



ASHESI UNIVERSITY

Ashesi Tutor Management System

Applied Project

BSc. Computer Science

Emmanuel Teye-Kofi Odonkor

April 2021

ASHESI UNIVERSITY

Ashesi Tutor Management System

Applied Project

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

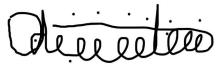
Emmanuel Teye-Kofi Odonkor

April 2021

DECLARATION

I hereby declare that this capstone 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:

Emmanuel Teye-Kofi Odonkor

.....

Date:

13th May, 2021

.....

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

Supervisor's Signature:



.....

Supervisor's Name: David Ebo Adjepon-Yamoah, Ph.D

.....

Date: 14th May 2021

.....

ACKNOWLEDGEMENT

Firstly, I would like to thank the Almighty God for his grace, mercy, and guidance in the development of this remarkable project. I also would like to thank my mother, Mary Odonkor, and my siblings, Gifty Odonkor, Evans Odonkor, and Collins Odonkor for their continuous encouragement and support during the development of this project. Thanks to you and may God richly bless you all.

Secondly, I also would like to express my appreciation and gratitude to my supervisor, Dr. David Ebo Adjepon-Yamoah for this support, guidance, advice throughout the execution of this project. I am very grateful to him for his continuous encouragement from the early phase of this project through to the development phase and the ending phase of this project. Thank you Dr. David Ebo Adjepon-Yamoah and may God richly bless you.

Lastly, I'd like to give special thanks to all lecturers at Ashesi University for their training and acquisition of knowledge from their designated disciplines. I must say this knowledge has helped to build myself in various disciplines. I have also applied this knowledge in the development of this project, therefore I am most grateful. Thank you.

ABSTRACT

The focus of this project is centered on the problem of understanding courses at Ashesi University and the implementation of a Tutor Management System to help address this problem. The topic of this project is relevant because the ultimate goal of an academic institution is to ensure that they equip students with the knowledge so they can positively contribute back to society. Hence the failure of students to thoroughly acquire understanding adversely affects them and the society at large. In this project, we would look into the problem of understanding university-level courses from the perspective of other related works and the thorough procedures followed in the implementation of the Ashesi Tutor Management System. Users' responses were obtained from a survey after the implementation of the Ashesi Tutor Management System. 100 % of the users described the approach in implementing the functionalities as excellent whiles 75% described the coloring and design of the application as outstanding.

Table of Contents

DECLARATION	i
ACKNOWLEDGEMENT	ii
ABSTRACT	iii
Chapter 1: Introduction	1
1.1 Background	1
1.2 Related Works	1
1.3 End-Product and its functionality	3
1.4 Sustainable Development Goals.....	4
Chapter 2: Requirement Analysis	5
2.1 Users and Stakeholders	5
2.2 Method of Gathering Requirements.....	6
2.2.2 Questionnaires for Requirement Elicitation and Results Obtained	7
2.3 Analysis of Results from Requirement Elicitation	11
2.4 Project Objectives	12
2.5 Research Approach	12
2.6 Set of Requirements.....	13
2.6.1 Functional Requirements	13
2.6.2. Non-Functional Requirements	14
2.7 User Story(Scenario).....	14
2.7.1 Key Tasks of Users.....	16
2.8 Unified Modeling Language(UML) Diagrams	16

2.8.1 Use Case Diagrams	16
2.8.2 Activity Diagrams	19
2.8.3 Sequence Diagram	22
2.8.4 Conceptual Database Design of the Ashesi Tutor Management System.....	24
Chapter 3: Architecture and Design.....	25
3.1 Architecture Overview	25
3.1.1 Model View Controller Architecture	25
3.1.2 Layered System Architecture.....	27
3.1.3 Repository Architecture	29
Chapter 4: Implementation.....	31
4.1 Front-End Development.....	31
4.1.1 Human-Computer Interaction (HCI)	31
4.1.2 HTML,CSS and JavaScript.....	31
4.1.3 AJAX and jQuery	32
4.1.4 SweetAlert library.....	32
4.1.5 Font Awesome Toolkit	32
4.1.6 EmojioneArea Plugin.....	33
4.1.7 Bootstrap Framework	33
4.2 Back-End Development	33
4.2.1 PHP(Hypertext Pre-Processor)	33
4.3 Database Development.....	34

4.3.1 Visual Paradigm.....	34
4.3.2 Structured Query Language	34
4.3.3 PhpMyAdmin.....	35
4.4 Local Development	35
4.5 Implementation Procedure	35
4.5.1 Landing Page of Application	35
4.5.2 Administrator Dashboard.....	36
4.5.3 Departmental Coordinators Designated Dashboards	37
4.5.4 Student Dashboard.....	39
4.5.5 Tutor Dashboard	41
4.5.6 Chat Application.....	43
4.5.7 Student-Tutor Matching Portal	45
4.6 Software Deployment.....	50
Chapter 5: Testing And Results	52
5.1 Development Testing	52
5.1.1 Unit Testing	52
5.1.2 Component Testing.....	53
5.1.3 Automated Testing (Selenium tool).....	54
5.1.4 Results from Selenium Testing.....	56
5.2 User / Acceptance Testing	60
5.3 Analysis of User Testing Survey Results.....	62

5.4 Application Requirements Testing Report.....63

Chapter 6: Conclusion and Recommendation.....65

6.1 Overview of Project65

6.2 Project Limitations and Challenges66

6.3 Future Works.....67

6.4 Conclusion68

References69

Appendix 171

Chapter 1: Introduction

1.1 Background

Getting the right academic support for university education is challenging [7]. The challenges faced include students facing difficulties in coping with course materials, lack of support for students in understanding concepts introduced in university-level courses, lack of good foundation in a course, the teaching of large student populations, and educating of different groups of students at different stages of their careers [7], [8]. In light of these challenges, universities endeavor to introduce new technologies and cutting-edge concepts to students to enable them to be well prepared for the rapidly evolving and competitive industry [7]. This problem is of prior importance because its repercussions include the lack of understanding of class concepts on the side of students, failure of students to pass their exams, a lower GPA which can eventually lead to the dropout of students from university institutions [7], [10]. These adverse effects also affect the progress of the University institution and the African society at large. Even the most progressive of universities in Africa like Ashesi, faces such challenges.

1.2 Related Works

A research work [7], brings to light a similar issue about first-year computer programming students facing difficulty in understanding Visual Programming 1 course at the University of South Africa. Visual Programming 1 is a first-year introductory programming course for novice students at the University of South Africa. The University has been experiencing a high dropout rate and a low pass rate of novice students enrolled in the course over the years [8]. This made it clear that novice students were finding it difficult to understand the concepts in the course. To curb this issue, the solution proposed was the implementation of an E-tutoring platform, myUnisa, a system used by e-tutors to

help get students to understand the concepts in Visual Programming 1 course as well as bringing them up to speed on e-learning resources that would help them to understand course concepts better.

In another work [5] also highlights the implementation of an online tutoring system with the sole purpose of helping nontraditional students learn subjects that require a significant amount of tutor assistance, who may only be available on campus. The subjects to be taught by tutors on this platform include but are not limited to Calculus, Algebra, and Physics. As an e-tutoring system to be used anytime by e-tutors and students, the paper highlights more on the design and implementation of major key subsystems of the project such as user account, problem account, tutoring, and accounting management systems.

In analyzing the work done concerning the implementation of myUnisa in the University of South Africa, the choice of the University to implement the e-tutoring system, myUnisa, was very efficient as it was able to appropriately curb the situation. To buttress this analysis, evidence from the paper shows how helpful the system was as qualitative analysis through interviews highlights how both e-tutors and students gave positive feedback about the usage of this system. Also, quantitative data taken from the university's routine statistics directorate shows major improvement in terms of student throughput as there was a reduction in the dropout rate of novice students taking the course. In the related work [5], the approach of using design visuals and architectural frameworks to help give readers a good mental model about the design and implementation was beneficial as it conveyed the intended meaning. Both related works bring to mind the two types of software products namely customized and generic products and why they must be considered in software implementation. The e-tutoring system, myUnisa, was a customized software product as the requirements and specifications for

the system were provided by the University and upon completion, the product was delivered and used by the University only. For the second related work, unlike myUnisa, it was generic as it was developed and published on the open market.

1.3 End-Product and its functionality

Currently, Ashesi University has an existing solution to help address students lacking understanding in a course. The solution is the tasking of tutors to assist students who need help on particular courses. Therefore what this project seeks to accomplish is to contribute to Ashesi's current solution by building an Ashesi Tutor Management System that would effectively manage these tutors, and give students a second chance to make things right in their quest for academic success. The tutor management system would be access-controlled with designated Students, Administrator, Department Coordinators, and Tutor user privileges.

Students in need of a tutor to help them with a course would log into the system to find tutors. Tutors can set up their profile which would be visible for students to find them. Departmental Coordinators would be representing their designated departments thus the Computer Science and Information Systems, Business Administration, Engineering, and Humanities and Social Sciences Department on the system. They can view tutors from their various departments, provide courses and the number of tutors required per course from their department. The setting of courses and the number of tutors by the departmental coordinators would help the Administrator to know which courses are available and the preferred number of tutors needed per course. The Administrator can find all tutors from all departments, know the number of tutors required per course, and set up courses for tutoring which would be visible to students who desire to be tutors.

1.4 Sustainable Development Goals

The fourth Sustainable Development Goal titled Quality Education is the guideline for this project. One target of SDG4 which characterizes the vision of this project is Target 4.6 titled Literacy and Numeracy. In brief, this target aims to ensure that the youth, both men and women obtain literacy and numeracy. The corresponding indicator which would be used as a hallmark to show success in this target is when a proportion of the population achieves a fixed level of proficiency in literacy, functional, and numeracy skills.

This project resonates with this goal because it seeks to build a Tutor Management System that would help students facing difficulty in courses to acquire knowledge in their respective areas of study and help to improve on their literacy, functional, and numeracy skills.

Chapter 2: Requirement Analysis

A user-centered approach to design enforces how the users and their tasks are the driving force behind a product's development. In using this approach as a motivation for requirement elicitation, below are the various sub-sections under Requirement Elicitation:

2.1 Users and Stakeholders

The users and stakeholders who have been identified to be the key individuals in my proposed software application are categorized into four groups. These are the Administrator, Ashesi Students, Ashesi Faculty, the Tutors, and lastly Heads of Departments (HODs).

Administrator

The Administrators act as the overseers of the Tutor Management System. They ensure the smooth running of the application through the provision of functionalities such as the setting up of courses for tutoring.

Ashesi Students

The Tutor Management System would best serve the needs of Ashesi students by providing them the opportunity to know the available tutors they can reach out to in case of course help. They also form part of the primary users who would constantly be engaging with the system.

Tutors

They are Ashesi students who can sign up on the platform to become tutors for specific courses provided on the platform by the Administrator. They would be in contact with the students by providing them course help. They also form part of the primary users.

Ashesi Faculty

The Faculty members also form part of the stakeholders as they would not directly interact with the Tutor Management System but would be interacting with the Tutors most often. Since it is their course that the tutors would be helping the Ashesi Students on, they are recognized as key stakeholders too.

Heads of Departments(HODs)

Similar to the faculties, the Heads of Departments are also key stakeholders and users of my project. As stakeholders, they would act as secondary overseers of the system. As users, they would have to log in as departmental coordinators for their designated courses and assist in the provision of courses and the number of tutors required.

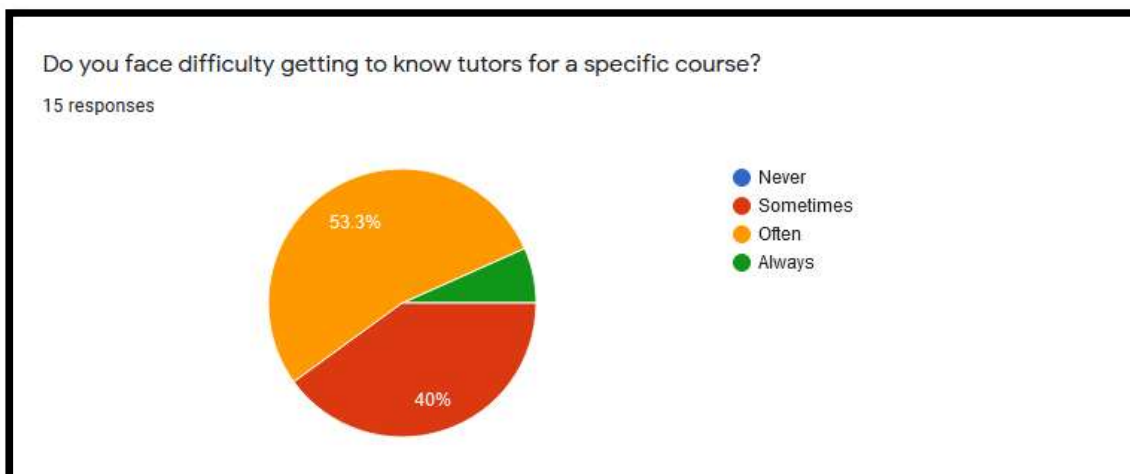
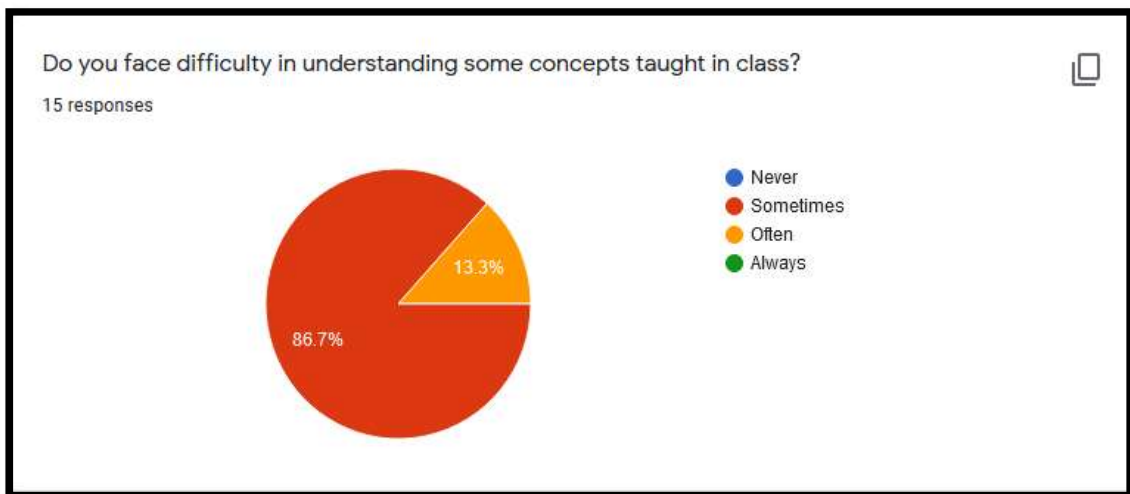
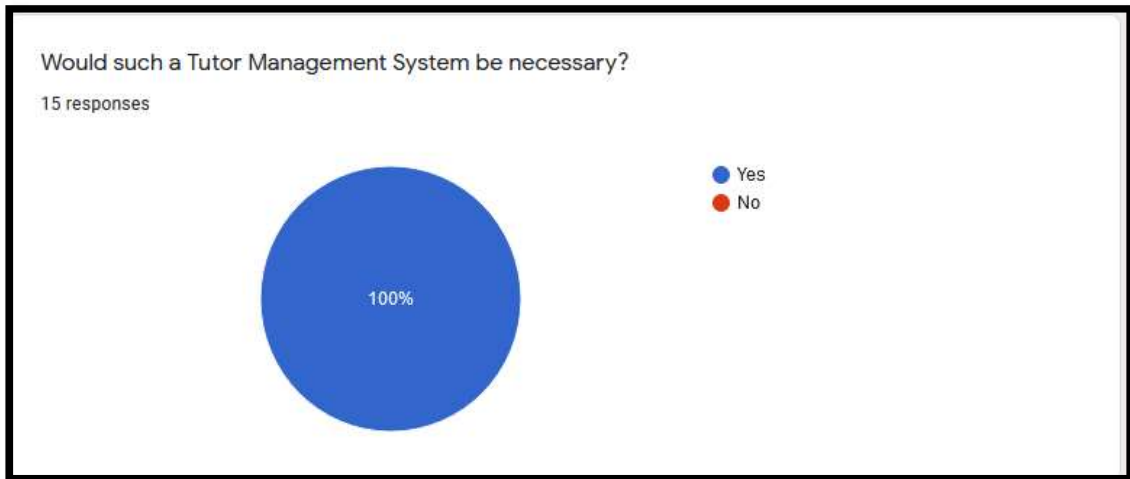
2.2 Method of Gathering Requirements

Questionnaires Survey

Due to the outbreak of the COVID-19, sending out an online form to share my questionnaires with my users and stakeholders would be appropriate. These questionnaires would be categorized such that inputs received would be able to be grouped and mapped out to know the overall view of each group. These questionnaires would help me to understand the problem, gain insight from users and stakeholders as well as find additional functionalities to be included in the system. In Section 2.2.2, the Questionnaires for Requirement elicitation and the results collected are shown.

2.2.2 Questionnaires for Requirement Elicitation and Results Obtained

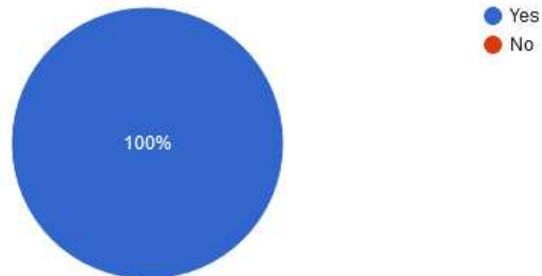
From Ashesi Students Survey:



Based on your response in Q2 and Q3, would a list of potential tutors provided on the Tutor Management System for specific courses be helpful to you?



15 responses



What other features do you want to see in the system?

9 responses

A rating system for the tutors

A feature with recommendations on extra material we can read after class

I will be fine with anything

I would want the tutors to be categorized based on the departments

With the tutors names and their respective courses attached.

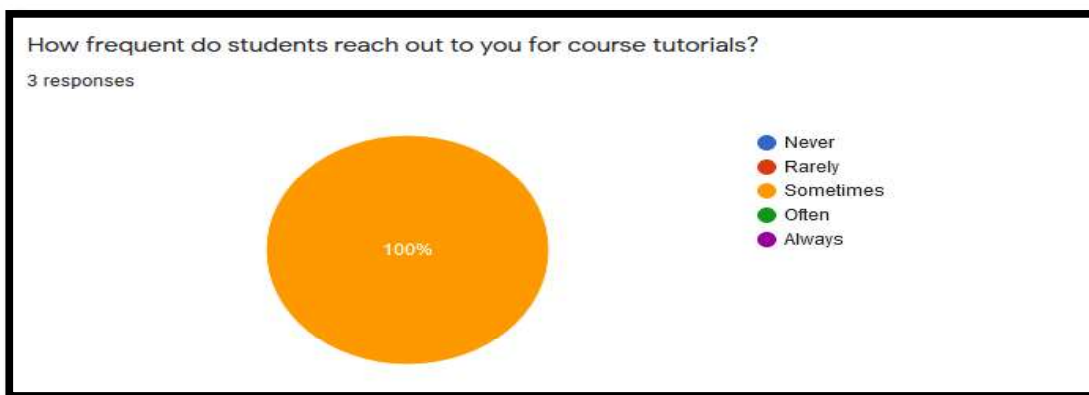
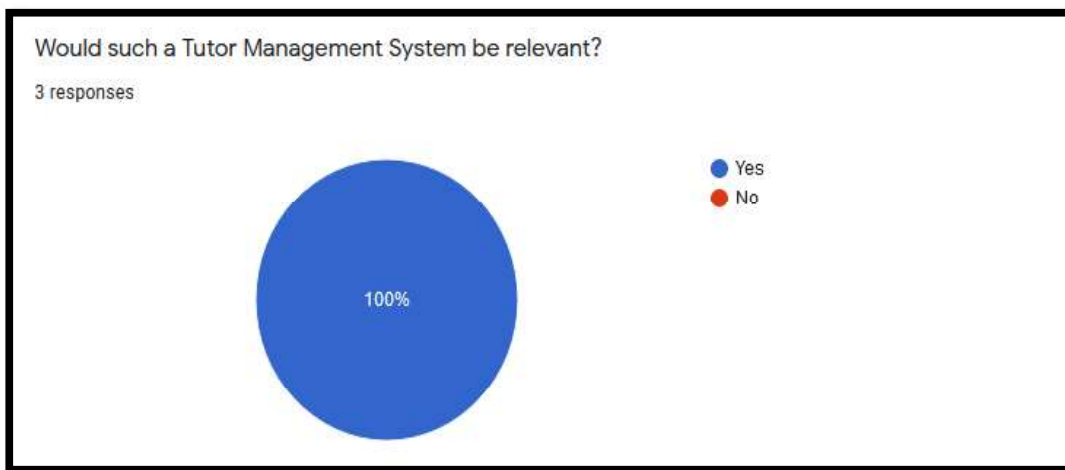
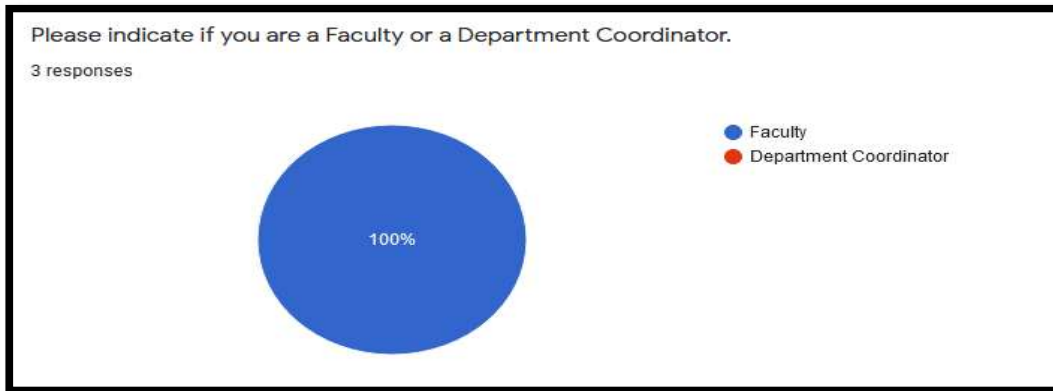
Contact details of tutors as well as a private chat section to be able to have an easier introduction before physical meet up.

Notification,calendar

A notification component of the system

I think the indicated features will suffice

Results From Ashesi Faculty Survey:



How many tutors do you prefer to be assigned to your course? (Please state the course and the number of tutors needed).

3 responses

2 minimum

Web technologies - 5

On the average, two per each of the following courses: Computer Organization and Architecture, Communication Systems, and Networks & Data Communication.

What other features do you want to see in the system?

2 responses

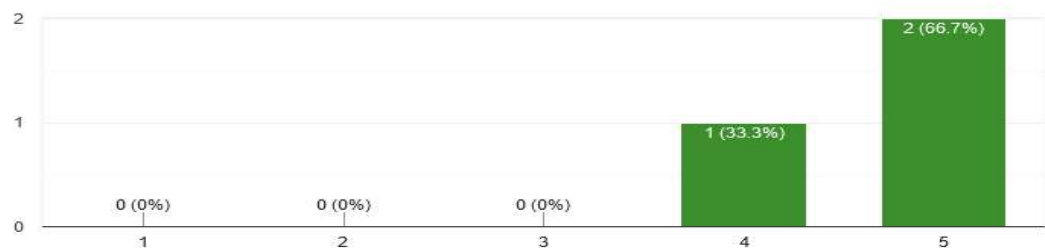
Calendar to see tutor busy schedule

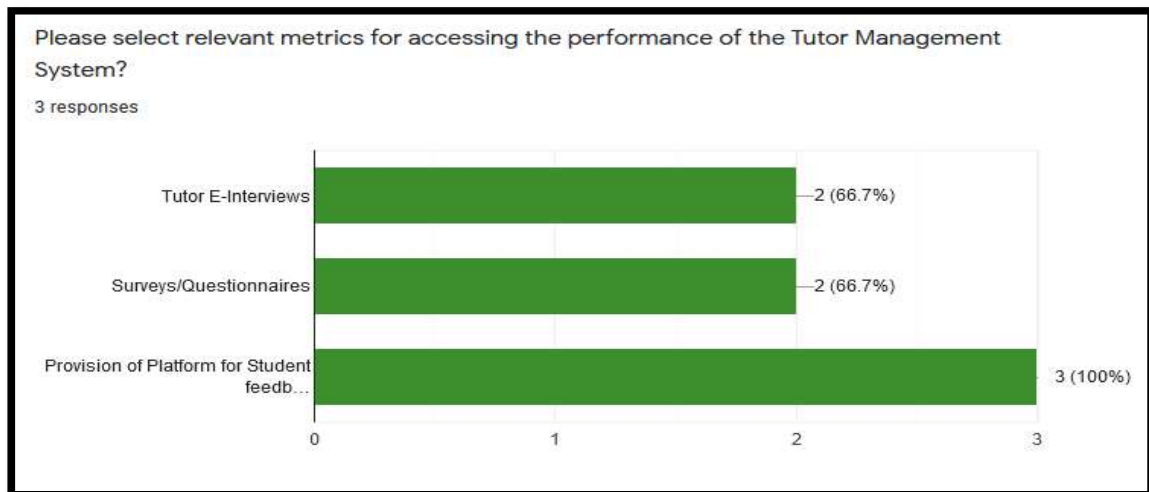
The system should permit students to clearly specify which aspect of a topic they have difficulty with. It will inform tutors who think they are strong in the stated area to engage them. Also, it will enable the faculty to discover over time, which aspect of the course students have problems with most.

Results From Departmental Coordinators Survey:

How relevant would such a Tutor Management System be for your department?

3 responses





Please provide some comments on the proposed Tutor Management System?(If any)

1 response

Useful system but you need to consider other technologies or strategies from a good collection of courses, as a capstone project. As it stands now, it is a typical Web Technologies class project. Highlighting an interesting algorithm, emphasising on some HCI strategies/heuristics, incorporating some NLP, Machine Learning, Data Science APIs, etc. can be interesting.

2.3 Analysis of Results from Requirement Elicitation

From the requirement elicitation, 100% of Ashesi students and faculty expressed the urgent need for a Tutor Management System. 66.7% of Departmental Coordinators from the chart also expressed that the development of this system would be relevant in helping to curb the problem of lack of understanding in courses. 87% of Ashesi Students from the chart expressed the fact that they truly face difficulty in understanding class concepts taught in class while 53.3% of Ashesi Students from the survey also express the idea of often finding difficulties in getting to know the tutors they need for a challenging course. Therefore the above pieces of evidence from the Related Works in section 1.2 and

results from the Requirement Elicitation testify that the lack of understanding of concepts by students is indeed a problem. Therefore the development of an Ashesi Tutor Management System to address this need would be preferred.

2.4 Project Objectives

Given the evidence of proof from the requirement elicitation that the development of an Ashesi Tutor Management System is relevant. Hence to achieve this aim, the following are the Objectives of this project:

OB1: To provide a system that will support the tutoring process at Ashesi University.

OB2: This project would seek to assist students in their academic journey.

OB3: The implementation of this system would assist Ashesi Faculty by providing tutors to assist students in course tutorials.

OB4: This project would seek to improve the academic performance of students at Ashesi University.

2.5 Research Approach

The research approach to be used in the implementation of this project is the Agile Development Approach. The tutor management system would be developed as a series of increments with users and stakeholders involved in its specification and evaluation. In using the Agile approach, the process of implementation, design, and testing of the tutor management system would be inter-leaved so a thorough process of user-developer negotiation can occur during its development process.

2.6 Set of Requirements

After sending out the set of questionnaires to Ashesi students, Faculty, and the Departmental-Coordinators and upon considerations from my supervisor, the following are the set of functional and non-functional requirements obtained.

2.6.1 Functional Requirements

Given the proof of evidence from the Requirement Elicitation, the need for an Ashesi Tutor Management System has proven to be very relevant. Hence upon analyzing the results from requirement elicitation, the following are the set of functional requirements the users and stakeholders would want to see implemented in the Ashesi Tutor Management System:

FR1: Students should be able to view tutors categorized under departments.

FR2: Tutors should be able to provide and update their profile which includes their course interests, available times, and days for tutoring so students can view them.

FR3: Students should be able to rate and evaluate tutors on their performance.

FR4: Students should be able to get matched to a Tutor upon request.

FR5: Users should a Chat Application for Tutor-Student Interaction.

FR6: Students should be able to find relevant academic resources on the System Platform.

FR7: The System should have a notification component to alert users of any information.

2.6.2. Non-Functional Requirements

As part of the provision of the set of requirements from the users and stakeholders, the general set of non-functional requirements expressed in the Ashesi Tutor Management System follows:

NFR1: Security: The tutor management system would be access-controlled with designated Administrator, Student, Tutor, and Departmental Coordinators privileges. Entry into the system would be based on Multi-Authentication.

NFR2: Maintainability: As a system built using the Agile approach, the system would be periodically updated to meet the changing needs of the users.

NFR3: User-Friendly: The Ashesi Tutor Management System would be structured such that students trying to find an available tutor would find the process easy. Tutors also would be able to interact with the application easily. The goal here is to improve the users' experience.

NFR4: Effectiveness: The Ashesi Tutor Management System should be able accurate and effective in the delivery of its functionalities.

2.7 User Story(Scenario)

The first persona for this scenario is called Emmanuel Wilson. He is 22 years of age and is currently a second-year student at Ashesi University studying Computer Science.

Scenario 1: After attending numerous lectures on Intermediate Computer Programming, Emmanuel still had difficulty in understanding Classes and Polymorphism. His lack of understanding greatly affected his course progress as he has obtained low grades in most of his assignments. He then realizes he needs a tutor to assist him with this

course. He logs into the Ashesi Tutor Management System with his credentials as a student. On the Student Dashboard, he navigates his way to the section of "Find Tutors" where tutors are categorized based on departments. He then goes into the CSIS department and goes through the list of tutors available and finally finds a tutor for ICP. He then views the tutor's available days and times for tutoring, skill-level, contact details, and email. Emmanuel begins constant interaction with his tutor via the Chat Application in the Tutor Management System, downloads helpful documents uploaded by his tutor, and constantly rates the tutor based on their performance and his improvement. Gradually, Emmanuel is picking up the theoretical skills in ICP and is now performing well in class.

The second persona for this scenario is called Mercy Gyan. She is 23 years of age and is currently a third-year student at Ashesi University studying Computer Engineering.

Scenario 2: Mercy over the past few weeks has witnessed her strength in the theoretical, and practical aspect of Programming hence wants to help fellow students who are finding difficulty in it. She then loads the Tutor Management System to register as a Tutor by providing relevant details during "sign up" including her course of interest to tutor thus, Data Structures and Algorithms. After creating an account, she logs in with her credentials as a tutor and lands on the Tutor Dashboard. On this page, she navigates to the "Set Up Tutor Profile" page and creates her profile by providing her available days and times for tutoring, contact details, and other relevant information. The information she provides would be visible to students who wish to find tutors available for help with Data Structures. To re-enforce her profile, she sets off to the "Student-Tutor Matching" portal where she provides details on her strengths in Data Structures, tutoring mode preferred, and other relevant information. After saving her data, she is sent to a page that shows her matched student. Since her strengths were the weaknesses of the student and also her skill level was above that of the student she was a suitable match to help the student. She

begins interactions via the Chat Application and has been helping the student in practical areas in Data Structures.

2.7.1 Key Tasks of Users

- Register or Sign in on the Platform
- Find a Tutor
- Create and Update tutor profile
- Set up courses for tutoring
- Upload and Download Academic resources
- Rate and Evaluate Tutors
- Engage in Chat Application
- Student – Tutor Matching

2.8 Unified Modeling Language(UML) Diagrams

2.8.1 Use Case Diagrams

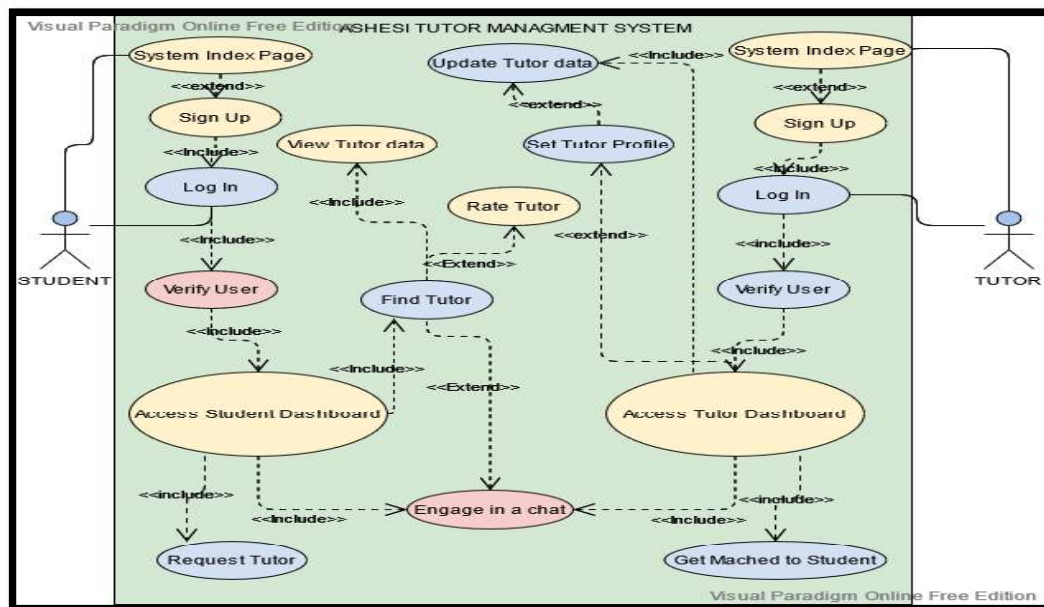


Figure 2.8.1a: Use Case Diagram for the Student and Tutor in the Tutor Management System

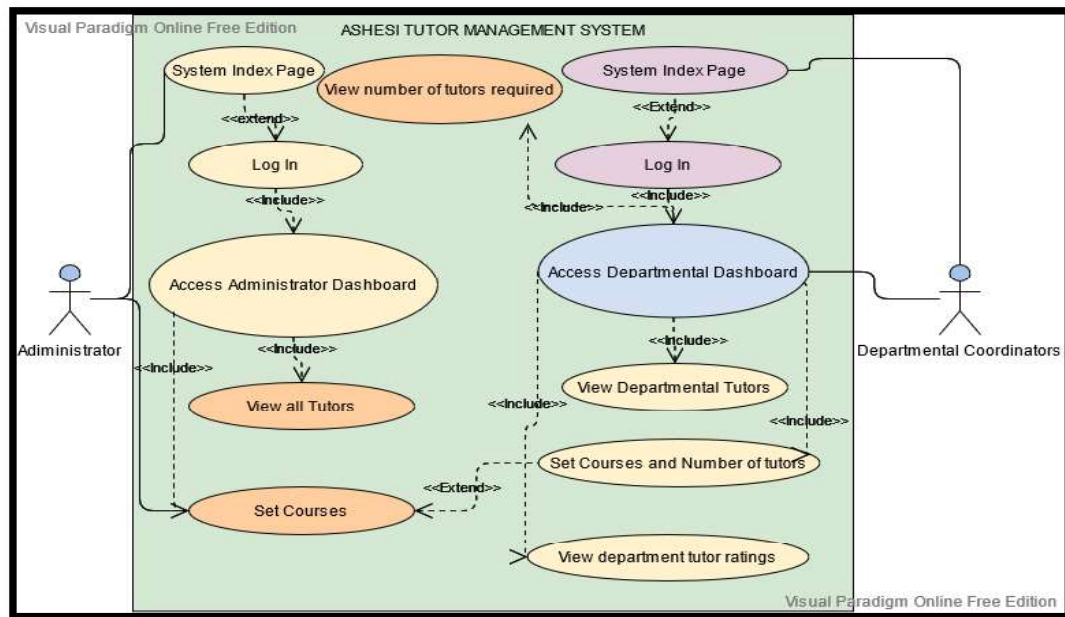


Figure 2.8.1b: Use Case Diagram for the Administrator and Departmental Coordinators in the Tutor Management System

Table 2.8.1: Table description of Use Case Diagrams

<u>TUTOR MANAGEMENT SYSTEM</u>	
Actors	Students, Tutors, Administrator, Departmental Coordinators
<u>Description</u>	After attending numerous lectures on Intermediate Computer Programming, Emmanuel still had difficulty in understanding Classes and Polymorphism. In that semester, the CSIS Departmental coordinators set the need for tutors in ICP privately and send this request to the Administrator. The Admin sets ICP publicly which is visible to students who want to be tutors. Mercy is good with programming hence sets up her profile as a tutor for ICP in the Tutor Management System. Michael in need of an ICP tutor logs into the system with his credentials as a student. On the Student Dashboard, he navigates his way to the section of “ <i>Find Tutors</i> ” where tutors are categorize based on departments. He then goes into the CSIS department and goes through the list of tutors available and finally finds Mercy, a tutor for ICP. He then views her profile, and skill level in the course. Emmanuel begins constant interaction with his tutor via the Chat Application within the System. He constantly rates the Mercy which is visible to the tutor's department. Gradually, Emmanuel is

	picking up the skills in ICP and is now performing well in class.
<u>Data</u>	Tutor Information, Course Information, Chat information, Availability of Tutors
<u>Stimulus</u>	Verification of Users access control by the system and the Chat Application
<u>Response</u>	Administrator's receiving of courses needed for tutoring, Students successfully receiving tutors information upon request
<u>Comments</u>	The tutoring courses must be created privately by the departmental coordinators before the Administrator makes them publicly available to students willing to be tutors.

2.8.2 Activity Diagrams

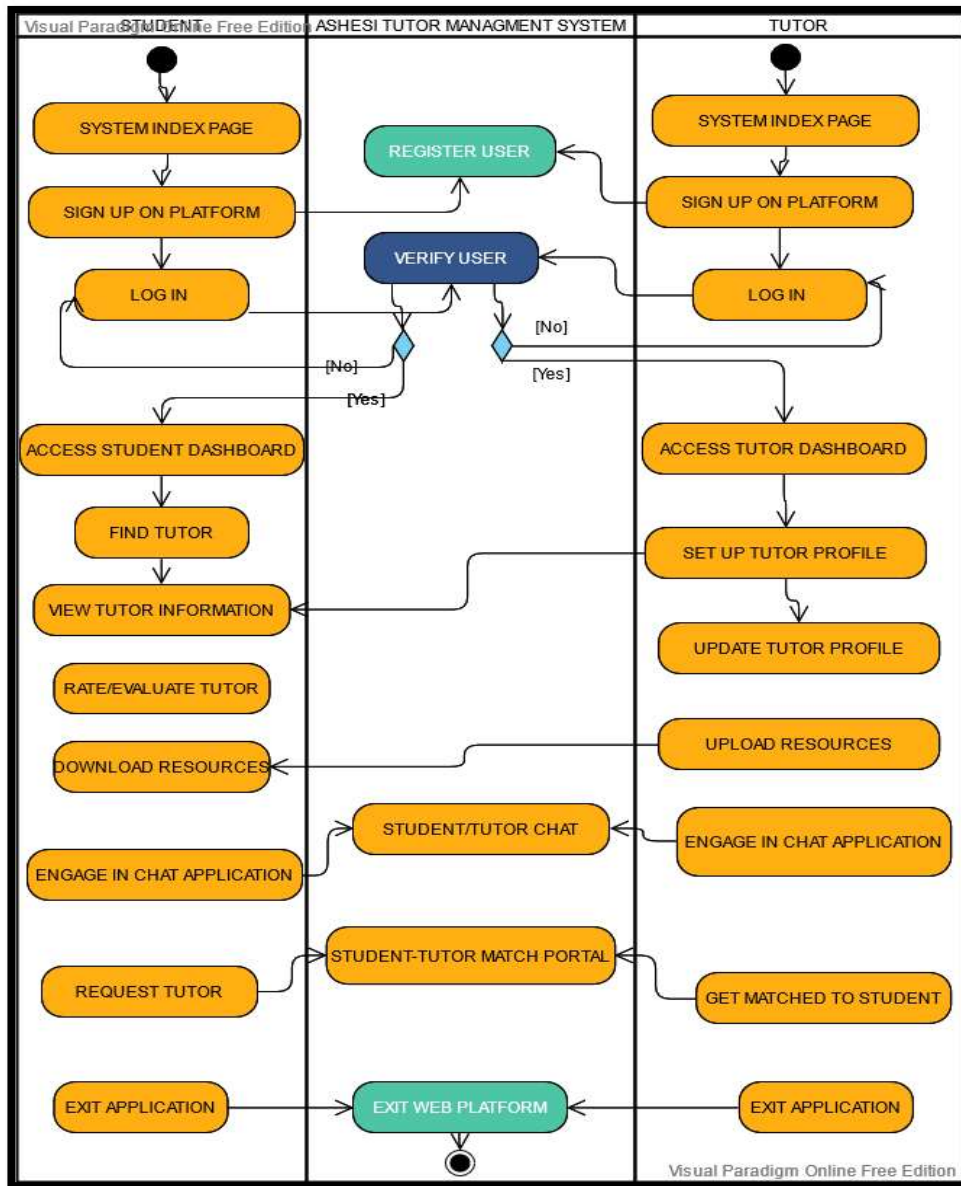


Figure 2.8.2a: Activity Diagram for the Student and Tutor in the Tutor Management System

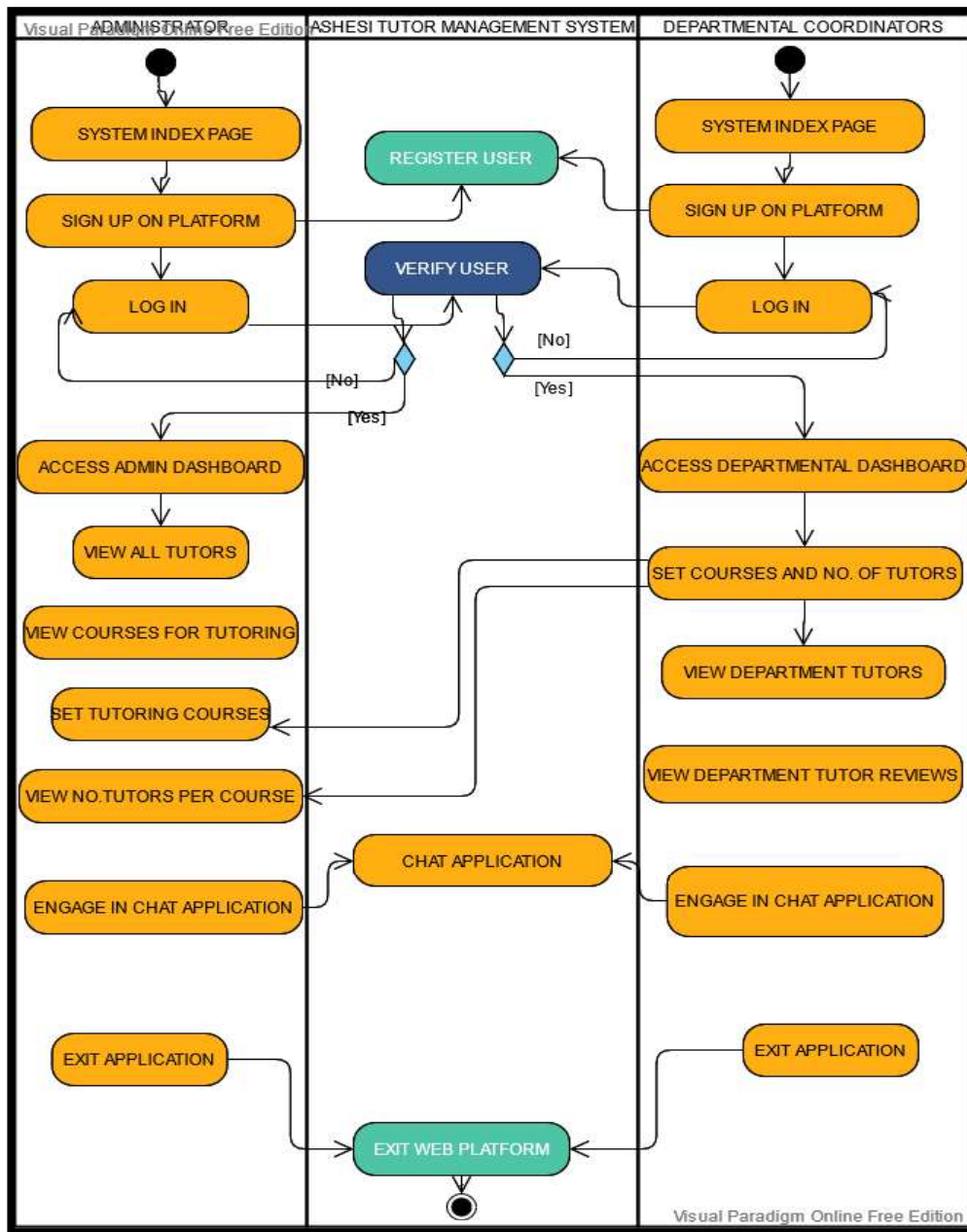


Figure 2.8.2b: Activity Diagram for the Administrator and Departmental Coordinators in the Tutor Management System

Table 2.8.2: Table description of Activity Diagrams

<u>TUTOR MANAGEMENT SYSTEM</u>	
<u>Actors</u>	Students, Tutors, Administrator, Departmental Coordinators
<u>Description</u>	After attending numerous lectures on Intermediate Computer Programming, Emmanuel still had difficulty in understanding Classes and Polymorphism. In that semester, the CSIS Departmental coordinators set the need for tutors in ICP privately and send this request to the Administrator. The Admin sets ICP publicly which is visible to students who want to be tutors. Mercy is good with programming hence sets up her profile as a tutor for ICP in the Tutor Management System. Michael in need of an ICP tutor logs into the system with his credentials as a student. On the Student Dashboard, he navigates his way to the section of " <i>Find Tutors</i> " where tutors are categorize based on departments. He then goes into the CSIS department and goes through the list of tutors available and finally finds Mercy, a tutor for ICP. He then views her profile, and skill level in the course. Emmanuel begins constant interaction with his tutor via the Chat Application within the System. He constantly rates the Mercy which is visible to the tutor's department. Gradually, Emmanuel is picking up the skills in ICP and is now performing well in class.
<u>Data</u>	Tutor Information, Course Information, Chat information, Availability of Tutors
<u>Stimulus</u>	Verification of Users access control by the system and the Chat Application
<u>Response</u>	Administrator's receiving of courses needed for tutoring, Students successfully receiving tutors information upon request
<u>Comments</u>	The tutoring courses must be created privately by the departmental coordinators before the Administrator makes them publicly available to students willing to be tutors.

2.8.3 Sequence Diagram

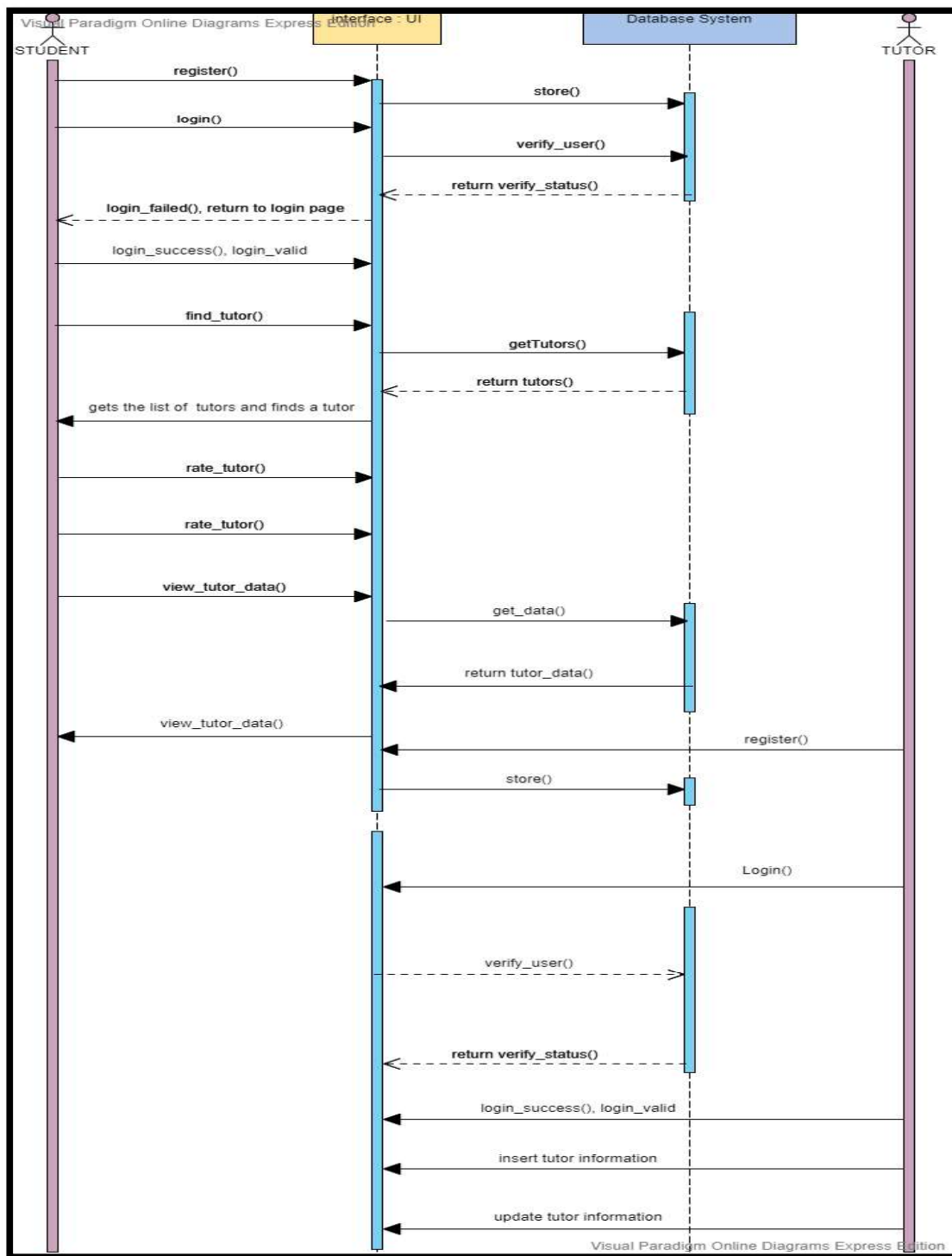


Figure 2.8.3: Sequence Diagram for the Tutor Management System

Table 2.8.3: Table description of Sequence Diagrams

<u>TUTOR MANAGEMENT SYSTEM</u>	
Actors	Students, Tutors
<u>Description</u>	Michael after signing up on the Tutor management system logs in with his credentials as a student. As a multi-authenticated application, Michael's access type as a student is verified and since he is authenticated to be a student, he successfully logs onto the application. On the student dashboard, he navigates his way to Mathematics and Quantitative Section to find a tutor for Pre-Calculus II. He then navigates through the list of tutors available and finally finds a tutor for Pre-Calculus II. He then views the tutor's data on their available days and times for tutoring and takes note of the days and times for tutoring. after which he logs out of the system. Other functionalities include Michael having access to a rating system to evaluate tutors, sending tutors e-mail on the platform, engaging in a chat with them, etc.
<u>Data</u>	Tutor Information, Availability of Tutors
<u>Stimulus</u>	Verification of User's access control by the application
<u>Response</u>	Students successfully receiving tutors information as well as available times and days for tutoring
<u>Comments</u>	The tutors should be approved by the departmental coordinators before appearing as a tutor on the application.

2.8.4 Conceptual Database Design of the Ashesi Tutor Management System

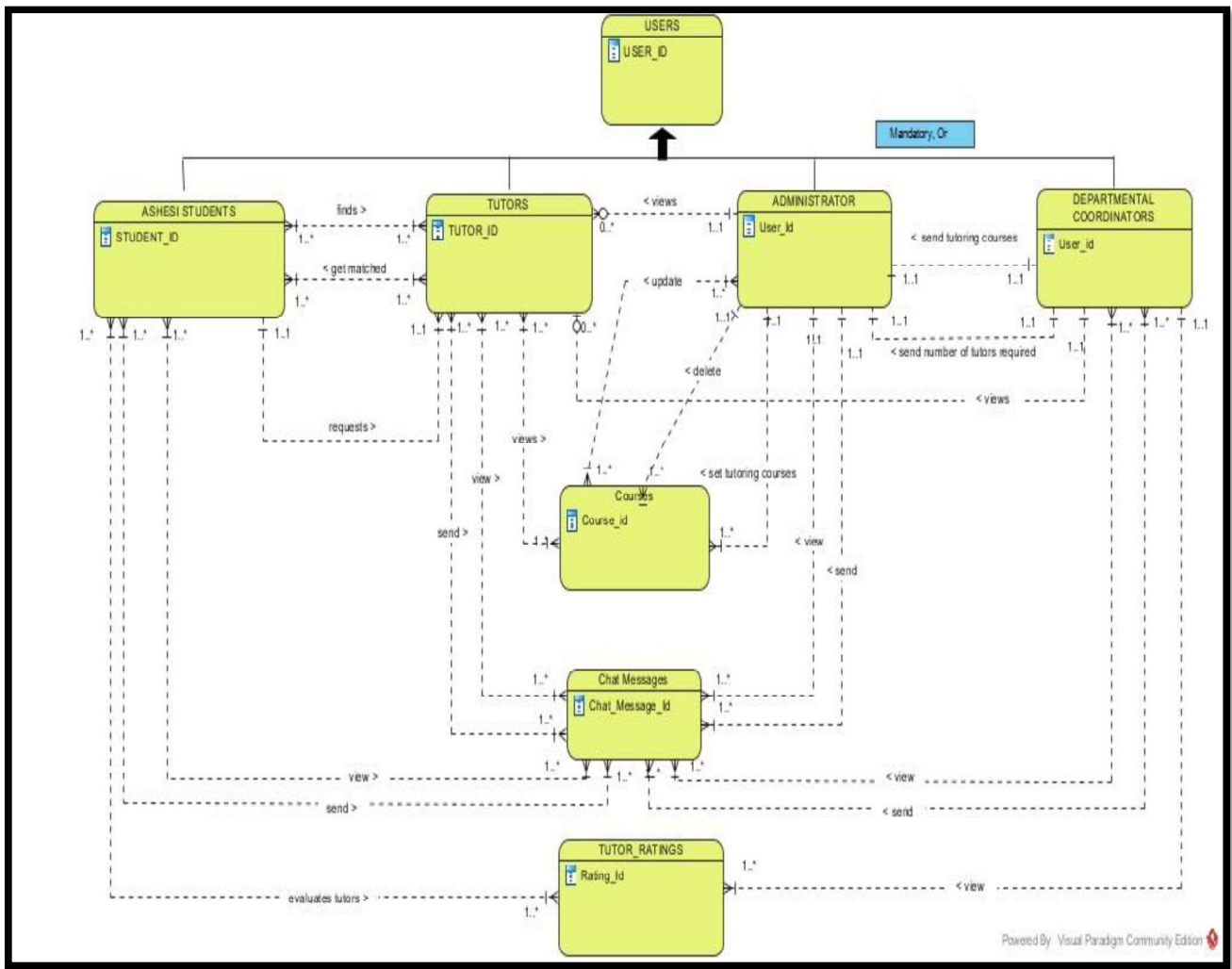


Figure 2.8.4: Entity Relationship Diagram for the Tutor Management System

Chapter 3: Architecture and Design

3.1 Architecture Overview

In building a software system, designing the architecture gives an understanding of how the software system should be organized as well as the overall structure of that system. For this Tutor Management System, three architectures have been developed to provide us the in-depth structure of the system.

3.1.1 Model View Controller Architecture

Table 3.1.1: MVC Architecture Pattern for the Tutor Management System

Name	Model View Controller (MVC)
Description	The MVC architecture shows the separation of the Tutor management system into three logical components that interact with each other. The View component shows the front-end of the application the users would be interacting with. When the user performs a functionality in the system, the command goes to the Controller component which processes the user's request by querying it into the database to be handled by the Model component , which manages the system data and associated operations on that data. When the processed request is complete, the output is then projected back to the View component to be seen by the user.
Example	Figure 1 shows the architecture of a web-based Tutor Management system organized using the MVC pattern.
When Used	The MVC architecture is used when in a given system, there are multiple ways of viewing and interacting with data. For example, in the given Tutor Management System, where both Tutors and Students, as well as Administrator and Departmental Coordinators, would be known to be interacting with the data.
Advantages	With the MVC architecture, the development of the Tutor Management System would become very fast. This architecture would make debugging also very easy.
Disadvantages	The MVC architecture can involve code complexity when the data model and interactions are simple.

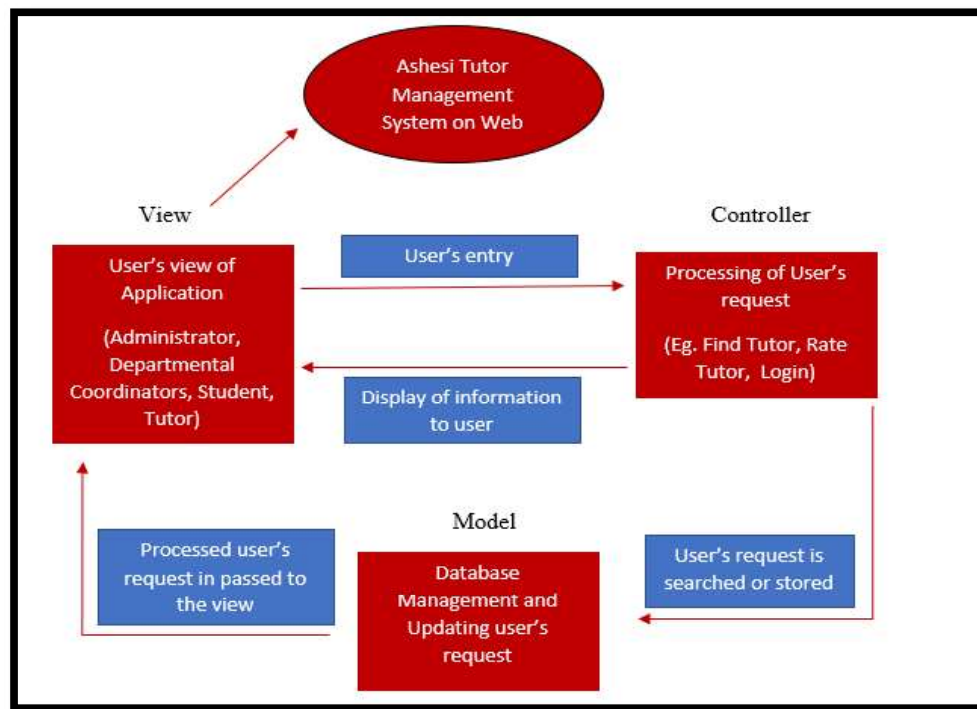


Figure 3.1.1: Model View Controller Architecture for the Tutor Management System

Justification for implementing a Model View Controller Architecture

The Tutor Management System to be implemented is a web-based application that would be structured into three logical components that interact with each other. Structuring this application in this manner would establish the division and effective execution of the View, Model, and Controller functions in terms of user experience, class definitions, and management of user interactions. The four key groups of users are the Administrator, Departmental Coordinator, Students, and Tutors tells that there would be multiple ways of viewing and interacting with data therefore using this architecture would help in the viewing and interaction of data by these multiple key users. Since also, the research approach used for the implementation of the tutor management system is the Agile Methodology, using of this architecture would help to make room for future requirements in terms of interaction and presentation of data.

3.1.2 Layered System Architecture

Table 3.1.2: Layered System Architecture Pattern for the Tutor Management System

Name	Layered System Architecture
Description	This architecture organizes the Tutor Management system into layers with related functionality associated with each layer. The browser-based User Interface layer shows the Landing page of the application on the browser-based hosting platform. From there they would arrive at the User Validation layer which would see to the verification and authentication of users before logging them into the system. After passing the User Validation layer, the user lands in the Application Functionalities layer and would be able to perform system functionalities. When the user for instance a Student decides to engage in a chat application with tutors, past information of chat is retrieved from the System Support (Database) layer which stores system data and other relevant information.
Example	Figure 2 shows the architecture of a web-based Tutor Management system organized using the Layered System Architecture pattern.
When Used	The Layered system architecture is used when building sub-systems on top of existing systems.
Advantages	The Layered system architecture would allow the replacement of layers that make up the Tutor Management system very easy so long as the interface is maintained.
Disadvantages	With this architecture, the performance can be a bit of a problem because of multiple levels/sub-systems of interpretation of a service request as it is processed at each sub-system/layer.

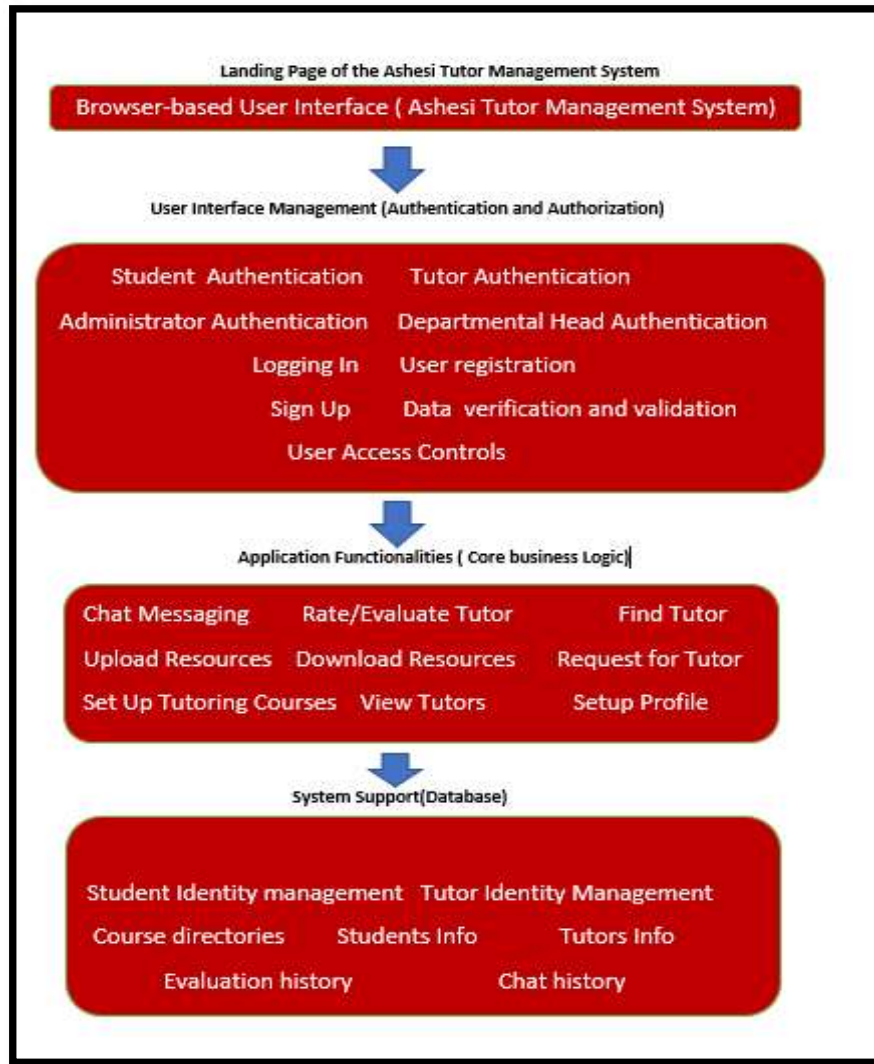


Figure 3.1.2: Layered System Architecture for the Tutor Management System

Justification for implementing a Layered System Architecture

The layered information system architecture would also effectively help in the identification of the sub-systems within the Tutor management system and what they all provide each other in terms of functionalities when divided into layers. For this Tutor management system, some key sub-systems are the multi-authentication, user interface, and database systems. The user interface layer would act as the front elevation in helping the user interact with the system, the multi-authentication layer would see to the thorough

verification of users as well as their access controls while the database would see to the storing of relevant information such as usernames, passwords, etc. So with the implementation of this architecture, the related functionalities associated with each layer would be very visible due to the structuring of the layers.

3.1.3 Repository Architecture

Table 3.1.3: Repository Architecture Pattern for the Tutor Management System

Name	Repository Architecture
Description	<p>In this architecture, all data in the Tutor Management system is managed in a central repository that is accessible to all system components. For example, the Student Repository would be accessible to the <i>user validation component</i> to sign in and register students into the system or to chat with a tutor. The Tutor Repository also, for example, would be accessible to the <i>student rating component</i> to allow students to rate and evaluate a tutor. The Chat Repository also would be accessible to the <i>chat messaging component</i> to enable students, administrators, tutors, departmental coordinators to store chat messages to history for future retrieval. The Courses Repository would be accessible to <i>upload courses component</i> to enable Administrators to store courses for tutoring</p> <p>NB: Components do not interact directly, only through the repository.</p>
Example	Figure 3 shows the architecture of a web-based Tutor Management system organized using the Repository Architecture pattern.
When Used	This architecture is used when you have a system in which large volumes of information are generated that has to be stored for a long time.
Advantages	This architecture would enable the data to be used in the Tutor Management system to be managed consistently. Also, another benefit is that changes made to one component which uses either the Courses, Student, and Tutor repository can be propagated to all components.
Disadvantages	The repositories act as a central storage location for system data to be used in the Tutor Management System hence, a single point of failure can jeopardize and affect the whole system.

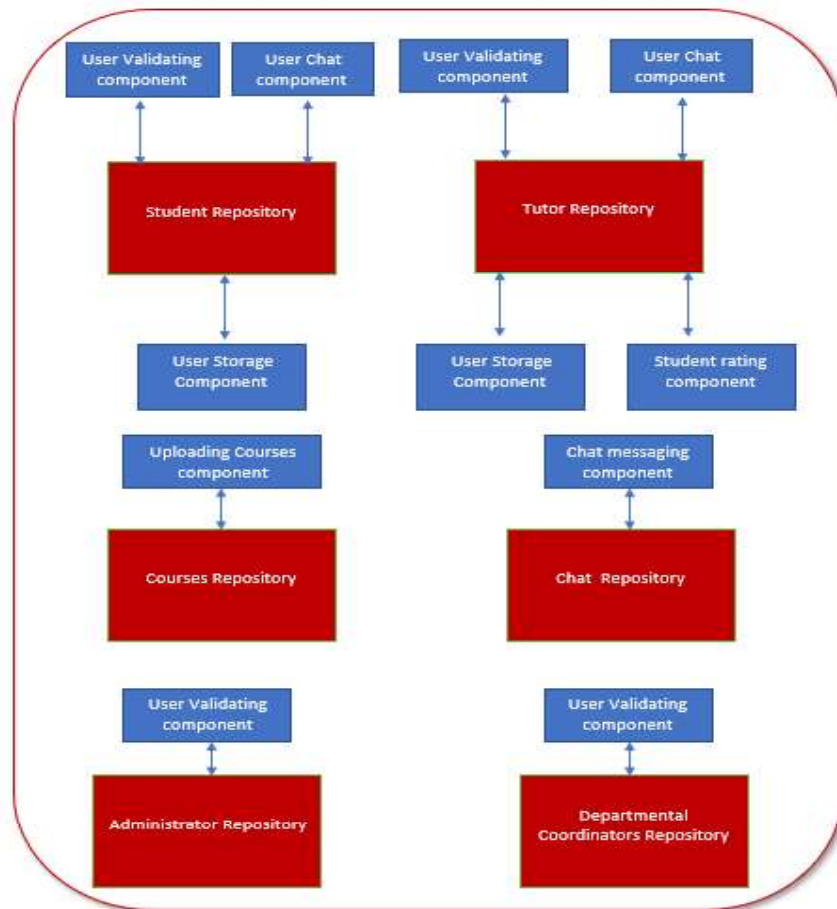


Figure 3.1.3: Repository Architecture for the Tutor Management System

Justification for implementing a Repository Architecture

In the Ashesi Tutor management system, students would log into the system to find a vast array of tutors to assist them in their troubling courses. Therefore having vast storage to store tutor and students details. Users would be interacting with the chat application hence, the chat messages should be stored for future references This is why the repository architecture is important in my project. With the implementation of this architecture, all student and tutor data in a system would be managed in a central repository that would be accessible to all system components for multi-user authentication, and retrieval of data for immediate use chat messaging, uploading of courses, etc.

Chapter 4: Implementation

4.1 Front-End Development

Front-End Development defines the looks and graphical layout of the application. They also form the point of contact users would interact with when using the application. The tools and academic disciplines used for the front-end implementation of the application include HTML(Hypertext Markup Language), CSS(Cascading Style Sheet), JavaScript, AJAX and jQuery, SweetAlert library, Font Awesome toolkit, EmojioneArea plugin, bootstrap framework, and Human-Computer Interaction(HCI) design concepts. The reasoning behind the use of the above-stated frameworks and tools used are explained below.

4.1.1 Human-Computer Interaction (HCI)

Human-computer interaction(HCI) concepts define the approach to designing your application that meets the needs of the user and the purpose of the application. HCI was utilized to help in the designing of the Front-End web pages to improve users' interaction and experience. The buttons, icons, coloring, structure of every page were implemented having in mind the Gestalt principles, Cognitive lenses, and the user-centered design model.

4.1.2 HTML,CSS and JavaScript

HTML, CSS, and JavaScript form the basics of programming with regards to the web. The Ashesi Tutor Management System is an application hosted on the web hence these tools were essential for its development. HTML(Hypertext Markup Language) helped with the overall structure of all the web pages within the system and its compatibility with the web browser. CSS(Cascading Style Sheet) helped with the styling

of all the web pages with colors to make them stand out. JavaScript helped with the programming of the front-end concerning issuing alerts to the user, user validation, and verification checks on the Register, Log In, and the Chat Application portal.

4.1.3 AJAX and jQuery

AJAX and jQuery were used to introduce advanced JavaScript functionalities within the system. For instance, they were used to help Administrators easily search for the number of tutors for a specific by just typing out the course name and projecting the tutors out using the "onkeyup" feature. They were also effectively utilized in the creation of the Chat Application to help project user chat histories and the overall setup of the Chat Application.

4.1.4 SweetAlert library

SweetAlert is a JavaScript library used to enrich alerts issued out to users more beautifully and conveniently. Its inclusion in the implementation of this application was to help make the appearance of alerts to the user in the Register and Sign In, and other subsequent pages appear nicely as a way of achieving user-friendliness.

4.1.5 Font Awesome Toolkit

Font-Awesome is an icon toolkit that provides recognizable icons to your application to help users easily identify what the purpose of a button or a section is. It also helps to improve the user experience. Hence using this knowledge, Font-Awesome, an icon toolkit was used in all the web pages to introduce icons in the system to help users to recognize what a particular section or a button seeks to achieve

4.1.6 EmojioneArea Plugin

As part of making interaction in the Chat Application very interesting, EmojioneArea is a jQuery plugin that helped to introduce emojis in the Chat Application. This plugin helped users to use this medium of expression aside from the normal texts.

4.1.7 Bootstrap Framework

Bootstrap framework helped with the provision of foundational front-end design blocks upon which the Ashesi Tutor Management System pages were built on. It made the structuring of the front-end layout very easy to configure and re-configure to suit the necessary design requirements.

4.2 Back-End Development

Back-End Development defines the server-side programming aspect of the application. The main programming language or tool used for the back-end implementation of the application is PHP (Hypertext Pre-Processor). The reasoning behind the use of PHP is explained below.

4.2.1 PHP(Hypertext Pre-Processor)

PHP(Hypertext Pre-Processor) stood out as the main server-side language for back-end development because it is an open-source application, easy and simple to code, as well as easy to maintain. PHP was used to retrieve user inputs from various forms within the application using the (\$_POST) method to be sent to the database. Using the (\$_SESSION) feature, PHP helped to avoid manual entry into any page from the URL bar and also helped to ensure user independence in all pages. PHP was used to assist JavaScript in terms of validations, verifications at the Register and login pages and used extensively in the Chat Application. Using PHP, the online and offline status of users was

made possible to allow users to know other users' availability state. Notifications after sending a chat message were also possible using PHP. Also, methods such as "implode()" in PHP helped with the receiving of multiple user inputs per column and storing them in the database.

4.3 Database Development

Database Development stands as the third-tier architecture when it comes to web programming. The main tool used for the database implementation includes Visual Paradigm, SQL (Structured Query Language), and PhpMyAdmin. The reasoning behind the use of these tools is explained below.

4.3.1 Visual Paradigm

Visual Paradigm was the first tool used in the development of the database for the Ashesi Tutor Management System. Constructing the database without a conceptual design of a blueprint can be problematic hence Visual paradigm was used to develop a conceptual database design of how the database is going to look like. Figure 2.8.4 presents the Entity Relationship Diagram with regards to the Conceptual Database Design of the Ashesi Tutor Management System.

4.3.2 Structured Query Language

SQL(Structured Query Language) was used to retrieve users' information from the database for usage. Chat Messaging used SQL to retrieve chat history from the database to foster easy recovery of data for use. All tutor details made visible to students upon request was as a result of SQL queries. Further into SQL, the "join" operator was used to link the Tutors and Students table to arrive at an effective matching of a student to tutor based on some of the users' attributes.

4.3.3 PhpMyAdmin

PhpMyAdmin was used as an open-source application to harbor the database for the Ashesi Tutor Management System. It allowed easy retrieval and storage of users' data, courses, chat messages, tutor ratings, etc. Since XAMPP was used for the local development, using PhpMyAdmin was a suitable choice to help interface my database system for my application with the local server.

4.4 Local Development

XAMPP server was implemented and used in the local development of the application with regards to the front-end, back-end, and database developments. XAMPP was best preferred because of the choice of technologies used in the development of the Ashesi Tutor Management System. These include PHP for the server-side language and usage of MySQL for database development. XAMPP presents itself to suit the occasion as it is best interfaces with my Windows OS.

4.5 Implementation Procedure

First and foremost, the methodology used in the development of this project was the Agile Approach. During the implementation process, Agile helped to adapt to the idea of change in the software functions. The implementation process for this project would be described based on the individual user dashboards and the embedded features.

4.5.1 Landing Page of Application

The first page upon launching the Ashesi Tutor Management System aims to introduce users to the vision of the application as well as the various aspects, features, functionalities within the system. This includes the various sections such as the

Motivation, Users, Chat Application, and Student-Matching. The Motivation seeks to explain the problem statement behind the project as well as the significance of the Ashesi Tutor Management System. The users' section aims to inform arriving users of the available users on the platform. The Chat Application seeks to inform users about a messaging section all users can use for communication while the Student-Matching section explains the Student-Matching portal and how it works.

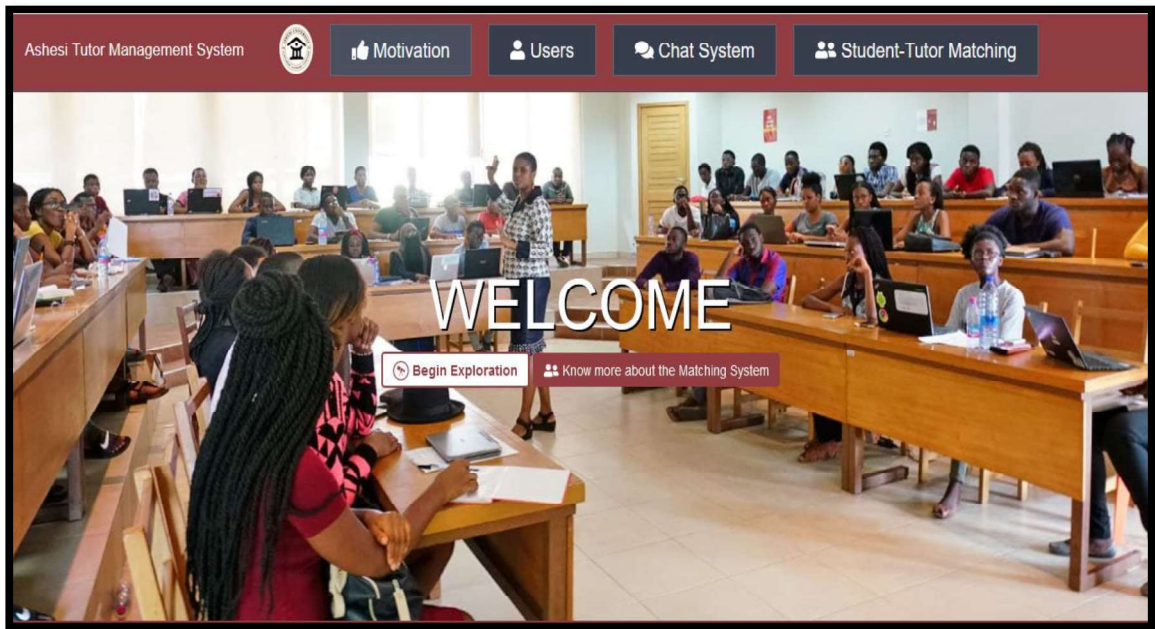


Figure 4.5.1: Landing Page of Ashesi Tutor Management System

4.5.2 Administrator Dashboard

The Administrator portal introduces the Admin to the portal of supervision and the power to oversee the tutoring courses set. At the index page of the Administrator Dashboard, the Admin has the power to view all tutors categorized based on departments, set tutoring courses publicly so tutors know which courses are available, view the set courses, and view the various courses from the departments and the number of tutors needed per course. Administrators are pre-configured into the system and hence only log in using their pre-configured credentials.

Using AJAX and jQuery, the Admin can filter through the tutors in the designated courses to know the number of tutors currently available per course. They receive preferred courses for tutoring from the departmental coordinators from all departments and based on that sets these courses publicly so tutors can see which courses require tutors. The Admin can view, update and delete a course when the number of tutors needed per course is full. The Admin can know this through the capability of filtering using AJAX and jQuery. Admin can also engage in the Chat Application Portal to be in communication with all other users on the system.

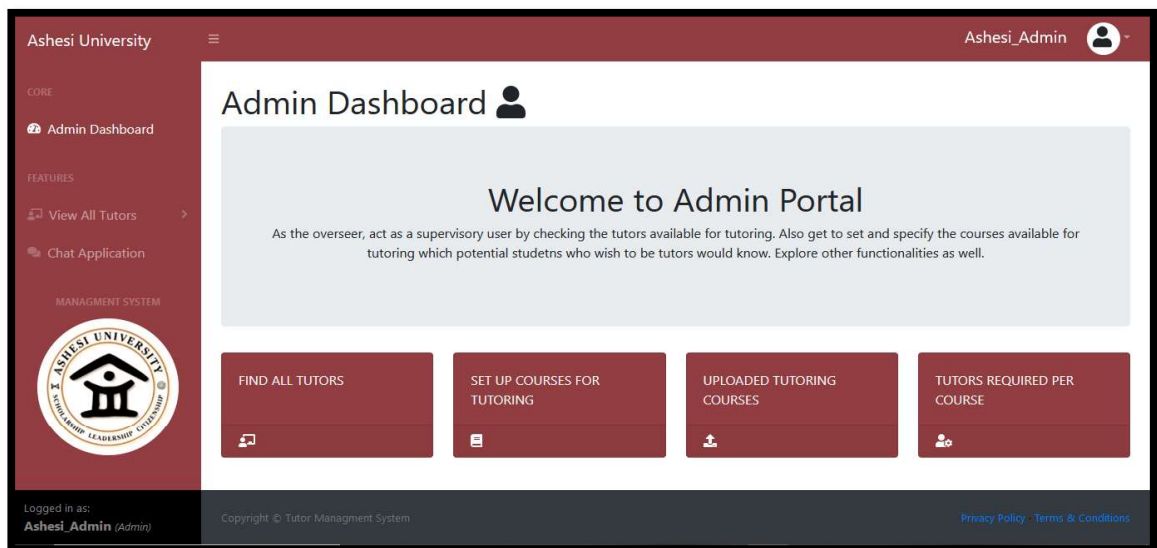


Figure 4.5.2: Administrator Dashboard

4.5.3 Departmental Coordinators Designated Dashboards

The Departmental Coordinators are categorized into four groups of users who perform the same role but these roles apply to their designated departments. These departments are the Computer Science and Information System, Business Administration, Engineering and Humanities, and Social Sciences Department. Departmental Coordinators just like the System Admin are pre-configured into the system. In their various dashboards, they have the privileged to view their designated department tutors, set

courses so the Administrator can know the courses needed for tutoring, view these courses made publicly available by Admins, and view their department tutor reviews/ evaluations from students. Departmental Coordinators also act as the secondary overseers of the Tutoring process but are limited to only their respective departments.

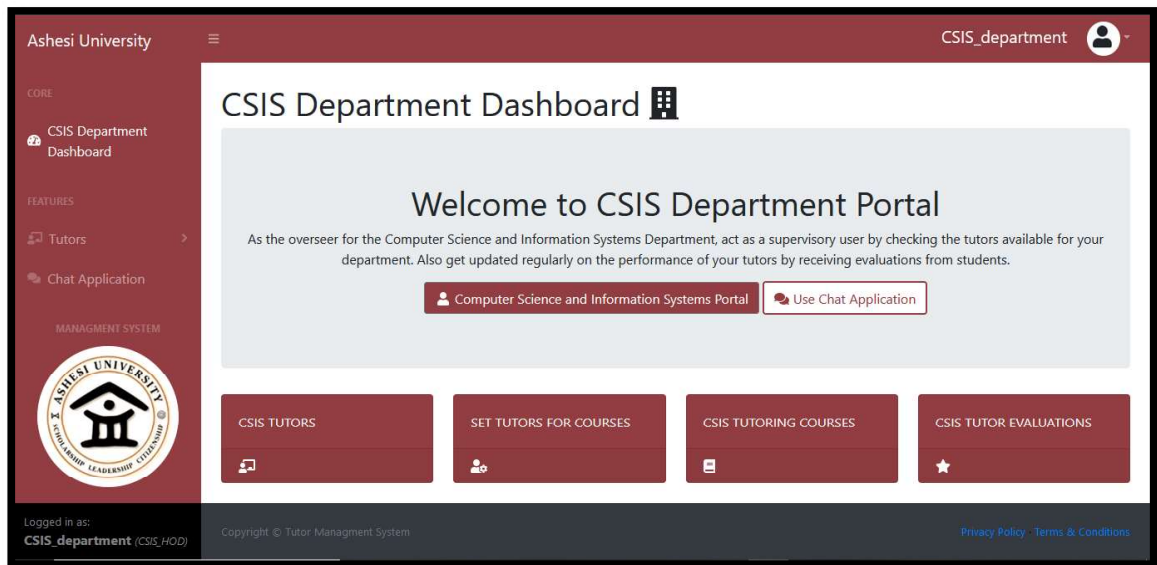


Figure 4.5.3a: Departmental Coordinators Dashboard: CSIS Department

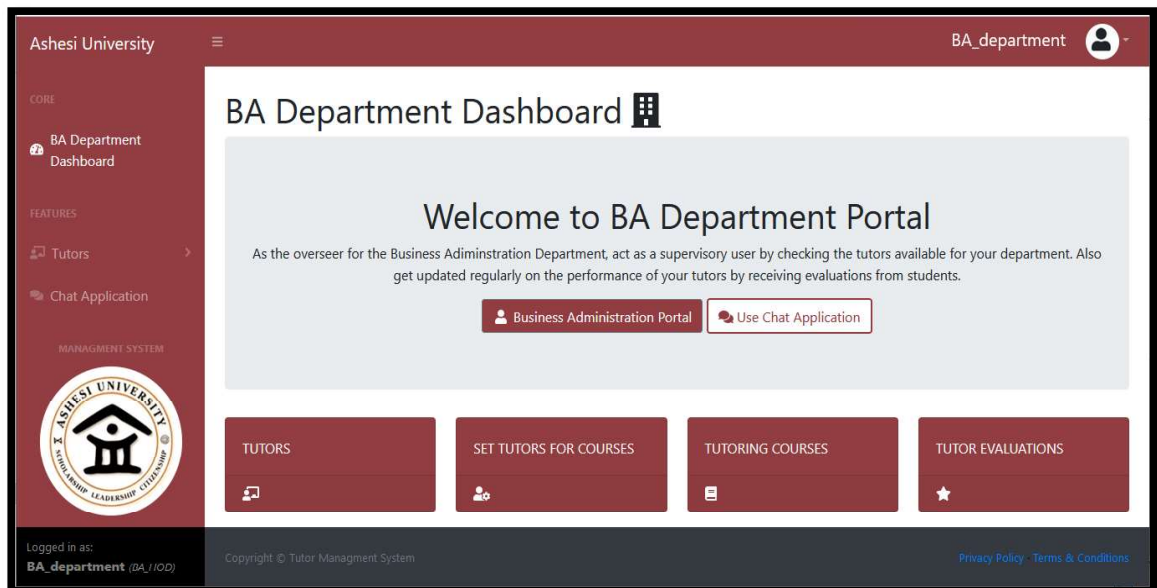


Figure 4.5.3b: Departmental Coordinators Dashboard: Business Administration

Department

4.5.4 Student Dashboard

In the Student-designed dashboard, the interface provides the student with the option of Finding a tutor categorized based on departments and the year the course is taught for easy identification of tutors. Students upon finding a tutor for their course can begin communications with the tutor through the Chat Application portal. Students can also download academic content uploaded by tutors and also rate and evaluate tutors regularly on their tutoring performance. To safeguard the identity of students who perform these evaluations, their uploaded evaluations of a tutor are made visible only to the Departmental Coordinator for the department whose course the tutor is helping the students with. Also, the name of the student is kept hidden.

To improve students' participation in the system and avoid the stress in navigating through the tutor lists searching for a tutor, the student can request a tutor in the Student-Tutor Matching portal. In this portal, students' weaknesses being it with the Theoretical, Prerequisite/Foundation, or Practical aspect and their skill level(Beginner or Intermediate) in their challenging courses are taking. In using this information, the portal makes a match of the student to a tutor. As an Ashesi Student seeking help in a specific course, the Ashesi learning goals for their tutoring process are also required in this portal to act as a check on their progress.

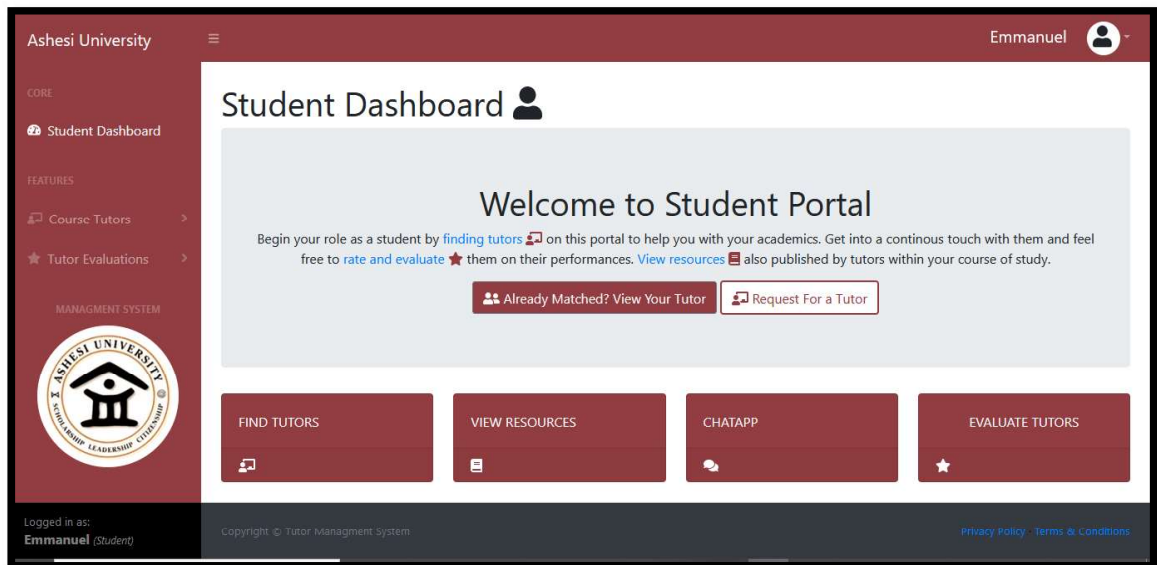


Figure 4.5.4a: Student Dashboard



Figure 4.5.4b: Student viewing a resource and about to download

Ashesi University

Emmanuel

CSIS-Computer

Give Tutor a rating (Out of 5)

★★★★★

Tutor Name:

Michael

Tutoring Course:

ICP

Tutor Department:

CSIS-Computer Science and Information Systems

Evaluation Message:

Comment on Tutor performance

Submit

Close

Logged in as: Emmanuel (Student)

Figure 4.5.4c: Student Evaluating/Rating a Tutor

4.5.5 Tutor Dashboard

The Tutor Dashboard presents tutors with the privilege to set up their tutor profile which would be visible to students who want to find or request a tutor. They can also update their profile picture and photo at any time. Other functionalities include the privilege to upload and view an academic resource onto the system which would be visible to students in need of resources for course help. They can engage in the Chat Application with Students and other users and interact with the Student-Matching Portal. In using this feature, they are required to provide information about their Skill-Level for their tutoring course and their strengths in this course. As an Ashesi Student who is a tutor, the Ashesi learning goals are also required here to act as a check on your tutoring progress.

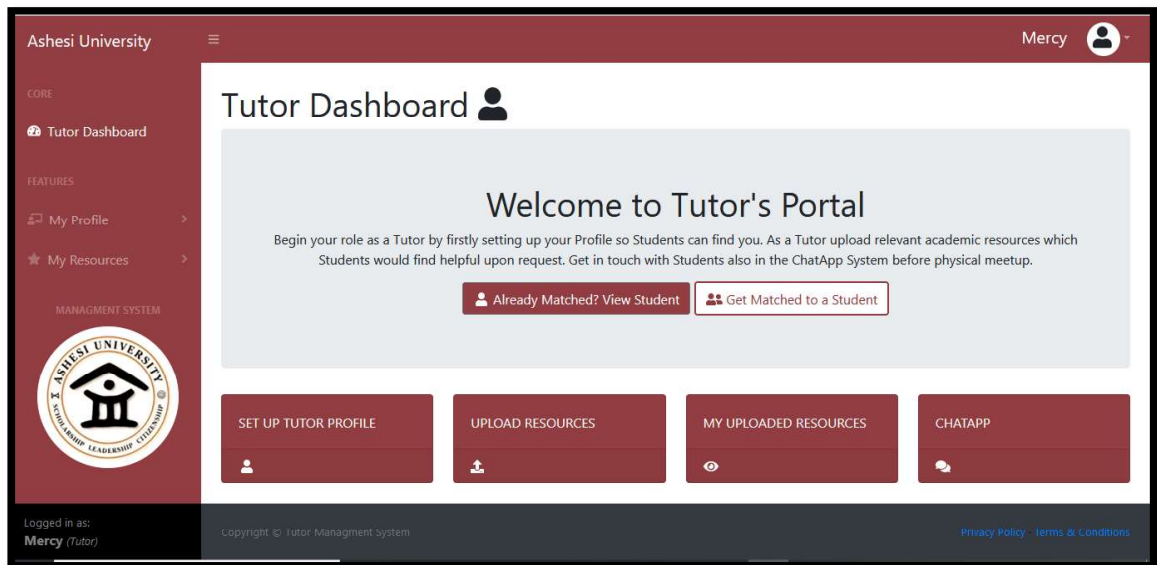


Figure 4.5.5a: Tutor Dashboard

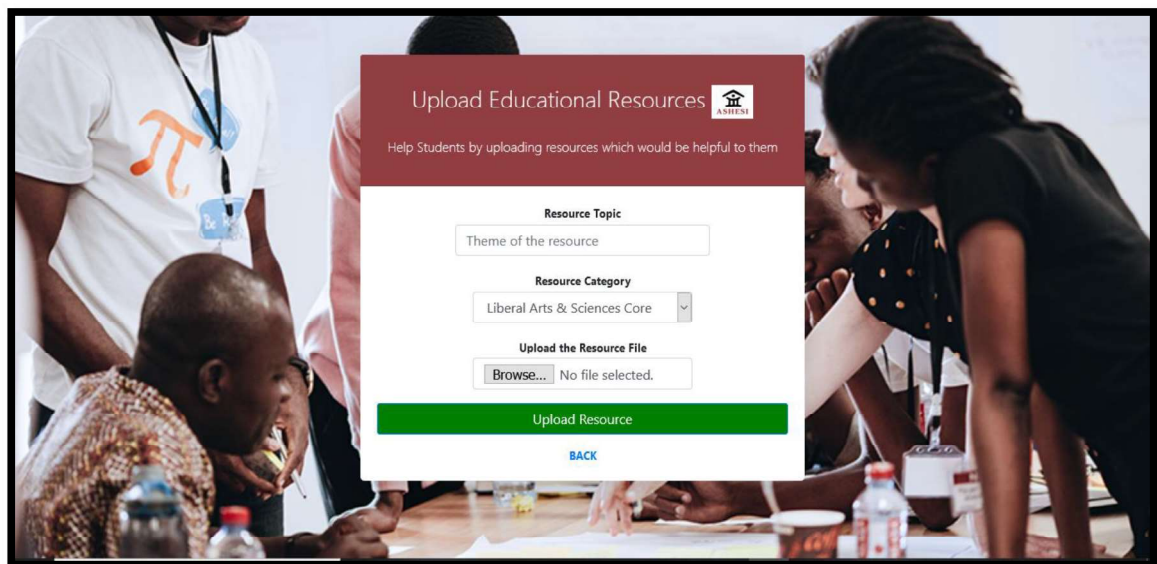


Figure 4.5.5b: Tutor Upload Resource Page

4.5.6 Chat Application

As part of delivering services to users of the Ashesi Tutor Management System, the Chat Application embedded in this project seeks to establish a user to user communication while on the platform. The Chat Application is accessible to all users especially, Students and Tutors with the sole purpose of helping them to have an interaction on the tutoring course before a physical meetup. Using PHP, AJAX, and jQuery, the Chat Application is set up to improve on the user's experience and interaction through relevant features such as online-offline status to show the current availability of users, notification feature to alert users off any incoming message, and privacy concerning secured communications on the portal.

Upon launching the Chat Application, users arrive at the Chat Login page to log in with their usernames. Upon validation, users arrive on the Chat Application landing page designed to improve their user experience and interactions. All available users their roles on the system i.e. Student and Status thus whether online or offline is visible to enable each user to know the availability status of others. To chat with a particular user, each user has a "Send Chat" button attached to their names. Upon clicking on it, a popup chat box appears with the receiver username and relevant features such as emojis provided by the "EmojiArea" jQuery plugin to allow the expression using emoticons to entice the conversation. Users can also delete any sent message they prefer and this operation deletes it on both the receiver and sender section.

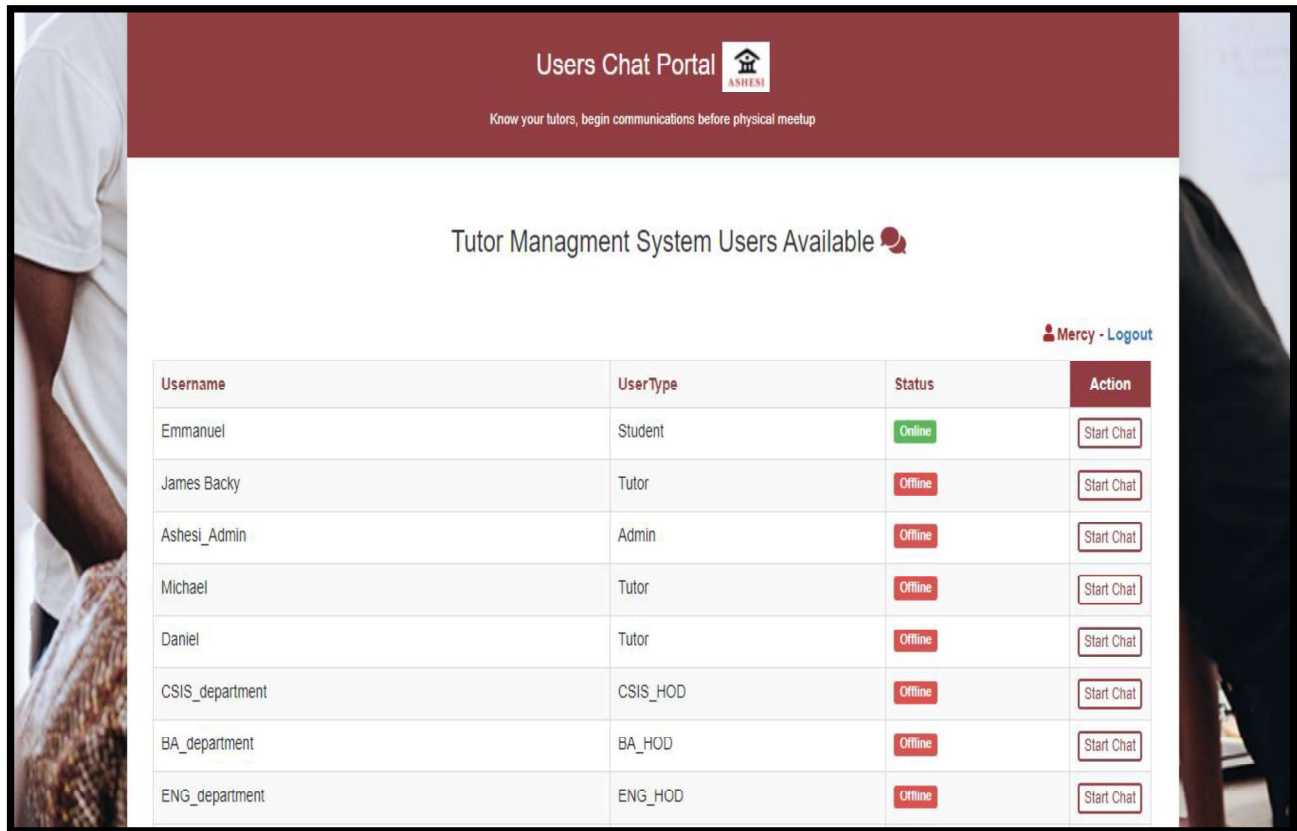


Figure 4.5.6a: ChatApp Portal showing users and Status

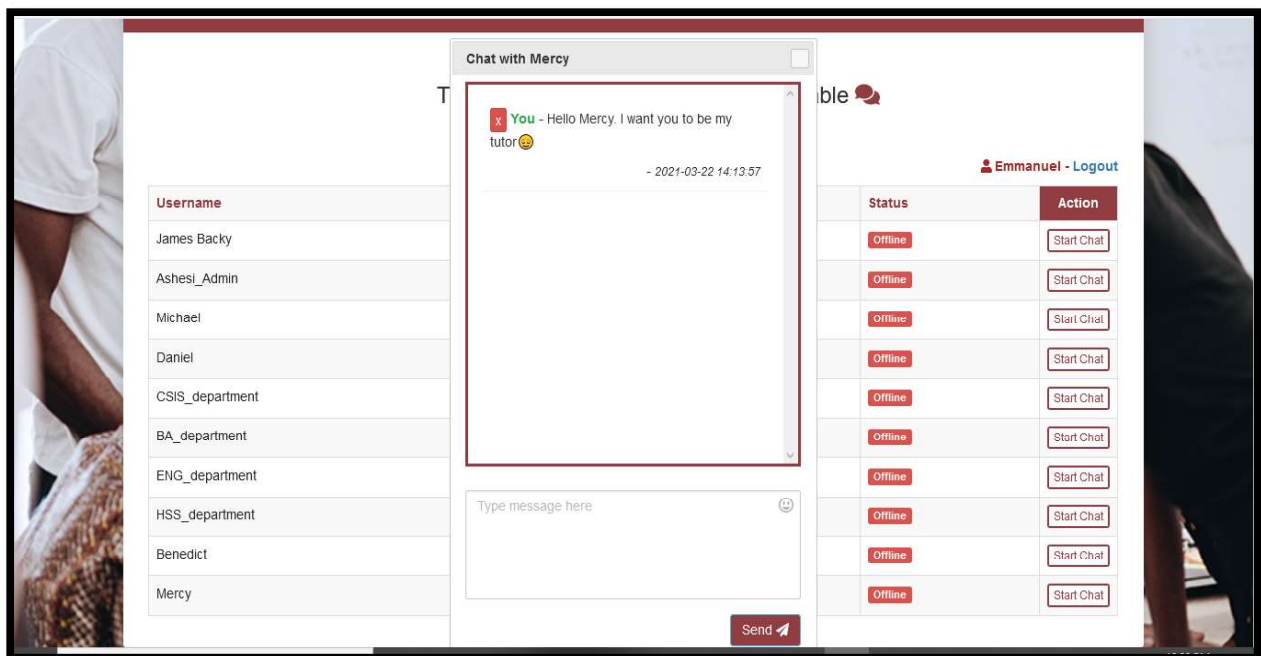


Figure 4.5.6b: Student chatting a Tutor, Mercy

4.5.7 Student-Tutor Matching Portal

The Student-Tutor Portal saves students the time and energy in navigating throughout the tutors' profiles to find which tutor suits them. One benefit of this portal is that it allows students to express the challenges that hinder them from progressing in a course being it the theoretical, Foundational/Prerequisite, or the Practical aspect of a given course. Other information taking from the student includes their skill level in the course, their mode of tutoring they prefer (One on One or Group-based), the topic they need help with, their expectations, and other relevant information. Tutors also using this portal can update their profile with their skill level, strengths in a course, and mode of tutoring. Other relevant information such as the applicable Ashesi learning goals to achieve are required by the system from both student and tutor to help each other track their progress with these set goals. Upon submission of a request for a tutor, students are provided a link to a page that shows them their matched tutors. Tutors are also provided a link to a page that shows them their matched student. Below is the layout of the Algorithm.

Student - Tutor Matching Algorithm

Input: A set of m major, mod mode of tutoring, l skill level, c tutoring course for help, t topics, s strengths, and w weaknesses by Students and a set of mod mode of tutoring, c tutoring course, l skill level, s strengths by Tutors

Output: A matching of Student to Tutor

Step 1: While c tutoring course for help by student S and c tutoring course by tutor T are equal and l skill level of tutor T is greater than l skill level of student S :

Check for equality between *mod* mode of tutoring by student *S* and *mod* mode of tutoring by tutor *T*.
Each *w* weaknesses of student *S* should be cross-checked with *s* strengths of tutor *T* to establish a match.

Step 2: If no matching, repeat Step 1.

Step 3: Return a matching of *S* student to *T* tutor.

For the code implementation, after students provide the information required by the system for matching, the system effectively in using the SQL Inner Join feature and the use of comparison operations finds a suitable tutor match for the student. The process of matching happens simultaneously hence once a tutor or a student is matched, each user automatically sees the profile information of their matched user immediately.

```
//SQL Inner Join query for matching a tutor to a student

$sql = "SELECT * FROM tutors INNER join students ON
tutors.tutoring_course =students.course AND
tutors.tutoring_mode = students.tutoring_approach AND
tutors.prerequisite = students.prerequisite OR tutors.theory
= students.theory OR tutors.practical = students.practical
WHERE userstudent_id = '$student_id' AND
tutors.skill_category != students.Skill_level";
```

Figure 4.5.7a: Code snippet of Inner Join Query at Student Page to let Student view
matched Tutor

```
//SQL Inner Join query for matching a student to a tutor

$sql = "SELECT * FROM students INNER join tutors ON
students.course =tutors.tutoring_course AND
students.tutoring_approach = tutors.tutoring_mode AND
students.prerequisite = tutors.prerequisite OR
students.theory = tutors.theory OR students.practical =
tutors.practical WHERE usertutor_id = '$tutor_id' AND
students.Skill_level != tutors.skill_category ";
```

Figure 4.5.7b: Code snippet of Inner Join Query at Tutor Page to let Tutor view matched Student

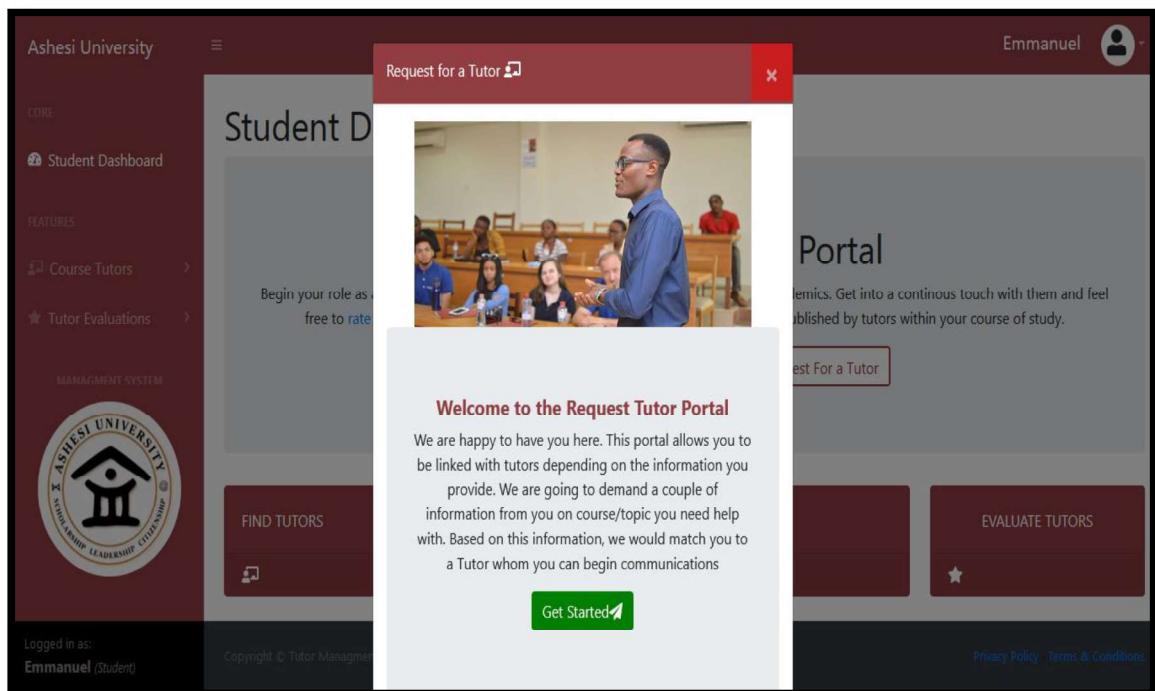


Figure 4.5.7c: Student using the Student-Tutor Matching Portal

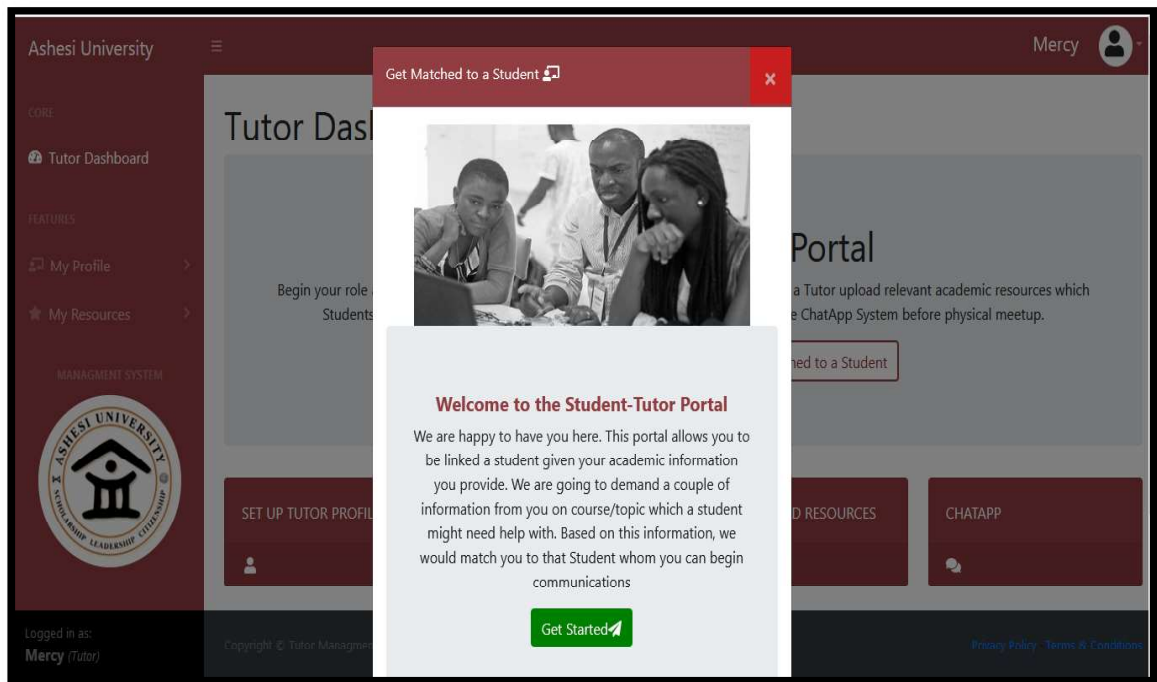


Figure 4.5.7d: Tutor using the Student-Tutor Matching Portal

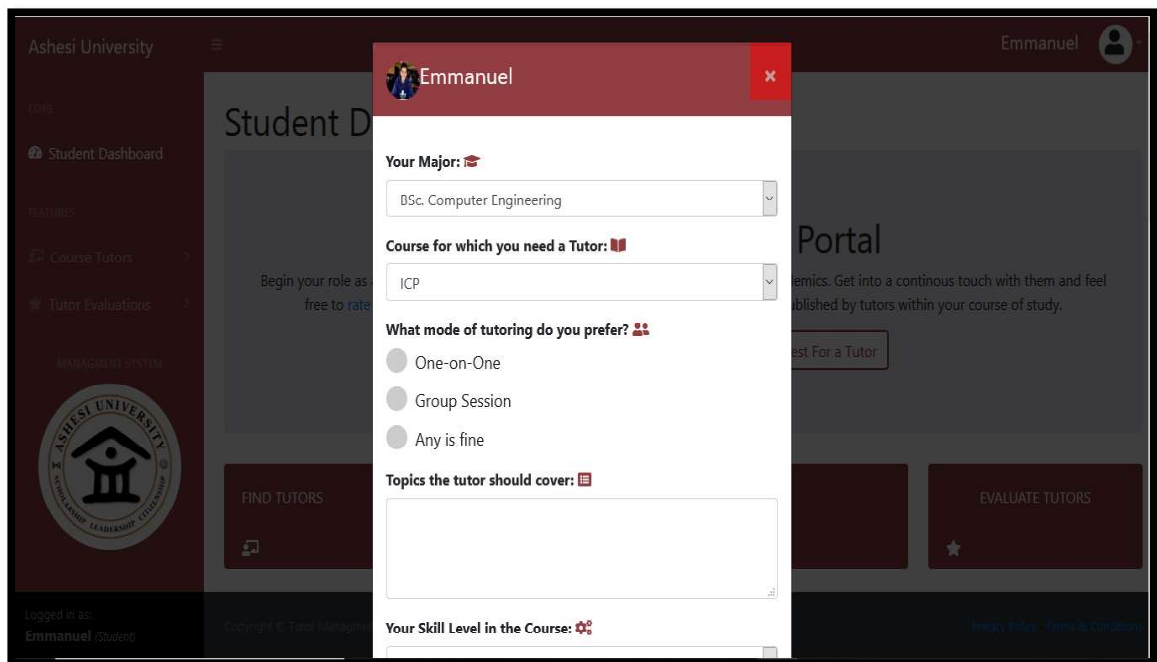


Figure 4.5.7e: Student providing info before matched to Tutor

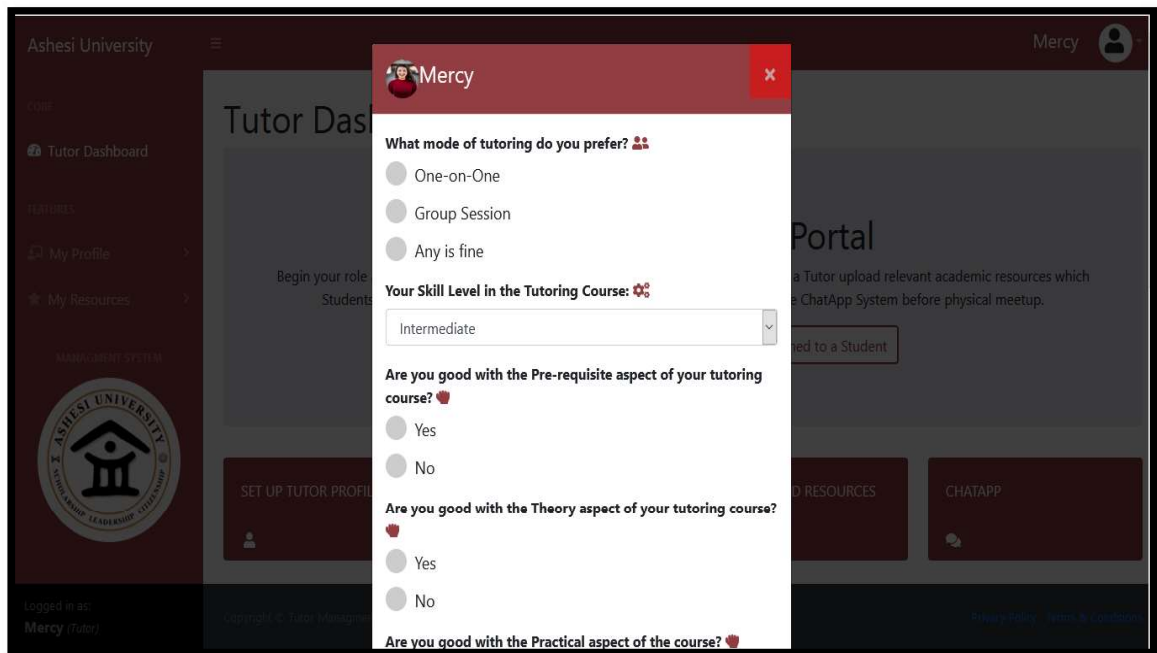


Figure 4.5.7f: Tutor Providing Info before been matched to Student

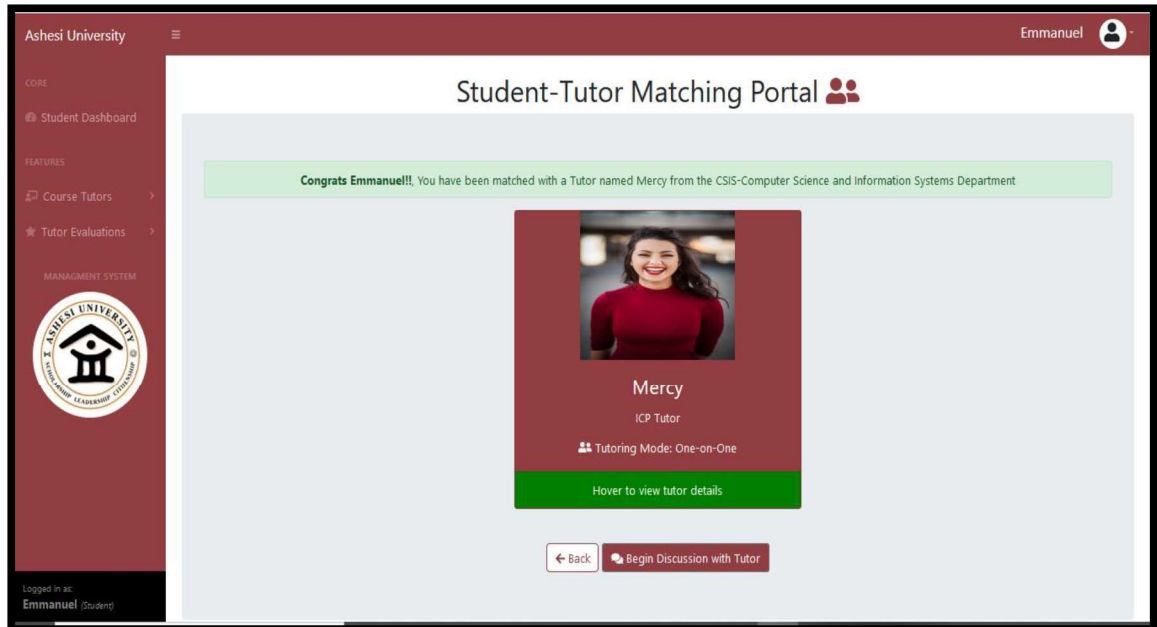


Figure 4.5.7g: Student gets matched to a Tutor

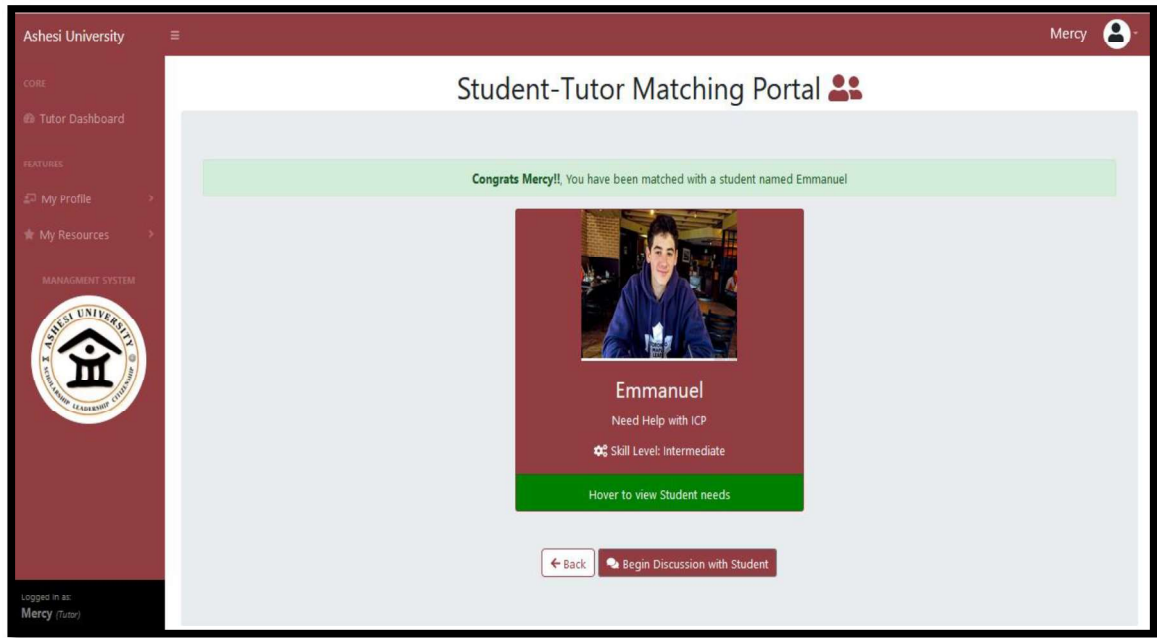


Figure 4.5.7h: Tutor gets matched to a Student

4.6 Software Deployment

The Ashesi Tutor Management System has been successfully deployed onto a live web server hence making its accessibility by users and stakeholders very easy. The software tools used for the deployment of this application are FileZilla, 000webhost hosting platform, and PhpMyAdmin(Database Manager within 000webhost). 000webhost web server helped with the hosting of the application hence making it accessible on port 443 (HTTPS). FileZilla helped with the transfer of the project files and folders to the public HTML directory on the 000webhost platform. The database of the project was also imported into 000webhost's database manager, PhpMyAdmin for easy retrieval and storing of data. 000webhost was best preferred because of its database extension with PhpMyAdmin. During the development of the application in localhost, PhpMyAdmin was used as the database manager hence with it been the database manager also for 000webhost, integration of the database for the project was easy.

The URL for the Ashesi Tutor Management System is

<https://ashesitutor.000webhostapp.com/>

Chapter 5: Testing And Results

The Ashesi Tutor Management System is made up of various components which were been used to implement the functional requirements as well as additional features. To prove the performance of these functionalities, testing was conducted in two phases. The first phase which is the Development testing looks into the testing of the various components, units, and the Ashesi Tutor Management System as a whole from the developer's view. An automated testing tool such as Selenium was also utilized for automated testing under Development testing. The second phase was the User/Acceptance testing where the users and stakeholders of the application tested the functionalities and components of the System.

5.1 Development Testing

5.1.1 Unit Testing

Administrator Unit testing

For Administrators, unit testing was performed on functionalities such as the “Set Up Tutoring Courses”, “Find Tutors”, “View Uploaded Tutoring Courses”, and “View Tutors Required Per Course”. From the development tests, all these functionalities were successful. For instance, the Administrator upon clicking on the “Set Up Tutoring Courses” tab is moved to a form page where he/she can set a tutoring course received from a departmental coordinator through the system. This tutoring course would be visible to students who wish to be tutors.

Departmental Coordinators Unit testing

Departmental Coordinators functionalities such as "View Tutors" and "View Tutor Reviews" were successful in unit testing. For instance, the CSIS Departmental

Coordinator upon clicking on the "CSIS Tutors" can view the profiles of tutors for the CSIS department. The departmental Coordinator also upon clicking on the "View Tutor Reviews" can view all the uploaded reviews from students concerning the tutoring performance of a tutor in their designated departments.

Tutor Unit testing

Tutor functionalities such as the “Set Up Profile”, “Edit Profile”, “Edit profile photo” and “Upload Resources” also proved successful during unit testing. For instance, tutors upon clicking on the Upload Resources tab are directed to a form page. Upon filling the form and choosing an academic resource from their designated directories, the resource is then uploaded and can be publicly viewed by both interested students and the Tutor as well.

Student Unit testing

Student functionalities such as "Find Tutor", "Upload Profile Photo", "View Resources" all proved successful during unit testing. For instance, students upon clicking on the "View Resources" tab are directed to a page where the academic resources are grouped based on departments. Clicking on any department tab reveals the resources uploaded by tutors from that department. These resources can be viewed on the platform or downloaded by the student.

5.1.2 Component Testing

Tutor Evaluation and Rating Portal

During the development testing, the Tutor Evaluation component as a key Student functionality was successful during testing. Students upon clicking on the "Evaluate Tutors" tab are directed to a page with tutors categorized into departments. Students upon clicking on a department, for instance, Computer Science and Information

System (CSIS) would see all CSIS tutors and upon clicking on the evaluate button attached to each user, a bootstrap modal form pops up given students the permission to rate tutors on a star scale and write an evaluating message. Upon clicking on submit, the evaluation message is sent and is displayed in the CSIS Departmental Coordinator's dashboard for review on tutors' performance.

5.1.3 Automated Testing (Selenium tool)

Launching of Ashesi Tutor Management System on the Browser

The launching of the Web Application address on the web browser was successful in the automated testing. Navigations on the index page were performed easily with the animations also been intact.

User Registration and Login

The Registration and Login of all users were successful during Selenium testing. Selenium revealed the Registration of students and tutors on the Register page. After which they were directed to the Login Page to Sign In with their credentials. Upon successful validation, they arrived in their respective dashboards. Departmental Coordinators on the other hand upon arriving in the Login Page can provide their credentials and also arrive at their designated dashboards based on departments.

User Verification and Validation

User Verification and Validation at applicable areas were all successful during Selenium testing. For instance, users especially students not providing email addresses in the Ashesi format and users providing wrong passwords during login into the application and at the Chat Application portal were denied access in the automated tests. Users who

had a mismatch in their password and confirm password at the Register page would be denied access until both passwords match.

Admin View All Tutors

Selenium test for Viewing all tutors by the Administrator was successful. The automated test reveals the ability of the Administrator to click on the "View All Tutors" tab and select a department to view its available tutors.

Tutor Setting Profile

Selenium test showed success in the test for tutors setting their tutoring profile. In the simulation by selenium, tutors upon clicking on the "My Profile" tab are directed to a page where they can set their tutoring profile. Upon successfully setting up their profile, they can be able to review their profile and be open to options of "Edit profile", "Upload Profile Photo", and "Delete Account".

Chat Application Component

Using Selenium, the automated test revealed the ability of the user to login into the Chat Application Portal and also send a message to another user successfully. The automated tests also revealed success in the ability of users to be able to send emojis together with texts, view the status(online or offline) of other users, and be notified of any unread messages in the Chat Portal. After which they can and log out successfully as well.

Student-Tutor Matching Component

The automated test successfully revealed the ability of students to launch the Student-Matching Portal and get started. The automated test also revealed the success of a student in filling the form requirements, clicking on the submit button, and been directed to the page with their matched tutor. From the tutors' perspective, the automated test successfully revealed the ability of tutors to launch the Student-Matching Portal, fill the

form requirements, click on the submit button, and been directed to the page with their matched student.

“Selenium tests were exported as JUnit test codes for future references. See section 5.1.4 for the pictorial view of the results of the above selenium tests”

5.1.4 Results from Selenium Testing

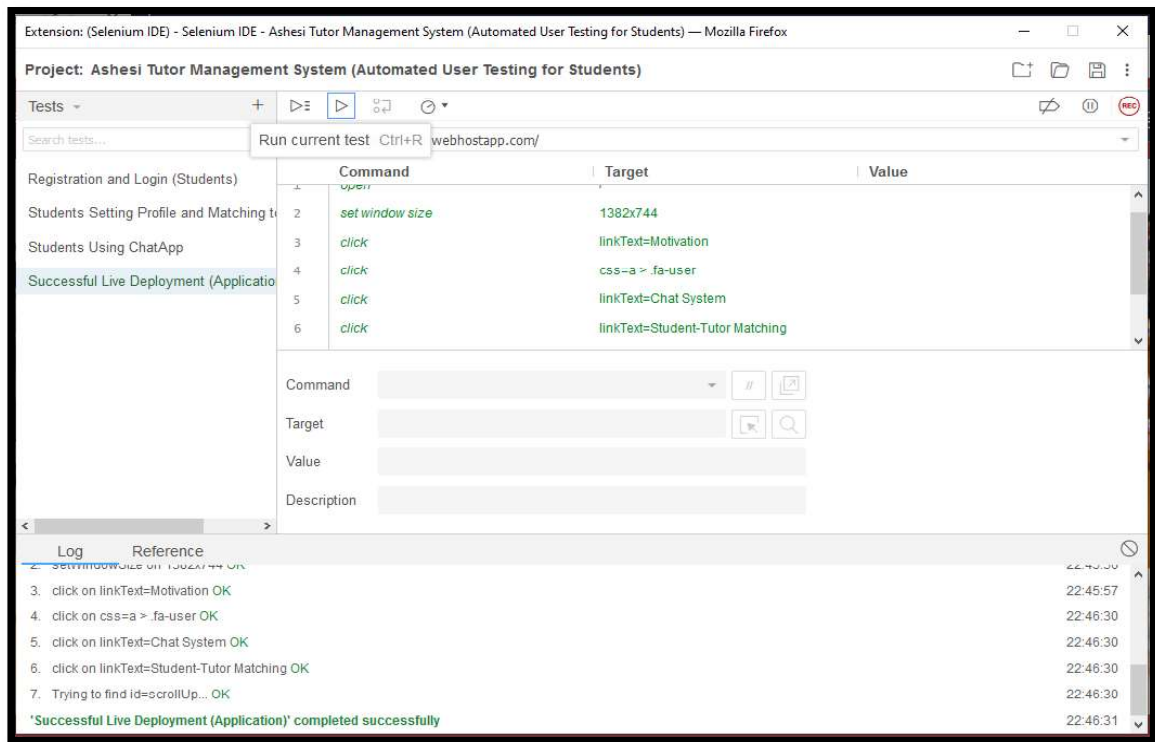


Figure 5.4.1a: Successful results of System deployment from Selenium

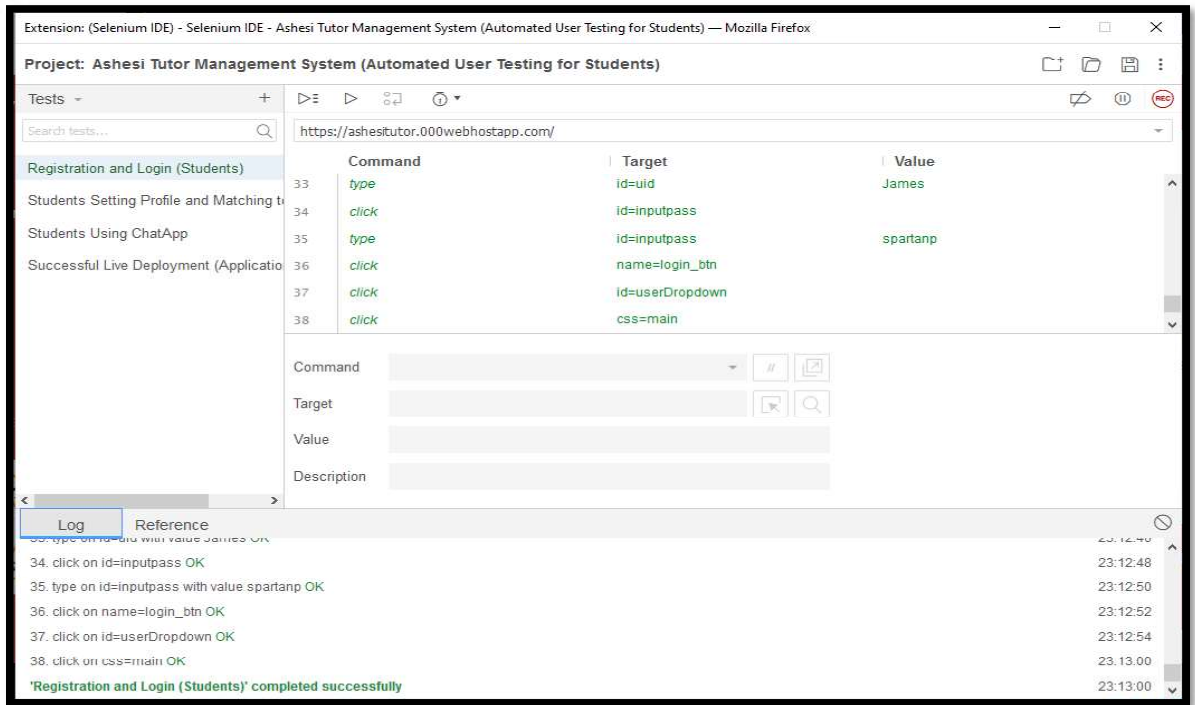


Figure 5.4.1b: Successful results of User Registration and Login (Students)

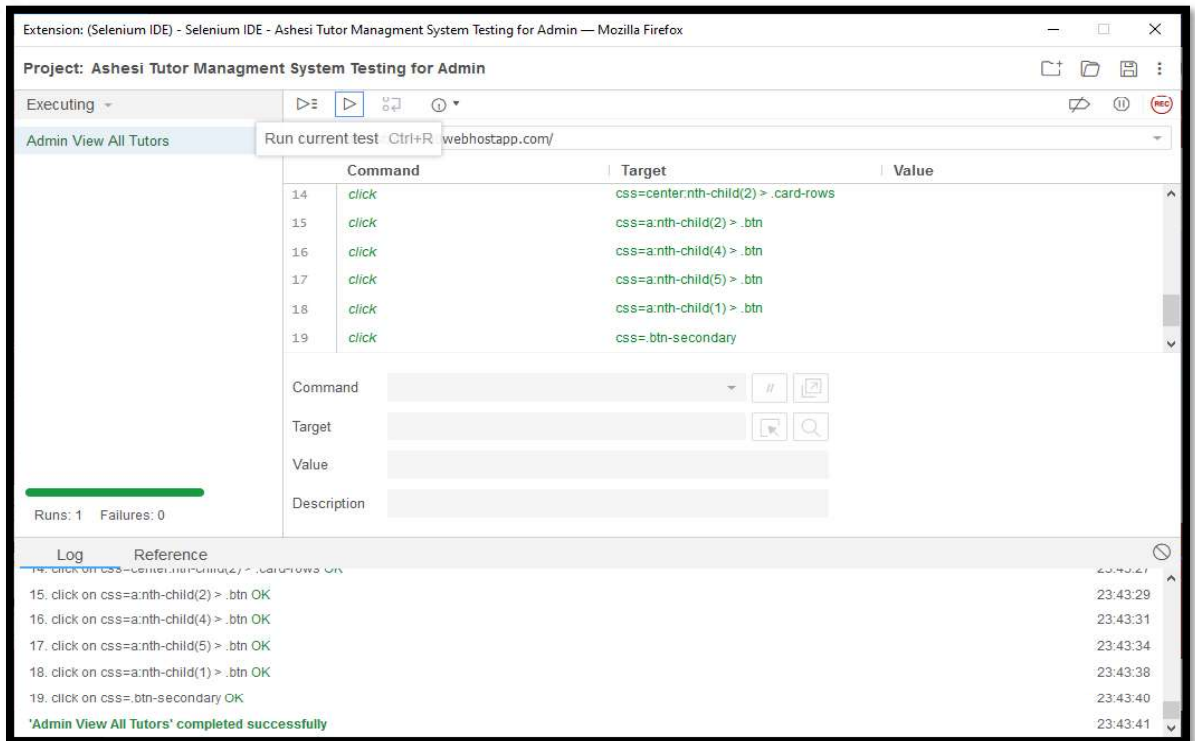


Figure 5.4.1c: Successful results of Admin viewing Tutors from Selenium

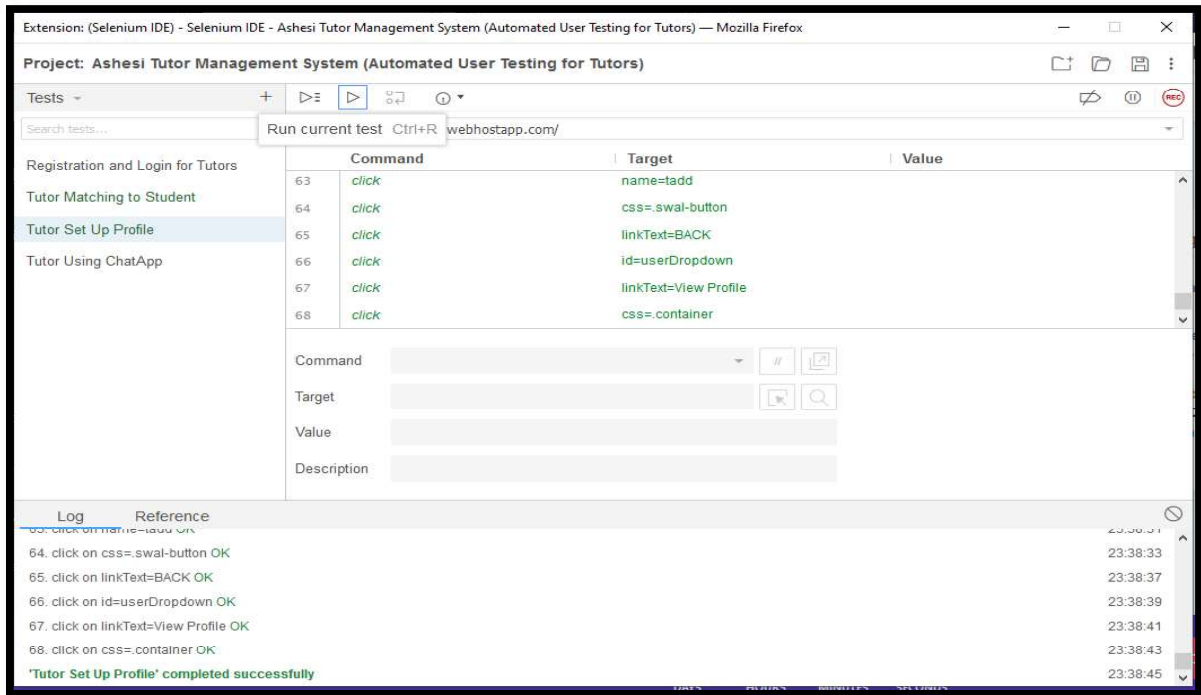


Figure 5.4.1d: Successful results of Tutors Setting Profile from Selenium

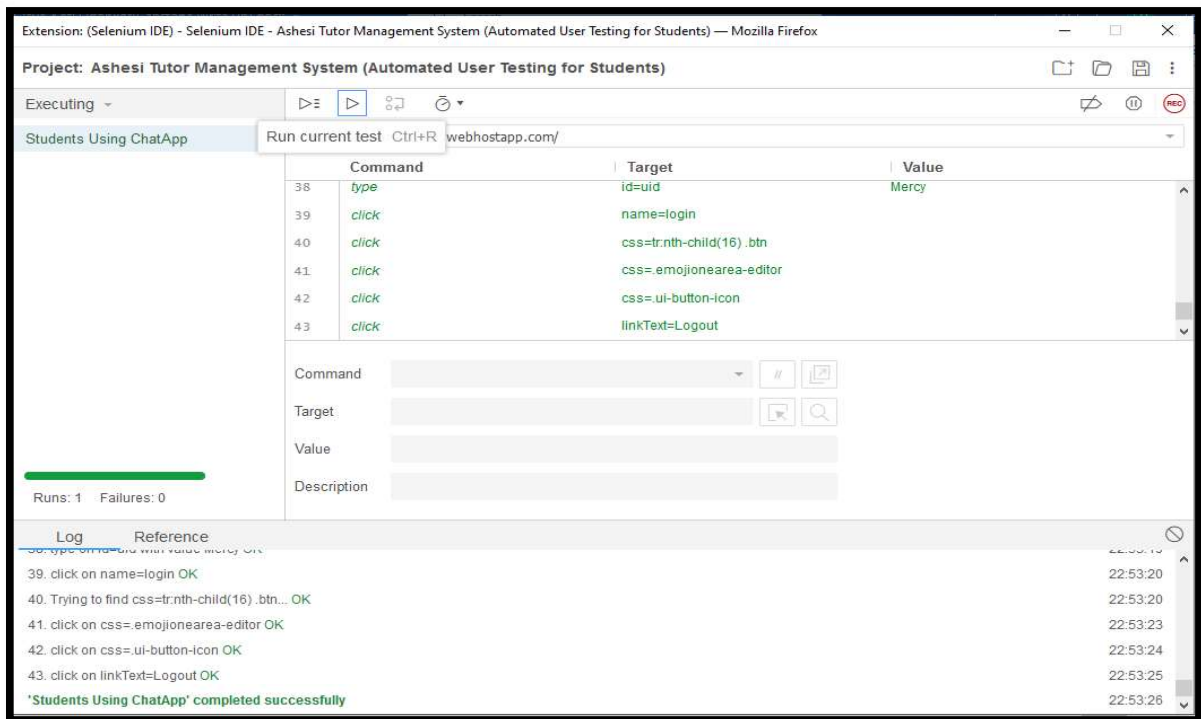


Figure 5.4.1e: Successful results of User(Student) using the Chat Application(ChatApp)

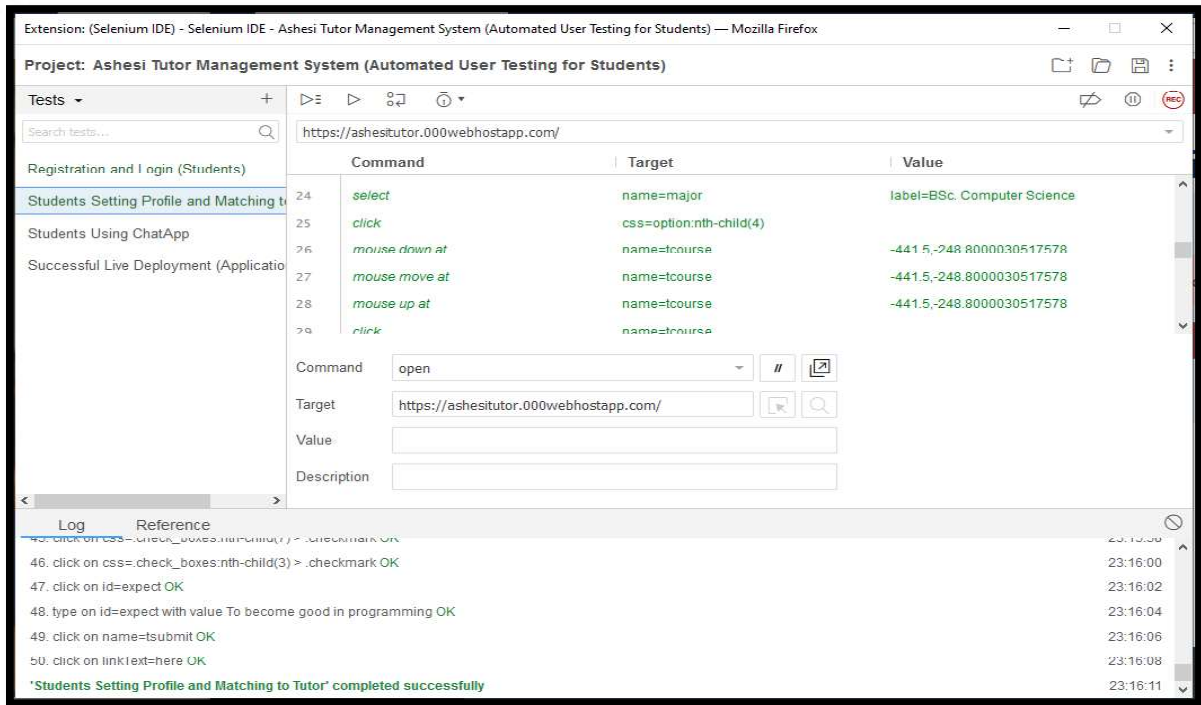


Figure 5.4.1f: Successful results for Student using Student-Tutor Matching Portal

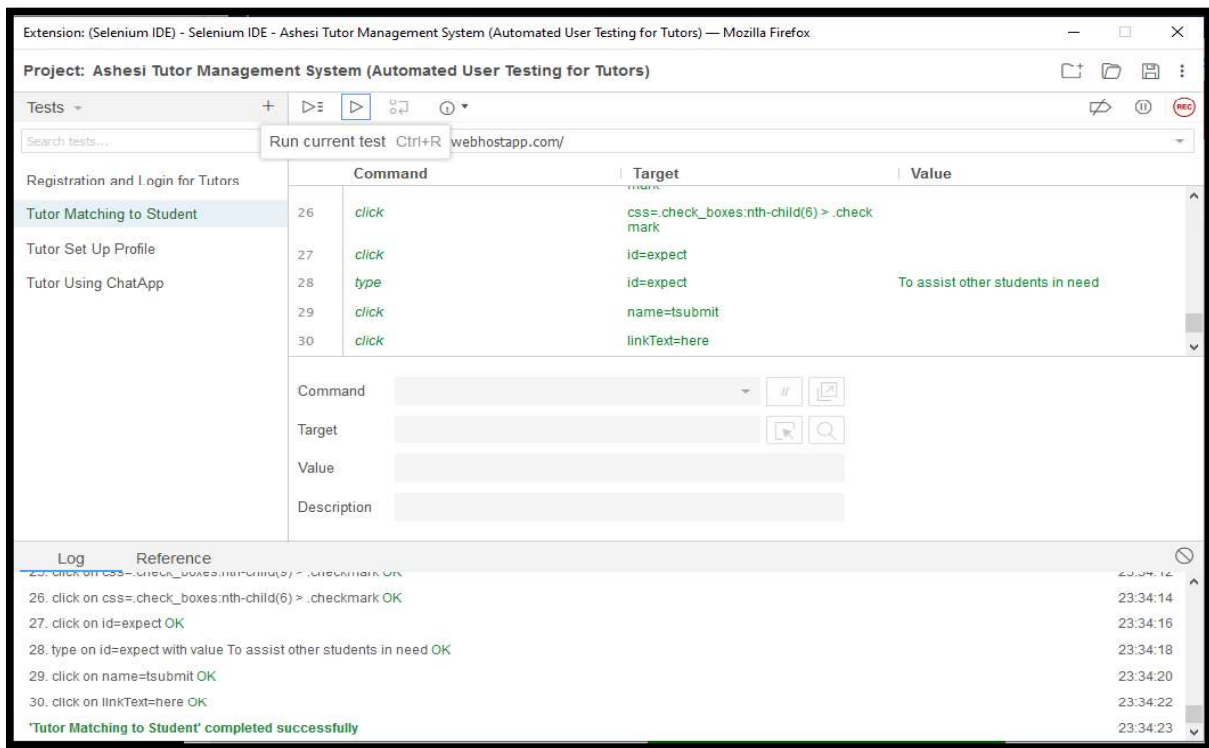
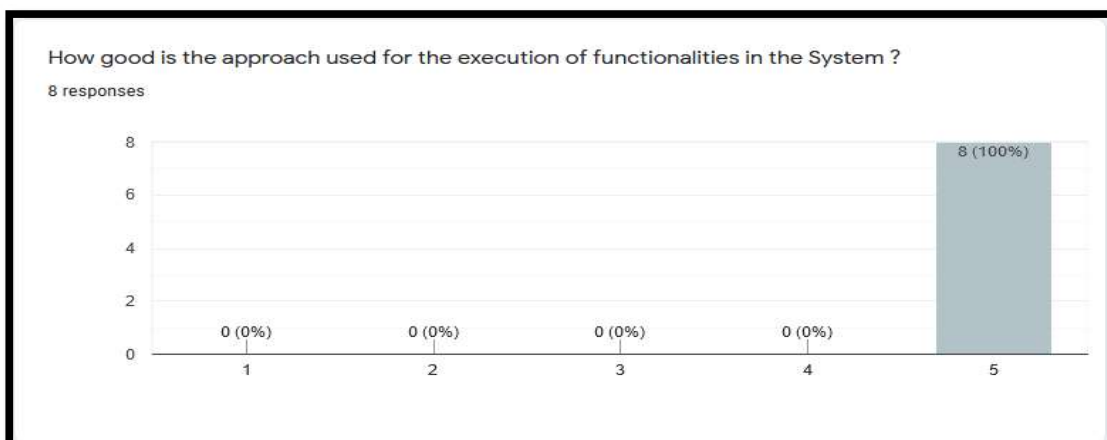
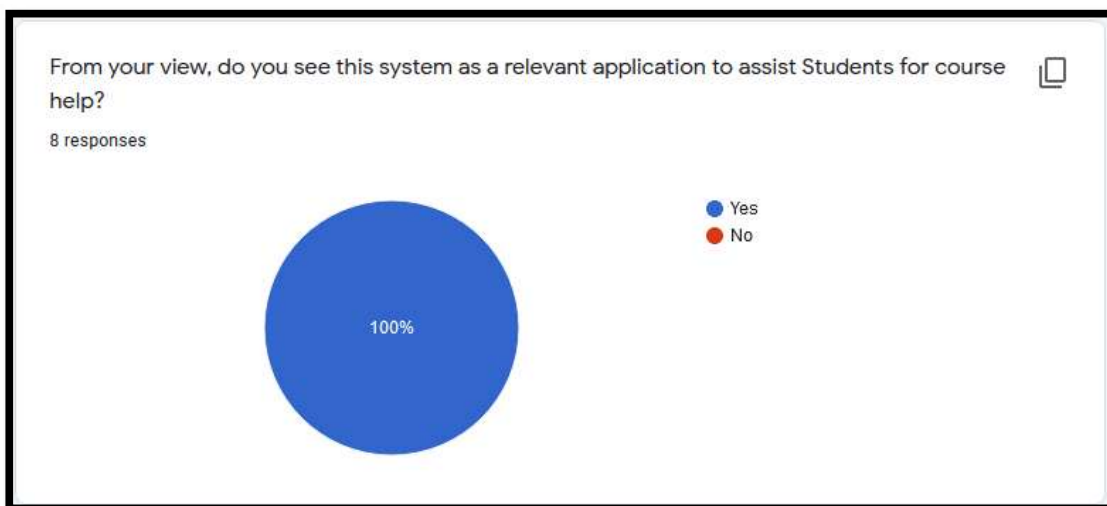


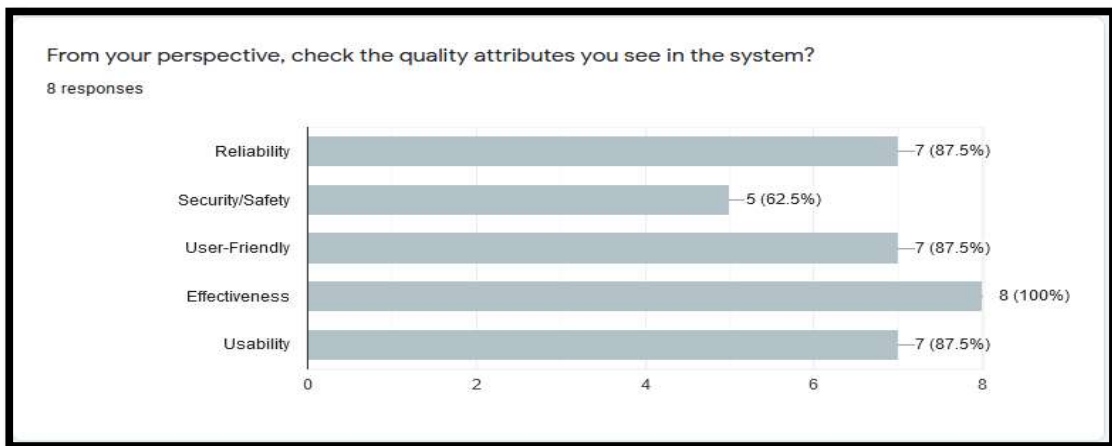
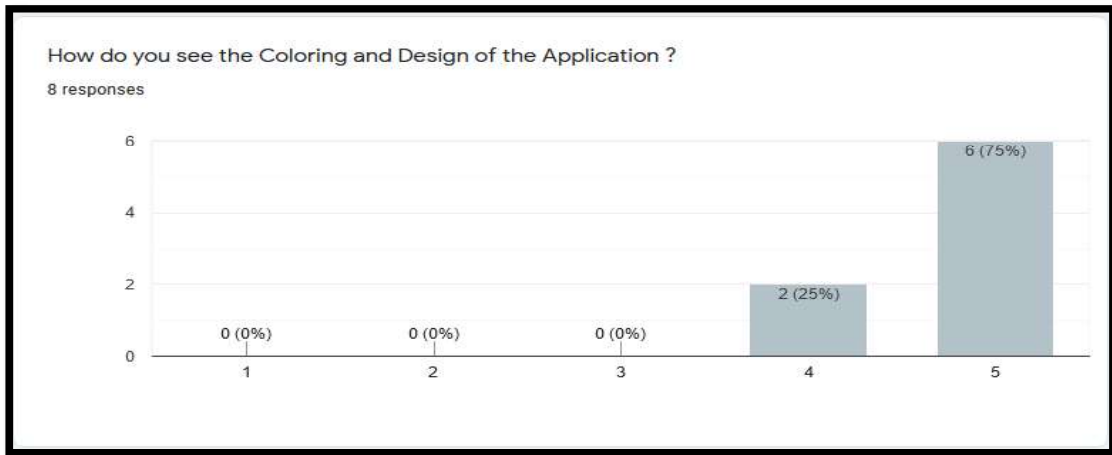
Figure 5.4.1g: Successful results of Tutor using the Student-Tutor Matching Portal and Getting Matched

5.2 User / Acceptance Testing

Development testing and the use of the Selenium tool gave an overview of the performance of the Ashesi Tutor Management System from the developer's perspective. Hence after going through the developer testing, the potential users tested the application to witness its functionalities, quality attributes, objectives, and relevance of the application. A survey was then carried out to enable Ashesi Students after going through the process of testing, to provide feedback on what they think of this application. Below are the results from users during the user testing phase.

Results from User / Acceptance Testing Survey





What are some ideas or areas of improvement you can think of ? (For future works)

7 responses

Implementation of a mobile-based version

A feature that can be added is the matching of key words in the profile of both the tutors and the students to match the right tutor to a student. The pay option can also be taken out since fees already obligate tutors to aid students in all things academics.

1. Having the unread chats (or recent) appearing on top of the page. Since a new chat is added to the database and displayed, then the display can be done in descending order such that the newly added chats appear first.
2. Once students or tutors are signed up on the platform, then notifications should easily pop up on their phones without necessarily opening the app first.
3. Perhaps an option for tutor to determine whether he or she prefers to have the session in-person or virtual (COVID has made us blend both). And if virtual, a zoom link could directly be generated using the zoom API and sent to both parties.
4. Possible sending of reminders as notifications that appear on the phones of users when meeting time or day is approaching.

What are some ideas or areas of improvement you can think of ? (For future works)

7 responses

1. Link it to the employers of the tutors(carrier service, library)
2. Include other file formats for upload.

I think the chatapp should not be a separate application from the main application in order to increase the usability of the product and make it more seamless. Also, I think that shade of green is a bit weird combined with the ashesi colours. Probably lemon green or a different colour would work better.

For future works, they could focus on a management system to assist students in reaching out to some lecturers. This is because some lecturer's receive over 100 mails in a day and it sometimes skips them. So a new system to solve this issue will be greatly appreciated.

Since it is tailored to Ashesi University Students, the level of proficiency for various courses or topics can be associated to the levels of the students (say Level 100, 200,...). This is because the use of Beginner/intermediate/advanced/expert might be vague and subjective. You can check other areas that mimic the above and use more of Ashesi specific terms.

It is a very good and useful application. Good job 🍌.

5.3 Analysis of User Testing Survey Results

From the survey, 100% of users indicated that the currently implemented system is relevant and can help students achieved their objective of getting help for courses. 100% of students also indicated that the approach used within the application in the achievement of functionalities was excellent with 75% expressing good remarks about the coloring and designing of the application. As part of testing for the functional requirements, Non-functional requirements were also defined as part of the requirements. In testing them, 100% of students indicated that indeed the system is effective whiles 87.5% of the users indicated the application is reliable, user-friendly, and usable. 62.5% of the users indicated that the application also achieves security and safety.

The results obtained above from Ashesi students generally indicate how good and remarkable the Ashesi Tutor Management System is. Ashesi students have proven that this system indeed can help to assist them when they face course difficulties. Some feedbacks obtained from Ashesi Students were incorporated within the system whiles others would serve as relevant information for further work.

5.4 Application Requirements Testing Report

The set of requirements that were documented to be implemented in the system have been tested and the results have been placed in a table together with their implementation status.

Table 5.4: Functional requirements and its implementation status and test results

Functional Requirements	Requirement Purpose	Implementation Status	Testing Results
View Tutors categorized under departments (FR1)	This allows students to easily know the various tutors available	Implemented	Testing was successful
Setting up of profile and updating them regularly (FR2)	This requirement allows tutors to set up their personality for students to know them	Implemented	Testing was successful
Rating and Evaluation of Tutor's Performance (FR3)	This requirement allows student's to evaluate the performance of tutors during the tutoring process	Implemented	Testing was successful
Student-Tutor Matching (FR4)	This requirement allows students to get matched to tutors upon request	Implemented	Testing was successful
Notification component in System (FR7)	This requirement allows users to be alerted of any information	Partially Implemented (Notification was implemented only in Chat Application)	Testing was successful only for the Chat Application

Chat Application for Tutor-Student Interaction (FR5)	Allows student and tutors to begin interaction before physical meetup	Implemented (Functionality was extended to all users)	Testing was successful
Uploading and downloading of Academic resources on the platform (FR6)	Allows tutors to upload academic resources for students to download them for review	Implemented	Testing was successful

Chapter 6: Conclusion and Recommendation

6.1 Overview of Project

The Ashesi Tutor Management System is a Software Application Portal that aims to provide university students the platform to interact with tutors concerning their areas of study they have difficulty in. Students upon logging into the system can find all tutors categorized based on departments and the year in which the course is taught. They can also view academic documents uploaded by tutors from all departments and download whichever resource they need. Students can also evaluate a tutor based on their performance regularly. The Application also allows students to use the Student-Tutor Matching Portal and request a Tutor.

Tutors on their dashboard can set up their profile, choose a course from the ones available, and upload relevant resources for review by interested students. Just like Students, they can engage in Student-Tutor Matching Portal and get matched to a Student. The Application allows the Administrator to set up courses for tutoring after receiving them from the various departments, update and delete them regularly, and view all tutors currently available. Departmental Coordinators in their respective dashboards can view tutors for their departments, approve courses for tutoring so Admin can set them, and view evaluations of tutors in their departments. The Application allows all users to engage in Chat Application (ChatApp) regularly. For Students and Tutors, the Chat Application can serve as a portal for communication before a physical meetup.

6.2 Project Limitations and Challenges

The Send Email Feature was proposed to be implemented in the application to enable students to forward messages to tutors to further initialize the tutoring process. Administrators and Departmental Coordinators on the other hand could benefit from this feature as well. To implement the Email feature, multiple email frameworks such as PHP Mailer, Zend Mail, and SwiftMailer were introduced to set up an email feature but encountered difficulties. These difficulties were with the set up of the email configurations such as ports, paths for Outlook mails, initialization of framework functions, and failure of the framework to identify existing emails for testing.

The Ashesi Tutor Management System at the early development stage was hosted on localhost hence XAMPP as the server helped with the testing of the web application. The database for the project was also stored on PhpMyAdmin locally. At some point, the XAMPP Server frequently runs into unexpected errors and as a result, the database becomes corrupted and the application fails to be executed. This resulted in the loss of the database during the early stages of development but I found a way to rectify this issue and prevented it from happening again.

Allowing the upload of multiple file formats was also a challenge in this project as tutors are limited to upload pdf documents only. Respective formats such as ".pptx" and ".docx" were considered as part of the upload formats at the tutors' disposal but multiple uploads trials using these specified file formats proved futile.

FileZilla was the FTP Client application used to establish a connection to the 000webhost platform on port 21 and to ensure the transfer of the project files. On multiple occasions, when the application was tested on the live server, some files were inactive hence upon checking the public HTML directory of the 000webhost platform, these files

were missing. The missing files then had to be uploaded individually as by doing so, they were successfully uploaded. This made the upload of missing files very challenging.

6.3 Future Works

Currently, the Ashesi Tutor Management System is a web-based application hence developing a mobile-based version stands out as one of the plans for this project. Developing a mobile-based application would help to increase the usage of the application more often as more students can download the application onto their mobile devices. The creation of the mobile-based version of this application would also make it more convenient for most users in terms of its usage.

The Establishment of the chat application serves as the main platform for effective user-to-user communication. The present features such as the emoji, notification, and online-offline status of users make the chat application very efficient. Hence, the addition of other useful features such as "Create a group feature" would be very beneficial as a tutor who is matched to a group of users can set up a group chat for these students and interact with them.

Currently, the Student-Tutor Matching works effectively using the SQL Inner Join feature and the use of comparison operations as students can be matched to their preferred tutors based on the information they provide. An upgrade into Student-Tutor Matching using Artificial Intelligence would be very beneficial as more features can be embedded in the development of the Student-Matching system.

Currently, the Ashesi Tutor Management System allows tutors to upload only documents to the system for students having difficulty in a course to download. As part of

the future works of this project, the application should allow tutors to upload videos, and set up a discussion portal on the system to facilitate student and tutor collaborations.

Lastly, the verification of users using their email before giving them the privilege to sign in to the system is a considerable feature that would help to observe security in the system. This forms part of the future works for this project because it would help to institute multiple user authentication before finally giving users the privilege to access the system.

6.4 Conclusion

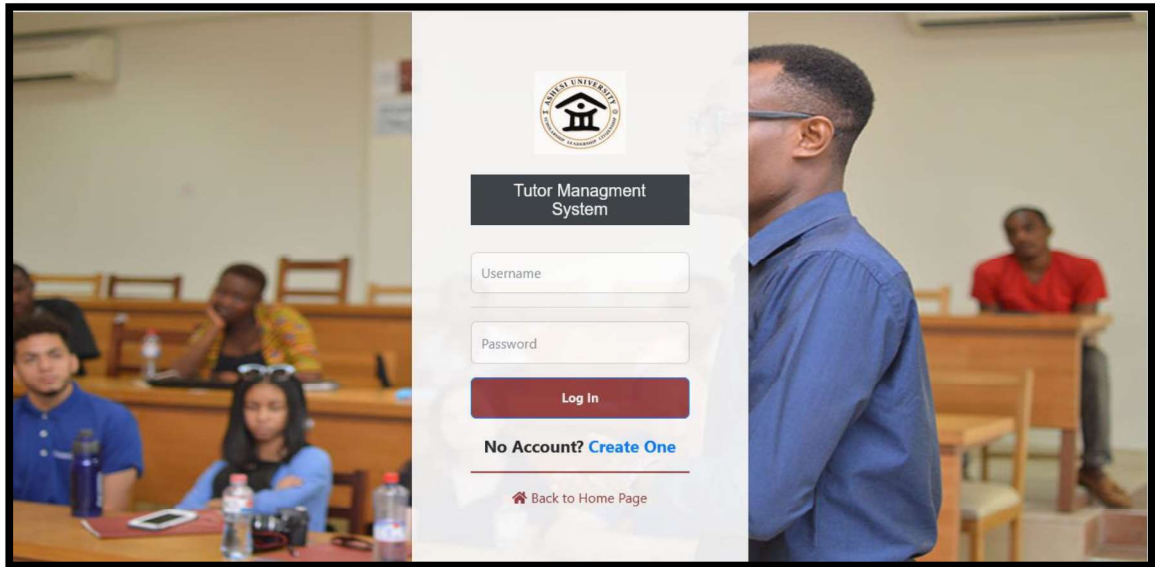
The implementation of this project shows how students can find tutors at any instance to support them concerning difficulties in understanding courses taught at Ashesi University. Faculties and faculty interns would be less overburdened with pressure from students with the provision of this application. This project has contributed to the achievement of target 4.6 under SDG4 by providing a Tutor Management System that ensures that students having difficulties in course concepts and lacking proficiency in literacy, numeracy, and functional skills in a course can have their problems sorted out by providing tutors specialized in these areas of the course. Hence, Ashesi University's aim of training ethical and entrepreneurial leaders can be achieved through the contribution of this application because it gives students a second chance to make things right in their quest for success.

References

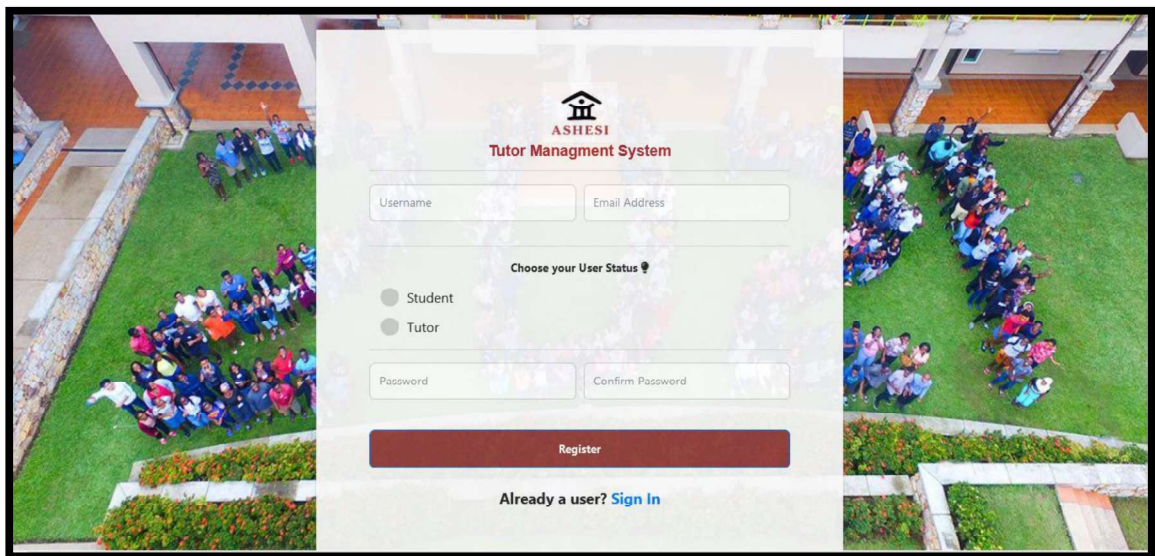
- [1] Ata Aburajab and Adnan Salman. 2019. Interactive Blackboard for Web-based Real-time Tutoring System. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, 63-68.
- [2] Bryn Jeffries, Timothy Baldwin, Marion Zalk, and Ben Taylor. 2020. Online Tutoring to Support Programming Exercises. In *Proceedings of the Twenty-Second Australasian Computing Education Conference (ACE'20)*, February 3–7, 2020, Melbourne, VIC, Australia. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3373165.3373172>
- [3] Charoenchai Wongwatkit. 2019. An Online Web-based Adaptive Tutoring System for University Exit Exam on IT Literacy. In *2019 21st International Conference on Advanced Communication Technology (ICACT)*, 563-568.
- [4] Chan Yuen Fook and Gurman Kaur Sidhu. 2015. Investigating Learning Challenges faced by Students in Higher Education. In *5th World Conference on Learning, Teaching and Educational Leadership, WCLTA 2014*, 604-612.
- [5] Dan Lo and Larry Wang. 2011. An online tutoring system with instant responses. In *Proceedings of the 49th Annual Southeast Regional Conference (ACM-SE '11)*. Association for Computing Machinery, New York, NY, USA, 316–317. DOI: <https://doi.org/10.1145/2016039.2016125>
- [6] Francis Goode. 2017. Approaches to Ghana's Higher Education Challenges Drawn from the U.S. Community College Model. In *Walden University ScholarWorks*, 1-2.
- [7] Horner and P. Gouws. 2016. E-tutoring Support for Undergraduate Students Learning Computer Programming at the University of South Africa. In *Proceedings of the Computer Science Education Research Conference 2016 (CSERC '16)*. Association for Computing Machinery, New York, NY, USA, 29–36. DOI: <https://doi.org/10.1145/2998551.2998557>
- [8] Kevin M. Guthrie. 2019. Challenges to Higher Education's Most Essential Purposes. In *2019 ITHAKA S+R*, 3-4. DOI: <https://doi.org/10.18665/sr.311221>

- [9] Muhammad Shukri Bin Safari, Muhammad Badiuzzaman Abdullah, and Abdul Rahman Ahmad Dahlan. 2018. eTutor4U Platform: Acquiring Knowledge and Skills with the Help of Great Tutors. In 2018 *International Conference on Information and Communication Technology for the Muslim World (ICT4M)*, 213-218.
- [10] Mohammad H. Hashem, Ahmed A. Al Khawaja, Saleh O. Edhah, Usman I. Hashmi and Al Hareth S. Al Akill. 2014. How Do Academic Issues affect College Students' Performance? In *ASEE2014 Zone I Conference, April 3-5, 2014, University of Bridgeport, Bridgeport, CT, USA*, 1-7.

Appendix 1



Appendix 1(a) Ashesi Tutor Management System Login Page



Appendix 1(b) Ashesi Tutor Management System Register Page