



ASHESI UNIVERSITY COLLEGE

A SOCIAL NETWORKING SITE FOR ASHESI STUDENTS AND
ALUMNI

APPLIED PROJECT

B.Sc. [Management Information Systems]

Selassie Golloh

2018

ASHESI UNIVERSITY COLLEGE

A Social Networking Site for Ashesi Students and Alumni

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.

Selassie Golloh

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:

ACKNOWLEDGEMENTS

I am giving all the glory to God who has been with me throughout my years in Ashesi University. I am also very grateful to my family for all the support and encouragement from the day I was born. I am equally thankful to all my friends and people I met and interacted with at Ashesi University. Finally, a special thanks to my supervisor Mr David Sampah for his academic guidance, dedication and support on this project.

ABSTRACT

Social networking sites (SNSs) are a 21st century phenomenon which has come to stay. There exist over a hundred social networking sites created for various reasons which support a wide range of interests. With over a thousand alumni since Ashesi's first graduate class in 2005, there is no Ashesi platform for both students and alumni to connect with one another to build relationships. The available platforms have either Students or Alumni but not both parties. Although there are popular social networking sites such as Facebook, Twitter and LinkedIn, these sites do not serve as an appropriate platform for alumni to connect with students due to its large population.

This project sought to create a Social Networking Site solely for the Ashesi community. This is to connect students with alumni to develop relationships, collaborate on projects, provide mentorship and increase job opportunities. In addition, the project sought to provide an easy and accurate way of generating a CV in the Ashesi format. This is to replace the manual approach employed by students which often resulted in formatting errors.

In summary, this project involved the creation of a web application with the basic features of social networking sites to allow for interaction between Ashesi students and alumni.

Table of Contents

DECLARATION.....	i
ACKNOWLEDGEMENTS	ii
ABSTRACT.....	iii
List of Figures.....	vi
Chapter 1 – Introduction.....	1
1.1 Project Topic	1
1.2 Background	1
1.3 The problem	2
1.4 Motivations/Reasons for pursuing this project.....	2
1.5 Literature Review and Related Works	3
1.6 Objectives	4
Chapter 2 - Requirements	8
2.1 Systematic Approach.....	8
2.2 Project Scope/ Requirement Definition.....	9
2.3 User Groups.....	10
2.4 Use Case	10
2.4.1 Use Case Scenario	10
2.4.2 Use Case Diagram	11
2.5 Sequence Diagram.....	12
2.6 Requirements Gathering.....	13
2.6 System Requirements	13
2.6.1 Functional Requirements	13
2.6.2 Non-functional requirements	13
Chapter 3 - Architecture and Design	15
3.1 Introduction	15
3.2 System Architecture	15
3.2.1 Presentation Layer	16
3.2.4 Database Architecture.....	18
3.3 ER Diagram	19
3.4 User identification	20
3.5 Architectural Pattern	20
3.6 Class Diagram	21
Chapter 4 - Implementation.....	23
4.1 Bootstrap	23
4.1.1 How to install and use bootstrap.....	23
4.2 PHP Validation and Verification.....	25
4.3 PHP Mailer	27
4.3.1 Setting up PHP Mailer	28
4.3.2 Sample code of how an email is sent using PHP Mailer	28

4.4 TCPDF – PDF Generator	30
4.5 Implementation techniques	30
4.5.1 Registration.....	30
4.5.2 Login.....	32
4.5.3 Edit Profile.....	32
4.5.4 Search Page.....	33
4.5.5 Connect.....	34
4.5.6 CV Generator.....	35
4.5.7 Message	36
4.5.8 Notifications	37
Chapter 5: Testing and Results	39
5.1 Software Testing Methods	39
5.1.1 Black Box Test	39
5.1.2 White Box Test.....	40
5.1.3 Differences between Black Box Test and White Box Test	41
5.2 Results of Unit Testing.....	41
5.3 Results of system-level testing.....	43
5.4 Results of user testing	44
5.5 Analysis of test results.....	44
Chapter 6: Conclusions and Recommendations	45
6.1 Summary of project.....	45
6.2 Limitations and Recommendations	45
6.3 Conclusion.....	46
References	47
Appendix.....	51
A. Requirement gathering	51
A.1 Interview Questions	51
A.2 Interview Observations & Insights	52

List of Figures

Figure	Description	Page
Fig 1.1	Snapchat of MSOE bridge	5
Fig 1.2	Snapchat of MSOE bridge	6
Fig 1.3	Snapchat of MSOE bridge	7
Fig 2.1	Software process Model - Waterfall model	9
Fig 2.2	Use Case Diagram	11
Fig 2.3	Sequence Diagram	12
Fig 3.1	3-tier architecture	15
Fig 3.2	Database	18
Fig 3.3	ER Diagram	19
Fig 3.4	MVC Architectural pattern	21
Fig 3.5	Class Diagram	22
Fig 4.1	Bootstrap folder structure	24
Fig 4.3	Sample PHP validation code for Login Page	25
Fig 4.4	Sequence diagram for login PHP validation	26
Fig 4.5	Register Page	31
Fig 4.6	Login Page	32

Fig 4.7	Edit Profile Page	33
Fig 4.8	Search Page	34
Fig 4.9	CV Generator	35
Fig 4.10	AJAX code for messaging	37
Fig 4.11	Home Page	38

Chapter 1 – Introduction

1.1 Project Topic

A Social Networking Site for Ashesi University College.

1.2 Background

Ashesi has a large network of people working at various companies and in various countries since its pioneer class graduated in 2005 (Ashesi Foundation, 2018). Thankfully, with the introduction and help of social media platforms such as Facebook, Twitter, LinkedIn and CCN, students and alumni can stay in touch with each other. All these sites provide a way to connect and interact with others, however, there is no clear assurance for Ashesi students and alumni to indeed connect with each other.

According to Jamey Tucker of WPSD local 6, a broadcasting station, accepting a Facebook request from a stranger exposes the user to risks such as being hacked or robbed (Tucker, 2017). With the risks associated with accepting requests from strangers on Facebook and other social networking platforms, there is a hypothetically low probability of an alumnus accepting and communicating effectively with an Ashesi student. According to interviews conducted, alumnus will be more likely to interact with a stranger on an authentic Ashesi social networking site.

To ensure that the users of this site are Ashesi students or alumni, this project will use Ashesi emails to verify new accounts.

1.3 The problem

There is no direct way for Ashesi students to get into contact with Ashesi alumni. Aside the Ashesi email and recently Ashesi Courseware, there is no Ashesi platform that allows students to communicate with one other or give general announcements. Another problem is the inability of most students to generate a CV in the standard format required by Ashesi Career Services. Yet, most students usually find it difficult to visit the career services for guidance. This project seeks to provide a platform for students to interact with one other and connect with alumni. It also seeks to provide a simple and easy mechanism for student to generate a standardized Ashesi CV.

1.4 Motivations/Reasons for pursuing this project

Over the years, networking has proved to be a very crucial aspect of life especially in the desired career path or business. Having the right connections and knowing the right people can open doors to a lot of opportunities in terms of help, advice and business. Networking builds relationships and usually benefits one's career. In a research, "Informal Innovation: Entrepreneurship and Informal Communities" released by the Economist Intelligence Unit, the research results suggested that informal professional networks and communities are more important for entrepreneurial success than formal structures such as incubators and accelerators (The Economist, 2016). This is to say that networking is very vital to individuals and start up projects.

This project will mainly help to build the network of Ashesi students. By connecting students to alumni, students will get to know more about life after college and probably be able to collaborate with some alumni on various projects.

1.5 Literature Review and Related Works

There have been numerous social networking sites (SNSs) since the introduction of the internet. SixDegrees.com is recognized as the first social networking site which supported several features that allowed users to affiliate with others, create profiles, list friends and send messages to others (Hale, 2015). It was launched in 1997, however it failed to remain a sustainable business and shut down in 2000. The founder believed the site failed because it was ahead of its time. At that time, people complained about having little or nothing to do after accepting friend requests and most users were also not interested in meeting strangers. From 1997 to 2003, several SNSs were developed and launched with a combination of different profiles; personal, professional and dating (Boyd & Ellison, 2007).

Facebook began in early 2004 as a Harvard-only SNS and the only way to join was with the Harvard.edu email address (Phillips, 2007). Later, it began supporting other schools whose users were also required to use their university email addresses. In September 2005, it expanded to include high school students, corporate professionals and eventually the public. A differentiating feature of Facebook compared to other SNSs is its ability for outside developers to build “Applications” which allows users to personalize their profiles and perform other tasks (Boyd & Ellison, 2007).

Though Facebook is a large social networking site with various renowned schools and universities available on the site, there are a few schools which still prefer a private social network for its students. These universities are aiming to cover the span from prospective student to accepted student, enrolled student and alumnus. The Milwaukee School of Engineering (MSOE) has a private social network site centered on the student experience. The MSOE SNS began because of the drop-off between acceptance and enrollment. It was a challenge keeping in touch with

prospective students. Now, to apply to the school, students must sign up for MSOE Bridge, the school's own social network. The implementation of the site saw an 8% increase in the number of accepted student who actually enrolled in the school (Carr, 2013).

1.6 Objectives

By the end of this project, the site should have the following features:

- A user-friendly site where users do not need guidance or tutorials on how to navigate.
- Attractive and secure pages.
- Ability to search for users by name, year group and major.
- Ability to send and accept friend requests.
- Ability to edit profile.
- Ability to generate a qualified Curriculum Vitae in the Ashesi format, that will be approved by career services.
- Ability to send and receive messages.
- Ability to post general announcement or comments moderated through an administrator

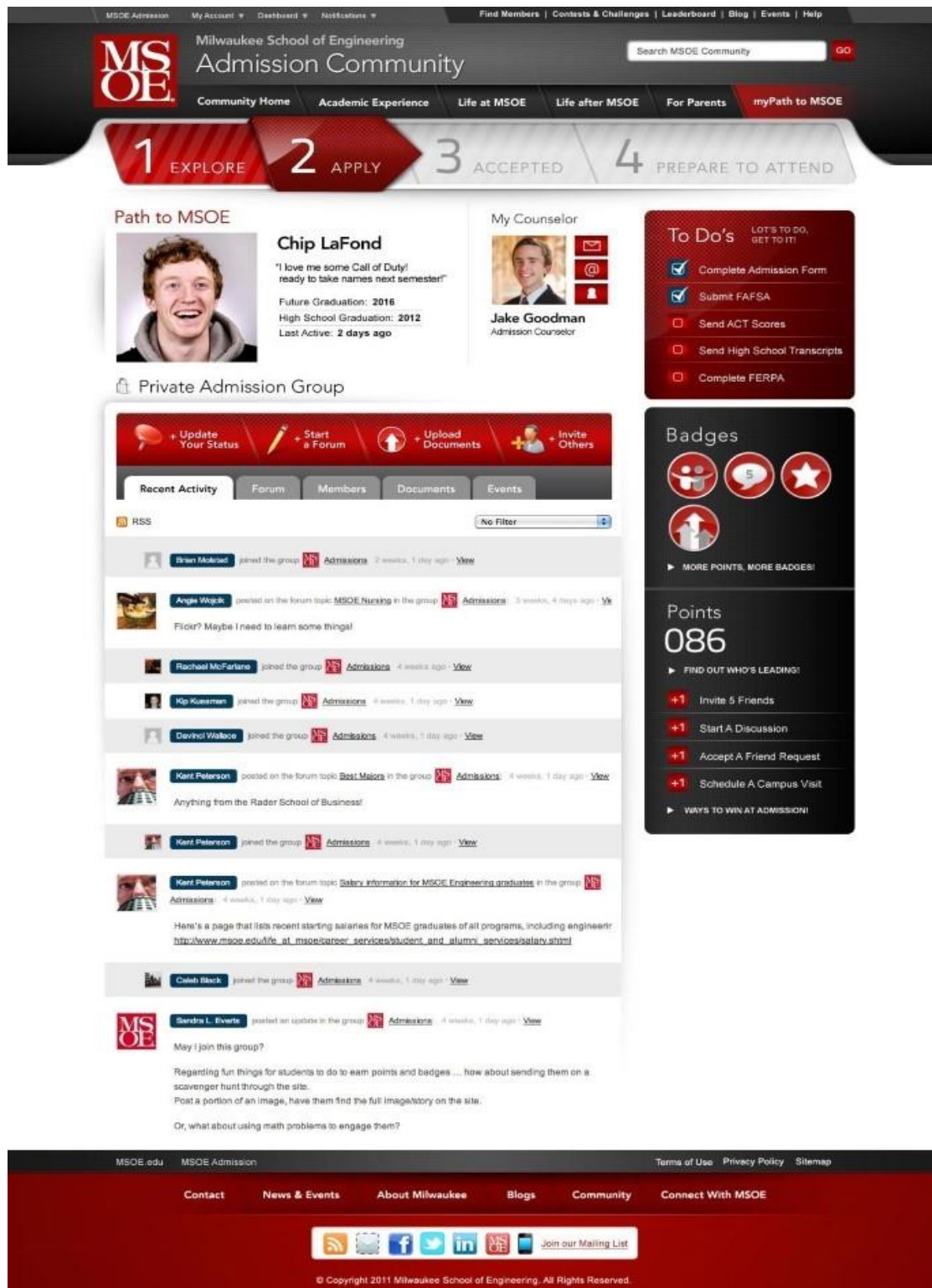


Fig 1.1 Snapchat of MSOE bridge (7summits, 2018)

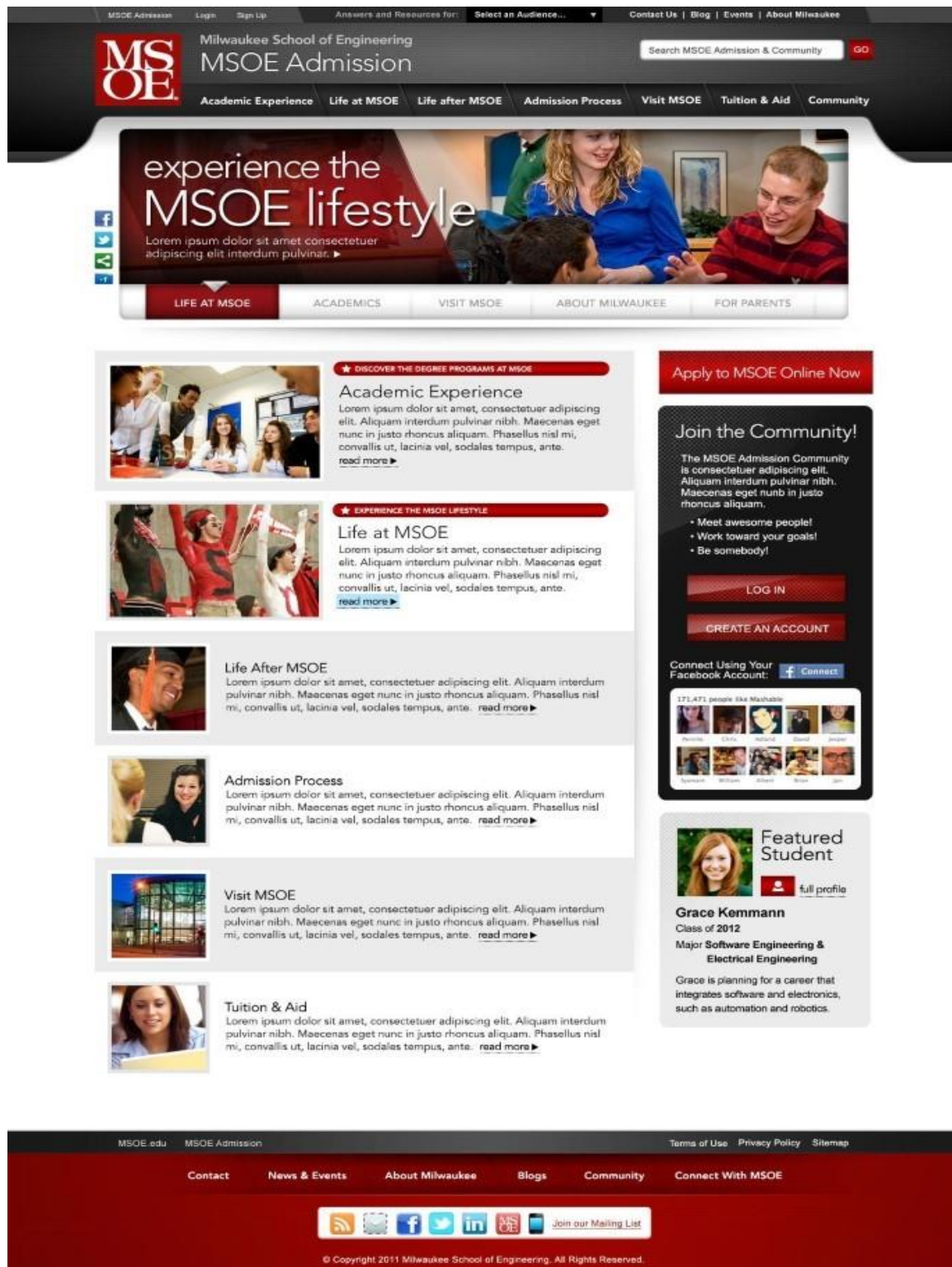


Fig 1.2 Snapchat of MSOE bridge (7summits, 2018)



Fig 1.3 Snapchat of MSOE bridge (7summits, 2018)

Chapter 2 - Requirements

2.1 Systematic Approach

A simplified representation of a software process is known as a software process model. The software process model adopted for this project is the waterfall model/software life cycle. With this model, the next phase is not supposed to start until the previous has been completed. This is to ensure consistency and completion as documentation is produced at each phase. This makes the process visible so managers or supervisors can monitor progress against the development plan. The waterfall model emphasizes on meticulous record keeping which makes it easy to improve the project in the future. The main flaw of the waterfall model is the inflexible partitioning of the project into distinct stages. The waterfall model is usually the best approach when the requirements are well understood and unlikely to change radically during system development (Sommerville, The waterfall model).

“Agile methods are incremental development methods in which the increments are small and, typically, new versions of the system are created and made available to customers every two or three weeks”. Customers are involved in the development process to get feedback on changing requirements (Sommerville, Agile Software Development, 2006). For this project, the requirements of the project have been identified, hence customers will not be involved in the developmental process as with an agile method.

The waterfall model below (Fig 2.1) presents the systematic approach in which this project will be tackled.

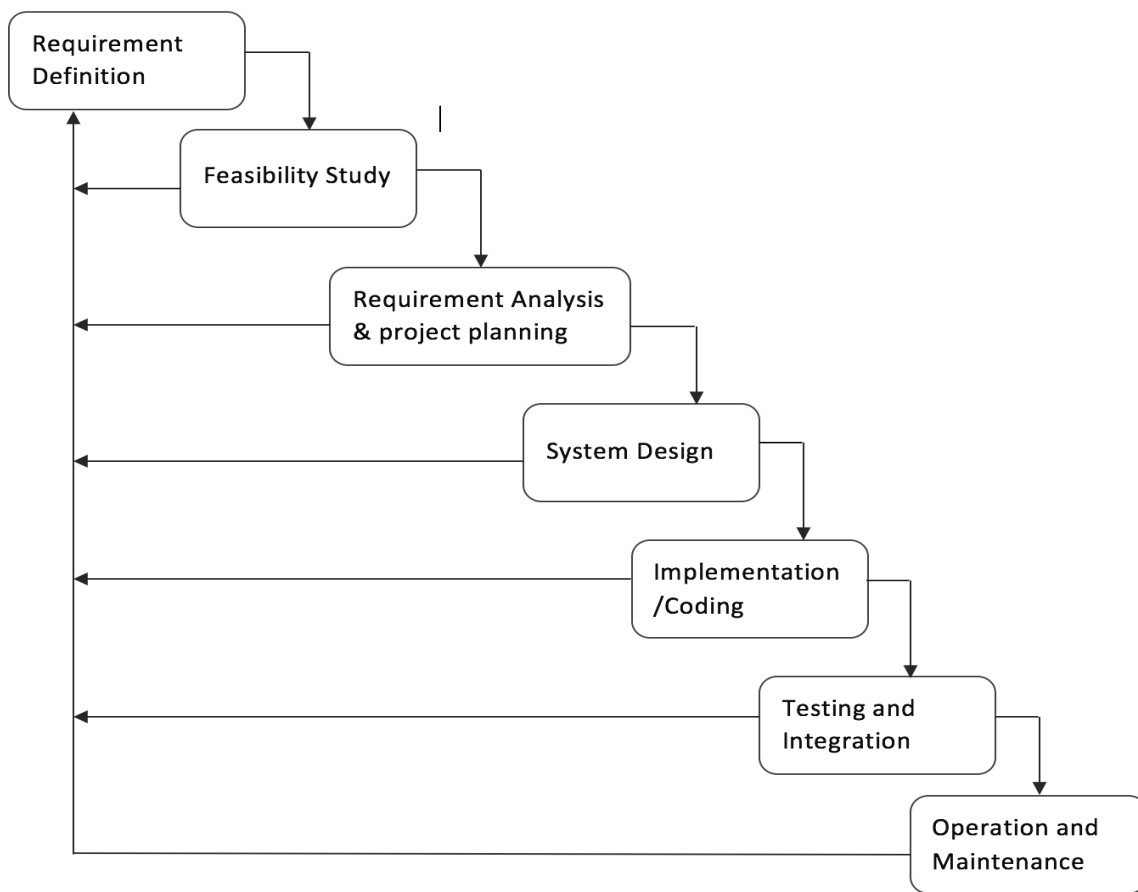


Fig 2.1 Software process Model - Waterfall model

2.2 Project Scope/ Requirement Definition

Just like every social networking site, this project seeks to connect people. However, the main requirement of this project is to connect Ashesi students to Ashesi Alumni, to create a bigger and better Ashesi community where almost everyone knows each other. This enables students to get help for alumni on various issues such as projects, job opportunities, internships and advice on living in school and life after school.

Another requirement of this project is to provide an easy and accurate way of creating a CV in the Ashesi CV format. This reduces the work of Career Services as they usually spend a lot of time reviewing the CV of students.

2.3 User Groups

The main users of the end-product of this project will be Ashesi Students and Alumni. Students and Alumni will have the same privileges: post general information, connect, send and receive messages, edit profile and generate CV.

2.4 Use Case

2.4.1 Use Case Scenario

This section presents a scenario of how a new user might access and use the site effectively. Upon entering the URL of the site, user will be directed to the login page. New user then clicks on sign up button to register for the site. Upon entering the required details requested by the page, user should receive a message in their Ashesi email notifying them of their action. This verifies the user's Ashesi membership.

User can now log in and is directed to the profile page to input more details. After this process, user may go to connect page and connect with students or alumni. Once, connect request has been accepted, user may message new connection. User may also go to homepage to post general information. User could also decide to create CV by entering CV details which will then be generated into a CV in the Ashesi format.

2.4.2 Use Case Diagram

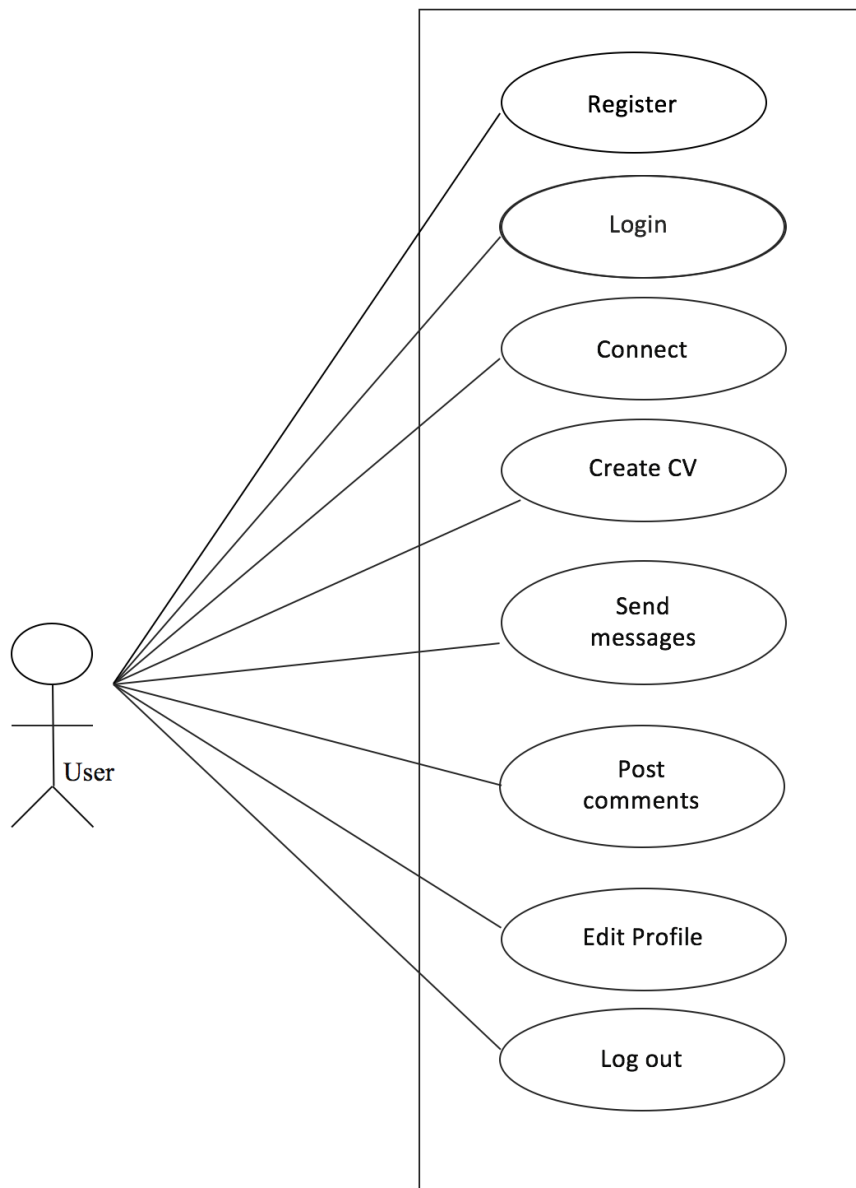


Fig 2.2 Use Case Diagram

2.5 Sequence Diagram

Below presents a sequence diagram that illustrates how a user navigates through the application. More information on sequence diagrams are provided at page (insert page number here) of this report.

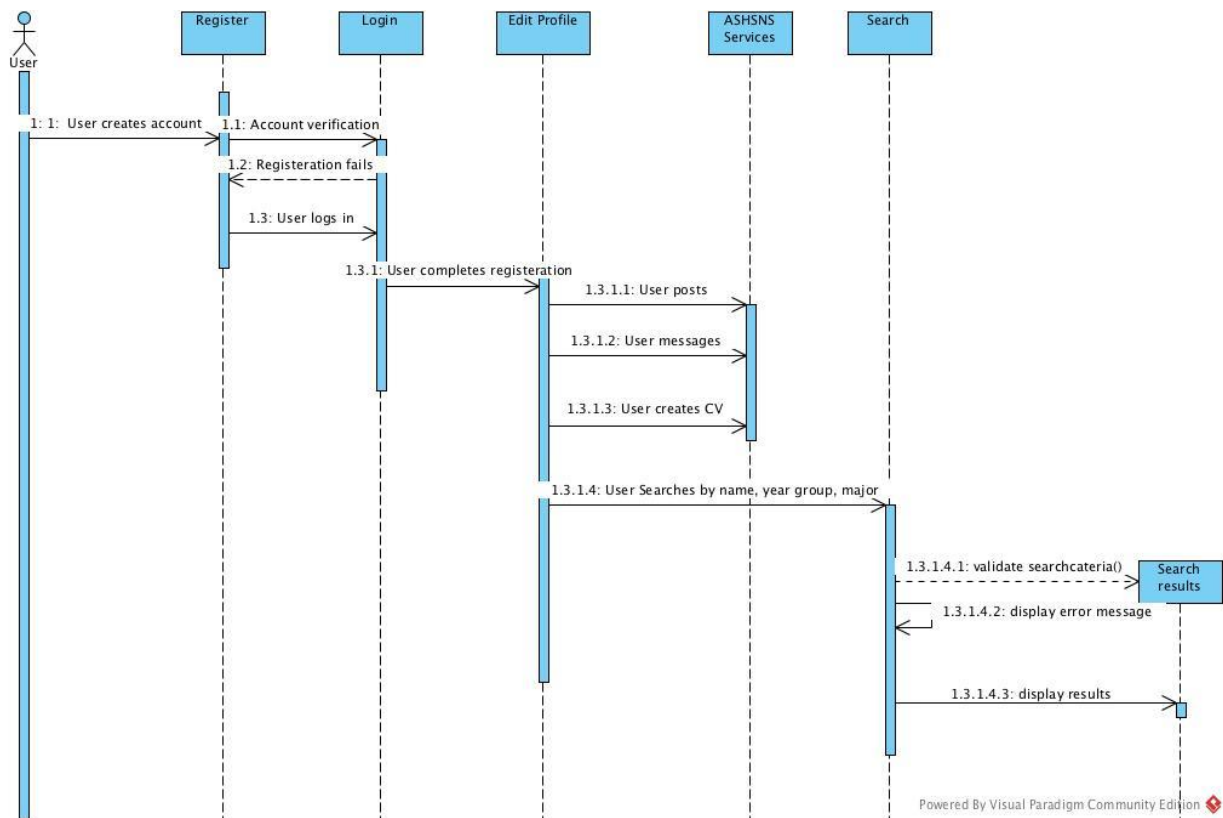


Fig 2.3 Sequence Diagram

2.6 Requirements Gathering

To gather the requirements for the project, the main end users of the system were interviewed. The results and the questions asked can be found in Appendix A of this document.

2.6 System Requirements

2.6.1 Functional Requirements

This presents the functionalities the system should possess:

Registration: This page should allow users to register with their Ashesi Office 365 email.

Login: This page should allow users to login with their Ashesi Office 365 email.

Profile Display: This page should display information about the user.

Edit user Profile: This page should allow user to edit profile.

CV generator: System should have a form which allows user to enter necessary details to be generated into a CV in the Ashesi CV format.

Messages: System should allow for direct or personal messages.

Search: User should be able to search for other users by name, year group and major.

Connect: User should be able to add other users to his or her network by sending a connect request.

Post: User should be able to post general information on site.

2.6.2 Non-functional requirements

Usability: The system should be easy to use by all users.

Reliability: The system should work wherever and whenever once internet connection is available.

Accuracy: All functions, buttons and links should work exactly as stated or labelled.

Appearance: The system should be attractive to all end users.

Privacy and security: User's personal information should be kept private and can only be accessed by authorized users.

Scalability: The system should be able to work well with increasing amount of data. It should not crash or break down when accessed by multiple users at a time. Also, there should be no redundancy in database to take extra memory space.

Usefulness: The system must be useful to the user; it must respond directly to their needs.

Performance: The system must always function efficiently and effectively regardless of platform and browser used.

Chapter 3 - Architecture and Design

3.1 Introduction

This section presents an overview of the high-level systems and architecture for this project. It also identifies the key modules and layers required in the implementation of the project.

3.2 System Architecture

This project will be a web application accessible mainly via the browser. The project makes use of the 3-tier layers comprising of a web browser (or client), a web application server, and a database server (IBM Knowledge Center, 2018). The main distinction between the 3-tier architecture and the traditional 2-tier client-server architecture is that, with a 3-tier architecture, the business logic is separated from the user interface and the data source (Homer, 2018). The benefits of the 3 tier architecture are ease of management, scalability, flexibility and security. Should the website be hosted live in the future, it can be accessed via its URL. Users will be required to register with their Ashesi email address and a password.

3-tier architecture

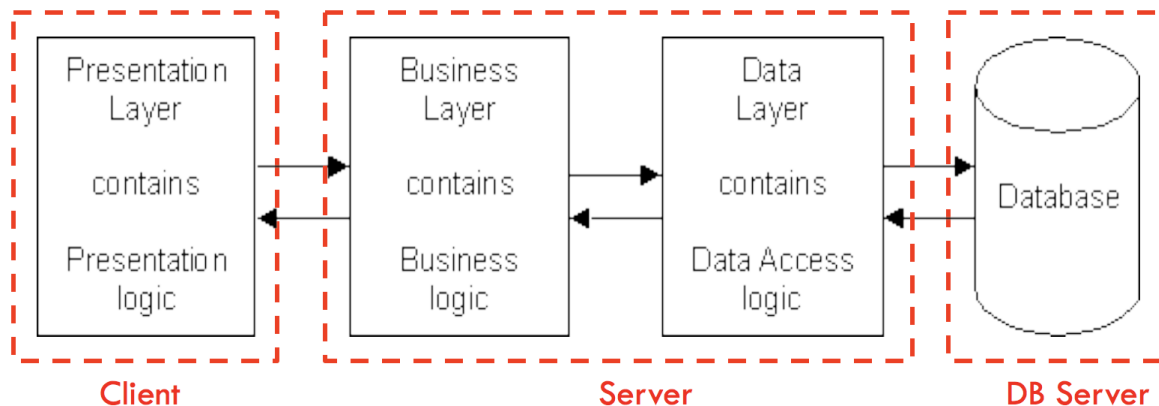


Fig 3.1 3-tier architecture

3.2.1 Presentation Layer

The presentation layer (Client) is the front-end. It could also be referred to as the Graphical User Interface (GUI). It is independent of the business and data layers. The software tools used in this layer and for this project are:

- a) **HTML** : HTML stands for HyperText Markup Language. It is used to create pages to be displayed on the World Wide Web. Each page is connected to other pages by hyperlinks. The HTML code ensures proper formatting of text and images so that the internet browser displays pages as they are intended to look. HTML provides a basic structure of the page upon which Cascading Style Sheets are overlaid to change its appearance (Computer Hope, 2017).
- b) **CSS**: Cascading Style Sheets
- c) **JavaScript**: It is the programming language of HTML and the web (w3schools, 2017). It is usually used for client-side validation.
- d) **Bootstrap**: Bootstrap is a free front-end framework for faster and easier web development. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins (w3schools, 2018).

3.2.2 Business/ Logic Layer

The business or logic layer is the middleware/backend. It is the set of rules for processing information and can accommodate several users. Software tools used are:

- a) **PHP**: Hypertext Preprocessor is a language designed for creating HTML content that interacts with databases. PHP can be used in three ways: server-side scripting, command

line scripting and client-side GUI applications (Tatroe, MacIntyre, & Lerdorf). However, for this project, PHP is used for server –side scripting.

- b) Asynchronous JavaScript And XML (AJAX): It is a client-side script that allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes (w3schools, 2018).

3.2.3 Data Layer

The Database Server manages access to the file system or database. The server currently being used for the project is XAMPP. XAMPP stands for:

- Cross-Platform (X)- this means it is compatible with most Operating systems,
- Apache (A)- an apache server
- MariaDB (M)- a database server
- PHP (P) – hypertext processor
- and Perl (P)- Practical extraction and reporting language, a programming language used for web application.

XAMPP is a popular PHP development environment. It is a server that can be installed on a laptop or desktop. Other software tools used in this layer are:

- a) phpMyAdmin: It is a free software tool written in PHP intended to handle the administration of MySQL over the Web. It supports a wide range of operations on MySQL and MariaDB (phpmyadmin, 2003).

- b) MySQL: It is currently one of the world’s most popular open source database. Due to its proven performance, reliability and ease of use, it is the leading database choice for most web-based applications (Oracle, 2017).

3.2.4 Database Architecture

A database is a shared collection of logically related data, and a description of this data, designed to meet the information needs of an organization (Connally & Begg, 2005). It is usually represented as a collection of rows (indicating the tuple) and columns (indicating the domain) of the table. The database for this project is named “projectConnect” and it currently consists of 4 entities with their various attributes. Below is a snapshot of the database entities.

Table	Action	Rows	Type	Collation	Size	Overhead
friendship	<input type="checkbox"/> Browse <input type="checkbox"/> Structure <input type="checkbox"/> Search <input type="checkbox"/> Insert <input type="checkbox"/> Empty <input type="checkbox"/> Drop	2	MyISAM	latin1_swedish_ci	2 KiB	-
messages	<input type="checkbox"/> Browse <input type="checkbox"/> Structure <input type="checkbox"/> Search <input type="checkbox"/> Insert <input type="checkbox"/> Empty <input type="checkbox"/> Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
post	<input type="checkbox"/> Browse <input type="checkbox"/> Structure <input type="checkbox"/> Search <input type="checkbox"/> Insert <input type="checkbox"/> Empty <input type="checkbox"/> Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
user	<input type="checkbox"/> Browse <input type="checkbox"/> Structure <input type="checkbox"/> Search <input type="checkbox"/> Insert <input type="checkbox"/> Empty <input type="checkbox"/> Drop	21	InnoDB	latin1_swedish_ci	16 KiB	-
4 tables	Sum	23	InnoDB	latin1_swedish_ci	50 KiB	0 B

Fig 3.2 Database

The friendship entity contains the details for accepting and sending friend request. The messages entity contains details about the receiver and sender of the message and the message itself. The post entity contains the details involved in any post. Example: post type (text, photo or video) and

post date and time. The user entity contains all the details about the user. Example: name, major, date of birth, password, year group etc.

Other software Tools used

Sublime Text: It is a text editor for code, mark-up and prose (Sublime Text, 2018).

3.3 ER Diagram

ER Diagram for ASHSNS

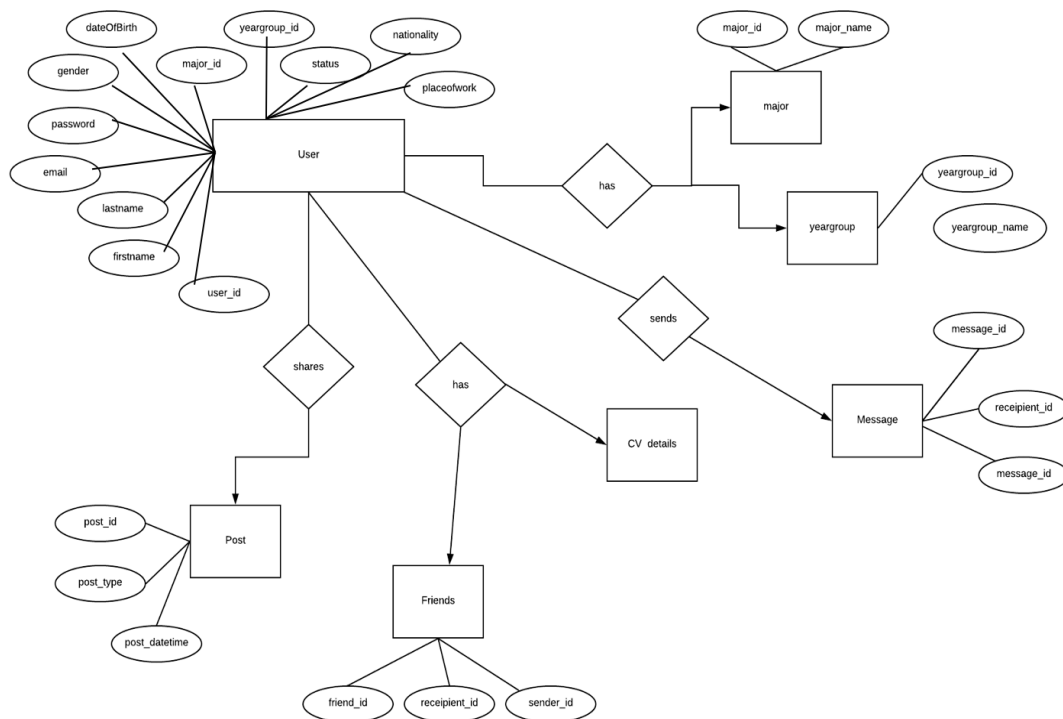


Fig 3.3 ER Diagram

3.4 User identification

This project is not intended for the public, hence the only form of identification is via the Ashesi email address. A form validation will be made to ensure that user's email is of the form @ashesi.edu.gh. For further verification, an email will be sent to the user's Ashesi email to verify account. User can then log in with Ashesi email and a specified password.

3.5 Architectural Pattern

The Model-View-Controller(MVC) architectural pattern will be used in the development of this project. This architectural pattern separates presentation and interaction from the system data. It has three logical components that interact with each other. The Model component manages the system data and associated operations on that data. The View component defines and manages how the data is presented to the user. The Controller component manages user interaction and passes these interactions to the View and the Model.

An advantage of the MVC architectural pattern is that it allows the data to change independently of its representation and vice versa. It also supports presentation of the same data in different ways with changes made in one representation reflected in all. The disadvantage however is that while data model and interactions may be simple, it can involve additional code and code complexity.

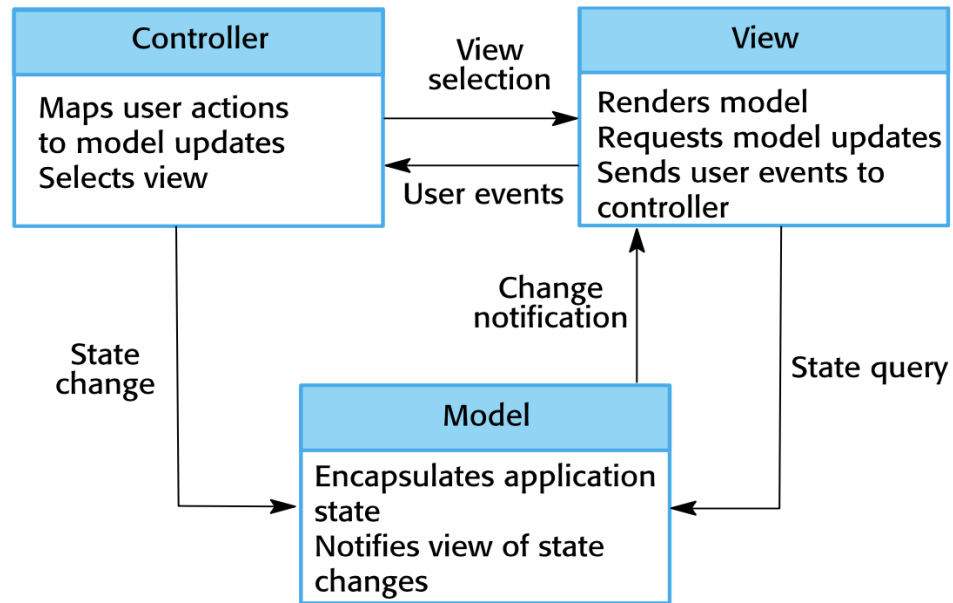


Fig 3.4 MVC Architectural pattern

3.6 Class Diagram

A class diagram is diagram which illustrates the attributes and operations of a class and the relationships between classes. The diagram below demonstrates the relationships between the classes of this project. User shares posts, sends messages, creates a CV and has friends.

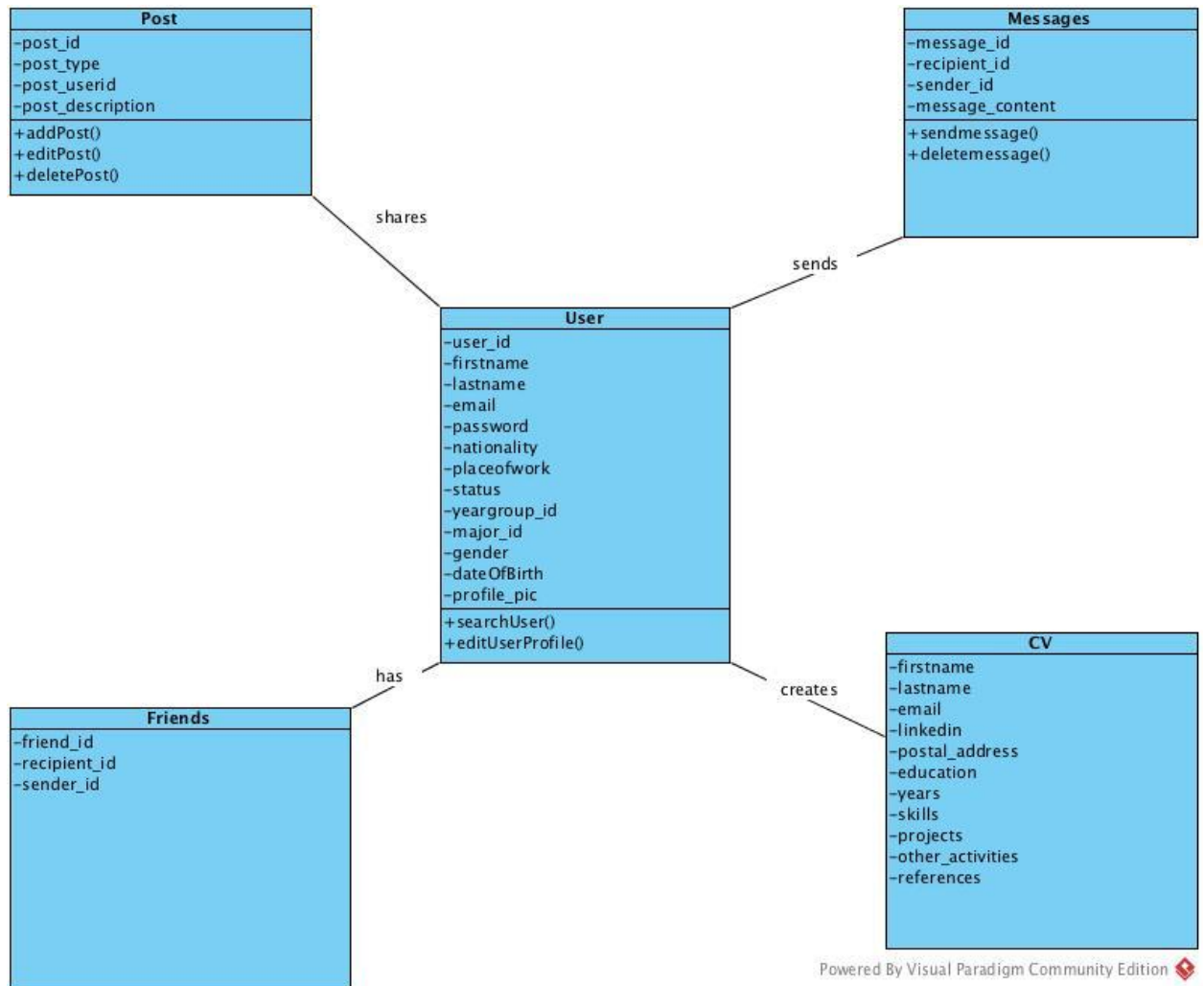


Fig 3.5 Class Diagram

Chapter 4 - Implementation

This chapter provides a description of the implementation tools, libraries, frameworks, APIs and other components used for the project.

4.1 Bootstrap

Bootstrap is an open source front end library used for designing web applications. Its package includes HTML, CSS and JavaScript codes.

4.1.1 How to install and use bootstrap

Bootstrap is downloadable in two forms:

- download bootstrap
- download source code.

For the purposes of this project the basic Bootstrap (Compiled and minified CSS, JavaScript, and fonts with no documentation or original source files included) was downloaded.

Once downloaded, the compressed folder requires unzipping to view the structure below (the compiled) Bootstrap (Bootstrap, n.d.).

```
bootstrap/
├── css/
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap.min.css.map
│   ├── bootstrap-theme.css
│   ├── bootstrap-theme.css.map
│   ├── bootstrap-theme.min.css
│   └── bootstrap-theme.min.css.map
├── js/
│   ├── bootstrap.js
│   └── bootstrap.min.js
└── fonts/
    ├── glyphs-halflings-regular.eot
    ├── glyphs-halflings-regular.svg
    ├── glyphs-halflings-regular.ttf
    ├── glyphs-halflings-regular.woff
    └── glyphs-halflings-regular.woff2
```

Fig 4.1 Bootstrap folder structure

A basic bootstrap template is of this format.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must come
    *after* these tags -->
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

Fig 4.2 Basic Bootstrap Template

The project was built upon this basic template and modified from time to time according to the functionalities of the web application.

4.2 PHP Validation and Verification

Validation of forms can be implemented in various ways. For this project, validation was done using JavaScript and PHP but mainly PHP.

```

1  <?php
2
3  /**
4   * @author Selassie Golloh
5   */
6
7  //include login class
8
9  include("../classes/loginclass.php");
10
11 $error_login;
12
13
14 if (isset($_POST['login'])){
15     validlogin();
16 }
17
18 function validlogin(){
19     $success = true;
20     $email = "";
21     $password = "";
22
23     if(isset($_POST['email']) && !empty($_POST['email'])){
24         $email = $_POST['email'];
25     } else{
26         $emailError = "Please enter your email";
27         $success= false;
28     }
29
30     if (isset($_POST['password']) && !empty($_POST['password'])){
31         $password = $_POST['password'];
32     }else{
33         $pwdError = "Please enter your password";
34         $success= false;
35     }
36
37     if($success){
38         verifyLogin($email, $password);
39         global $error_login;
40         $error_login = false;
41     }
42
43
44 }
45
46
47 /*
48 *function to verify user login and log a user into the system
49 */
50 function verifyLogin($email, $password){
51
52     global $status;
53
54     $login = new Login();
55     $dbquery = $login->verifylogin($email, $password);
56 }
57
58 /**a function to display error or success message upon login
59 */
60 function loginstatus()
61 {
62     global $error_login;
63     if (!isset($error_login)) {
64         echo "";
65     } else {
66         echo "<h4 style='color:red'>User Doesn't Exist or Wrong Credentials</h4><br>";
67     }
68 }
69
70
71 ?>

```

Fig 4.3 Sample PHP validation code for Login Page

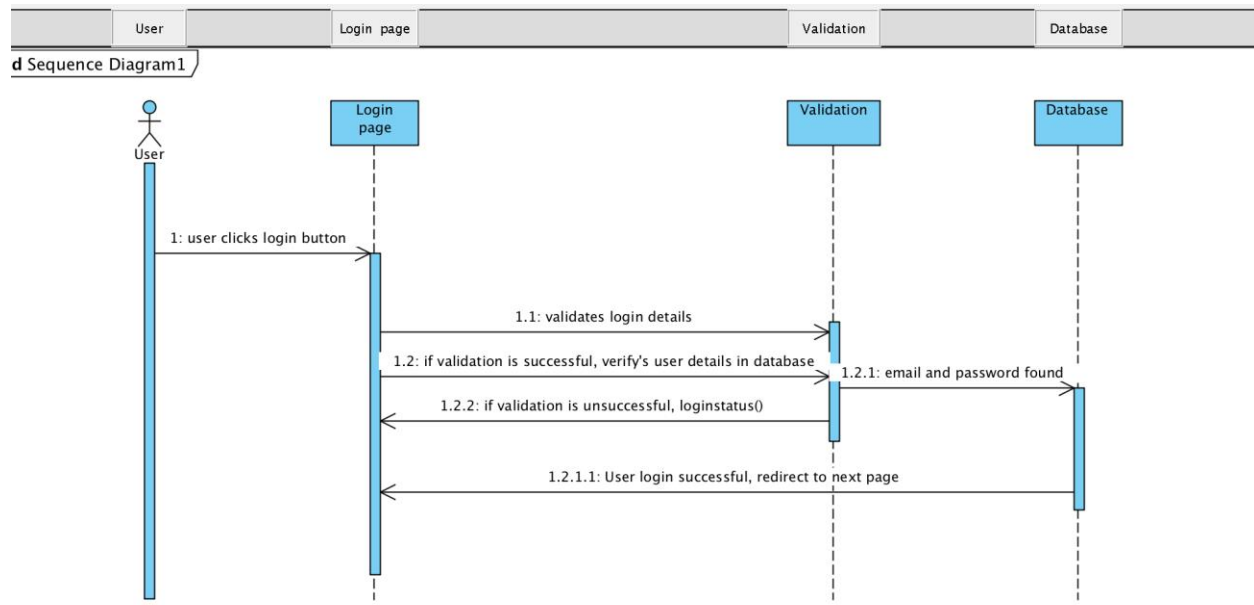


Fig 4.4 Sequence diagram for login PHP validation

4.3 PHP Mailer

PHP Mailer is an email creation and transfer class for PHP. PHP has a mail () function which allows developers to utilize email in their codes, however PHP Mailer provides the advantage of having certain features such as HTML-based emails and attachments. With PHP Mailer, developers can use their own SMTP server and avoid Sendmail routines used by the mail() function on Unix platforms. The mail() function in PHP has so many hidden problems hence it is recommended to use established email library such as PHP Mailer, eZcomponents, SwiftMailer, Zend/Mail among others (PHPMailer, n.d.).

PHP Mailer was chosen over other established email libraries for this project for the following reasons;

- It is probably the world's most popular code for sending email from PHP.
- It has integrated SMTP support, allowing users to send emails without local mail servers.
- It provides SMTP authentication with LOGIN, PLAIN, CRAM-MD5 and XOAUTH2 mechanisms over SSL and SMTP+STARTTLS transports.
- It validates email addresses automatically.
- It protects against header injection attacks.
- It is compatible with PHP 5.5 and later
- It is name spaced to prevent name clashes (PHPMailer, n.d.).

4.3.1 Setting up PHP Mailer

PHP Mailer can be downloaded from Github into the project directory or may be installed via composer.

4.3.2 Sample code of how an email is sent using PHP Mailer

```
<?php
// Import PHPMailer classes into the global namespace
// These must be at the top of your script, not inside a function
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

//Load composer's autoloader
require 'vendor/autoload.php';

$mail = new PHPMailer(true);          // Passing `true` enables exceptions
try {
```

```

//Server settings

$mail->SMTPDebug = 2;           // Enable verbose debug output

$mail->isSMTP();                // Set mailer to use SMTP

$mail->Host = 'smtp1.example.com;smtp2.example.com'; // Specify
main and backup SMTP servers

$mail->SMTPAuth = true;         // Enable SMTP authentication

$mail->Username = 'user@example.com'; // SMTP username

$mail->Password = 'secret';     // SMTP password

$mail->SMTPSecure = 'tls'; // Enable TLS encryption, `ssl` also
accepted

$mail->Port = 587;              // TCP port to connect to


//Recipients

$mail->setFrom('from@example.com', 'Mailer');

$mail->addAddress('joe@example.net', 'Joe User'); // Add a
recipient

$mail->addAddress('ellen@example.com'); // Name is optional

$mail->addReplyTo('info@example.com', 'Information');

$mail->addCC('cc@example.com');

$mail->addBCC('bcc@example.com');


//Attachments

$mail->addAttachment('/var/tmp/file.tar.gz'); // Add attachments

$mail->addAttachment('/tmp/image.jpg', 'new.jpg'); // Optional name


//Content

$mail->isHTML(true);           // Set email format to HTML

$mail->Subject = 'Here is the subject';

$mail->Body = 'This is the HTML message body <b>in bold!</b>';

$mail->AltBody = 'This is the body in plain text for non-HTML mail
clients';

$mail->send();

```

```
        echo 'Message has been sent';  
    } catch (Exception $e) {  
        echo 'Message could not be sent. Mailer Error: ', $mail->ErrorInfo;  
    }
```

4.4 TCPDF – PDF Generator

TCPDF is a PHP library for generating PDF documents in PHP. When downloaded from Sourceforge.net, it comes as a Zip archive. Once unzipped, access is granted to the TCPDF directory which contains everything needed. Users can then go ahead to configure their TCPDF installation according to the Operating System in use (Herborth, 2010). There are several examples on the TCPDF website (<https://tcpdf.org/>) that assist with how to use TCPDF.

4.5 Implementation techniques

This section discusses the various components and the implementation techniques used in each component.

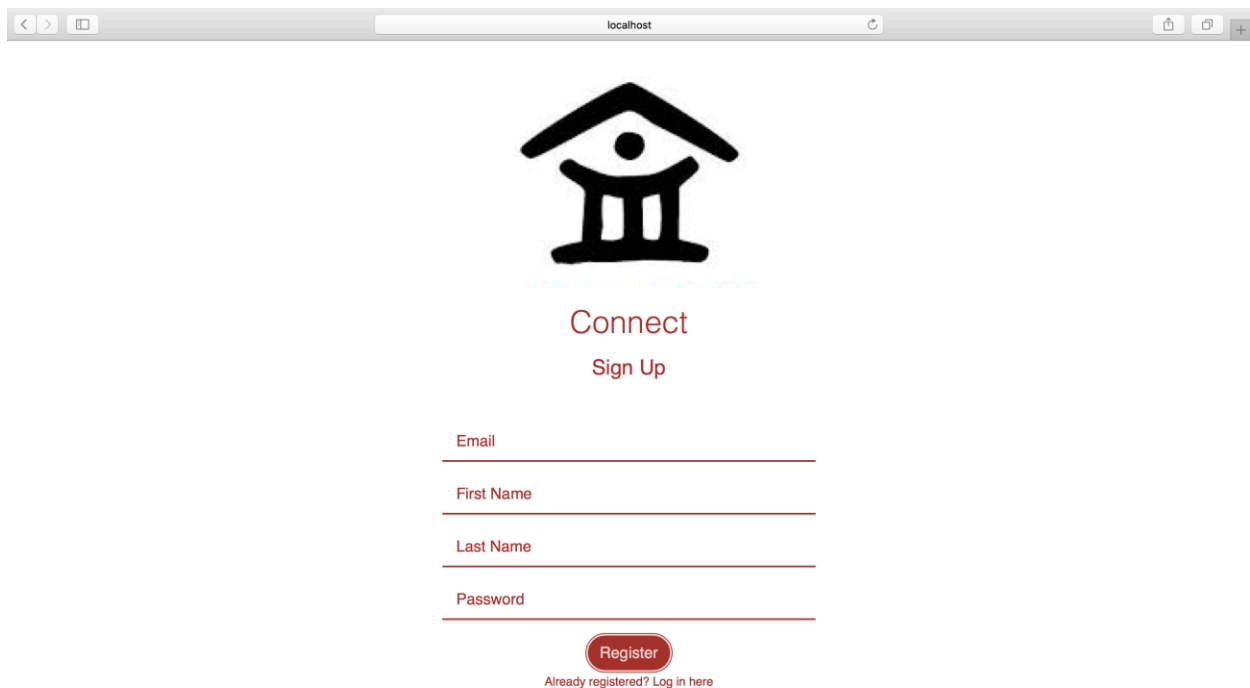
4.5.1 Registration

This is the register page, which takes in user's email, first name, last name and password. A link was provided to redirect already registered users to the login page. The design of this page was achieved with the use of a bootstrap form.


PHP validation was created to ensure that users do not provide incorrect details. A regular expression was made for the email to ensure that user's email is of the form name@ashesi.edu.gh or name@alumni.ashesi.edu.gh. A check was also made on the user's first name and last name to

ensure that users do not enter names with numbers or symbols. Only letters and hyphens are accepted. Users passwords are required to be between six to sixteen values. The values must contain at least a letter and a number.

Once user completes registration, a link is sent to the user's Ashesi email to verify his or her account. Clicking on the link in the email sets the user's status as active in the database and user has access to use the application.



localhost



Connect
Sign Up

Email

First Name

Last Name

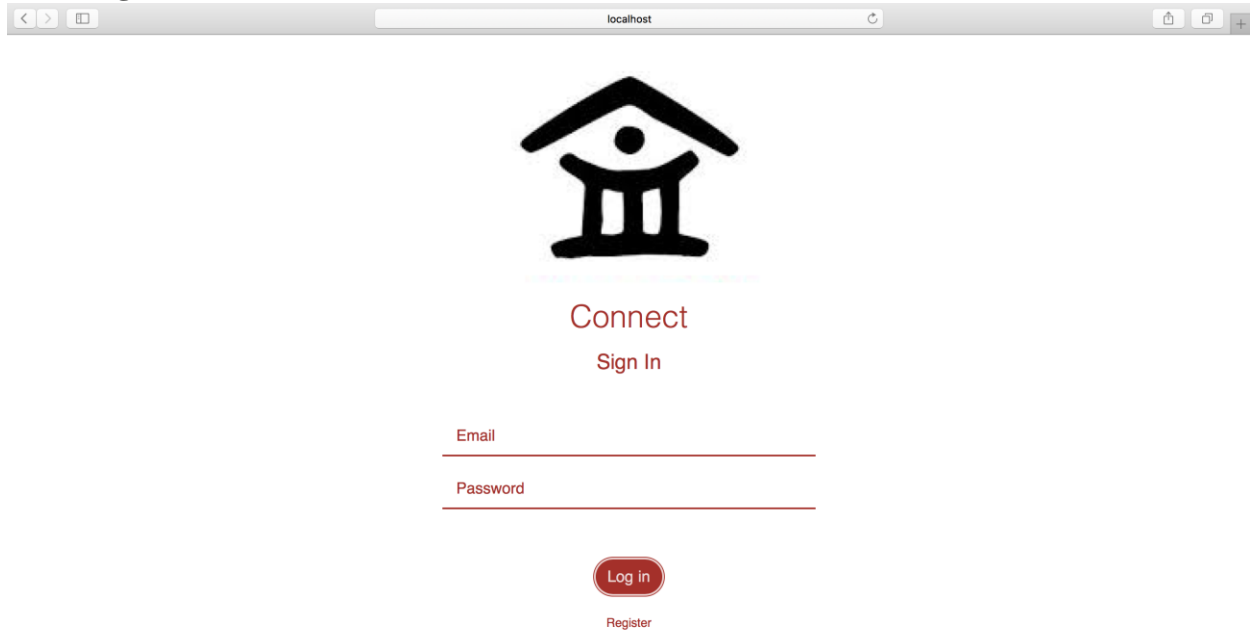
Password

Register

Already registered? Log in here

Fig 4.5 Register Page

4.5.2 Login



The screenshot shows a web browser window with the address bar set to 'localhost'. The login page has a minimalist design with a black logo of a person standing under a house-like shape. Below the logo, the text 'Connect' is in a large, dark font, and 'Sign In' is in a smaller, red font. There are two input fields: 'Email' and 'Password', both with red borders. Below the 'Password' field is a red 'Log in' button. At the bottom, there is a red 'Register' link.

Fig 4.6 Login Page

Like the register page, the login page was designed with bootstrap and has a PHP validation to check if user's details can be found in the database. A link to send users to the register page is available.

4.5.3 Edit Profile

Users are redirected to the edit profile page once a verified user logs in. Here, users are required to preview information about themselves. This page updates the database with the last name, gender, year group, major, status (Student/ Alumnus), nationality, work place and profile picture of the user. Just like the other pages, this page also uses a bootstrap form. A session is also started from the login page to identify a user in order to retrieve the details of that user.

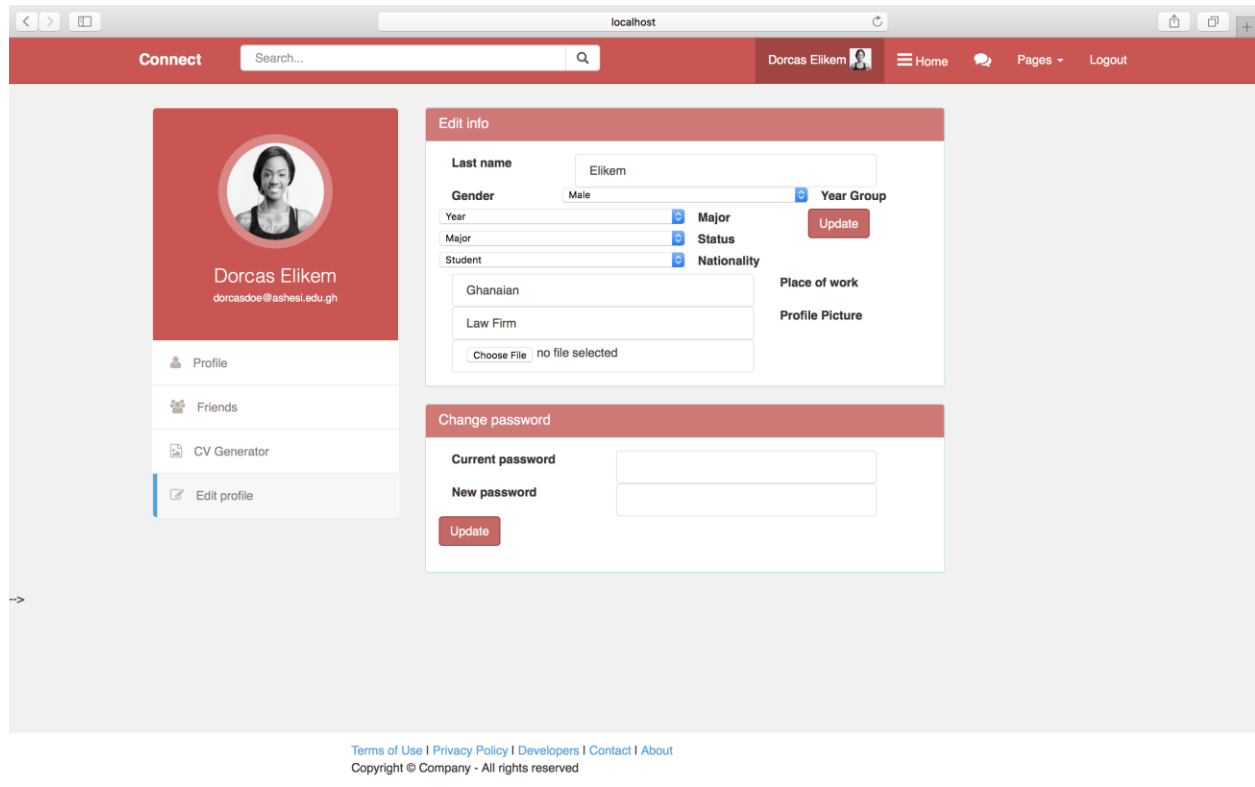


Fig 4.7 Edit Profile Page

4.5.4 Search Page

This is a search page. Here, users can search for other users by year group, major or name. The results are displayed as seen above. To achieve this, an ajax code was used to check for the selected options and results are displayed according to what is chosen. Users can also use all three parameters to search for other users.

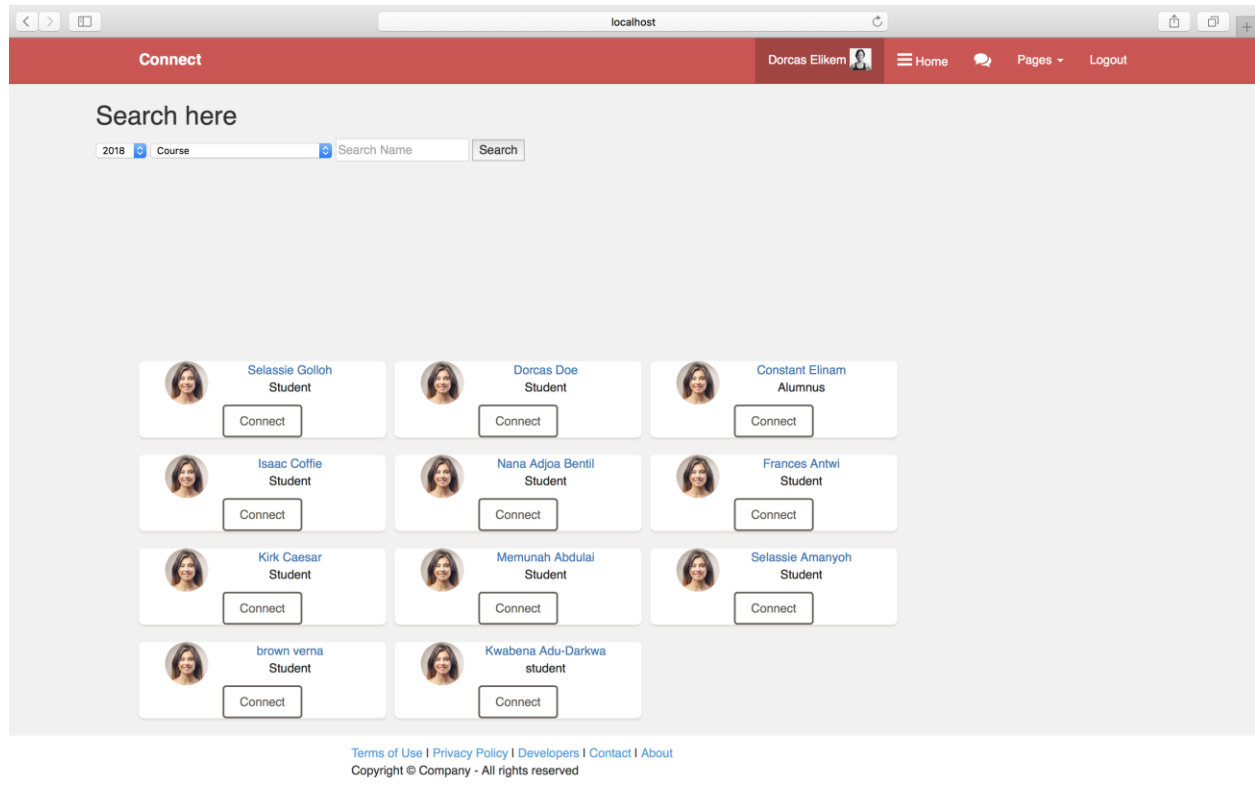
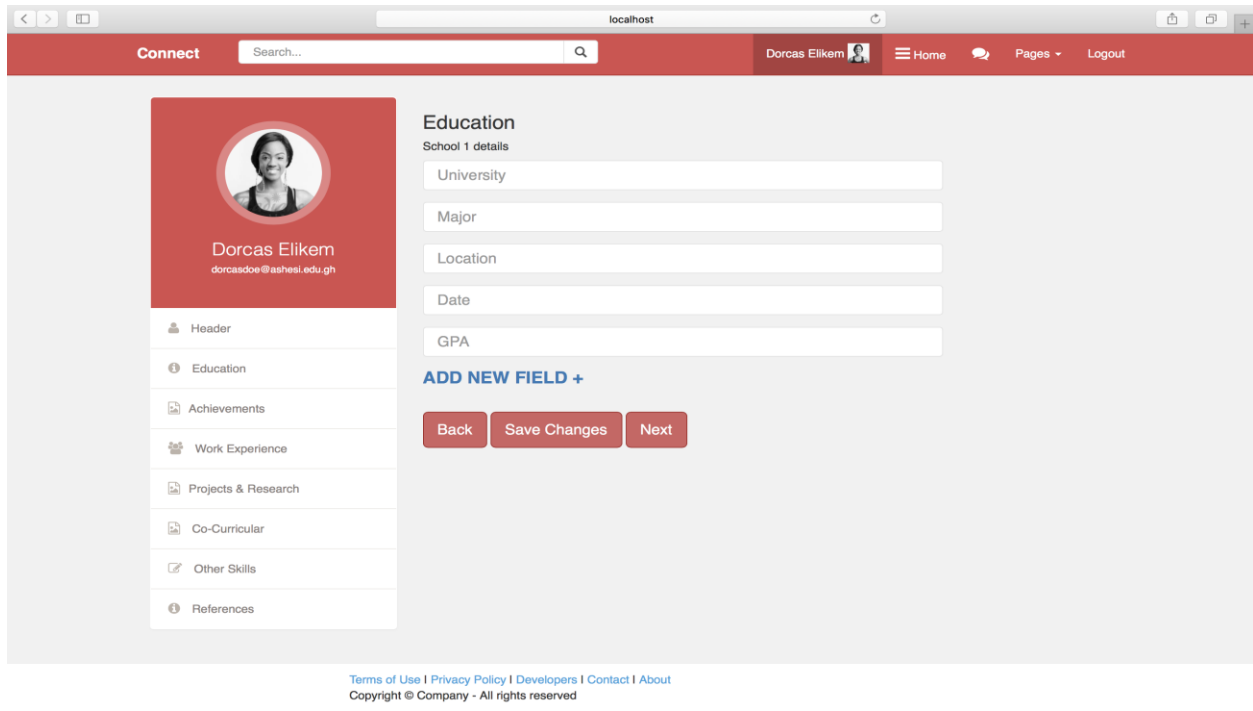


Fig 4.8 Search Page

4.5.5 Connect

When the search results show up, a user can click on the ‘Connect’ button to link up with other users. In this application, once a user connects to another user, both users are automatically connected, that is, become friends. However, both users have the option of blocking each other. This approach was chosen because as per the objectives of this project, users are to connect with each other to grow their network, hence no need to accept a connect request. As such, the block option is available for all users who do not wish to connect with some users anymore.

4.5.6 CV Generator



The screenshot shows the 'Education' page of the CV Generator. The page has a red header with a 'Connect' button, a search bar, and a user profile for 'Dorcas Elikem'. The left sidebar contains a menu with options: Header, Education (selected), Achievements, Work Experience, Projects & Research, Co-Curricular, Other Skills, and References. The main content area is titled 'Education' and includes a sub-section 'School 1 details'. It features five input fields: 'University', 'Major', 'Location', 'Date', and 'GPA'. Below these fields is a blue link 'ADD NEW FIELD +' and three buttons: 'Back', 'Save Changes', and 'Next'. At the bottom, there is a footer with links for 'Terms of Use', 'Privacy Policy', 'Developers', 'Contact', and 'About', along with a copyright notice.

Connect Search... Dorcas Elikem Home Pages Logout

Education
School 1 details

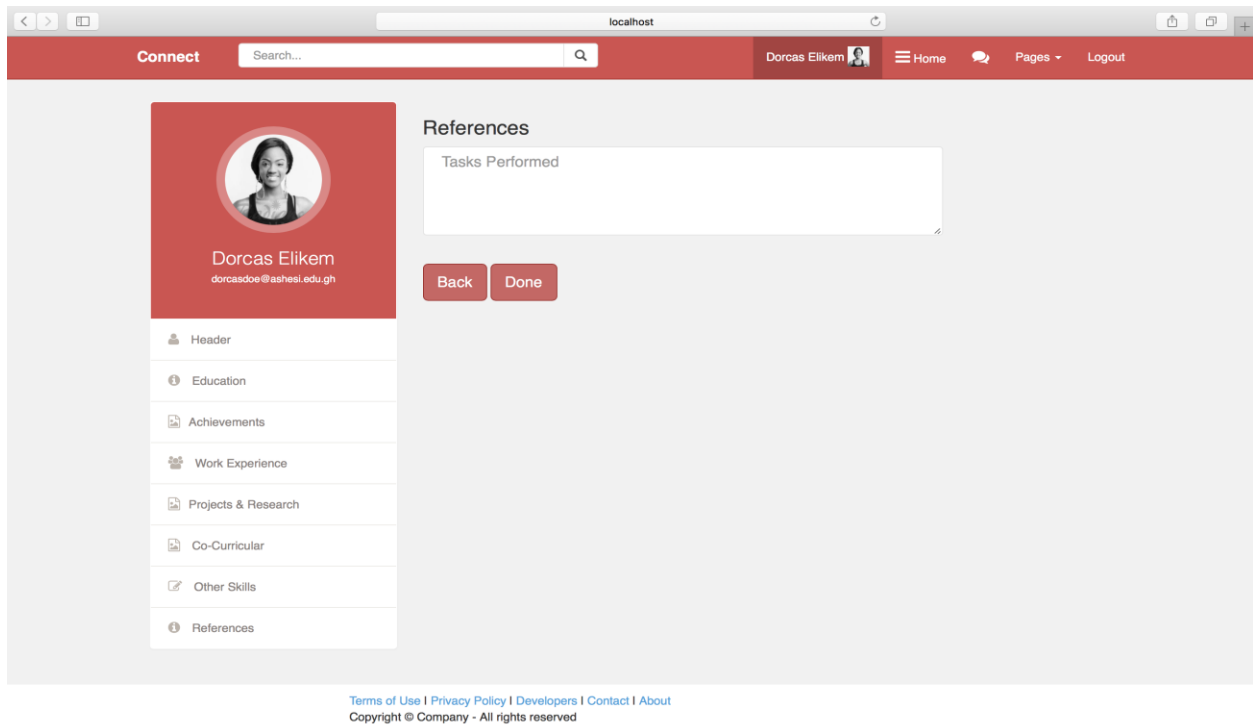
University
Major
Location
Date
GPA

[ADD NEW FIELD +](#)

Back Save Changes Next

[Terms of Use](#) | [Privacy Policy](#) | [Developers](#) | [Contact](#) | [About](#)
Copyright © Company - All rights reserved

Education page of CV Generator



The screenshot shows the 'References' page of the CV Generator. The page has a red header with a 'Connect' button, a search bar, and a user profile for 'Dorcas Elikem'. The left sidebar contains a menu with options: Header, Education, Achievements, Work Experience, Projects & Research, Co-Curricular, Other Skills, and References (selected). The main content area is titled 'References' and includes a sub-section 'Tasks Performed' with a large text area for input. Below this is a 'Back' button and a 'Done' button. At the bottom, there is a footer with links for 'Terms of Use', 'Privacy Policy', 'Developers', 'Contact', and 'About', along with a copyright notice.

Connect Search... Dorcas Elikem Home Pages Logout

References
Tasks Performed

Back Done

[Terms of Use](#) | [Privacy Policy](#) | [Developers](#) | [Contact](#) | [About](#)
Copyright © Company - All rights reserved

References page of CV Generator

Fig 4.9 CV Generator

The CV Generator consists of several forms which accepts user details to generate a CV in Ashesi CV format. This was done using bootstrap for design, jQuery for adding an additional field and JavaScript validation for the forms. Above are screenshots of the education and references pages. Once a user enters all details and clicks “Done” a preview of the CV is generated for the user. The user can then go ahead to click on the “Convert to pdf” button to export the CV in pdf format.

4.5.7 Message

This is one of the main functionalities of the application. This allows users to communicate with their friends. Here, a user selects the friend with whom he/she wishes to communicate and sends a message. This part of the project was done using AJAX.

```
<script type="text/javascript">
let senderId = '', receiverId = '';
document.querySelector("#button").addEventListener("click", (event)=>{
    event.preventDefault();
    sendMessage();
})
function sendMessage(){
    let message = document.getElementById("message").value;
    let placeholder = document.getElementById("chat");
    if(message !== ''){
        let xhr = new XMLHttpRequest();
        xhr.open("GET", `./sendMessage.php?message=${message}&sender=${senderId}&receiver=${receiverId}`, true);
        xhr.onreadystatechange = function() {
            let output = '';
            if(xhr.readyState == 4 && xhr.status == 200) {
                let result = JSON.parse(this.responseText);
                if(result.length > 0){
                    result.forEach(function(elem){
                        if(elem.hasOwnProperty('receiver')){
                            output+=
                                `<li class="left clearfix">
                                    <div class="chat-body clearfix">
                                        <div class="header">
                                            <strong class="primary-font"></strong>
                                            <small class="pull-right text-muted"><i class="fa fa-clock-o"></i>${elem['createdAt']}</small>
                                        </div>
                                        <p>
                                            ${elem['message']}
                                        </p>
                                    </div>
                                </li>`
                        }
                    })
                }
            }
        }
    }
}
```

```

    }else{
        output+=
        `<li class="right clearfix">
            <div class="chat-body clearfix">
                <div class="header">
                    <strong class="primary-font"></strong>
                    <small class="pull-right text-muted"><i class="fa fa-clock-o"></i>${elem['createdAt']}</small>
                </div>
                <p>
                    ${elem['message']}
                </p>
            </div>
        </li>`
    }
    })
    }else{
        output = `<h1>No Result match your query</h1>`
    }
    placeholder.innerHTML = output;
}
}
xhr.send();
}else{
    alert("Add Message");
}
}

//code to get user by clicking
$(".users").click(function(e){
    placeholder = document.getElementById("chat")
    placeholder.innerHTML = '';
    receiverId = $(this).data('receiver');
    senderId = $(this).data('sender');
    document.getElementById("receiverName").innerHTML = 'Chatting With: ' + $(this).data('name');
});
</script>

```

Fig 4.10 AJAX code for messaging

4.5.8 Notifications

Once a message is sent to a user, the receiver should get a notification informing them of the message. The notification is an important feature that makes users aware of messages sent to them.

4.5.9 Homepage

Below is a snippet of the basic landing page of the user. This page displays posts made by friends. The news page keeps getting updated as long as the user's friends keep posting. User may also post information on this page. Notifications are at the right side of the page for easy and quick notification of any information.

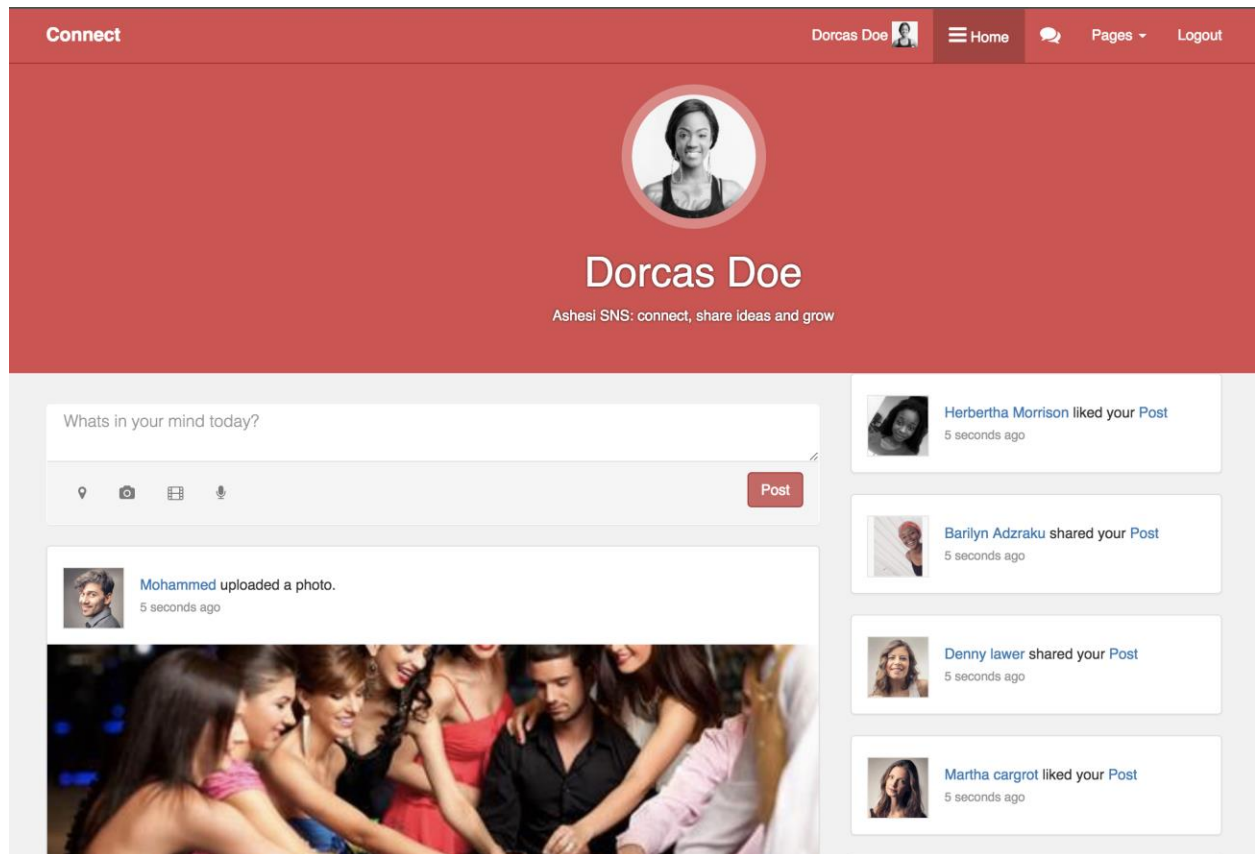


Fig 4.11 Home Page

Chapter 5: Testing and Results

Following the systematic approach used for this project, after implementation comes testing. Testing involves conducting tests to validate the functionalities of the application to determine if they work as expected. This section presents the test methods, test cases and test results of this application.

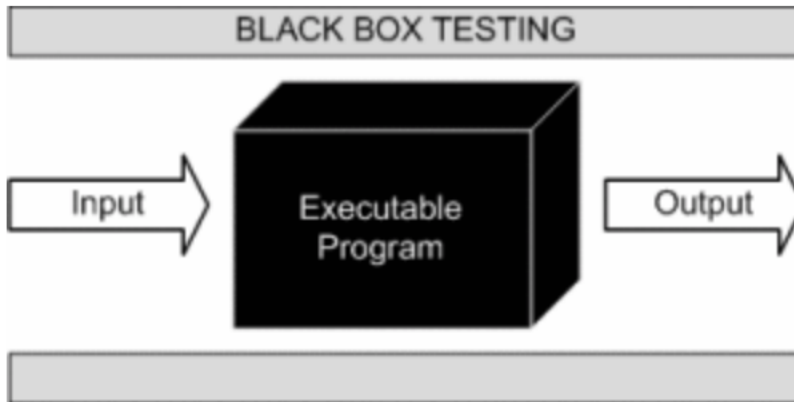
5.1 Software Testing Methods

This section discusses the software testing methods used for this project. The software testing methods used were the black box test and the white box test.

5.1.1 Black Box Test

Black Box Testing, also known as Behavioural Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester (Software Testing Fundamentals, n.d.). It is a test that attempts to find errors in the following categories:

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



The Black Box Testing method was used for the following levels of software testing: Unit Testing → Integration Testing → System Testing.

Unit Testing involves testing the individual components of the application. Integration Testing is testing for detecting flaws in the interaction between the various components. It is performed by the developers themselves or independent testers. System Testing, on the other hand and according to ISTQB (International Software Testing Qualifications Board), involves the process of testing an integrated system to verify that it meets specified requirements (Software Testing Fundamentals, n.d.).

5.1.2 White Box Test

According to ISTQB, white-box testing is testing based on an analysis of the internal structure of the component or system (Software Testing Fundamentals, n.d.). Here, the developer evaluates all the outputs of the components tested and verifies it against the expected outcome by studying the implementation code.

5.1.3 Differences between Black Box Test and White Box Test

Criteria	Black Box Testing	White Box Testing
Definition	Black Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is NOT known to the tester	White Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester.
Levels Applicable To	Mainly applicable to higher levels of testing: System Testing Acceptance Testing	Mainly applicable to lower levels of testing: Unit Testing Integration Testing
Responsibility	Generally, independent Software Testers	Generally, Software Developers
Programming Knowledge	Not Required	Required
Implementation Knowledge	Not Required	Required
Basis for Test Cases	Requirement Specifications	Detail Design

(Software Testing Fundamentals, n.d.)

5.2 Results of Unit Testing

Login

No.	Input Values	Test Case	Expected message / outcome	Test Result
1	Email	Empty	Please enter your Ashesi email	Successful
2	Email	User does not exist / wrong credentials	Sorry, user does not exist or wrong credentials	Successful
3	Password	Empty	Please enter password	Successful

4	Password	wrong password	Incorrect password	Successful
----------	----------	----------------	--------------------	------------

Registration

No.	Input Values	Test Case	Expected message / outcome	Test Result
1	Email	Empty / Not an Ashesi email	Invalid email address	Successful
2	First Name	Empty / invalid name e.g.: name with numbers	Please enter First name	Successful
3	Last name	Empty / invalid name e.g.: name with numbers	Please enter Last name	Successful
4	Password	Empty / does not fulfil password requirement	Password should be at least 6 letters and a number	Successful

Search

No.	Input Values	Test Case	Expected message / outcome	Test Result
1	Major	Selects major	Display all users with selected major	Successful
2	Year Group	Selects year group	Display all users with selected year group	Successful
3	Name	Input first name/last name or both	Display all users with given name	Successful

4	Major/Year Group/ Name	All parameters selected	Display all users who possess similar details	Successful
----------	---------------------------	----------------------------	---	------------

Message

No.	Input Values	Test Case	Expected message outcome /	Test Result
1	Message	Empty	Please enter a message	Successful
	Message	Enters Message	Display on message page	Successful

CV Generator

No.	Input Values	Test Case	Expected message outcome /	Test Result
1	CV details	Enters cv details	Enters database	Successful
2	Preview	Clicks “preview” button	Display CV	Successful
3	Generate PDF	Clicks “Generate PDF” button	Generates CV in PDF format	

5.3 Results of system-level testing

Component	Results	Flaws
Registration	Works as expected	
Login	Works as expected	
Search	All three parameters (major, year group, name) work as expected	Users cannot search on any other part of the application aside this page.

Profile	Works as expected	
Edit Profile	Works as expected	
CV Generator	Performs basic functionality of creating a CV in Ashesi format	No limit on user's input, thus page has no limit. User cannot edit previous CV details
Message	Works as expected	Message textbox should clear input after message is sent.
Homepage	Works as expected	

5.4 Results of user testing

Ten users, five students and five alumni were contacted to test the system. Users seemed satisfied with the application and below were a few suggestions and concerns raised by the testers at the various components.

Registration: A hint should be given to users on the password requirements of the system.

Search page: A notification should be provided when user does not exist in database.

Connect: Users should still be allowed to make the decision of who they will like to accept as friends.

Friends: A page should be provided where users can view all their friends.

Photos: A page where users can view all photos and photos of friends.

5.5 Analysis of test results

The test results demonstrated that the system basically performs its functional requirements however it lacks a few features that satisfies all the needs of its targeted users. The results also showed that though the users will like to connect with others they will also like to have a hand in the decision-making process of who they connect with.

Chapter 6: Conclusions and Recommendations

This section provides a summary of the project, what it entails, and the degree to which it meets its functional requirements. It also states the limitations and flaws of the work. Furthermore, it points out what can be done or added in the future according to the increasing requirements or needs of its direct or potential users.

6.1 Summary of project

This project involved the creation of a social networking site limited to Ashesi Students and Alumni. The project's aim was to bridge the gap between students and alumni. It aimed to build the network of students by connecting them to alumni. It provided a direct link to communicate with alumni on similar interests, such as projects and possible job opportunities.

6.2 Limitations and Recommendations

According to the functional requirements of this project (page 13), this project fulfilled all its requirements. Recommendations or suggestions to further enhance this application will be to add the following functionalities:

- A feature that allows alumni to post job opportunities.
- A feature that allows both alumni and students to post present and past projects.
- A feature that allows users to identify other users based on interests or place of work.
- A feature that allows for video and audio calls.

6.3 Conclusion

In conclusion, this project has a great potential of achieving its aim of connecting Ashesi students with alumni; thus, building a strong network. It also has the potential of becoming a frequently used application by the entire Ashesi community (past and present).

References

- 7summits. (2018, February 05). Students Admissions Community . Retrieved from 7summitsinc.com: <http://www.7summitsinc.com/case-studies/msoe-admissions/>
- Ashesi Foundation. (2018, February 05). Ashesi History. Retrieved from Ashesi Foundation: <http://www.ashesi.org/about/history/>
- Bootstrap. (2018, March 19). *Getting started*. Retrieved from Bootstrap: <https://getbootstrap.com/docs/3.3/getting-started/>
- Boyd, D. M., & Ellison, N. B. (2007, October). Social Network Sites: Definition, History, and Scholarship. Retrieved from Wiley Online Library: <http://onlinelibrary.wiley.com/doi/10.1111/j.1083-6101.2007.00393.x/full>
- Carr, D. F. (2013, July 31). Universities Create Their Own Social Networks For Students. Retrieved from information week: <https://www.informationweek.com/universities-create-their-own-social-networks-for-students/d/d-id/1110997>
- Computer Hope. (2017, 10 11). HTML. Retrieved 10 29, 2017, from computerhope.com: <https://www.computerhope.com/jargon/h/html.htm>
- Connally, T. M., & Begg, C. E. (2005). Chapter 1: Introduction to Databases. In T. M. Connally, & C. E. Begg, *Database Systems* (4th Edition ed., p. 15). Harlow: Pearson Education Limited
- Hale, B. (2015, June 16). The History of Social Media: Social Networking Evolution! Retrieved February 02, 2018, from History Cooperative: <http://historycooperative.org/the-history-of-social-media/>

Herborth, C. (2010, August 10). *Create PDFs on the fly using TCPDF*. Retrieved from IBM.com:
<https://www.ibm.com/developerworks/library/os-tcpdf/index.html>

Homer, A. (2018, February 10). *Components and Web Application Architecture*. Retrieved from
Microsoft: <https://msdn.microsoft.com/en-us/library/bb727121.aspx>

IBM Knowledge Center. (2018, February 10). *Components of web-based applications*. Retrieved
from IBM Knowledge Center:
https://www.ibm.com/support/knowledgecenter/en/SSEPEK_10.0.0/intro/src/tpc/db2z_componentsofwebapplications.html

Phillips, S. (2007, July 25). A brief history of Facebook. Retrieved from The Guardian :
<https://www.theguardian.com/technology/2007/jul/25/media.newmedia>

PHPMailer. (n.d.). *Introduction*. Retrieved from GitHub:
<https://github.com/PHPMailer/PHPMailer/wiki/Tutorial>

PHPMailer. (n.d.). *PHPMailer - A full-featured email creation and transfer class for PHP*.
Retrieved from GitHub: <https://github.com/PHPMailer/PHPMaile>

phpmyadmin. (2003). *About*. Retrieved from phpmyadmin: <https://www.phpmyadmin.net/>

Software Testing Fundamentals . (n.d.). White Box Testing. Retrieved from Software Testing
Fundamentals : <http://softwaretestingfundamentals.com/white-box-testing/>

Software Testing Fundamentals. (n.d.). Black Box Testing. Retrieved from Software Testing
Fundamentals: <http://softwaretestingfundamentals.com/black-box-testing/>

Software Testing Fundamentals. (n.d.). Differences Between Black Box Testing and White Box
Testing. Retrieved from Software Testing Fundamentals:

<http://softwaretestingfundamentals.com/differences-between-black-box-testing-and-white-box-testing/>

Software Testing Fundamentals. (n.d.). System Testing . Retrieved from Software Testing Fundamentals: <http://softwaretestingfundamentals.com/system-testing/>

Sommerville, I. (2006). Agile Software Development. In I. Sommerville, *Software Engineering* (9th Edition ed., p. 57). Boston: Addison-Wesley.

Sommerville, I. (2006). Architectural Design. In I. Sommerville, *Software Engineering* (9th Edition ed., p. 155). Boston: Addison Wesley.

Sommerville, I. (2006). Architectural Design. In I. Sommerville, *Software Engineering*. Boston: Addison Wesley.

Sommerville, I. (n.d.). The waterfall model. In I. Sommerville, *Software Engineering* (9th Edition ed., pp. 30-32). Boston: Addison-Wesley.

Sublime Text. (2018, February 10). *Sublime Text*. Retrieved from Sublime Text: <https://www.sublimetext.com/>

Tatroe, K., MacIntyre, P., & Lerdorf, R. (n.d.). Chapter 1: Introduction to PHP. In K. Tatroe, P. MacIntyre, & R. Lerdorf, *Programming PHP* (3rd Edition ed., p. 1). Sebastopol: O'Reilly Media, Inc.

tcpdf. (2004). *TCPDF*. Retrieved from TCPDF.org: <https://tcpdf.org>

The Economist. (2016, November 10). Retrieved February 02, 2018, from Informal Innovation :
<http://informalinnovation.economist.com>

Tucker, J. (2017, November 02). Risks of accepting strangers' Facebook friend requests. Retrieved February 02, 2018, from WPSD Local 6: <http://www.wpsdlocal6.com/2017/11/02/risks-accepting-strangers-facebook-friend-requests/>

Visual Paradigm. (2018, March 05). *What is Sequence Diagram?* Retrieved from Visual Paradigm:
<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>

w3schools. (2018). *Bootstrap Get Started*. Retrieved from w3schools.com:
https://www.w3schools.com/bootstrap/bootstrap_get_started.asp

Appendix

A. Requirement gathering

A.1 Interview Questions

Below is a list of the questions that were asked in the interview to gather the requirements for this project.

Alumni

- What do you think about an Ashesi-only Social Networking site?
- Will you be interested in using such a site? Why or why not?
- What features will you like to see in an Ashesi University College social networking site?
- How do you think the current students of Ashesi can benefit from this site?

Students

- What features will you like to see in an Ashesi University College social networking site?
- Why will you like to use the site?
- Will you be interested in interacting and collaborating with alumni on projects? Why?
- What do you think about a CV generator and how it will impact your life in school.

Career Services

- What do you think about an Ashesi Social Networking Site for Ashesi student and alumni?
- What features will you like to have on this site?
- What are the major problems involved in editing Students CVs?
- What do you think about an Ashesi CV generator?
- How might we further connect or bridge the gap between Ashesi Students and Alumni?

A.2 Interview Observations & Insights

The results of the interview were positive. All parties interviewed were intrigued about the idea of an Ashesi Social networking site. Most students expressed more interest in the CV generator functionality as they believed it provided an easy way to create an accurate CV approved by the Ashesi career services.

Other students were interested in the notion that the site provided a way to communicate with alumni including alumni they probably did not meet in Ashesi. They believed and agreed it could allow for collaboration and assistance on projects and possible job opportunities.

Seven out of Ten alumni interviewed expressed interest in the social networking site. An individual's reason to use the site was to contact friends whose names he had forgotten- this is in relation to the search page as the search page allows users to search by name, year group and major.

Career Services believed that the site will make their work easier as it builds and networks of students and most especially reduces the work of checking everyone's CV. They also recommended a page that allows alumni to post jobs and a feature that allows both alumni and students to post projects they have worked on or are currently working on.