ASHESI

ASHESI UNIVERSITY COLLEGE

ZER0CLOCK: AN ONLINE APPOINTMENT BOOKING SYSTEM

APPLIED PROJECT

B.Sc. Management Information Systems

Warihana Gumah

2018

ASHESI UNIVERSITY COLLEGE

zer0Clock: An Online Appointment Booking System

APPLIED PROJECT

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

Warihana Gumah

April 2018

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 College.

Supervisor’s Signature:

…………………………………………………………………………………………

Supervisor’s Name:

…………………………………………………………………………………………

Date:

.……………………………………………………………………………………….

ACKNOWLEDGEMENT

I would like to thank the Almighty Allah for giving me strength to complete this project. I would like to express my profound gratitude to all who assisted me most especially my project supervisor, Mr. David Amatey Sampah, for his priceless guidance, criticisms and overall support towards the successful completion of this project.

Abstract

Research shows that queuing is a problem in Ghana especially in hospitals and public-sector offices. Due to long queues, most Ghanaians are demotivated to take part in health benefitting activities such as vaccinations. Again, some Ghanaians pay bribes to public sector officials to avoid long queues and multiple visits to public sector offices for a service. This is one of many factors that accounts for the high bribery and corruption rate in Ghana. In this regard, this project sought to develop “zer0Clock”: a generic online appointment booking system that allows prior booking before an appointment. Requirements for this application were gathered from prospective users. On this platform, businesses and individuals can register and schedule their events. Clients can visit this platform and book an appointment to guarantee service completion on the day of visit and to avoid long queues. zer0Clock also has a video and live chat feature for virtual appointment execution. It also has a notification feature that ensures that clients who book appointments do not miss them by sending a reminder (usually 30 minutes before time). There is also a google map functionality that provide detailed direction for clients to navigate venues of appointments. There is a high level of conviction that this application will help reduce long queues if not eliminate them, because there has been a steep increase in internet usage in Africa over the past 10 years and all these vast number of internet users have the opportunity to use this application.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

1.1 Background

The conventional way for Ghanaians to have access to most services in Ghana is through queuing. Queuing is ubiquitous in Ghana especially in places such as hospitals, passport and license offices, among others. Considering that Ghana is a developing country (United Nations, 2014), inadequate basic infrastructure is one of the probems the citizens of the country face. One of these basic infrastructure the country lacks is sitting areas. As such, citizens have to queue in the sun for long hours which posses severe health hazards.

In the case of hospitals in Ghana, people have to queue for hours during vaccinations and health check-ups to see a specialist or a doctor. Again, this is attributable to the inadequate healthcare facilities in a developing country (United Nations, 2014) of 28.21 million people (World Bank, 2017). More often than not, people are demotivated to take part in important activities at hospitals in Ghana. When others take time out of their busy schedules to visit the venue to take part in vaccinations or health check-ups, greater percentage of them leave the queue after having spent considerable amount of time, especially when there are many people ahead of them (Zhou & Soman, 2003).

Moreover, in public service sectors such as passport, Driver and Vehicle Licensing Authority (DVLA), birth and death registry offices, the issue of queuing for hours still arises. Beinpuo (2015) vividly expressed his frustrating experience at the Ridge Passport Office in Ghana. He complained about the "long queue in front of the passport office, delays in processing and issuance of passports" (Beinpuo, 2015). Aside the fact that citizens are being exposed to the sun while standing outside for hours, most of them have to make several visits to the passport office to join long queues in acquiring a passport (Beinpuo, 2015). Recently, queuing is the norm in Ghana.

1.2 Problem Statement

Queuing for long hours is one of the major problems Ghanaians face. Queuing has continuously led to immense waste of time that could have been used to engage in productive activities to make valuable contribution to economic development. When people consider the time that will be spent waiting in a queue and its opportunity cost, they might not take part in certain key activities. Due to queuing, some Ghanaians are demotivated to participate in health benefiting activities that can reduce disease severity, complications, and comorbidities (Bonanni, Picazo, & Remy, 2015). Similarly, to avoid these long queues and for that matter, time wastage and additional cost of visiting the venue, citizens pay bribes to the public officials (Riley, 1999). This is one of the accounting factors for Ghana being one of the countries with high bribery and corruption rates in the world.

1.3 Objectives

The objectives of this project are to design a generic but customizable online appointment booking system that ensure the following:

- Save clients time spent in long queues by allowing them to book an appointment prior to the day when the activity is taking place. Saving time of clients will in turn address its related problems such as petty bribes and demotivation to take part in health benefiting activities.
- Save clients time wasted searching for unfamiliar locations where major activities are taking place by providing them google map direction to the venue.
- Ensure that clients attend appointments they have booked by receiving notification when an appointment has successfully been booked and a reminder when it is 30 minutes to the appointment.

- Provide a live and video chat functionality which will further save time, because with this feature, they can consult with their specialists from home.
- Provide a platform for businesses to improve their services by making it possible for clients to give feedback regarding services rendered by the business.
- Increase awareness about businesses through the online platform since a web-based application can be accessed anywhere in the world.
- Provide tools to analyse bookings which will give business an insight about the patronage of their products.

1.4 Motivation

There has been a continuous increase in internet usage in Africa and this growth surpasses the growth in any part of the world (Shapshak, 2015). The internet has made it possible for several innovations. Examples of such innovation is the invention of WhatsApp, Facebook and Instagram which has helped made communication easier. The developed countries have explored the internet and have used it to solve most of the problems they face. The internet and technology have been a motivation to push the boundaries of knowledge and try to figure out ways that it can be used to address pressing problems that Ghanaians face, just as the developed countries have taken advantage of it to solve their problems. Having realised that queuing is a problem and have had first-hand experience of being in long queues for hours, implementing a platform that can eliminate or reduce queuing became an area of interest. Reducing time wasted in long queues is important especially for the 72.65% employed (World Bank, 2017) because this will help save time to engage in their routine activities and continuously contribute to economic development.

1.5 Related Works

This section analyses existing software packages and puts this proposed project;

appointment booking system in perspective with other related works. The basic functionality of this software package is to provide a platform for Ghanaians to book appointments and save the time spent in long queues. Some related works to this proposed project include but not limited to Acuity Scheduling, Simplybook.me, AppointmentCare, Appointy and Setmore. The major distinguishing features for the proposed system from these other related works include but not limited to the following:

- Price: Most of these applications have at most 30 days free trial. After the 30 days, a user must pay for continuous usage. For instance, premium, standard, and basic users of Simplybook.me pay $59.90, $29.90 and $9.90 per month respectively. Again, premium, standard, and basic users of Acuity Scheduling pay $50, $25 and $15 per month accordingly. Moreover, an Acuity user with multiple business locations pays $59.99 per month, a user with medium size business pays $39.99 per month while a user with small business pays $29.99 a month. Contrary, the proposed system will allow users free access to all the functionalities of the application once they are registered users. This will motivate the public, private as well as the NGO sectors to use this software package.
- Number of bookings: Another distinguishing feature of this proposed system from similar applications is the number of bookings it offers users. Most of these existing related software packages have a limit on the number of bookings. Taking SimplyBook.me for instance, there are limited number of bookings even for premium users. This is different in the case of this project. It is important to provide unlimited booking for users (Ghanaians) because considering the culture of the people is key in designing appointment scheduling rules and system (Ho, Lau, & Li, 1995). The Ghanaian culture is such that people prefer to attend events in large numbers when they are in serious need of it. As such, this proposed system offers

unlimited number of bookings so that everyone interested can get the opportunity to book an appointment.

- Popularity: Existing software packages are not well known by Ghanaians. However, this application is designed primarily for Ghanaians and will take the initiative to increase awareness.

1.6 Major Functionalities Implemented

- Scheduling: This functionality is concerned with scheduling events. A registered administrator of a business can generate a calendar for an event.
- Booking: With this functionality, a client of a business can book an appointment from a scheduled event calendar.
- SMS notification: This functionality deals with sending Short Message Service (SMS) notification to clients when an appointment has successfully been booked and when it is 30 minutes to the appointment.
- Google map direction: This functionality provides a detailed google map direction to the venue for an appointment.
- Feedback: With this functionality, a client can give feedback to businesses about their services. This will give businesses the opportunity to improve their services.
- Report generation: This provides a comprehensive dashboard which shows summaries of clients' bookings as well as their feedback.
- System configuration: This component handles configuring and creating a customizable booking website for businesses.
- Live and video chats: This allow clients to communicate with their consultants from home.

Chapter 2: Requirements

2.0 Chapter Overview

This chapter outlines the requirements of this system based on the international standard for software requirement specification. It first describes the requirements gathering process and analysis. Again, it identifies the various users of the system, their requirements and the data requirements in their interactions with the system. Moreover, it discusses the system and non-functional requirements of this application.

2.1 Requirements Gathering and Analysis

2.1.1 Requirements Gathering

The requirements of this application were gathered through interviewing the various stakeholders which included companies and their clients. The interviews revolved around obtaining information regarding the following:

I. How do companies appropriately schedule for activities to ensure efficient utilization of time and other resources?

II. What major functionalities are key to users to make the system useful to them?

III. How do businesses track the level of patronage of their products?

IV. What is users' level of education and technological competence?

V. What interface and level of technicality is desirable?

VI. How do firms obtain customer feedback to be able to improve their services?

2.1.2 Requirement Analysis

Detailed analysis of the information gathered from the interviews showed that the following are of major importance to the various stakeholders: businesses and their clients.

These findings are grouped according to the interview questions in section 2.1.1 and the individual stakeholders.

2.1.2.1 Perspective of Businesses

I. Examination of responses for interview question number I showed that businesses are interested in a platform that allow clients to book an appointment prior to when an activity is taking place so that they can effectively plan.

II. Users reply for interview question II indicated that a system that will increase people's awareness of their businesses worldwide and widen their customer scope is important to the success of their businesses. Again, businesses mentioned that the system should provide for customization since it is a generic application.

III. Analysis of users' feedback for interview question III indicated that they need a system that provide comprehensive statistical analysis of the patronage of their products.

IV. With regards to interview question IV and V, administrators of businesses have at least an undergraduate degree and have had experience in using web applications. Moreover, these users are medium technological competent thus, they require an interface that is attractive and easy to use.

V. A platform that give room for customer feedback for improvement of services rendered was users' response for interview question VI.

2.1.2.2 Business Clients' Perspective

The responses of business clients stressed much on interview question II. The insights drawn from the interview with these users showed they are interested in the following functionalities:

VI. A flexible system that allow for easy booking, cancellation and rebooking of appointments.

VII. A way to remind clients when it is 30 minutes to an appointment.

VIII. A platform that makes it possible to interact with specialists via live chats or video call.

IX. A system that allows calendar generation.

X. Detailed and accurate directions to venue where an activity is taking place.

2.2 Purpose of Software

The goal of this project is to design a generic web-based appointment booking system called "zer0Clock". This system is designed for businesses in the public, private, NGO and other sectors and their clients.

2.3 Product Functions

The functions of the system which indicate the services it provides (Belitsoft, 2016) are discussed below. These functions address the various needs of users in section 2.1.2. They include the following:

- Scheduling: This application generates a schedule for registered users of “zer0Clock” based on their specified activity, time needed to attend to a person, working hours and break times. This function addresses users’ requirement I.
- Booking: Since the schedule of registered businesses or individuals can be seen by everyone, clients can book a slot from the schedule and attend an appointment. This gives customers the freedom to choose the appropriate time to book an appointment based on their schedules. This functionality of "zero0Clock" responds to users’ requirement VI.

- Notifications: After a user has successfully booked an appointment, an SMS notification is sent to the user to indicate that the appointment has successfully been booked. Again, when it is 30 minutes to the appointment, another SMS notification is sent to the client to remind them of the appointment. This addresses users' requirement VII
- Live chats and video calls: Customers expressed that it is time saving to be able to chat with a specialist from home. This, and considering users' requirement VIII, "zer0Clock" provides live and video chat functionality for users. This particular feature is of much importance to clients of family doctors, counsellors, lawyers and other individual and private organizations.
- Google map direction: To provide for users' requirement X, this system will provide users who have successfully booked an appointment access to a google map direction to the venue where an event such as vaccination is taking place. This gives clients the chance to patronize the service of unfamiliar service providers.
- Report: This functionality allows administrators and management of businesses to view the patronage of their services. Businesses can view summaries of all data relating to their client's bookings on their business dashboard. This feature addresses users' requirement III
- Client feedback: In response to requirement V, this application provide a functionality which allows clients to give feedback about the services rendered to them to provide room for businesses to improve. The feedback is anonymous so that clients can give honest feedback without fear of being identified.
- Customization: "zer0Clock" is a generic but customizable web-application. This feature addresses user requirement II. This enables registered firms to customize their booking websites to their specifications.

- Calendar generation: "zer0Clock" allow standard users who have successfully booked an appointment to generate a calendar. This functionality helps clients of businesses to keep track of their appointment and this addresses users' requirement IX.

2.4 Users Classes and Interactions

2.4.1 User Classes and Characteristics

There are two major classes of users of this application are businesses and their clients. In this application, businesses are known as administrative users and clients are standard users. The various categories of users of “zer0Clock” and their characteristics are outlined below:

- Administrative users: Individuals and businesses that have registered to use “zer0Clock” have one major user who will be the administrator of their customizable booking website. The administrator is responsible for scheduling firm’s events and customizing the booking of the website of the company. As such, it is key that the administrator must be technologically competent. He or she has the authorization to read and write.
- Standard users: The second class of users of this application include clients of businesses. They are interested in viewing and understanding a schedule and being able to book an appointment. A client must understand the English Language to be able to effectively use this application.

2.4.2 Use Case Diagram

This application has two (2) major categories of users. The use case diagram below indicates how the various users utilize this appointment booking system.

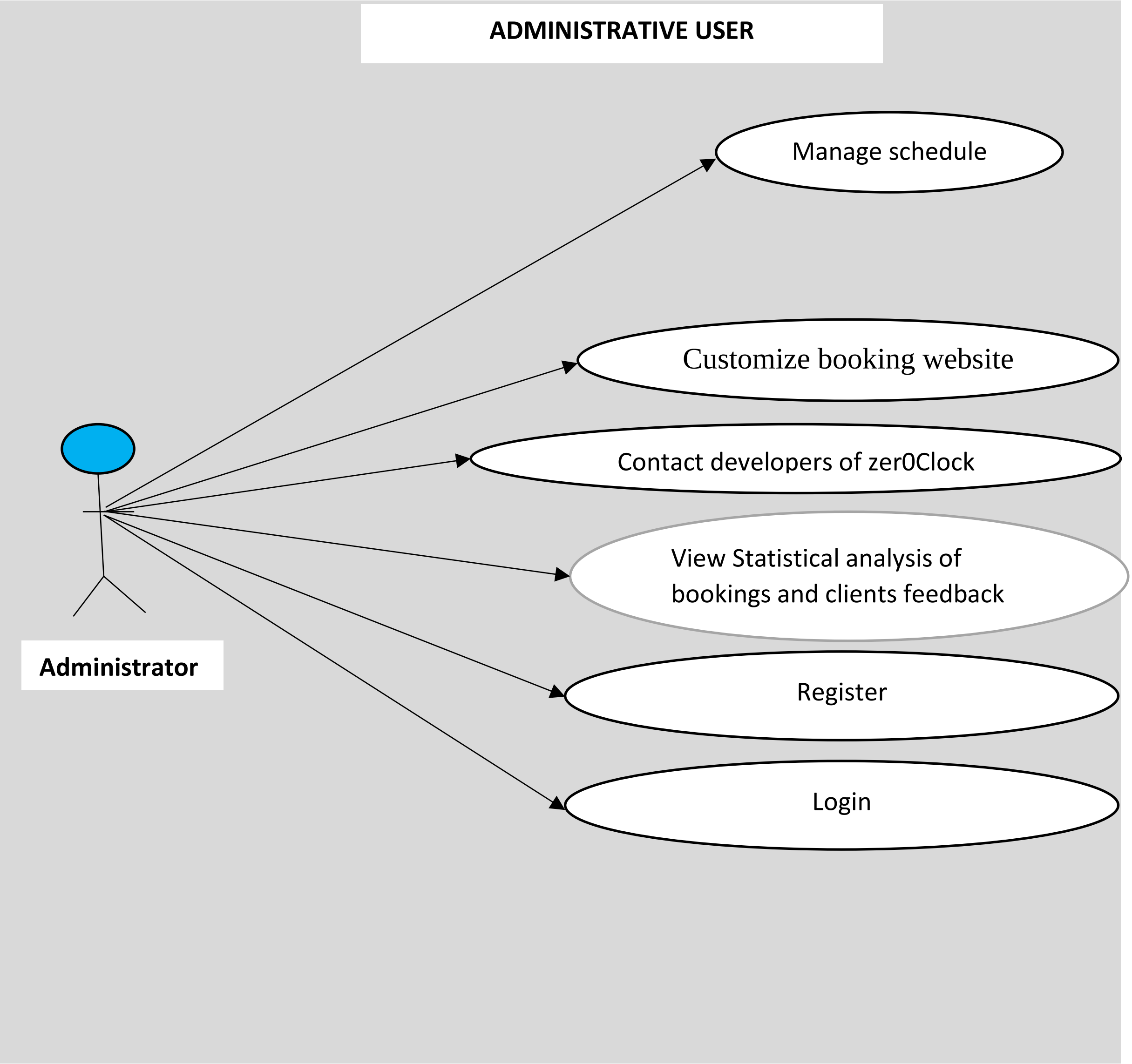

ADMINISTRATIVE USER
Manage schedule
Customize booking website
Contact developers of zer0Clock
View Statistical analysis of bookings and clients feedback
Register
Login
Administrator

Figure 2.1: Use case diagram showing the requirements of an administrative user

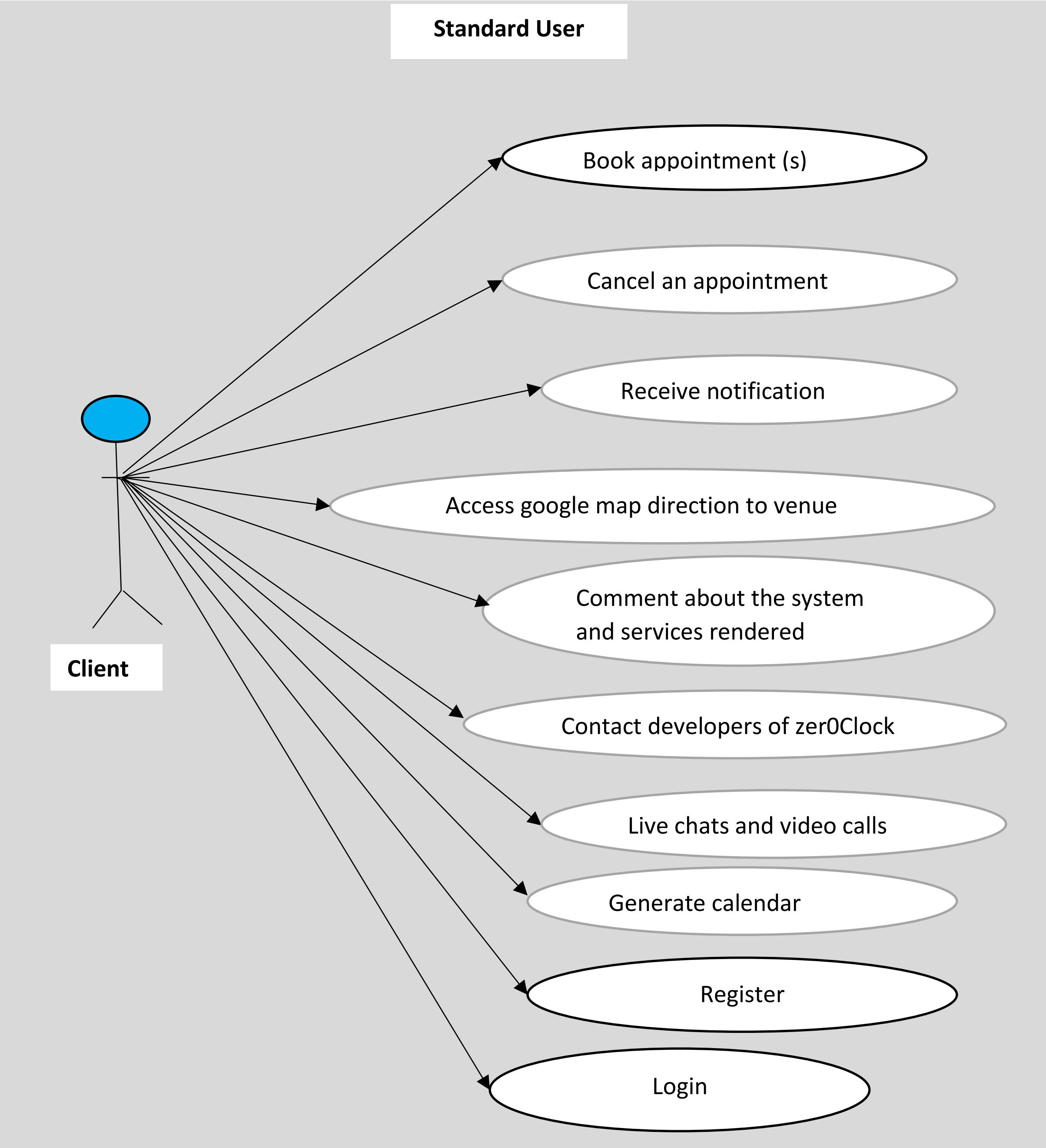

Standard User
Book appointment (s)
Cancel an appointment
Receive notification
Access google map direction to venue
Comment about the system and services rendered
Client
Contact developers of zer0Clock
Live chats and video calls
Generate calendar
Register
Login

Figure 2.2 Use case diagram showing the requirements of a standard user

2.5 Data Requirements of Users

This section outlines in detail the data users are required to provide at every stage of their interaction with this system. These requirements are grouped according to the user classes: administrative and standard users.

2.5.1 Data Requirements of Administrative Users

The administrative users interact with this system in various ways. The tables below show the data requirements of selected interactions with "zer0Clock":

Table 2. 1 Administrative user data requirement during registration

FR-BR-1: A business should be able to register

Description	This requirement allows businesses to register and become registered users of “zer0Clock”
Input	Company name, email, business category, phone, password, description
Source	A form filled by a business
Output	Alert user to enter data when form fields are empty or alert user to enter valid input when invalid or wrong input is provided. Submission of the form can either be a success or failure. When form submission is successful, display “Registration successful” and send an email to user email provided during registration to inform the firm that it is now a registered user of “zer0Clock”
Destination	Alert in the business view and a notification to the business provided email address
Trigger	User clicks the registered button
Pre-condition	None
Post-condition	Business is added to the databases of registered users

Table 2. 2: Administrative user data requirement during login

FR-BR-2: A business must be able to login

Description	This requirement allows registered users to login and use the system
Input	email, password
Source	A form filled by a business

Output	Alert user to enter data when form fields are empty or alert user to enter input provided during registration when inputs provided does not match data in the database. Form submission can be a success or failure. When form submission is successful, display “Login successful” and redirect business to its dashboard
Destination	Alert in the business view
Trigger	User clicks the login button
Pre-condition	User must be a registered user of “zer0Clock”
Post-condition	Business is redirected to their dashboard and a session created for the user

Table 2. 3: Administrative user data requirement in scheduling an event

FR-BR-3: A business should be able to schedule an event

Description	This requirement allows administrative users to schedule for events or activities
Input	Activity title, start date of activity, end date of activity, time required to attend to a person, working hours and break times
Source	A form filled by a business
Output	Alert user to enter data when form fields are empty or alert user to enter valid input when invalid or wrong input is provided. Submission of the form be either be a success or failure. When form submission is successful, display “Schedule successfully created” and redirect business to view all ongoing scheduled activities of the firm
Destination	Alert in the business view
Trigger	User clicks the schedule button
Pre-condition	Business must register and logged in
Post-condition	Business schedule is added to the database of schedules

2.5.2 Standard User Data Requirements

This section describes the data required by standard users in their various interactions with the system. The tables below indicate the data requirements of users in some interactions with the system:

Table 2. 4: Standard user data requirement in booking an appointment

FR-CR-1: A client of a firm should be able to book an appointment online

Description	This requirement allows standard users to book an appointment with a firm prior to when an activity or event is taking place
Input	Full name, email address, purpose, Phone number
Source	A form filled by a client
Output	When a slot is also successfully taken, display a dialog containing a form that requires user details. Alert user to enter data when form fields are empty or alert user to enter valid input when invalid or wrong input is provided. Submission of the form can either be a success or failure. When form submission is successful, display "appointment successfully booked" and send an SMS to the phone number provided by the client
Destination	Alert in the client view and a notification sent to the client's phone number
Trigger	User clicks a slot in the business schedule
Pre-condition	None
Post-condition	Client's appointment start and end timestamp, email, full name and phone number are stored in the database

Table 2. 5: Standard user data requirement in giving feedback

FR-CR-2: A standard user should be able to give feedback

Description	This requirement allows clients of a firm to give feedback regarding the services provided by the firm
Input	Message or feedback
Source	A form filled by a client
Output	Alert user to enter data when form fields are empty or alert user to enter valid input when invalid or wrong input is provided. Submission of the form can either be a success or failure. When form submission is successful, display " Feedback successfully sent"
Destination	Alert in the client view
Trigger	User clicks send feedback button
Pre-condition	A client must have booked and attended and appointment
Post-condition	Feedback stored in the database

2.6 System Requirements

The system requirements regarding this web-based application are discussed below:

1. The application must be available and responsive on all devices that support any kind of browser such as Firefox, Google Chrome, Opera etc
2. Users must be able to customize their booking website to their specification
3. The application must to available on the internet at all times
4. Before a user can have full access to all the functionalities of the application (zer0Clock), the user must first be a registered user and must log in with the appropriate credentials

2.7 Non-functional Requirements

The non-functional requirements discussed below are concerned with the constraints on the services provided by the system:

1. Security Requirements: one major non-functional requirement of this application is security for all users. All users must be authenticated before being allowed access to the systems (Sommerville, 2011).
2. Dependability Requirements: Considering that the system is web-based application, it should be available 24 hours a day (Sommerville, 2011). The systems should be regularly maintained, and all necessary support should be provided for users when needed. Users should be able to rely on this application to provide the services it promises to offer.
3. Usability Requirements: The system should be easy to use especially by new users. The system should be very usable in a sense that a new user should be able to navigate around it without any guidance. The user interface should be very elaborate and easy to understand.

Chapter 3: Architecture and Design

3.0 Chapter overview

This chapter illustrates how this software will be designed for implementation. It also shows the high-level architecture of the overall structure of the system. Designs such as the system architecture, database design, activity diagram, interface design and sequence diagrams will be explored. These are proposed designs thus, not all components stated in these diagrams will be implemented.

3.1 System Overview and Architecture

3.1.1 System Overview

The architectural design of this system considers the various components needed to design the system and the interaction between them. The tools needed to design this web-based application include PHP, HTML, CSS, jQuery, JavaScript, database, APIs and relevant libraries. This system will be developed using the Model-View-Controller (MVC) pattern.

The view component defines how data retrieved is presented. These include the various pages of the application such as home, login, register, contact, dashboard, admins view, booking, and feedback page. The model component contains the various classes and manages the entire system and interfaces between the database and controller. The controller component contains the logic and controls user interaction from the views to the model components and the vice versa.

3.1.2 System Architecture

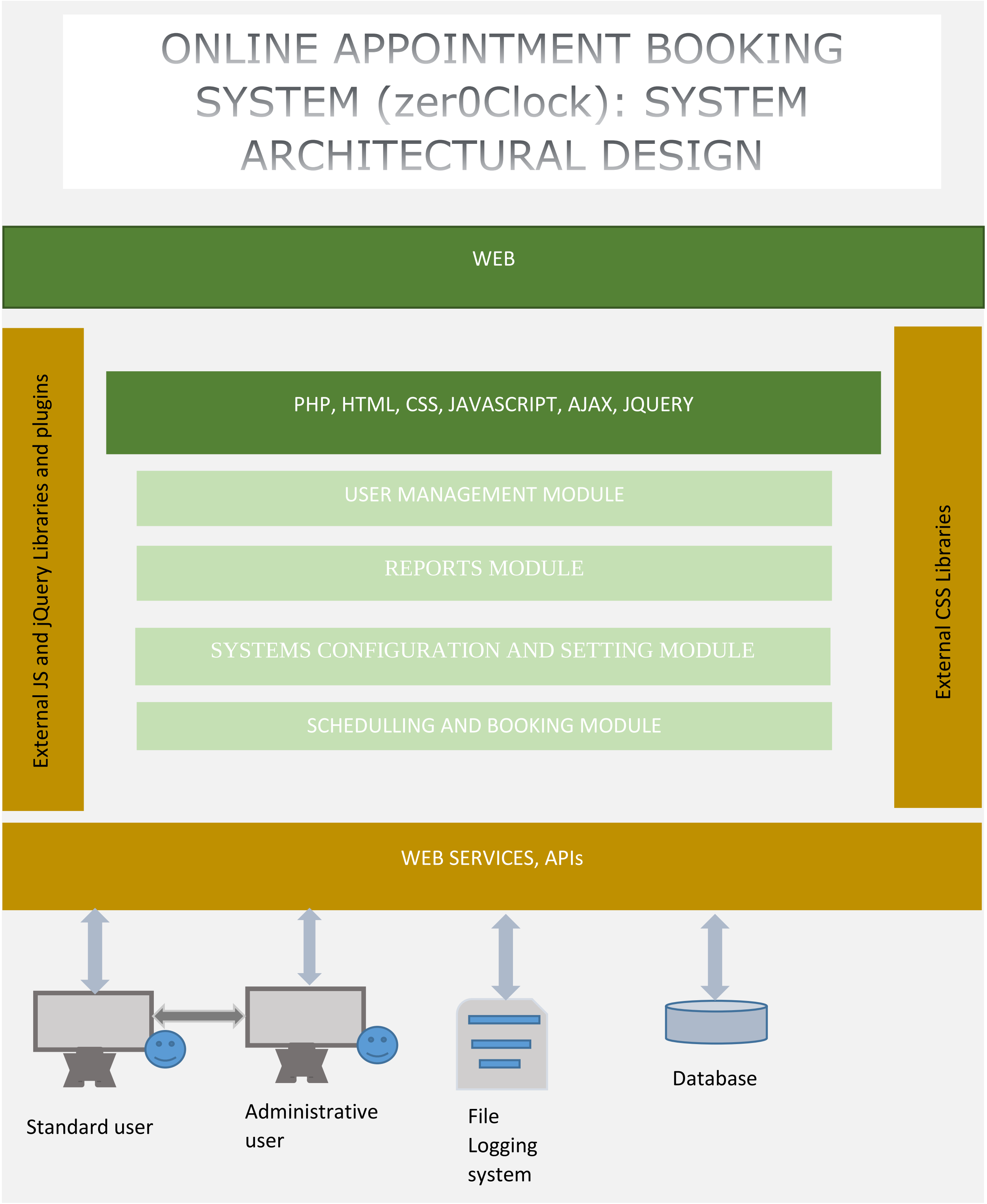

ONLINE APPOINTMENT BOOKING SYSTEM (zer0Clock): SYSTEM ARCHITECTURAL DESIGN
WEB
External JS and jQuery Libraries and plugins
PHP, HTML, CSS, JAVASCRIPT, AJAX, JQUERY
USER MANAGEMENT MODULE
REPORTS MODULE
SYSTEMS CONFIGURATION AND SETTING MODULE
SCHEDULLING AND BOOKING MODULE
External CSS Libraries
WEB SERVICES, APIs
Standard user
Administrative user
File Logging system
Database

Figure 3.1: System architecture diagram

3.2 Database Design

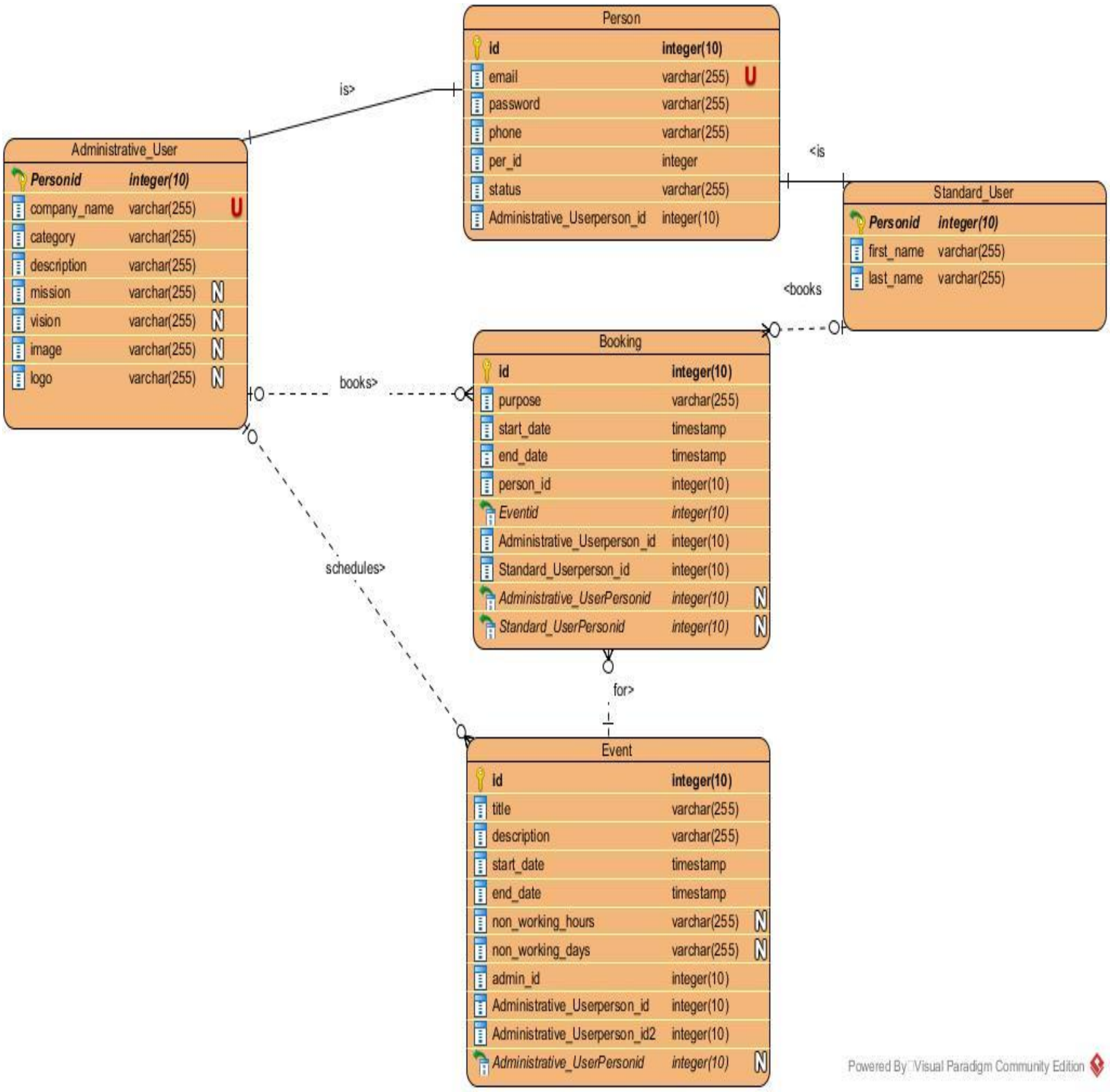


Figure 3.2: Diagram showing database design of the system

Client’s interaction with the database will be implemented using PHP. The data that will be stored in the database server are modelled into classes. Data in these classes will be stored in MySQL database and retrieved in JSON format. All communications with the database should be checked to prevent MySQL injection and other kinds of attack on the system.

3.3 Activity Diagram

The first activity diagram below shows the flow of activities of an administrative user. An administrative user is required to register to gain full access to the functionalities of the system. When registration is successful, the next stage is login; where a user provides email and password. After login, the necessary configurations are done and the company's customizable website is created. On this customizable booking website, the administrator can schedule an upcoming activity, personalize company website, etc.

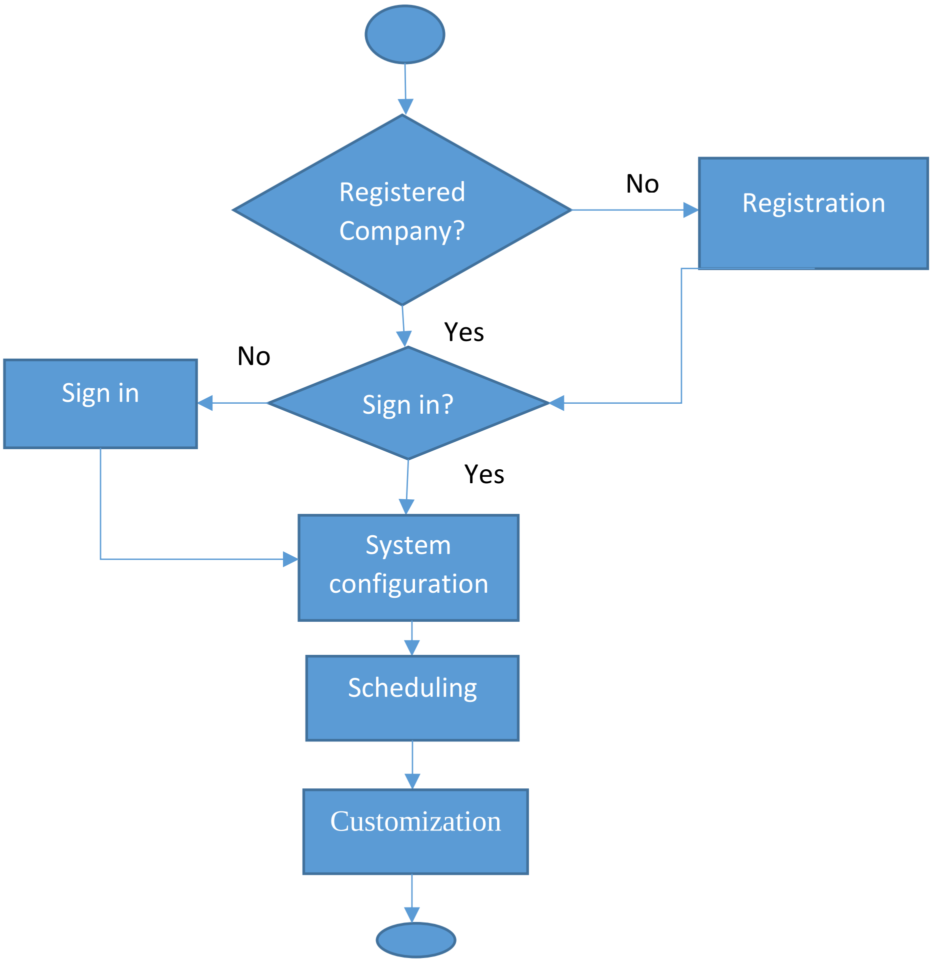


Figure 3.3: Activity diagram showing workflow of an administrative user

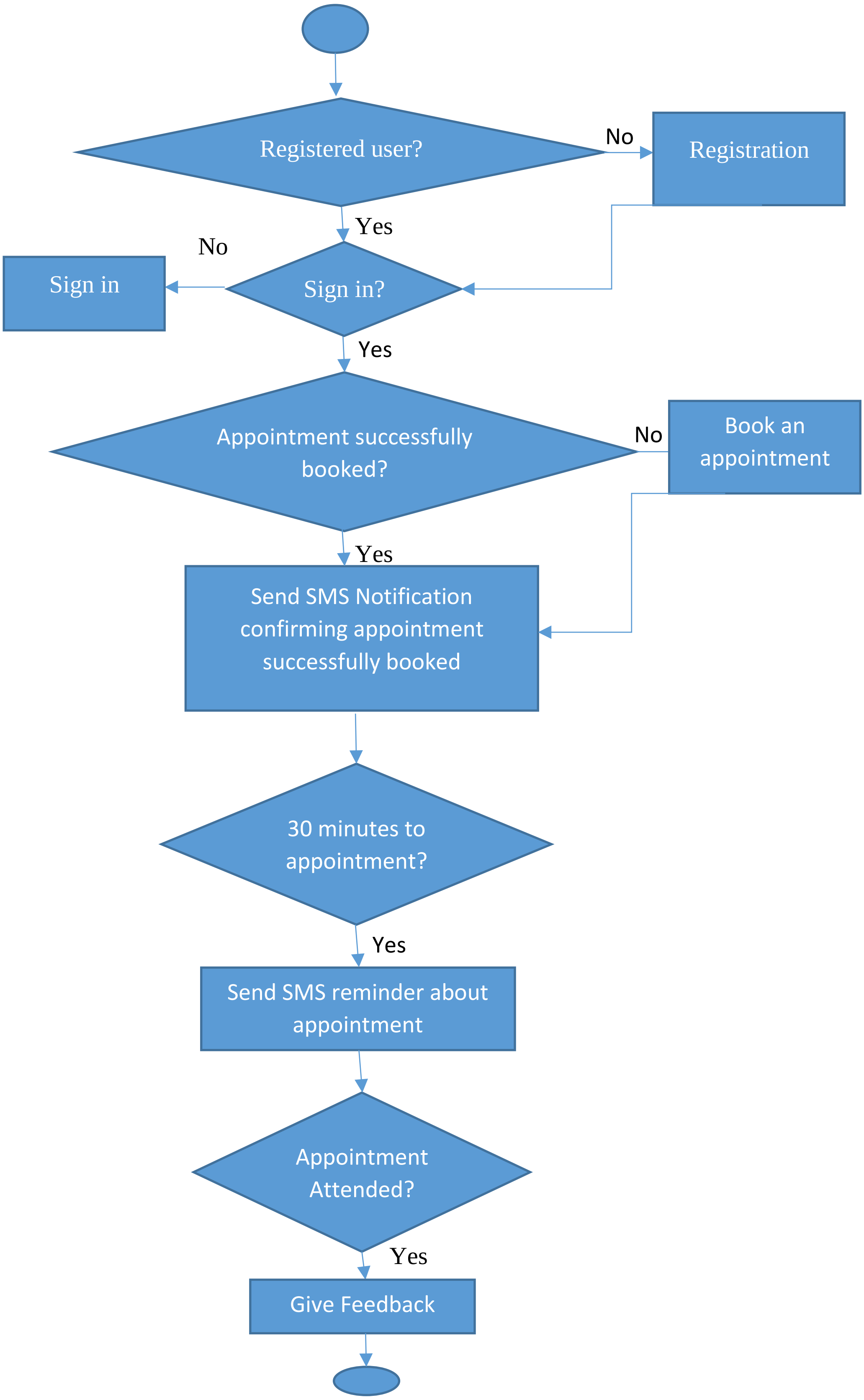
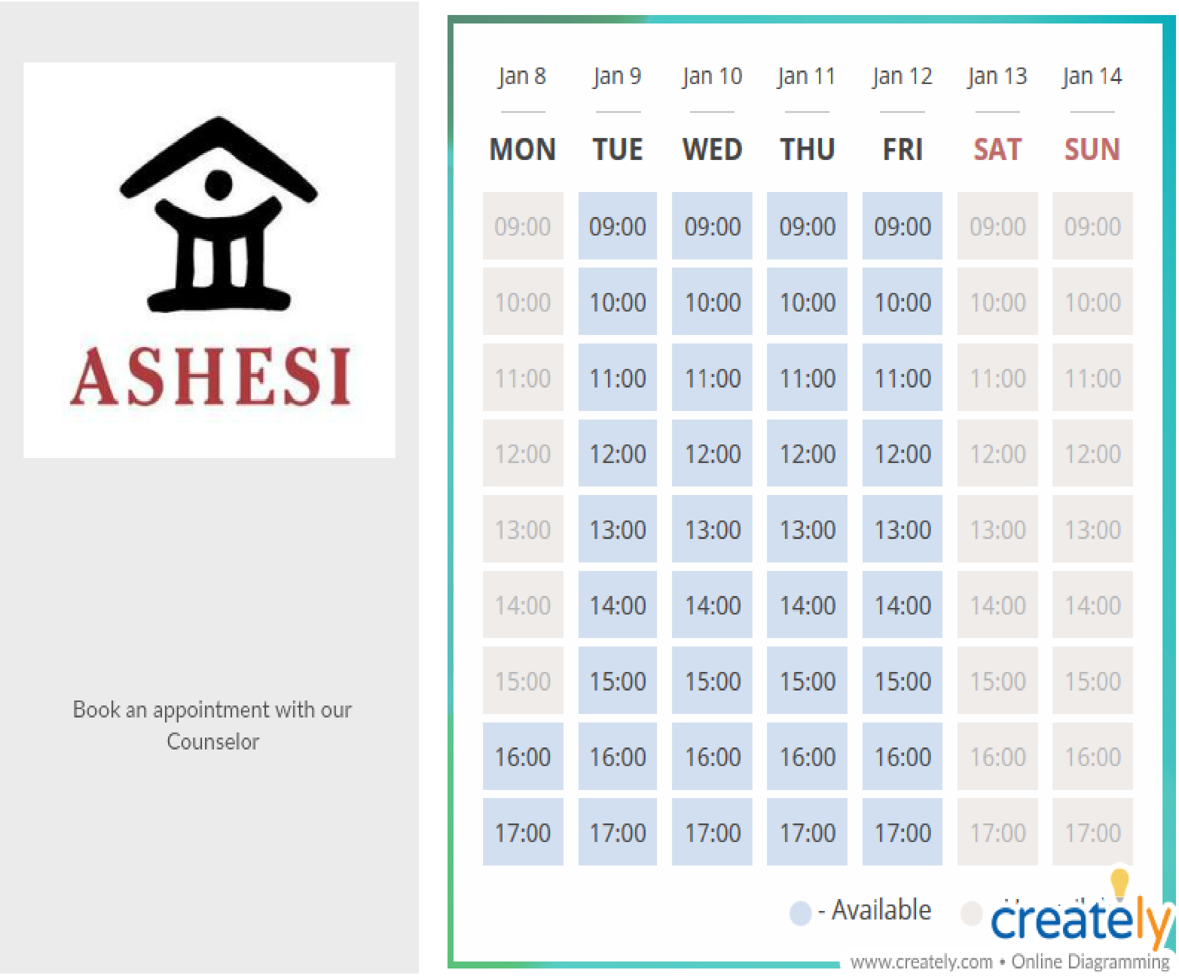

Registered user?
No
Registration
Yes
No
Sign in
Sign in?
Yes
Appointment successfully booked?
No
Book an appointment
Yes
Send SMS Notification confirming appointment successfully booked
30 minutes to appointment?
Yes
Send SMS reminder about appointment
Appointment Attended?
Yes
Give Feedback

Figure 3.4: Activity diagram showing workflow of a standard user

The second activity diagram above shows the flow of activities of a standard user who wants to book an appointment through to providing feedback after attendance.

3.4 Prototypes

This section shows prototypes of selected pages of this application. Figure 3.5 shows a prototype of the booking page where clients can book an appointment. On this page, a client can book an appointment by clicking on any of the available slots. When an available slot is chosen, there is a pop up that prompt user to enter email, purpose, name and phone number. When appoint is successfully booked, an SMS notification is sent to the client confirming that the appointment has successfully been booked.



ASHESI

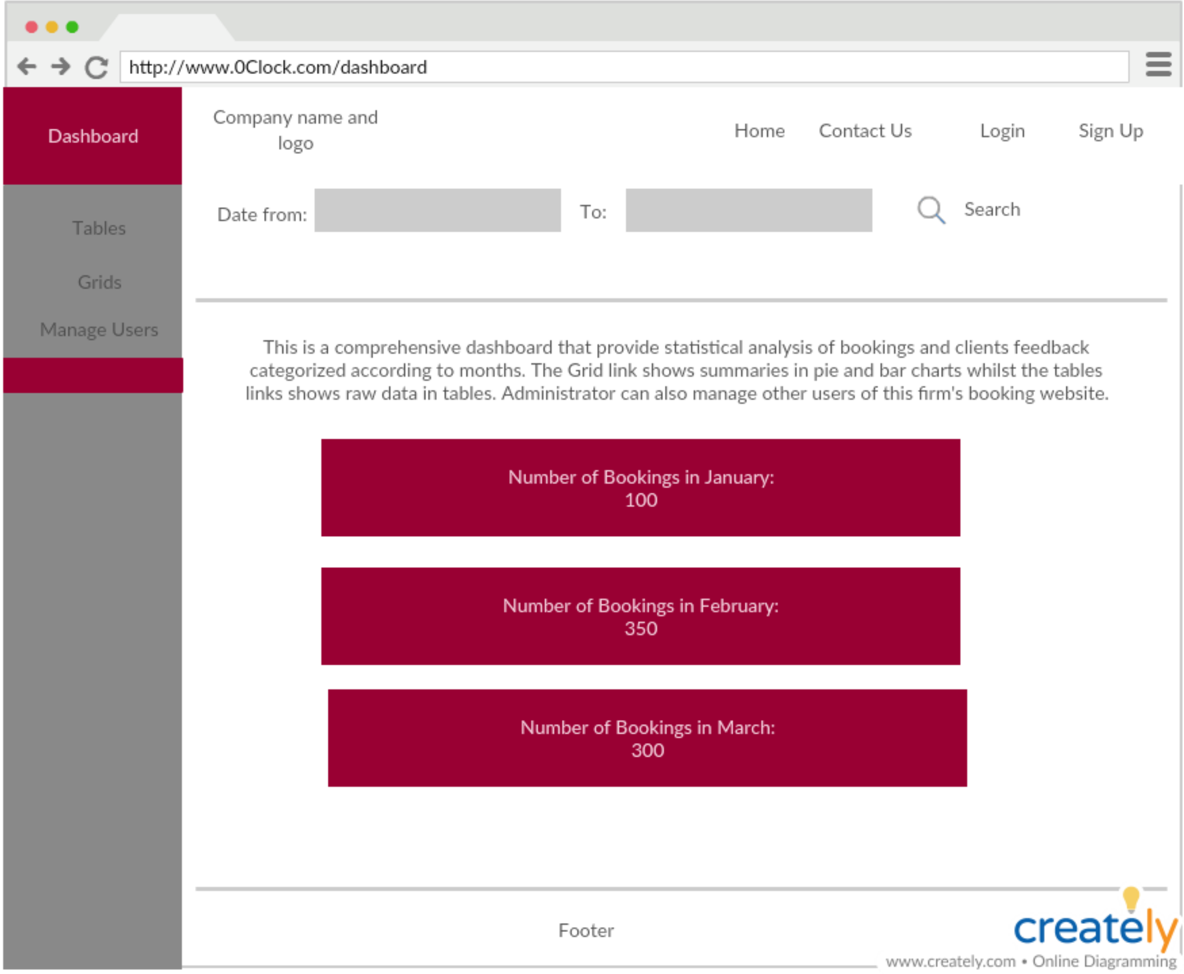
Book an appointment with our Counselor

Jan 8	Jan 9	Jan 10	Jan 11	Jan 12	Jan 13	Jan 14
MON	TUE	WED	THU	FRI	SAT	SUN
09:00	09:00	09:00	09:00	09:00	09:00	09:00
10:00	10:00	10:00	10:00	10:00	10:00	10:00
11:00	11:00	11:00	11:00	11:00	11:00	11:00
12:00	12:00	12:00	12:00	12:00	12:00	12:00
13:00	13:00	13:00	13:00	13:00	13:00	13:00
14:00	14:00	14:00	14:00	14:00	14:00	14:00
15:00	15:00	15:00	15:00	15:00	15:00	15:00
16:00	16:00	16:00	16:00	16:00	16:00	16:00
17:00	17:00	17:00	17:00	17:00	17:00	17:00

● - Available ●

Figure 3.5: Prototype of the booking (calendar) page

Figure 3.6 shows the prototype of the administrator’s dashboard. This page shows summaries of clients’ bookings as well as their feedback.



This is a comprehensive dashboard that provide statistical analysis of bookings and clients feedback categorized according to months. The Grid link shows summaries in pie and bar charts whilst the tables links shows raw data in tables. Administrator can also manage other users of this firm's booking website.

Number of Bookings in January: 100

Number of Bookings in February: 350

Number of Bookings in March: 300

Figure 3.6: Prototype of the administrator’s dashboard

3.5 Sequence Diagrams

This sections shows the sequence diagrams which indicate the interaction between the various objects and models of the system and the outcome of these interactions.

3.5.1 Administrative user registration

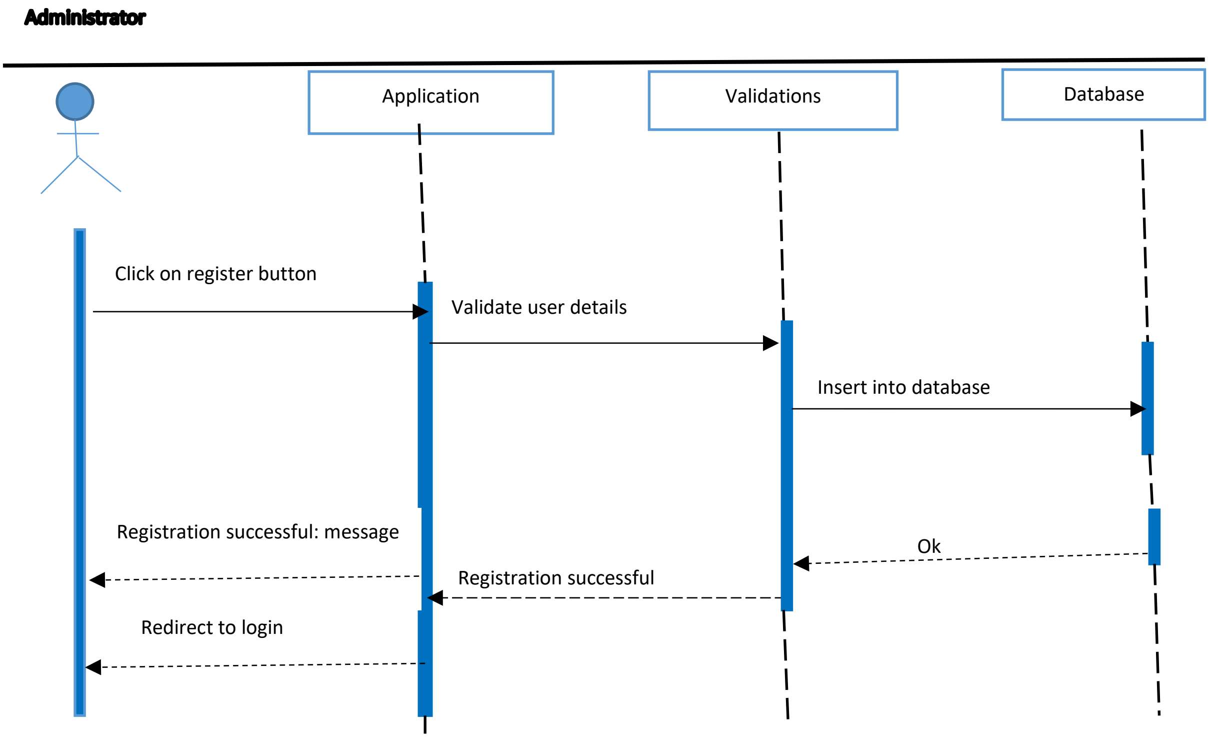


Figure 3.7 sequence diagram of a new administrator registering a company

3.5.2 Administrative user login

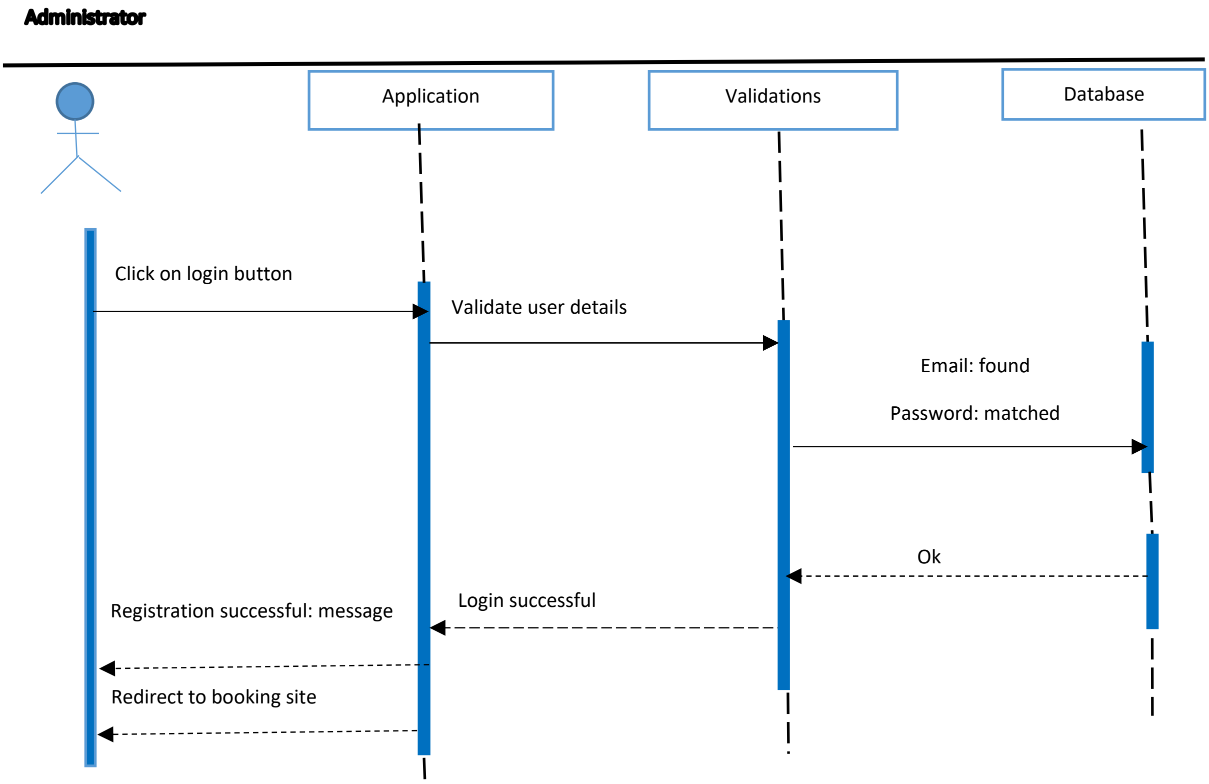


Figure 3.8: sequence diagram of an administrator’s login

3.5.3 Administrative user's other interactions

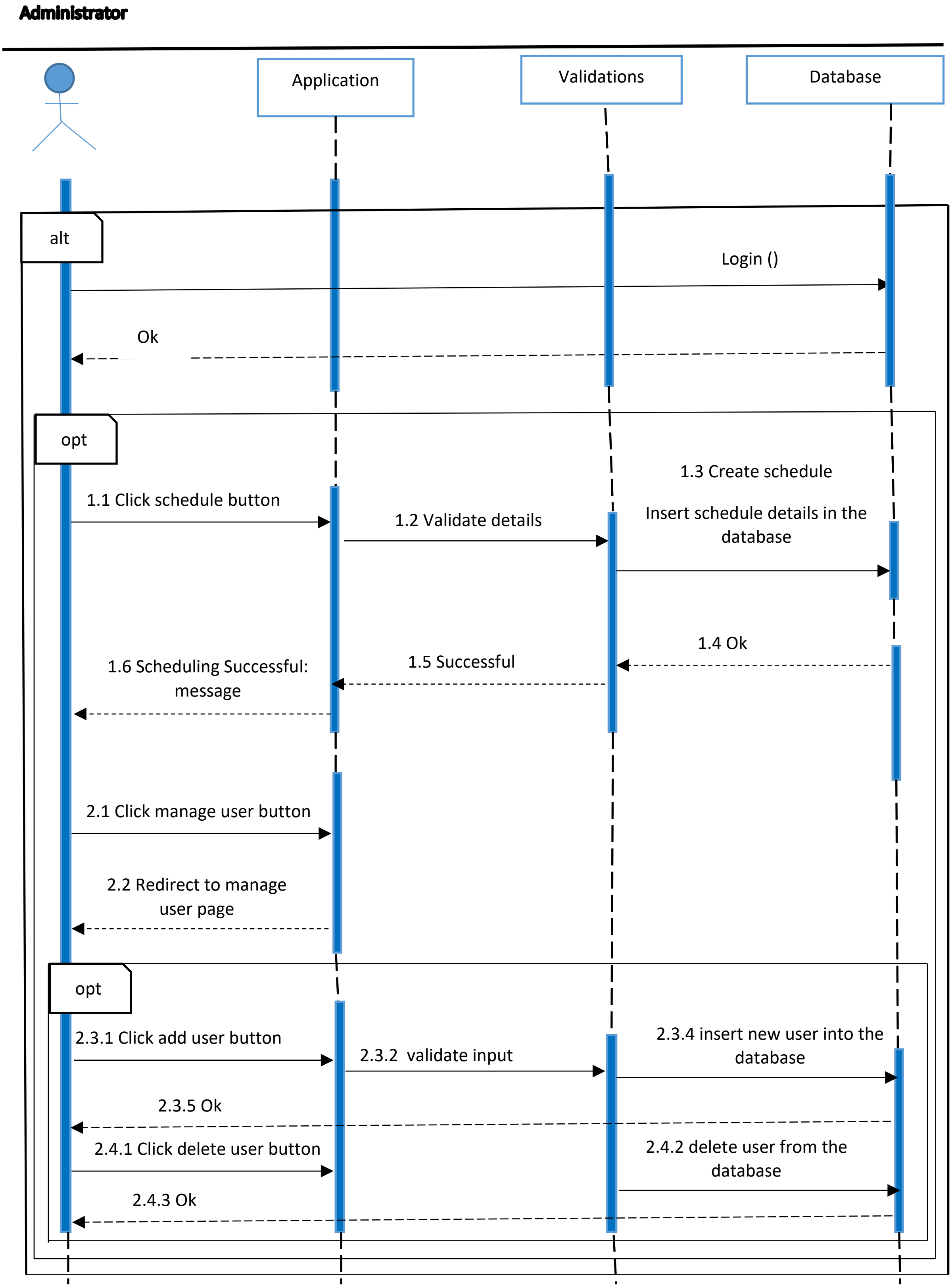

Administrator
Application
Validations
Database
alt
Login ()
Ok
opt
1.1 Click schedule button
1.2 Validate details
1.3 Create schedule
Insert schedule details in the database
1.4 Ok
1.5 Successful
1.6 Scheduling Successful: message
2.1 Click manage user button
2.2 Redirect to manage user page
opt
2.3.1 Click add user button
2.3.2 validate input
2.3.4 insert new user into the database
2.3.5 Ok
2.4.1 Click delete user button
2.4.2 delete user from the database
2.4.3 Ok

Figure 3.9: sequence diagram of an administrator's interactions aside login and registration

3.5.4 Standard user (client's) interaction with the system

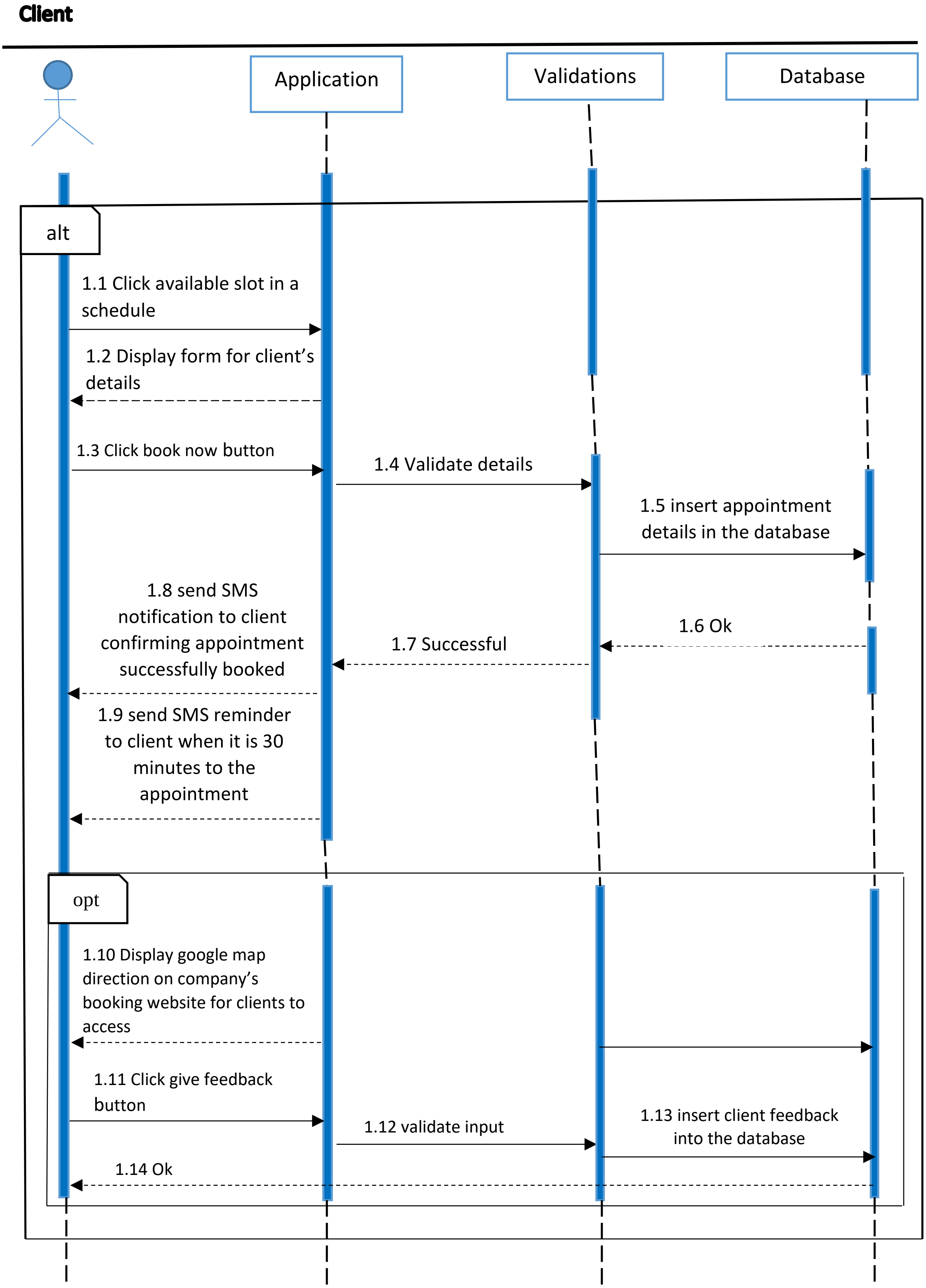

Client
Application
Validations
Database
alt
1.1 Click available slot in a schedule
1.2 Display form for client's details
1.3 Click book now button
1.4 Validate details
1.5 insert appointment details in the database
1.6 Ok
1.7 Successful
1.8 send SMS notification to client confirming appointment successfully booked
1.9 send SMS reminder to client when it is 30 minutes to the appointment
opt
1.10 Display google map direction on company's booking website for clients to access
1.11 Click give feedback button
1.12 validate input
1.13 insert client feedback into the database
1.14 Ok

Figure 3.10: sequence diagram of a standard user interaction with the system

3.6 Traceability and Requirements Matrix

In this section, the major components of the system are mapped to user perspectives drawn from the requirements analysis in chapter 2 section 2.1.2. In the traceability and requirement matrix, the requirements are represented on the vertical axis whilst the components are represented on the horizontal axis. "X" is used to map a requirement to its component(s).

Table 3. 1: Requirement and traceability matrix

Requirements	**Components**			
	Scheduling	**Booking**	**System Configuration**	**Report Generation**
Book Appointment	X	X		
Customization			X	
Statistical analysis of bookings	X	X		X
Easy to use interface			X	
Customer feedback	X	X		X
Booking, cancellation and rebooking of appointments	X	X		X
Appointment reminder	X	X		
Live and video chats	X	X		X
Calendar generation	X	X		
Directions to venue	X	X		X

3.7 Technologies Used

3.7.1 Integrated Development Environment (IDEs)

The IDE used to edit, compile and debug code was Sublime Text. Sublime Text is a cross platform text editor for code, prose and markup. It has a Python API. It supports many programming and markup languages and prose such as AppleScript, ActionScript, ASP, C, C#, HTML, JavaScript, PHP, Java, Perl, Pascal, C++, CSS, Clojure, OCaml and many others.

3.7.2 Programming Languages

- PHP: PHP which stands for Hypertext Processor is a server side scripting language suitable for web application development. PHP code is executed on the server and the result is displayed on a web page as HTML. In this project, PHP is used for all database and other server side interactions. PHP is used instead of other server side scripting languages because it is fast, simple and offers many other professional features for programmers.
- JavaScript: JavaScript is an object-oriented programming language used to make webpages interactive. JavaScript is used in this project for responses when a button is click and other user interactive events.
- HTML: HTML stands for Hypertext Markup Language. In this application, HTML is used to create web pages and various user interfaces.
- SQL: SQL stands for Structured Query Language. SQL is used in this project to manipulate the database. Specific SQL commands used in this project are insert, update, delete, select and retrieve data.
- jQuery: This is a portable JavaScript library used in this project for event handling. Specifically, it is used to call and manipulate the jQuery full calendar plugin for scheduling an event.

3.7.3 Development Platforms

- GitHub: This development platform is used in this project to host and review code. Written codes are push and pulled from GitHub.

- XAMPP: XAMPP is a cross platform web server. XAMPP stands for Cross-Platform (X), Apache (A), MySQL (M), PHP (P) and Perl (P). In this project, XAMPP is used to create a local http server for development testing.

3.7.4 Frameworks and Templates

- Superlist - directory template: This template was designed by the Aviators (2015). It makes use of enhanced HTML 5 and Bootstrap 3.x which is an open source framework with HTML, CSS and JavaScript. This template was used for responsive front-end designs of this application.

3.7.5 APIs

- Google charts API: This API is used in this project to create graphics with data from the database. It is used to draw PNG images of pie and bar charts indicating summaries of clients' bookings and feedback on the administrator's dashboard.
- PHP mail: PHP mail is an emailing service used in this application to enable users of "zer0Clock" to contact its developers during system failures or to make suggestion that can improve user experience. It is also used to notify users when a password reset request is made and when a password has successfully been reset. Moreover, it used to send clients who have booked and attended an appointment a link to the feedback page.
- Web RTC: RTC stands for Real-Time Communications. It is used in this software to enable clients of firms and individuals to communicate via live chats or video calls from home. Insights and JavaScript configuration files written by Sanni (2016) were used to design this component.

- Hubtel SMS API: Hubtel messaging API makes it possible to send and receive SMS. In this project, this API is used to send SMS to clients when an appointment has been booked successfully and when it is 30 minutes to the appointment.

3.7.6 Plugin

- jQuery full calendar plugin: The jQuery full calendar plugin is used in this project to create a calendar for events. It shows the calendar view for the month, week and day level. This plugin creates an easy to use calendar for users. It has features which makes it possible to drag an appointment to a different time or day to reschedule that appointment. A tutorial by Webslesson (2017) provided major insights into the development of this component.
- Add to calendar plugin: An add to calendar button interacts with this plugin from Eventable (2017). This plugin allow users who have booked an appointment to add the event to their calendar. This plugin allow integration with Google calendar, Apple calendar, Outlook calendar, Outlook desktop calendar, Yahoo calendar and other calendars which gives users flexibility.

3.7.7 Utility

The software utility used in this application is **cron job**. This is a time based scheduler for automation. In this project, cron job is used to run the PHP scripts for SMS notifications. Cron job made it possible to automatically send an SMS notification to a client when it is 30 minutes to their appointment. Again, with cron job, a link to the feedback page is automatically sent to a client after they have attended an appointment.

Chapter 4: Implementation

4.0 Chapter Overview

This chapter outlines the implementation techniques employed in developing this application. It also describes the tools, APIs, libraries and frameworks used in developing this software package.

4.1 Approach and Implementation Limitations and Techniques

The approach employed in developing this application is web-based. This application is an online application that can be accessed by any device with a browser and internet connectivity. The reasons for choosing a web-based approach to develop this application and not the other approaches are outlines below:

- Platform independence. A web-based application runs inside a web browsers hence does not have to be installed on a device. This makes it possible to use a web-application on any computer and smart devices regardless of the platform.
- Cost effective development: Web-based applications are accessed through the internet which is a uniform environment. As such, a web-based application is developed for a single operating system but tested on different web browsers such as Google Chrome, Firefox, and Internet Explorer, among many others.
- Easily customizable. Web-based application interfaces can be modified to fit different users with distinct needs and expertise. For this reason, this approach is advantageous over desktop applications because desktop applications have fixed user interfaces. This feature is much important to the development of this application because users stated that they wanted a customizable booking website since this application is generic. Thus, the web-based approached was an appropriate choice.

- User independence and choice. Web-based applications can be accessed anywhere at any time by a user on any device connected to the internet and with a browser. Unlike a traditional or desktop application, the user is in charge of where and when they use the application, and this give users full control and independence.

4.1.2 Implementation Limitations

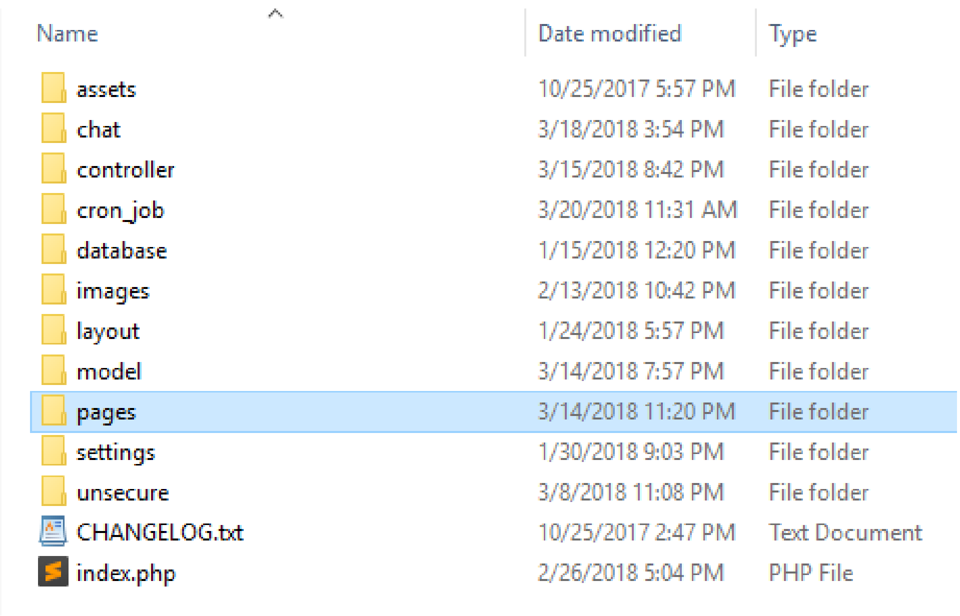
One major constraint to the implementation of this application is the software design process. The software design process adopted in designing this application is the agile method of development. The agile method necessitates users' involvement which ensures that what is needed by the user will be developed. Again, this method embrace change. This is important because we are in a changing business environment thus, requirements are subject to change. Moreover, the agile method of software development focuses on simplicity in both the software being developed and the development process. These will ensure that the software package developed will be useful.

4.1.3 Implementation techniques

A step-by-step technique was adopted in developing this application. The entire components of the application were broken down into sub-components and developed in stages. The first stage of development was the design of the various user interfaces. Next, all interfaces with forms were validated.

Following the agile method of development, users were involved during the implementation. The most important components; scheduling and booking were developed first. The feedback from user testing were incorporated in the development process. Again, new requests from users were analyzed through the requirements engineering and specification process and components were designed accordingly to address these new requirements. An example of a new requirement that was added by users is the calendar generation feature. The components were divided into sub-components during implementation to ensure easy development.

Again, the MVC (Model View Controller) model was used to divide this application into 3 interconnected parts. This made it possible to reuse code and practice parallel development. The diagram below shows the folder structure and files of this system:



Name	Date modified	Type
assets	10/25/2017 5:57 PM	File folder
chat	3/18/2018 3:54 PM	File folder
controller	3/15/2018 8:42 PM	File folder
cron_job	3/20/2018 11:31 AM	File folder
database	1/15/2018 12:20 PM	File folder
images	2/13/2018 10:42 PM	File folder
layout	1/24/2018 5:57 PM	File folder
model	3/14/2018 7:57 PM	File folder
pages	3/14/2018 11:20 PM	File folder
settings	1/30/2018 9:03 PM	File folder
unsecure	3/8/2018 11:08 PM	File folder
CHANGELOG.txt	10/25/2017 2:47 PM	Text Document
index.php	2/26/2018 5:04 PM	PHP File

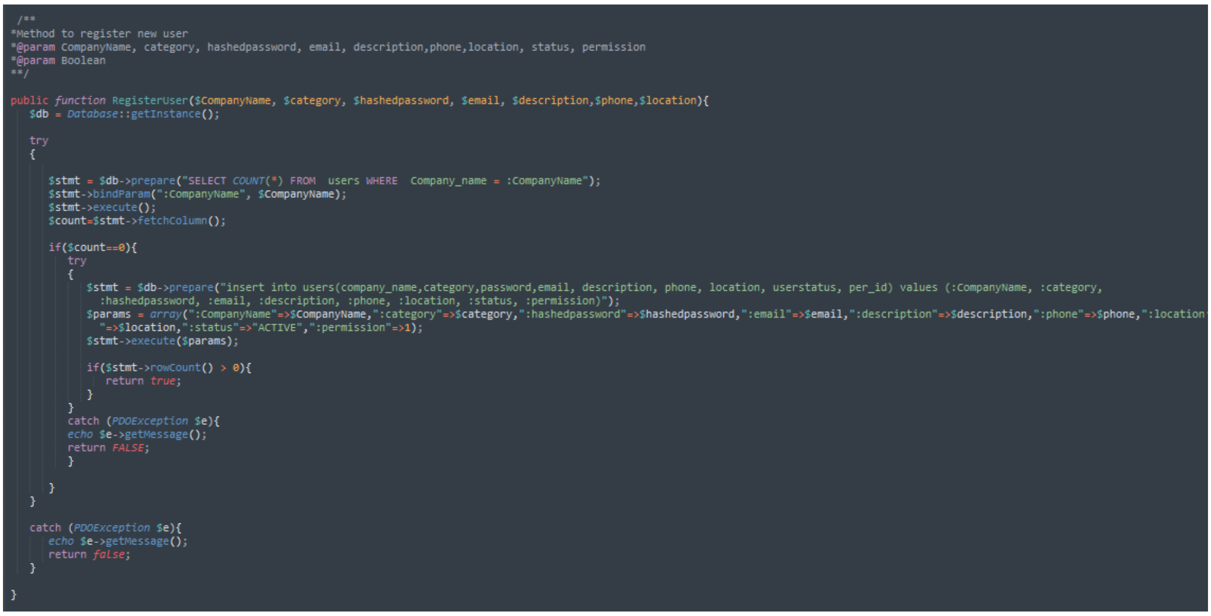
Figure 4.1 folder structure: MVC

4.2 Description of Components

As indicated in section 4.1.3, all components of the system were broken down into sub sections for easier development. The major and sub-components are described below:

- Scheduling: This major component is concern with all steps a firm must follow to be able to put an event for booking. The sub-components under this major functionality includes:
    - Registration: All firms that wishes to use this application must register. This component ensures that new users (administrative users) register and become

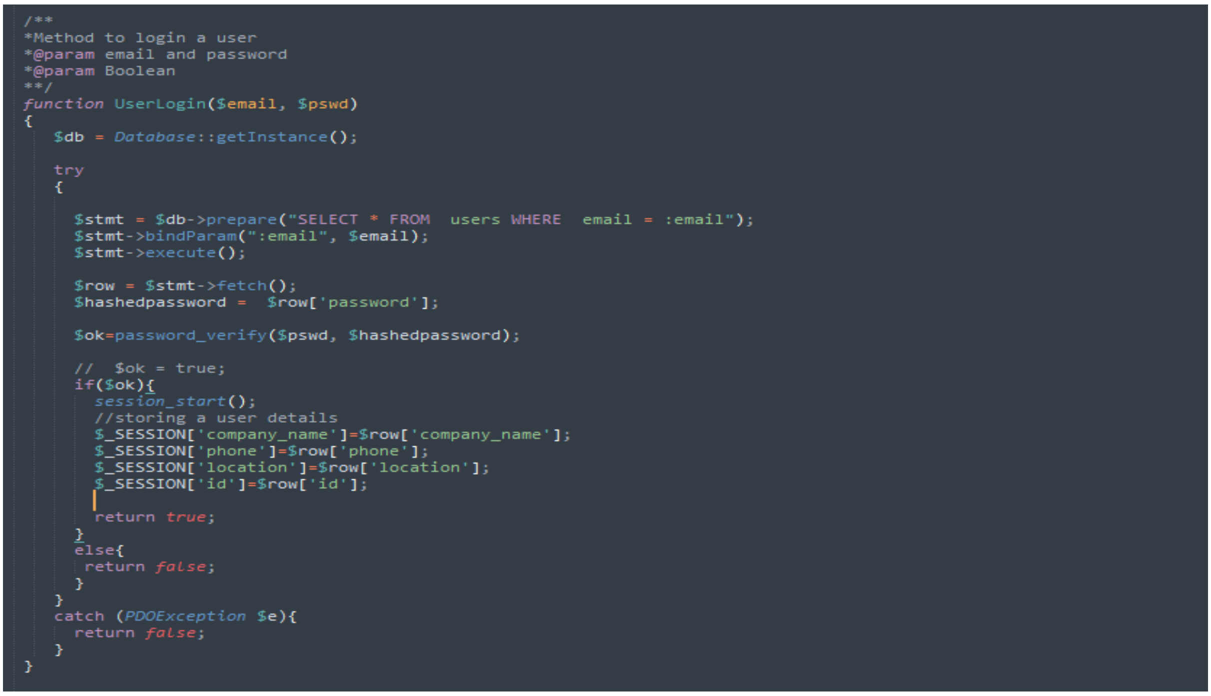
registered users of “zer0Clock”. The figure below shows a snippet of code for registration:



```
 /**
*Method to register new user
*@param CompanyName, category, hashedpassword, email, description,phone,location, status, permission
*@param Boolean
**/

public function RegisterUser($CompanyName, $category, $hashedpassword, $email, $description,$phone,$location){
   $db = Database::getInstance();

   try
   {

      $stmt = $db->prepare("SELECT COUNT(*) FROM  users WHERE  Company_name = :CompanyName");
      $stmt->bindParam(":CompanyName", $CompanyName);
      $stmt->execute();
      $count=$stmt->fetchColumn();

      if($count==0){
         try
         {
            $stmt = $db->prepare("insert into users(company_name,category,password,email, description, phone, location, userstatus, per_id) values (:CompanyName, :category,
              :hashedpassword, :email, :description, :phone, :location, :status, :permission)");
            $params = array(":CompanyName"=>$CompanyName,":category"=>$category,":hashedpassword"=>$hashedpassword,":email"=>$email,":description"=>$description,":phone"=>$phone,":location
              "=>$location,":status"=>"ACTIVE",":permission"=>1);
            $stmt->execute($params);

            if($stmt->rowCount() > 0){
               return true;
            }
         }
         catch (PDOException $e){
         echo $e->getMessage();
         return FALSE;
         }

      }
   }

   catch (PDOException $e){
      echo $e->getMessage();
      return false;
   }

}
```

Figure 4.2 Snippet of code for registration

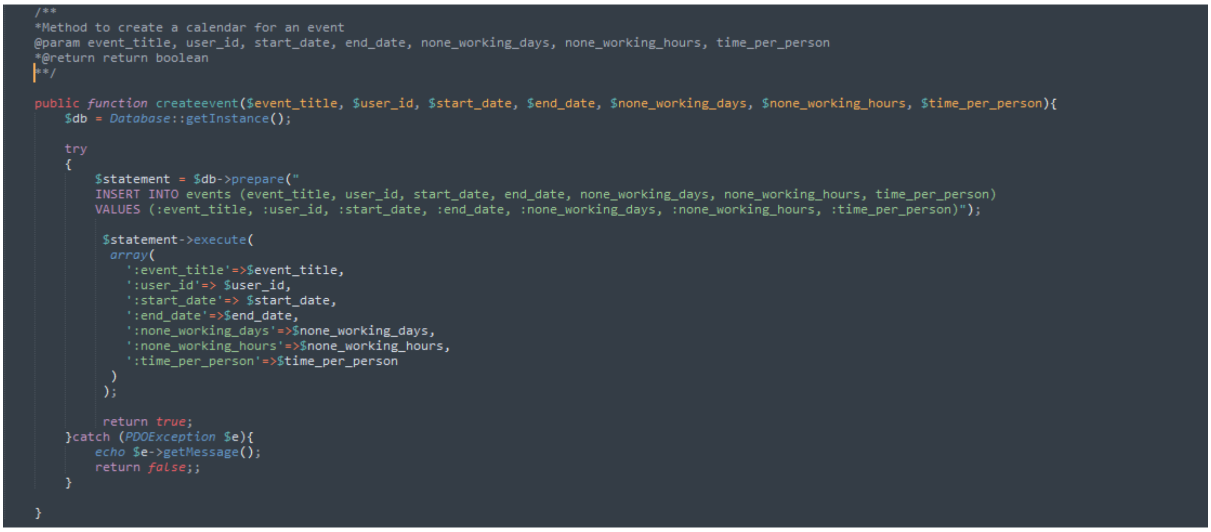
- ○ Login: This component allows registered users to login and have full access to the functionalities of the application. The diagram below shows a snippet of code for login:



```
/**
*Method to login a user
*@param email and password
*@param Boolean
**/
function UserLogin($email, $pswd)
{
   $db = Database::getInstance();

   try
   {

      $stmt = $db->prepare("SELECT * FROM  users WHERE  email = :email");
      $stmt->bindParam(":email", $email);
      $stmt->execute();

      $row = $stmt->fetch();
      $hashedpassword =  $row['password'];

      $ok=password_verify($pswd, $hashedpassword);

      //  $ok = true;
      if($ok){
         session_start();
         //storing a user details
         $_SESSION['company_name']=$row['company_name'];
         $_SESSION['phone']=$row['phone'];
         $_SESSION['location']=$row['location'];
         $_SESSION['id']=$row['id'];

         return true;
      }
      else{
       return false;
      }
   }
   catch (PDOException $e){
      return false;
   }
}
```

Figure 4.3 Snippet of code for login

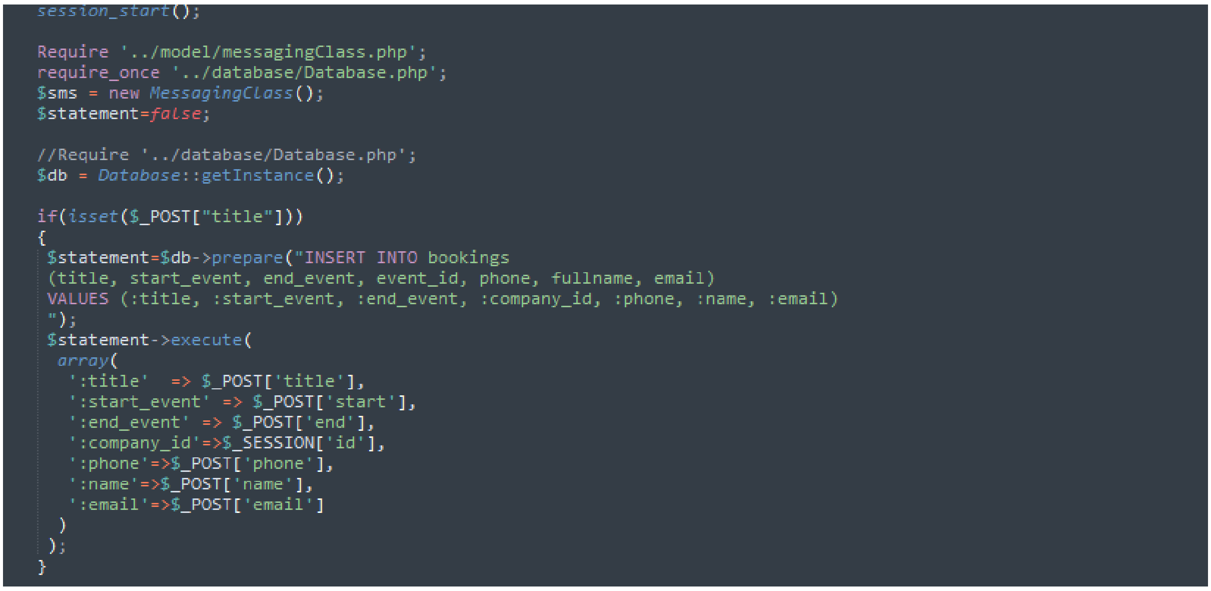
- Schedule specification: Once a user is logged in, he/she can put up their schedule by specifying the activity or event name, time required to attend to a query, working hours and break times. With this information, a schedule will be created and displayed on the firm’s booking website. The diagram below shows a snippet of code for creating a calendar for an event:



```
/**
*Method to create a calendar for an event
@param event_title, user_id, start_date, end_date, none_working_days, none_working_hours, time_per_person
*@return return boolean
**/

public function createevent($event_title, $user_id, $start_date, $end_date, $none_working_days, $none_working_hours, $time_per_person){
    $db = Database::getInstance();

    try
    {
        $statement = $db->prepare("
        INSERT INTO events (event_title, user_id, start_date, end_date, none_working_days, none_working_hours, time_per_person)
        VALUES (:event_title, :user_id, :start_date, :end_date, :none_working_days, :none_working_hours, :time_per_person)");

         $statement->execute(
          array(
            ':event_title'=>$event_title,
            ':user_id'=> $user_id,
            ':start_date'=> $start_date,
            ':end_date'=>$end_date,
            ':none_working_days'=>$none_working_days,
            ':none_working_hours'=>$none_working_hours,
            ':time_per_person'=>$time_per_person
          )
         );

         return true;
    }catch (PDOException $e){
        echo $e->getMessage();
        return false;;
    }

}
```

Figure 4.4 Snippet of code for schedule specification

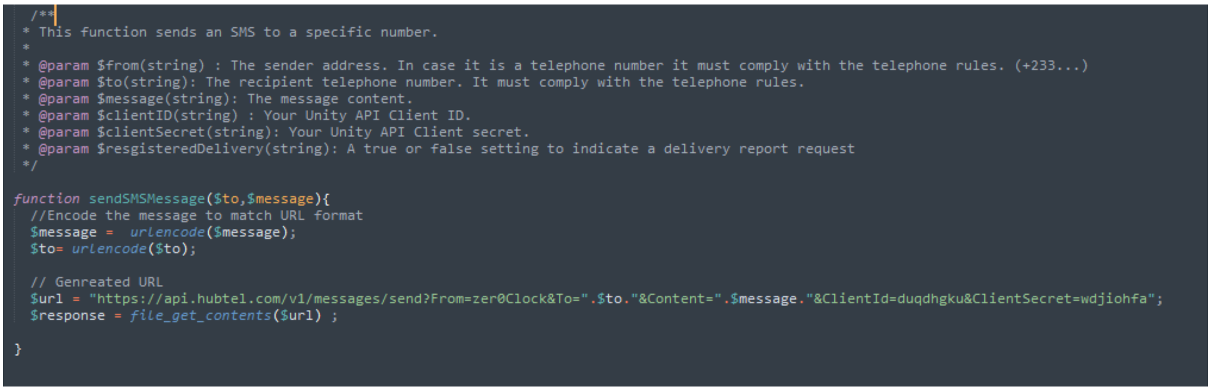
- Booking: This component deals with all the events involved in booking and managing an appointment. The sub events or components involved in booking appointment are:
    - Pick a slot: This component enables clients to visit a firm's booking website and view a schedule. On the schedule, a client can pick a suitable slot that has not been already booked by another client through a double click on that slot. The figure below is a sample for picking a slot: booking an appointment:



```php
session_start();

Require '../model/messagingClass.php';
require_once '../database/Database.php';
$sms = new MessagingClass();
$statement=false;

//Require '../database/Database.php';
$db = Database::getInstance();

if(isset($_POST["title"]))
{
 $statement=$db->prepare("INSERT INTO bookings
 (title, start_event, end_event, event_id, phone, fullname, email)
 VALUES (:title, :start_event, :end_event, :company_id, :phone, :name, :email)
 ");
 $statement->execute(
  array(
   ':title'  => $_POST['title'],
   ':start_event' => $_POST['start'],
   ':end_event' => $_POST['end'],
   ':company_id'=>$_SESSION['id'],
   ':phone'=>$_POST['phone'],
   ':name'=>$_POST['name'],
   ':email'=>$_POST['email']
  )
 );
}
```

Figure 4.5 Snippet of code for booking an appointment

- SMS notification: This component deals with notifying a client via an SMS when an appointment has successfully been booked and when it is 30 minutes to the appointment. The figure below shows a snippet of code for sending SMS notifications:

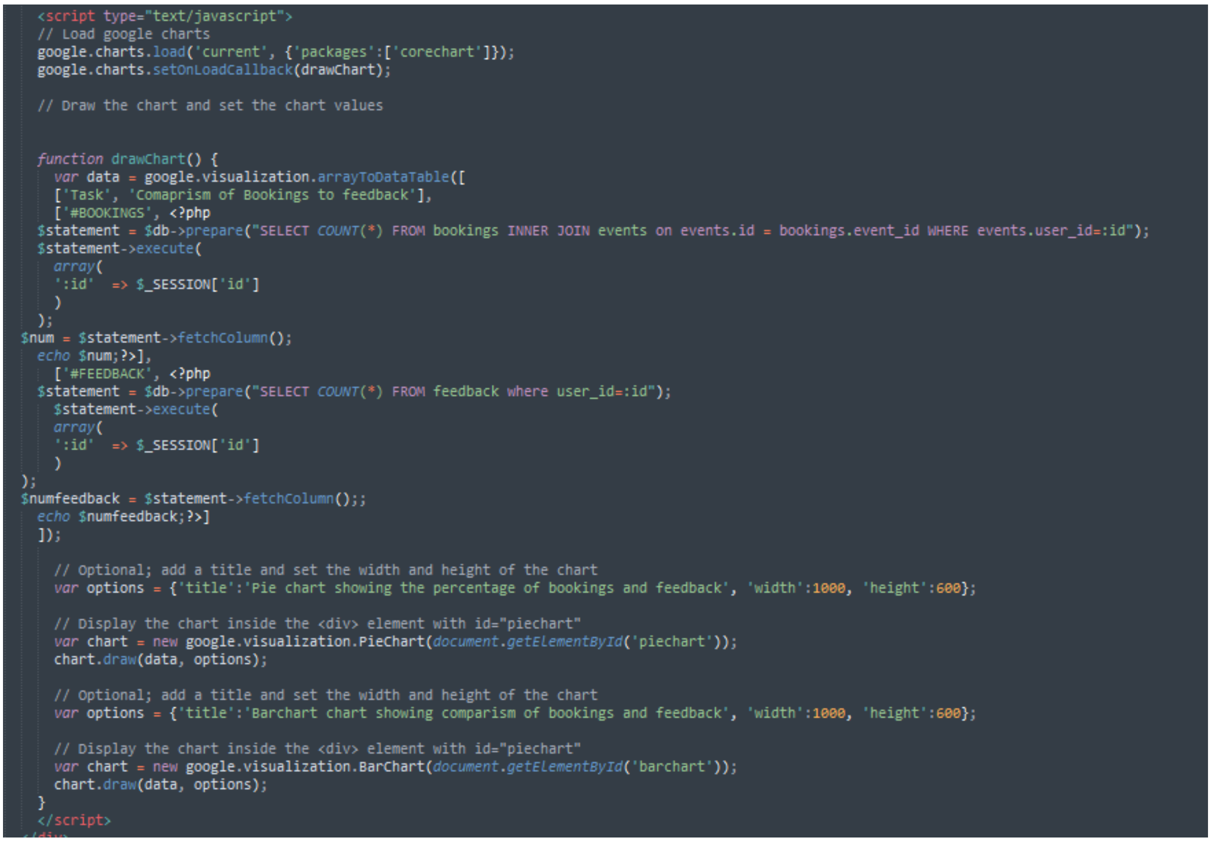


```php
  /**
 * This function sends an SMS to a specific number.
 *
 * @param $from(string) : The sender address. In case it is a telephone number it must comply with the telephone rules. (+233...)
 * @param $to(string): The recipient telephone number. It must comply with the telephone rules.
 * @param $message(string): The message content.
 * @param $clientID(string) : Your Unity API Client ID.
 * @param $clientSecret(string): Your Unity API Client secret.
 * @param $resgisteredDelivery(string): A true or false setting to indicate a delivery report request
 */

function sendSMSMessage($to,$message){
  //Encode the message to match URL format
  $message =  urlencode($message);
  $to= urlencode($to);

  // Genreated URL
  $url = "https://api.hubtel.com/v1/messages/send?From=zer0Clock&To=".$to."&Content=".$message."&ClientId=duqdhgku&ClientSecret=wdjiohfa";
  $response = file_get_contents($url) ;

}
```

Figure 4.6 Snippet of code for sending an SMS notification

- Google map direction: This component provides users who have booked an appointment access to a google map direction to the venue where the appointment is taking place.

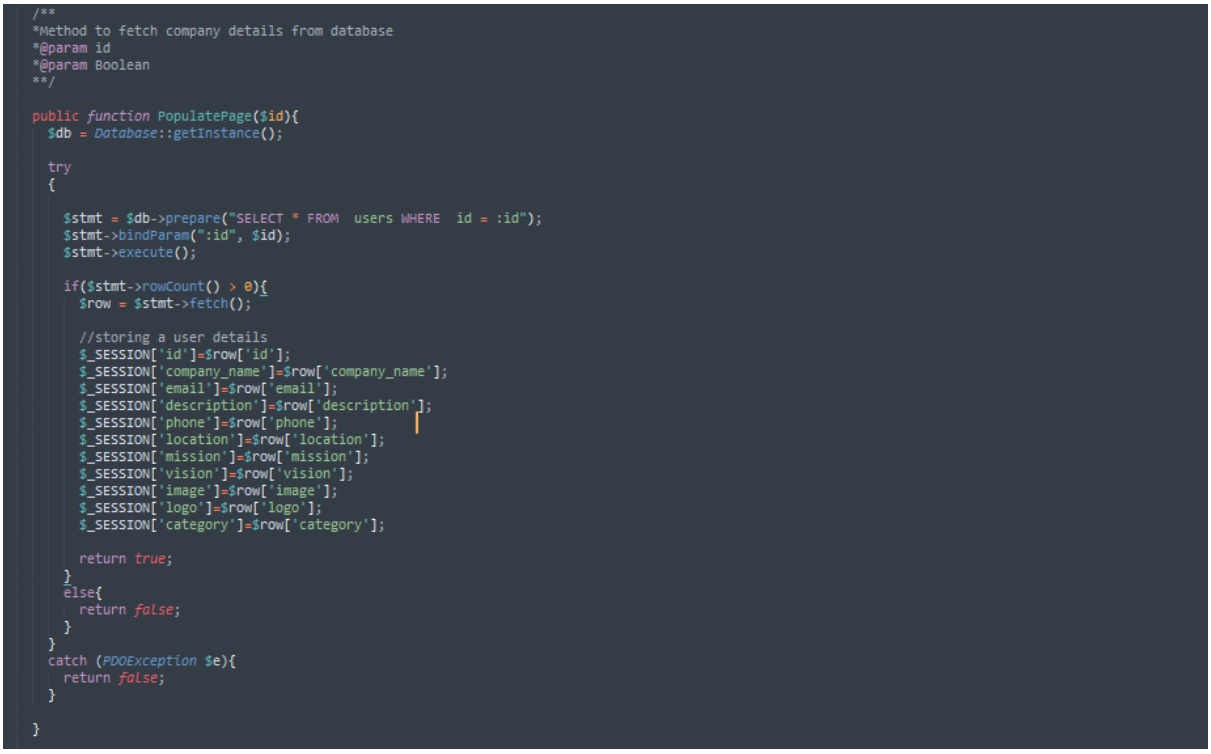
- Feedback: After a client has attended an appointment, this sub-component enables clients to provide feedback about the services rendered for firms to further improve their services.
- Report generation: This component provides a comprehensive dashboard which shows summaries of clients' bookings as well as their feedback. This statistical analysis of bookings and clients' feedback are shown in pie and bar charts and tables. The figure below shows a snippet of code for report generation:



```
  <script type="text/javascript">
  // Load google charts
  google.charts.load('current', {'packages':['corechart']});
  google.charts.setOnLoadCallback(drawChart);

  // Draw the chart and set the chart values


  function drawChart() {
    var data = google.visualization.arrayToDataTable([
    ['Task', 'Comaprism of Bookings to feedback'],
    ['#BOOKINGS', <?php
  $statement = $db->prepare("SELECT COUNT(*) FROM bookings INNER JOIN events on events.id = bookings.event_id WHERE events.user_id=:id");
  $statement->execute(
    array(
    ':id'  => $_SESSION['id']
    )
  );
$num = $statement->fetchColumn();
  echo $num;?>],
    ['#FEEDBACK', <?php
  $statement = $db->prepare("SELECT COUNT(*) FROM feedback where user_id=:id");
    $statement->execute(
    array(
    ':id'  => $_SESSION['id']
    )
);
$numfeedback = $statement->fetchColumn();;
  echo $numfeedback;?>]
  ]);

    // Optional; add a title and set the width and height of the chart
    var options = {'title':'Pie chart showing the percentage of bookings and feedback', 'width':1000, 'height':600};

    // Display the chart inside the <div> element with id="piechart"
    var chart = new google.visualization.PieChart(document.getElementById('piechart'));
    chart.draw(data, options);

    // Optional; add a title and set the width and height of the chart
    var options = {'title':'Barchart chart showing comparism of bookings and feedback', 'width':1000, 'height':600};

    // Display the chart inside the <div> element with id="piechart"
    var chart = new google.visualization.BarChart(document.getElementById('barchart'));
    chart.draw(data, options);
  }
  </script>
```

Figure 4.7 snippet of code for report generation

- System configuration: This component handles configuring and creating a customizable booking website for administrative users. The figure below shows a snippet for customizing a company website:



```
/**
*Method to fetch company details from database
*@param id
*@param Boolean
**/

public function PopulatePage($id){
  $db = Database::getInstance();

  try
  {

    $stmt = $db->prepare("SELECT * FROM  users WHERE  id = :id");
    $stmt->bindParam(":id", $id);
    $stmt->execute();

    if($stmt->rowCount() > 0){
      $row = $stmt->fetch();

      //storing a user details
      $_SESSION['id']=$row['id'];
      $_SESSION['company_name']=$row['company_name'];
      $_SESSION['email']=$row['email'];
      $_SESSION['description']=$row['description'];
      $_SESSION['phone']=$row['phone'];
      $_SESSION['location']=$row['location'];
      $_SESSION['mission']=$row['mission'];
      $_SESSION['vision']=$row['vision'];
      $_SESSION['image']=$row['image'];
      $_SESSION['logo']=$row['logo'];
      $_SESSION['category']=$row['category'];

      return true;
    }
    else{
      return false;
    }
  }
  catch (PDOException $e){
    return false;
  }

}
```

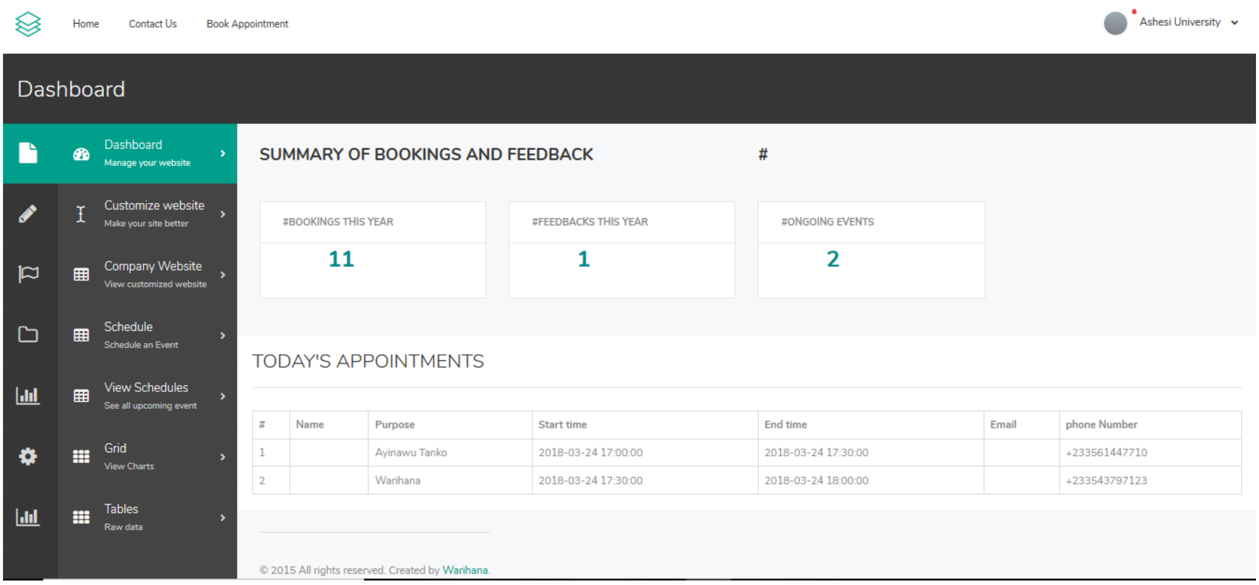
Figure 4.8 Snippet of code for customizing a company website

4.4 Evidence of Implementation

This section shows screenshots of the various pages implemented. The screenshots are categorized according to the various users; administrative and standard users.

4.4.1 Administrative users' views

The dashboard of this application is designed for administrative users. As such, the screenshots shown in this section are views of the administrators' dashboard.



SUMMARY OF BOOKINGS AND FEEDBACK #

#BOOKINGS THIS YEAR 11

#FEEDBACKS THIS YEAR 1

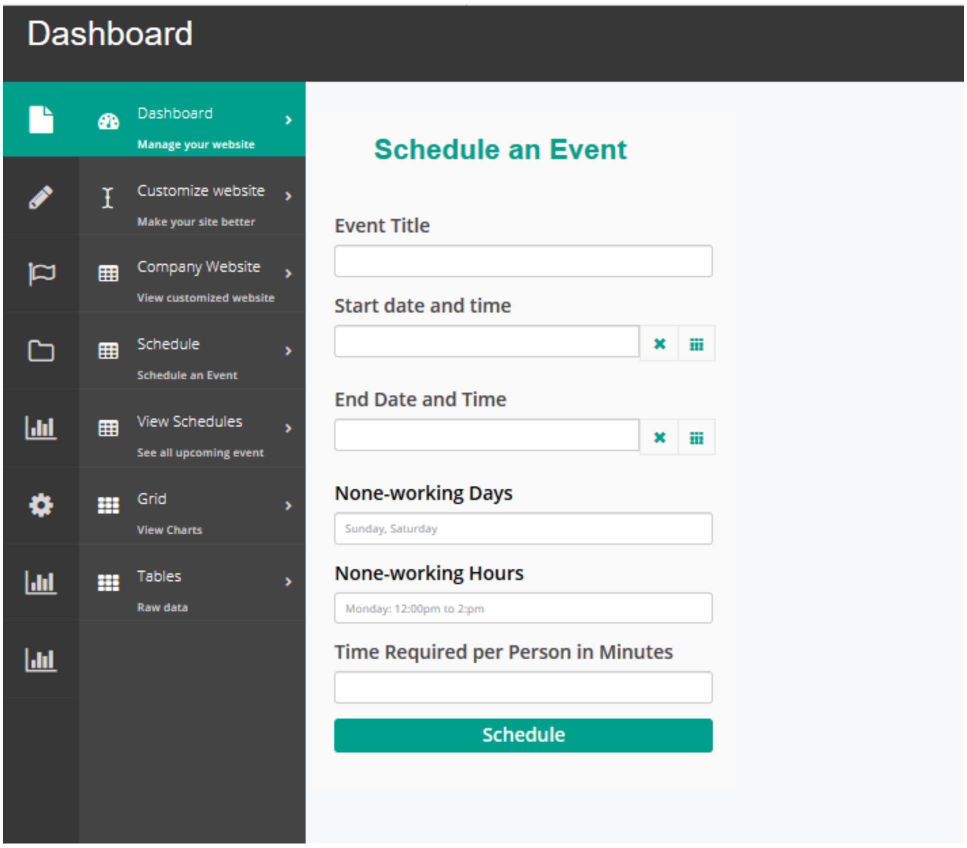
#ONGOING EVENTS 2

TODAY'S APPOINTMENTS

#	Name	Purpose	Start time	End time	Email	phone Number
1		Ayinawu Tanko	2018-03-24 17:00:00	2018-03-24 17:30:00		+233561447710
2		Warihana	2018-03-24 17:30:00	2018-03-24 18:00:00		+233543797123

© 2015 All rights reserved. Created by Warihana.

Figure 4.9: Interface of the main administrators' dashboard



Schedule an Event

Event Title

Start date and time

End Date and Time

None-working Days

Sunday, Saturday

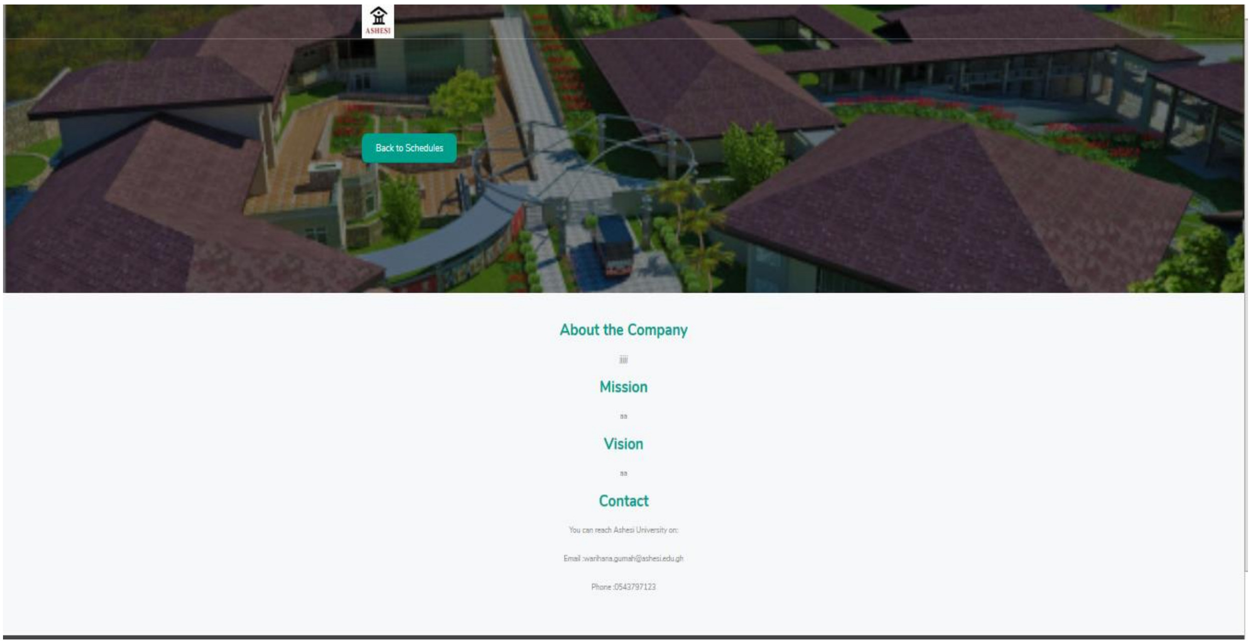
None-working Hours

Monday: 12:00pm to 2:pm

Time Required per Person in Minutes

Schedule

Figure 4.10: Interface to schedule and event



About the Company

Mission

Vision

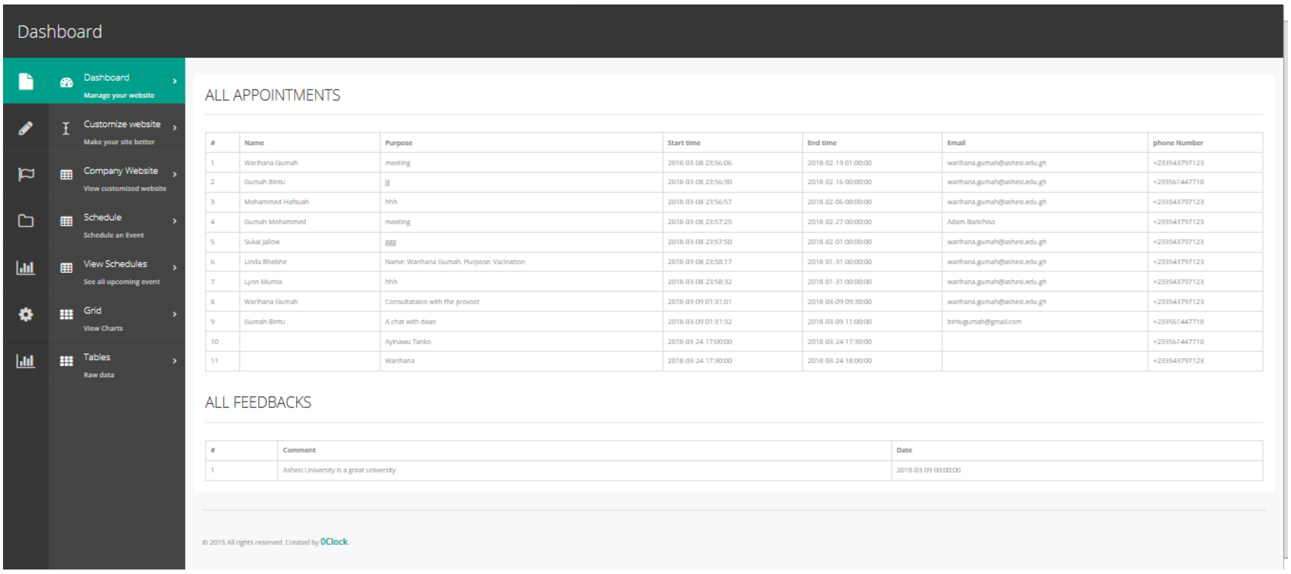
Contact

You can reach Ashesi University on:

Email :warihana.gumah@ashesi.edu.gh

Phone :0543797123

Figure 4.11: View of company customizable website



ALL APPOINTMENTS

#	Name	Purpose	Start time	End time	Email	phone Number
1	Warihana Gumah	meeting	2018-03-08 23:56:06	2018-02-19 01:00:00	warihana.gumah@ashesi.edu.gh	+233543797123
2	Gumah Bintu	jjj	2018-03-08 23:56:30	2018-02-16 00:00:00	warihana.gumah@ashesi.edu.gh	+233561447710
3	Mohammed Hafsuah	hhh	2018-03-08 23:56:57	2018-02-06 00:00:00	warihana.gumah@ashesi.edu.gh	+233543797123
4	Gumah Mohammed	meeting	2018-03-08 23:57:25	2018-02-27 00:00:00	Adam Barichiso	+233543797123
5	Sukai Jallow	ggg	2018-03-08 23:57:50	2018-02-01 00:00:00	warihana.gumah@ashesi.edu.gh	+233543797123
6	Linda Bhebhe	Name: Warihana Gumah. Purpose: Vacination	2018-03-08 23:58:17	2018-01-31 00:00:00	warihana.gumah@ashesi.edu.gh	+233543797123
7	Lynn Mumia	hhh	2018-03-08 23:58:32	2018-01-31 00:00:00	warihana.gumah@ashesi.edu.gh	+233543797123
8	Warihana Gumah	Consultataion with the provost	2018-03-09 01:31:01	2018-03-09 09:30:00	warihana.gumah@ashesi.edu.gh	+233543797123
9	Gumah Bintu	A chat with dean	2018-03-09 01:31:32	2018-03-09 11:00:00	bintugumah@gmail.com	+233561447710
10		Ayinawu Tanko	2018-03-24 17:00:00	2018-03-24 17:30:00		+233561447710
11		Warihana	2018-03-24 17:30:00	2018-03-24 18:00:00		+233543797123

ALL FEEDBACKS

#	Comment	Date
1	Ashesi University is a great university	2018-03-09 00:00:00

© 2015 All rights reserved. Created by 0Clock.

Figure 4.12 Interface of tables of raw data; all appointments and feedback

4.4.2 Standard users' views

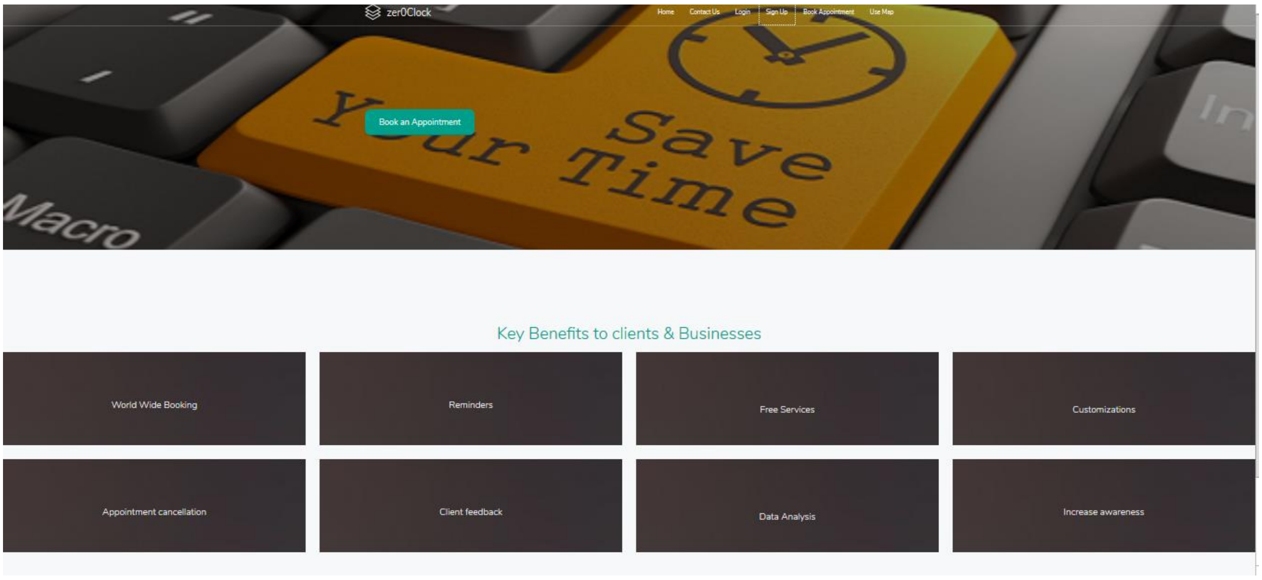


Figure 4.13: View of the home page

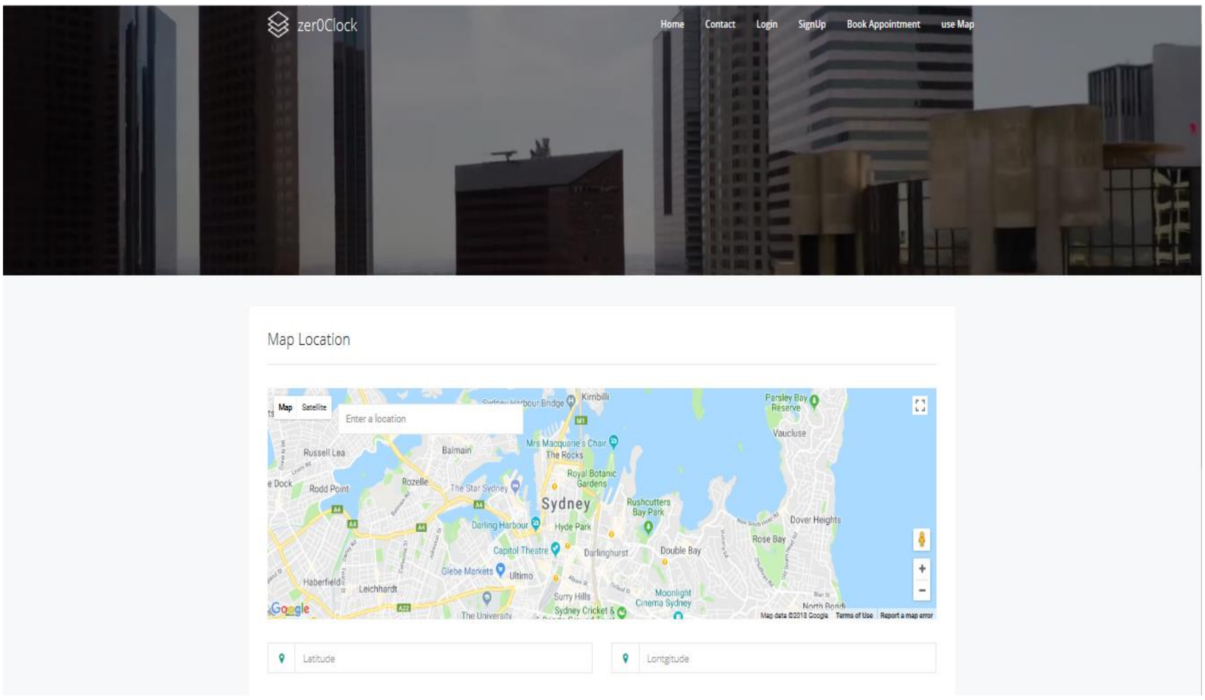
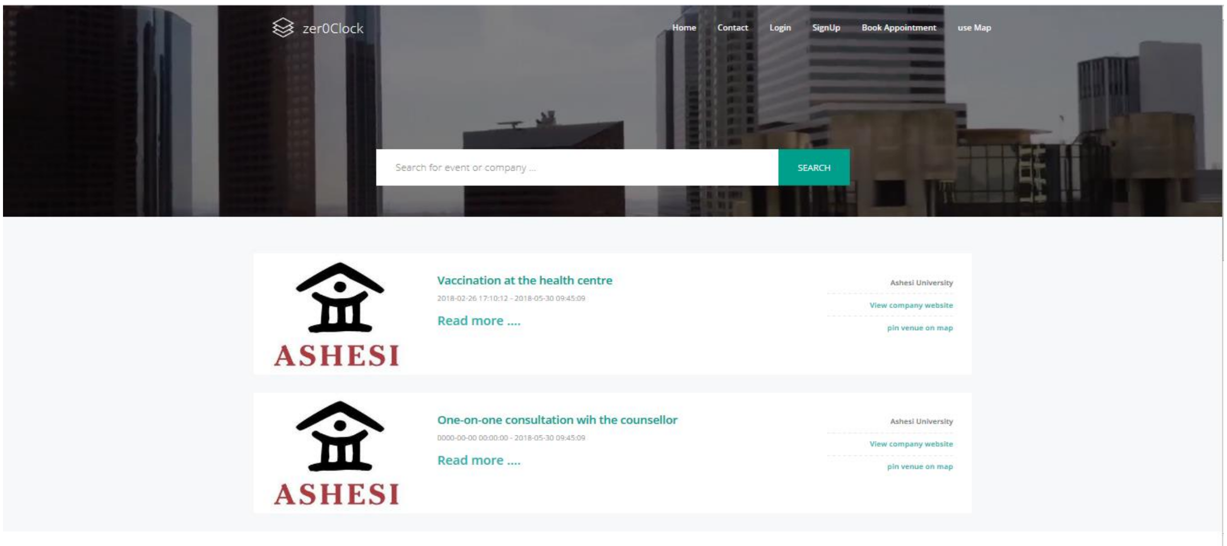


Figure 4.14: Interface of the map page for directions



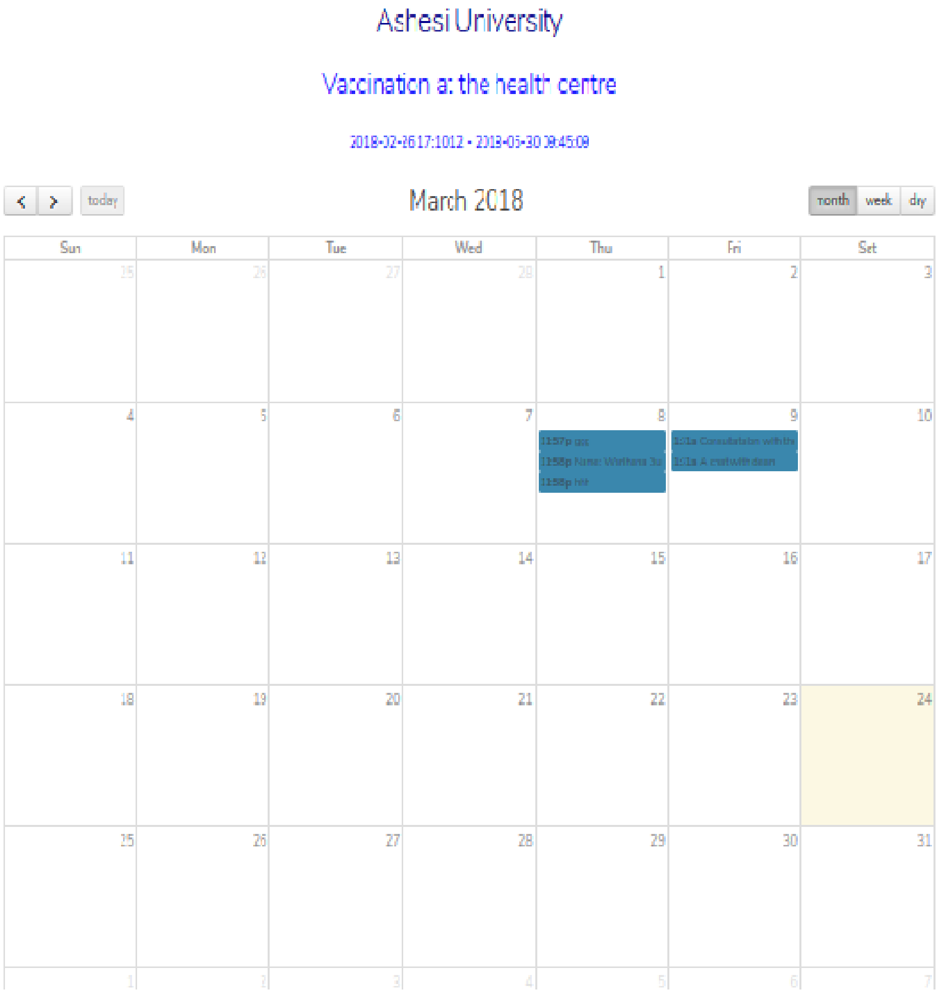
Vaccination at the health centre
2018-02-26 17:10:12 - 2018-05-30 09:45:09
Read more ....

Ashesi University
View company website
pin venue on map

One-on-one consultation wih the counsellor
0000-00-00 00:00:00 - 2018-05-30 09:45:09
Read more ....

Ashesi University
View company website
pin venue on map

Figure 4.15: View of all scheduled events



Ashesi University

Vaccination at the health centre

2018-02-26 17:10:12 - 2018-05-30 09:45:09

March 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	1	2	3
4	5	6	7	8 11:57p qqq 11:58p Nana Wiafe... 3u 11:58p hht	9 1:01a Consultation with the 1:01a A meet with dean	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Figure 4.16: Interface of calendar for an event

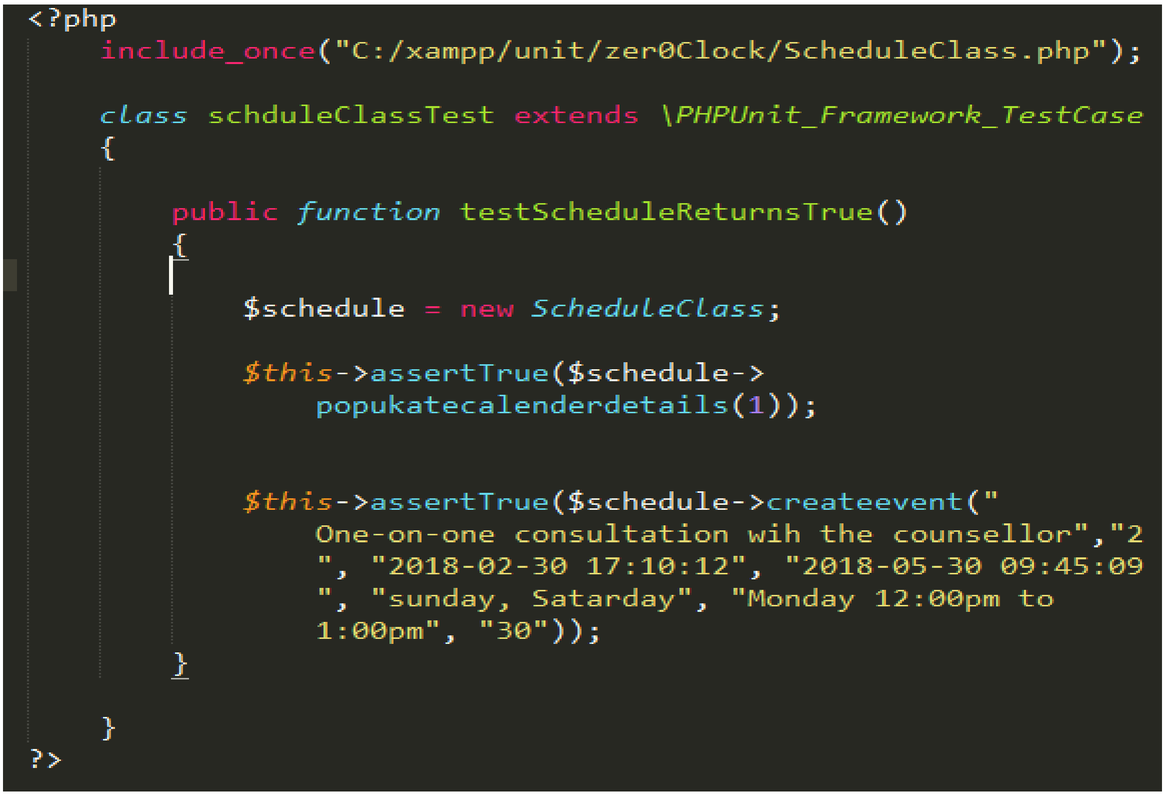
Chapter 5: Test and Results

5.0 Chapter Overview

This chapter elaborates on the testing techniques that were employed to test this application to ensure that it meets all requirements outlined in chapter 2. The testing techniques adopted in testing this application included unit testing, component testing and system testing.

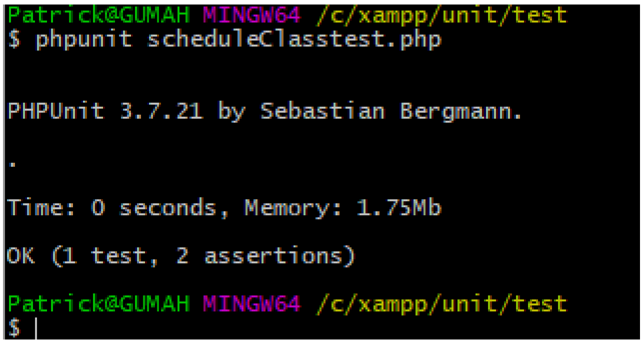
5.1 Unit Testing

Under this testing technique, the PHPUnit testing framework was used to test objects of classes holistically. Functions of the various classes were individually tested to ensure proper operation. The code snippets of the PHPUnit test and a screenshot of the test result for two functions in the schedule class of this application is shown below:



```
<?php
    include_once("C:/xampp/unit/zer0Clock/ScheduleClass.php");

    class schduleClassTest extends \PHPUnit_Framework_TestCase
    {

        public function testScheduleReturnsTrue()
        {

            $schedule = new ScheduleClass;

            $this->assertTrue($schedule->
                popukatecalenderdetails(1));


            $this->assertTrue($schedule->createevent("
                One-on-one consultation wih the counsellor","2
                ", "2018-02-30 17:10:12", "2018-05-30 09:45:09
                ", "sunday, Satarday", "Monday 12:00pm to
                1:00pm", "30"));
        }

    }
?>
```

Figure 5.1: Snippet of PHPUnit test code to test the schedule class.



```
Patrick@GUMAH MINGW64 /c/xampp/unit/test
$ phpunit scheduleClasstest.php

PHPUnit 3.7.21 by Sebastian Bergmann.

.

Time: 0 seconds, Memory: 1.75Mb

OK (1 test, 2 assertions)

Patrick@GUMAH MINGW64 /c/xampp/unit/test
$ |
```

Figure 5.2: PHPUnit test result for the 2 functions in the schedule class

5.2 Component Testing

This section discusses the testing of specific modules of this application. Individual units tested in section 5.1 were integrated to form components. These components were then tested for defects as a result of the integration of units. Below are the test results of selected components of this system:

5.2.1 Test case 1: Create a calendar for an event

- Precondition: An SQL statement that takes event title, start and end date, non-working days, non-working hours and time required to attend to a person.
- Expected Results: Event details are inserted into the database and fetched to populate the calendar for that event.
- Test Results:

Table 5. 1: Test result for creating a schedule for an event

Valid Input	**Result**
• Event title as a String: e.g. Vaccination at Ashesi • Start date as timestamp: e.g. 2018-02-30 06:30:00 • End date as timestamp: e.g. 2018-05-30 18:00:00	Returns result (a valid JSON output) greater than 0.

- Non-working days as a String: e.g. Sunday, Saturday - Non-working hours as a String: e.g. Monday - Friday 12:00pm to 1:00pm - Time required per person as a String: e.g. 30 minutes	Fetch and populate calendar with event details from the database
Invalid Input	**Result**
- No event title provided - No start date provided or an input that is not a time stamp - No end date provided or an input that is not a time stamp - No non-working days provided - No non-working hours provided - No time required provide	Returns a JSON result being 0 (unsuccessful). Event calendar is not created

5.2.2 Test case 2: Booking an appointment

- Precondition: An SQL statement that takes full name, start date, end date, phone number and email.
- Expected Results: Appointment details are inserted into the database and SMS notification sent to the phone number provided
- Test Results:

Table 5. 2: Test result for booking an appointment

Valid Input	**Result**
- Full name as a String: e.g. Warihana Gumah - Start date as timestamp: e.g. 2018-02-30 06:30:00 - End date as timestamp: e.g. 2018-02-30 7:00:00 - Phone number as a String: e.g. +233543797123	Returns result (a valid JSON output) greater than 0. Send SMS notification to client's provided phone

- Email address as a String: e.g. warihanagumah@gmail.com	number to confirm that an appointment has successfully been booked.
Invalid Input	**Result**
- No full name provided - No start date provided or an input that is not a time stamp - No end date provided or an input that is not a time stamp - No phone number provided or invalid phone number - No email address provided or invalid email address	Return a JSON result being 0 (unsuccessful). Appointment is not booked and an SMS is not sent

5.3 System Testing

This application was hosted on 000webhost and tested in its entirety to reveal bugs and evaluate users' interaction and ease of usage. Three major system testing techniques were adopted; usability testing, load testing and functional testing. Details of each of the above testing techniques are discussed below:

- Usability testing: This testing technique focused on the ease of usage of this application. The application had redundant and self-explanatory links which made navigation for users easy. This testing technique was adopted since the early implementation phases to determine expectation of the user. Since the early phase to the final stages of development, approximately 15 users have tested this application in a period of 2 months to determine its usability. Overall feedback from these users indicated that this system is user friendly.

- Load testing: This testing technique is concerned with determining how well this application will perform under real-time loads and pressure. The main focus of this test was measuring the efficiency of queries in relation to response time of performing a functionality such as signup, login, view schedules, schedule an appointment, among many others. At the final stage of development, 11 users were contracted to access this system simultaneously. This test revealed that this application is 80% efficient based on the response time and performance of database queries.
- Functional testing: This testing technique was adopted during the mid-phase of implementation to test for functional completeness. This approach of testing showed that functionalities conform to the requirements outlined in chapter 2. During this testing, users suggested additional functionalities such as feedback and SMS notification automation.

Chapter 6: Conclusion and Recommendation

6.0 Chapter Overview

This chapter outline the areas of further research that can improve the efficiency, functionality and usability of this application. It also summarizes the entire processes undertaken to design this system as well as the challenges encountered

6.1 Recommendations

- Company's individual domain: The decision to display all calendars for upcoming events on a single page was to help increase people's awareness about registered businesses on this platform. This is true for people looking for an industry specific event but has no clue where to start from. With a simple search using industry keyword, a list of upcoming events will be displayed for the users' perusal. Again, when a client of a particular company uses this application to book an appointment, they might see certain interesting upcoming events posted by other companies. These helps to promote and increase public awareness about businesses registered and utilizing this platform. For further work, businesses should be given the chance to choose whether they want to have an independent domain URL that they can easily share with clients to follow for appointment booking. This will give businesses exclusivity and increase usability of this application.
- Mobile application platform: Currently, there is only a web version of this application. The decision to develop for web is because web applications are platform independent. Thus, it was feasible to develop considering the time frame. An area of further research that can improve the usability and functionality of this application is a mobile application platform. For instance, a native application will

give users more convenience and ease of use. This will improve the efficiency of this application.

6.2 Major Challenges

- One major challenge accounted was developing the video and live chat feature. The web RTC API was used to develop this feature. This API required several dependencies and plugins to work. The area of challenge was finding the right dependency and plugins for the various browsers as well as making these work. As such, majority of the code for implementing this feature was gotten from Sanni (2016).
- Another major challenge was implementing the calendar feature. The plugin used to develop this was the jQuery full calendar plugin. The area of challenge was the strict jQuery syntax and format used in implementing this plugin. For this reason, the jQuery format from Webslesson (2017) was used.
- Another major area of challenge was getting users for testing so that their feedback can be incorporated in the development process to design a useful application.

6.3 Conclusion

In conclusion, this project sought to develop zer0Clock: a generic online appointment booking system that aims to reduce the problem of long queues in Ghana. Requirements were gathered from prospective users and the various components were designed to address these requirements. On this application, a registered user can schedule an event and a calendar will be generated for this event. Aside scheduling for an event, an administrative user can also customize their company website as well as view statistical analysis of bookings and feedback. This feedback will help firms to improve their services and better satisfy the needs of their clients.

In addition, a client can book for an appointment, cancel or reschedule an appointment on this calendar. To ensure that clients attend their appointments, reminders are sent to them when it is 30 minutes to the appointment. A google map is also developed on this platform to give directions to clients to unfamiliar venues for appointments. The relevance of this project can be measured by how much it has helped to reduce queuing in Ghana especially at hospitals and public-sector offices.

References

Aviators. (2015). *Superlist - Directory Template*. Retrieved from themeforest: https://themeforest.net/item/superlist-directory-template/12623631

Beinpuo, E. S. (2015, May 7). *Clearing queues at passport office*. Retrieved from Graphic Online:https://www.graphic.com.gh/features/opinion/clearing-queues-at-passport-office.html

Belitsoft. (2016, August 8). *Custom Software Requirements Specification Document Example (International Standard)*. Retrieved from Belitsoft: Software Development Company: https://belitsoft.com/php-development-services/software-requirements-specification-document-example-international-standard

Bonanni, P., Picazo, J. J., & Remy, V. (2015). The intangible benefits of vaccination - what is the true economic value of vaccination? *Journal of Market Access & Health Policy*, 1-6

Eventable. (2017). *Add to Calendar Button*. Retrieved from Eventable: https://add.eventable.com/free-add-to-calendar-button

Ho, C.-j., Lau, H.-S., & Li, J. (1995). Introducing variable-interval appointment scheduling rules in service systems. *International Journal of Operations & Production Management, 15*(6), 59-68

Riley, S. P. (1999). Petty Corruption and Development. *Development in Practice*, 189-193

Sanni, A. (2016, December 27). *Building a Text, Audio and Video Chat Web App Using WebRTC and Ratchet*. Retrieved from Medium: https://medium.com/@amirsanni/building-a-text-audio-and-video-chat-web-app-using-webrtc-and-ratchet-45863446e79a

Shapshak, T. (2016). Forbes Welcome. Forbes.com. Retrieved from http://www.forbes.com/sites/tobyshapshak/2015/09/02/african-internet-capacitygrowth-contin-ues-to-outstrip-the-world-2/#1d0ae4557666

Sommerville, I. (2011). *Software engineering.* Boston: Pearson

Transparency International. (Corruption Perceptions Index in 2016). *2017.* Washington, DC: Transparency International

Webslesson. (2017, 12). *Jquery Fullcalendar Integration with PHP and Mysql*. Retrieved from Webslesson: https://www.webslesson.info/2017/12/jquery-fullcalandar-integration-with-php-and-mysql.html

World Bank. (2017). *Employment to population ratio, 15+, total (%) (modeled ILO estimate).* Washington, D.C: World Bank Group. Retrieved from https://data.worldbank.org/indicator/SL.EMP.TOTL.SP.ZS

World Bank. (2017). *Population 2016.* Washington, D.C: World Bank Group.

Zhou, R., & Soman, D. (2003). Looking Back: Exploring the Psychology of Queuing and the Effect of the Number of People Behind. *Oxford Journals,* 517-530