 Addis Ababa University

SEEK WISDOM, ELEVATE YOUR INTELLECT AND SERVE HUMANITY!

ADDIS ABABA UNIVERSITY

School of Information Science

Remote IT User Support and Maintenance System for AAU

Submitted in partial fulfillment of Bachelor’s Degree in  
Information Systems (Industrial Project-1)

Group Members

|  |  |  |
| --- | --- | --- |
| No. | Name | ID Number |
| 1 | Dagim Assefa | UGR/6782/12 |
| 2 | Fikir Kebede | UGR/9485/12 |
| 3 | Getnet Bantie | UGR/4623/12 |
| 4 | Leul Gezahegne | UGR/4919/12 |
| 5 | Mohammed Kedir | UGR/7600/12 |
| 6 | Senay Alemneh | UGR/1397/12 |

Submission Date: Feb 15, 2023

Examination Board

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Advisor Name Signature Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Examiner Name Signature Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Examiner Name Signature Date

Acknowledgment

First and foremost, we would like to express our sincerest gratitude to everyone who has supported us on our journey. Without your encouragement, guidance, and assistance, achieving our goal would have been much more difficult.

Your kindness and generosity have been invaluable to us. The assistance, advice, and support you provided have been instrumental in our success. We are truly grateful for all that you have done.

We owe a great deal of gratitude to our advisor Wendwessen Endale for his unfailing patience, continuous support, unreserved and valuable advises during the course of writing the whole paper. He gave us support for all the difficulties encountered starting from writing the project proposal to the finalization of this portion of the project, filled with sharp intelligence and understanding.

Finally, we would like to thank anyone who has ever believed in us and encouraged us to stay focused on our goal. Your guidance and advice have been invaluable, and we thank you for helping us stay on track. We are truly grateful for all the support we have received.

Contents

[Acronyms vi](#_Toc127222137)

[List of Tables vii](#_Toc127222138)

[List of Tables viii](#_Toc127222139)

[Chapter One 1](#_Toc127222140)

[1. Introduction 1](#_Toc127222141)

[1.1. Overview 1](#_Toc127222142)

[1.2. Background of the organization 2](#_Toc127222143)

[1.3. Statement of the problem 3](#_Toc127222144)

[1.4. Objectives of the Project 4](#_Toc127222145)

[1.4.1. General Objective 4](#_Toc127222146)

[1.4.2. Specific Objective 4](#_Toc127222147)

[1.5. Feasibility Analysis 4](#_Toc127222148)

[1.5.1. Economic Feasibility 5](#_Toc127222149)

[1.5.2. Operational Feasibility 6](#_Toc127222151)

[1.5.3. Legal and Contractual Feasibility 6](#_Toc127222157)

[1.5.4. Technical Feasibility 6](#_Toc127222158)

[1.5.5. Political feasibility 7](#_Toc127222167)

[1.5.6. Schedule Feasibility 8](#_Toc127222168)

[1.6. Significance of the Project 8](#_Toc127222170)

[1.7 Beneficiaries of the Project 8](#_Toc127222171)

[1.8 Methodology 9](#_Toc127222172)

[1.8.1. Data Collection 9](#_Toc127222173)

[1.8.2. System Development Methodology 10](#_Toc127222176)

[1.9. Development Tools and Methodology 11](#_Toc127222177)

[1.9.1 Front-end technologies 11](#_Toc127222178)

[1.9.2 Back-End Technologies 12](#_Toc127222179)

[1.9.3 Documentation and Modeling tools 13](#_Toc127222180)

[1.9.4 Deployment Environment 14](#_Toc127222181)

[1.10. Scope 14](#_Toc127222182)

[1.11. Risks, assumptions and constraints 15](#_Toc127222183)

[1.12 Phases and Deliverables of Project 15](#_Toc127222184)

[1.13. Work-breakdown structure 17](#_Toc127222185)

[1.14 Project Schedule 23](#_Toc127222186)

[Chapter Two 24](#_Toc127222187)

[2.1 Overview 24](#_Toc127222188)

[2.2 Business area analysis 24](#_Toc127222189)

[2.2.1 Activities/functions of the organization 25](#_Toc127222190)

[2.2.2 Problems of the existing system in PIECES framework 26](#_Toc127222191)

[2.2.3 Forms and reports of the current system 29](#_Toc127222192)

[2.2.4 Players of the existing system 30](#_Toc127222193)

[2.3 Requirements Definition 30](#_Toc127222194)

[2.3.1 Functional Requirement 30](#_Toc127222195)

[2.3.2 Non-Functional Requirements 32](#_Toc127222196)

[Chapter 3 34](#_Toc127222197)

[3.1 Overview 34](#_Toc127222198)

[3.2 Use Case Modeling 34](#_Toc127222199)

[3.2.1. User Interface Identification 34](#_Toc127222200)

[3.2.2 Business rules identification 37](#_Toc127222201)

[3.2.3 Actor identification 38](#_Toc127222202)

[3.2.4 Designing the use case diagram 39](#_Toc127222203)

[3.2.5. Use Case Description 40](#_Toc127222204)

[3.3 Conceptual Modeling 62](#_Toc127222205)

[3.3.1 Class diagram 62](#_Toc127222206)

[3.3.2 Class description 63](#_Toc127222207)

[3.4 Sequence Diagramming 68](#_Toc127222208)

[3.5 User Interface Prototyping 73](#_Toc127222209)

[Appendix 83](#_Toc127222211)

[Appendix 1 – Forms and report collected from organization 83](#_Toc127222212)

[Appendix 2-Interview Questionnaire used to collect information from technicians and team leaders 84](#_Toc127222213)

[Reference 88](#_Toc127222214)

## Acronyms

|  |  |
| --- | --- |
| Abbreviation | Description |
| AAU | Addis Ababa University |
| AAUNet | Addis Ababa University Network |
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| CSS | Cascading Style Sheet |
| CoBE | College of Business and Economics |
| GUI | Graphical User Interface |
| HR | Human Resource |
| HTML | Hyper Text Markup Language |
| ICT | Information Communication Technology |
| ID | Identification |
| IT | Information Technology |
| JS | Javascript |
| MVC | Model View Controller |
| MySQL | My Structured Query Language |
| NoSQL | Not only Structured Query Language |
| PIECES | Performance, Information and Data, Economics, Control and Security, Efficiency, and Service. |
| PHP | Hypertext Processor |
| SDLC | System Development Life Cycle |
| SQL | Structured Query Language |
| UC | Use Case |
| UI | User Interface |
| UML | Unified Modeling Language |
| URL | Uniform Resource Locator |
| UX | User Experience |
| WAN | Wide Area Network |

## List of Tables

[Table 1. 1 Work breakdown among team members 22](#_Toc127182084)

[Table 3. 1 User Interface identification 37](#_Toc127219525)

[Table 3. 2 Business rule identification 38](#_Toc127219526)

[Table 3. 3 Actor identification 38](#_Toc127219527)

[Table 3. 4 Landing Page Use Case Description 40](#_Toc127219528)

[Table 3. 5 AI Bot Use Case Description 41](#_Toc127219529)

[Table 3. 6 Give Suggestion to Problems to Include In the Bot Use Case Description 42](#_Toc127219530)

[Table 3. 7 Sign Up Use Case Description 43](#_Toc127219531)

[Table 3. 8 Login Use Case Description 44](#_Toc127219532)

[Table 3. 9 Forgot Password Use Case Description 45](#_Toc127219533)

[Table 3. 10 Add Team Leader Use Case Description 46](#_Toc127219534)

[Table 3. 11 View Report Generated By Team Leader Use Case Description 47](#_Toc127219535)

[Table 3. 12 Generate a Report for ICT Directorate Office 48](#_Toc127219536)

[Table 3. 13 Register technicians and provide a default password 49](#_Toc127219537)

[Table 3. 14 View Unsolved Issues Use Case Description 50](#_Toc127219538)

[Table 3. 15 Assign Support Request to Technicians Use Case Description 51](#_Toc127219539)

[Table 3. 16 View Solved Support Issues Use Case Description 52](#_Toc127219540)

[Table 3. 17 View Report Generated by Technicians Use Case Description 53](#_Toc127219541)

[Table 3. 18 View List of Out of Use Devices Use Case Description 54](#_Toc127219542)

[Table 3. 19 Generate a Report for the Admin Use Case Description 55](#_Toc127219543)

[Table 3. 20 View List of Assigned Task Use Case Description 56](#_Toc127219544)

[Table 3. 21 Contact Users via Chat Use Case Description 57](#_Toc127219545)

[Table 3. 22 Fix Issues in Person Use Case Description 57](#_Toc127219546)

[Table 3. 23 Document Problem Findings Use Case Description 58](#_Toc127219547)

[Table 3. 24 Add Out of Use Devices Use Case Description 59](#_Toc127219548)

[Table 3. 25 Train AI Chat Bot Use Case Description 60](#_Toc127219549)

[Table 3. 26 Generate a Report for the Team Leader Use Case Description 60](#_Toc127219550)

[Table 3. 27 Request for Support Use Case Description 61](#_Toc127219551)

[Table 3. 28 View if Their Request is Approved or Not Use Case Description 62](#_Toc127219552)

[Table 3. 29 Logout Use Case Description 62](#_Toc127219553)

## List of Tables

[Figure 1. 1 ICT organizational structures 3](file:///C:\Users\Lenovo\Documents\Desktop\Industrial%20project\RIMaUS%20system.docx#_Toc127221200)

[Figure 1. 2 Iterative model 11](#_Toc127221201)

[Figure 1. 3 Work breakdown structure 17](#_Toc127221202)

[Figure 1. 4 Project schedule 23](#_Toc127221203)

[Figure 2. 1Forms and reports of the current system 29](#_Toc127221213)

[Figure 3. 1 Use Case Diagram 39](#_Toc127221753)

[Figure 3. 2 Class Diagram 63](#_Toc127221754)

[Figure 3. 3 User Sequence Diagram 69](#_Toc127221755)

[Figure 3. 4 Guest Sequence Diagram 69](#_Toc127221756)

[Figure 3. 5 Technician Sequence Diagram 70](#_Toc127221757)

[Figure 3. 6 Team Leader Sequence Diagram 71](#_Toc127221758)

[Figure 3. 7 Admin Sequence Diagram 72](#_Toc127221759)

[Figure 3. 8 Technician Dashboard 73](#_Toc127221760)

[Figure 3. 9 Landing Page(Home) 73](#_Toc127221761)

[Figure 3. 10 Landing Page(About) 74](#_Toc127221762)

[Figure 3. 11 Landing Page(About) 74](#_Toc127221763)

[Figure 3. 12 Register 75](#_Toc127221764)

[Figure 3. 13 Login 75](#_Toc127221765)

[Figure 3. 14 Login Error Message 76](#_Toc127221766)

[Figure 3. 15 Verify Email Address 76](#_Toc127221767)

[Figure 3. 16 Reset Password Verify 77](#_Toc127221768)

[Figure 3. 17 Forgot Password 77](#_Toc127221769)

[Figure 3. 18 Landing Page(Services) 78](#_Toc127221770)

[Figure 3. 19 Landing Page(About) 78](#_Toc127221771)

[Figure 3. 20 Register new team leader form 79](#_Toc127221772)

[Figure 3. 21 Register new technician form 80](#_Toc127221773)

[Figure 3. 22 Request support form 80](#_Toc127221774)

[Figure 3. 23 View report 81](#_Toc127221775)

[Figure 3. 24 List of new support request 81](#_Toc127221776)

[Figure 3. 25 Generate report 82](#_Toc127221777)

[Appendix 1. 1User problem reporting form 83](#_Toc127221778)

[Appendix 2. 1 User Google form questionnaire 85](#_Toc127221798)

[Appendix 2. 2 Admin Google form questionnaire 87](#_Toc127221799)

## Chapter One

## 1. Introduction

## 1.1. Overview

In these generation, we are engaged in highly computerized technology aiming to make individuals' and organizations' lifestyles more comfortable and easier, especially in the world of business. Online remote IT user support and maintenance is now very common to widen the target market and improve transparency if anything goes wrong in the maintenance process.

This project is remote IT user support and maintenance system for Addis Ababa University, focusing on College of Business and Economics. In this system five entities will be considered. The first one is admins who control the workflow with in every campus and manages team leader. The second entity is team leader who are the head of the maintenance department in the specific campus and generally control the work within that department. The third entity is technician who provides technical support and advice for the company and all its users. Those technicians will be responsible to teach the AI bot which will be developed and provide support when the technicians are not available and if users are facing minor problems. The fourth entity of this system is users who are staff members of the campus requesting for support. The system will enable them to contact the technician by logging into their account and provide a full description of the problem and take a picture of the device that they are having an issue with so that the technician can provide the suitable solution. The fifth entity of the system is guest users who are able to view the landing page and use the AI bot without the need to register in the system. When a user requests for support the technicians will receive a real-time notification (form of request) and they will provide the support for that specific user. Users can also get help from the AI bot. Technicians will always provide a report about their job to the admin.

Replacing the current system with this automated one makes the job more efficient to the customer, as it can save time and hustle-free. In addition, the information needed by the company to decide matters in relation to purchasing a device when the device becomes out of use can be easily generated.

The rest of this chapter deals with all the main features of this software and the problems of the current manual support system that they are using. It not only describes various functions but also gives details about how these functions are related to each other and how feasible they are when compared to the current one. Apart from this, the document also contains cost estimates for developing this system. Various risks associated with the system have also been mentioned along with the ways to mitigate them.

## 1.2. Background of the organization

Addis Ababa University (AAU) was established in 1950 as the University College of Addis Ababa and is the oldest and the largest higher learning and research institution in Ethiopia. Since inception, the University has been the leading center in teaching-learning, research and community services. At present the University has 10 colleges, 4 institutes that run both teaching and research, and 6 research institutes that predominantly conduct research. Within these academic units, there are 55 departments, 12 centers, 12 schools, and 2 teaching hospitals. (Addis Ababa University, 2023)

**Information Communication Technology Directorate (ICTD)**

The ICT Directorate was established in 1996 through visionary leadership of a few individuals who realized that the AAU would be wise to join the information age by adopting the technology that has been transforming the world. The newly formed office initiated a project named AAUNet that has resulted in a wide area network (WAN) whose first phase of construction was completed in November 2001. The network, which connects all the 17 widely distributed campuses of the university, has been growing since. The services delivered through the infrastructure have also been increasing. (Addis Ababa University, 2023)

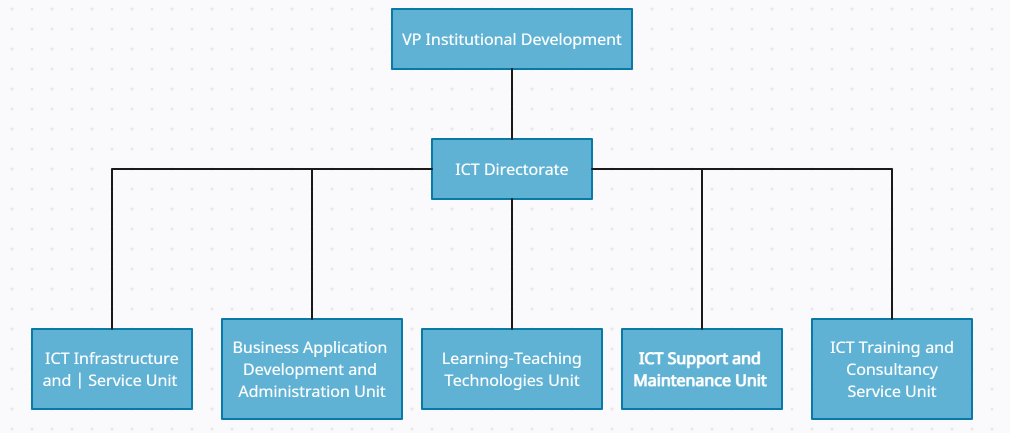


Figure 1. 1 ICT organizational structures

## 1.3. Statement of the problem

The problem faced by the university is that some of the staff members request for support while the problem is minor or negligible. When using a computer in an office, it’s known that problems can happen at any time with any of the system's components. In such situations the current system requires the users to call for support or go in person to the technicians’ office but if those support teams are not there for any reason their computer will not be fixed.

The current system has the following overall problems.

* It is difficult to trace the assigned technician for a given problem
* It is difficult to identify solved, unsolved or escalated problems
* It is difficult to identify work load of technicians
* It is difficult to generate consolidated reports regarding previously recorded problems, their status and findings, performance of technicians and others.
* There is no knowledge repository that guides users to troubleshoot minor problems by their own.

The problems that are stated above are faced by all campuses of the university but we are focusing on CoBE, but this can be modified to be used in all the campuses with a minimal change.

## 1.4. Objectives of the Project

## 1.4.1. General Objective

The general objective of this project is to develop a system that automates the process of maintenance and user IT support within Addis Ababa University, College of Business and Economics campus starting from requesting for support to generating a report.

## 1.4.2. Specific Objective

The specific objectives of this project are to develop a system that:-

* Automates report generation of IT user support for analyzing performance
* Request management
* Integrates an AI bot that is able to learn and support users with some of their problems and provide self-service online
* Centralizes all service requests by allowing tasks to be assigned only by the team leader

## 1.5. Feasibility Analysis

This feasibility analysis will be carried out in order to determine, if the company possess the required resources or technologies; as well as whether, the proposed system has a reasonable return compared to the risk of the investment. A feasibility study contains a detailed analysis of what's needed to complete the proposed project. The report may include a description of the new product or venture, a market analysis, the technology and labor needed, as well as the sources of financing and capital. The report will also include financial projections, the likelihood of success, and ultimately, a go-or-no-go decision. (Investopedia, 2023)

## 1.5.1. Economic Feasibility

# The development and implementation of the proposed system, which is to develop a system that will straighten up the communication between the technical assistants and users and make maintenance faster and easier for the users and technicians, will require development cost but once deployed operational cost is minimal and usage of additional trivial communication materials will be cut down.

**Cost Analysis**

Since most users have got up-to-date desktops, the proposed system does not require any additional hardware. The proposed system will be developed using HTML, Tailwind CSS, Java Script, Bootstrap, and CSS3 as a front end and MySQL, Laravel, EmailJS API, and Firebase as a back end. And all these software are distributed freely.

**Benefits**

1. Tangible Benefits
   * Cost reduction and/or avoidance
   * It eliminate the duplication of requests
   * Increase flexibility
   * Improvement in management planning and control
   * Job transparency
   * Change work culture
2. Intangible Benefits
   * Better service provision
   * Faster decision making
   * Timely and accurate information
   * Builds employee moral
   * Create trustworthy service provision

The new system provides both tangible and intangible benefits to the office. These benefits are necessary and desirable.

## 1.5.2. Operational Feasibility

# The proposed system will enable users to communicate directly with technical assistants without the need to address the issue with a physical presence or make a busy phone call with a primary intention of cutting down maintenance response time.

# The current working system where the users report their issues directly in person, via a phone call or make a report for item replacement has a major issue when it comes to response time, even after reporting the users have to wait for the confirmation of their support requests. This system aims to resolve these issues.

# Based on certain feasibility assessments, we expect the relevant stakeholders to accept, use, and support the system given the reason that the system conforms to the following points:

# The system utilizes virtually all available resources, time being the most important of those.

# The system will not need much from both the technicians and the users as it will be an easy to use that will only require a few trials to get used to.

## 1.5.3. Legal and Contractual Feasibility

The proposed system will be secured and will have restricted access by only allowing authenticated users access to the system, mostly enforced by the admin. No user of the system will be asked to enter personal information; the only requirement is to enter their email and password to access the system.

## 1.5.4. Technical Feasibility

# The study is to measure the predictability of the technical solution and whether the organization possesses the necessary technology to solve the problem as projected. It also measures whether there is enough technical expertise in the organization so as to support the proposed system.

# Even though the office has practiced a manual system before the office has the required equipment, the office has to do some minor maintenance and debugging.

# The devices that will be used in order to access the system are available with the office and the software that will be developed is reliable enough to avoid any problems in the future. The system will be extendible and be able to handle the projected growth of the organization in the future. The system development will not require any high-end technology to develop, maintain and support.

# The technical feasibility analysis is accepted based on the following points:

# The proposed system is practical for both the users and technical teams

# The team members all contributing their skill sets will contribute for the manpower needed to develop the system including designing, coding and testing.

# All the technical resources to use the system could be easily available, if not; any verified user with a digital device with a browser installed can access the system to carry out the needed task.

## The technology used to develop the system is carefully selected to meet the requirements of the users from response time to security.

## 1.5.5. Political feasibility

This system automates and digitalizes the technical support system in Addis Ababa University. It aligns with the national strategy to digitalize major systems in the country. The system will not implicate any political conflict of interest or concern among any groups. This project will not harm anyone; instead, it will provide an interactive service for help.

## 1.5.6. Schedule Feasibility

## The time-consuming tasks in the development of the project are going to be data collection, the system design, understanding the workings of the current system, collecting the requirements of the users and taking most of the implementation period is developing the actual system that meets the requirements of the users.

After the successful development of the system, the subsequent task will be testing it in different circumstances to identify if there is any bug in our program before it gets into an implementation phase. An approximate time table that is required for developing remote support and maintenance system is displayed at the end of the chapter.

## 1.6. Significance of the Project

The significance of the project is to improve the operational efficiency of technicians in respect to the handling of data, elimination of time-consuming bureaucracy staff faces when electronic equipment in the office malfunctions.

Staff in the universities, technicians, and the IT department as a whole would benefit immensely when the new system is implemented, in the sense that:

* Staff would get services in a short period of time
* Technicians would also experience minimal efforts in maintaining equipment, as they would only need to access the problem details through the system.
* The admin can easily carry out an assessment to know the total number of staff satisfied by the services given through the system.
* Increased efficiency.

## 1.7 Beneficiaries of the Project

1. **University Staff:** Remote IT maintenance and user support system works all the time.

* Get faster services: This system will reduce the time required to get items maintained. In the manual system technicians would have to go to the place where the item is found. As such, it takes longer time to get the services given by the technicians. Now the staffs get services by using the system for minor functional issues.

1. **Technicians:** Remote IT maintenance and user support system reduces workloads for technicians and optimize customer service as they can now give efficient support online for their customers for less critical issues.
2. **The Guest users:** The guest includes anyone whose email address domain isn’t of the university (@aau.edu.et). Guests visit the website to get some information of the system but also use the AI bot to get support on minor issues.

## 1.8 Methodology

Various techniques and methods have been used to achieve the objective of the project. This techniques and methods are selected for their effectiveness in analyzing, designing and implementation of a system.

## 1.8.1. Data Collection

## We used the following methods during data collection: Document review, Interview and Questionnaires as our major data collection methods.

## Through this we collected raw data for the System, where existing reports on the current system are available. Verbal interview techniques were also used to interview staff from Addis Ababa University.

1. **Analyzing form, file and document**

We used this method to analyze the system requirement. This includes sample business form and report, Organizational chart and documentation of Existing system.

We have mainly used the user report problem reporting form (figure 5) in order to represent most of the forms implemented in the system. Analyzing the organizational structure mainly obtained from the Addis Ababa University website aided to understand the hierarchical structure of the University concerning ICT and maintenance office.

1. **Interviewing**

In this method, we contacted the team leader in the College of Business and Economics. Interviews were conducted to find out what difficulties they have encountered with the existing maintenance and support system.

Interviewing the team leader we were able to get a gist of the maintenance and user support process and flow in the university, we were able to identify who the entities involved in the process were and how they were involved.

1. **Questionnaire**

Questionnaire was issued to the technicians and team leader of College of Business and Economics; this enabled us to address separately the technicians and the team leader and understand their perspective separately.

## 1.8.2. System Development Methodology

The **systems development life cycle (SDLC)** is a common methodology for systems development in many organizations that marks the phases or steps of information systems development. The SDLC methodology that we have used is iterative model.

In the Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed.

We don’t attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed to identify further requirements. This process is then repeated, producing a new version of the software at the end of the iteration of the model.

**Iterative Model – Design**

We will start with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. In each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).

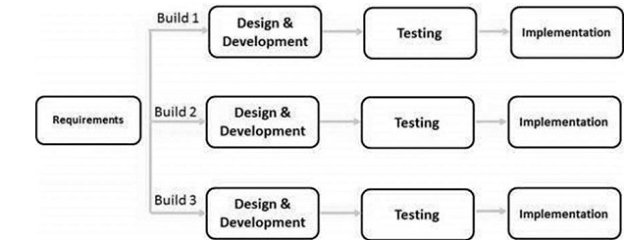


Figure 1. 2 Iterative model

In this incremental model, we divide the whole requirement into various builds. In each iteration, the development modules go through the requirements, design, implementation and testing phases. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is ready as per the requirement.

The key that makes us to use an iterative software development lifecycle is rigorous validation of requirements, and verification & testing of each version of the software against those requirements within each cycle of the model. As the software evolves through successive cycles, tests must be repeated and extended to verify each version of the software.

## 1.9. Development Tools and Methodology

Development Tools are tools that assist in the creation, modification and/or test application programs. Some examples include compilers, debuggers, visual programming methods, GUI designers, and framework generators.

## 1.9.1 Front-end technologies

Front-end development manages everything that users visually see first in their browser or application. Front-end developers are responsible for the look and feel of the site. The front-end development focuses on the client-side of the development, meaning it focuses on what the users visually see first in their browser or application and interact with them. Front end languages include HTML, CSS, and JavaScript.

* HTML (Hypertext Markup Language)

HTML is the most basic building block of the Web. It defines the meaning and structure of web content. (w3schools, 2023)

* CSS (Cascading Style Sheets) and Bootstrap

CSS describes how HTML elements are to be displayed on the screen, on paper, or in other media. It controls the layout of multiple web pages all at once. (w3schools, 2023)

Bootstrap is the most popular CSS framework for developing responsive and mobile-first websites. This project uses Bootstrap 4 which supports all major browsers except the Internet Explorer.

* JavaScript

JavaScript is a scripting language that enables you to create dynamically update content, control multimedia, animate images, and pretty much everything else, it can update both HTML and CSS, it calculates, manipulates, and validates data. (w3schools, 2023)

* Tailwind CSS

Tailwind CSS can be used to style websites in the fastest and easiest way. Tailwind CSS is basically a utility-first CSS framework for rapidly building custom user interfaces. It is a highly customizable, low-level CSS framework that gives you all of the building blocks you need to build bespoke designs without any annoying opinionated styles you have to fight to override.

## 1.9.2 Back-End Technologies

Back-end development refers to the server-side of an application and everything that communicates between the database and the browser. This type of web development usually consists of three parts: a server, an application, and a database. Code written by back-end developers is what communicates the database information to the browser.

For the project we have chosen Laravel, a PHP framework for our server-side scripting.

* **Laravel**

The below-mentioned points are more agreeable for this project, hence the chosen framework for this study.

* + The MVC model it implements eases our work and gives an easily manageable structure.
  + Laravel makes implementing authentication very simple.
  + It allows for easier mail services integration.
  + Laravel helps to secure the web application by protecting it against the most serious security risks: SQL injection, cross-site request forgery, and cross-site scripting.
  + Laravel is built with testing in mind.
  + All Laravel routes are defined in the app/Http/routes.php file, which is automatically loaded by the framework.
  + The Laravel queue service provides a unified API across a variety of different queue backend.
* **MySQL**

MySQL is an open-source relational database management system. As with other relational databases, MySQL stores data in tables made up of rows and columns. Users can define, manipulate, control, and query data using Structured Query Language, more commonly known as SQL.

* **Firebase**

The Firebase real-time Database is a cloud-hosted NoSQL database that allows to store and sync data between users in real-time.

We have chosen to use firebase to store data on the cloud that will be for the AI bot, and the AI bot will be developed using javascript, html and CSS and it will hosted using netlify.

## 1.9.3 Documentation and Modeling tools

This project will use the following modeling tools:

* **Google Docs:** We chose Google docs because it allowed us to seamlessly work on the same projects at the same time while also being a free tool.
* **Microsoft** **Word:** Microsoft word was used for document creation and sharing among group members.
* **Enterprise architect:** Sparx Systems Enterprise Architect is a visual modeling and design tool. The platform supports the design and construction of software systems, modeling business processes, and modeling industry-based domains. We have chosen to use this design tool because it encompasses different modeling techniques that able to create attractive and editable models.
* **Adobe:** Is a UI (User Interface) and UX (User Experience) design tool and it can be used to design anything from smart watch apps to fully fledged websites. Our project will use this to design the UI for more upscale human computer interaction.

## 1.9.4 Deployment Environment

This is the environment where we develop the product locally on the computer and then each of us will create a branch on our git repository and push each change to our own branch. After finishing each task, we’re going to merge it after a considerable review. The database for the AI bot will be hosted using firebase and the whole website will be hosted on Netlify (which is a remote-first cloud computing company that offers a development platform that includes build, deploy, and server less backend services for web applications and dynamic websites.)

## 1.10. Scope

As the project's name suggests, our project's working area is the development of remote IT user support and maintenance system, with a particular emphasis on creating a web-based version of the system for university users and technicians of CoBE campus. The following points define the project's boundaries:

* The system allows the Admin to control the job of the technician and makes sure that the user is satisfied by the support that is provided to them.
* The system allows the Technician to provide support for users.
* The system allows Users to request for support.

## 1.11. Risks, assumptions and constraints

**Risks of the project**

Risks associated with the development and implementation of the project:-

* The time taken for system development and implementation may take longer than expected.
* The users and technicians may fail to accept and use the system.

**Constraints of the project**

These are constraints associated with the development and implementation of the project. The project is expected to meet some organizational and other aspects of constraints, some of them being:

* Meeting the deadline for the project, delivering the project and all its requirements on time.
* Delivering all the promised functionalities with additional features with added quality.

**Assumption of the project**

* The development of the project will be met with all the required resources.
* All relevant stakeholders will be in favor of the development and deployment of the project.
* The team members of the group will all see through the development and implementation of the project providing all the required skills.
* All development equipment is in and will remain in good condition.

## 1.12 Phases and Deliverables of Project

**Phases of the project**

**I.  System Planning**

Problem identification is the first step in system planning phase, a need for the remote IT user support and maintenance system, and also plan how to develop the functional requirements of a system. The deliverable of this phase is determining all the user and system requirements needed to develop the system.

**II. System Analysis**

In this step, we will assess the state of the campus’ technical support infrastructure and existing office property inquiries. At this point, additional sources of information regarding the new requirements and the current working habits would also be looked into. The deliverable of this phase is studying the cost and schedule feasibility for the project.

**III. System Design**

After the requirements have been determined, the necessary specifications for the hardware, software, people, and data resources, and the information products that will satisfy the functional requirements of the proposed system can be determined. The deliverable of this project is identifying and creating the design and interface of the system.

**IV. Implementation**

  The actions that start as soon as the system design is finished are referred to as the implementation phase. These stages produce software code in accordance with the system design, analysis, and planning stages. The process of developing the finished system involves coding and debugging. The deliverable of this project is coding, developing and testing the final output of the project, the system.

## 1.13. Work-breakdown structure

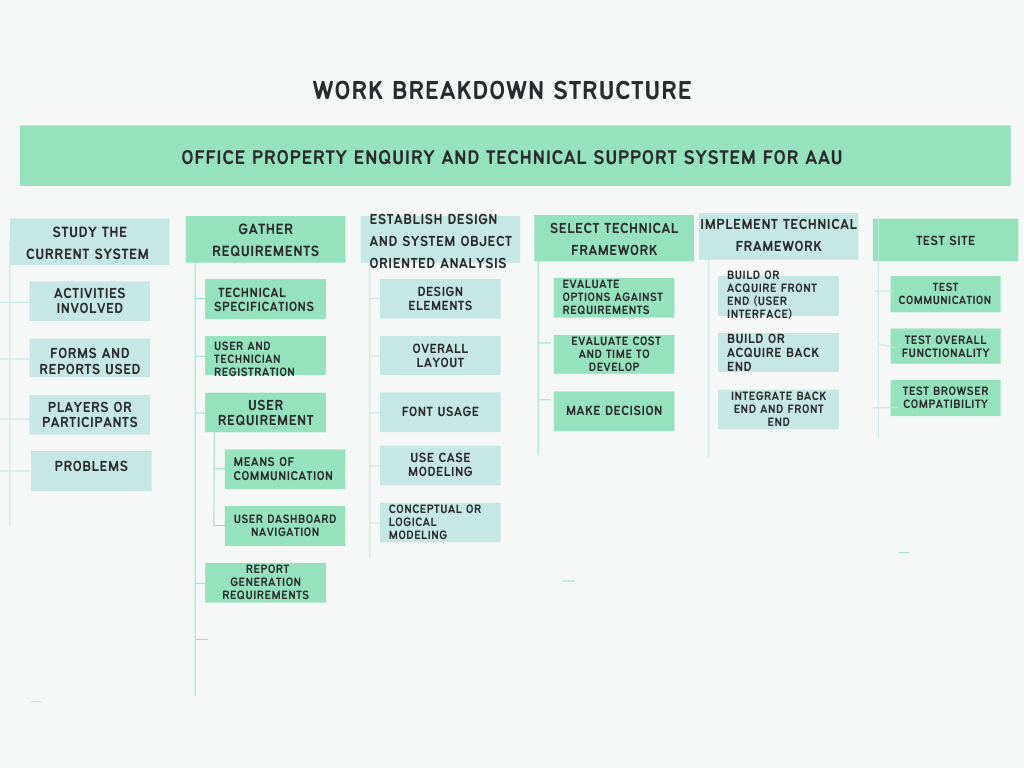


Figure 1. 3 Work breakdown structure

|  |  |  |
| --- | --- | --- |
| **No** | **Work break down** | **Responsible Person** |
| 1. | Chapter one - project proposal | |
| 1.1 | Overview | Dagim Assefa |
| 1.2 | Background of Organization and Study | Mohammed Kedir |
| 1.3 | Statement of the Problem | Dagim Assefa |
| 1.4 | Objectives of the project   1. General objective 2. Specific objectives | Dagim Assefa |
| 1.5 | Feasibility Analysis   1. Economic Feasibility 2. Operational Feasibility 3. Schedule Feasibility 4. Legal and Contractual Feasibility 5. Political Feasibility 6. Social Feasibility 7. Technical Feasibility | Fikir Kebede |
| 1.6 | Significance of the Project | Leul Gezahegn |
| 1.7 | Beneficiary of the project | Getnet Bantie |
| 1.8 | Methodology   1. Data collection 2. System Collection Method 3. System Development Methodology | Getnet Bantie |
| 1.9 | Development Tools and Technologies   1. Frontend Technologies 2. Backend Technologies 3. Documentation and Modeling Tools 4. Development environment 5. Deployment Environment 6. Project management tools | Dagim Assefa |
| 1.10 | Scope and Limitation | Senay Alemneh |
| 1.11 | Risks, assumptions and constraints   1. Risks 2. Assumptions 3. Constraints | Fikir Kebede |
| 1.12 | Phases and Deliverables of the project | Senay Alemneh |
| 1.13 | Work Breakdown Structure | Leul Gezahegn |
| 1.14 | Project Scheduling | Leul Gezahegn |
| 2. | Chapter Two-Business Area Analysis and Requirement Definition | |
| 2.1 | Over view | Dagim Assefa |
| 2.2 | Business area analysis   1. Activities/functions of the organization 2. Problems of the current system 3. Forms and Reports of the current system | Senay Alemneh, Mohammed Kedir and Leul Gezahegn |

|  |  |  |
| --- | --- | --- |
|  | 2.2.4. Players of the existing system |  |
| 2.3 | Requirements Definition   1. Functional Requirements 2. Non-functional requirement | Fikir Kebede, Getnet Bantie, and Dagim Assefa |
| 3. | Chapter Three-Object Oriented Analysis | |
| 3.1. | Over view | Dagim Assefa |
| 3.2. | System use case   1. UI identification 2. Business rule identification 3. Actor identification 4. Designing the use case diagram 5. Use case Description | Mohammed Kedir, Fikir Kebede, Getnet Bantie, Leul Gezahegn and Senay Alemneh |
| 3.3 | Conceptual Modeling   1. Class diagram 2. Class description | Leul Gezahegn and Mohammed Kedir |
| 3.4 | Sequence Diagramming | Getnet Bantie |
| 3.5 | User interface prototyping | Senay Alemneh, Fikir Kebede and Dagim Assefa |

Table 1. 1 Work breakdown among team members

## 1.14 Project Schedule




Figure 1. 4 Project schedule

## Chapter Two

## 2.1 Overview

The operations and roles that each player performs inside the organization are shown in this chapter, along with how they are carried out. It provides a broad explanation of what happens when a campus employee needs assistance and how each technician responds. The chapter also includes a breakdown of all the issues that the existing system has and how they impact the campus's overall working environment.

This chapter also explains the players who have a contribution when executing maintenance support and related tasks. The current system has two major participants: staff members who request assistance (also known as users) and maintenance providers (also known as technicians).

The function that each participant performs is also explained in this chapter. Finally, this chapter will cover all the requirements of the new system. This system will remove all the barriers that are faced by the current system, and it will make the technicians more accountable to their job. Detailed functions and processes involved in the current system will also be provided in the following content.

The assessment was made based upon

* Interviews of user, technicians, team leader and ICT Support and Maintenance Unit leader,
* A review of supporting document provided by the office

## 2.2 Business area analysis

Business analysis is a combination of gaining insight from data using specific techniques, and performing tasks to identify the needs of a business—then, recommending changes and providing solutions that produce value for the stakeholders. It includes analyzing a business or organization, by documenting its systems and processes, assessing its business model, identifying vulnerabilities, and devising solutions. It evaluates how well business processes, business rules, business data, organizational structure, and people skills match the mission of the organizational unit. (Simplilearn, 2023)

## 2.2.1 Activities/functions of the organization

The University recognizes ICT as a powerful enabling tool for institutional change and transformation to achieve its strategic objectives, missions and vision.

The ICT support and maintenance unit provides the primary management and support for computing and technology infrastructures, services and systems within the University. The office also manages the data network infrastructures, central services provision such as official website, institutional emails, video conferencing systems, total antivirus security systems, internet and data services, bandwidth optimization services and business applications. The office also provides troubleshooting, maintenance and technical support services on computers, office machines and related devices.

The unit has the following activities that are listed below:

1. Fix the issues of the staff in the university related to their equipment.

* Users submit their request for support to the office in one of the following three ways.
  1. Using Users Problem Report Form.

The offices prepared and distributed user problem report form to all faculties and department administration offices and, users collect and fill-out this form and submit to local ICT support and maintenance unit. Users are required to fill-out their name, faculty, department, email, floor or office number, telephone, date and time along with problem type, description and severity level.

1. Using Telephone.

The office allocates a direct telephone line for users in order to report their issues to the team leader via a phone call.

1. In person.

Sometimes users visit the office in person and report their issues directly to the available technicians and request for support.

1. Assign technician for reported problems:

After the report has been submitted to the office, the problem is usually reported using users’ problem request form or using telephone then the technician will be assigned by the team leader.

1. Register Devices:

Devices which can’t be fixed with in the campus should be registered as out of use. Equipment check-in and check-out form is filled in order to control whose equipment is received for maintenance. In the form name of equipment, item number, received date issue and name and signature of the technician are filled.

1. Report generation:

Reports collected from technicians are presented in a form of paper and that include:

* Number of problems reported
* Number of problems solved
* Number of unsolved problems
* Number of devices that are out of use
* Status of reported problem
* Types of problems
* Time taken to solve a given problem

## 2.2.2 Problems of the existing system in PIECES framework

Based on the staff interviews and analysis of the working document the following problems are identified that hiders the unit not to provide its service to the university’s community. We classify and present the problems in PIECES framework.

1. Performance problems

* If users need to get service from the office, he/she must fill the form and present to the office. This kind of scenario takes much time because the user has to go to the office to submit the form.
* The nature of the work forces the technician to move from one office to the other. In this way it is also taking much time to respond for reported problem.

1. Information problems

* Sometimes users and technicians don’t fill the user problems report form properly and this led to:
* Difficulty to locate user’s office.
* Difficulty to understand the problem type.
* Difficulty to prepare a report based on reported problems.
* Difficulty to trace whether a problem was solved or unsolved
* It is difficult to identify work load of technicians.
* The manager doesn’t have well formatted record keeping mechanism to keep a record of who is assigned for which problem and who is free and this led the manager to assign technicians randomly.
* There are limitations of information to evaluate technicians’ performance.

1. Economic problems

The unit has been providing a service to a large number of users and all requests for help should be presented using the user problems report form. This leads to

* High paper consumption per day which means the office incurs expense.
* Users who cannot get the user problems report form uses telephone to present their request and this also incur telephone bill.

1. Efficiency issues

* User problems report form are not properly shelved or kept, which has resulted in difficulties to compile a report.
* The user problems report form does not have unique case number. Thus, there is no way to find a specific user problems report form.
* If the user problems report form is completely lost, there is no way to trace whether:
* The specified problem was reported.
* The assigned technician was involved or not.
* The problem was solved, unsolved or escalated.

1. Service issues

The team leader writes summary report by reading each user problems report form one by one. The existing user problems report form is not complete enough to accommodate the details of all problem types.

## 2.2.3 Forms and reports of the current system

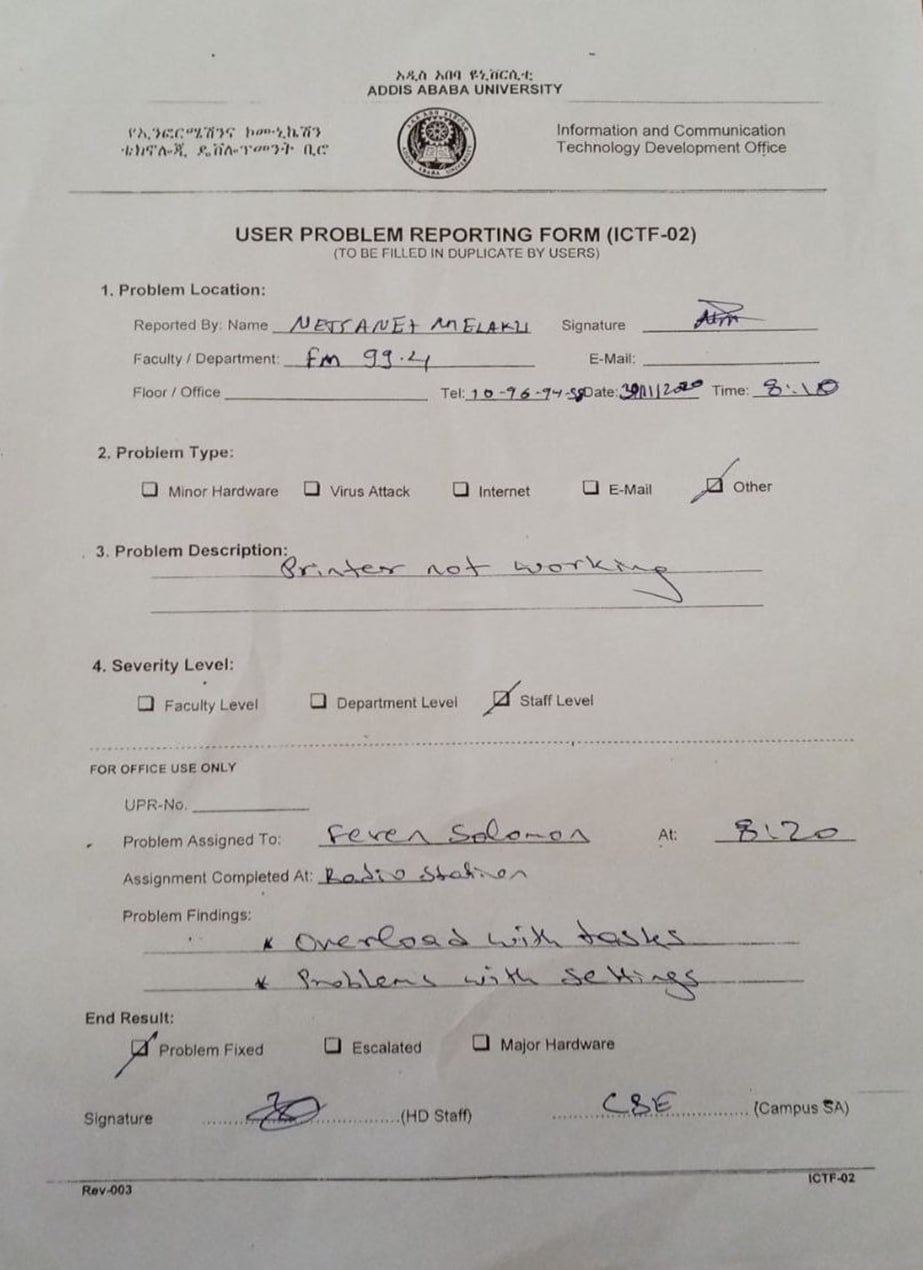


Figure 2. 1Forms and reports of the current system

## 2.2.4 Players of the existing system

* ICT Support and Maintenance Unit manager
  + ICT Support and Maintenance Unit manager supervises the technicians and manages and supervises the workflow.
* ICT Director
* Views report generated by the admin.
* Technicians
  + Solve problems assigned to them and report solved and unsolved problems that are assigned to them.
* Users
* The university community as a whole is the user of the system. They report ICT related problems to the local ICT support and maintenance unit as per the business rule of the system

## 2.3 Requirements Definition

Elaborately discussed below are some of the most important functional and non-functional requirements of our project which is a web-based office property enquiry and technical support system for Addis Ababa University, Faculty of Business and Economics Campus.

## 2.3.1 Functional Requirement

I. Admin (ICT Support and Maintenance Unit manager)

* View report

The team leaders should prepare reports for the admin about the tasks which already performed and based on the reports the admin can easily evaluate the performance of each maintenance department.

* Add new team leaders

As per request the company may hire a new team leader and the admin will give him a password for his workspace.

* Prepare and submit a report to ICT Directorate officer.

1. Team leader

* View number of support requests
* View number of solved support requests
* View number and detail of unsolved requests
* Add a new technician
* Assign tasks to a technician
* List of devices that are out of use
* View report
* Generate report for the admin

1. Technician

* View support requests assigned to him/her
* Contact users via chat for issues
* Generate report per week for team leader
* Receive equipment’s to be maintained.
* Train chat bot that will help users for minor issues.
* Document problem findings.

1. User

* Sign up

The user should have an account to use the system.

* Sign in

The user uses their username and password to be authenticated as a legal user.

* Request for support

The user will request for support from technicians for problems via calls or chats.

* Get AI support

Users will get AI support for minor errors.

* View whether their request approved or not, and when their issues will be addressed in person.
* Communicate with technicians via chat

## 2.3.2 Non-Functional Requirements

Non-functional requirements describe user-visible aspects of the system that are not directly related with the functional behavior of the system.

The system will address the following general non-functional requirements.

1. Performance

The system will perform at optimal speed given that other factors (i.e. network infrastructure, internet speed) operate normally as expected.

1. Security

Laravel helps to secure the web application by protecting it against the most serious security risks: SQL injection, cross-site request forgery, and cross-site scripting.

Moreover, passwords in the login processes must be encrypted before stored in database. When a user login they will be directed to a specified page based on their roles (i.e., Admin, Technician, User) to protect the system from any unauthorized access.

Public users can access the system without any authorization but they only view information but are not able to add, delete or alter any information.

1. Usability

The system has the potential to give service to many users in different fields, in order to support these users; it needs to have different types of user interfaces. It will have friendly interfaces to include individuals with low knowledge of computers and be interactive and simple to use. Since the system has basic functionalities that are easy to understand without any complexities the need for help is low.

1. Interoperability

The system after it is developed is not restricted to any operating system device for usage. Any authorized user with a digital device can access and use the system without the need to have a specified device.

1. Maintainability

Laravel is known for its ease of maintainability and its use of the MVC architecture will allow us to easily maintain, upgrade features and allow for better error management.

1. Availability

The system will be easily maintainable with the aid of technical support and continue its normal operation. The system will be available to authorized users whenever they want to access the system.

1. Reliability

This non-functional requirement depends on the availability and maintainability of the system. The system works on Amazon’s cloud servers and using Laravel will make the system highly reliable.

1. Scalability

This is the main reason the team chose Laravel as the framework for the project as it allows for scalability with its built-in scalability features.

## Chapter 3

## 3.1 Overview

The current IT user support and maintenance system in AAU is discussed in the previous chapter, specifically in CoBE campus. It identified the current system's functionalities, issues, forms, and reports. The current system, the shortcomings, the proposed system, and a diagram that depicts the general outline of the new system are also discussed.

Object-oriented analysis looks at the real-world environment in which a system will operate, with this environment consisting of people and things interacting to create some result. The people and things are first analyzed in the most abstract form and these abstractions become the class. The abstraction is analyzed and reanalyzed in multiple iterations until all objects are uniquely identified. Object characteristics and their behaviors are then analyzed to establish the various states an object can have and to define the methods the object will use to create action. This analysis effort will identify the objects that will need to be created and supported as well as the methods and the messages used by the objects to cause actions.

Accordingly, this study will demonstrate the analysis of the business area in terms of the new system that is to be developed. This phase will essentially focus on what the system is supposed to do. This will be dealing with activities including user interface and business rules identification, system use cases modeling, and documentation. This chapter is the continuation of the essential use case diagram but its modeling will be influenced by the technology to be used for the systems development.

## 3.2 Use Case Modeling

## 3.2.1. User Interface Identification

User interface is the first impression of a software system from the user’s point of view. Therefore any software system must satisfy the requirement of user. UI mainly performs accepting the user's input and displaying the output. (tutorials point, 2023)

In this section, it will go through the process of identifying the user interface with the correspondents' actors. Mainly it tries to identify who is allowed access to that specific UI page. The below table will identify the user interface with: ID, Name, who has access to that page and description.

|  |  |  |  |
| --- | --- | --- | --- |
| UI ID | UI Page | Actors who access the page | Description |
| UI01 | Landing Page | All Actors | A page that contains basic description about the service and a gateway for the Guests, Staff/User, Admin, Team Leader, Technicians |
| UI02 | User Sign in Page | Users/Staff, Admin, Technician, Team Leader | A page where User can fill their credentials to enter the system |
| UI03 | User Register Page | Users/Staff | A page that prompts the User to enter basic information to create an account |
| UI04 | User Dashboard Page | Users | A page that contains the links to the services and notification tab |
| UI05 | User Profile Page | Users/Staff, Admin, Technicians, Team Leader | A page that contains all lists of basic information about the User |
| UI06 | Admin Dashboard Page | Admin |  |
| UI07 | Technician Dashboard Page | Technician |  |
| UI08 | Admin Team Leader Registration Page | Admin | A page that will allow the admin to register team leaders assigned to each campus |
| UI09 | User Issue Report Page | Users/Staff | A page that allows Users/Staff to report their problems |
| UI10 | Admin Delete Team Leader Page | Admin | A page that prompts the admin to enter information in order to remove a Team Leader |
| UI11 | Admin List Technician and Team Leader Page | Admin | A page that contains all the list of Team Leaders and Technicians of that organization |
| UI12 | User Edit Profile | All | A page that enables the User to Edit his/her profile |
| UI13 | Request Page | Team Leader, Admin | A page that contains all the Users issues and support request |
| UI14 | Report Status Pending Page | Team Leader, Admin | A page that holds the pending request |
| UI15 | Admin Report Page | Admin | A page that contains all the request and maintenance report |
| UI16 | User Sign Out Page | All | A page that allows users to sign out |

Table 3. 1 User Interface identification

## 3.2.2 Business rules identification

Business rules guide the everyday decision-making within businesses by outlining the relationships between objects, such as customer names and their corresponding orders. This translation of an organization's business activities into concrete business logic allows software engineers and business analysts to apply these rules within workflow tools or other applications to enable process automation. Without them, updating processes can become more arduous and time-consuming, and documents can be subject to more human error and inconsistencies. By implementing business rules across an organization, a business can save time and money by streamlining work to the right stakeholders and reducing churn.

|  |  |  |
| --- | --- | --- |
| **ID** | **Business rule title** | **Description** |
| Br1 | One admin | There is only one admin for the system |
| Br2 | Only valid users use system | Users with emails that full fill the requirements of the system can use and request support from technicians |
| Br3 | Five times password try | Users with a lost password can try for a maximum of five times then after they should reset their password |
| Br4 | One team leader | There is only one team leader per campus |
| Br5 | Admin issues username and password | Team leaders get their username and password from admin |
| Br6 | Team leader issues username and password | Technicians get username and password from team leader |
| Br7 | AI Bot training | Anyone with technician role has the authority of training a chat bot |
| Br8 | Get AI support | Anyone can get AI support without role restriction |
| Br9 | Prepare Report | All subordinate roles have to prepare weekly report for their immediate upper role |
| Br10 | Team leader assigns for maintenance | Technicians get assigned for the maintenance service they will provide for staff |

Table 3. 2 Business rule identification

## 3.2.3 Actor identification

This section pertains to all users of the system. We have identified the actors and have listed them below

|  |  |  |
| --- | --- | --- |
| ID | ACTOR NAME | DESCRIPTION |
| A01 | Admin | The admin assigns the team leader |
| A02 | Team Leader | The team leader assigns his team of technicians and assigns them tasks |
| A03 | Technician | The technicians are managed directly by their team leader and they communicate with staffs who have requested Support |
| A04 | Staff | The staff are the member of the users who register and request for support and |
| A05 | Guest | The Guest includes all visitors of the site that are able to use the AI bot without the ability to register |

Table 3. 3 Actor identification

## 3.2.4 Designing the use case diagram

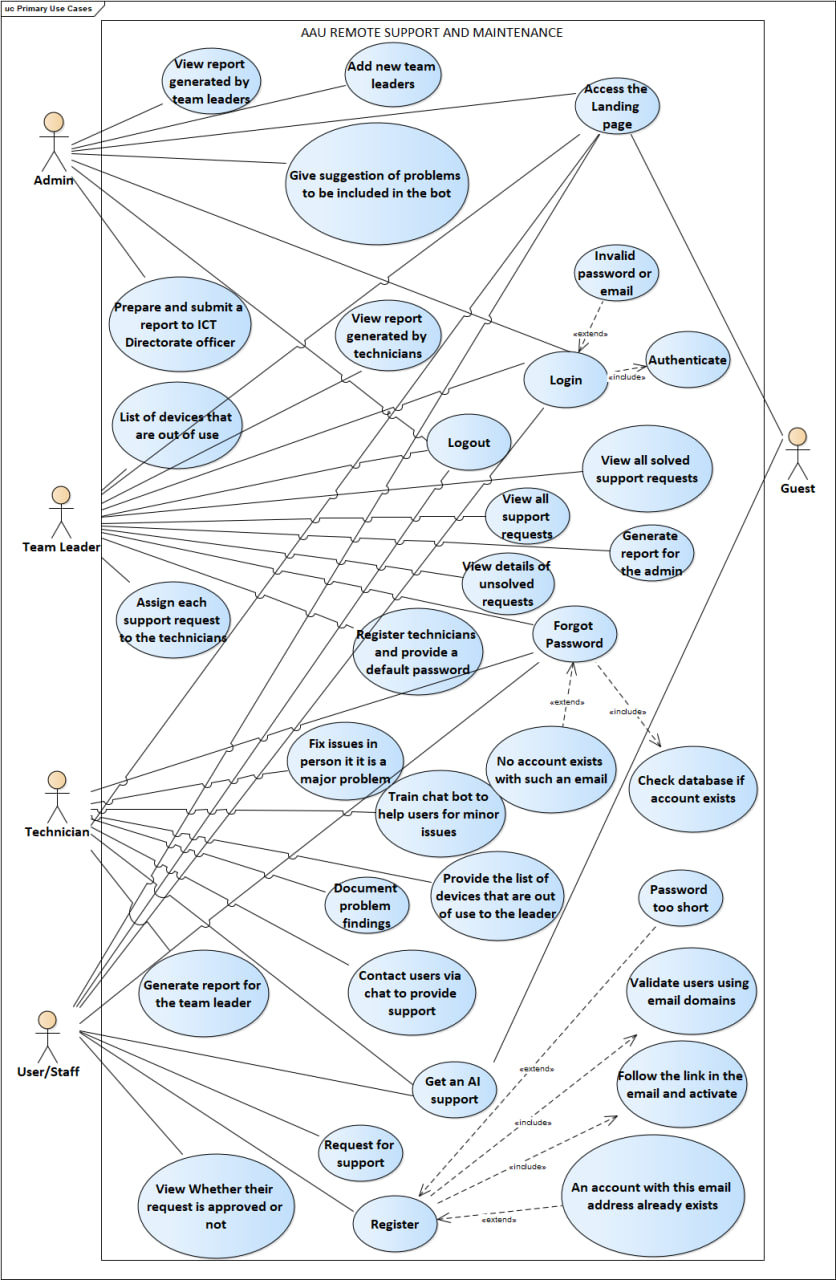


Figure 3. 1 Use Case Diagram

## 3.2.5. Use Case Description

|  |  |
| --- | --- |
| **Name** | **Access Landing Page** |
| **Identifier** | UC01 |
| **Description** | Enables users to understand about AAU’s support team and the services that they provide |
| **Actor** | Admin, Team Leader, Technician, Staff Member, Guest |
| **Pre-condition** | Users should have the url of the website to access it |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when users opens the website  2. The system displays the Landing page  3. Use case ends |
| **Alternative course of action** | A1. User types the url of the website  A2. The browser displays page not found error  A3. User is prompted to turn on wifi or mobile data, troubleshoot their network or enter the url correctly. |
| **Post-condition** | Landing page will be displayed to the user |

Table 3. 4 Landing Page Use Case Description

|  |  |
| --- | --- |
| **Name** | **Use AI bit for support** |
| **Identifier** | UC02 |
| **Description** | Provides users for answers for their minor issues |
| **Actor** | Admin, Team Leader, Technician, Staff Member, Guest |
| **Pre-condition** | Users should login using the email and password provided in the bot |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when users opens the website  2. The system displays the Landing page  3. User selects the AI bot icon displayed in the bottom right corner.  4.User selects what kind of problem they are facing from the drop down menu  5. Users login using the credentials provided in the bot  6. User describes their issues and the bot will try to fix it  7.use case ends here |
| **Alternative course of action** | A1. User fills out login information  A2. System displays “Login error”  A3. User is prompted to check login information. |
| **Post-condition** | The user will be able to solve some of the problems using the bot |

Table 3. 5 AI Bot Use Case Description

|  |  |
| --- | --- |
| **Name** | **Give suggestion of problems to be included in the bot** |
| **Identifier** | UC03 |
| **Description** | Enables users to submit a list of problems that they want them to be trained to the bot. |
| **Actor** | Admin, Team Leader, Technician, Staff Member, Guest |
| **Pre-condition** | Users must have an email address and all the inputs in the form must be filled. |
| **Extends** | Displays error message in user input |
| **Includes** | Validates user inputs |
| **Basic course of action** | 1. Use case starts when users open the website  2. The system displays “UI01 Landing page”  3. User scrolls to the contact page  4. User fills all the inputs in the contact form  5. User inputs are validated  6. Users suggestion has been sent to the technicians email.  7. Use case ends |
| **Alternative course of action** | A1. User fills all the input fields  A2. The system displays error message  A3. User is suggested to improve their input |
| **Post-condition** | Password changed successfully |

Table 3. 6 Give Suggestion to Problems to Include In the Bot Use Case Description

|  |  |
| --- | --- |
| **Name** | **Sign up** |
| **Identifier** | UC04 |
| **Description** | Enables users to create an account only using their email address on the website |
| **Actor** | Admin, Team Leader, Technician, Staff Member |
| **Pre-condition** | Must have AAU’s domain (@aau.edu.et) |
| **Extends** | Password too short and account already exists with the specified email address |
| **Includes** | Validate users using email domains. |
| **Basic course of action** | 1. Use case starts when users open the website  2. The system displays “UC01 Landing page”  3. User selects “Sign up” button  4. The system displays “UC04 sign up” page  5. User inputs required information  6. The system displays user dashboard page  7. Use case ends |
| **Alternative course of action** | A1. User input required information  A2. The system displays signup error  A3. User is prompted to check inputted sign-up information |
| **Post-condition** | Account creation |

Table 3. 7 Sign Up Use Case Description

|  |  |
| --- | --- |
| **Name** | **Login** |
| **Identifier** | UC05 |
| **Description** | Enables users to login to AAU remote support and maintenance website |
| **Actor** | Admin, Team Leader, Technician, Staff Member, Guest |
|  |  |
| **Extends** | Display Login error |
| **Includes** | Authenticate login information |
| **Basic course of action** | 1. Use case starts when any user opens the website  2. The system displays “UC01 Landing Page”  3. user selects “Login” button  4. The system displays sign in page depending on who is accessing the website  5. User fills out login information  6. User gets logged in the system  7. The use case ends |
| **Alternative course of action** | A1. User fills out login information  A2. System displays “Login error”  A3. User is prompted to check login information |
| **Pre-condition** | All users must have an account |
| **Post-condition** | Successful Login |

Table 3. 8 Login Use Case Description

|  |  |
| --- | --- |
| **Name** | **Forgot Password** |
| **Identifier** | UC06 |
| **Description** | Enables users to recover their account using their email address and change their password. |
| **Actor** | Admin, Team Leader, Technician, Staff Member |
| **Pre-condition** | An account must exist in that specified email address. |
| **Extends** | No account exists with this email address. |
| **Includes** | Checks the database if an account exists. |
| **Basic course of action** | 1. Use case starts when users open the website  2. The system displays “UC01 Landing page”  3. User selects “Login” button  4. The system displays “UC05 Login” page  5. User selects “Forgot Password” button  6. User inputs Email address  7. Password reset link is sent over the email address  8. User is promoted to enter a new password  9. User password is changed successfully and user is logged in.  10. Use case ends |
| **Alternative course of action** | A1. User provides a new password  A2. The system displays error message if the new password is the same as the old one.  A3. User is prompted to enter a new password again |
| **Post-condition** | Password changed successfully |

Table 3. 9 Forgot Password Use Case Description

|  |  |
| --- | --- |
| **Name** | **Add and assign new team leader** |
| **Identifier** | UC07 |
| **Description** | Admins can add and assign a new team leader to any campus of AAU. |
| **Actor** | Admin |
| **Pre-condition** | Each new technician first needs to be hired by the HR first. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when a new technician is hired with a title team leader by the HR.  2. The admin will assign the team leader to a campus.  3. Use case ends |
| **Alternative course of action** |  |
| **Post-condition** | A team leader is assigned to a campus by the admin. |

Table 3. 10 Add Team Leader Use Case Description

|  |  |
| --- | --- |
| **Name** | **View report generated by the team leader** |
| **Identifier** | UC08 |
| **Description** | Enables the Admin to view a report which is generated by each team leader who works in each campus. |
| **Actor** | Admin |
| **Pre-condition** | Must provide correct credential and has a role of Super Admin. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when users open the website  2. The system displays “UC01 Landing page”  3. User presses login and fills out login information  4. If the user credentials are correct and user has a role or Super Admin  5. User is logged in to the system and views the report  6. Use case ends |
| **Alternative course of action** | A1. Admin navigates to show report  A2. System displays no report found if there is no report generated by the team leaders  A3. Admin is suggested to remind the team leaders to generate the report |
| **Post-condition** | Admin can view the report. |

Table 3. 11 View Report Generated By Team Leader Use Case Description

|  |  |
| --- | --- |
| **Name** | **Generate a report for ICT directorate office** |
| **Identifier** | UC09 |
| **Description** | Enables the Admin to generate a report for the ICT directorate about the general overview of ICT support and maintenance unit in accordance with the time line. |
| **Actor** | Admin |
| **Pre-condition** | Must have access to all the reports that came from a team leader from each campus. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when user is logged in with a role super admin.  2. The system will display each report from the campus  3. Admin will prepare a report using each team leader’s report.  4. Use case ends |
| **Alternative course of action** | A1. Admin navigates to show report  A2. System displays no report found if there is no report generated by the team leaders  A3. Admin is suggested to remind the team leaders to generate the report |
| **Post-condition** | Admin can view the report. |

Table 3. 12 Generate a Report for ICT Directorate Office

|  |  |
| --- | --- |
| **Name** | **Register technicians and provide a default password** |
| **Identifier** | UC10 |
| **Description** | Enables the team leader to add a new technician to their campus. |
| **Actor** | Team Leader |
| **Pre-condition** | The technicians must be hired by the HR to that campus. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when user is logged in with a role an Admin.  2. The team leader will navigate to add a new technician.  3. Team leader will provide a default password for the technician  4. Use case ends |
| **Alternative course of action** |  |
| **Post-condition** | Team leader can add a new technician to the campus. |

Table 3. 13 Register technicians and provide a default password

|  |  |
| --- | --- |
| **Name** | **View details of unsolved requests** |
| **Identifier** | UC11 |
| **Description** | Enables the team leaders to view which support requests have not been addressed or that haven’t been solved. |
| **Actor** | Team Leader |
| **Pre-condition** | Must provide correct credential and has a role of Admin. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when users open the website  2. The system displays “UC01 Landing page”  3. User presses login and fills out login information  4. If the user credentials are correct and user has a role of Admin  5. User is logged in to the system and views the list of problems which are not solved and the respective reason for it.  6. Use case ends |
| **Alternative course of action** |  |
| **Post-condition** | Team leader can view list of unsolved problems. |

Table 3. 14 View Unsolved Issues Use Case Description

|  |  |
| --- | --- |
| **Name** | **Assign each support request to a technician** |
| **Identifier** | UC012 |
| **Description** | Enables the team leader to assign a technician for every support request that might come from the users. |
| **Actor** | Team Leader |
| **Pre-condition** | The technicians to be assigned the task must be available within the campus. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when user is logged in with a role an Admin.  2. The team leader will assign a team leader when a new issue arises  3. Use case ends |
| **Alternative course of action** |  |
| **Post-condition** | Team leader can assign a support request to the technicians on duty |

Table 3. 15 Assign Support Request to Technicians Use Case Description

|  |  |
| --- | --- |
| **Name** | **View all solved support request** |
| **Identifier** | UC13 |
| **Description** | Enables the team leaders to view which support requests have been addressed and are solved successfully. |
| **Actor** | Team Leader |
| **Pre-condition** | Must provide correct credential and has a role of Admin. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when users open the website  2. The system displays “UC01 Landing page”  3. User presses login and fills out login information  4. If the user credentials are correct and user has a role of Admin  5. User is logged in to the system and views the list of problems which are addressed by the assigned technician.  6. Use case ends |
| **Alternative course of action** |  |
| **Post-condition** | Team leader can view list of solved problems. |

Table 3. 16 View Solved Support Issues Use Case Description

|  |  |
| --- | --- |
| **Name** | **View report generated by technicians** |
| **Identifier** | UC14 |
| **Description** | Enables the team leaders to view a report which is generated by each technician who works in that specific campus. |
| **Actor** | Team Leader |
| **Pre-condition** | Must provide correct credential and has a role of Admin. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when users open the website  2. The system displays “UC01 Landing page”  3. User presses login and fills out login information  4. If the user credentials are correct and user has a role of Admin  5. User is logged in to the system and views the report  6. Use case ends |
| **Alternative course of action** | A1. Team leader navigates to show report  A2. System displays no report found if there is no report generated by the technicians  A3. Team leader is suggested to remind the technicians to generate a report |
| **Post-condition** | Team leader can view the report. |

Table 3. 17 View Report Generated by Technicians Use Case Description

|  |  |
| --- | --- |
| **Name** | **View list of devices that are out of use** |
| **Identifier** | UC015 |
| **Description** | Enables the team leader to see all list of devices that are out of use or devices that cannot be fixed. |
| **Actor** | Team Leader |
| **Pre-condition** | The devices must be added by the technicians |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when user is logged in with a role an Admin.  2. The team leader will navigate to out of used devices to access the list.  3. Team leader can view all list of devices and their respective details.  4. Use case ends |
| **Alternative course of action** |  |
| **Post-condition** | Team leader can view the list of out of use devices. |

Table 3. 18 View List of Out of Use Devices Use Case Description

|  |  |
| --- | --- |
| **Name** | **Generate a report for the admin** |
| **Identifier** | UC016 |
| **Description** | Enables the team leader to generate a report for the admin. |
| **Actor** | Team Leader |
| **Pre-condition** | Must have access to all the reports that came from each technician in their campus. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when user is logged in with a role an Admin.  2. The system will display a report from each technician  3. Team leader will prepare a report using each team technician’s report.  4. Use case ends |
| **Alternative course of action** | A1. Team leader navigates to show report  A2. System displays no report found if there is no report generated by the technicians  A3. Team leader is suggested to remind the technician to generate the report |
| **Post-condition** | Team leader can view the report. |

Table 3. 19 Generate a Report for the Admin Use Case Description

|  |  |
| --- | --- |
| **Name** | **View a list of assigned task** |
| **Identifier** | UC017 |
| **Description** | Enables technicians to see a list of tasks or support request to complete by the team leader. |
| **Actor** | Technicians |
| **Pre-condition** | A task must be assigned to the technicians by the admin. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when user is logged in with a role Technician.  2. The technician will navigate to view list of assigned task.  3. Technicians can view all list of devices and their respective details.  4. Use case ends |
| **Alternative course of action** |  |
| **Post-condition** | Technicians will be able to see what task is assigned to them and grasp an insight about the problem at hand. |

Table 3. 20 View List of Assigned Task Use Case Description

|  |  |
| --- | --- |
| **Name** | **Contact users via chat to provide support** |
| **Identifier** | UC18 |
| **Description** | Enables the technicians to provide an online walk through to the users in order to remove their issues. |
| **Actor** | Technicians |
| **Pre-condition** | Technicians must be assigned to that specific issue by the team leader. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when user is logged in.  2. Technicians will initiate a chat with the users to provide their support.  3. Use case ends |
| **Alternative course of action** |  |
| **Post-condition** | Technicians will help the users to remove their issues by themselves. |

Table 3. 21 Contact Users via Chat Use Case Description

|  |  |
| --- | --- |
| **Name** | **Fix issues in person of if it is a major problem** |
| **Identifier** | UC19 |
| **Description** | Enables the technicians to fix any problem either in person or in a remote support from their office. |
| **Actor** | Technicians |
| **Pre-condition** | The support request must be from the campus staff member. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when user is logged in with a role technician.  2. The technician will check for any task that is assigned to them by the team leader.  3. The technicians will fix those problems  4. Use case ends |
| **Alternative course of action** |  |
| **Post-condition** | Technicians will remove any difficulties that the campus staff members face. |

Table 3. 22 Fix Issues in Person Use Case Description

|  |  |
| --- | --- |
| **Name** | **Document problem findings** |
| **Identifier** | UC20 |
| **Description** | Enables the technicians describe what problems they face in their daily activity and document their findings about the support problem. |
| **Actor** | Technicians |
| **Pre-condition** | There should exist a support request to study and determine its problem finding and document it for generating a report. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when user is logged in.  2. Technicians will fill the form that is used to determine all the problem description and how it was solved.  3. This document is sent to the team leader by the technician  4. Use case ends |
| **Alternative course of action** |  |
| **Post-condition** | Technicians will document their problem finding and send it to the team leader. |

Table 3. 23 Document Problem Findings Use Case Description

|  |  |
| --- | --- |
| **Name** | **Add a list of devices that are out of use** |
| **Identifier** | UC21 |
| **Description** | Enables the technicians teach to add a list of devices in the campus that they are unable to fix. |
| **Actor** | Technicians |
| **Pre-condition** | The device needs to be out or it cannot be fixed. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when user is logged in.  2. Technicians will navigate to add out of use devices and provide the detail  3. List of out of use devices is updated  4. Use case ends |
| **Alternative course of action** |  |
| **Post-condition** | Technicians will be add a device whenever they face a device that they couldn’t fix. |

Table 3. 24 Add Out of Use Devices Use Case Description

|  |  |
| --- | --- |
| **Name** | **Train the chat bot** |
| **Identifier** | UC22 |
| **Description** | Enables the technicians teach the bot in order to improve its knowledge base based on the problems that they mainly face when executing their task |
| **Actor** | Technicians |
| **Pre-condition** | Must be a technician at AAU |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when user is logged in.  2. Technicians will add a new answer and question to train and improve the bot.  3. Use case ends |
| **Alternative course of action** |  |
| **Post-condition** | Technicians will be able to update the knowledge base of the bot. |

Table 3. 25 Train AI Chat Bot Use Case Description

|  |  |
| --- | --- |
| **Name** | **Generate a report for the team leader** |
| **Identifier** | UC23 |
| **Description** | Enables the technicians to generate a report for the team leader. |
| **Actor** | Technicians |
| **Pre-condition** | Must analyze all their problem findings to generate a generalized report. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when user is logged in with a role technician.  2. Technicians will prepare a report using their analysis.  4. Use case ends |
| **Alternative course of action** |  |
| **Post-condition** | Technicians will prepare a generalized report for the team leader. |

Table 3. 26 Generate a Report for the Team Leader Use Case Description

|  |  |
| --- | --- |
| **Name** | **Request for support** |
| **Identifier** | UC24 |
| **Description** | Enables users or staff members of AAU to request for support. |
| **Actor** | Users/Staff members |
| **Pre-condition** | Must be a staff member of AAU and have an email address with a suffix of @aau.edu.et. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when user is logged in with a role user.  2. Users will navigate to request for support.  3. Users will fill the form which an input like name, description, model, photo, etc...  4. Submit their request to the team leader  5. Use case ends |
| **Alternative course of action** |  |
| **Post-condition** | Users will be able to request for a support by submitting the form. |

Table 3. 27 Request for Support Use Case Description

|  |  |
| --- | --- |
| **Name** | **View if their request has been approved or not** |
| **Identifier** | UC25 |
| **Description** | Enables users or check if their request has been approved or not. |
| **Actor** | Users/Staff members |
| **Pre-condition** | They should have already requested for a support. |
| **Extends** |  |
| **Includes** |  |
| **Basic course of action** | 1. Use case starts when user is logged in with a role user.  2. Users will navigate to request status.  3. Users will find if their request has been approved or if it’s on pending.  4. Use case ends |
| **Alternative course of action** |  |
| **Post-condition** | Users will be able to check the status of their request. |

Table 3. 28 View if Their Request is Approved or Not Use Case Description

|  |  |
| --- | --- |
| **Name** | **Logout** |
| **Identifier** | UC26 |
| **Description** | Enables users to log out of AAU’s remote support and maintenance website |
| **Actor** | Admin, Team Leader, Technician, Staff Member, Guest |
|  |  |
| **Extends** |  |
| **Includes** | Destroys sessions |
| **Basic course of action** | 1. Use case starts when any user presses the logout button  2. The system displays the login page  3. The use case ends |
| **Alternative course of action** |  |
| **Pre-condition** | All users must be logged in first |
| **Post-condition** | Successfully logged out |

Table 3. 29 Logout Use Case Description

## 3.3 Conceptual Modeling

## 3.3.1 Class diagram

A class diagram is a graphical representation of the system's static view that shows various aspects of the application. The system is represented by a collection of class diagrams. The static view of an application is represented by a class diagram. Class diagrams are the only diagrams that can be directly mapped with object-oriented languages, making them popular during the construction phase.

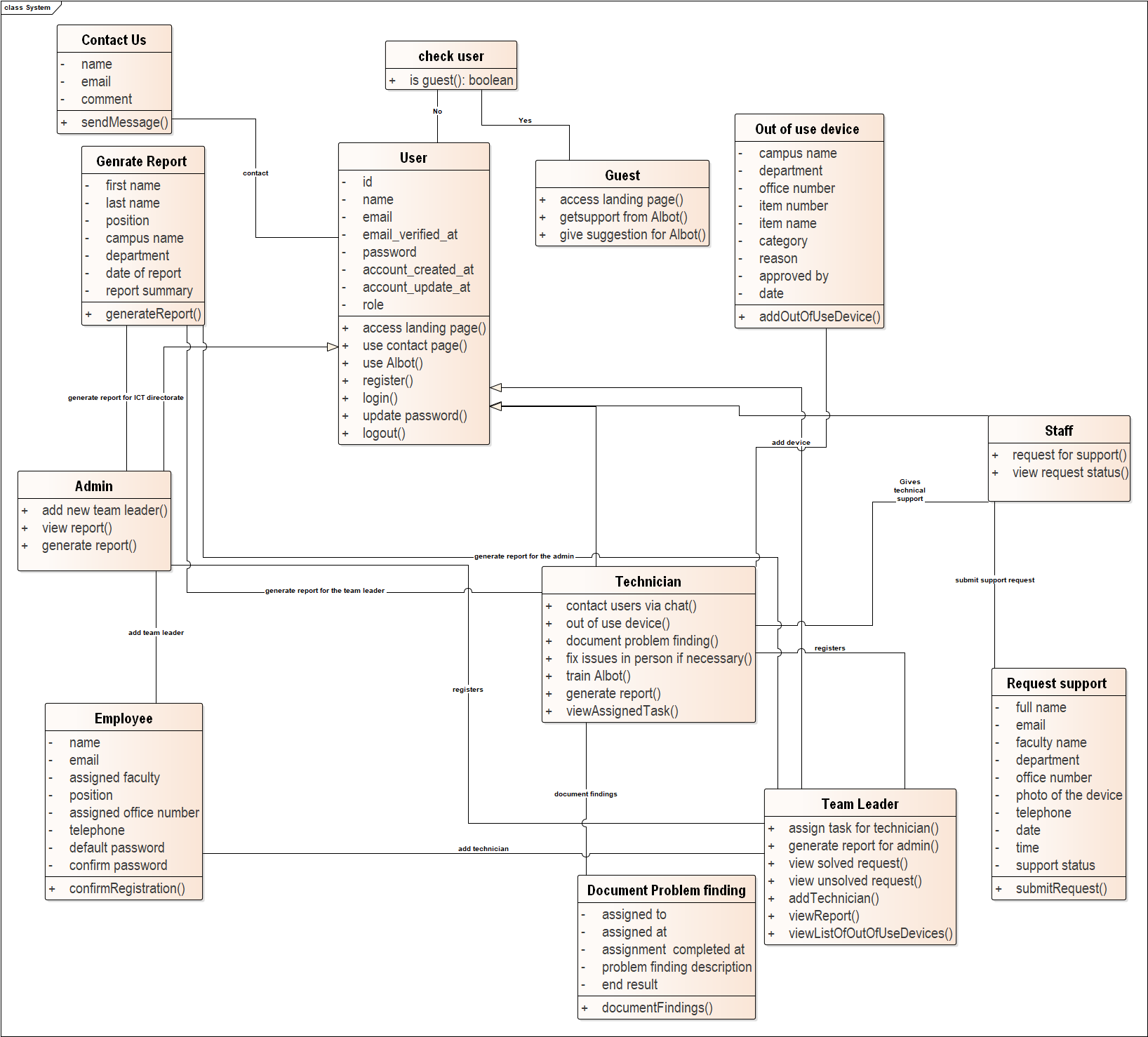


Figure 3. 2 Class Diagram

## Class description

**Class Name Check User**

**Method**

* **isGuest()**
* Check whether the user is guest or not.

**Class Name Users**

**Attribute**

* Email
  + Is one of the credential used by the user to log in to the system.
* Email\_Verified\_At
  + Is the date that the email is verified at.
* Id
  + Assigned to each user as a unique identifier
* Name
  + Is used when the user is registered and is important to verify each user.
* Password
  + It is used during login to authenticate users.
* Role
  + It used for authorization to determine the level of access users have.

**Method**

* LogIn()
  + Initializes a method for the user to login.
* LogOut()
  + Initializes a method for the user to logout.

**Class Name Admin Extends User Class**

**Method**

* AddNewTeamLeader()
  + Allows the admin to add and assign new team leader.
* GiveBotSuggestion()
  + Allows the admin to give a suggestion on what should be updated on the bot.
* ReportToICTDirector()
  + The admin prepares a report to the ICT Directorate.
* AccessLandingPage()
  + The admin accesses the landing page of the website.
* GenerateReport()
  + The admin generates reports gathered from the team leaders.

**Class Name Team Leader Extends User Class**

**Method**

* AssignTechnician()
  + Allows the team leaders to assign tasks to technicians.
* GenerateReport()
  + The team leaders generate reports to the admin based gatherings from technician reports.
* ViewSuportRequest()
  + The team leader is able to view support request which have been issues by users.
* ViewUnsolvedSupportRequest()
  + The team leader is able to view support requests which have been issued by users and still not solved by technicians.

**Class Name Technician Extends User Class**

**Method**

* ContactUsers()
  + Allows the technicians to directly contact the users who requested for support.
* DocumentProblem()
  + The technicians report their problem findings in each task they carry out.
* FixProblem()
  + The technicians view their assigned tasks in order to fix problems.
* GenerateReport()
  + The technicians generate report of their work.
* OutOfUseDevice()
  + The technicians create a log of devices which cannot be fixed(out of use).
* TrainBot()
  + The technicians are able to train the bot to respond well to users issues.

**Class Name Staff Extends User Class**

**Method**

* BotSupport()
  + The staffs are able to get support from the bot on minor technical issues.
* Register()
  + Staff members with the valid credential of AAU domain email address are able to register to use the system.
* RequestSupport()
  + Staff members who have registered to the system are able to request support on their issues.
* CheckRequestStatus()
  + Staff who requested for support are able to view the status of their issues.

**Class Name Guest**

**Method**

* BotSupport()
  + The guest users are able to get support from the bot on minor technical issues.
* AccessLandingPage()
  + Any guest user is able to view the landing page of the website.

**Class Name Out of use device**

**Attribute**

* Campus name
* Contains the name of the campus in which the device is located in.
* Department
* Contains the name of the department in the specific campus in which the device in currently located in.
* Office number
* Contains the number of the office in the department which holds the device.
* Item number
* This attribute contains the data about the numerical identity of the device.
* Category
* Contains data about the category of the device that is out of use.
* Reason
* Reason is the description of why the device was registered as out of use.
* Approved by
* The technician writes his/her name after he approves the device out of use.
* Date
* Contains the date in which the device was registered as out of use.

**Method**

* addOutOfUseDevice()
* This method enables the technician to register a device which is out of use.

**Class Name Document problem finding**

**Attributes**

* Assigned to
* This represents the technician which was assigned the problem by the team leader.
* Assigned at
* Contains the date in which the problem was assigned at.
* Assignment completed at
* Contains the date in which the task was completed at by the technician.
* Problem finding description end result
* This contains the general description of the problem and the end results found the technician.

**Method**

* documentFindings()
* This method allows the technician to document the end result of the task.

**Class Name Employee**

**Attribute**

* Name
* Contains the name of the employee.
* Email
* Contains the email of the employee which was assigned by the university.
* Assigned Faculty
* Contains the faculty in which the employee is assigned at.
* Position
* Contains the position of the employee which was assigned to him/her.
* Assigned office number
* Contains the office number in which the employee will be working at.
* Telephone
* Contains the phone number of the employee
* Default password
* Contains the password which will be given to the employee during registration.
* Confirm password
* Contains the assigned password for confirmation during system access.

**Method**

* confirmRegistration()
* This method allows the registering individual to complete the action.

**Class Name Contact us**

**Attributes**

* Name
* Identifies the name of the person who wants to contact with.
* Email
* The email address of the person who wants to contact with.
* Comment
* Any suggestion, feedback and recommendation forwarded

**Methods**

* Send message ()
* Send the comment

**Class Name Generate report**

**Attributes**

* First name
* The last name of the person who generates the report
* Last name
* The last name of the person who generates the report
* Position
* The position of the person who generates the report
* Campus name
* The campus the person works
* Date of report
* The date at which the report is generated.
* Report summary
* The report itself.

**Methods**

* Generate report ()
* Creates report with the listed contents.

**Class Name Request support**

**Attributes**

* **Full name**
* The name of the requestor
* **Email**
* The email address of the requestor
* Faculty name
* The name of the faculty the requestor works.
* Department
* The department the requestor is part of.
* Office number
* The office of the requestor
* Photo of the device
* The requestor should take a photo of the device to be fixed and send it as part of the request.
* Telephone
* The phone number of the requestor
* Date
* The date when the request is created.
* Time
* The time when the request is created.
* Support status
* The status(pending, assigned) of the request after it is submitted.

**Methods:**

* Submit request ()
* After the request is created, it will be submitted.

## 3.4 Sequence Diagramming

UML sequence diagrams model the flow of logic within a system in a visual manner, enabling both to document and validate logic, and are commonly used for both analysis and design purposes (Ambler, 2004). For this project, the chosen to map are based on the role of the actors, the Users, Guests, Technician, Team leader, and Admin.

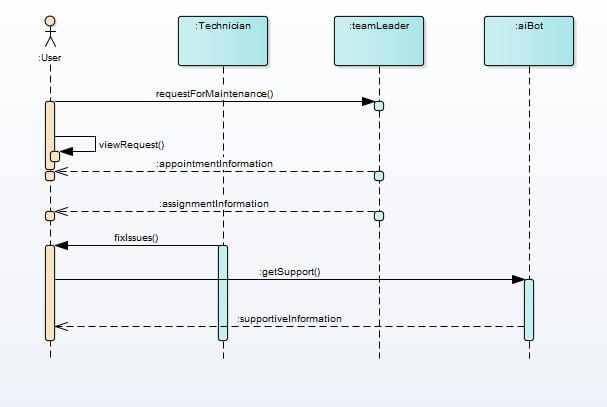


Figure 3. 3 User Sequence Diagram

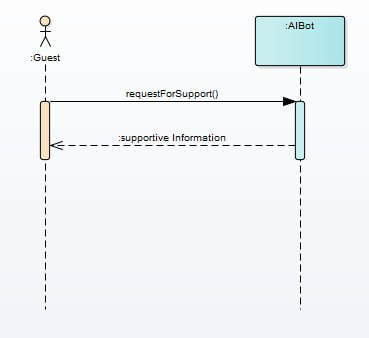


Figure 3. 4 Guest Sequence Diagram

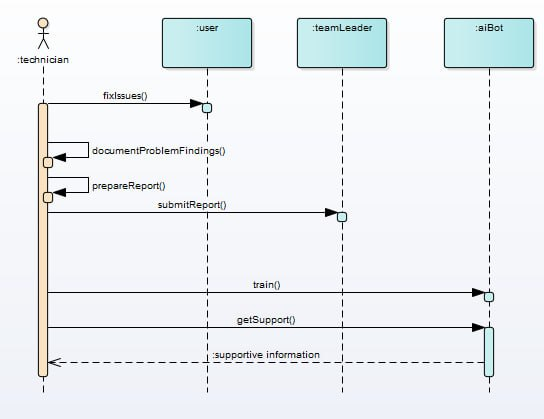


Figure 3. 5 Technician Sequence Diagram

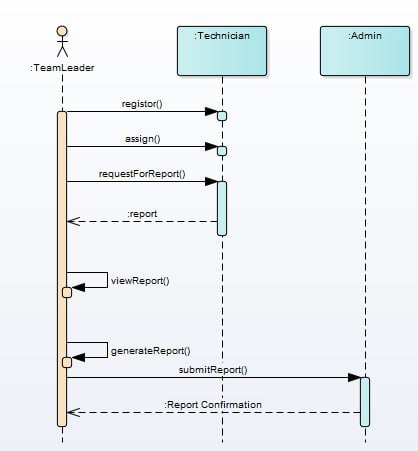


Figure 3. 6 Team Leader Sequence Diagram

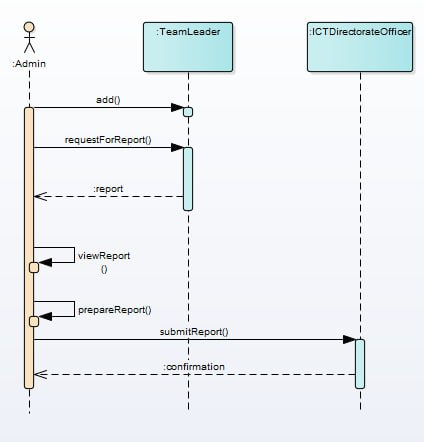


Figure 3. 7 Admin Sequence Diagram

## 3.5 User Interface Prototyping

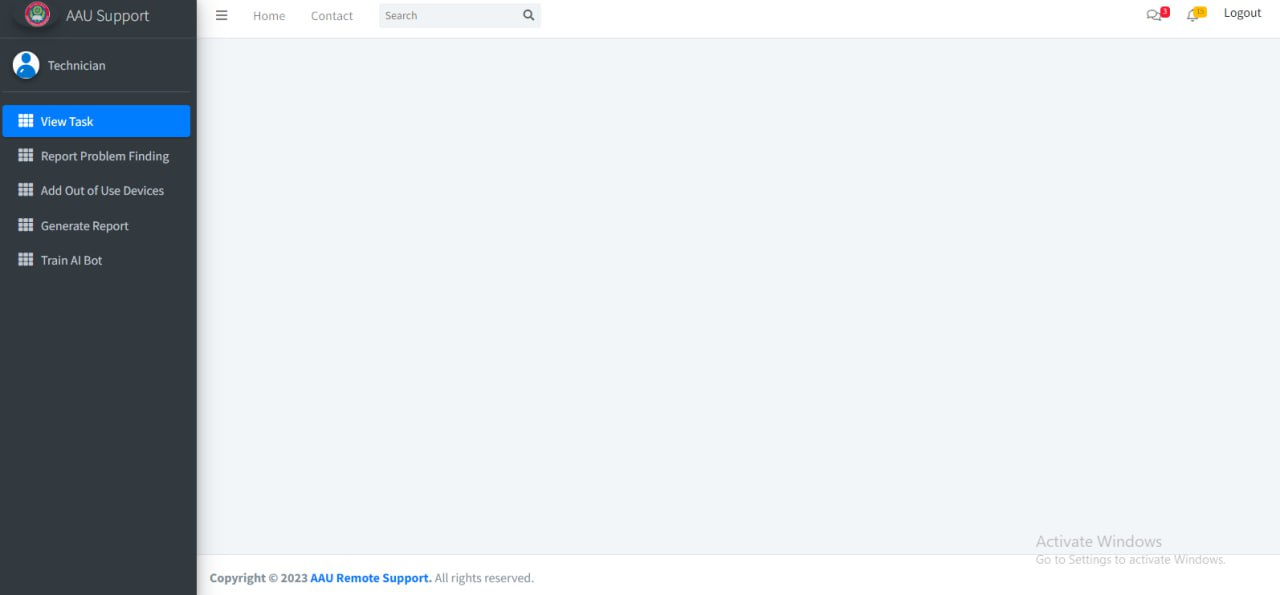


Figure 3. 8 Technician Dashboard

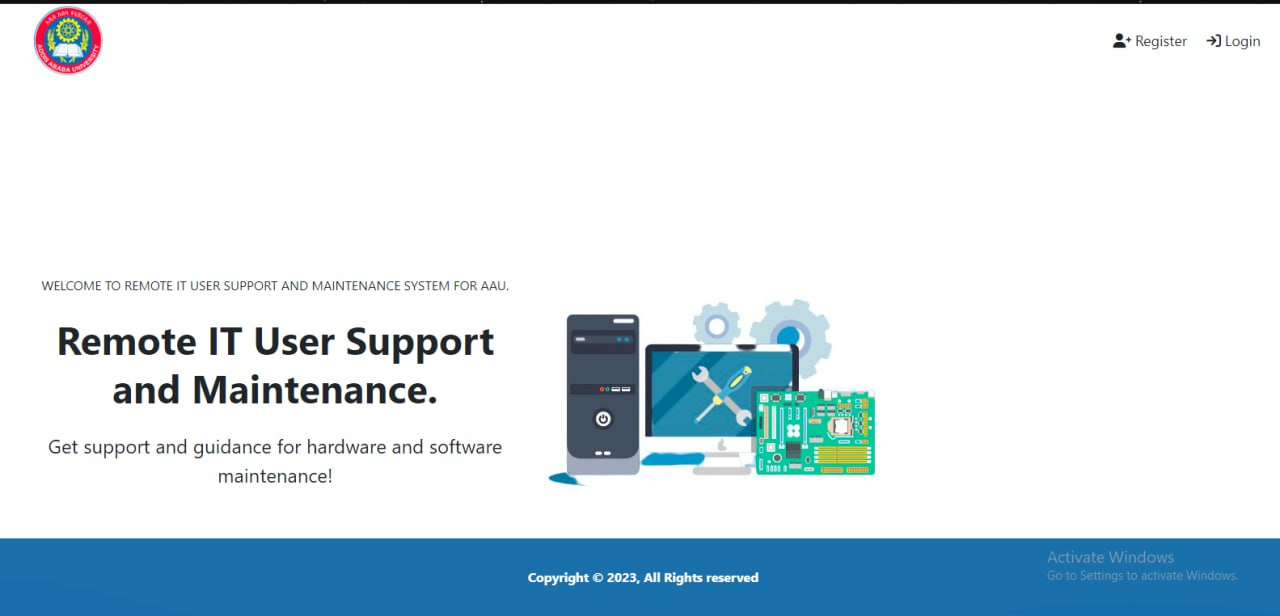


Figure 3. 9 Landing Page(Home)

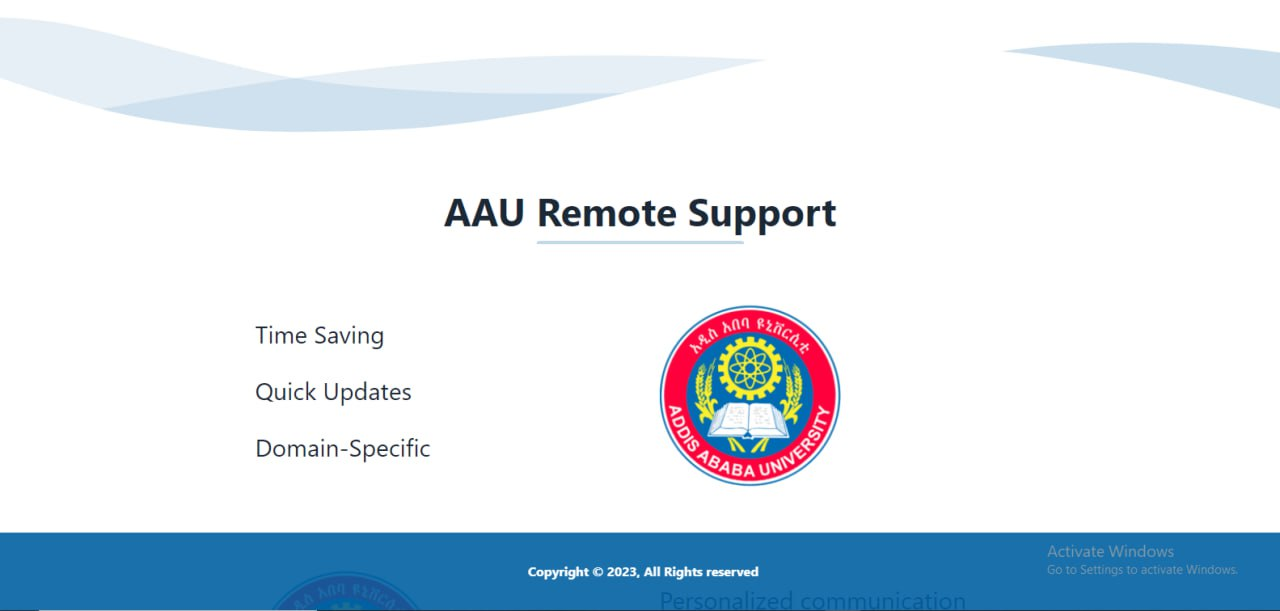


Figure 3. 10 Landing Page(About)



Figure 3. 11 Landing Page(About)

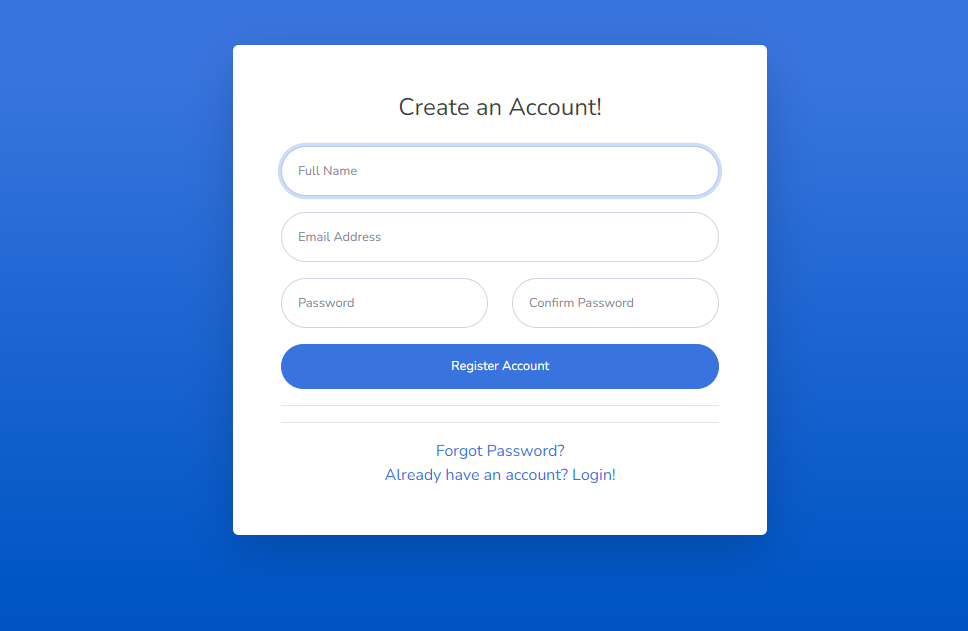


Figure 3. 12 Register

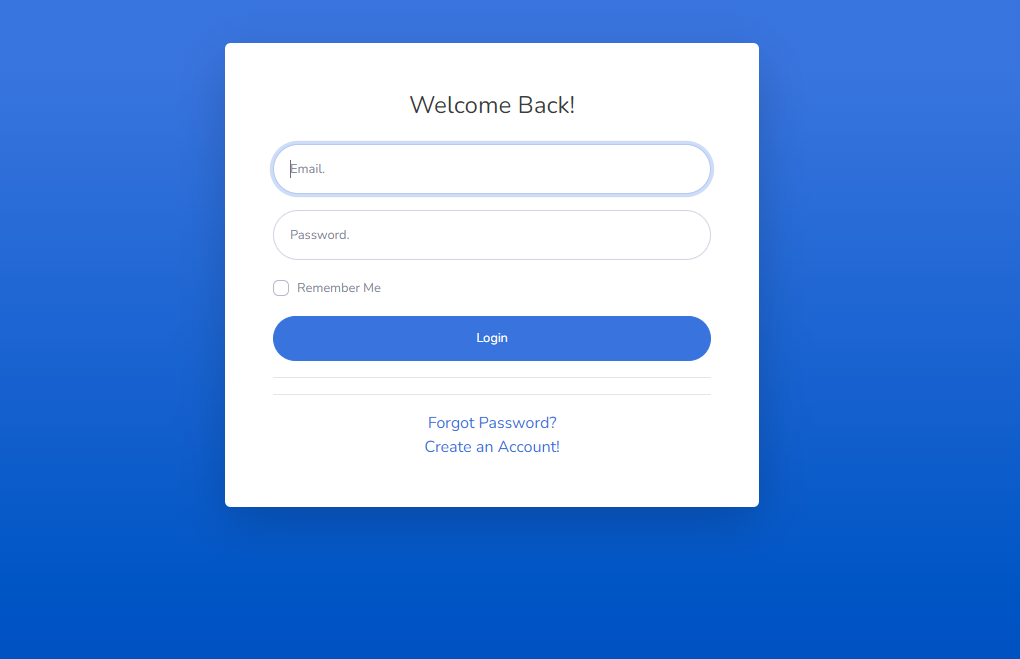


Figure 3. 13 Login

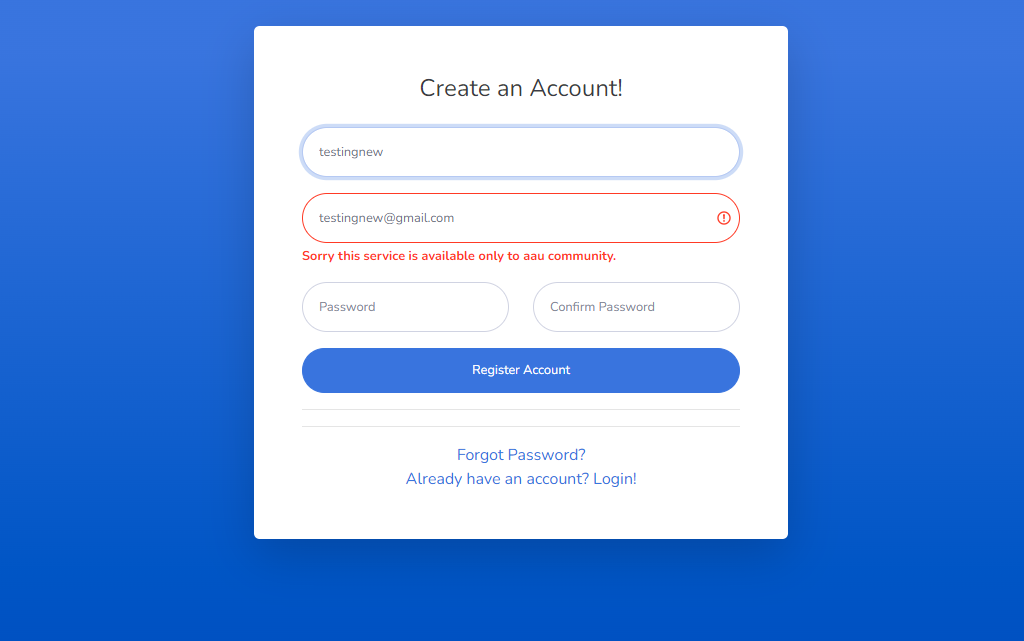
**

Figure 3. 14 Login Error Message

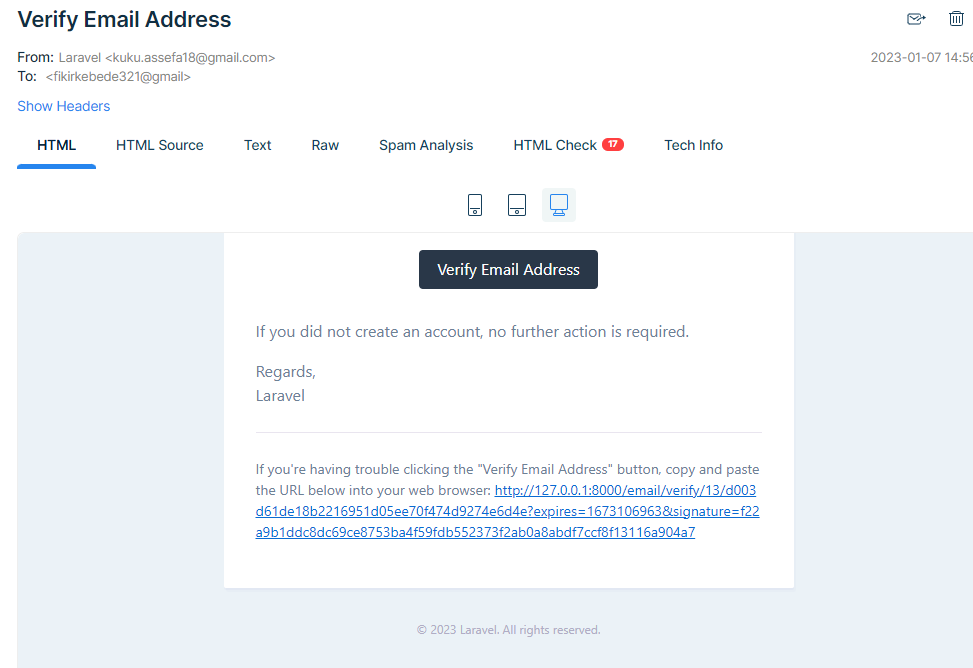
**

Figure 3. 15 Verify Email Address

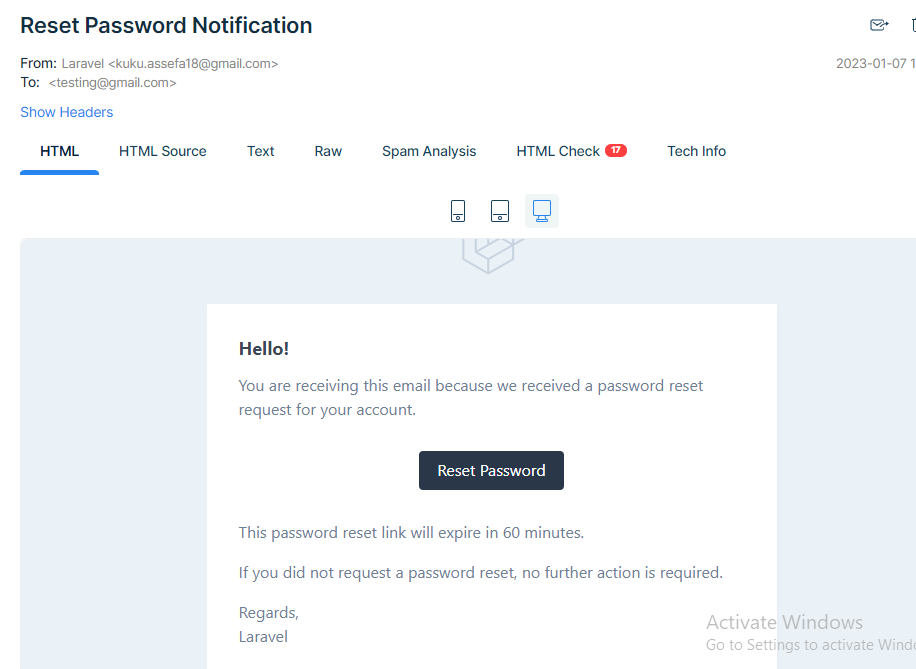
**

Figure 3. 16 Reset Password Verify

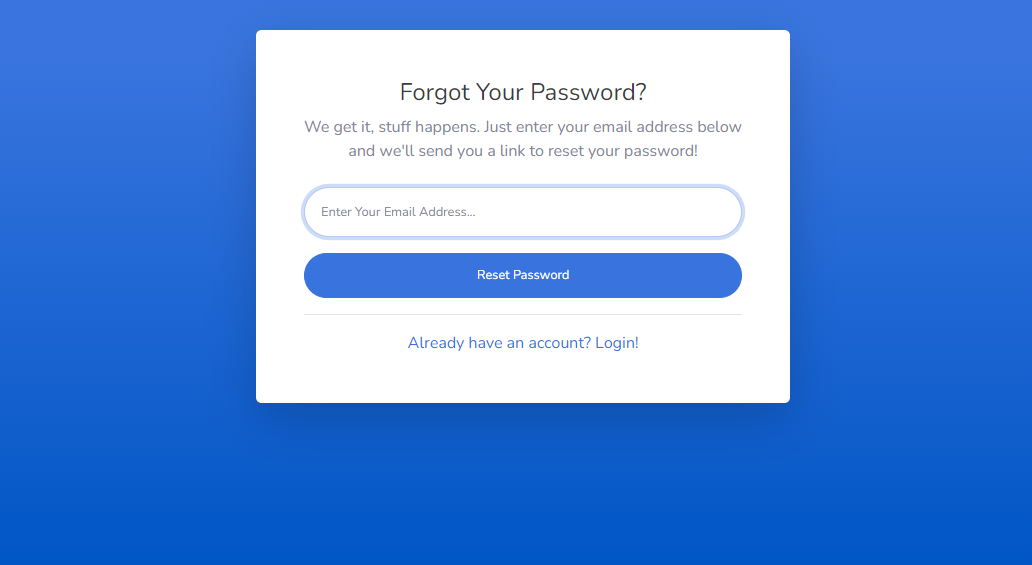


Figure 3. 17 Forgot Password

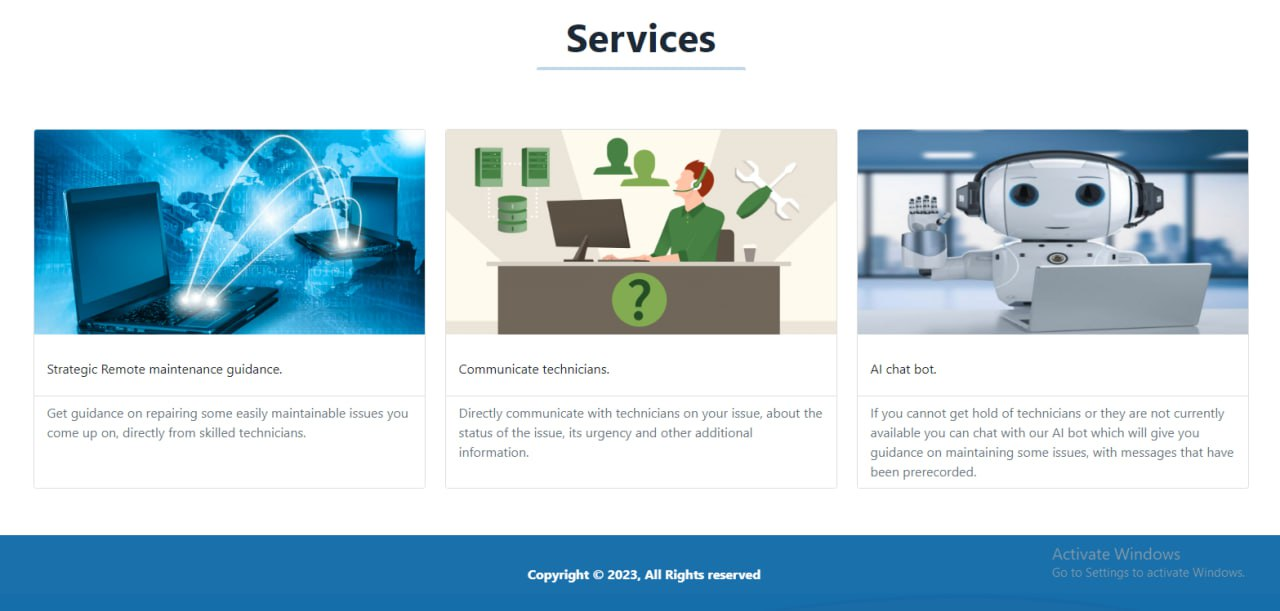


Figure 3. 18 Landing Page(Services)



Figure 3. 19 Landing Page(About)

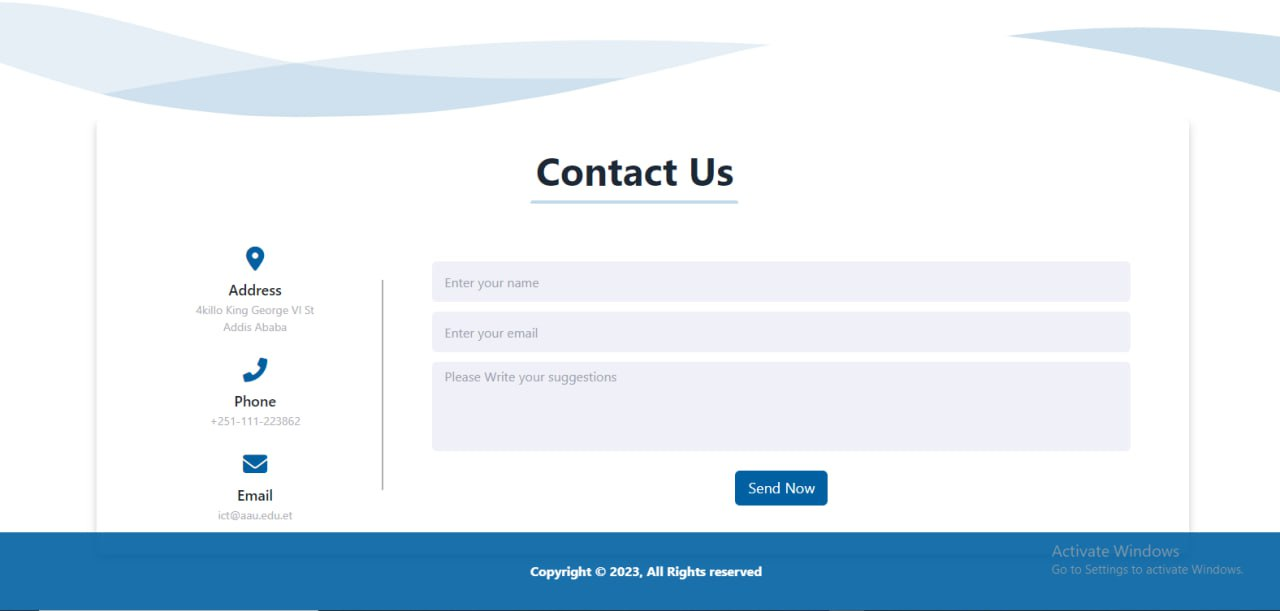


Figure 41 Landing Page(Contact Us)

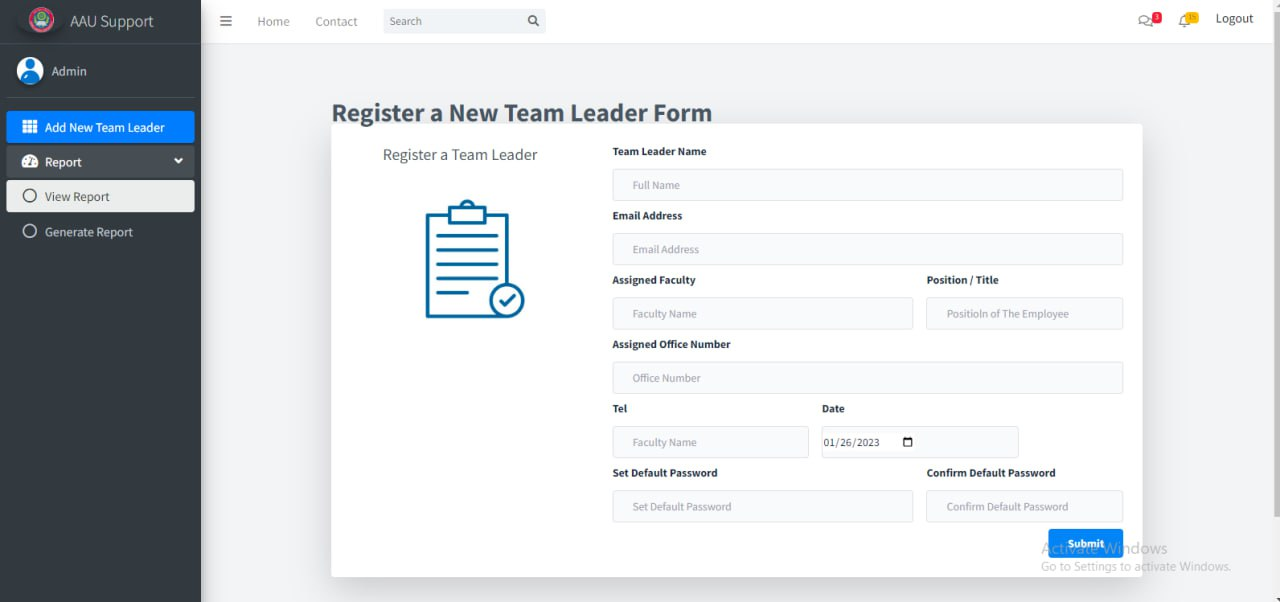


Figure 3. 20 Register new team leader form

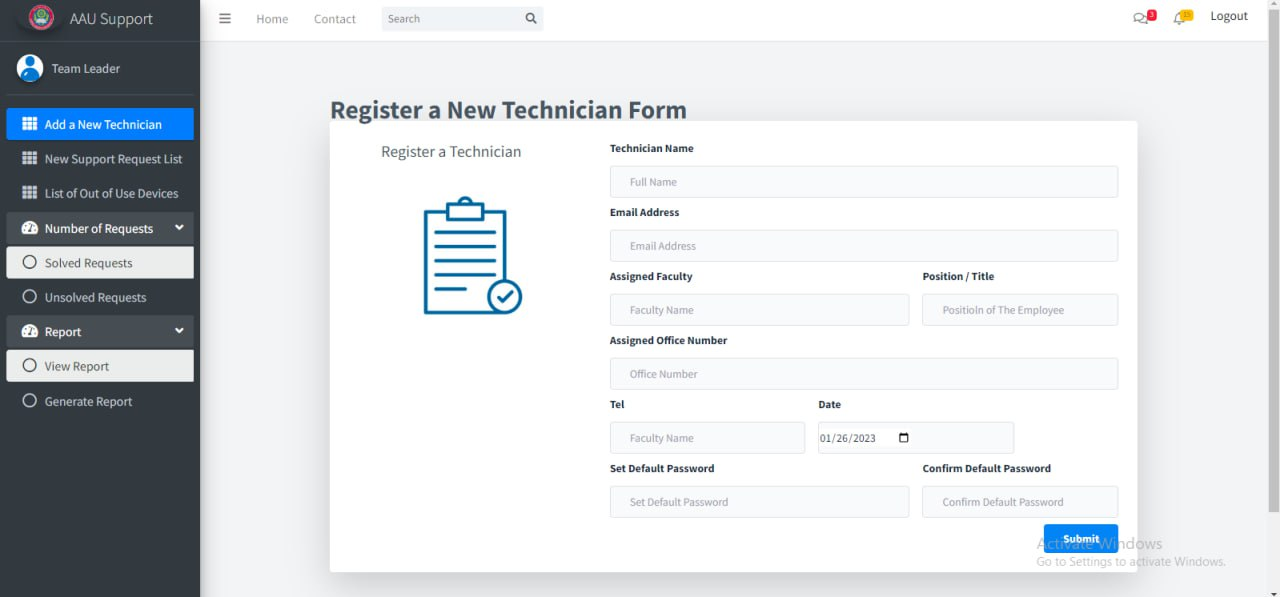


Figure 3. 21 Register new technician form

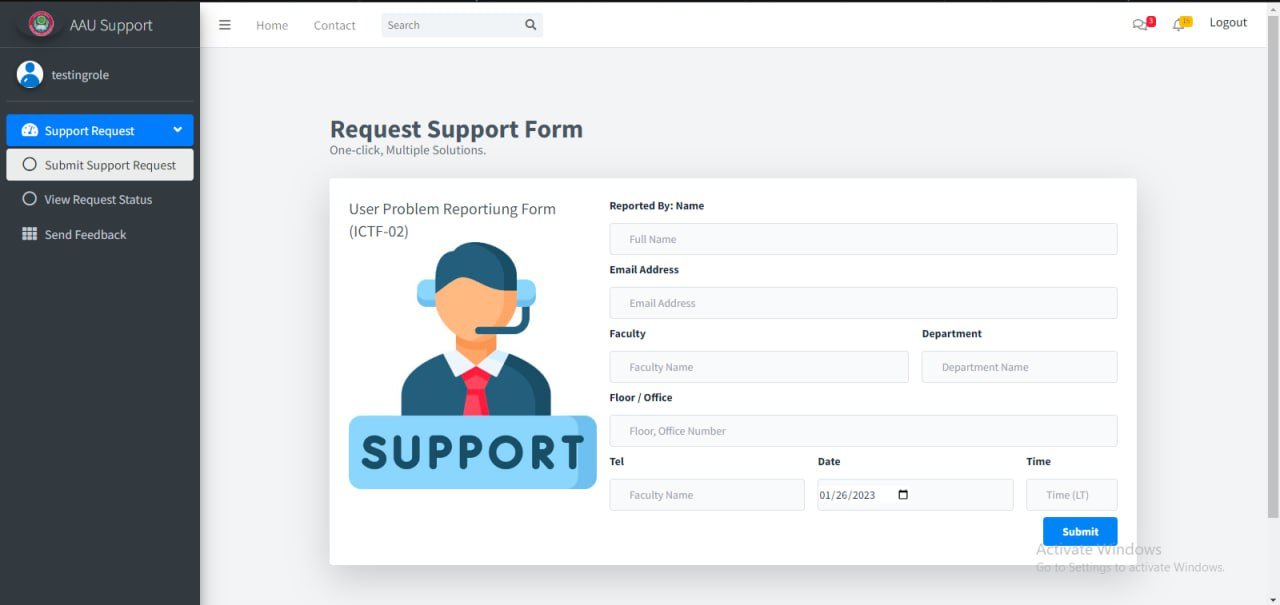


Figure 3. 22 Request support form

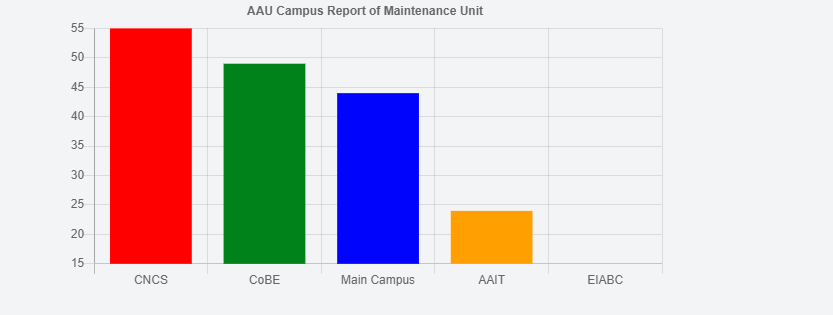


Figure 3. 23 View report

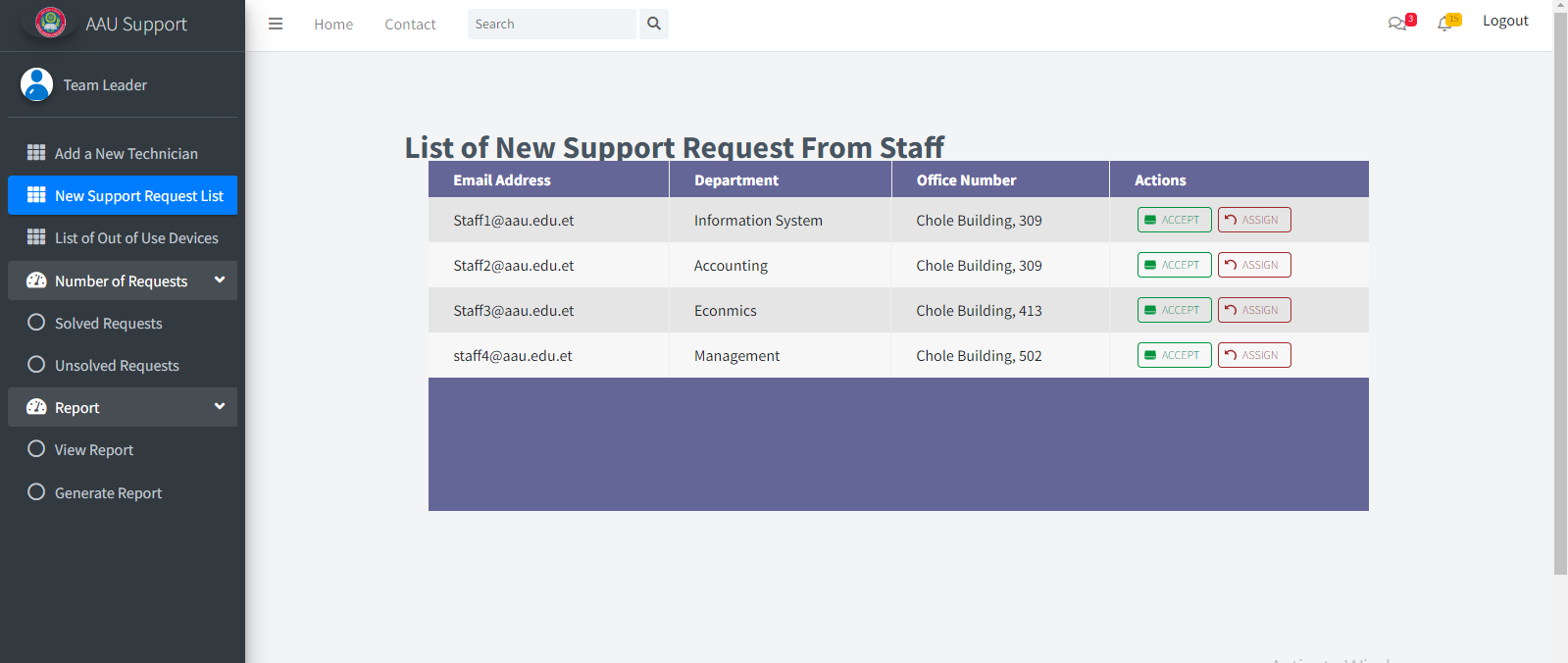


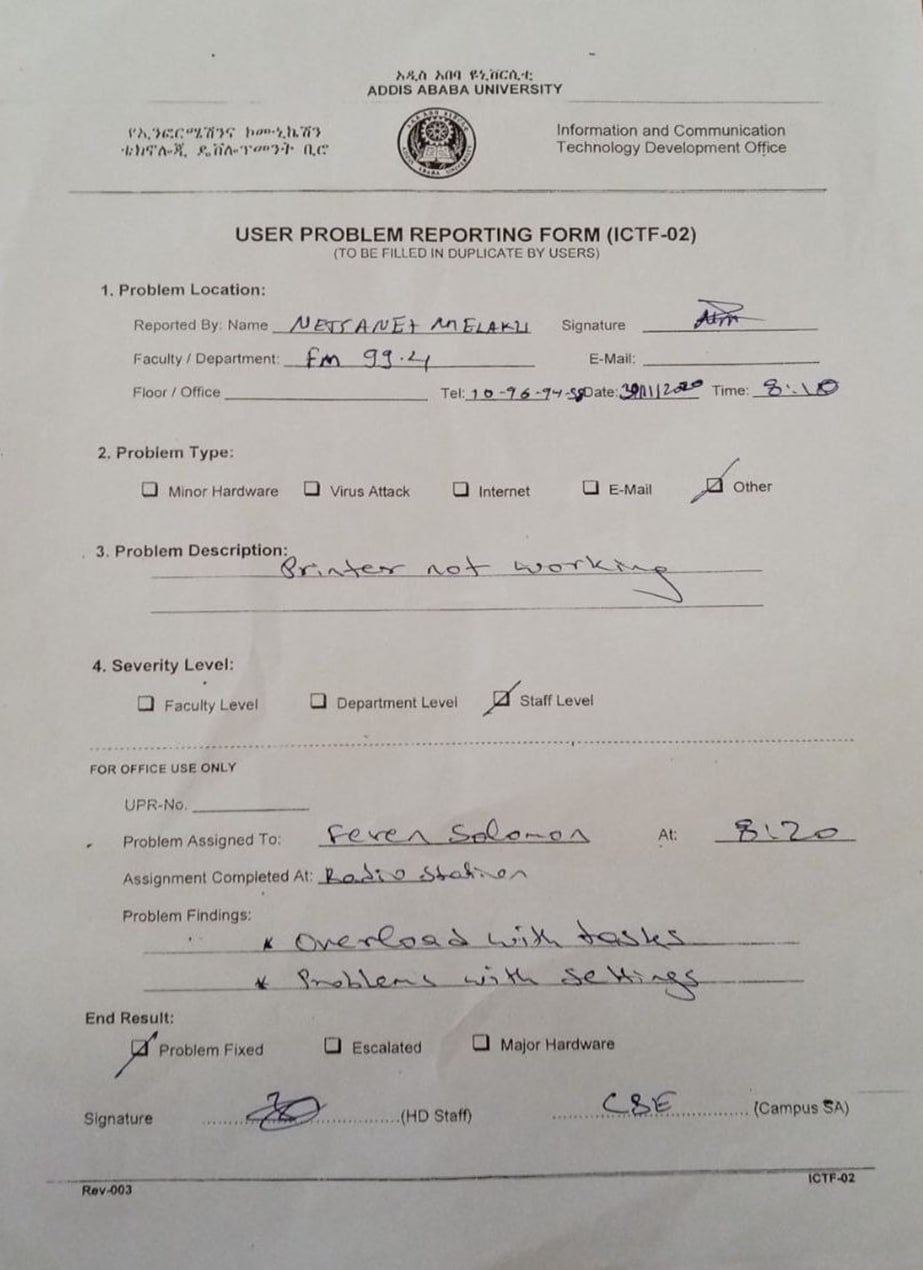
Figure 3. 24 List of new support request

## C:\Users\Lenovo\Documents\hw\AutoCAD Sheet Sets\Downloads\Telegram Desktop\Screenshot (415).png

Figure 3. 25 Generate report

## Appendix

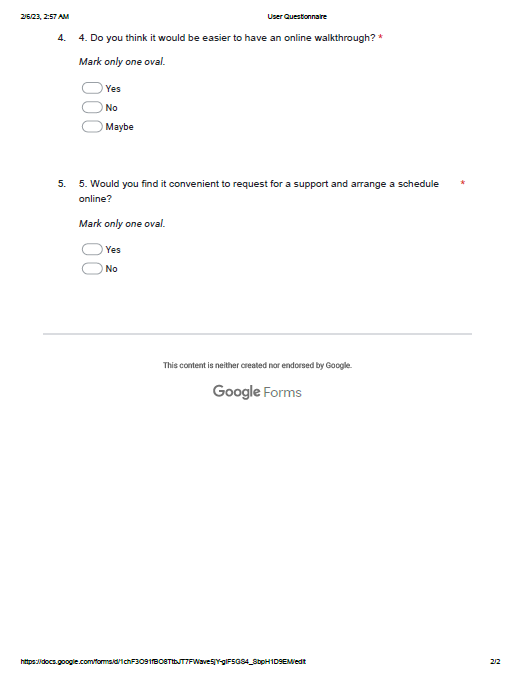
## Appendix 1 – Forms and report collected from organization



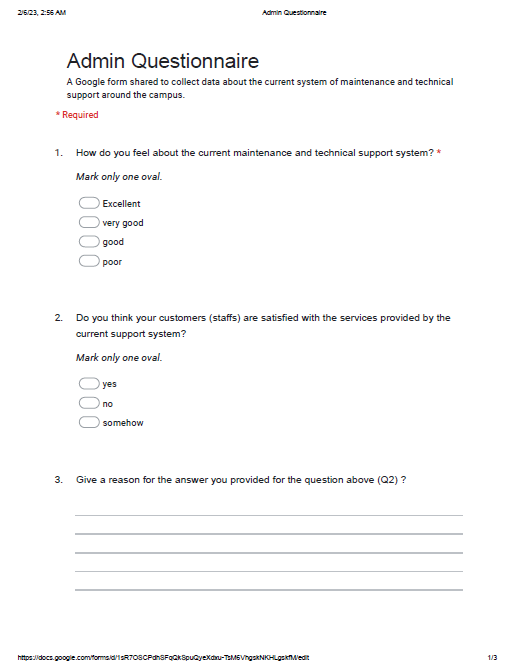
Appendix 1. 1User problem reporting form

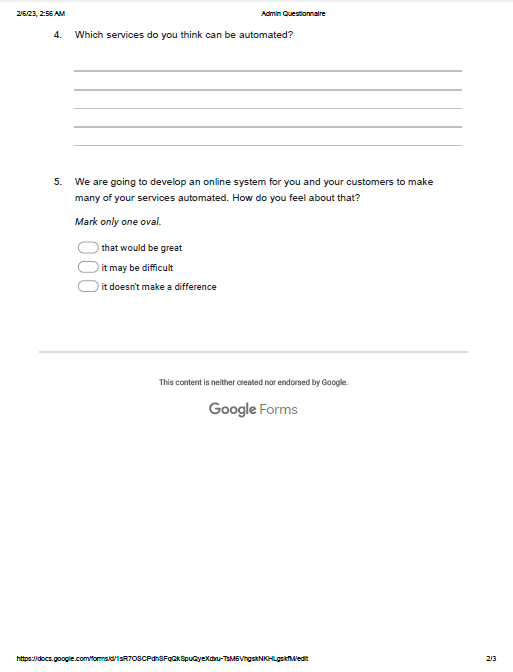
## Appendix 2-Interview Questionnaire used to collect information from technicians and team leaders





Appendix 2. 1 User Google form questionnaire





Appendix 2. 2 Admin Google form questionnaire

# Reference

Addis Ababa University. (2023). *Background*. Retrieved from aau.edu.et: http://www.aau.edu.et/offices/v\_president-office/office-of-the-academic-vise-president/continuing-and-distance-education-office/background/

Addis Ababa University. (2023). *Services*. Retrieved from aau.edu.et: http://www.aau.edu.et/services/ict/

Investopedia. (2023). *Feasibility study*. Retrieved from Investopedia.com: https://www.investopedia.com/terms/f/feasibility-study.asp

Simplilearn. (2023). *What is a Business Analysis and What does a Business Analyst Do*. Retrieved from Simplilearn.com: https://www.simplilearn.com/what-is-a-business-analyst-article

tutorials point. (2023). *User Interface* . Retrieved from tutorialspoint.com: https://www.tutorialspoint.com/software\_architecture\_design/user\_interface.htm

University, A. A. (2023, February 7). *AAU-ICT*. Retrieved December 15, 2022, from AAU portal: http://www.aau.edu.et/services/ict/overview/

w3schools. (2023). *CSS tutorial*. Retrieved from w3schools.com: https://www.w3schools.com/css/

w3schools. (2023). *HTML Tutorial*. Retrieved from w3schools.com: https://www.w3schools.com/html/

w3schools. (2023). *Javascript tutorial*. Retrieved from w3schools.com: https://www.w3schools.com/js/