Here agility is the pervasiveness of change. We learnt that software engineers are always on their feet if they are accommodated into that. Here software engineers and stakeholders or customers should work on an agile team. A self organizing team, and everything should be in control if possible when they work mutually with coordination and understanding between them. It is very important because we observed that now-a-days modern business environment is based on computer based systems and keeps on changing technologies and is quite challenging. At the same time we learned it is important to deliver the software product with a high success rate within time and deploy the software product within a customer environment. Here we see framework activities play a vital role like communication, planning, modeling, construction and deployment delivery. Generally I observed the politics of Agile such as how Agile is more self organizing and due to this we are not expecting the team should be self-organized. It is highly impossible because in a team some engineers aren't able to do the work within time, some people can't attend the meeting in time, some might don't have understanding between them. So automatically it leads to a product failure without having good understanding between software engineers and stakeholders.

Secondly, the Scrum process is also very popular with the Agile software development team. We learned backlog refinement meetings with customers to create a backlog, partitioned sprints, daily Scrum meetings as well as sprint review and sprint retrospective which needs improvement. The main thing we observed is that it is quite difficult to handle the cost as well as we are expecting team members which are highly advanced in content and context. Moreover this does not work for larger projects. Instead, these methodologies should be used for smaller projects. Due to this reasoning the group chose to implement Kanban board for the software development methodology.

Thirdly, we observed that the Kanban board will show the list of what to do and the process and the final outcome But Scrum is a different process, this Scrum is a process to plan and project in a particular time and budget. In order to visualize the workflow process in a Kanban board, the task states are arranged into columns which represent the development stages. By limiting the work in processes at any given time, this reduces the time and improves the workflow. This method works by making process politics better.

In Chapter 7 “Understanding requirements” we have learned about the importance and the need for understanding the requirements to ensure that the project is perfectly managed and maintained [5]. Here different phases exist such as the Inception, Elicitation, Elaboration, Negotiation, Specification, Validation and Management [5]. Here through the Inception phase we were able to understand the scope and the need of the project, by analyzing the scope of the project now we were able to estimate the time requirements and the risk factors for our projects. In the Elicitation phase we were able to understand all the requirements and they were properly elicited. All the user tasks are identified. And again in the next phase in the Elaboration phase we had to refine all the requirements which we have gathered. Here we had to make sure that we classify all the requirements based on the user needs, function based, class based and interaction based. Once all the requirements were refined, now we had to negotiate within the team members. We have made different efforts to make sure that all the tasks have been prioritized and then we have removed tasks which are unconvincing and that are not doable. Then we have specified all the requirements and the features and mostly the walkthrough is the same as the software requirement specification. Then we validated and managed the project by usability testing. So here the different phases which we have learned in class has helped us to make sure that at each and every step we were successful and effective.

# XI. Discussion

In the existing systems there are many online platforms such as web sites and cloud based systems which can have the features of saving and managing the user or the customers data like that of data which might be useful for communication. Many of the customer relationship management systems exist such as Thryv, NetSuite, Salesforce Sales Cloud, Zoho CRM and many more.

But the main disadvantage of these customer relationship management systems is that your customers' data is completely centralized. Here centralized is nothing but your data is stored in a database which is in the form of a distributed type of database. Here distributed databases are nothing but the different databases connected through a network, which are logically interconnected and where the data is replicated or fragmented with them to provide a high availability of data to the customer. Like if one database fails then the other database has the replicated data. So this centralized data can be at risk by the attackers or hackers.

Another disadvantage is they are too costly. We need to subscribe for the monthly subscription which they offer. So, here we need to pay as we are using their services. And also we need to train and learn on how to use these platforms.

The main ultimate disadvantage of these websites or platforms is that they are not even customizable. There are already some of the predefined data elements or structure which has been hard coded. Here whether we are in need of the information or not in need of a particular feature we will be getting access to it and as a whole we need to pay for it.

So by using these management tools we most often end up paying for the features which are indeed not required for our organization. So based on reliability, security and feasibility these management tools are a drawback for many organizations.

The paper “Customer Relationship Management (CRM) implementation: A soft issue in knowledge management scenario” focuses on the implementation of the customer relationship management. It gives a complete scenario on the scope and the nature of the problem as of the inception phase in the requirement gathering [2]. Since the customer relationship management is the combination and aggregation of different types of elements or attributes such as the people who work there which are the people, the methodology or standards used by them for the implementation of the software problems and to make sure to deliver the product on time which is nothing but the process which the steps of the sequence of activities which are required to be successful. And at last the technology. So this paper states that there are many difficulties in creating customization management systems for the users.

This thorough discussion and analysis with the help of background work has led us to create an independent and individual working python application with a graphical user interface which allows us to manage the customer data as required by us. So here we can add data and manage data the way we want to. Here the data is not centralized and is locally saved in the mysql database. So we also do not have the fear of loss or mishandling of the data by different outsiders or attackers.

And also we only pay for what we use. Here, based on the number of modules which we are using, that is the required number of features that we need, only those features will be implemented as modules and we can pay only for what is needed for us.

So more sophisticated customization is possible in the python based GUI application for customer information management system.

# XII. Conclusion

So to conclude here we have been able to successfully implement a python based Graphical User Interface application through which we can easily and effectively manage the customer data which is to be gathered and maintained for proper customer relationship. As we know stakeholder is a very important person in determining the flow and the output of the project. So here we have successfully created the application through which the contact details and the communication parameters never can be lost.

And also the system or the application which is developed is very simple and easy to use. Any naive user who is not even a basic programmer can use the application. The graphical user interface which has been generated is very simple and customizable. So in conclusion our project helps people to manage and handle their customer data effectively and efficiently. So any information now through our application is safe and secure.

## *Current State*

In the current state or the project increment we have made the application which interacts with the customer by prompting for data input and also at the same time the customer's details can also be seen and viewed as required by the user. Here the application completely interacts with the database where we can add the customers data by inserting the data into the database. So in this increment we have been able to successfully implement the application which can perform different activities such as to add the customer information, search for the customer information. Display the whole customer's information, verify the customer's communication details such as the electronic email, clear the data in the application and terminate the application. These steps are very simple and effective to be used by the project managers or users.

## *Future Enhancement*

For the future enhancement we would like to add more features to the application which can perform more tasks. Like we would like to add more features and functionality like to update the customer details whenever required to do so, by using the update query in the mysql where we can update the existing data. And also we can use the drop data to delete the data based on a particular attribute. So we would also like to add a feature to delete the customer information when required by the user. We would also like to add features to establish more security to our python application.

# Appendix A

## Bibliography

[1] Blackwell A., Gowthami K., Gupta C., Mandava H., (2023). *Project Proposal #2*. Unpublished manuscript, University of North Texas.

[2] F. Khodakarami and Y. E. Chan, “Exploring the role of Customer Relationship Management (CRM) systems in Customer Knowledge Creation,” *Information & Management*, vol. 51, no. 1, pp. 27–42, 2014.

[3] M. Nath and A. Arora, "Content management system: Comparative case study," 2010 IEEE International Conference on Software Engineering and Service Sciences, Beijing, China, 2010, pp. 624-627, doi: 10.1109/ICSESS.2010.5552271.7

[4] N. H. M. Ariffin, A. R. Hamdan, K. Omar and N. Janom, "Customer Relationship Management (CRM) implementation: A soft issue in knowledge management scenario," 2012 IEEE Colloquium on Humanities, Science and Engineering (CHUSER), Kota Kinabalu, Malaysia, 2012, pp. 485-489, doi: 10.1109/CHUSER.2012.6504363.

[5] Pressman, R.S. and Maxim, B.R. (2019) *Software engineering: A practitioner's approach*. 9th edn. Boston, MA: McGraw Hill.

[6] “Software, tools, resources for your business,” *HubSpot*. [Online]. Available: https://www.hubspot.com/. [Accessed: 20-Mar-2023].

[7] “The customer company,” *Salesforce*. [Online]. Available: https://www.salesforce.com/. [Accessed: 20-Mar-2023].

[8] “Unity Customer Data Platform,” *Oracle*. [Online]. Available: https://www.oracle.com/cx/customer-data-platform/. [Accessed: 09-Mar-2023].

[9] “Why you need [the right] CRM,” *Oracle*. [Online]. Available: https://www.oracle.com/cx/what-is-crm/why-crm-is-important/. [Accessed: 09-Mar-2023].

## GitHub Link

<https://github.com/blackwell7/Software-Engineering-CSC5430-Project.git>

## 