ASHESI

ASHESI UNIVERSITY

CAPSTONE MANAGEMENT SYSTEM

APPLIED PROJECT

B.Sc. Management Information Systems

Philip Owusu-Afriyie

2020

ASHESI UNIVERSITY

CAPSTONE MANAGEMENT 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.

Philip Owusu-Afriyie

2020

DECLARATION

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

Candidate's Signature:

……………………………………………………………………………………………

Candidate's Name:

……………………………………………………………………………………………

Date:

……………………………………………………………………………………………

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

Supervisor's Signature:

……………………………………………………………………………………………

Supervisor's Name:

……………………………………………………………………………………………

Date:

……………………………………………………………………………………………

Acknowledgements

Firstly, and most importantly I would like to thank the Lord almighty for providing me the strength and knowledge to complete this project. I would like to express my gratitude to Mr. David Sampah for his guidance and advice in carrying out my applied project.

I also give thanks to my family and friends for their unmatched love and support.

Abstract

A Capstone is a project carried out by final year students usually before graduating or completing a course of study. In Ashesi University, a capstone project can either be a thesis, an applied, or an entrepreneurship project. The absence of a centralized system often results in a constant back and forth between students and faculty members. While the current method of operation works, it involves an appreciable amount of paperwork, which is an ineffective way to handle data.

A possible solution presented in this paper is a capstone management system that holds the ability to streamline the entire capstone process between students and faculty members. An essential feature discussed is a feature that makes capstone-worthy projects available to all students. This is especially beneficial to students with little to no idea of projects to pursue.

Table of Contents

List of Figures

Chapter 1: Introduction

Every final year college student looks forward to the day when they put on their black robes and lift their hats off to mark the achievements over their years of study. Prior to this day however, most students are required to complete a project to demonstrate knowledge accrued in their field of study. "*One aspect of the degree program that has received particular attention is the capstone project, a culminating experience in which students are expected to integrate, extend, critique and apply the knowledge gained in the major*" [7]. A capstone project is a hands-on course carried out by final year students pursuing their undergraduate or post-graduate degrees [6]. In addition, it of worthy note that some high school students undergo this process before University. A capstone project is the zenith of the education of a college student which fundamentally seeks to provide the opportunity to express and demonstrate knowledge and skill. As per the scope of an Ashesi University education, a capstone project can take shape in the form of these three: Entrepreneurship; students work in groups as they go through all the aspects of starting a scalable business, a thesis project; the student either chooses to work alone or to be in a group to conduct intrinsic research into any area of interest and applied project; students either work in groups or alone as they solve typical real-life projects [6].

In many instances, several students face an arduous task of making a cogent decision on the capstone topic to work on. Sometimes, students are unprepared to answer questions such as where to start from, what exactly a capstone project is, what is involved in a capstone project, how to select topics, whom to select as a supervisor, how large does the Capstone project need to be.

In Ashesi University, for example, interactions with a number of computer science faculty members revealed the validity of the above statement. Amidst the confusion on selecting topics, one would notice that some faculty have capstone-worthy projects in mind.

Nevertheless, they are unable to contact the right (in terms of interest and skill) students to carry them out. In its success, this project will not only help students answer the above questions but will also make the entire capstone process much more convenient and easier for both students and supervisors.

1.1 Benefits of the System

The system is meant to be used by students, faculty and the overall capstone coordinator. Users of the system should be able to perform the following basic activities:

Students

1. Register on the system
2. Create and edit profile
3. View dashboard
4. Propose topics for capstone depending on interest.
5. Select a maximum number of preferred supervisors to work with in the order of most preferred to least preferred.
6. Express interest to work on faculty proposed project
7. Capstone progress tracking
8. Submit final work: separate abstract, main work, references, etc.

Faculty

1. Register on the system
2. Create and edit profile
3. View dashboard
4. Publicize projects to attract interested students.
5. Approve / Decline topics proposed by students depending on interest.
6. Set capstone activity and monitor process

Capstone Coordinator

1. Put a limit on the total number of students each supervisor is allowed to have.
2. Manage the entire capstone process
3. Communicate with students and faculty.

1.2 System Benefits

The Capstone Management System (CMS) is ultimately beneficial to students and faculty.

In advance, students can select the faculty they would want to work with and vice versa. Faculty members also get the opportunity to review a topic or project proposed by a student before accepting to work with that student. The system can serve as a central point for the dissemination of capstone related information, while streamlining the capstone process between supervisors and students.

The scenarios given below elucidate how the system can be beneficial to both students and faculty.

Scenario 1

Dr. Appiah has available projects he would want to work on but has difficulty finding prospective students whose interests fall within the scope of these available project. With the aid of the CMS, Dr. Appiah can communicate the details of his projects to determined final year students on the system with a broad range of interests. He is also able to make public his interests and preferences in the hopes of being approached by students with similar interests.

Scenario 2

Kwasi is a final year student with broad interests across multiple fields. Due to this, the selection of a capstone topic is taking longer than usual. However, with the aid of the CMS, Kwasi has access to a number of available projects posted by Dr. Appiah. Kwasi immediately feels interested and challenged to take on the project.

In addition to the two scenarios above, the design and implementation of the System will allow for the monitoring and management of the entire capstone process.

1.4 Motivation

The initial introduction to the idea of a CMS was from my supervisor; Mr. David Sampah. Nevertheless, the urge to pursue this project was as a result of student experiences similar to scenario 2 in chapter 1.3. Due to indecisiveness, initial ideas for my capstone project changed twice over a period of a month. This challenge made the need for such a system more plausible because, in its success, faculty with ideas for capstone projects will have the opportunity to reach a broader number of students for supervision.

1.5 Related Work

Management systems, as the name suggests, are primarily used to manage, oversee or monitor the processes and procedures of a particular task or sequence of activities in an organization. When successful, management systems make these processes easier. Two of such existing management systems are discussed below:

a. **Dynamic Project Management Software**

This was a project carried out by MCI Corporation, a telecommunications company based in Washington DC. The purpose of this project was to come up with a system that has the capabilities to both manage and monitor the activities and processes of a typical project. "*A project planning tool is used to effect the project plan including a plurality of tasks to be performed by the users in accordance with respective time schedules*" [4]. Similarly, an important feature of the CMS is for faculty to make tasks available to students. Tasks can be considered as milestones in the capstone process. Where each task submitted is essentially a milestone achieved. Examples of such tasks include *submission of References, submission of Requirements Analysis or submission of a Concept Report. Upon completion of tasks, files will be sent back to the faculty for review.*

FIG. 1

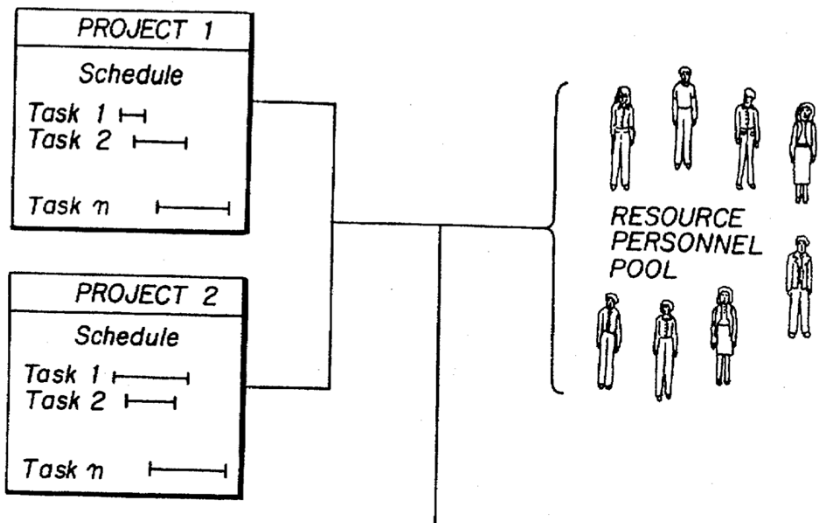


Figure 1: [4] People & Projects

Figure 1.0 above shows several people who are to engage in take on some projects. This can be likened to entrepreneurship projects since projects are worked on in teams or groups. The main difference between an entrepreneurship project and a thesis or applied project is that while students undertaking entrepreneurship projects are expected to works in teams, thesis or applied project students are expected to work solo on a selected project.

b. **Web-Based Project Management System**

This was a project management system invented by Harry A. Frolick and Robert M. Wilson in the early 2000s. The invention aimed to create a management system with a shared member workspace for several team members to collaborate on any project [1].

An essential feature of the Web-based management systems is for the project manager to create a specific task. Under the task created, the project manager provides a description of the task, a title for the task, and the status of the task. This feature can be directly likened to a faculty member creating a new Activity or a new Milestone for his or her students on the CMS. Similarly, the faculty, acting as the project manager in this instance, provides a title and a description amongst other features that are not discussed in the Web-Based Project Management System.

Chapter 2: Requirements Specification

This chapter provides a detailed description of the functionalities of the proposed Capstone Management System (CMS).

2.1 Project Scope

This project involves building a management system to primarily monitor and accompany all activities involved in the capstone process. The system is designed for 3 types of users: students, faculty and the computer science (cs) coordinator. The geographical scope of this project is the Ashesi University campus. This project is important because it solves two pressing needs; it simplifies the entire capstone process and presents students with the opportunity to work on interesting projects that have been proposed by faculty.

2.2 System Features

2.2.1 Functional Requirements

When building software, the functional requirements simply communicate how the system will behave, its features and its functions. Specifications for each user have been provided below. The users mentioned here are students, faculty and the computer science (cs) coordinator.

Students

1. Sign up and log in to the system.
2. Control items and manipulate information on their dashboard.
3. Propose several capstone topics.
4. Select a maximum number of faculty members of interest to supervise capstone.
5. View tasks on the capstone topic.
6. View meeting details with the faculty member.

Faculty

1. log in to the system.
2. Control information on the dashboard.
3. Post possible student projects.
4. Accept capstone projects suggested by students.
5. Declare their interests to work with a student

Coordinator

1. Limit the number of students each faculty can take on.
2. Add a new faculty member
3. View the number of students each faculty member is working with, the names of the students and the project details.

<u>Additional Features</u>

1. Students and faculty should be able to send messages/mail to each other on the platform.

2. Show activities/milestones for each project being worked on.

2.2.2 Non-Functional Requirements

Usability:

1. The system will be easy to use and adapt to. To allow this, the User Interface / User Experience (UI/UX) of the system will make use of icons and color schemes which can be found on applications popularly used by students.
2. Users of the system would be able to accomplish simple tasks and navigate their way without any difficulty.

3. Feedback needs to be provided to the user after the accomplishment of any task to eliminate any confusion. That is, the system needs to engage the customer when performing any functionality.
4. The system will be deployable on all platforms. Including but not limited to smartphones, laptops, desktops, etc.

Security:

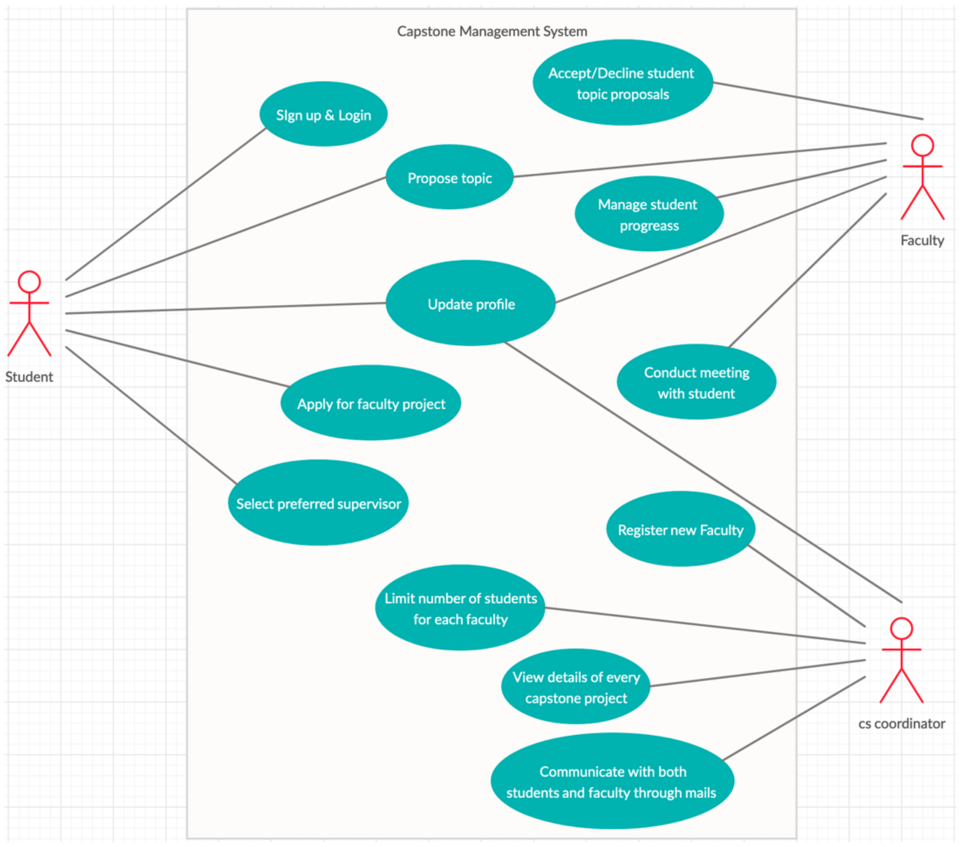
1. Users should be restricted to sections of the system depending on their level of authorization. For example, a student should not be able to view the dashboard of a faculty member when logged in.

Reliability:

1. The system will be void of errors and failures.
2. The system needs to provide accurate and real-time information per change made.

2.3 Use Case Diagram

"The purpose of a use case diagram in Unified Modelling Language (UML) is to demonstrate the different ways that a user might interact with a system" [25]. As discussed in chapter 2.2.1, there are 3 users of the system which translates to 3 actors of the use case diagram. Each functionality mentioned is considered as a scenario.


Capstone Management System
Accept/Decline student topic proposals
SIgn up & Login
Propose topic
Manage student progreass
Update profile
Conduct meeting with student
Apply for faculty project
Select preferred supervisor
Register new Faculty
Limit number of students for each faculty
View details of every capstone project
Communicate with both students and faculty through mails
Student
Faculty
cs coordinator
Figure 2: Use case diagram

Chapter 3: Architecture and Design

3.1 Design Specification

3.1.1 Architecture Description

In this chapter, a breakdown of the structure and behavior of the CMS is provided.

3 - Tier Architecture

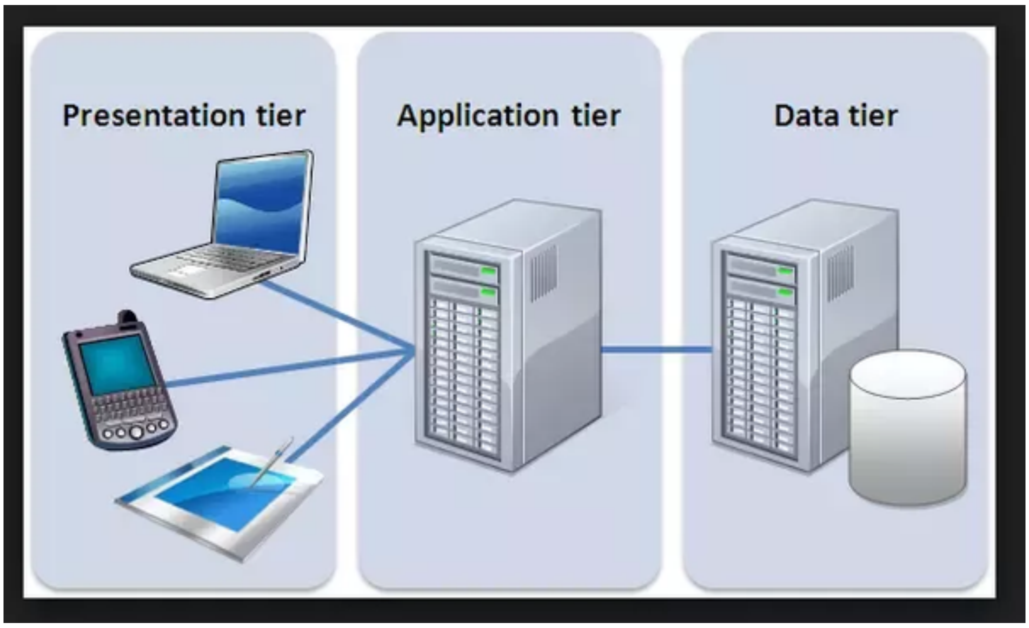


Figure 3: 3-tier architecture

"*A 3-tier architecture is a type of software architecture which is composed of three "tiers" or "layers" of logical computing. The logical processing, access to data and interaction by users are built independently*" [8]. The 3 tiers are presentation layer, application layer and database layer. Implementing this project using the 3 – tier architecture provides a major advantage of making changes or updating any of the tiers individually without touching the others. Additionally, since each layer is built independently, the maintainability of code is tolerable and feasible.

Presentation Layer: This is the uppermost layer of the architecture. It represents the front-end of the application, thus, it is the platform through which users make their interactions. This tier will be built with the aid of basic web technologies such as HTML, CSS, and JavaScript. Frameworks such as Laravel can also be used. For this project, Bootstrap will be used to implement the presentation layer.

Application Layer: Being the middle layer, the application layer acts as the logic of the entire system. All functionalities and processing required for the smooth running of the application are carried out on the application layer. The application layer can be built with Python, Java, PHP, etc. Laravel will be used to implement the application layer.

Database Layer: This is the data store of the entire system. On this layer, information is essentially stored and retrieved for processing in the application layer. MySQL, PostgreSQL, etc. can be used to build this layer. For this project, MySQL will be used.

3.1.2 Design Pattern

Model – View – Controller Architecture

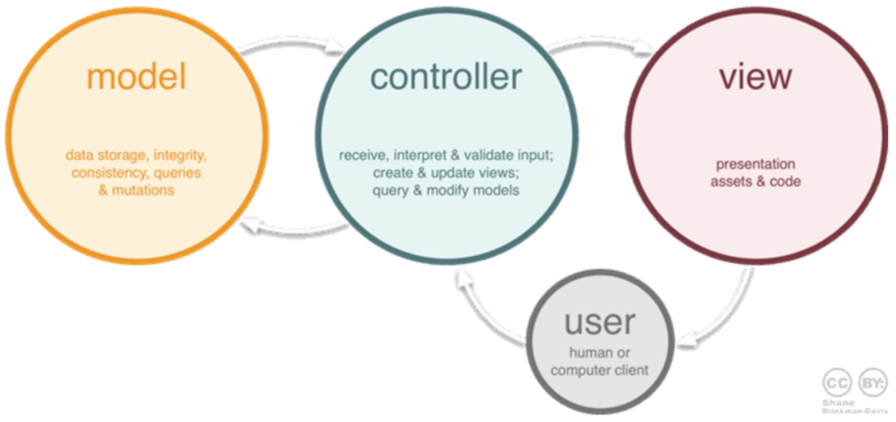

model
data storage, integrity, consistency, queries & mutations
controller
receive, interpret & validate input; create & update views; query & modify models
view
presentation assets & code
user
human or computer client

Figure 4: MVC Architecture (Zeeshan, 2015)

The idea behind MVC architecture is to separate code depending on its purpose. As such, code is grouped depending on 3 categories:

1. Does it help hold some data for the system? (Database)

2. Does it make the application look appealing and user -friendly? (Front- end)
3. Does it handle the functionalities of the application? (Core functionalities)

View: This aspect of the architecture is mainly concerned with how data is represented to the user. It focuses solely on the User Interface/User Experience of the application. It interacts with the controller and dictates how data from the model can best be represented and displayed to the user. The view is also an avenue for users to alter data in the database.

Controller: The controller is responsible for handling and reporting any actions carried out by the user on the application. It communicates with both the view and the model.

Model: The model stores data and handles queries. (A query is a request for data from the database.) The model is also responsible for all the logic of the application that is data related. For this project, MySQL will be used to handle the storage and modification of data. MySQL works with tables and relationships, as such, the creation of a relational database – a database that keeps track of the relationship between its items - can easily be carried out. Some important fields found in the database of this project are listed below:

1. Users – this table contains information on all users (Both students and Faculty).
2. Faculty – this table contains information on all faculty members.
3. Student - this table contains information on all students.
4. Project – this table contains information on all projects. Projects could either be proposed by students or lecturers.
5. Capstone table – this table contains information on students, their corresponding supervisors, the project the students are working on and the starting date of the project.
6. Faculty_student – this table contains information on the faculty member each student chose.
7. Meetings – this table contains information on each meeting held between a student and his or her supervisor.

Below is an entity-relationship diagram of the database for the CMS. An entity-related diagram portrays and describes the relationship between fields in a given database.

Entity Relation Diagram

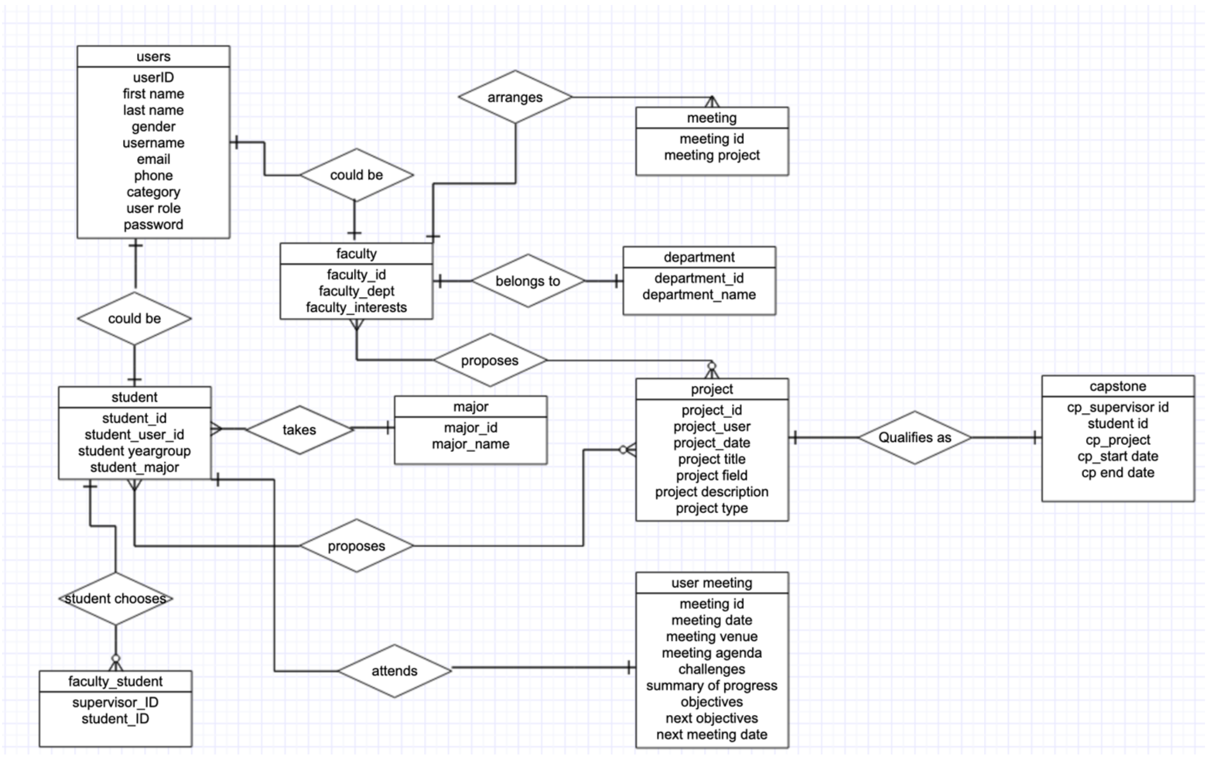


Figure 5: Entity-Relationship Diagram

From Figure 3.3, it can be deduced that a user is likely to be a student or a faculty member. The diagram also shows that both faculty and students can propose projects by providing the title, field, description and type of the project.

3.1.3 Activity Diagrams

The diagram below shows the series of actions a student would go through when proposing a topic and selecting a faculty as a supervisor.

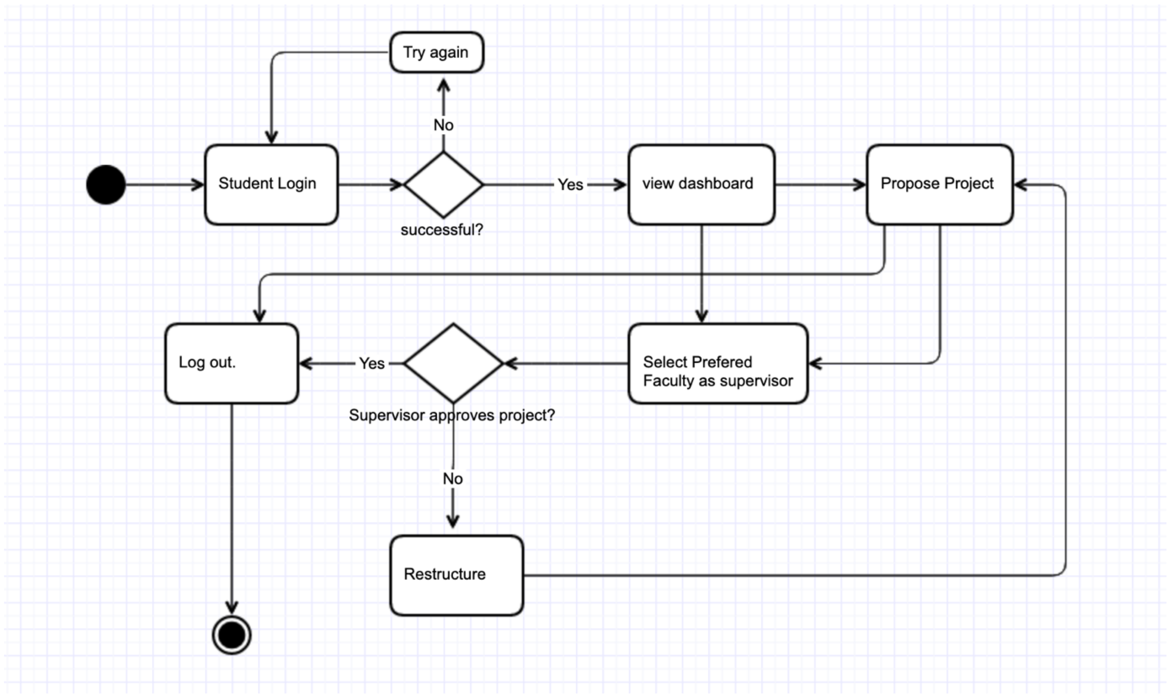


Figure 6: Activity diagram for a student

The diagram below shows the series of activities a faculty member would go through to accept the proposal of a student or to propose a project himself or herself.

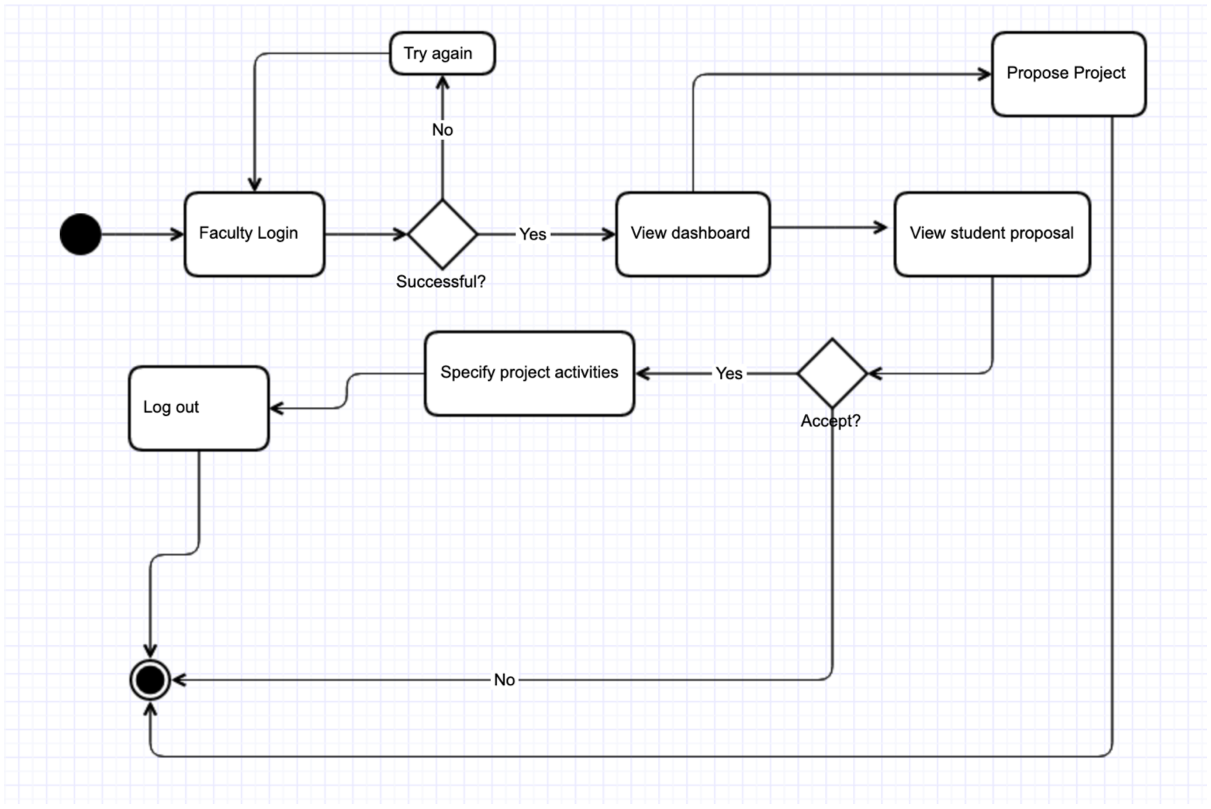


Figure 7: Activity diagram for accepting student projects

3.2 Technology & Framework Used

HTML – "*Hypertext Markup Language is a standard markup language for creating content on-pages of websites*" [9].

CSS – "*CSS is a language that describes the style of an HTML document.*" [10]

JavaScript – "*JavaScript is a programming language that adds interactivity to your website ...*" [11]

Bootstrap – "Bootstrap is a free front-end framework for faster and easier web development."

Laravel– "*Laravel is a powerful MVC PHP framework, designed for developers who need a simple and elegant toolkit to create full-featured web applications.*" [12]

MySQL – "*MySQL is the most popular Open Source Relational SQL Database Management System.*" [13]

Ajax – "*AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page*" [23].

jQuery - "*jQuery is a lightweight, "write less, do more", JavaScript library*" [24]. It simply makes it easy for the use of JavaScript.

Chapter 4: Implementation

This chapter seeks to provide the details of the various functionalities that are implemented in the CMS. It also provides information on processes used during implementation. Two major frameworks were employed in the development of this system. These frameworks are Laravel and Bootstrap.

4.1 Setting Up

To use the Laravel framework, the installer is downloaded using the composer – a dependency manager. Laravel composer is installed from the terminal or command prompt using the code below:

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === 'e0012edf3e80b6978849f5eff0d4b4e4c79ff1609dd1e613307e16318854d24ae64f26d17af3ef0bf7cfb710ca74755a') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
php composer-setup.php
php -r "unlink('composer-setup.php');"
```
[17]

A new Laravel project is then easily created with the command below:

```
composer create-project laravel/laravel your-project-name 4.2. *
```
[16]

Bootstrap, on the other hand, is a Cascading Style Sheets (CSS) framework that makes front-end development easy for developers by providing CSS and JavaScript designs. Bootstrap can be used by simply including its name tags in the header of the HTML page as shown below:

```
<link rel="stylesheet"href="https://stackpath.bootstrapcdn.com/bootstrap
/4.1.0/css/bootstrap.min.css" crossorigin="anonymous">
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/js/
bootstrap.min.js" crossorigin="anonymous"></script>
```

4.2 Project File Structure

- ***Capstone_Management_Systemtapp***
    - *Http*
        - *Controllers*
    - *Model*
- *Bootstrap*
- *Config*
- *Database*
- *Public*
- *Resources*
    - *Views*
- *Routes*
- *Storage*
- *Tests*
- *Vendor*
- *. editorconfig*
- *. env*
- *.gitattributes*
- *. gitignore*

Above is a breakdown of the default file structure for all Laravel projects. As described in chapter 3, the software architecture employed for this project is a Model_View_Controller architecture. The structure above shows how each component can be accessed in the editor.

- ***Model***
    - *Capstone_table.php*
    - *Department.php*
    - *Faculty.php*
    - *Faculty_student.php*
    - *Major.php*
    - *Pending_request.php*
    - *Project.php*
    - *Student.php*

By convention, this model is responsible for executing queries that manipulate the database. For example, should a faculty member add a new project, the query to retrieve and store information related to the project would be located in the model.

- Views
    - Dashboard.blade.php
    - Home.blade.php
    - Layout.blade.php
    - Login.blade.php
    - Milestones.blade.php
    - Profile.blade.php

        - Student_dashboard.blade.php
        - Student_milestones.blade.php
        - Student_topics.blade.php
        - studentLayout.blade.php
        - students.blade.php
        - topics.blade.php
        - viewProject.blade.php
        - welcome.blade.php

The view handles HTML pages to be accessed by the user. The view also allows the user to alter and modify data through inputs and interactions. All user interactions with the system are carried out via the view. The view dictates how data from the model can be organized and displayed to the user.

- Controllers
    - Controller.php
    - FacultyController.php
    - HomeController.php
    - ProjectsController.php
    - StudentController.php

Located under the Http folder, the controllers are responsible for handling requests from the user and sending responses back. The controllers act as middlemen between the model and the view. For example, when a user clicks a button to visit the dashboard of the Capstone Management System, the controller checks whether he or she is logged in or not. The controller then checks for the level of authorization of the user (either faculty or student) and then finally renders the respective dashboard to the user.

Other essential components of the project include the Database folder and the *.env file*. Both can be identified in the project file structure breakdown above. The database folder contains the migrations ("*Migrations are like version control for your database, allowing your team to easily modify and share the application's database schema*". [15]) for the project. Laravel migrations provide version control for the database. Thus, it allows developers to easily make changes to the database.

Details on the configuration of the database and its connection can be found in the *.env file.* These details include the name of the database, the host address of the application, the database port, the database username and finally the database password. In the project file structure breakdown provided above, the location of this file is shown.

4.3 User Interface and Functionality

This system is designed ideally for three types of users; faculty, students and the computer science (cs) coordinator. Although similar, all three categories of people have different functionalities. The design of this system is such that it responds to its environment, taking into consideration screen size and screen orientation.

4.3.1 Login

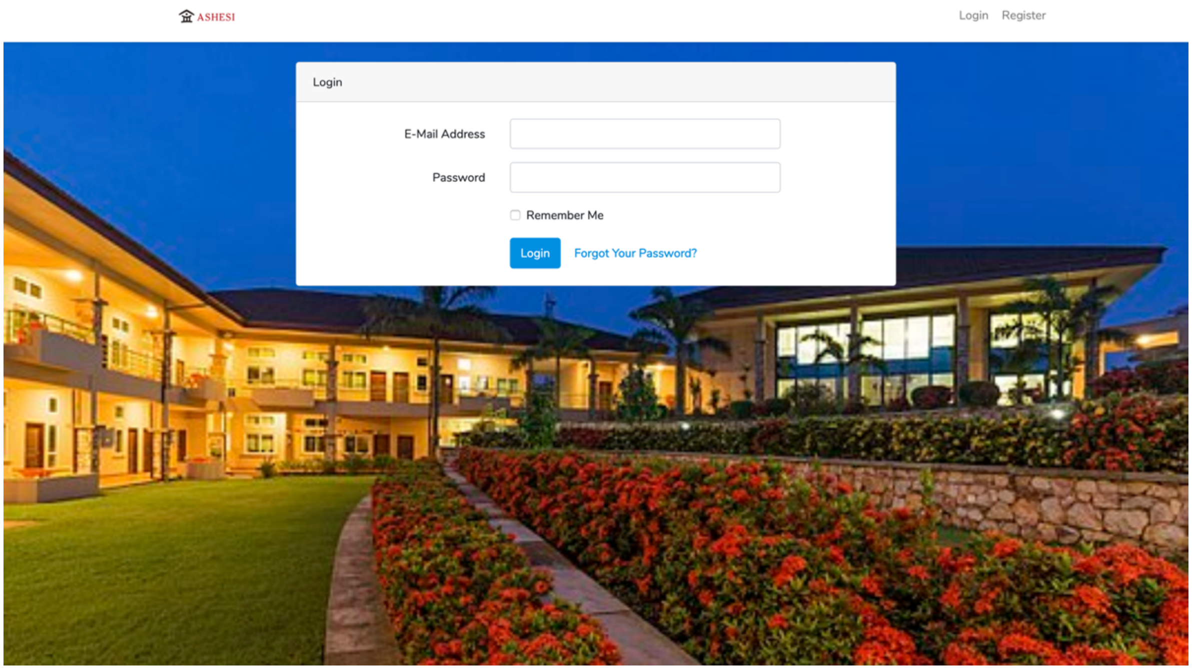


Figure 8: Student login page.

A user logs in with their username and password. Verified users are redirected to their dashboard.

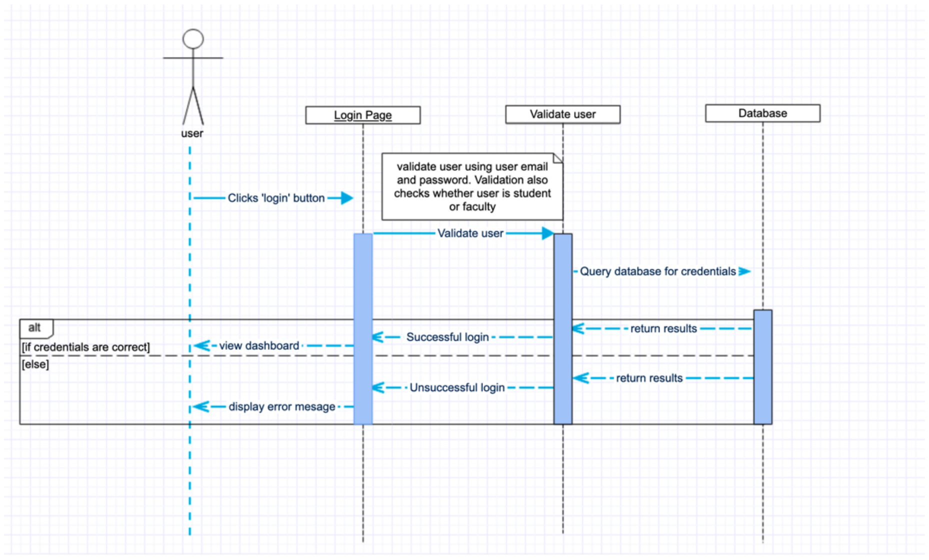
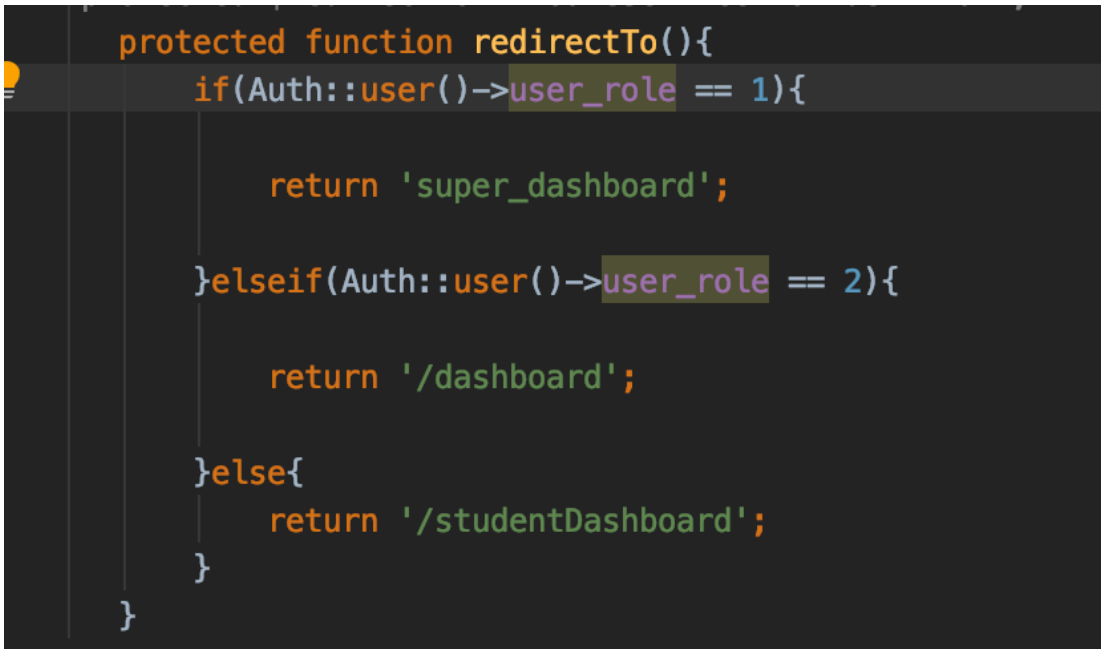


Figure 9: Student login sequence diagram

The credentials of users are crosschecked in the database to ensure validity before they are redirected. Figure 4.2 shows this process.

"*Laravel provides a quick way to scaffold all of the routes and views you need for authentication, using one simple command: PHP artisan make: auth*" [14]. This command reduces the workload of programmers when trying to create a registration, login, verification, and authentication for users of any system. "*This command should be used on fresh applications and will install a layout view, registration and login views, as well as routes for all authentication end-points*" [14]. The image below shows the verification process used when determining whether a user trying to log in is a student, a faculty member or the computer science (cs) coordinator. Depending on their status, the respective dashboard is loaded.


```
protected function redirectTo(){
    if(Auth::user()->user_role == 1){

        return 'super_dashboard';

    }elseif(Auth::user()->user_role == 2){

        return '/dashboard';

    }else{
        return '/studentDashboard';
    }
}
```

Figure 10: Code snippet for authenticating users.

After the credentials of the user have been verified, the role of the user i.e. student, supervisor or coordinator is then checked to identify his or her status. Depending on the role of the user, he or she is redirected accordingly.

4.3.2 Registration

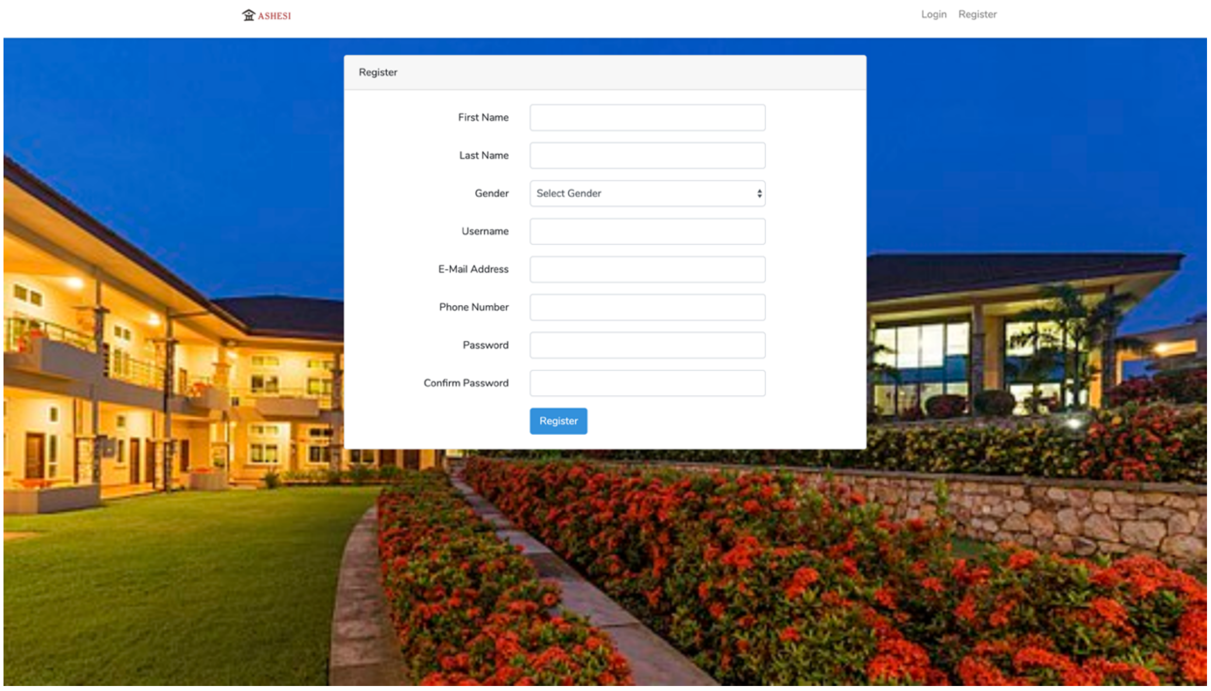


Figure 11: Student registration page

The fields in Figure 4.4 are required of every student undergoing registration.

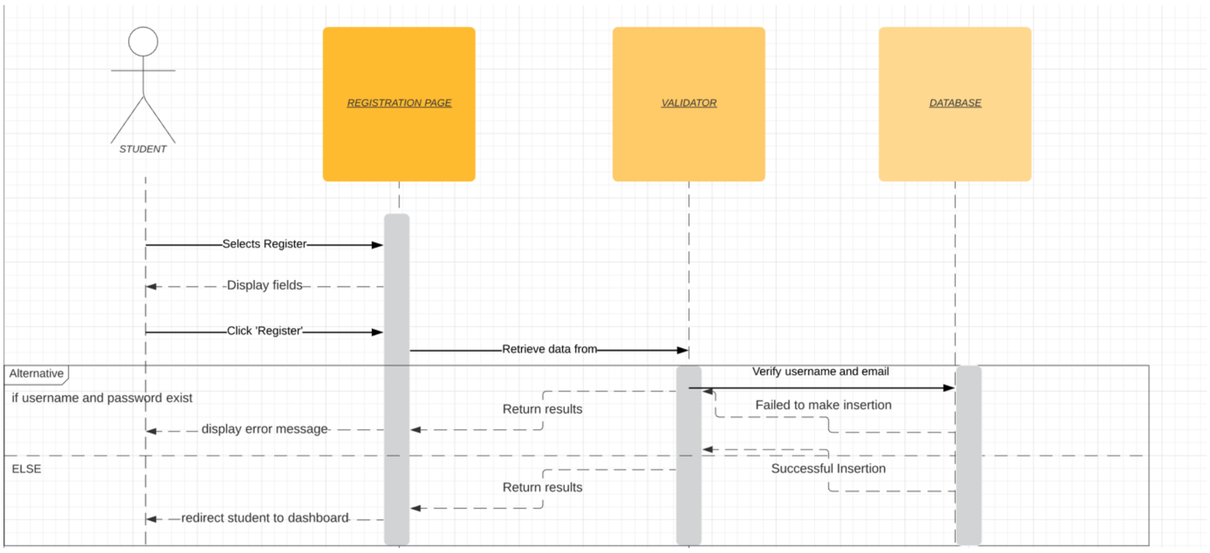
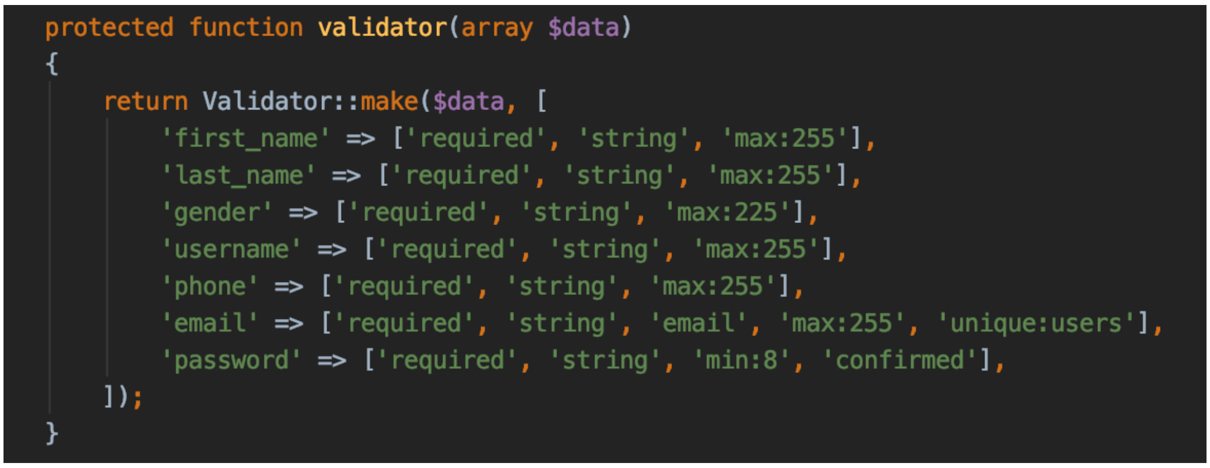


Figure 12: Student registration sequence diagram.

Before any student can successfully register, he or she must produce a unique email and username. Upon successful registration, the student is redirected to the dashboard.

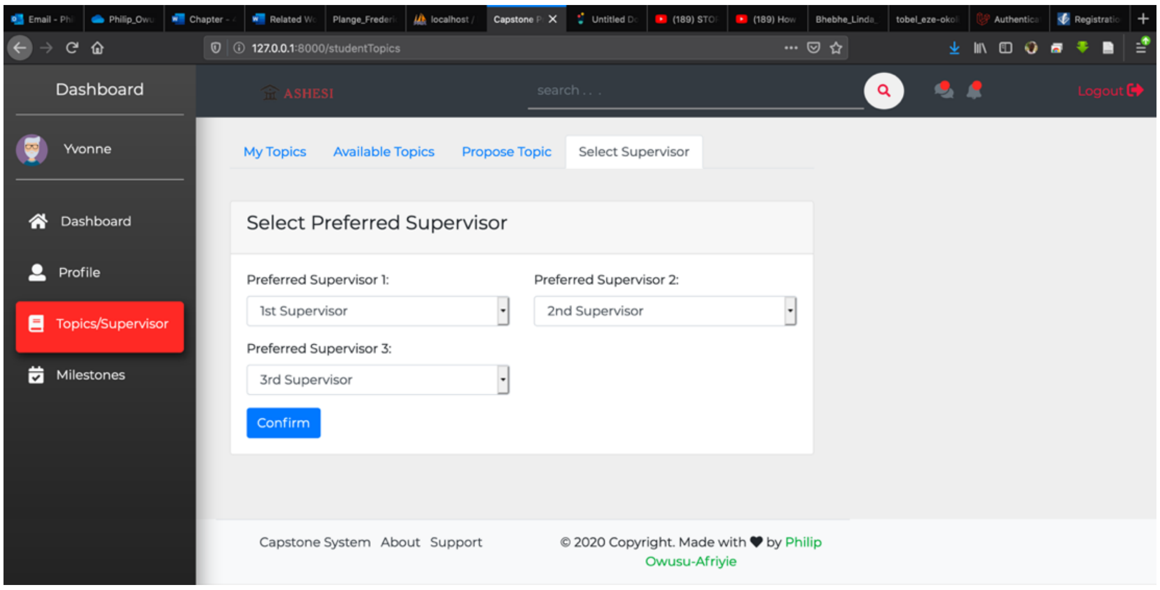


```php
protected function validator(array $data)
{
    return Validator::make($data, [
        'first_name' => ['required', 'string', 'max:255'],
        'last_name' => ['required', 'string', 'max:255'],
        'gender' => ['required', 'string', 'max:225'],
        'username' => ['required', 'string', 'max:255'],
        'phone' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
        'password' => ['required', 'string', 'min:8', 'confirmed'],
    ]);
}
```

Figure 13: Code snippet for creating & registering students.

The code displayed in figure 4.6 ensures that critical fields such as the email and username of the student are unique. In case entries made by the student already exist, he is alerted to make changes.

4.3.3 Student Selecting Topic & Supervisor



Select Preferred Supervisor

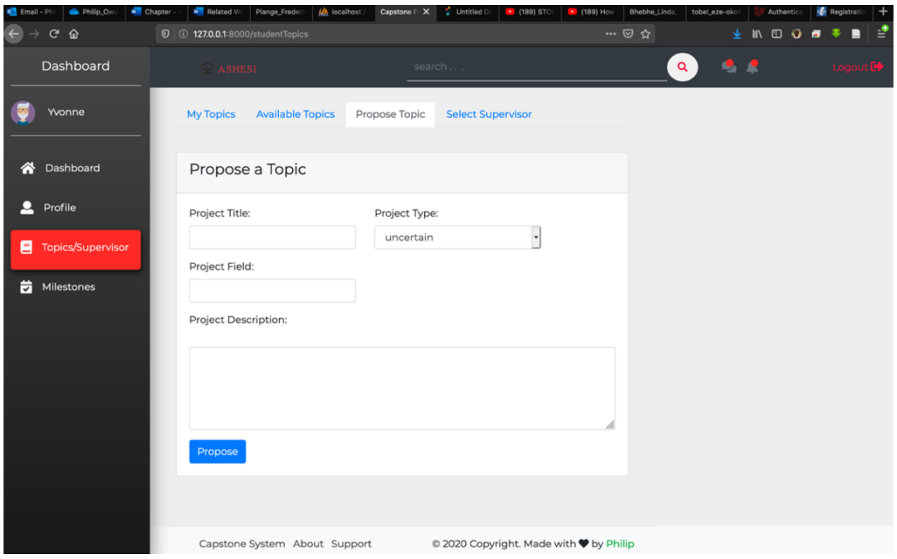
Preferred Supervisor 1: 1st Supervisor

Preferred Supervisor 2: 2nd Supervisor

Preferred Supervisor 3: 3rd Supervisor

Figure 14: Selecting a preferred supervisor.

Students are given the chance to select three (3) supervisors they would prefer to work with on their capstone projects. Ideally, students figure out their capstone topics before selecting the faculty members they would want to work with. Nonetheless, students are also allowed to select their preferred faculty members as supervisors before figuring out their capstone topics. The topics of students will only be shown to the three (3) faculty members they selected. The topics of students will only be shown to the three (3) faculty members selected.



Propose a Topic

Project Title:

Project Type: uncertain

Project Field:

Project Description:

Figure 15: Proposing a topic.

Topic proposals are meant for students who already have an idea for a project in mind. Proposing a capstone topic and selecting preferred faculty members to work with have intentionally been split into two separate tasks. This is to cater for students who know the faculty members they would like to work with but have little knowledge of the specific projects they will work on. Faculty members are also allowed to propose topics to students. Topics proposed by faculty members are available on the dashboards for the viewership of the entire student body. The project only disappears when a student applies for the project and is approved by that faculty member who owns the project.

4.3.4 CS Coordinator Adding New Faculty Member

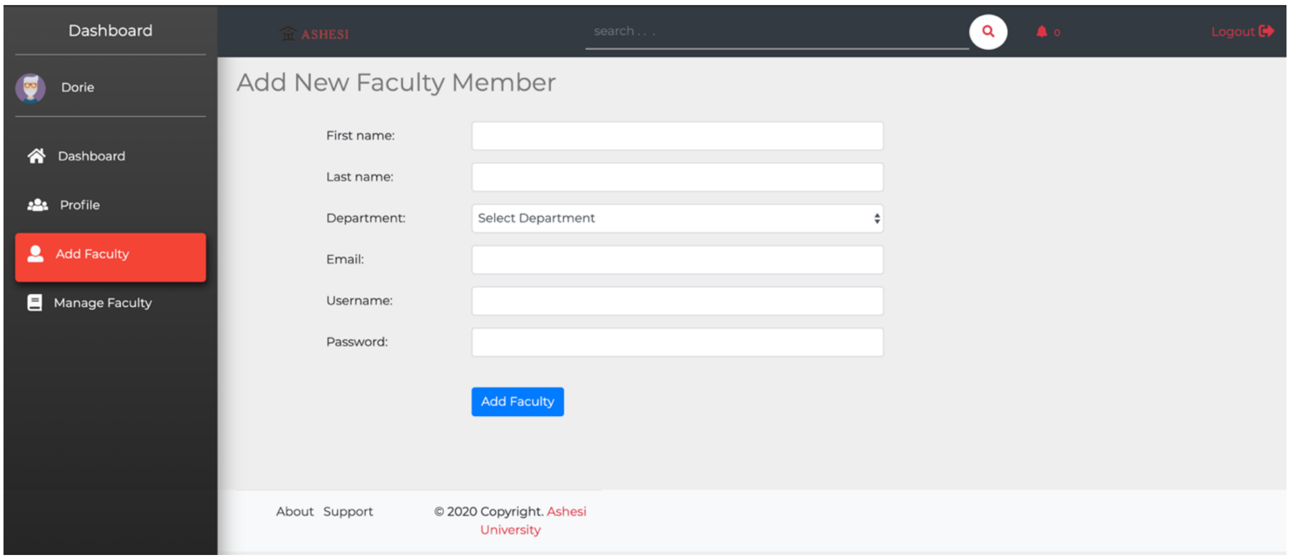
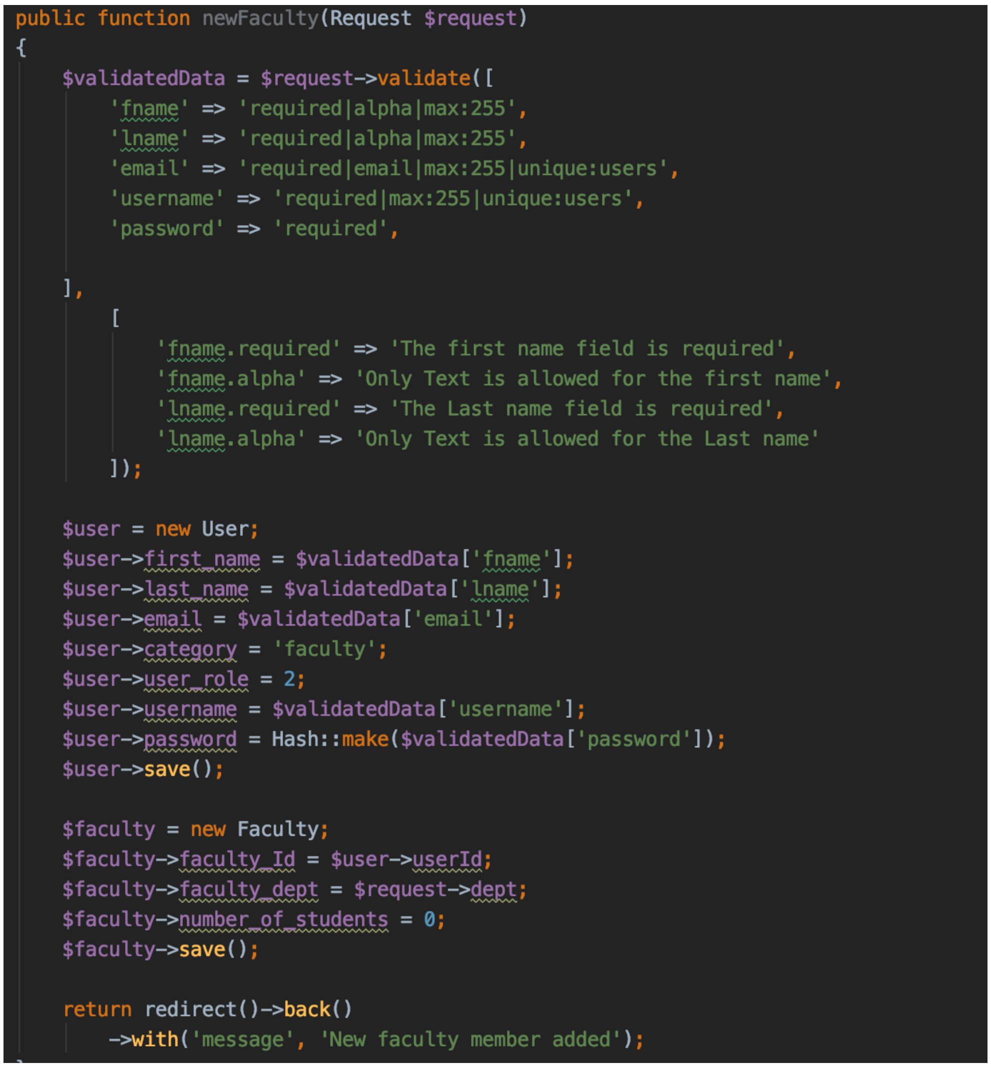


Figure 16: Adding new faculty members.



```
public function newFaculty(Request $request)
{
    $validatedData = $request->validate([
        'fname' => 'required|alpha|max:255',
        'lname' => 'required|alpha|max:255',
        'email' => 'required|email|max:255|unique:users',
        'username' => 'required|max:255|unique:users',
        'password' => 'required',

    ],
        [
            'fname.required' => 'The first name field is required',
            'fname.alpha' => 'Only Text is allowed for the first name',
            'lname.required' => 'The Last name field is required',
            'lname.alpha' => 'Only Text is allowed for the Last name'
        ]);

    $user = new User;
    $user->first_name = $validatedData['fname'];
    $user->last_name = $validatedData['lname'];
    $user->email = $validatedData['email'];
    $user->category = 'faculty';
    $user->user_role = 2;
    $user->username = $validatedData['username'];
    $user->password = Hash::make($validatedData['password']);
    $user->save();

    $faculty = new Faculty;
    $faculty->faculty_Id = $user->userId;
    $faculty->faculty_dept = $request->dept;
    $faculty->number_of_students = 0;
    $faculty->save();

    return redirect()->back()
        ->with('message', 'New faculty member added');
```

Figure 17: Code snippet for adding new faculty members.

Standard checks are made to ensure that the email and username being entered by the computer science (cs) coordinator do not already belong to another faculty member. Each faculty member is assigned the user role of ***2*** to differentiate them from students and the computer science (cs) coordinator. After being signed up, the faculty member can now log in and perform his or her various activities independent of the computer science (cs) coordinator.

4.3.5 Placing Limit on The Number of Students Supervised by Faculty Members

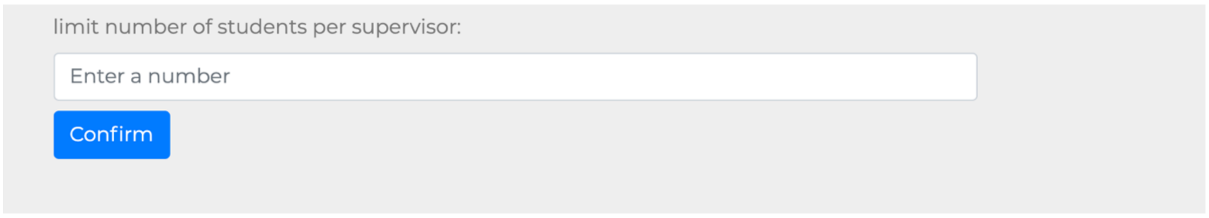
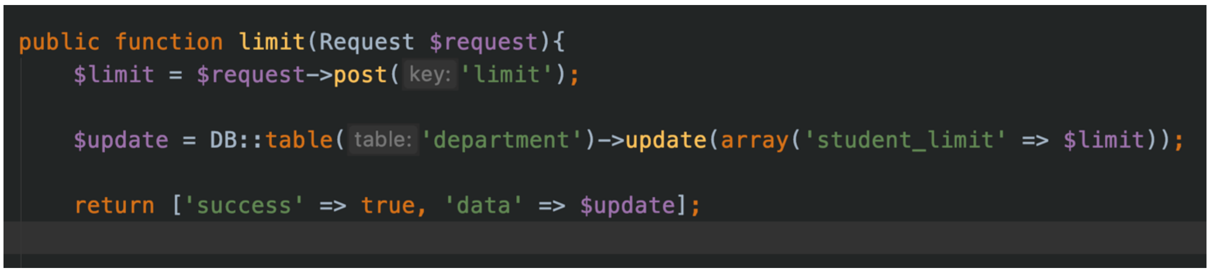


Figure 18: Placing a cap on the number of students.



```php
public function limit(Request $request){
    $limit = $request->post( key: 'limit');

    $update = DB::table( table: 'department')->update(array('student_limit' => $limit));

    return ['success' => true, 'data' => $update];
```

Figure 19: Code snippet for placing a cap on the number of students.

When the number of students supervised by a particular faculty member reaches the limit set by the computer science (cs) coordinator, the particular faculty member is no longer available as an option when other students are choosing their preferred supervisors.

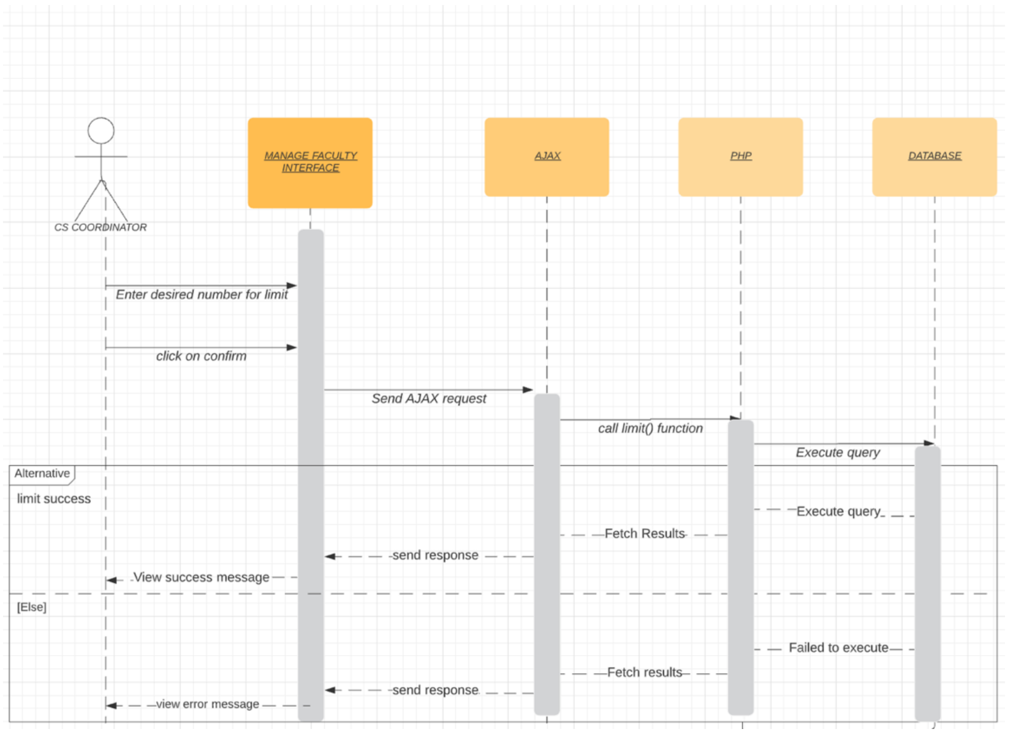


Figure 20: Sequence diagram for placing cap on number of students.

To ensure that faculty members do not supervise a large number of students which might prove counter-productive, a cap is always placed on the number of students each faculty member can supervise. By this means, the students can get the attention and help they need. On this system, after a supervisor has chosen the students to supervise which is equal to the cap placed by the computer science (cs) coordinator, other students will not find that particular faculty member among the list of supervisors.

Chapter 5: Testing

To ensure that a system performs as expected in addition to maintaining an error-free system, it was necessary to put it through several tests. To achieve this feat, three types of testing were carried out on the system; development testing, browser compatibility testing, and user testing.

5.1 Development Testing

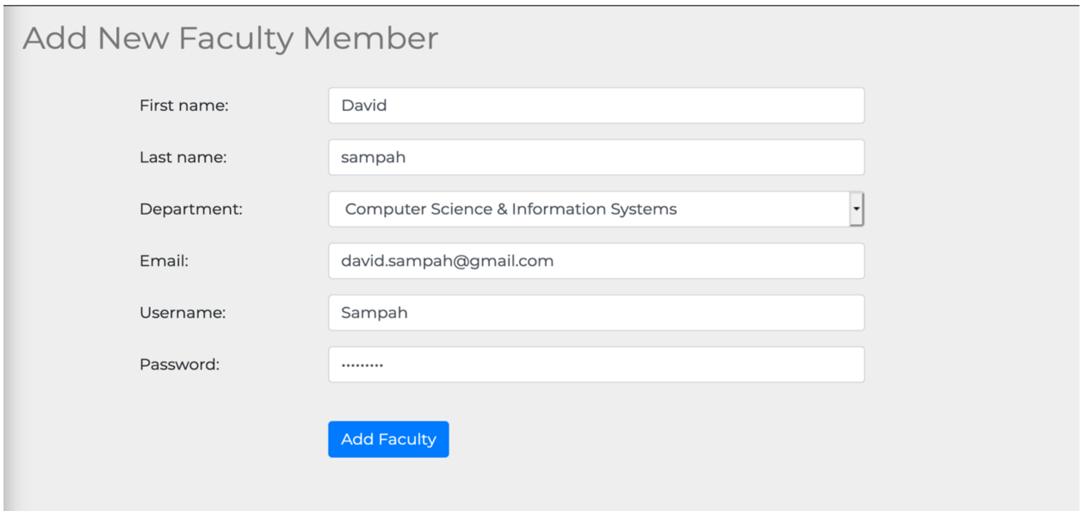
As the name suggests, development testing is carried out in the development phase of an application, and its purpose is to discover bugs and defects. For this system, two types of development testing were employed; unit testing and system testing.

5.1.1 Unit Testing

"Unit testing is a type of software testing where individual units or components of the software are tested." [18]. Unit testing aims to ensure that small isolated parts of the entire code function perfectly before they are assembled and tested as a whole. The main users of the CMS are faculty members, students, and the computer science (cs) coordinator. Functionalities for these three users were grouped and tested accordingly. Each action in the system was performed separately, and the success or failure of the action was confirmed with the help of PHPUnit. "PHPUnit is a programmer-oriented testing framework
" [21]. This was possible because Laravel provides support for testing with PHPUnit. "…support for testing with PHPUnit is included out of the box and a phpunit.xml file is already set up for your application" [19].

A sample of testing is provided below:

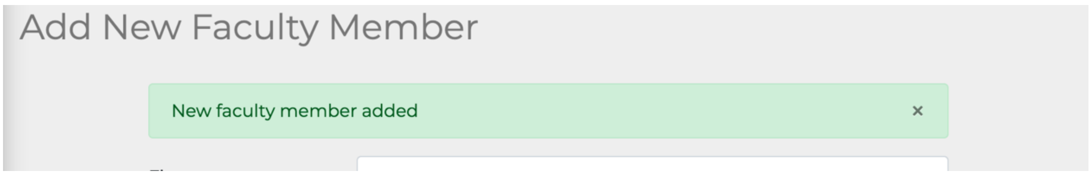
PS: This is a test for the addition of a new faculty member by the computer science (cs) coordinator.



Add New Faculty Member

First name:	David
Last name:	sampah
Department:	Computer Science & Information Systems
Email:	david.sampah@gmail.com
Username:	Sampah
Password:	·········

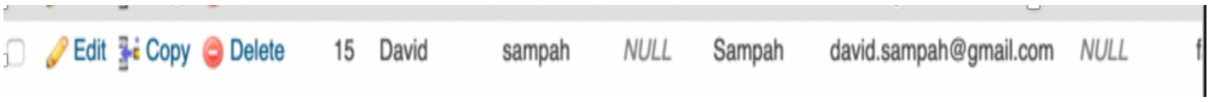
Figure 21: Computer Science (cs) coordinator adding a new Faculty member.



Add New Faculty Member

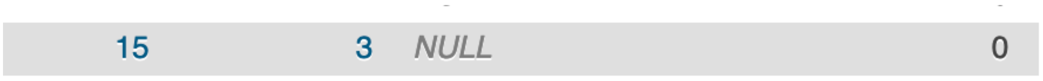
New faculty member added ×

Figure 22: Success message after adding faculty



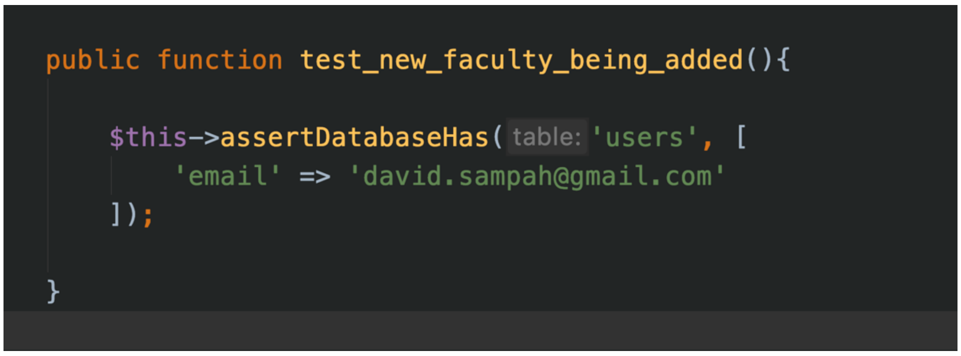
Edit	Copy	Delete	15	David	sampah	NULL	Sampah	david.sampah@gmail.com	NULL

Figure 23: Insertion into the users' table



15	3	NULL	0

Figure 24: Insertion into the faculty table



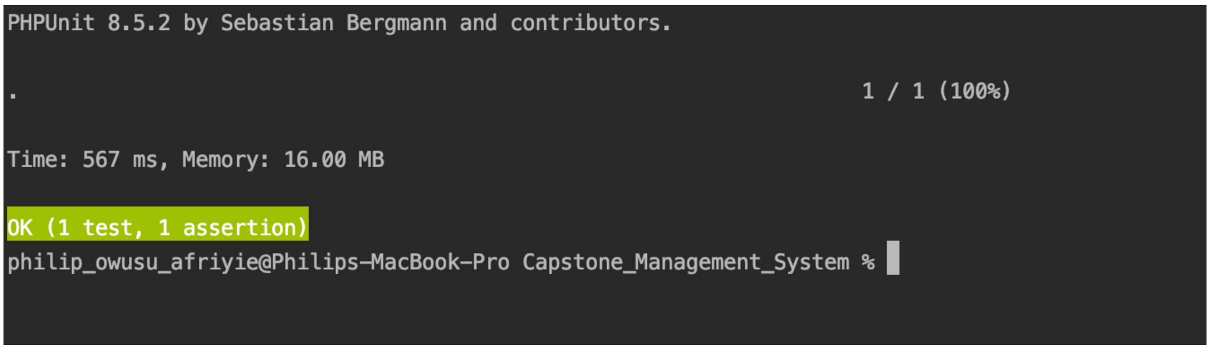
```
public function test_new_faculty_being_added(){

    $this->assertDatabaseHas( table: 'users', [
        'email' => 'david.sampah@gmail.com'
    ]);

}
```

Figure 25: PHPUnit test case

Figure 5.5 Displays a test case written in PHPUnit to check for whether a user with the email david.sampah@gmail.com exists in the users' field in the database.



```
PHPUnit 8.5.2 by Sebastian Bergmann and contributors.

.                                                                  1 / 1 (100%)

Time: 567 ms, Memory: 16.00 MB

OK (1 test, 1 assertion)
philip_owusu_afriyie@Philips-MacBook-Pro Capstone_Management_System %
```

Figure 26: PHPUnit successful test

Figure 5.6 Shows that the test was successful, this shows that the faculty member was successfully added to the database.

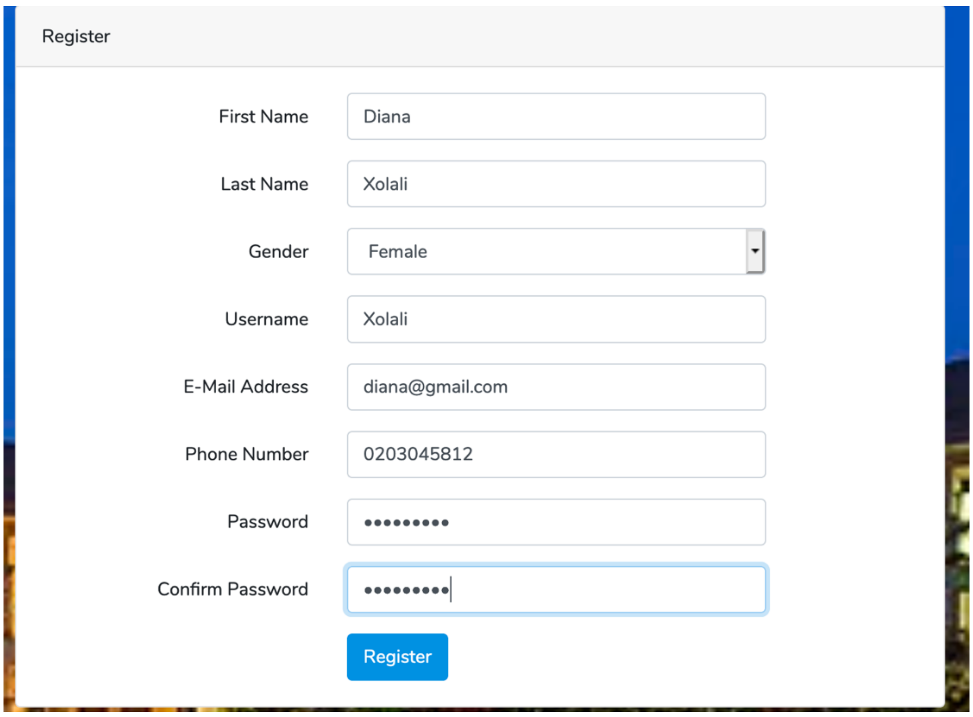
5.1.2 System Testing

"System testing is a level of software testing where a complete and integrated software is tested. [22]" Application or software components are a combination of multiple components that interact with one another. The objective of system testing is to ensure that the interacting components are working as expected. To test this application, a full cycled activity was carried out. Steps to the test are as follows:

- A student registers herself and completes her profile.
- The computer science coordinator registers four faculty members.
- The student proposes a topic and selects three out of the four faculty members as her preferred faculty members.
- The dashboards of the three faculty members are checked to ensure that the project proposal appears, and that of the last faculty is checked to make sure it is empty.

- One faculty out of the three accepts to supervise the student after viewing the full details of the project.
- The student and project information are then added to the list of supervised students of the faculty.
- The faculty member hosts a meeting with the student and specifies the objectives and dates for the next meeting.

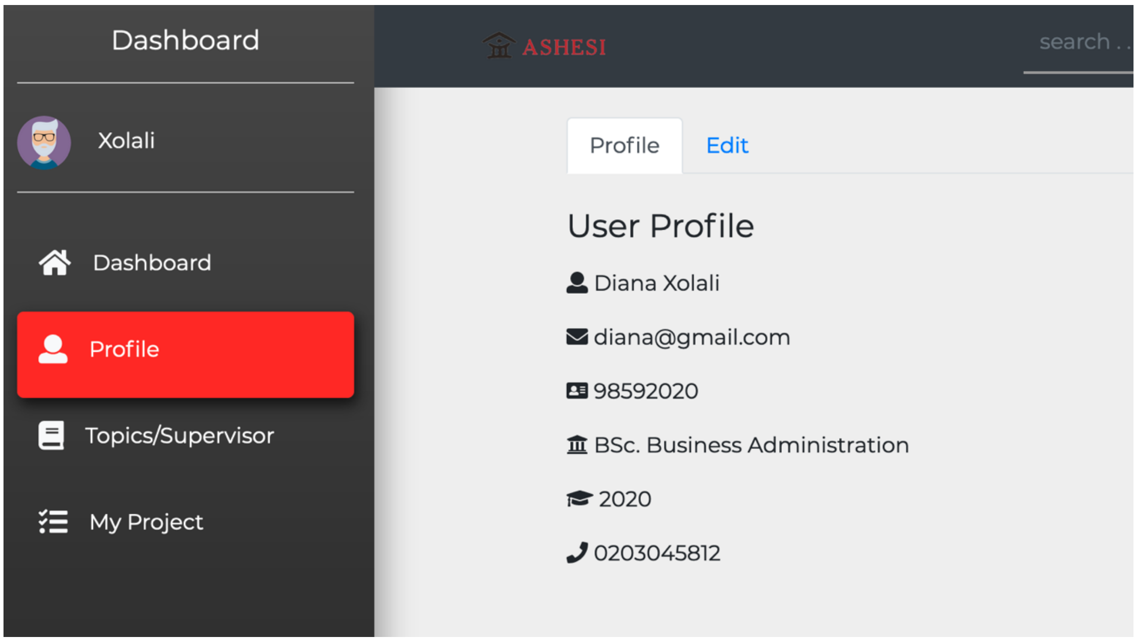
All actions tested executed seamlessly.



Register

First Name	Diana
Last Name	Xolali
Gender	Female
Username	Xolali
E-Mail Address	diana@gmail.com
Phone Number	0203045812
Password	•••••••••
Confirm Password	•••••••••

Figure 27: Student registering



User Profile

Diana Xolali

diana@gmail.com

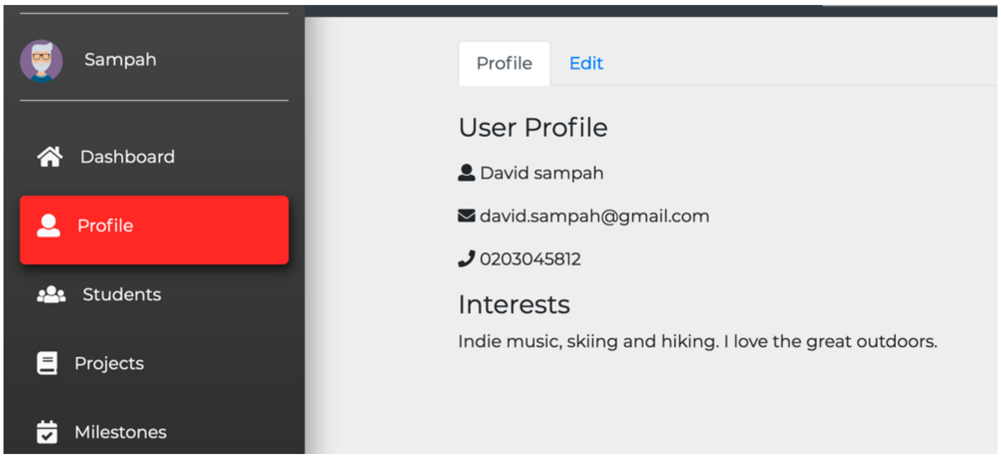
98592020

BSc. Business Administration

2020

0203045812

Figure 28: Successful student register



User Profile

David sampah

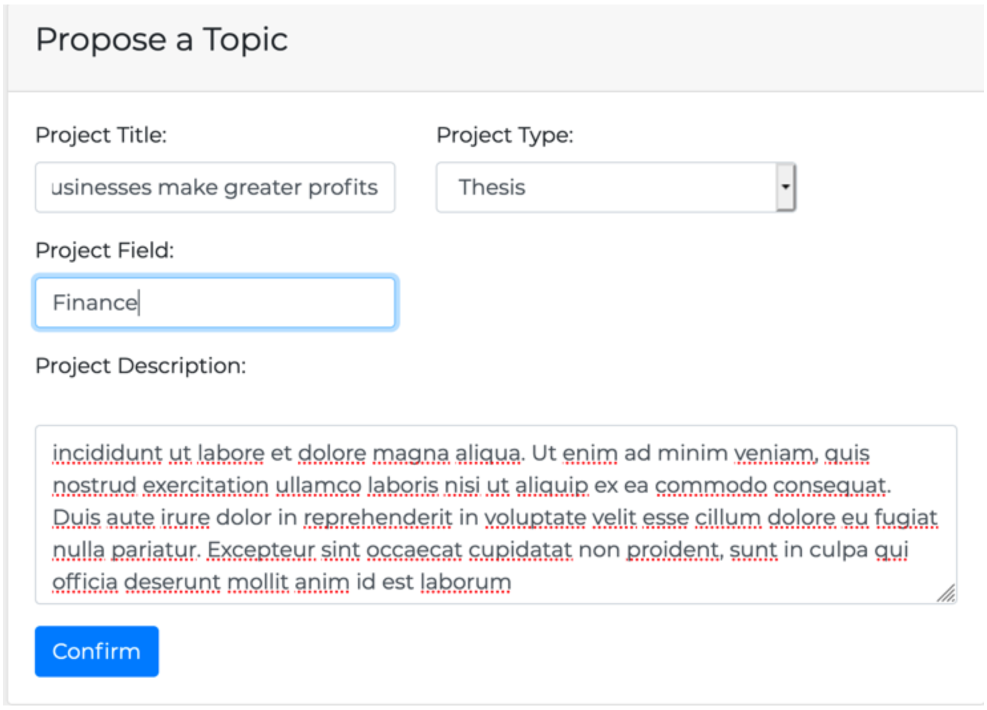
david.sampah@gmail.com

0203045812

Interests

Indie music, skiing and hiking. I love the great outdoors.

Figure 29: Faculty member registered by cs coordinator


Propose a Topic

Project Title: usinesses make greater profits

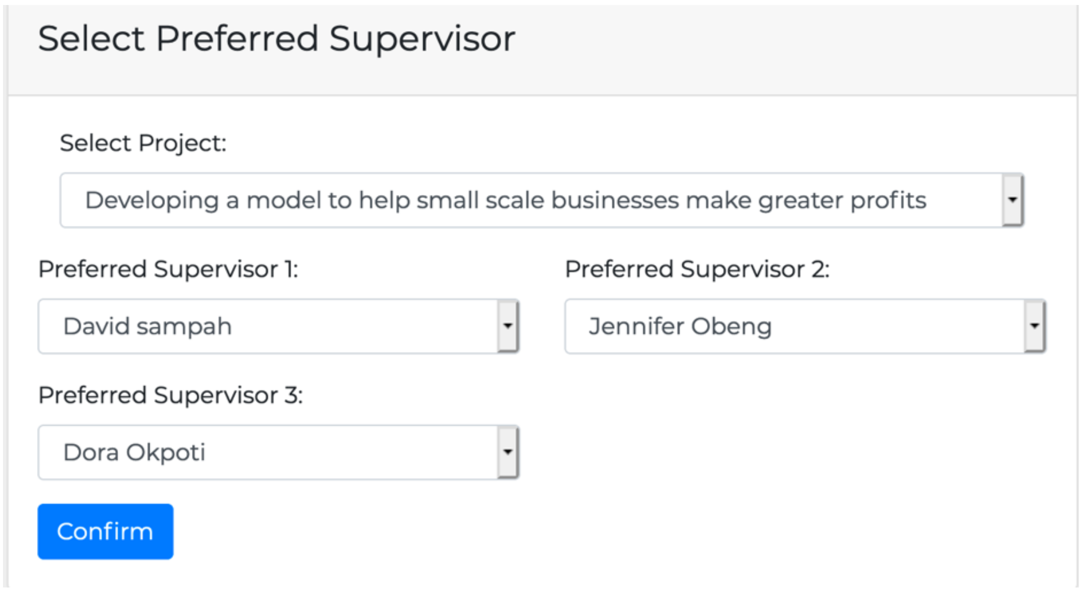
Project Type: Thesis

Project Field: Finance

Project Description:

incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Figure 30: Student proposing topic/project


Select Preferred Supervisor

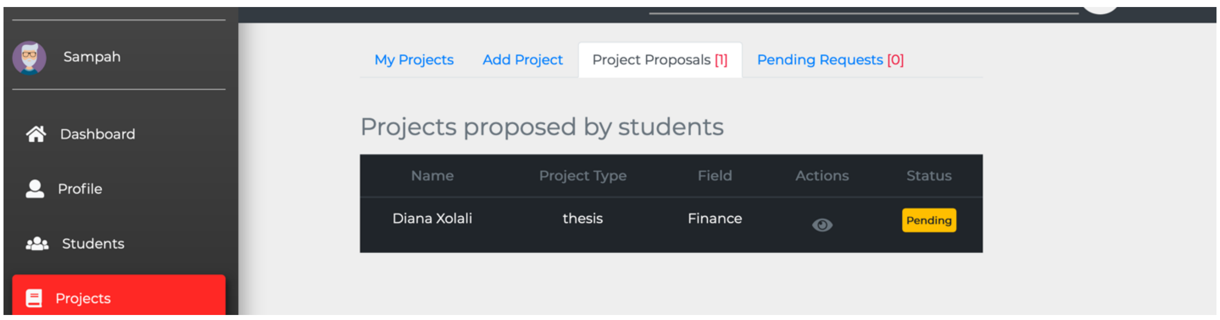
Select Project: Developing a model to help small scale businesses make greater profits

Preferred Supervisor 1: David sampah

Preferred Supervisor 2: Jennifer Obeng

Preferred Supervisor 3: Dora Okpoti

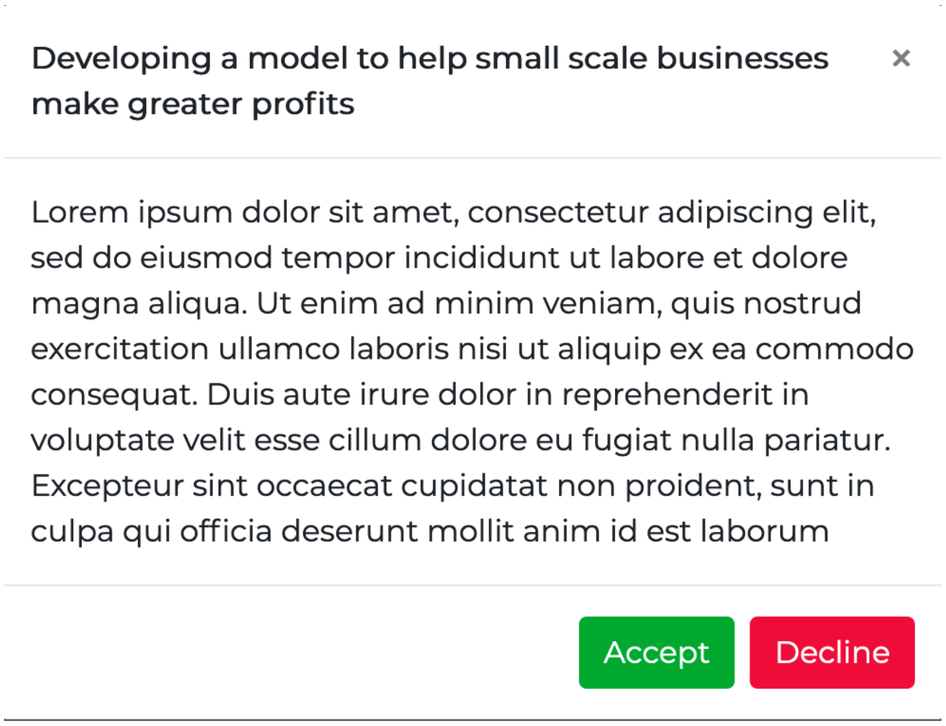
Figure 31: Student choosing supervisors:



Projects proposed by students

Name	Project Type	Field	Actions	Status
Diana Xolali	thesis	Finance	👁	Pending

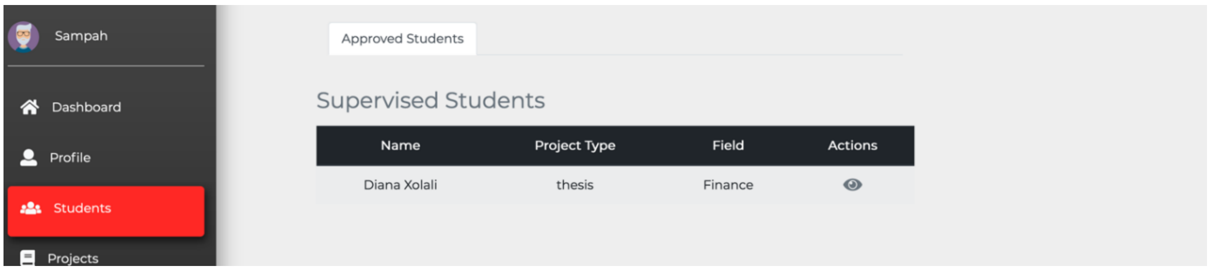
Figure 32: Faculty viewing student's proposal



Developing a model to help small scale businesses make greater profits

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

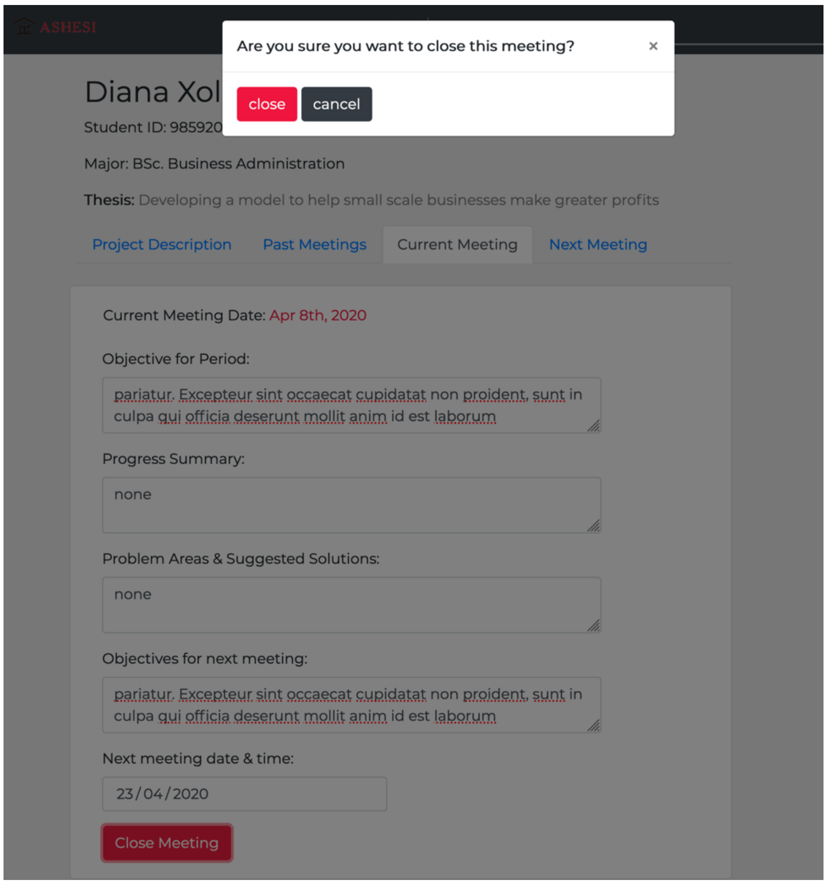
Figure 33: Faculty viewing and accepting student's project



Supervised Students

Name	Project Type	Field	Actions
Diana Xolali	thesis	Finance	👁

Figure 34: Student added to the faculty member's list of supervised students


ASHESI

Are you sure you want to close this meeting?

close cancel

Diana Xol

Student ID: 985920

Major: BSc. Business Administration

Thesis: Developing a model to help small scale businesses make greater profits

Project Description | Past Meetings | Current Meeting | Next Meeting

Current Meeting Date: Apr 8th, 2020

Objective for Period:

pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Progress Summary:

none

Problem Areas & Suggested Solutions:

none

Objectives for next meeting:

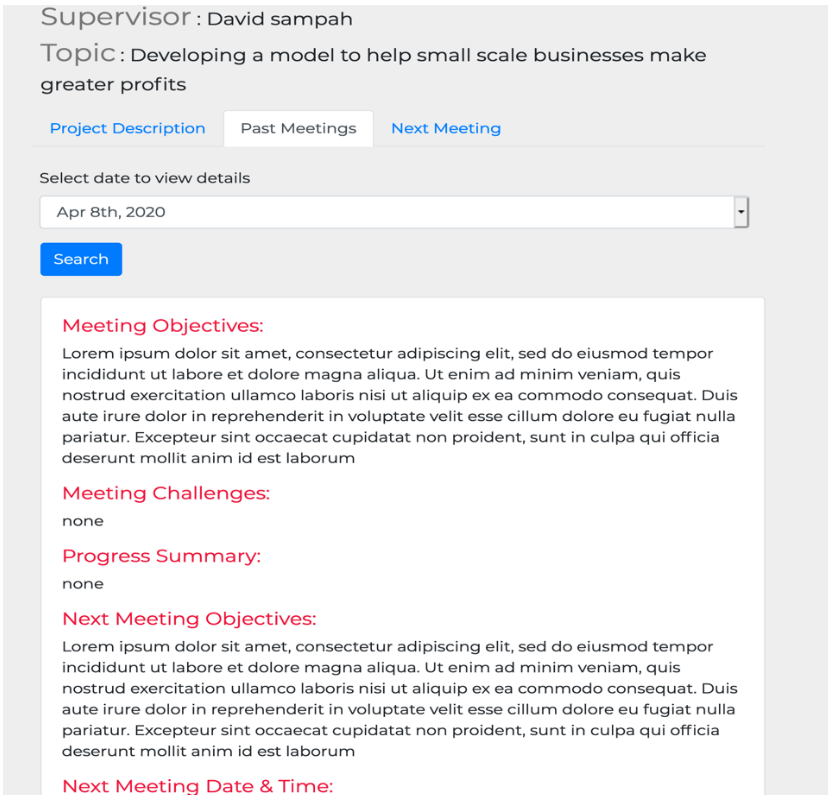
pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Next meeting date & time:

23/04/2020

Close Meeting

Figure 35: Faculty conducting a meeting with the student


Supervisor : David sampah

Topic : Developing a model to help small scale businesses make greater profits

Project Description | Past Meetings | Next Meeting

Select date to view details

Apr 8th, 2020

Search

Meeting Objectives:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Meeting Challenges:

none

Progress Summary:

none

Next Meeting Objectives:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Next Meeting Date & Time:

Figure 36: Student viewing details of the meeting

5.2 Browser Compatibility Testing

"Compatibility Testing is non-functional testing in which ensures that the application/website/system is capable of running on various objects like on various Browser, On various Resolution, on various Operating System, with some other application, on the network. [27]" This system was launched on many popular web browsers including Google Chrome, Mozilla Firefox, Safari and Opera Mini to observe how well it executes. All mentioned browsers run the system without any hitches. However, Safari and Google Chrome failed to recognize a bootstrap DateTime format class.

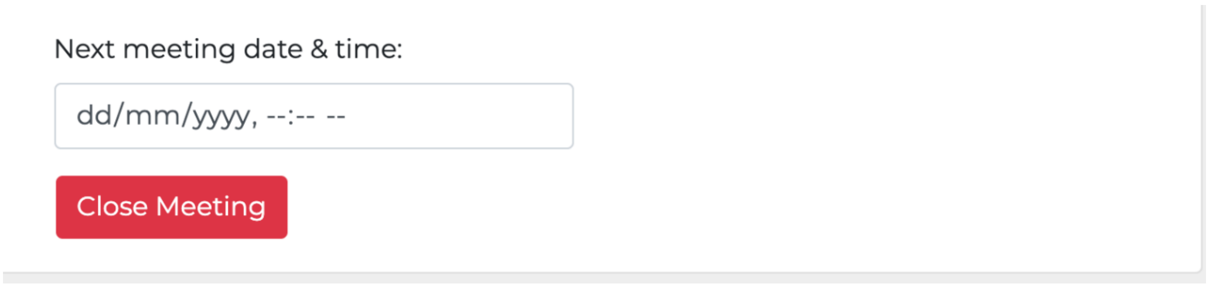


Figure 37: Mozilla Firefox display for datetime field.

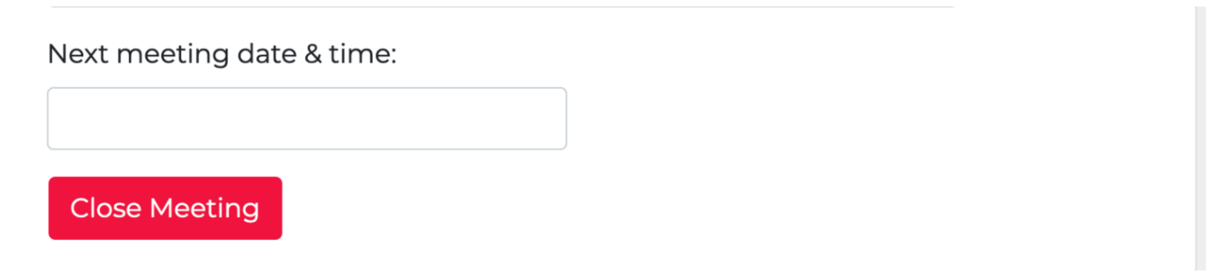


Figure 38: Google Chrome and Safari display for datetime field.

5.3 User Testing

"*User testing is the process through which the interface and functions of a website, app, product, or service are tested by real users who perform specific tasks in realistic conditions*". [26]

User testing is integral because it helps programmers decide whether an application or system in development mode is ready to be launched or not. User testing also allows developers to evaluate a system through the perspective of the typical user of the system. As such, the flow and sequence of the application can be perfected.

To conduct a thorough user testing, all 3 categories of users for the system were involved.

1. **Students**

   The key functionalities available to the student include signing up, proposing a topic, selecting preferred faculty members as supervisors and applying for topics proposed by faculty members. After interacting with the system and performing the tasks listed above, students suggested the following changes:

   a. The system could be integrated with Microsoft outlook to enable quick and easy log in. This can be likened to numerous systems already in use on the Ashesi campus.
   b. The system could allow the submission of capstone chapters to the computer science coordinator.

2. **Faculty**

Key functionalities for faculty members include proposing topics, accepting projects proposed by students, accepting or declining students who have applied for their projects, organizing meetings with students and providing feedback to students after capstone presentations. After interacting with the system, faculty members suggested the following:

a. The profile of students should be available prior to accepting or declining their request.
b. Faculty members should be able to view the projects of students without supervisors even if they are not chosen as preferred supervisors.

3. Computer Science (cs) Coordinator

Currently, the key functionalities of the coordinator are to add new faculty members and to set a limit on the number of students each faculty member is allowed to supervise. Other functionalities include, the coordinator being able to observe details such as information on faculty members and students with and without supervisors. After interacting with the system, the computer science (cs) coordinator gave the following suggestions:

a. To ease the disseminating of information such as reminders on upcoming presentations, the system should allow the contacting of all students at a go (Bulk mailing).

After conducting feedback with all 3 categories of users, they expressed their thoughts on the importance of such a system on the Ashesi Campus. They asserted that a system of this nature is even more important during the outbreak of the Novel COVID19; where education has been moved completely online. Moreover, the system does not only centralize activities to one platform, but it also makes it easier and seamless.

After gathering feedback and comments from the users above the following were modified and completed:

2 (a & b) and 3 (a)

Chapter 6: Conclusion

This chapter is going to focus mainly on the challenges faced during the development phase of this system and future work to be considered.

6.1 Challenges

Ultimately, there were two outstanding challenges throughout the course of this project. The first challenge encountered was the adaption to Laravel. Learning and development had to be carried out simultaneously due to the limited time frame for the project. Due to this, various errors were encountered at the early stages of this project which slowed down its progress. The second challenge was the inability to attain a large number of students and faculty members to conduct user testing. This problem was a direct result of the outbreak of the novel COVID-19 which led to the closure of the Ashesi University campus. With the schedules of both students and faculty being unexpectedly and quickly changed, only a few students and faculty members could avail themselves for user testing. One other challenge faced was learning the usage of new technologies such as AJAX and jQuery, which were fundamental to some functionalities of the system.

6.1 Future Work

Most features and functionalities considered for future work are based on some current limitations of the system. As such, should the project be continued, it would be beneficial to consider the following features:

- Allowing students and their supervisors to send messages to each other.

- Addition of a mobile application version of the system.
- As it stands, most functionalities on the part of the computer science (cs) coordinator make use of AJAX and jQuery. Create, Read, Update and Delete functionalities of students and faculty members can also be changed to use AJAX to allow the use of lightweight requests which do not need a page reload to return a response.

Conclusion

In conclusion, the Capstone Management System (CMS) provides a substantial contribution to the continuous advancement of the life cycle of capstone projects. In its success, this CMS will not only significantly reduce the workload of the computer science (cs) coordinator, it also presents an opportunity for students to have access to a large pool of projects to choose from.

References

[1] Harry A. Frolick and Robert M. Wilson. 2003. Web-Based Project Management System. (March 2003), 27.

[2] Kamchon Chio. 2010. Course Management System. (2010), 52.

[3] Knudson James George, William Lawrence Vivian, and Mark S. Crego. 1998. Dynamic Project Management System. (June 1998), 10.

[4] Richard E. West, Greg Waddoups, and Charles R. Graham. 2007. Understanding the experiences of instructors as they adopt a course management system. *Education Tech Research Dev* 55, 1 (February 2007), 1–26. DOI:https://doi.org/10.1007/s11423-006-9018-1

[5] Final Year/Capstone Project - BA - Ashesi University. Retrieved February 8, 2020, from http://v6.ashesi.edu.gh/academics/business-administration/business-administration-courses/1109-final-yearcapstone-project-ba.html

[6] Theodore C. Wagenaar. 1993. The Capstone Course. Teaching Sociology 21, 3 (1993), 209. DOI:http://dx.doi.org/10.2307/1319011

[7] 3-Tier Architecture: A Complete Overview. *JReport*. Retrieved February 28, 2020, from https://www.jinfonet.com/resources/bi-defined/3-tier-architecture-complete-overview/

[8] Introduction to HTML. Retrieved March 4, 2020, from https://www.w3schools.com/html/html_intro.asp

[9] CSS Tutorial. Retrieved March 4, 2020, from https://www.w3schools.com/css/

[10] JavaScript basics. *MDN Web Docs*. Retrieved March 4, 2020, from https:// developer.mozilla.org/en-US/docs/Learn/ Getting_started_with_the_web /JavaScript_basics

[11] Laravel Tutorial - Tutorialspoint. Retrieved March 4, 2020, from https://www.tutorialspoint.com/laravel/index.htm

[12] MySQL Tutorial - Tutorialspoint. Retrieved March 4, 2020, from https://www.tutorialspoint.com/mysql/index.htm

[13] Authentication - Laravel - The PHP Framework For Web Artisans. Retrieved March 4, 2020, from https://laravel.com/docs/5.7/authentication

[14] Database: Migrations - Laravel - The PHP Framework For Web Artisans. Retrieved March 23, 2020, from https://laravel.com/docs/5.8/migrations

[15] Laravel QuickStart - Laravel - The PHP Framework For Web Artisans. Retrieved March 24, 2020, from https://laravel.com/docs/4.2/quick

[16] Composer. Retrieved March 24, 2020, from https://getcomposer.org/download/

[17] Unit Testing Tutorial: What is, Types, Tools, EXAMPLE. Retrieved April 6, 2020, from https://www.guru99.com/unit-testing-guide.html

[18] Testing: Getting Started - Laravel - The PHP Framework For Web Artisans. Retrieved April 7, 2020, from https://laravel.com/docs/5.6/testing

[19] Jyothi. PHPUnit: Unit Testing | How to write Unit test cases. Retrieved April 8, 2020, from https://www.valuebound.com/resources/blog/understanding-phpunit-and-how-to-write-unit-test-cases

[20] 2011. System Testing. *Software Testing Fundamentals*. Retrieved April 8, 2020, from http://softwaretestingfundamentals.com/system-testing/

[21] AJAX Introduction. Retrieved April 8, 2020 from https://www.w3schools.com/xml/ajax_intro.asp

[22] jQuery Introduction. Retrieved April 8, 2020 from https://www.w3schools.com/jquery/jquery_intro.asp

[23] UML Use Case Diagram Tutorial. *Lucidchart*. Retrieved April 13, 2020 from https://www.lucidchart.com/pages/uml-use-case-diagram

[24] What is User testing? Definition. *Omniconvert*. Retrieved April 30, 2020 from https://www.omniconvert.com/what-is/user-testing/

[25] Dwarika Dhish Mishra. 2012. Browser Compatibility testing- When, Where and how it is done. *Abode QA*. Retrieved April 8, 2020, from https://abodeqa.com/browser-compatibility-testing-when-where-and-how-it-is-done/