ASHESI UNIVERSITY

A DATA MANAGEMENT PLATFORM AND BULK SMS PROGRAM FOR THE RITZZ EXCLUSIVE GUEST HOUSE

APPLIED PROJECT

B.Sc. Computer Science

Mohamed Hijazi

2020

ASHESI UNIVERSITY

A DATA MANAGEMENT PLATFORM AND BULK SMS PROGRAM FOR THE RITZZ EXCLUSIVE GUEST HOUSE

Applied Project

Applied Project submitted to the Department of Computer Science, Ashesi University in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science.

Mohamed Hijazi

2020

DECLARATION

I hereby declare that this Applied Project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……

Candidate's Name:

……

Date:

……

I hereby declare that preparation and presentation of this Applied Project were supervised in accordance with the guidelines on supervision of Applied Project laid down by Ashesi University.

Supervisor's Signature:

……...

Supervisor's Name:

……...

Date:

……...

Acknowledgement

I would like to thank Jehovah God who granted me the wisdom and ability needed to make this project a reality. Next to him is my caring mother who through her hard work and admirable single parenting skills put me through the best schools. Without the management of The Ritzz Exclusive Guest House, this project would not be possible. I would like to offer special thanks to Cynthia Awuni, who although no longer with us, helped make this project a success. Cynthia was dedicated to her job, and always responded to my concerns and questions about the guest house. I want to thank my friends and family who lovingly supported me during some difficult times. I would like to thank my advisor, Dr Ayorkor Korsah for her able guidance and support in completing this project. Finally, I am thankful to all my lecturers and classmates who extended a helping hand along with encouragement throughout the year.

Abstract

Businesses are using technology to gain a competitive advantage over players in their respective industries. The Ritzz Exclusive Guest House, a hospitality business in Ghana, needed an application that would integrate into its existing processes and enable it to communicate with their Ghanaian based clients to increase customer loyalty. This project proposes a two-in-one application, Bulker. The proposed solution will enable the guest house to send text messages to its local clients and allow them to receive responses in a cost-effective and user-friendly way.

Table of Contents

Chapter 1: Introduction

Businesses are leveraging technology today more than ever to drive innovation in their respective industries [1]. As a result, businesses have become more competitive among themselves, allowing customers to have superior experiences. Marketing departments around the world are also constantly striving to keep up with technology as shifts from traditional forms of marketing to digital marketing have become more efficient, cost-effective and secure [2]. Enterprises that are utilizing technology today, no matter their size, boast stronger customer relations, efficient operations, and enhanced security. This applied project focuses on the development of marketing tools using existing technologies such as SMS to bolster customer relationship and retention at The Ritzz Exclusive Guest House.

1.1 Background

Many companies in Ghana have begun adopting technology to maintain customer loyalty and boost revenue. The Ritzz Exclusive Guest House seeks to adopt such a trend.

The Ritzz Exclusive Guest House is a hospitality business that provides personalized experiences to each of their guests. They do this by providing tourists, business travelers and locals with bed and breakfast facilities along with other amenities. Guests have access to complimentary concierge services, laundry facilities, airport shuttle service, luggage storage and an executive lounge.

In 2017, PwC reported that there are about 2,723 hotels in Ghana [3]. That number has evidently increased since then, as the tourism industry has seen steady annual growth at 5.1 percent [3]. Along with this is competition – competition among hotels to fill their rooms with guests. Located in Osu, Accra, The Ritzz Exclusive's main competitors are Urbano, Frankie's Hotel, Roots Apartments, La Villa Boutique Hotel, and Olma Colonial Suites.

The Ritzz Exclusive Guest House is listed on online travel agencies which have played a significant role in attracting foreign nationals and locals to hotels in Accra. However, these travel agencies have other hotels listed and offer each hotel business equal opportunity in receiving bookings. These travel agencies charge high commission percentages on bookings made through their sites. The guest house also uses its website and social media networks for advertising, especially to potential local customers.

During check-in at The Ritzz Exclusive Guest House, customers fill a customer requisition form. This form allows the business to collect customer details, such as their name, telephone number, email address and the number of nights they are staying. This information is then inputted manually into an excel sheet and further used for administrative purposes which includes marketing.

The Marketing and Systems Administrator of The Ritzz Exclusive Guest House has noticed a downward trend in recent times, in the retention of its local customer base. Although new and existing clients are sent marketing offers and promotional products through email, he individually reaches out by phone call to those who do not respond to offers. Although this is effective, it is often expensive and inefficient. The general impression created is that email is not an effective way to reach their local client base.

Bulk SMS involves the sending of a large number of SMS messages to the mobile phones of already known recipients [4]. With a shift toward businesses using bulk SMS services in Ghana, The Ritzz Exclusive Guest House desires to follow this innovative technology that will allow it to send personalized SMS to its new and regular local clients to build trust and maintain customer relations while still potentially increasing its revenues.

1.2 Motivation

The Marketing and Systems Administrator of The Ritzz Exclusive Guest House is primarily concerned about the high cost and inefficiency associated with keeping in touch with their local customer base. With many companies shifting toward a low-cost, practical approach by using bulk SMS, the enterprise seeks to leverage such technologies.

However, bulk SMS technologies also has its limitations. For example, customers cannot respond to a call-to-action or fill a form because a Sender ID is sent to clients. A Sender ID is a limited number of characters that does not include a phone number. Without a phone number, or designated shortcode by mobile operators, replies cannot be sent by end-users.

Sender ID's increase the credibility of an incoming text. However, replying to a call to action or filling out a form is essential as it is a way of evaluating the effectiveness of a marketing campaign [5].

Alternatively, businesses may choose to buy a shortcode from mobile operators. A shortcode is typically a four-digit number that is used to send out messages. Businesses can receive replies from these messages using a simple call-to-action, i.e. a 'yes' or a 'no' answer. However, this approach is expensive and limited. The business has to buy the same shortcode from each mobile operator, costing thousands of cedis, and contact forms and surveys cannot be evaluated.

The Ritzz Exclusive Guest House is looking for a robust, low-cost business solution to not only allow it to send out personalized marketing campaigns but also to receive responses to those campaigns to enhance its customer relations. The system must enable it to engage its local client base to increase customer loyalty and its revenues in the competitive hospitality industry, while still maintaining their marketing budgets and existing administrative processes.

A custom-built solution that meets the requirements of The Ritzz Exclusive Guest House will be built using creativity and design thinking, to enable it to send out marketing campaigns and surveys.

1.3 Project's Focus

The focus of this project is to design a solution that enables the guest house to send out personalized marketing campaigns to its new and existing customers and receive responses to maintain customer relationships and increase revenue.

1.4 Related Work

The American University in Cairo conducted a study that explored the usefulness of short message service (SMS) and its effectiveness. The study consisted of a series of qualitative interviews and surveys. It was observed that targeting and personalization ensure that SMS advertisements are useful and relevant to recipients [6]. Further, SMS advertisements that include the brand name as the sender increase value and trust. If recipients are already connected with the brand, advertising is also more effective.

A study was conducted in the Bundelkhand University Library in India to assess the effectiveness of using SMS-based mobile alerts. The Library uses SMS mobile alerts for marketing its services and in providing value-added services [7]. The study found that library users were successfully motivated and engaged in the usage of the library's resources through SMS messaging. The findings further suggested that a more robust system can be implemented to take advantage of such marketing potential.

A study was conducted in various universities located in South Korea to examine key drivers of the acceptance of SMS advertising. Based on a series of interviews, the study indicated that for SMS advertising to be effective, the trust needed to be established with advertising

companies and its recipients [8]. Companies that abused user data and unreasonably sent out text messages were ineffective at making an impact.

1.5 Project Objectives and Significance

The main objective of this project is to develop a solution for The Ritzz Exclusive Guest House, named **Bulker**. Bulker will allow The Ritzz Exclusive Guest House to send out tailored marketing campaigns via SMS to its local client base. The system will allow responses to be received from clients using a low-cost approach. The systems administrator can further modify this data and download it for further administrative work.

The Ritzz Exclusive Guest House will benefit considerably from this project as it will allow the firm to gain a higher competitive advantage over its competitors in the hospitality industry by being the first to adopt this new marketing technique. This will enable it to drive up its customer retention and revenues. The guest house will also be able to reach out to marketing companies and use data for further analytical purposes.

Chapter 2: User Requirement Specifications

This chapter discusses the procedures used in the creation of **Bulker**; specifically, how user requirements were gathered and how users of Bulker will interact with the system. It will also outline the functional and non-functional requirements of the system.

2.1 Requirements Elicitation Technique & Procedure

Open-ended questions were used in understanding the concerns of management, the businesses current workflow and to clarify ambiguity. Closed-ended questions were used to identify requirements and solicit opinions and ideas from the system's administrator of the guest house.

After the fundamental prerequisites were obtained via interviews, sketched prototypes and user-flows were presented to the systems administrator of the guest house for review. Suggestions were provided by management, and changes were made. The modifications were presented to management for final approval. The overall security of the system was also further reviewed.

The creation of a system which includes two individual applications will be created based on the requirements of the guest house.

2.2 Functional Requirements

Functional requirements are the product features that will be developed to enable users to accomplish tasks. This section outlines the functional requirements of the data management platform and the bulk SMS program.

2.2.1 Bulk SMS Program

The bulk SMS program is mainly responsible for the administration of bulk SMS to clients of The Ritzz Exclusive Guest House by the systems administrator. The bulk SMS program has only one user.

Systems Administrator

The systems administrator is responsible for the administration of bulk SMS sent out on behalf of The Ritzz Exclusive Guest House.

- The user must be able to select a CSV file of customer records.
- The user must be able to compose a message or select a message template.
- The user must be able to send out an SMS with a unique URL to each customer.
- The user must be able to see a sample of the text message sent.
- The user must be able to bulk-send text messages.

2.2.2 Data Management Platform

The data management platform will receive and manage responses from clients of The Ritzz Exclusive Guest House. Clients that receive text messages that include a URL will be able to record their responses using this system. The systems administrator uses this system to manage customer responses.

Systems Administrator

The systems administrator is responsible for the management of incoming replies from clients of the guest house.

- The user must be able to sign into the application with authorized credentials.
- The user must be able to upload a CSV file of customer records.

- The user must be able to view all customer records.
- The user must be able to view responses recorded by customers.
- The user must be able to edit customer responses.
- The user must be able to delete customer records.
- The user must be able to view comprehensive graphs and charts for review.
- The user must be able to download a CSV file of customer records.

Customer

The customer is a client of the guest house. The customer may be a first-time client or a repeat client.

- The user must be able to respond to an SMS via a URL which contains a survey, marketing campaign or a call to action.

2.3 Non-functional Requirements

Non-functional requirements are requirements that are not directly concerned with the specific services delivered by the system to its users [9].

Efficiency – The system should be able to send personalized SMS in bulk and run campaigns with minimal effort.

Performance – The desktop and web applications must respond quickly to user input both on the user side and on the server-side. This requires the use of efficient and appropriate data structures and algorithms.

Usability – The application must follow clear design principles that will allow users to use it intuitively.

Security – The application must securely store data, retrieve data, and prevent unauthorized access.

2.3.1 Product Requirements

- The web application shall be available to all users always, 24 hours a day. Users must always be able to access all features.
- Downtime should not exceed five seconds in any day.

2.3.2 Product Standards

Code deployed should be well modularized for future development. Comments should exist throughout the code under all functions and classes. These should explain in detail the actions they perform.

2.4 System Requirements

System requirements describe the features and behavior of a system. It mainly deals with critical performance parameters that the system must meet.

Bulk SMS Application

- The system should be able to process a CSV file.
- The system should be able to append a unique URL to each message dynamically.
- The system should be able to process and send SMS in bulk.

Data Management Application

- The system should be able to read a CSV file and store its data in an online database.
- The system should be able to read, write or modify the online database.
- The system should be able to display graphs and charts of user responses.
- The system should be able to read and write to a CSV file.

2.5 Use Case Diagram

A use case diagram is a simple representation of a user's interaction with the system.

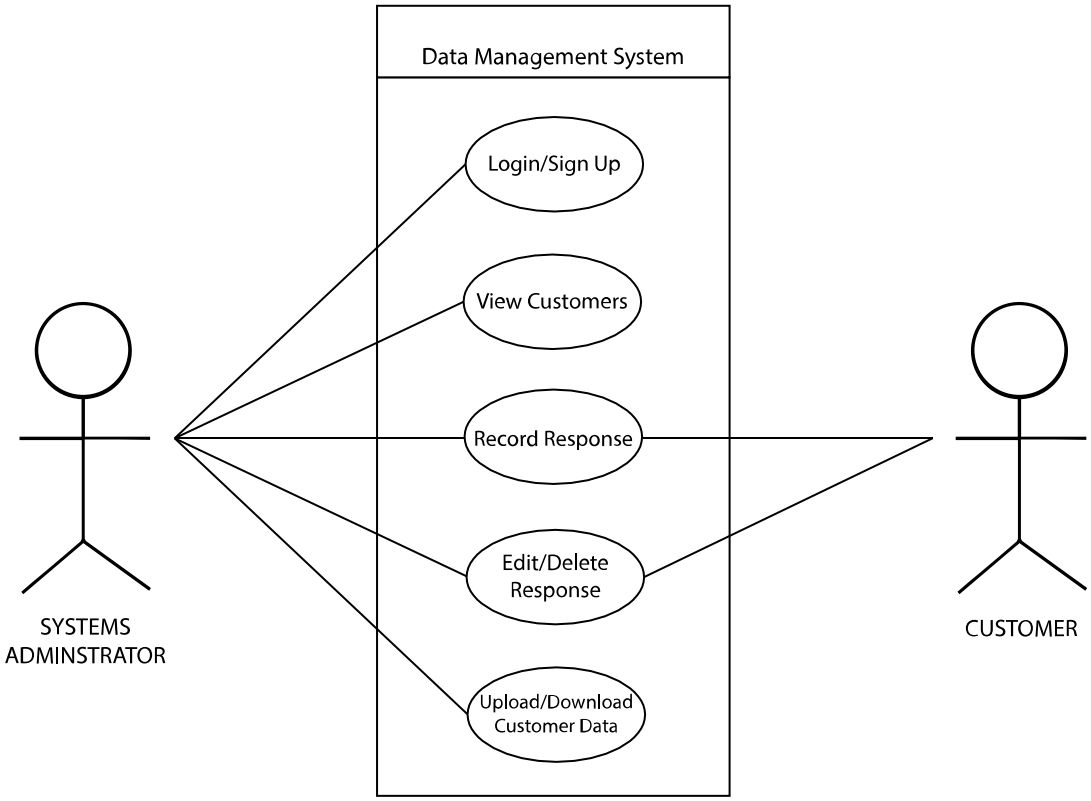

***Figure 2.1**: Use Case Diagram of the data management system*

Figure 2.1 shows two users of the data management system. The systems administrator can login/sign up, view customers, record responses, edit or delete responses and upload or download customer data. The customer is only able to record a response or delete/edit a response.

2.6 Design Decisions

Following the functional and non-functional requirements outlined above, including design principles and upon consultation with the administrator of The Ritzz Exclusive Guest House, two separate applications that complement each other will be created. The bulk SMS program will be a desktop-based application responsible for sending out personalized SMS in bulk to chosen clients

by the user via a CSV file. The type of message being sent can be customized and a campaign can be run if the user wishes. If a campaign is running, responses will be collected using the web-based application and displayed appropriately to the administrator of the guest house after signing in. The user-flow is narrated below:

- Administrator generates CSV file from a spreadsheet application. The guest house already collects data using a spreadsheet application. A CSV file format was chosen because of simplicity and its widespread use in data handling.
- Administrator uploads CSV file to Bulk SMS application – Application pre-processes the file by generating a unique encryption key for each customer record. The encryption key will be used to identify each client. It will also prevent hackers from easily identifying a client and altering data.
- A user types a promotional message or runs a campaign or survey. If a campaign or survey is run, a short URL is sent with the key attached in the SMS.
- The administrator sends out SMS in bulk to clients.

If a campaign or survey is run:

- Administrator uploads CSV file to the data management application.
- Users respond to campaign or survey via short URL.
- The administrator receives responses from clients.
- The administrator can edit responses from clients or delete users and their responses.
- The administrator can download data as CSV file for further processing.

Chapter 3: Architecture and Design

3.1 System Architecture

Architectural design is the creative process of designing an organized system that will satisfy the proposed functional and non-functional requirements [9]. The architecture pattern chosen for the data management platform is the client-server architecture. The architecture pattern chosen for the bulk SMS program is the Model View Controller.

The **bulk SMS program** will send out text messages to local clients of The Ritzz Exclusive Guest House. The program has one main component:

A user side: The systems administrator will use this to upload user data and send SMS to all clients. The program will then interact with the Bulk SMS API to send out messages securely.

The architecture of the system is as follows:

- View: The view is the graphical interface the user sees and uses.
- Model: The model stores the core logic and algorithms.
- Controller: The controller manages data flow to the view and the model.

The **data management platform** will receive and process responses from text messages. This program has three main components:

1. A user side: The system administrator will use this to securely upload user data, make changes to user responses, and download user data. Clients will use this to input data.
2. A server side: The server will act as a bridge between the user and the database by storing the code which will process incoming and outgoing data. This will be done using PHP. PHP is a scripting language used in web-development by HTML embedding [10].

3. A database server: The database will store all data that is used within the system. This will be accessed via PHP code and stored in MySQL. MySQL is a data management system by Oracle [11].

The architecture of the system is as follows:

- Client: The user will interact with the interface to view, edit, or download customer responses.
- Application Server: Read and write a CSV file and pre-process data that will be stored in the online database.
- Network: Internet services to facilitate communication between client and server.
- Database: Stores client data.

3.1.1 Bulk SMS Program

View

This is the user interface of the desktop application that the administrator or user of The Ritzz Exclusive Guest House accesses to send bulk text messages. The application will send information to the controller that will then further communicate with the model if necessary.

Controller

The controller handles a request from the view and passes it on to the model. When data is processed and ready, the controller passes it on to the view. For example, if the user sends bulk SMS to clients, the controller passes it on to the model to run the appropriate code. The model then informs the controller if the messages have been sent or not, and this is passed onto the view. The model further produces an output file outlining the outcome of the API call.

Model

The model keeps the core code of the program and runs it appropriately per the controller's request. For example, if the user selects a CSV file, the controller informs the model to run the encryption generator which will read the CSV file and appropriately assign unique keys to each customer. If the customer sends SMS to clients, the model will run code which will trigger the API to send out messages.

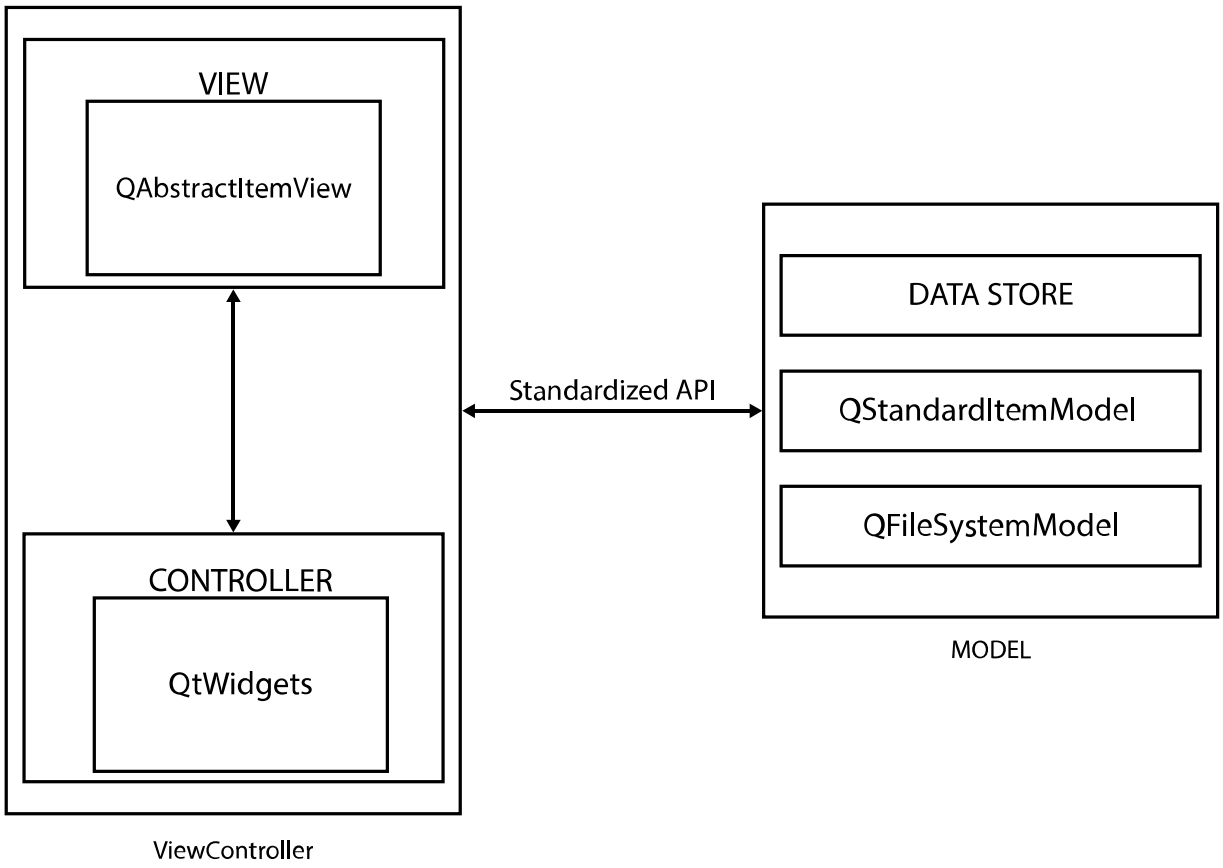

***Figure 3.1**: The Model View Controller of the bulk SMS application*

3.1.2 Data Management Application

Client Side

The data management application would be viewed in a web browser. The client side of the application has two client views. The end-users view, and the back-end view. The end-users view is the interface the clients would see and respond to when they get sent a campaign by the bulk SMS desktop application. The back-end view would show the administrator the responses to the campaign sent out.

Application Server

The application server controls the implementation of application-specific logic and information storage and retrieval requests [9]. In this case, the application server will handle PHP script requests made to and from the database. The application server will also hold the core code of the web application.

Database Server

The database server manages transactions between the application server per requests made on the client side. It also enforces database rules and displays data to appropriate users. For example, if the administrator loads the home page of the data management application, the server ensures that appropriate records in the database are displayed to the user.

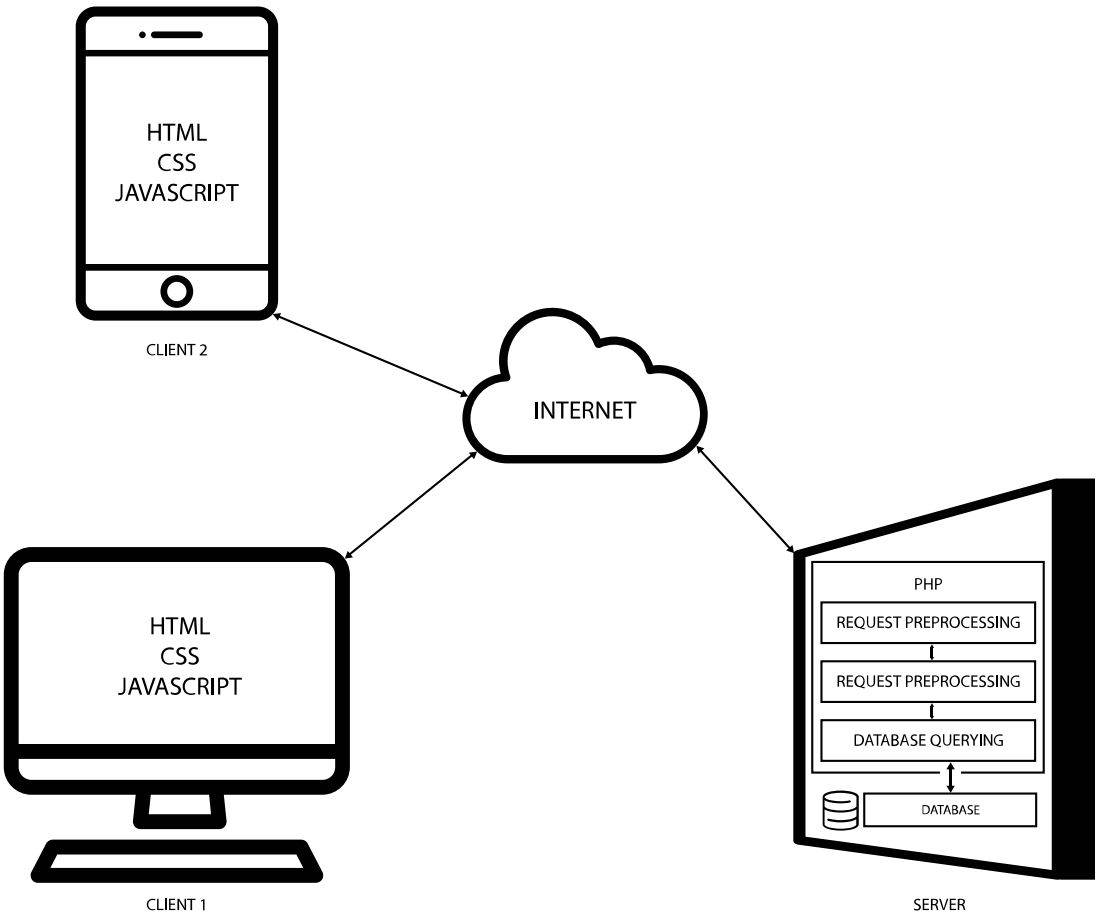

***Figure 3.2**: The Client-Server Architecture of the data management platform*

3.2 Sequence Diagram

A sequence diagram shows object interactions that are arranged in time sequence. It details operations that are carried out by a user.

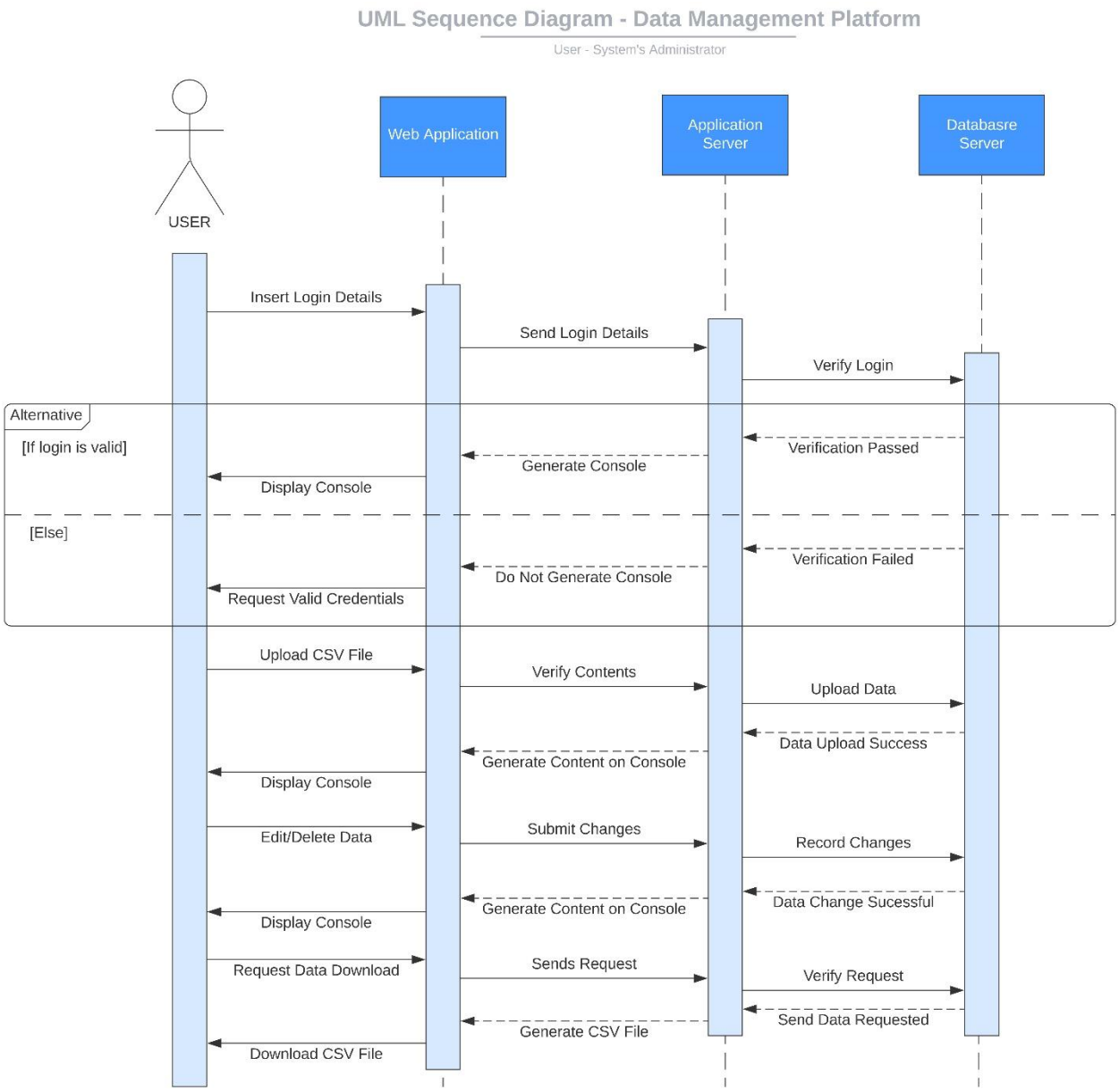

***Figure 3.3**: UML Sequence Diagram of the data management platform*

Figure 3.3 shows the interaction between the systems administrator (the user), and the data management platform. The user begins his interaction by logging in or creating an account. After login/registration, the user can perform administrative functions such as uploading a CSV file, editing/deleting data, and downloading customer data.

3.3 Database Schema

A database schema is a structure that represents the logical view of the entire database.

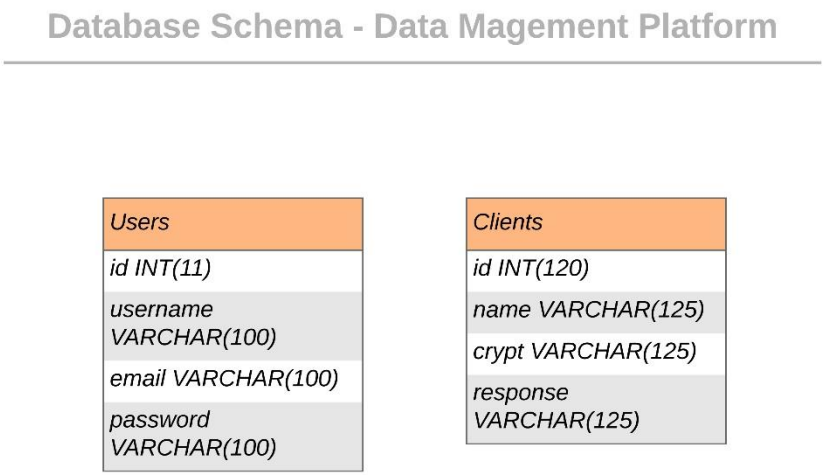

***Figure 3.4**: Database Schema for the data management platform*

Figure 3.4 shows the database schema for the data management platform. Two tables are created and unrelated. The Users table handles the storage of the system administrator's login details and registration.

Chapter 4: Implementation

In this section, the list of technology services used to build both applications will be discussed. The frameworks, programming languages and tools that were used in the development of Bulker will be examined. Design acquired from requirements analysis informed the choice of technology and implementation techniques. An object-oriented approach was used in the development of the bulk SMS program. This was necessary to facilitate long-term development and changes as well as interactions with the framework used; PyQt. A functional approach was used in the development of the data management application as its implementation was straightforward.

4.1 Project Feasibility

The availability of free and open-source software has significantly driven down the cost of software development in recent times [12]. In this project, most of the resources used are free and open source.

The desktop application will be developed using Python programming language. “Python is an interpreted, object-oriented, high-level programming language with dynamic semantics” that is open source [13]. For the interface of the application, PyQt will be used. PyQt is a Python plugin and framework that is free and used in creating cross-platform Graphical User Interfaces (GUI). In this project, it will be used to create the interface of the desktop application. SMS will be sent to clients of The Ritzz Exclusive using an API that will incur charges and this will be addressed during the development of the application and with the systems administrator of the guest house.

The web application will be written using HTML, CSS, JavaScript and PHP. Bootstrap, an open-source toolkit, will be used as a CSS framework for the program’s interface during development. This application will be hosted on the already existing and paid servers of The Ritzz Exclusive Guest House and managed by the systems administrator.

4.2 Implementation Techniques and Evidence

Bulk SMS Application

In the development of the bulk SMS application, a step by step approach was used. The application was broken down into various modules and developed individually. The various functions of the application were developed first and individually tested. The user interface of the application was then created using PyQt's Designer. Designer is a tool from PyQt which allows for the quick development of Graphical User Interfaces, giving simple drag-and-drop functionality. The modules developed were then merged with the user interface. See Appendix A.1 for key modules.

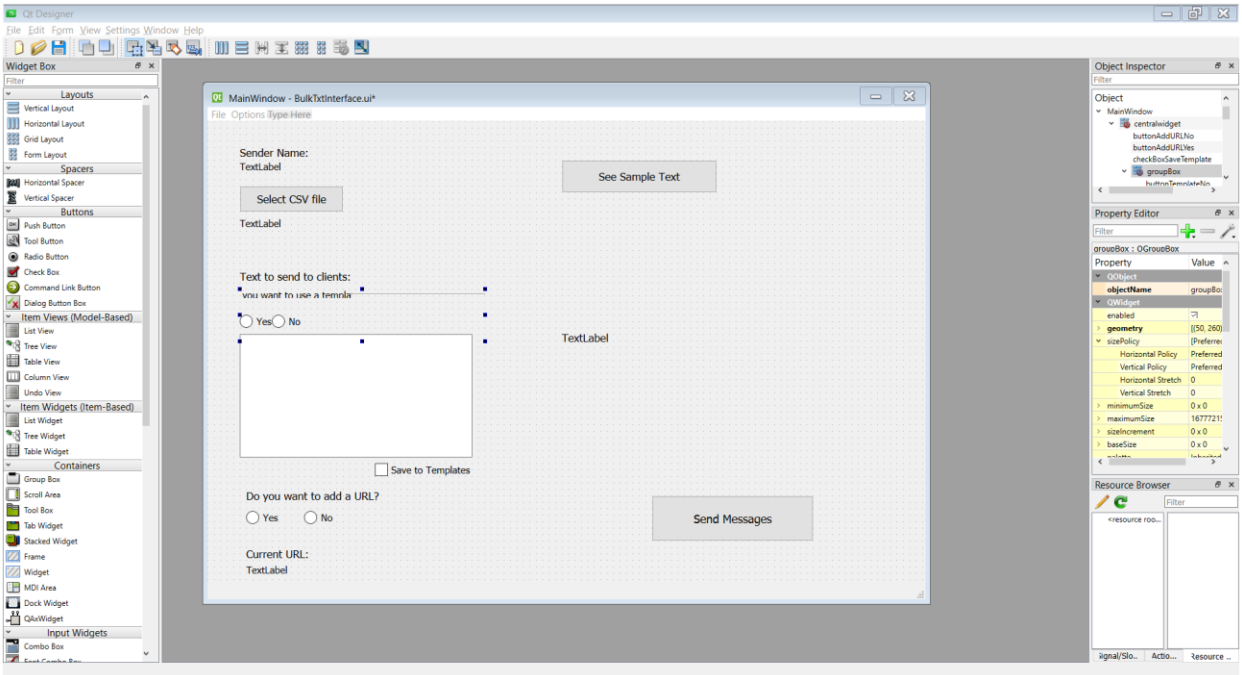

***Figure 4.1**: PyQt's Designer interface*

```
C:\Users\Mohamed Hijazi\OneDrive - Ashesi University\Desktop\School>pyuic -x BulkTxtInterface.ui -o BulkTxtInterface.py
```

***Figure 4.2**: Conversion of interface file (.ui) to Python file (.py)*

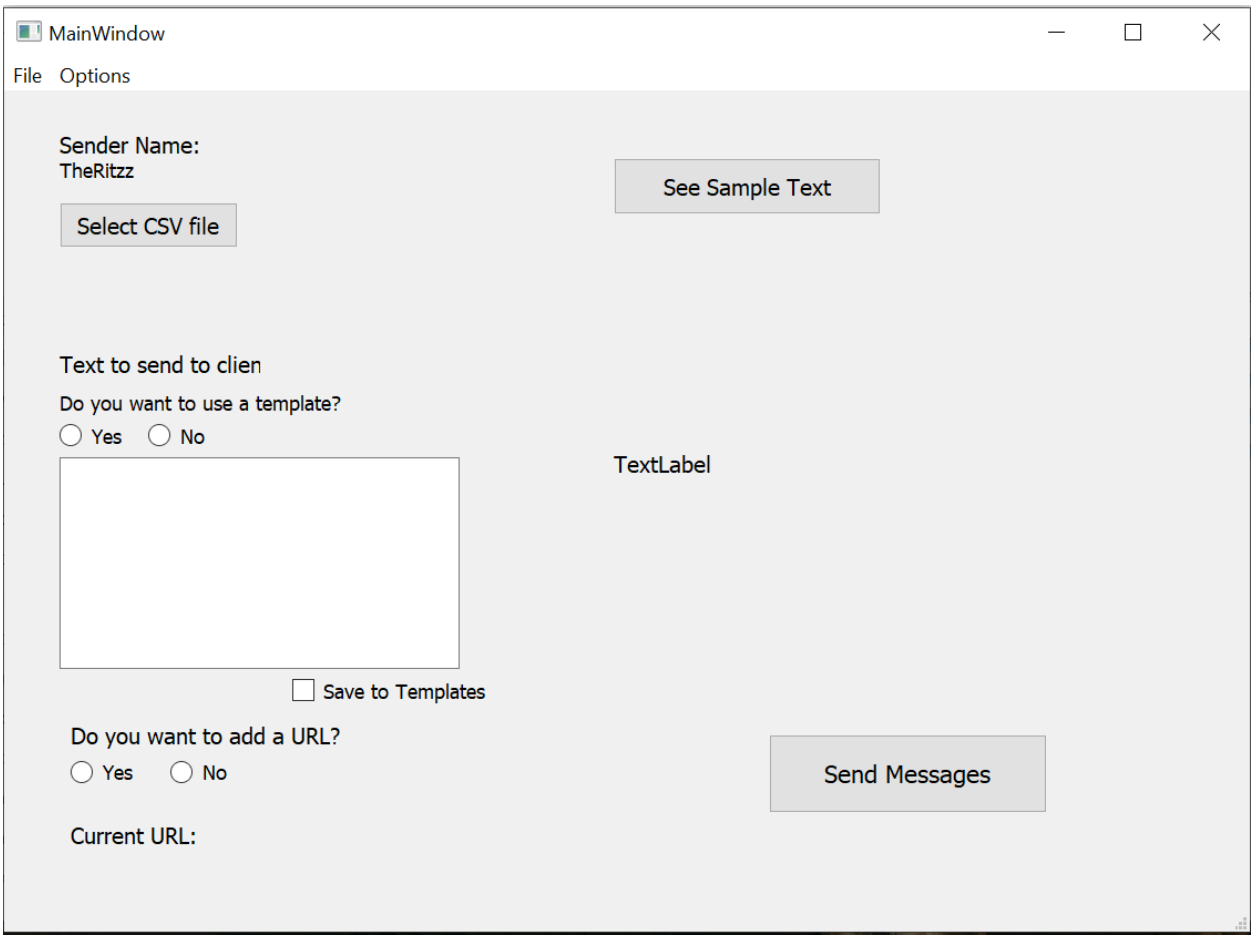

***Figure 4.3**: bulk SMS application interface*

Figure 4.3 shows the main interface of the bulk SMS application. A CSV file is selected, a template is chosen, or a message is composed. A URL is added if the administrator wants to run a campaign, and a sample text is previewed before the messages are sent.

Data Management Platform

During the development of the data management platform, the key functionality of the system was broken down and developed individually. The database was created first using phpMyAdmin. Core Database operations CRUD (Create, Read, Update, and Delete), were then developed individually.

The interface that supports these functions was then developed and merged. The login and sign-up page were then added to the entire program. See Appendix A.2 for modules.

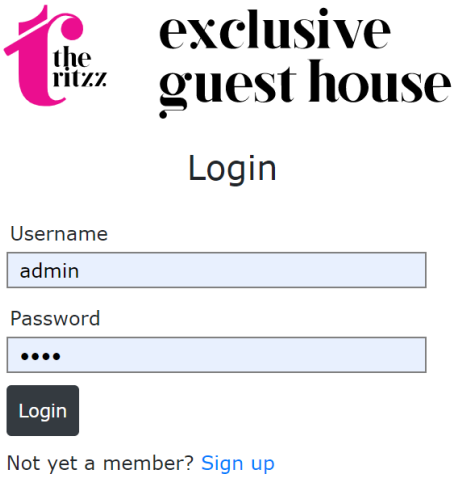

***Figure 4.4**: Login page of the data management application*

Figure 4.4 shows the login page of the data management application. The systems administrator of the guest house accesses this portal to gain access to customer data and responses.

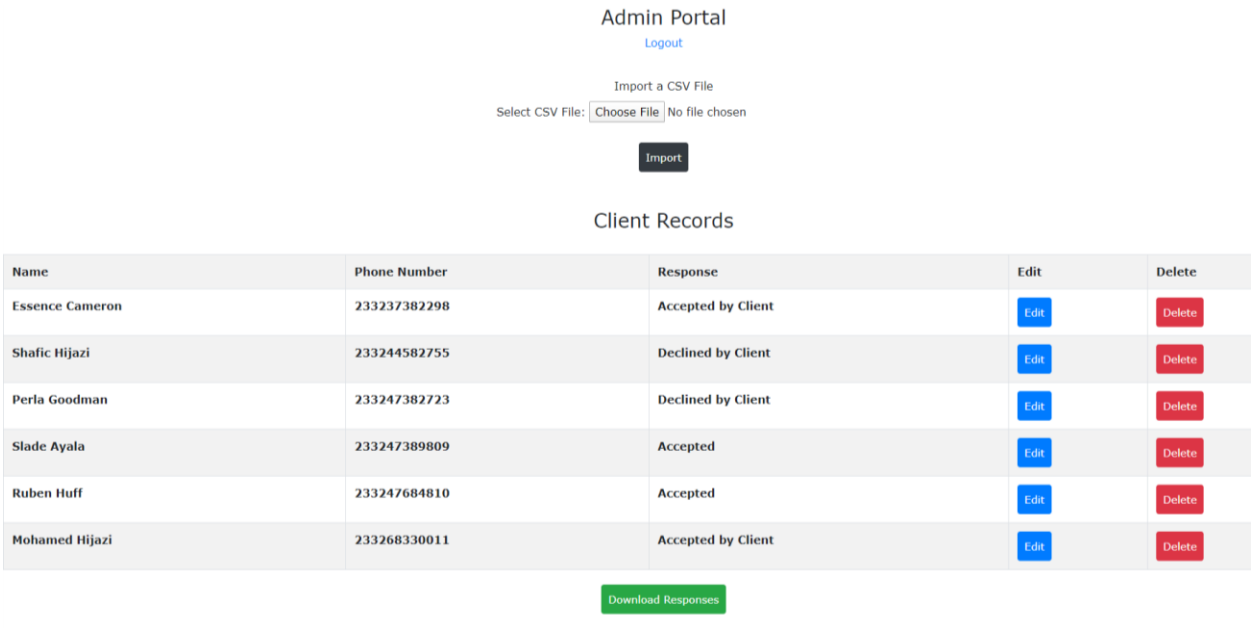

***Figure 4.5**: Administrators portal*

Figure 4.5 shows the index page of the application after login. The administrator can upload the CSV file used to run a campaign and receive responses from clients. Changes can be made to the responses, and all responses can be downloaded for further administrative work.

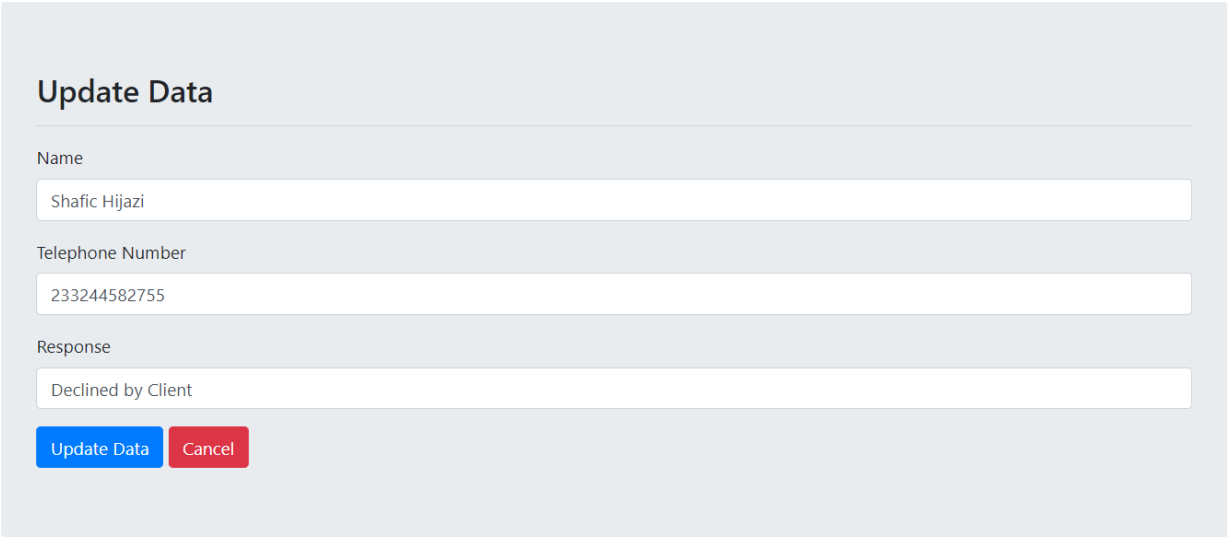

***Figure 4.6**: Updating a response*

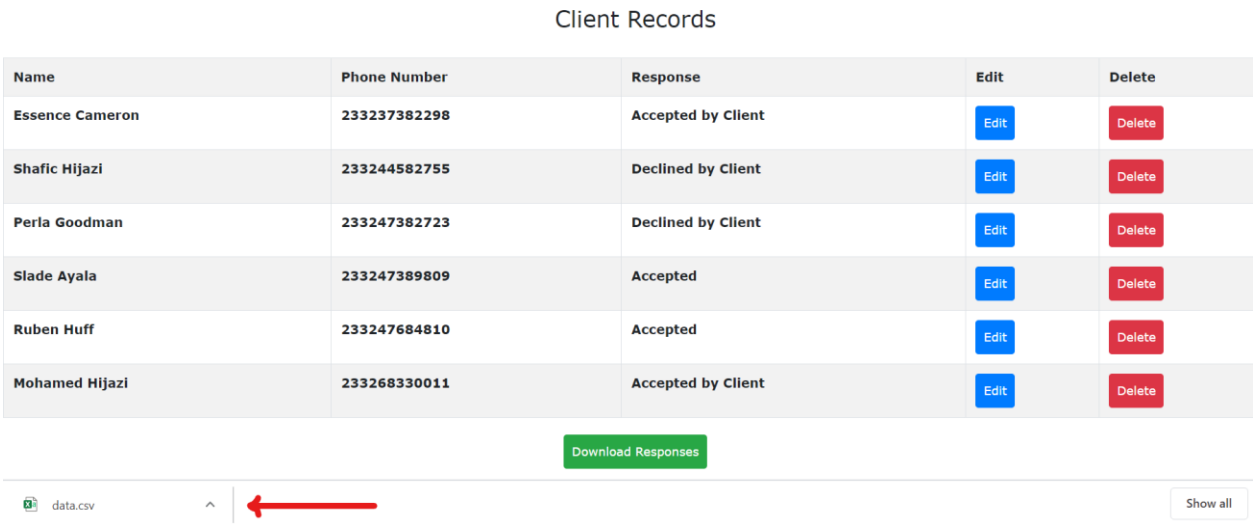

***Figure 4.7**: Successful download of responses*

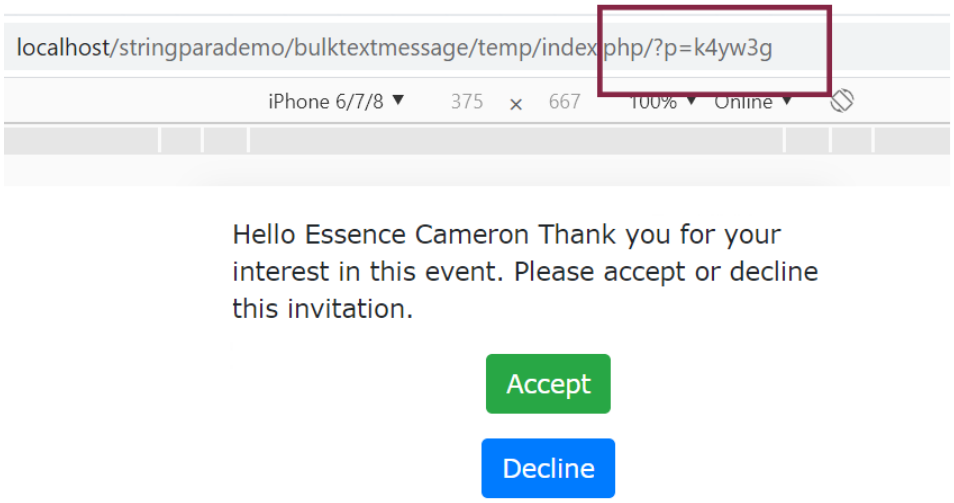

***Figure 4.8**: Client side – accepting or declining a response*

Figure 4.8 shows a URL that has been opened by a client. The encryption key highlighted in red is used to identify the client. His name is fetched from the database for personalization. His response is recorded safely in the database using that same key when the 'accept' or 'decline' button is clicked. The client-side screen can be adapted. For example, a survey or contact form can be sent to clients instead of a call to action as depicted in Figure 4.8.

4.3 Implementation Issues

4.3.1 Reuse

Component reuse has become an important aspect of software development. It is necessary to reuse existing code as much as possible because of costs and schedule pressure [9]. During the implementation of this project, component reuse was possible at several levels.

The Abstraction Level

During the development of both applications, knowledge, successful abstractions, design patterns and architectural patterns were obtained by passively viewing YouTube videos. Ideas were obtained by reviewing developer documentation by the API providers and viewing online implementations of core database operations; CRUD (Create, Read, Update, and Delete) in PHP.

The Object Level

At this level, objects from an existing library was reused during implementation. This was necessary in the case of the bulk SMS application. The following libraries were used:

- csv – This library supports the reading and writing of a CSV file.
- urllib – This is a library that handles HTTP URL's in Python.
- random – A necessary library to append a unique identifier of each record for SMS campaigns.
- string – Common string operations module for the generation of the unique identifier as described above.
- PyQt5 – Library for the PyQt cross-platform framework necessary for the desktop interface.

The Component Level

At this level, a framework is reused in the development of the system. The Bootstrap framework was used for the data management platform. A log in and sign-up page was adapted from CodeWithAwa [14] for web-based application. The bulk SMS program interface was developed using PyQt, a cross-platform application framework.

The System Level

At the system level, entire application systems that already exist are used. In the development of both applications, there was no adoption of an already existing application system.

4.3.2 Configuration Management

This involves the general process of managing a changing software system. During this project, version changes were tracked using Git. Git is a version control system that allows authors to track changes made to projects under development. By using Git, any unforeseeable event that breaks software under development can easily be reverted to an earlier version. Version releases were pushed to GitHub, a cloud-based service that stores code privately or publicly.

4.3.3 Host-Target Development/Implementation Method

Software was developed using a local web server solution stack known as XAMPP. The development environment also supports PHP, making it ideal to test the data management application locally before uploading it to the online servers of the guest house. The local web server also played a significant role in the agile development of the application. The execution platform which will be on a Linux server, will allow for a smooth transition once development is completed.

4.4 Technology Stack

4.4.1 HTML

HyperText Markup Language (HTML) is used in the data management application. HTML is a universally understood language that allows web browsers to generate an interface for the application. The language can generate online web pages that have headings, tables, text, lists, photos, forms, spread-sheets, video clips and sound clips [15].

4.4.2 CSS

Cascading Style Sheets (CSS) largely relieve HTML of the responsibilities of presentation and gives authors control over how web pages are documented or styled [15]. The interface of the data management application was designed with design principles that will make the system intuitive, user friendly, and appealing to the eyes. Bootstrap is used as a CSS framework for the program's interface during development.

4.4.3 JavaScript

JavaScript is a client-side scripting language that allows authors to implement complex features on web pages. Authors can use the language to create content that is updated dynamically, control media, animate images, etc [16].

4.4.4 PHP

Hypertext Preprocessor (PHP) is a server-side scripting language that is open source and suited for web development. Since PHP code is executed on the server-side and then generates HTML which is sent to the client-side, PHP allows for the secure handling and storing of user data in MySQL databases. The data management application uses PHP extensively to store user responses in a database, edit user responses, upload user data, and download user data.

4.4.5 phpMyAdmin

phpMyAdmin is a tool that is written in PHP that allows authors to handle and use MySQL databases over the web [10]. The data management application uses this tool to store, retrieve and edit data stored in MySQL securely.

4.4.6 Python

“Python is an interpreted, object-oriented, high-level programming language with dynamic semantics” [13]. The core code of the bulk SMS application is written in Python and executed by the user via the application interface.

4.4.7 PyQt

The desktop interface of the Bulk SMS Application is developed using PyQt. This plugin and framework are free and allows authors to create cross-platform Graphical User Interfaces (GUI).

4.4.8 Darrel Technologies

Darrel Technologies provides web hosting and cloud infrastructure, productivity and messaging solutions to businesses in Ghana. In this project, their SMS API is used to send out messages to users.

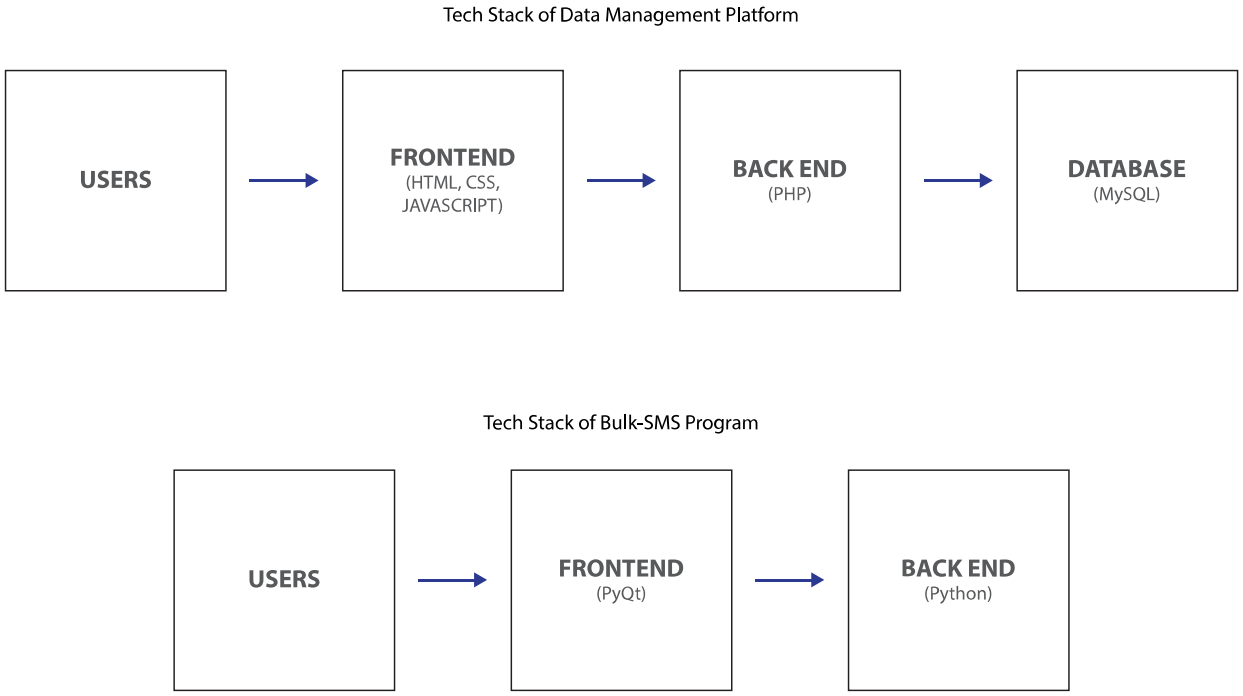

***Figure 4.9**: Technology stack of the data management platform and bulk SMS program*

Chapter 5: Testing and Results

Testing is required to know if software developed demonstrates to the developer and the customer that it meets its requirements [9]. It also allows the author to find behavior in the software that is incorrect or undesirable. For both applications, the test methods used are unit testing, component testing and system testing.

5.1 Unit Testing

Unit testing involves the testing of individual components to make certain that they work as intended [9]. This was made possible during the development of both applications.

5.1.1 Data Management Platform

Upload a CSV

The database was emptied and data from the CSV file was uploaded. The upload was successful. See Figure 5.1.

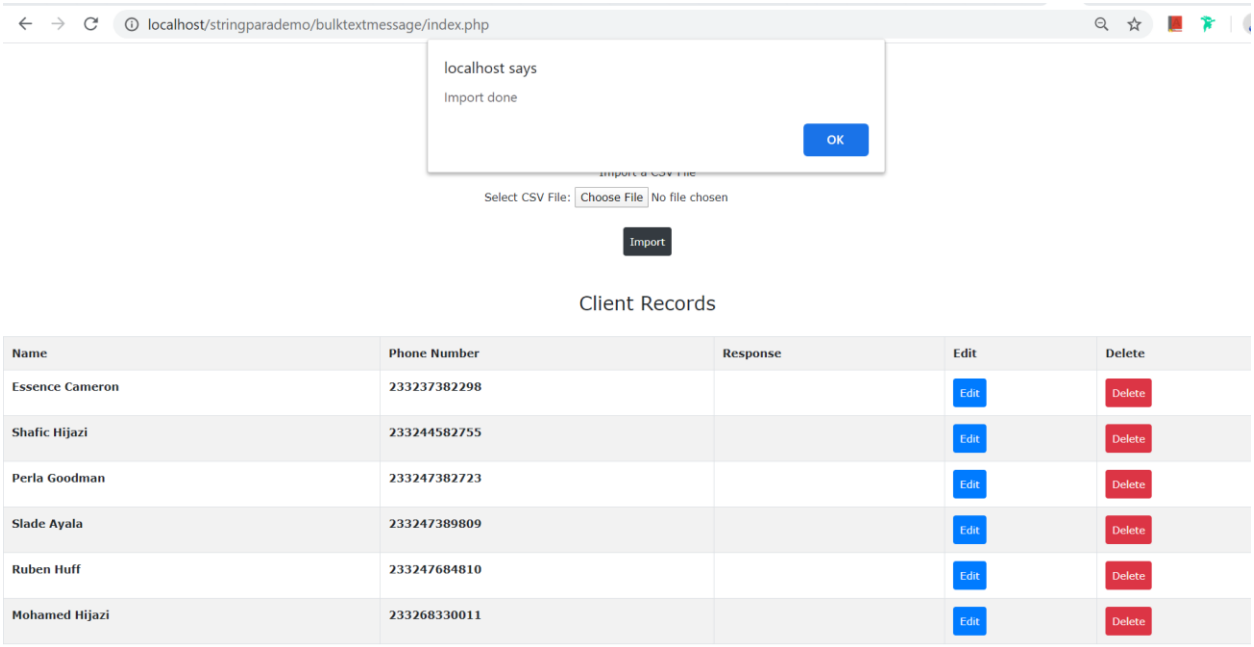

***Figure 5.1**: Database successfully uploaded from CSV file*

Update a Record

A record without a response was edited from the administrator view and the client view. Compare Figure 5.1 with Figure 5.2 and 5.3.

Name	Phone Number	Response	Edit	Delete
Essence Cameron	233237382298	Accepted	Edit	Delete
Shafic Hijazi	233244582755		Edit	Delete
Perla Goodman	233247382723		Edit	Delete
Slade Ayala	233247389809		Edit	Delete
Ruben Huff	233247684810		Edit	Delete
Mohamed Hijazi	233268330011		Edit	Delete

***Figure 5.2**: Response from Essence Cameron successfully edited from the admin panel*

Name	Phone Number	Response	Edit	Delete
Essence Cameron	233237382298	Accepted	Edit	Delete
Shafic Hijazi	233244582755		Edit	Delete
Perla Goodman	233247382723		Edit	Delete
Slade Ayala	233247389809	Accepted by Client	Edit	Delete
Ruben Huff	233247684810		Edit	Delete
Mohamed Hijazi	233268330011		Edit	Delete

***Figure 5.3**: Response from Slade Ayala successfully edited from client's interface*

Delete a Record

A customer record is deleted. Compare Figure 5.3 with Figure 5.4.

Name	Phone Number	Response	Edit	Delete
Essence Cameron	233237382298	Accepted	Edit	Delete
Shafic Hijazi	233244582755		Edit	Delete
Slade Ayala	233247389809	Accepted by Client	Edit	Delete
Ruben Huff	233247684810		Edit	Delete
Mohamed Hijazi	233268330011		Edit	Delete

***Figure 5.4**: A record successfully deleted*

Download as CSV

In Figure 5.5, the ‘download as CSV’ button was tested.

***Figure 5.5**: Successful download at bottom left corner*

5.1.2 Bulk SMS

Read CSV file

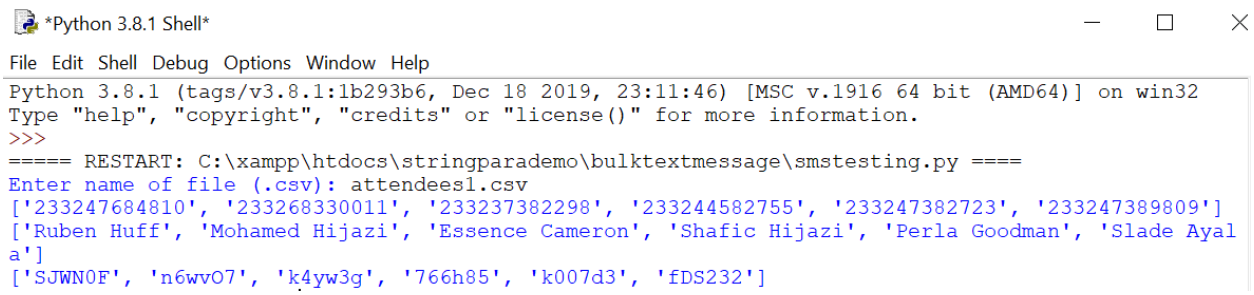

***Figure 5.6**: CSV file successfully read, and encryption keys appended*

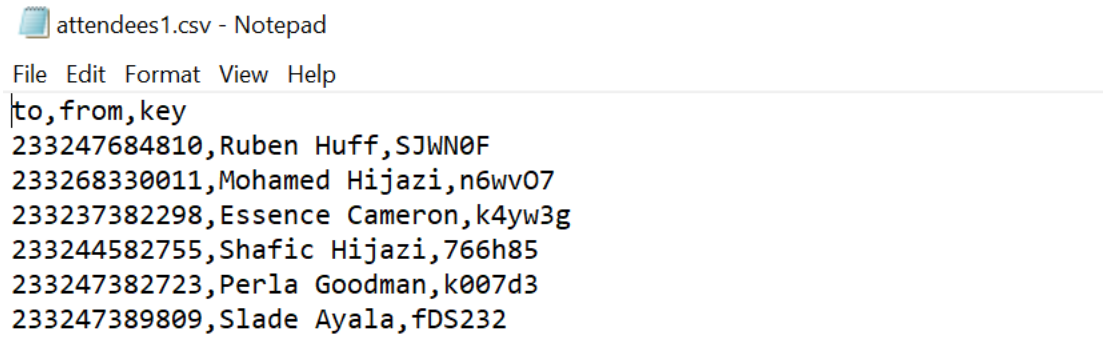

***Figure 5.7**: Encryption key successfully appended to last column*

Sending SMS

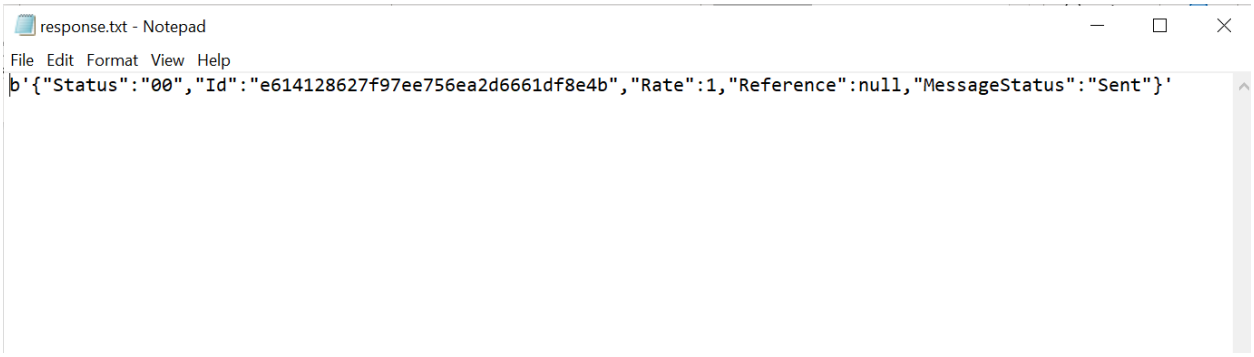

***Figure 5.8**: An output file is generated after a message is successfully sent*

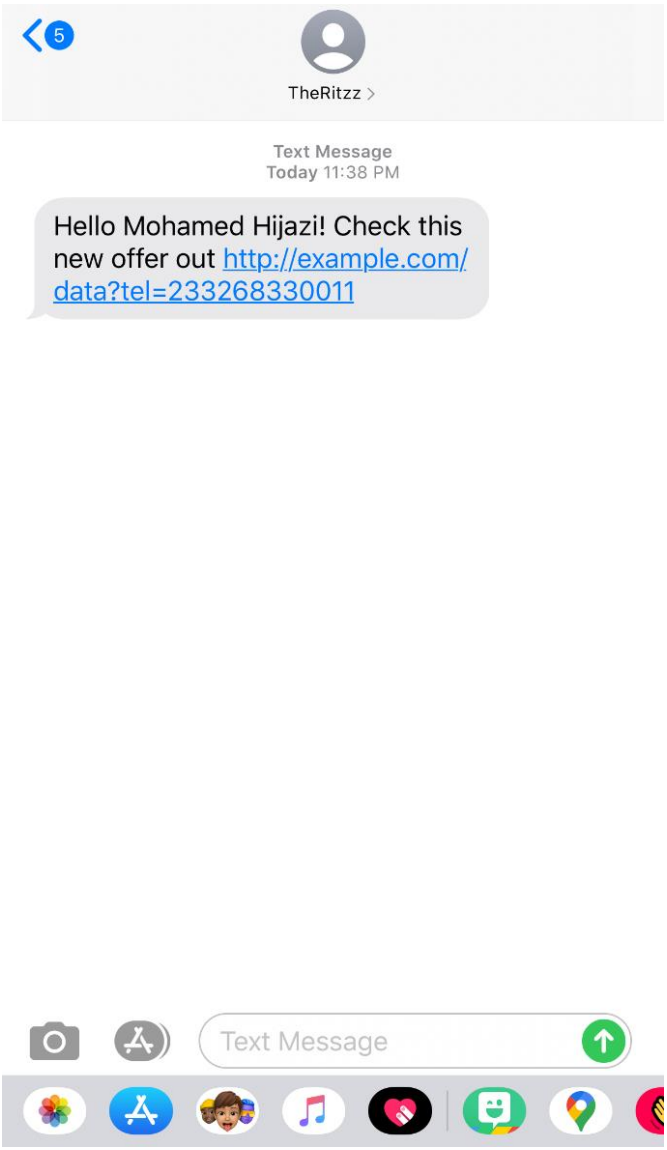

***Figure 5.9**: Message successfully sent. It includes the name of the customer*

Figure 5.9 shows a message successfully sent which includes the customer name. During later development, the encryption key was appended to the URL instead of the client's phone number for enhanced security as shown in Figure 5.10.

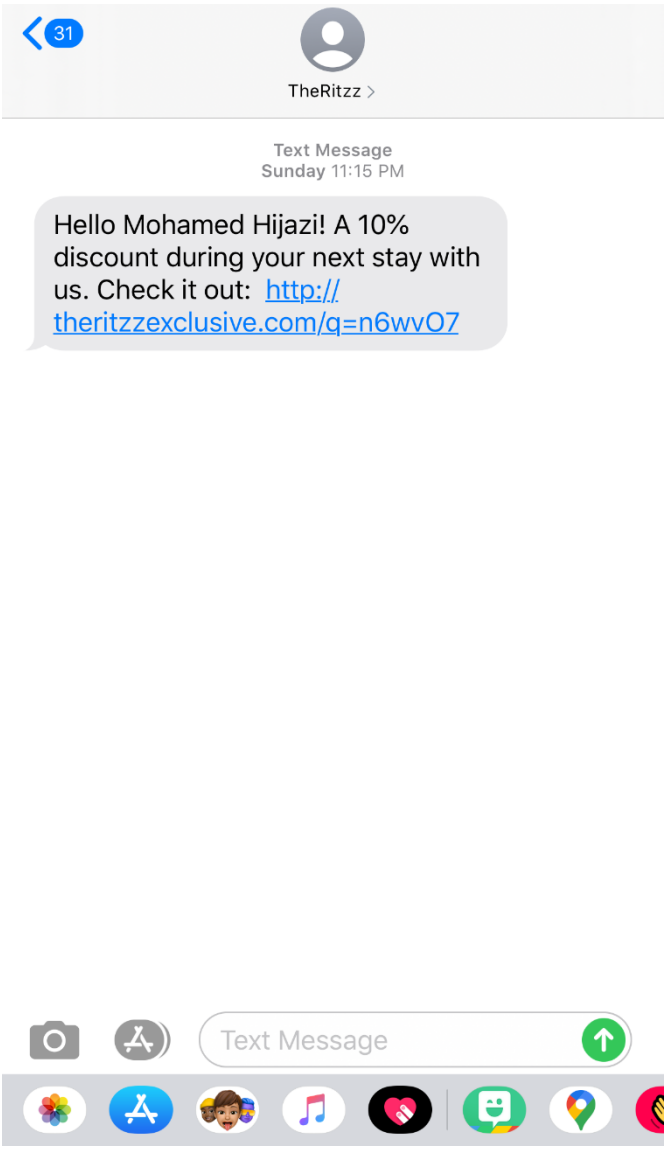

***Figure 5.10**: Message successfully sent. Includes name of customer and unique key.*

5.2 Component Testing

Component testing involves testing multiple components that are made up of several interacting objects [9].

5.2.1 Data Management Platform

Component testing was carried out where data or function references is passed from one interface to another.

Editing a User Response

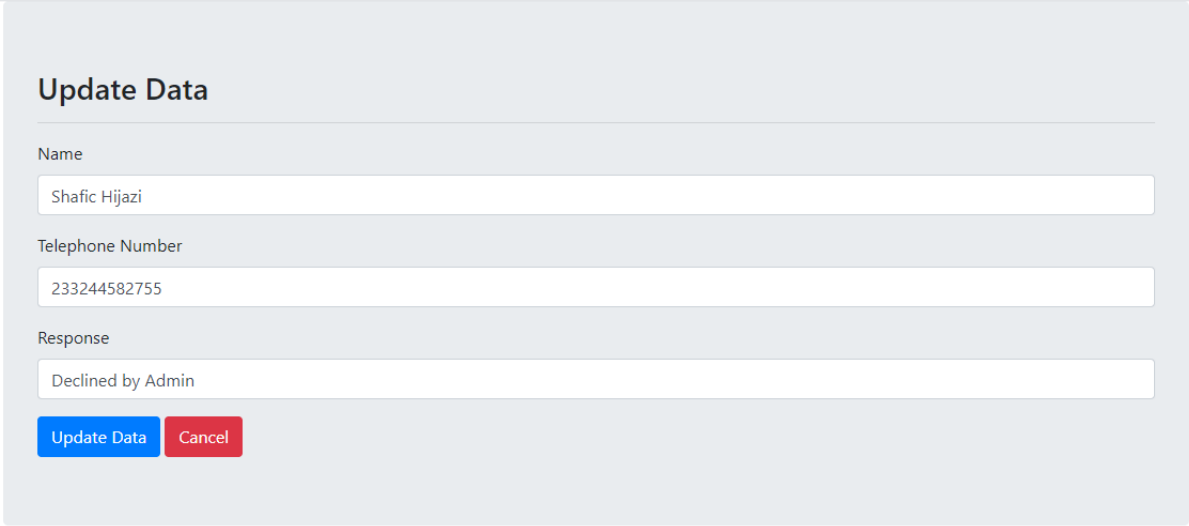

***Figure 5.11**: Information successfully carried from main view to update page*

Name	Phone Number	Response	Edit	Delete
Essence Cameron	233237382298	Accepted	Edit	Delete
Shafic Hijazi	233244582755	Declined by Admin	Edit	Delete
Slade Ayala	233247389809	Accepted by Client	Edit	Delete
Ruben Huff	233247684810		Edit	Delete
Mohamed Hijazi	233268330011		Edit	Delete

***Figure 5.12**: Information successfully updated from update page*

URL and User Interface

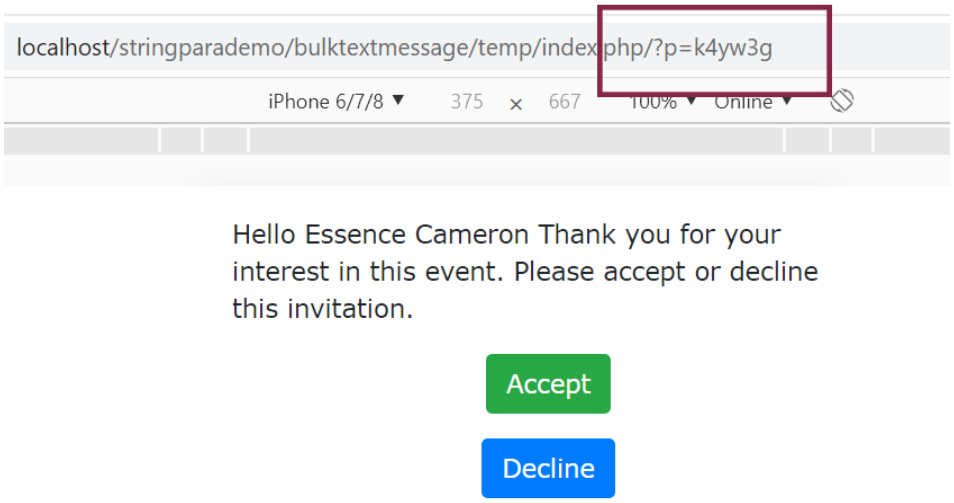

***Figure 5.13**: Fetching data from database using encryption key. Name successfully displayed*

5.3 System Testing

System testing involves merging all major components of the system into a complete integrated system and then testing it for usability. System testing makes sure that all components are "compatible, interact correctly and transfer the right data at the right time across their interfaces" [9].

System Testing was carried out after migration onto the servers of The Ritzz Exclusive Guest House. The two applications that complement each other were tested according to the user flow in Figure 3.3. The marketing and systems administrator of The Ritzz Exclusive Guest House was involved with this process to ensure that the project meets all the requirements that were outlined during requirements gathering. The system was mainly tested for usability and functionality.

Chapter 6: Conclusion and Recommendations

This chapter will outline proposed changes to the existing software that will improve its functionality, overall usability and efficiency. Further, challenges faced during development and limitations of the system will be discussed.

6.1 Recommendations

The decision to create two individual applications was as a result of discussions and requirements with management of the guest house. This was not entirely disputed as it offered the developer the opportunity to demonstrate his knowledge and skills using different programming languages and frameworks. Moving forward, usability and functionality of the system can be improved by the consolidation of both systems into a single web-based application.

6.2 Challenges

One major challenge encountered was using the PyQt Designer drag and drop feature to develop the interface of the application. After generating the Python front-end file and running the application, items on the screen remained small. See Figure 6.1.

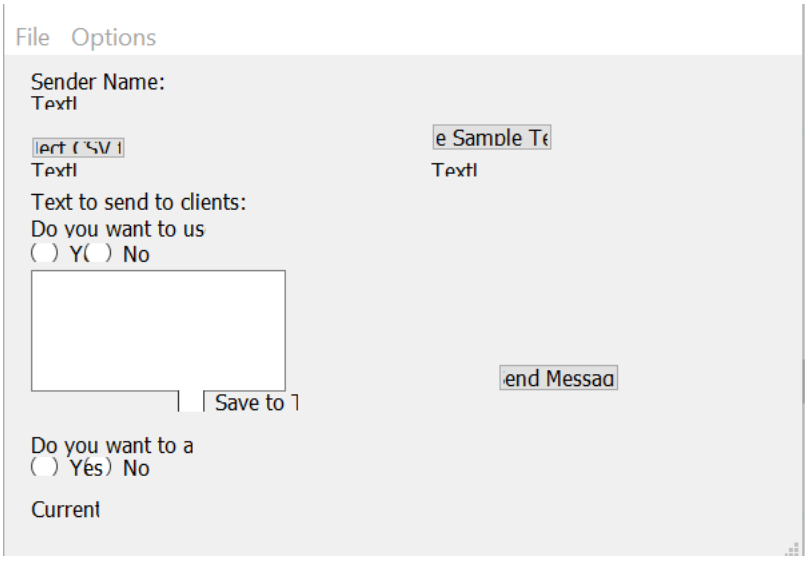

***Figure 6.1**: Distorted view during development*

After several hours spent looking for a solution with no results, the developer experimented with adjusting the screen resolution in the PC's personalization settings. This immediately resolved the issue although another interface had to be developed.

Another challenge emerged during the implementation of the core database operations, specifically with uploading data from the csv file to the database. The challenge was as a result of restrictions that were introduced during the setup of the database and was adjusted accordingly.

Outdated developer documentation from the owners of the API, Darrel Technologies, also stalled the development of the desktop application. However, this was resolved upon consulting recent Python library documentation.

6.3 Recommendations for Future Work

Much can still be done to further improve the system. In the future, the guest house may decide to consolidate both applications into one web-based application. Changing the client's view to match a campaign or survey can be implemented. Graphs and charts can be added to the data management platform as proposed during requirements analysis.

References

[1] G. G. Seleka and F.-M. E. Uzoka, "B2C e-commerce development in Africa: Case study of Botswana," in *Proceedings 7th ACM Conference on Electronic Commerce*, Ann Arbor, 2006.

[2] Business2Community, "The Importance of Information Technology In Business Today," 3 December 2015. [Online]. Available: https://www.business2community.com/tech-gadgets/importance-information-technology-business-today-01393380.

[3] E. K. Dogbevi, "Ghana has 2,723 hotels and lodges – PwC," 23 August 2017. [Online]. Available: https://www.ghanabusinessnews.com/2017/08/23/ghana-has-2723-hotels-and-lodges-pwc/.

[4] Celerity Systems Ltd, "What is bulk SMS messaging?," 2 June 2018. [Online]. Available: https://www.bulksms.com/resources/insights/what-is-bulk-sms-messaging.htm.

[5] M. Solcansky and . I. Simberova, "Measurement of marketing effectiveness," *Economics and management 15,* pp. 755-759, 2010.

[6] Y. Bakr, A. Tolba and H. Meshreki, "Drivers of SMS advertising acceptance: a mixed-methods approach," *Emerald Insight,* pp. 96-118, 2018.

[7] J. P. Anbu K. and M. R. Mavuso, "Old wine in new wine skin: marketing library services through SMS-based alert service," *Emerald,* vol. 30, no. 2, pp. 310-320, 2011.

[8] S. Dix , K. Jamieson and A. S. Shimul, "SMS advertising the Hallyu way: drivers, acceptance and intention to receive," *Asia Pacific Journal of Marketing and Logistics,* vol. 28, no. 2, 2016.

[9] I. Sommerville, Software Engineering - 9th Edition, Boston: Addison-Wesley, 2011, p. 251.

[10] PHP, "Documentation - Introduction," 2020. [Online]. Available: https://www.php.net/manual/en/intro-whatis.php.

[11] Oracle, "Documentation - MySQL Server," 2020. [Online]. Available: https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html.

[12] M. Heron, V. L. Hanson and I. Ricketts, "Open source and accessibility: advantages and limitations," *Journal of Interaction Science,* 2013.

[13] Python, "What is Python? Executive Summary," [Online]. Available: https://www.python.org/doc/essays/blurb/.

[14] A. Melvine, "Complete user registration system using PHP and MySQL database," [Online]. Available: https://codewithawa.com/posts/complete-user-registration-system-using-php-and-mysql-database.

[15] D. Raggett, A. L. Hors and I. JacobsIan, HTML 4.0 Specification, W3C Recommendation, 1998.

[16] MDN, "What is JavaScript?," 30 March 2020. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript.

Appendix

A.1 Bulk SMS Program Code Snippets

```
#Reads from a given .csv file and appends items into their respective lists
#decleared above.

    with open(input_file) as csv_file:
        count = 0
        csv_reader = csv.reader(csv_file, delimiter = ',')
        for row in csv_reader:
            if count == 0:
                count+=1
            else:
                #Generates a string if there are only two elements in a row.
                if len(row) == 2:
                    recipientNumberList.append(row[0])
                    recipientNameList.append(row[1])
                    encryptionKey.append(stringGeneraton(6))
                #Generates a string if a rows ecryption key is missing.
                elif row[2] == '':
                    recipientNumberList.append(row[0])
                    recipientNameList.append(row[1])
                    encryptionKey.append(stringGeneraton(6))

                else:
                #If the file looks fine with respective encryption keys,
                #it adds to their respective lists.
                    recipientNumberList.append(row[0])
                    recipientNameList.append(row[1])
                    encryptionKey.append(row[2])
```

***Figure A.1.1**: CSV file handling under various cases*

```
    #Generates the rows using a temp list
    #for each element from the previous list.
    finalList.append(['to','from','key'])
    for i in range(len(recipientNumberList)):
        tempList=[]
        tempList.append(recipientNumberList[i])
        tempList.append(recipientNameList[i])
        tempList.append(encryptionKey[i])
        finalList.append(tempList)
    csv_file.close()

#Updates the .csv file with respective encryption keys.
with open(input_file, mode= 'w') as csv_file:
    count = 0
    csv_writer = csv.writer(csv_file, delimiter = ',', lineterminator='\n')

    for i in range(len(recipientNumberList) + 1):
        csv_writer.writerow(finalList[i])
    csv_file.close()
```

***Figure A.1.2**: Writing to CSV file with key*

```
#Sending Text Messages with URL added using API from DTechGhana
for i in range(len(recipientNumberList)):
    recipientNumber = recipientNumberList[i]
    recipientName = recipientNameList[i]
    baseParamsForUrlSent = {"tel":recipientNumber}
    urlSent = "http://example.com/data?{}".format(urllib.parse.urlencode(baseParamsForUrlSent))
    content = 'Hello {0}'.format(recipientName)+'! ' + body + ' ' + urlSent
    baseParams = {'to':recipientNumber, 'from':sender, 'content':content, 'id':'mohieezy@gmail.com', 'secret':secret}
    res = urllib.request.urlopen("https://sms.dtechghana.com/api/v1/send?%s" % urllib.parse.urlencode(baseParams)).read()
```

***Figure A.1.3**: Sending an SMS campaign*

A.2 Data Management Platform Code Snippets

```php
    //User Login
    if (isset($_POST['login_user'])) {
        $username = mysqli_real_escape_string($db, $_POST['username']);
        $password = mysqli_real_escape_string($db, $_POST['password']);

        if (empty($username)) {
            array_push($errors, "Username is required");
        }
        if (empty($password)) {
            array_push($errors, "Password is required");
        }

        if (count($errors) == 0) {
            $password = md5($password);
            $query = "SELECT * FROM users WHERE username='$username' AND password='$password'";
            $results = mysqli_query($db, $query);

            if (mysqli_num_rows($results) == 1) {
                $_SESSION['username'] = $username;
                $_SESSION['success'] = "You are now logged in";
                header('location: index.php');
            }else {
                array_push($errors, "Wrong username/password combination");
            }
        }
    }
```

***Figure A.2.1**: Login and validation*

```php
if (isset($_POST['reg_user'])) {
        //Receive all input values from the form
        $username = mysqli_real_escape_string($db, $_POST['username']);
        $email = mysqli_real_escape_string($db, $_POST['email']);
        $password_1 = mysqli_real_escape_string($db, $_POST['password_1']);
        $password_2 = mysqli_real_escape_string($db, $_POST['password_2']);

        //Form validation: ensure that the form is correctly filled
        if (empty($username)) { array_push($errors, "Username is required"); }
        if (empty($email)) { array_push($errors, "Email is required"); }
        if (empty($password_1)) { array_push($errors, "Password is required"); }

        if ($password_1 != $password_2) {
            array_push($errors, "The two passwords do not match");
        }

        //Register user if there are no errors in the form
        if (count($errors) == 0) {
            $password = md5($password_1);//encrypt the password before saving in the database
            $query = "INSERT INTO users (username, email, password)
                      VALUES('$username', '$email', '$password')";
            mysqli_query($db, $query);

            $_SESSION['username'] = $username;
            $_SESSION['success'] = "You are now logged in";
            header('location: index.php');
        }
    }
```

***Figure A.2.2**: Registration and validation*

```php
if(isset($_POST["submit"])){
    if($_FILES['file']['name']){
        $filename = explode(".", $_FILES['file']['name']);
        if($filename[1] == 'csv'){
            $handle = fopen($_FILES['file']['tmp_name'], "r");
            $data = fgetcsv($handle,1000,",");
            while($data = fgetcsv($handle,1000,",")){
                $item1 = mysqli_real_escape_string($connect, $data[0]);
                $item2 = mysqli_real_escape_string($connect, $data[1]);
                $item3 = mysqli_real_escape_string($connect, $data[2]);
                $query = "INSERT into clients (id, name, crypt) values ('$item1','$item2','$item3')";
                mysqli_query($connect, $query);
                }
        fclose($handle);
        echo "<script>alert('Import done');</script>";
        }
    }
}
```

***Figure A.2.3**: Uploading CSV file to database*

```php
if($query){
    while($row = mysqli_fetch_array($result)){
?>

    <tr>
      <th><?php echo $row['name']; ?> </th>
      <th> <?php echo $row['id']; ?> </th>
      <th> <?php echo $row['response']; ?> </th>

        <form action="update.php" method="post">
            <input type="hidden" name="id" value="<?php echo $row['id'] ?>">
            <th> <input type="submit" name="edit" class="btn btn-primary" value="Edit"> </th>
        </form>

        <form action="delete.php" method="post">
            <input type="hidden" name="id" value="<?php echo $row['id']?>">
            <th> <input type="submit" name="delete" class="btn btn-danger" value="Delete"> </th>
        </form>
    </tr>
<?php
   }
}
else{
    echo "No record found";
}
```

***Figure A.2.4**: Displaying table dynamically after data upload*

```php
<?php
if(isset($_POST['update'])){
    $response = $_POST['response'];
    $query = "UPDATE clients SET response='$response' WHERE id=$id";
    $run_query = mysqli_query($connect, $query);

    if($run_query){
        echo '<script> alert("Data Updated"); </script>';
        header("location:index.php");
    }
    else{
        echo '<script> alert("Data not updated"); </script>';
    }

}
?>
```

***Figure A.2.5**: Updating client responses (admin side)*

```php
if(isset($_POST["export"])){
     header('Content-Type: text/csv; charset=utf-8');
     header('Content-Disposition: attachment; filename=data.csv');
     $output = fopen("php://output", "w");
     fputcsv($output, array('phone', 'name', 'crypt', 'response'));
     $query = "SELECT * from clients ORDER BY id DESC";
     $result = mysqli_query($connect, $query);
     while($row = mysqli_fetch_assoc($result)){
          fputcsv($output, $row);
     }
     fclose($output);
 }
```

***Figure A.2.6**: Fetching and downloading data as CSV from the database*