CAaRT

(Configuration Assessment and Remediation Toolkit)

CAPT SAAD TARIQ
CAPT ESSA KHATTAK
CAPT AKIF SHER
CAPT OVAIS IRFAN KHAN

SUPERVISOR
MAJ WAJAHAT SULTAN

Submitted to the Faculty of Software Department
National University of Sciences and Technology, Islamabad
in partial fulfillment for the requirements of a B.E Degree in
Computer Software Engineering
June 2021

CERTIFICATE OF CORRECTION & APPROVAL

This is to officially state that the thesis work contained in this report titled ***"CAaRT" (Configuration Assessment and Remediation Toolkit)*** *Is carried out by:* ***Saad Tariq, Essa Khatak, Akif Sher, Ovais Irfan khan ,*** *under my supervision and that in my judgment, it is fully ample, in scope and excellence, for the degree of Bachelor of Computer Software Engineering from National University of Sciences and Technology (NUST).*

Approved By:

Signature: _____

Supervisor: Maj Wajahat Sultan
MCS, NUST Rawalpindi

DECLARATION OF ORIGINALITY

We hereby declare that the work contained in this report and the intellectual content of this report are the product of our work. This thesis report has not been formerly published in any structure nor does it include any verbatim of the published resources which could be treated as violation of the international copyright decree. We also affirm that we do recognize the terms ‘plagiarism’ and ‘copyright’ and that in case of any copyright infringement or plagiarism established in this thesis, we will be held fully accountable of the consequences of any such violation.

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached at the end of the document.

Signature: _____
Supervisor: *Maj Wajahat Sultan*

ACKNOWLEDGEMENTS

In the name of Allah, the most merciful and the most Beneficent, Who led us to this extent. May all glory, honor and Adoration be unto Thy Name.

Our special thanks go to our supervisor **Maj Wajahat Sultan** for guiding us throughout the process that resulted in the successful completion of our project. We would also like to thanks to the faculty of Software Department specially **Dr. Athar Mohsin Zaidi , Lec Mobeena Shehzad, Maj Zeeshan Zulkifl** for their guidance in the development of the project.

A deep gratitude towards **Dr. Adnan Ahmed Khan** (Head of Computer Software Department) for his guidance and facilitation for the Project.

At last, we are most obliged to our Parents, their support contributed immensely to the success of this project.

Dedicated to our exceptional parents and adored siblings whose tremendous support and cooperation led us to this wonderful accomplishment.

ABSTRACT

It is widely accepted in most organizations that threats from trusted insiders pose a significant risk to the organization and are exceedingly difficult to defend against. Auditing is a widely accepted technique for detecting malicious activity on computer networks. In comparison, currently available auditing methods are typically applied uniformly and may not be an appropriate strategy for mitigating the insider threat in all circumstances.

We present a management information system auditing system, dubbed CIS-Audit, in this project. CIS-Audit is intended to aid in the auditing process of management information systems. It is defined by two primary characteristics:

1. It covers all facets of the Windows Operating System, both administrative and technical.
2. It will be possible to update and configure the Audit in real time.

Table of Contents

List of Tables

List of Figures

1 CHAPTER 1: INTRODUCTION

On a worldwide scale, information technology has revolutionized and significantly impacted the production processes today. Computerization has a substantial impact on organizational control, information flow inside documents, and other aspects of business operations. Though auditing in a CIS environment has had no effect on the core essence of auditing, the method of evidence gathering and evaluation has undergone significant transformation. Additionally, auditors must establish a working knowledge of computer settings and stay current on quickly evolving technology, including the use of sophisticated audit tools.

Vulnerabilities in privacy logs and monitoring progressively increasing to hide their location, malicious software, and target systems. Even though the victims are aware of penetration of their systems, they are not aware of the nature of the attack and the action taken by the attackers in the absence of secure, complete log data. Without adequate audit logs, an attack may remain undiscovered for an extended period of time, causing irreversible damage.

At times, the sole sign that an assault was effective is logged data. Although many firms maintain audit logs for regulatory compliance reasons; nonetheless, hackers take advantage of the fact that many organizations do not regularly review their audit logs and are therefore unaware of an intrusion into their systems. Due to inadequate or non-existent log analysis processes, attackers are occasionally able to maintain control of victim machines for months or even years without the target firm being aware, despite proof of the attack being documented in unopened log files.

1.1 Purpose

The goal of this thesis is to provide enough detail about a system's design to enable software development to proceed with a clear understanding of what will be built and

how it will be built. This document contains critical information about the software and the auditing process that will be used to create the system.

1.2 Scope

Scope of the Project is a Web Application that is applicable to Windows environments above 2016 and Windows Servers. Our goal is to provide the Automated Auditing of the platform within minimum possible time, cost and reliability.
Following are the benefits of the Project:

- Real-Time Audits.
- Low clerical error.
- Concentration of duties.
- Ability to investigate logs.
- Exception reporting.
- Low Cost.
- Disappearance of manual reasonableness.
- Shifting of control base.

1.3 Overview

The thesis is organized into eight sections, each of which has a number of subsections. It is divided into the following sections:

- Introduction
- CIS Standard
- System Overview
- System Architecture
- Data Design
- Component Design
- Human Interface Design
- Requirement Matrix
- Appendices

1.4 Definitions

1.4.1 Real-Time Audits

Information can be created relatively quickly in a CIS setting. Even complex reports in a specified report format can be prepared quickly for audit reasons. This saves time, allowing the auditor to widen their analytical assessment for under coverage. Additionally, the auditor can expand their substantive procedures to collect additional evidence to support their judgement.

1.4.2 Auditing

Auditing is a widely available resource that can be used to protect a company from the insider threat. To begin, it must be comprehended, and its possibilities examined through an examination of its history, varied functions, and purposes.

1.4.3 Auditing Policy

In order to effectively employ auditing to meet security objectives, it is necessary to build an auditing strategy that is thorough and well thought out. The foundation for developing an effective auditing policy is a strong security policy; in this software, we are utilizing the CIS Auditing Policy.

1.4.4 Low clerical error

Low clerical error refers to a systematic and sequentially scheduled course of action in which the chances of committing an error are significantly decreased, compared to the alternative.

1.4.5 Black Box Auditing

The Black Box methodology, known as computer-based auditing, is an audit method in which the auditor concentrates on input and output, ignoring technological aspects of the processing of data or transactions by computers. If the input and output are

identical, the auditor concludes that the transaction/data processing occurred as intended:

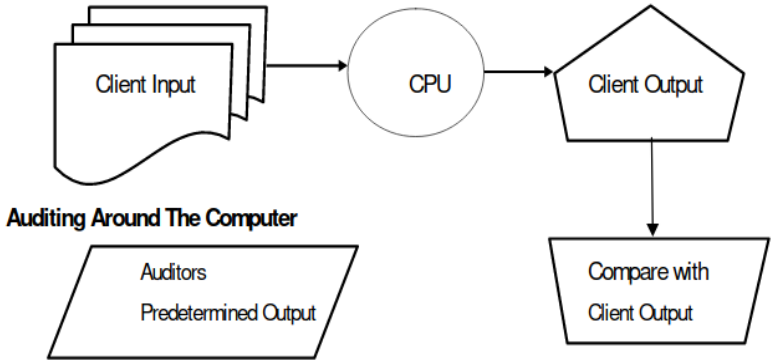


Figure 1: Black Box Auditing

1.4.6 White Box Auditing

Not only are the processes and controls around the subject audited, but also the processing controls that operate over this process are examined. Computer audit software can be used as stated below in order to enable the auditor to access these processes:

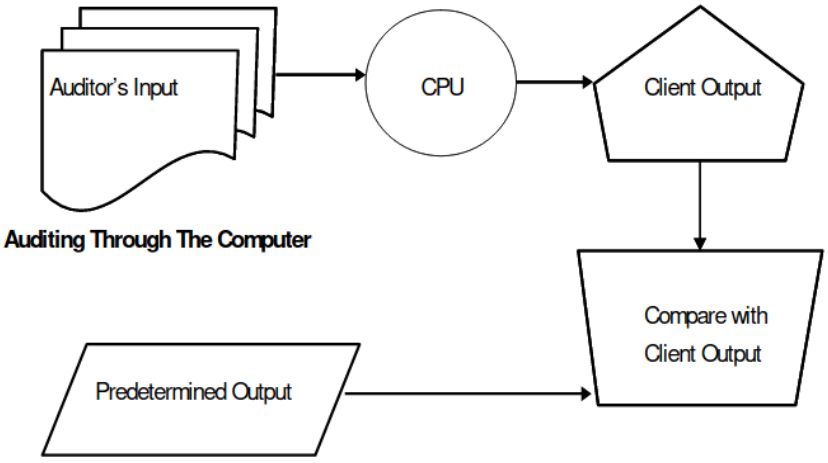


Figure 2: White Box Auditing

1.4.7 Impact of Poor System

When systems and designs dropped below the projected performance standards, the integrated corporate operation can be harmed more than good through a computerized information system environment.

1.4.8 Insider Threat

The threat inside is often misunderstood and disregarded. It damages an organization severely and requires attention and awareness to be safeguarded and mitigated.

2 CHAPTER 2: System Overview

2.1 Background

Because infrastructure serves as the foundation for your entire business, it is only intuitive that an infrastructure audit considers the big picture. If your infrastructure has an impact on your email, applications, timesheets, communication with employees, and anything else, the audit should have an impact on those things as well, as well.

Unfortunately, in practice, this does not always work as intended. Infrastructure works when it is removed from its silo, but not everyone has adapted to a DevOps mindset as a result of our experience. As a result, so many infrastructure audits are focused on finding quick fixes rather than learning why things are the way they are or how to make them better. Infrastructure audits to help you get the most out of your organization. We want to position our partners for long-term success, and we are willing to put in the necessary effort to do so correctly. When we arrive, we'll begin by gaining an understanding of how your system operates and what it is required to do. Afterwards, we will recommend changes to your organization's architecture and software tools, as well as improvements in culture and communication among your teams. Too often, the divisions between your IT staff and your developers result in a lack of collaboration on potentially game-changing ideas. By evaluating your organization from the front-facing applications all the way back to the bare bones of your infrastructure, you can make purposeful, deliberate decisions that will provide you with the results you want today as well as the capacity for future expansion.

2.1.1 CIS Standard Basics

CIS Controls and CIS Benchmarks provide global standards for internet security and are widely recognized as global standards and best practices for protecting information

technology systems and data against attacks. CIS Benchmarks provide frameworks to assist organizations in enhancing their security and are developed through an independent consensus process.

2.1.2 Internal Controls of CIS environment

It is vital for any system to have internal control in order for it to be handled properly and successfully. The policies and methods devised by management to achieve specific firm objectives, such as physical verification of assets, periodic review and reconciliation of accounts, and special control over computer-generated data, are, in essence, what they are. When we say, "internal control," we are referring to a computerized information system that operates on the same principles as a manual system. As a result, the organizational structure, delegation of power, system authorization, and assignment of responsibilities are all defined in a manner similar to how they are determined in a traditional manual system. However, because of the changes in approach, there are a range of additional types of controls that are particularly specific to the CIS environment that must be considered in a CIS context. It is necessary to break down internal control systems in a CIS environment into specific subsystems and build controls that address each function independently in order for auditors to be able to rely on them. The following are the primary control classes that this software will need to examine:

- ***Authenticity Controls*** - Authenticity control is used to ensure that the individuals or processes involved in a system are who they claim to be before they can be trusted.
- ***Accuracy Control*** - Accuracy control is the process of ensuring that data and processes in a system are correct.
- ***Completeness Control*** - Controlling for completeness seeks to verify that no data is missing and that all processing is completed to the appropriate conclusion.

- ***Redundancy Control*** - Attempts are made through redundancy measures to ensure that data is processed only once.
- ***Privacy Controls*** - In order to prevent personal information from being disclosed mistakenly or without authorization, privacy rules must be in place.
- ***Audit Trail Controls*** - Audit trail control assures that all events occurring in a system may be tracked back to their source. This record is required in order to respond to inquiries, comply with legal obligations, minimize irregularities, detect the repercussions of errors, and so on.
- ***Existence Controls*** - Existing controls make an attempt to ensure that all system resources remain available on an ongoing basis.
- ***Asset Safeguarding Controls*** - The goal of asset safeguarding control is to protect all resources included within a system from being destroyed or corrupted in any way.
- ***Effectiveness Controls*** - The goal of effectiveness control is to make certain that systems fulfil their objectives.

2.2 System Analysis

2.2.1 Functional Requirements

2.2.1.1 Dashboard Access

This web Application provides access to the auditing dashboard which:

- Encapsulates the latest auditing progress.
- List of previous audits.
- A Table describing which tests have passed and failed.
- Create a new Auditing Process.

2.2.1.2 Auditing Services

Users can access the Auditing Services in real-time. We propose the application must provide below functionalities:

- Save Audits
- Create new Audits
- Audit Analytics
- Multiple Audits

2.2.1.3 Mitigation Services

User can mitigate or hardened the policies on host computer that we authenticated and provided some.

2.2.1.4 Backup Services

- Users can Store log data in the cloud.
- Users can access/ Delete / replicate this data anytime.

2.2.2 Non-Functional Requirements

- Reactive Components and UI
- Material Design for Themes and Styling.
- Redux Store Management.
- React Router for Client-Side Routing.
- Django CSRF Security
- Python Client Authentication
- Python Client Token Sessions.

3 Chapter 3: System Architecture

3.1 Architectural Context

The entire Application is divided into **two-tier** architecture with Django Server application communicating with the client server using REST(GET, PUT, POST, DELETE) calls that are bi-directional and the second element of the whole infrastructure is a python program, whose only job is to communicate, audit and apply mitigation to a host OS, it is packaged as an .msi or .exe file written in python, to provide project consistencies.

3.1.1 Python Client Program

This program is compiled and updated in a separate codebase and provided to the user using the file server of the Django as an msi or exe. This program has three jobs to perform:

1. Communicate with the server
2. Audit the host OS.
3. Apply mitigation on the OS

The Program is consistently performing any job above using asynchronous Python calls, which means it doesn't wait for audit to end to start applying mitigation or communicate with the server which boosts the performance of the application exponentially. These the steps that the program took first:

1. Check for APPDATA directory to check for cache log if software is already installed.
2. If there are no log files searched then the program will run using default configuration and authenticate the user with username and password, if successful, the program will start auditing the HOST OS.
3. send the results to the server asynchronously using a stream of packets called socket streams.
4. If the user asks the server to start applying mitigation, the client will asynchronously start applying mitigation to the policies which are identified in the auditing process even if the audit is not ended.

5. If audited and mitigation are ended, close the program.

3.1.2 Django Server

The headless Django Server is used to display templates, views and save workspace, roles, of the users' data, we use Server-Side rendering for security, performance.

3.1.2.1 Why we use the Django framework

3.1.2.1.1 Simple and efficient

One of Django's primary objectives is to make the work of developers easier. Specifically, the Django framework employs: Rapid development principles, which allow developers to complete more than one iteration at a time without having to restart the entire schedule from the beginning; DRY philosophy (Don't Repeat Yourself), which allows developers to reuse existing code and concentrate on the unique one.

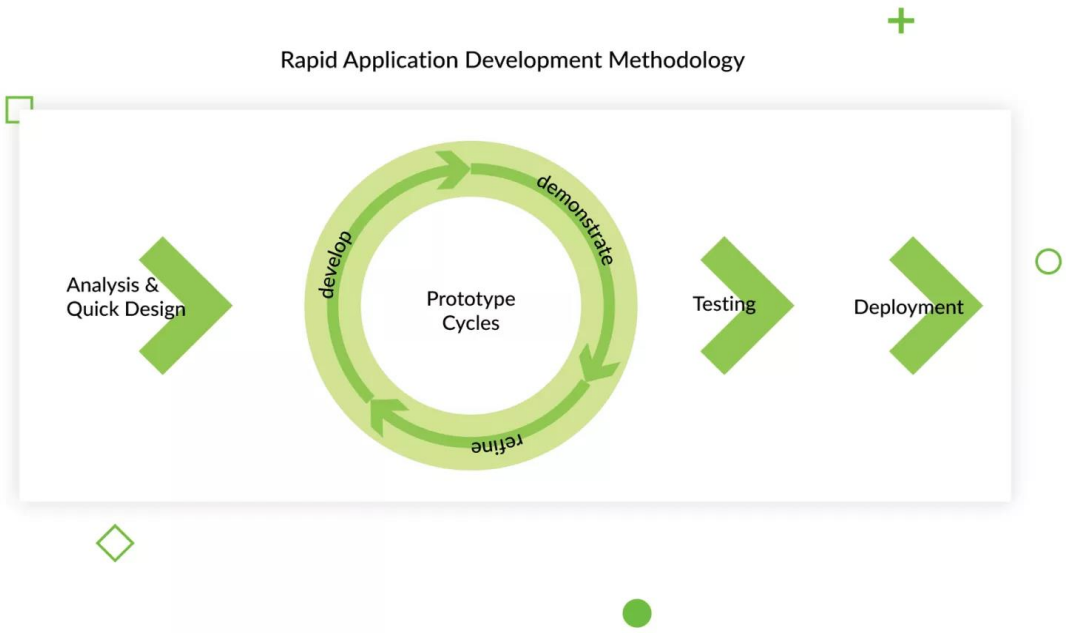


Figure 3: Rapid Application Development Methodology

3.1.2.1.2 Provides security:

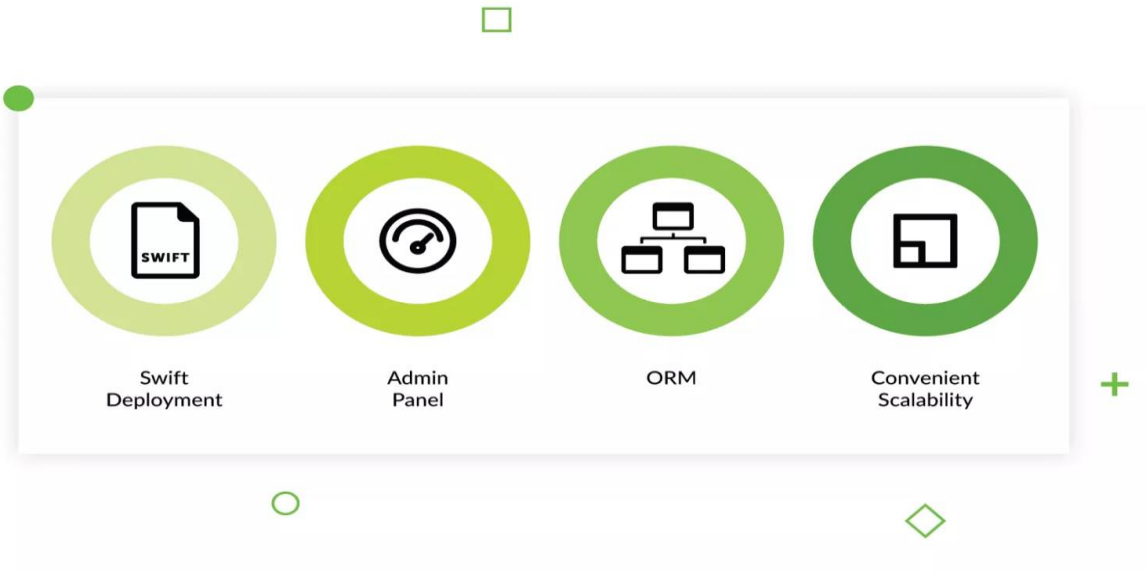
Django places a great value on the security of its users. It offers one of the greatest out-of-the-box security systems available, and it aids developers in avoiding typical security issues like as clickjacking, cross-site scripting, and SQL injection, among others. SQL injection is a type of attack. Django provides new security updates on a regular basis. It is typically the first to respond to vulnerabilities and notify other frameworks of the problem.

3.1.2.1.3 Well-established:

Django has been tried and true over time and by a large number of people. It features a large and supportive community, which can be accessed through a variety of forums, channels, and dedicated websites. When there is an issue with a function in the code, it is simple to obtain assistance, and finding developers is simple if your organization wants to use Django as the foundation for its next project.

Django got off to a terrific start, with some of the best documentation available for any open-source framework. And it is still maintained at a high level, with new features and bug fixes being added on a regular basis, allowing you to readily adjust to changes.

You can rest assured that any faults with the framework will be addressed as soon as they arise. a. new packages are released to make working with Django even more convenient than it already is. The software is continually being updated.



3.2 Architecture Diagram

1. Django Server fulfills requests by sending responses back to client.
2. Client authenticate with the server and send/receive instructions.

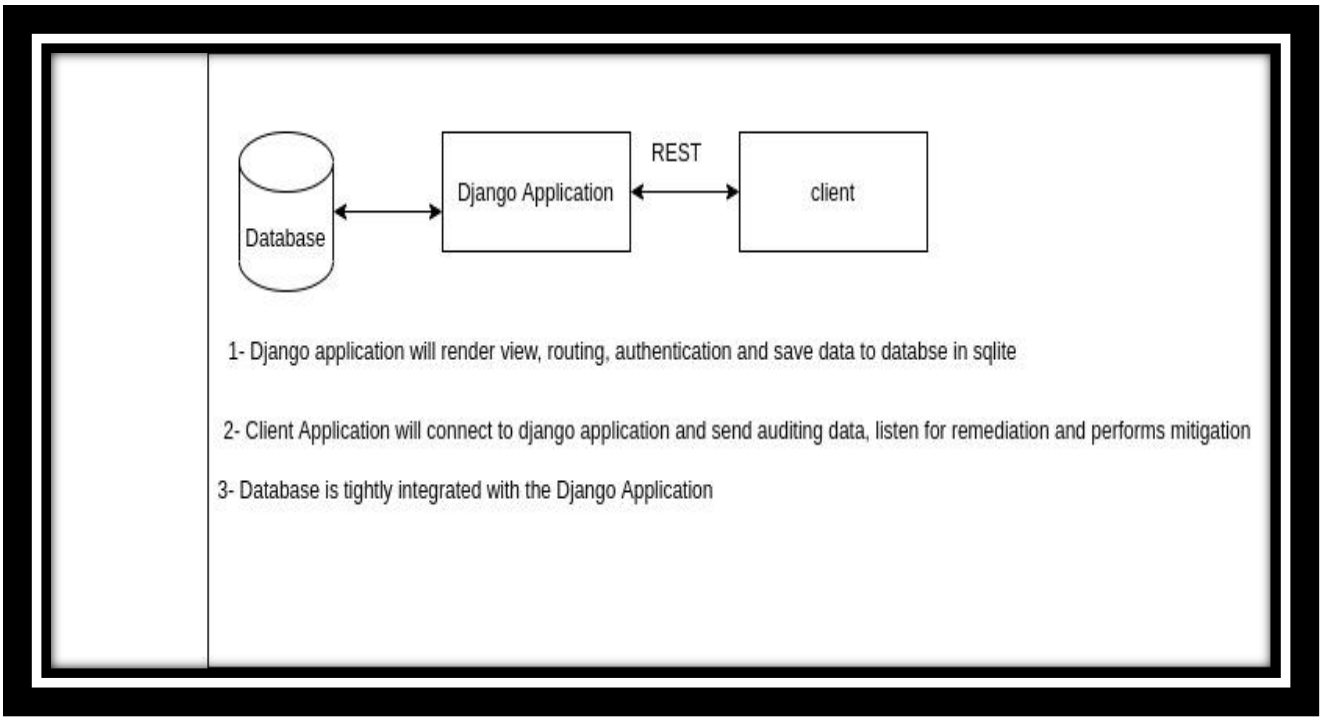


Figure 4: Architecture Diagram

3.3 Decomposition Description

3.3.1 Use Case Diagram

Use Case diagrams are divided into the three actor elements i.e Actor, Server, Client described above in detail, which performs functional jobs that are expected of these elements.

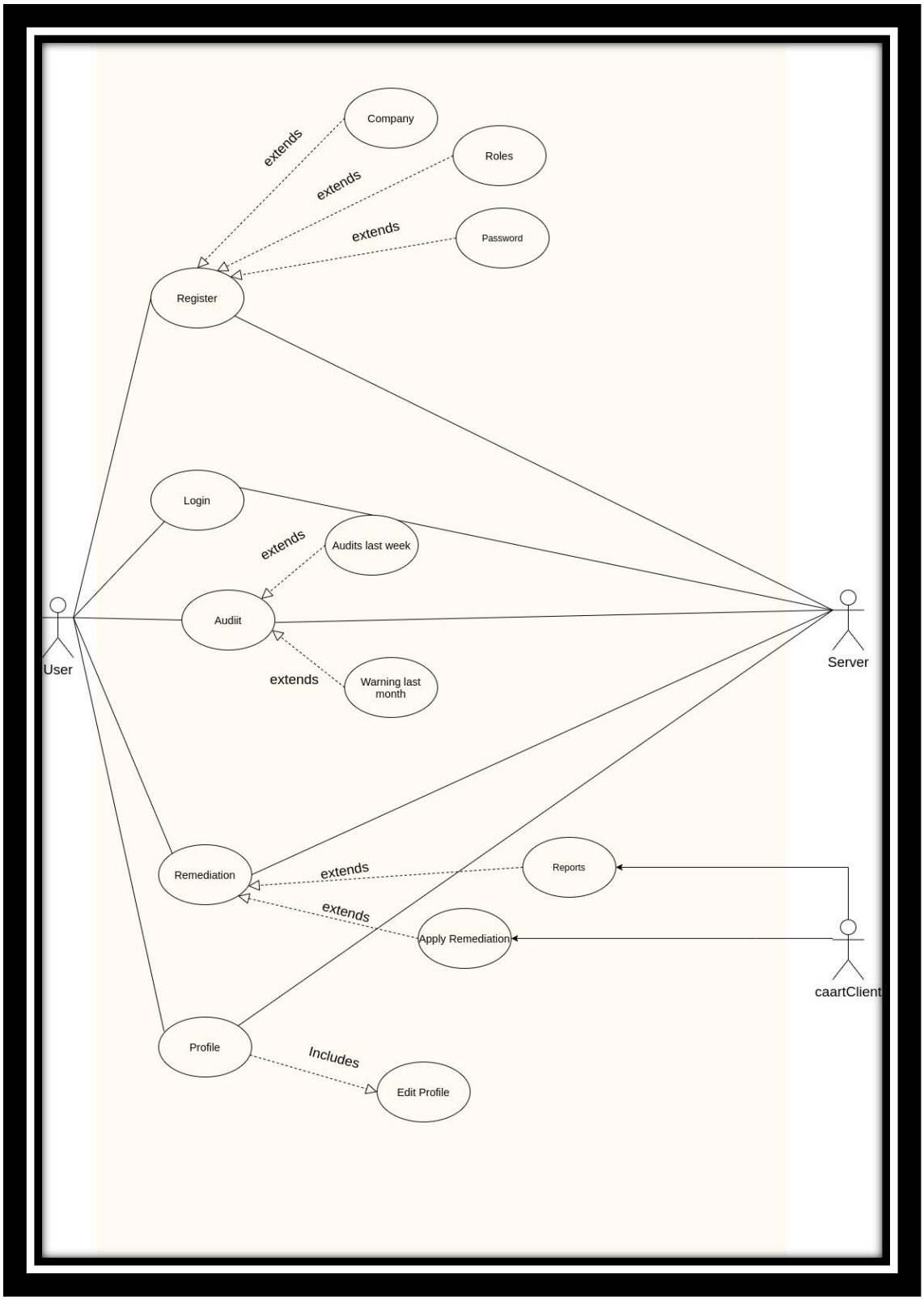


Figure 5: Use Case Diagram

3.3.2 Use Case UC1: Sign Up

Name: Sign up

Scope: CIS-Audit

Primary Actor: User

Description: User will load the webpage and click on the Sign-Up button to sign up.

Stakeholders and Interests: User must sign up to register their account to access the services

Pre-Condition: Must have web application for registration

Success Guarantee: Unregistered users shall get registered after they provide the valid information and store it in a database with a unique id.

User:	System:
Step 1: User requests to create an account by clicking the Sign-Up button	Step 2: The system shall show the Sign-Up form.
Step 3: User enters the details e.g. first name, last name, Email, Password.	Step 4: The System checks authenticity of the values and if the user is already registered or not, if not the system shall redirect to the dashboard
	Step 5: System create unique user id with the provided details and save the data to the database.

Table 1: Use Case - Sign Up

Extension

In extension we explain the alternative scenarios of use cases.

Alternative Flow 1: Username/email already exists

- System displays a message that the email already exists.
- System asks the user to enter another username/email.

Alternative Flow 2: Invalid email ID

- System displays a message that the email is invalid.
- System asks the user to enter a valid email id.

Alternative Flow 3: Mismatched passwords

- System displays a message that the password does not match.
- System asks the user to enter the password again.

Alternative Flow 4: Invalid username/password/email format

- System displays a message that the format is invalid.

3.3.3 Use Case UC2: Log In

Name: Log In

Scope: CIS Audit

Primary Actor: Registered User

Description: First, the user will open the webpage and click on the Signup/Login Button.

Stakeholders and Interests: Unregister user: User must sign up to register their account.

Preconditions: Users have already registered.

Success Guarantee (Post conditions): Registered users will be granted access to the system.

User:	System:
Step 1: Click the login Button	Step 2: System Show the login Screen
Step 3: Enter the Login Credentials	Step 4: System will authenticate the credentials, if valid, authorization is given.

Table 2: Use Case - Login

3.3.4 Use Case UC3: Create Company

Name: Create Company

Scope: CIS Audit

Primary Actor: User

Description: First, the user will register to the system, then login to the system using correct credentials.

Stakeholders and Interests: Users will be able to create companies to handle multiple audits at once.

Preconditions: Must be Registered to the System and logged In the System.

Success Guarantee (Post conditions): User will be Able to create a company.

User:	System:
	Step 1. System will show the company form page.
Step 2: User will enter the required form elements as, Company name, Postal Address, Client Operating System.	Step 3: The System will save the data to the database and redirect to the dashboard.

Table 3: Use Case - Create Company

3.3.5 Use Case UC4: Create Roles

Name: Create Roles

Scope: CIS-Audit

Primary Actor: User

Description: First, the user will register to the system, then login to the system using correct credentials, then create the Company and then create a User or role if needed.

Stakeholders and Interests: Users would be able to create users and define their role in the auditing process.

Preconditions: User must have registered and logged In to the System.

Success Guarantee (Post conditions): Users would be able to create users and define their role in the auditing process.

User:	System:
	Step 1: The System will show the Create Role Screen form.
Step 2: User will fill the form with predefined input values for Roles and a name of User	Step 3: The system will save the form to the database and shall redirect to the main dashboard.

Table 4: Use Case - Create Roles

3.3.6 Use Case UC5: Create Stream

Name: Create Stream

Scope: CIS-Audit

Primary Actor: User

Description: First, the user will register to the system, then login to the system using correct credentials and create a Company with a single user and its role.

Stakeholders and Interests: Users can create Audits.

Preconditions: User must be registered and logged In the System.

Success Guarantee (Post conditions): Create Can Perform CRUD operation with Auditing.

User:	System:
	Step 1: The System will show the User a dashboard through which the user can Perform CRUD Operation.
Step 2: Users can click on the Create Stream.	Step 3: The System will Show a popover banner to download the client program.
Step 4: The user Download and run the Client program	Step 5: The system connects to client program

Table 5: Use Case – Create Stream

Alternative Flow 1: Client Download Failed

- Server will open the connection for half an hour after that it would be assumed failed.
- A client can Create a new Stream.

Alternative Flow 2: Client Couldn’t connect to Server

- Server will send a network error response to the System (ERR CODE > 500)
- The Client Program will exit.
- Clients can create a new Stream.

3.3.7 Use Case UC6: Observe Stream

Name: Observe Stream
Scope: CIS-Audit
Primary Actor: Client Program
Description: Client Program is installed and authenticated on the user Device, then Client program will start the audit and send Stream of data to Server, which is parsed, saved then redirected to user dashboard in real time.

Stakeholders and Interests: Unregister user: User must sign up to register their account.

Preconditions: User should be registered.
Success Guarantee (Post conditions): Successful transmission of data.

Client Program	Server:
Step 1: Client will start the audit process.	Step 2: System will send a response to the client to verify if it is running or not.
Step 3: Client program will send a response of 200 Status code upon successful authorization.	Step 4: System will listen for data change events async.

Table 6: Use Case – Observe Stream

Alternative Flow 1: Authorization Failed

- Server will send an error signal to the System.
- The Client Program will exit.

Alternative Flow 2: Failure of network connection

- Server will send a network error response to the server.
- The Client Program will exit.

3.3.8 Use Case UC7: Stop Stream

Name: Stop Stream

Scope: CIS-Audit

Primary Actor: Client Program

Description: Client Program is installed and authenticated on the user Device, then Client program will start the audit and send Stream of data to Server, which is parsed, saved then redirected to user dashboard in real time.

Stakeholders and Interests: Unregister user: User must sign up to register their account.

Preconditions: User should be registered.

Success Guarantee (Post conditions): Successful transmission of data

Client Program	Server:
Step 1: Client will start the audit process.	Step 2: System will send a response to client to verify if it is running or not.
Step 3: Client program will send a response of 200 Status code upon successful authorization.	Step 4: System will listen for data change events async.

Table 7: Use Case – Stop Stream

Alternative Flow 1: Authorization Failed

- Server will send an error signal to the System.
- Client Program will exit.

Alternative Flow 2: Failure of network connection

- Server will send a network error response to the server.
- Client Program will exit.

3.3.9 Use Case UC8: Invite Members

Name: Invite Members
Scope: CIS-Audit
Primary Actor: Registered User, Unregistered External User, web portal
Description: Client can send invites as IAM users to External users.

Stakeholders and Interests: Unregister user: User must sign up to register their account.

Preconditions: User should be registered.
Success Guarantee (Post conditions): Successful invitation of IAM users.

Client Program	Server:
Step 1: Client will start the audit process.	Step 2: Client will send email invites to IAM users.
Step 3: External Users will follow the link and register themselves on the platform.	Step 4: Registered External Users can now manage the infrastructure of auditing.

Table 8: Use Case – Invite Members

Alternative Flow 1: Authorization Failed

- Server will send an error signal to the System.
- The Client Program will exit.

Alternative Flow 2: Failure of network connection

- Server will send a network error response to the server.
- The Client Program will exit.

3.3.10 Use Case UC9: Apply Remediation

Name: Apply Remediations

Scope: CIS-Audit

Primary Actor: Client, Host OS, Server

Description: Sever will message the client program to start applying remediations.

Stakeholders and Interests: Unregister user or User should be logged in the client program.

Preconditions: User should be authenticated.

Success Guarantee (Post conditions): Successful completion of the remediation

Client Program	Server:
Step 1: User asks the server to start remediation.	Step 2: Server will send a message to client program to start remediation service.
Step 3: Client Program will start remediation service and send results to server.	Step 4: Server will send received messages to User.

Table 9: Use Case – Apply Remediation

Alternative Flow 1: Authorization Failed

- Server will send an error signal to the System.
- The Client Program will exit.

Alternative Flow 2: Failure of network connection

- Client will send a network error response to the server.
- The Client Program will exit.

3.4 Activity Diagram

1. Frontend in below diagram denotes Django views or templating engine.
2. Backend in the diagram below denotes Django core functionality.
3. Server denotes a tightly integrated SQLite database.

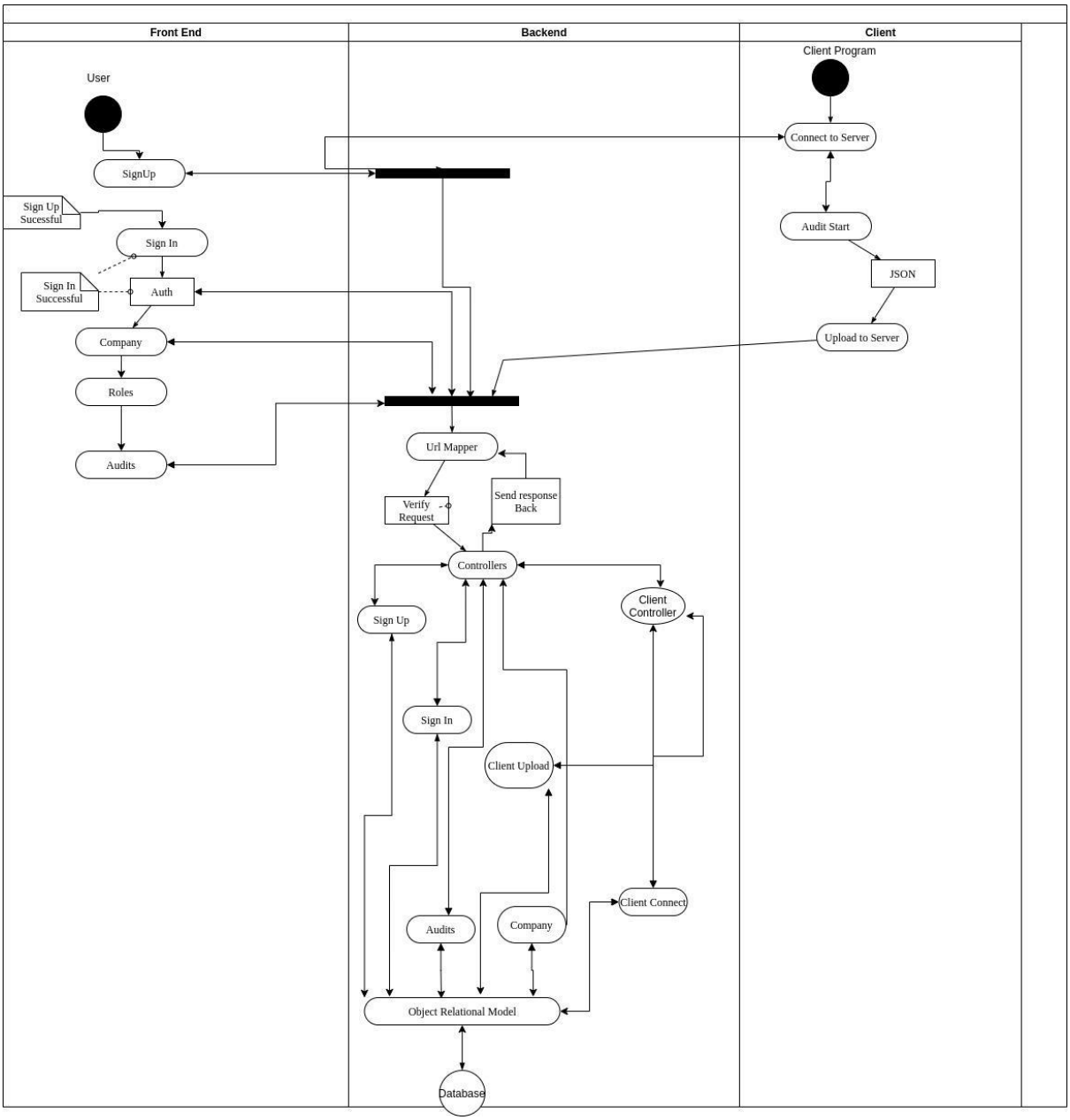


Figure 6: Activity Diagram

3.5 Database Diagram

1. Company is composed of users and a set of users can have many roles.
2. A user can have only one audit at a time.
3. and many users can create many audits.

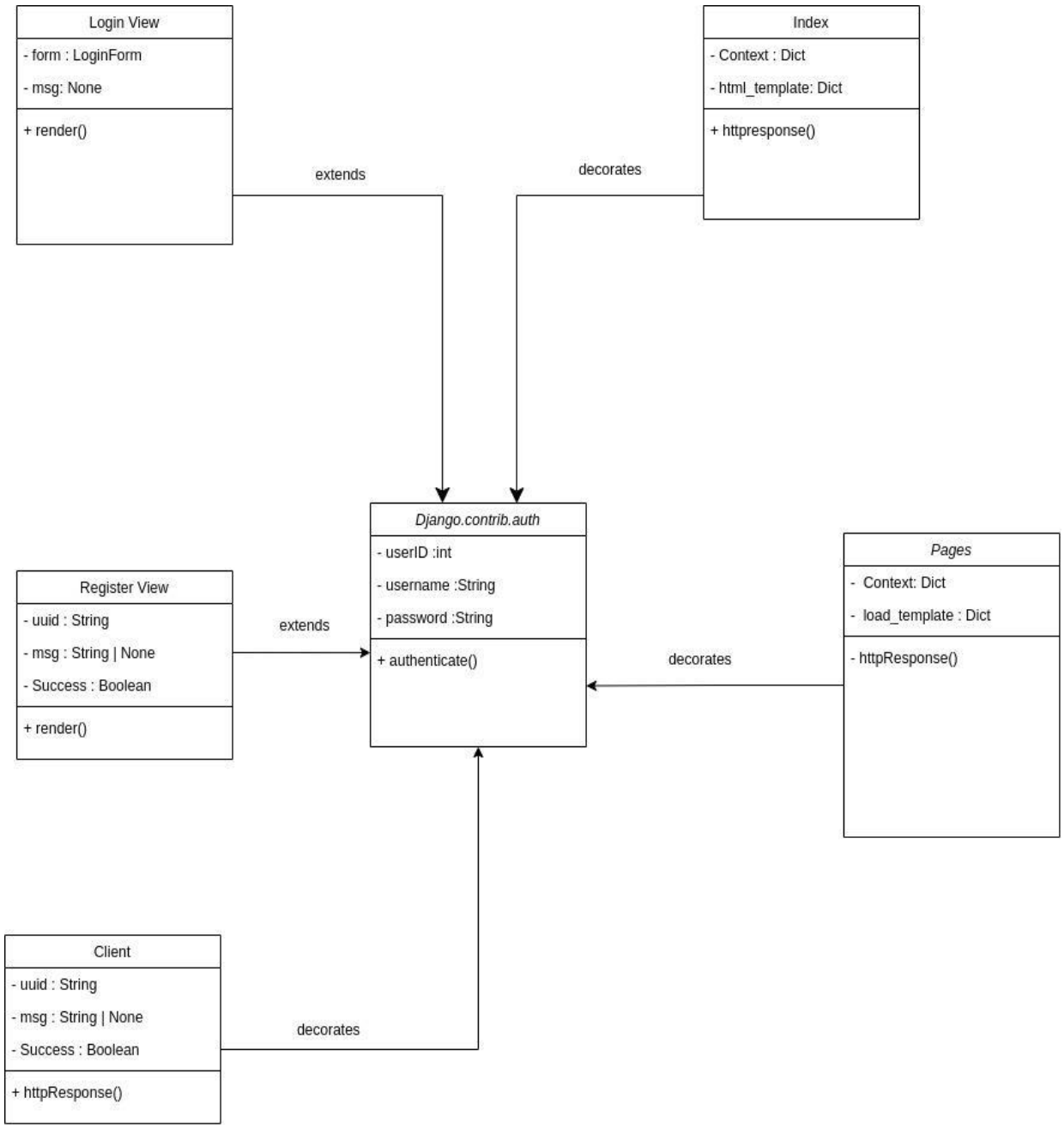


Figure 7: Database Diagram

3.6 Sequence Diagram

3.6.1 Sign Up

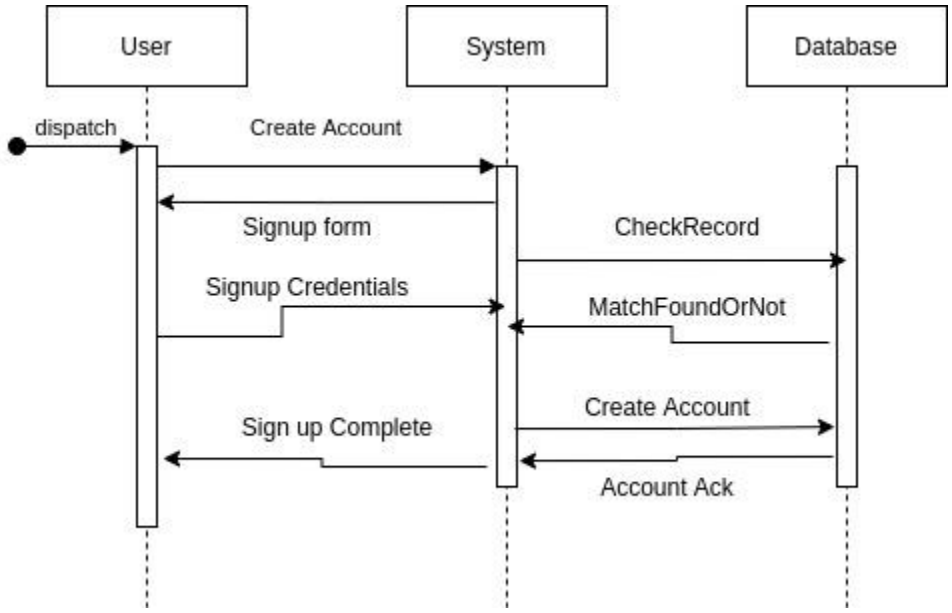


Figure 8: Sequence Diagram Signup

3.6.2 Login

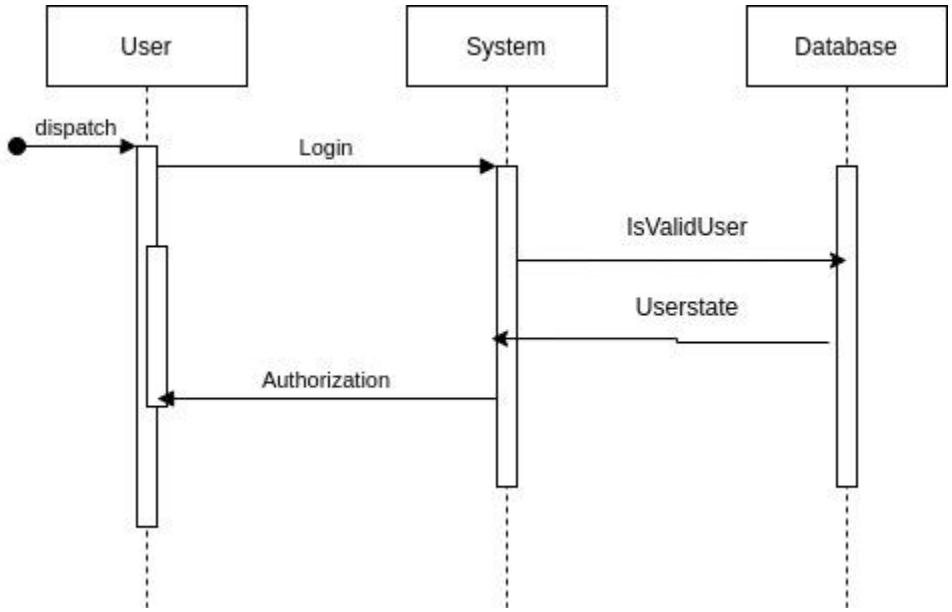


Figure 9: Sequence Diagram Login

3.6.3 Creating Company

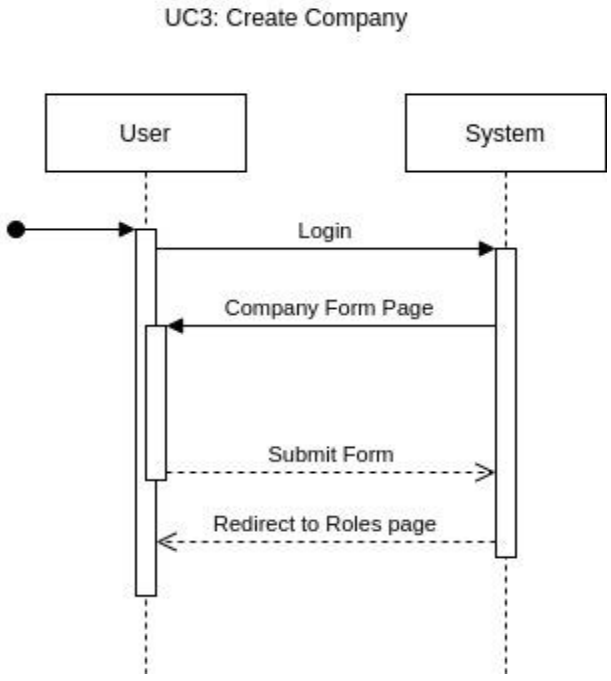


Figure 10: Sequence Diagram Creating Company

3.6.4 Creating Roles

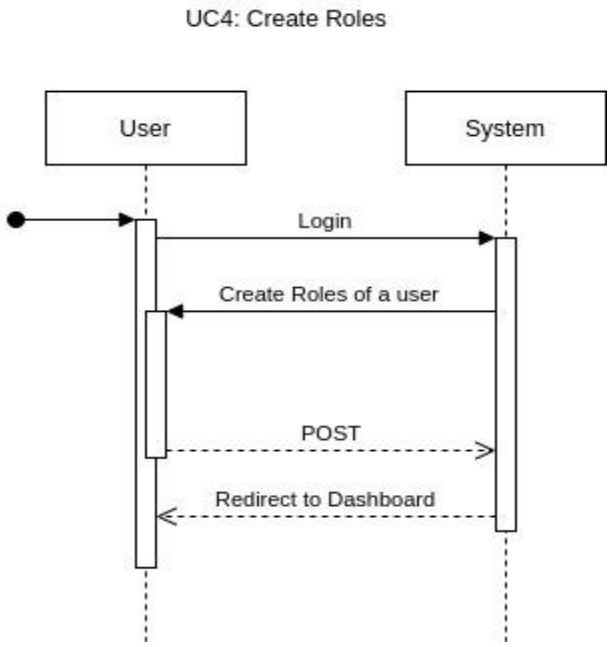


Figure 11: Sequence Diagram Creating Roles

3.6.5 Observe Stream

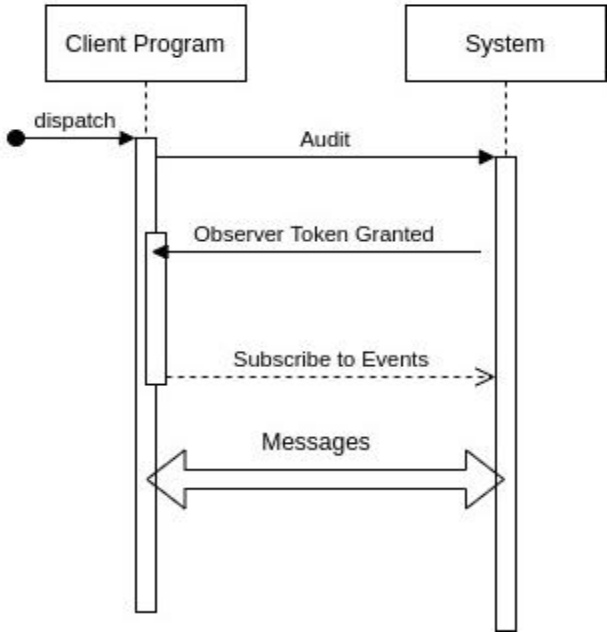


Figure 12: Sequence Diagram Observe Stream

3.6.6 Stop Stream

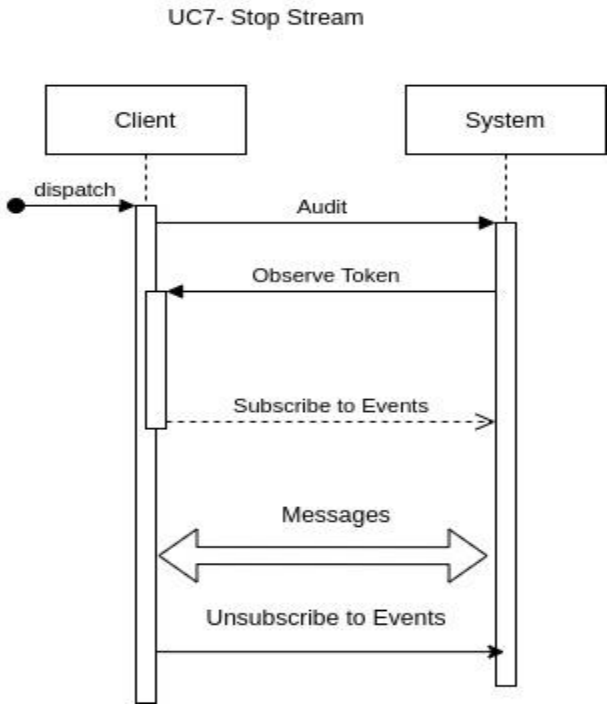


Figure 13: Sequence Diagram Stop Stream

3.6.7 Invite IAM members

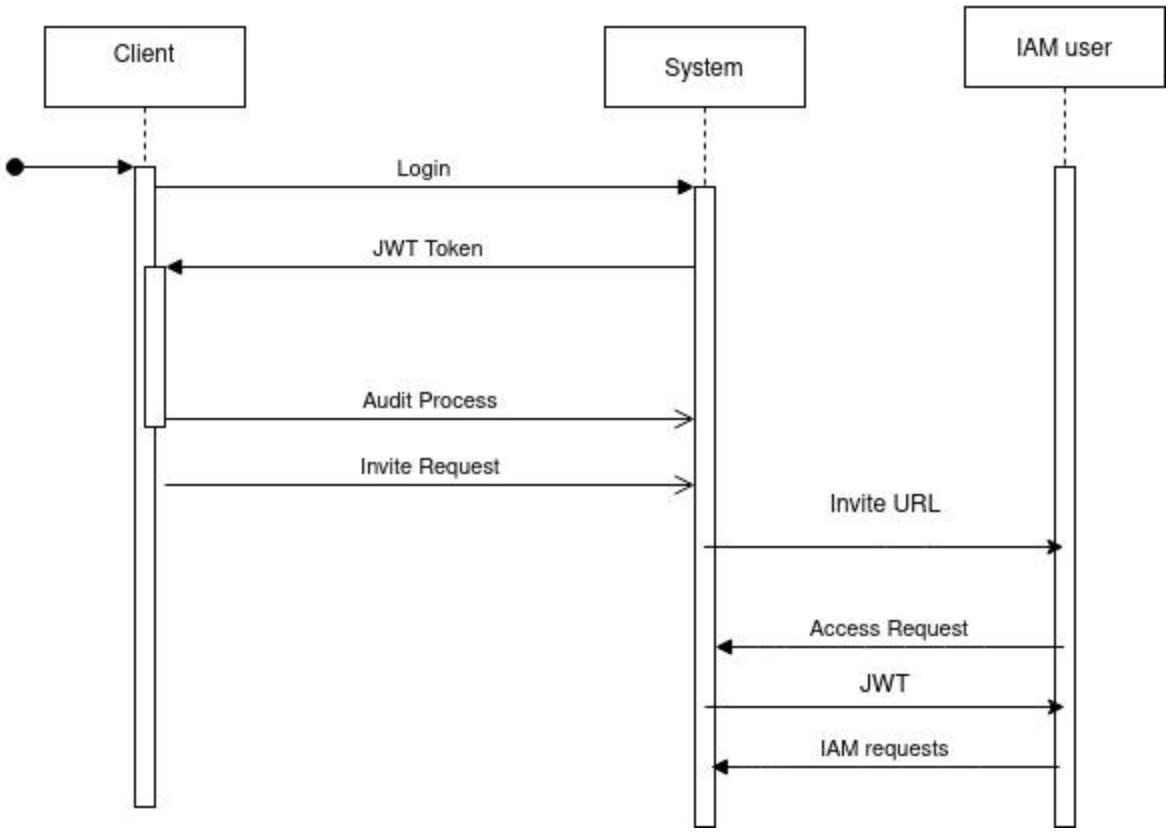


Figure 14: Sequence Diagram Invite IAM members

4 Chapter 4: Future Work

In the Future, we would like to increase the security of our client program, to decrease obfuscated parameters a malicious program can introduce. Auditing and Hardening infrastructure for AWS, Heroku , docker containers for minimal risk of zero day attacks and Expand our service to Enterprise software to use Fuzzer as a Service, that would be helpful to discover Zero-Day Vulnerabilities in them, at last we would like to expand our services to open-source software to find vulnerabilities in them and provide mitigation routes for the problem.

5 Chapter 5: Conclusion

In this project, we applied our vision of a portable and reliable Auditing process of hundreds of devices for critical inspections and to provide hardening of the windows environment, which is crucial to enterprise infrastructure.

We developed a AaaS (Audit-as-a-Service) that is intuitive, frictionless which can handle hundreds of audits concurrently with zero downtime and minimal data-storage, we applied advanced techniques of two-way-communication like web sockets, WebRTC in our infrastructure which helped us to achieve significant amount of reduction of time in the auditing process.

6 Chapter 6: System Implementation

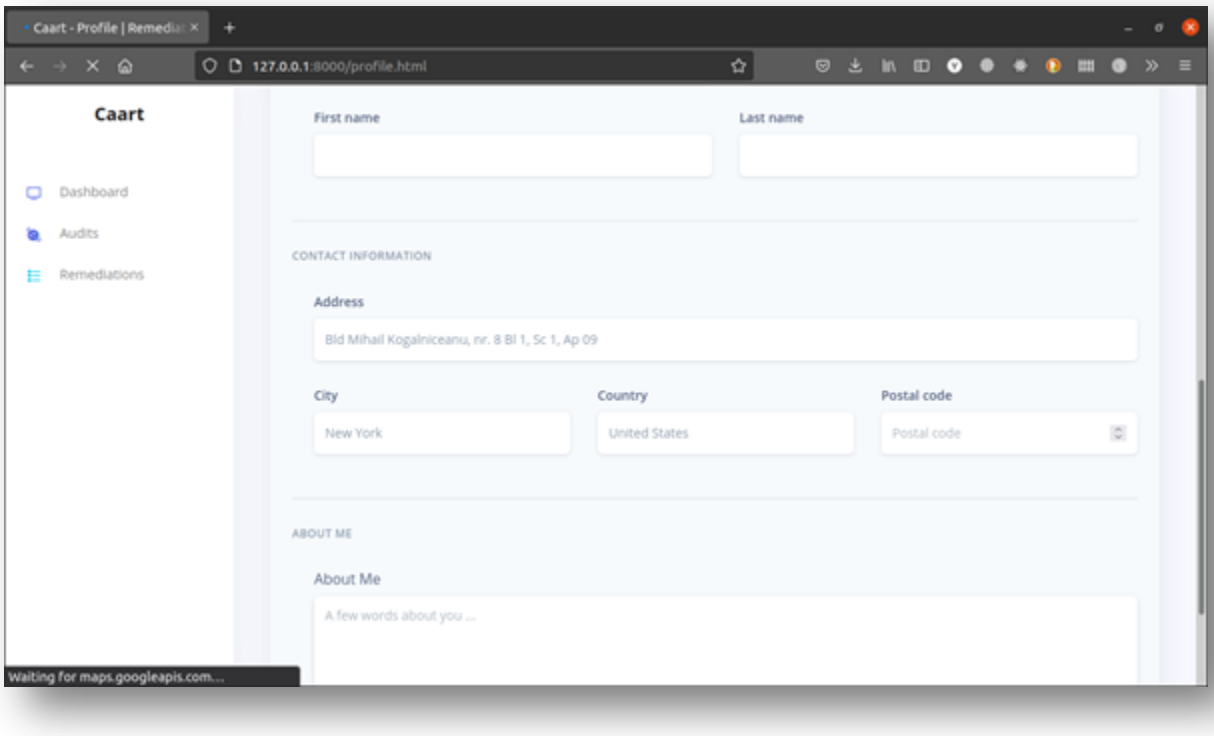


Figure 15: Cart Module

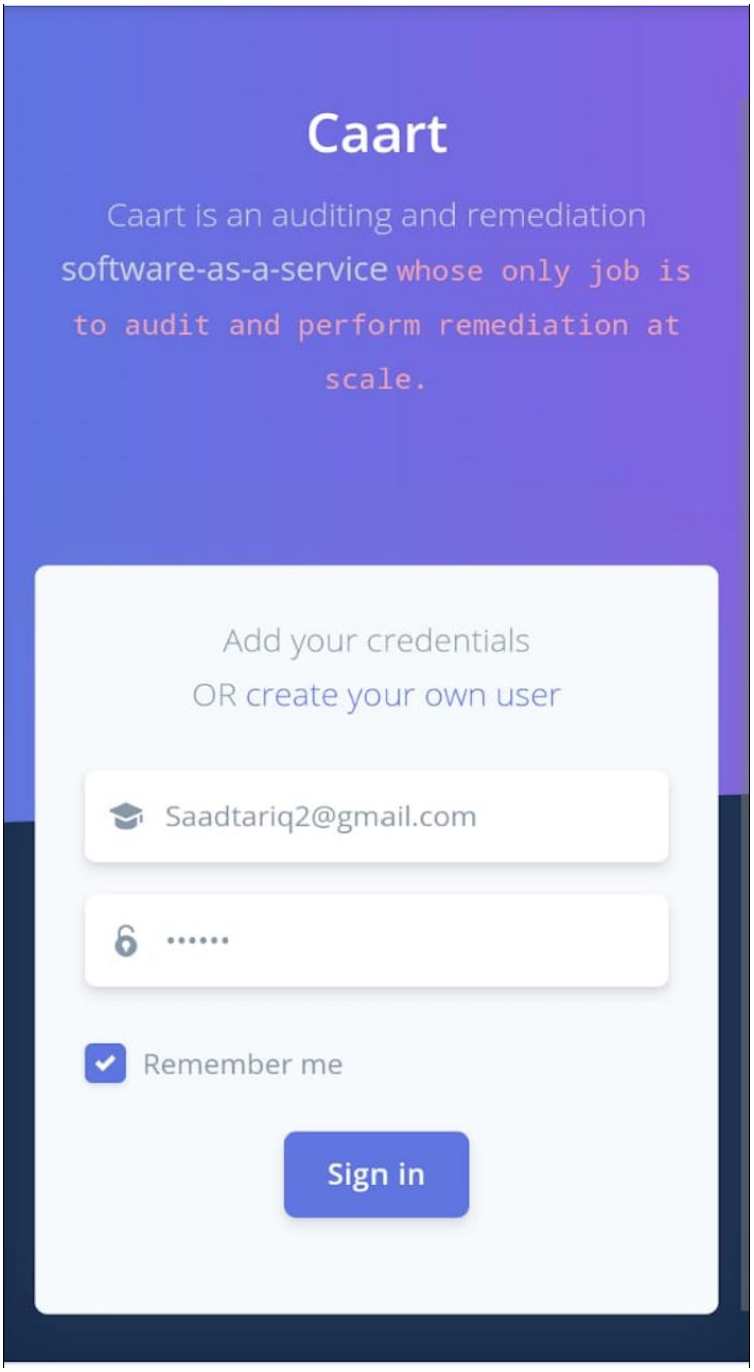


Figure 16: Sign In Screen

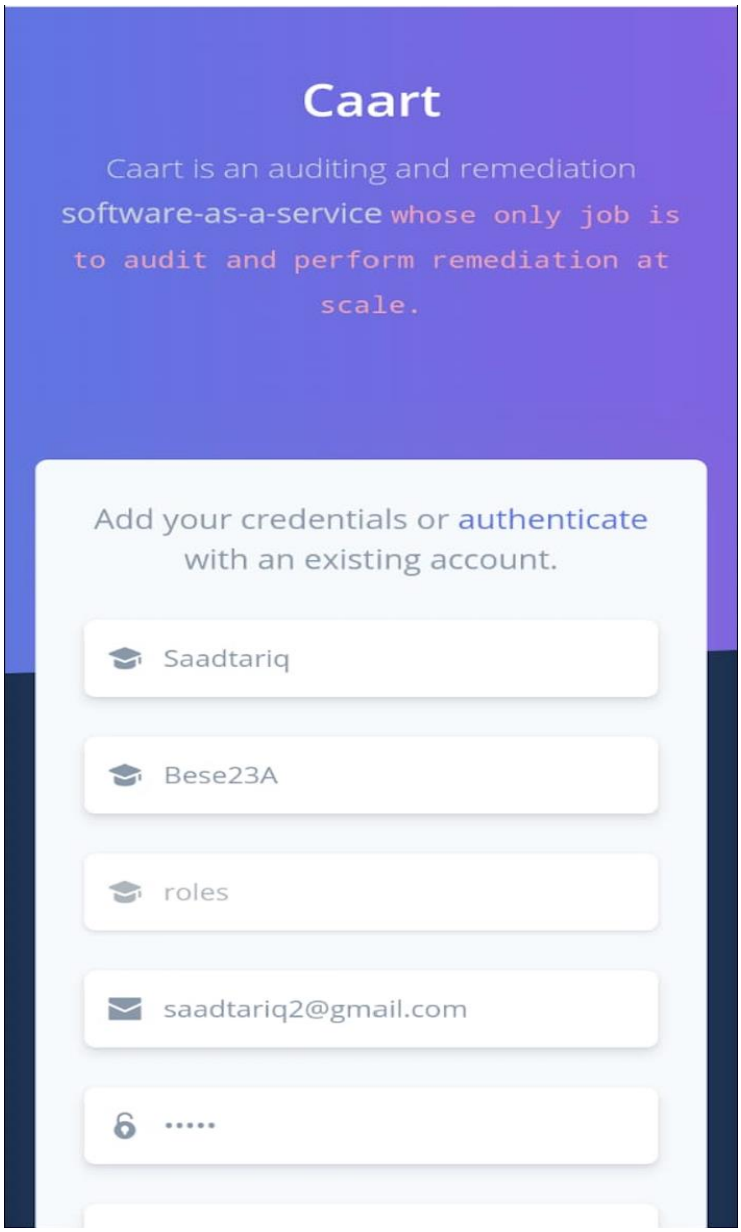


Figure 17: Sign Up Screen

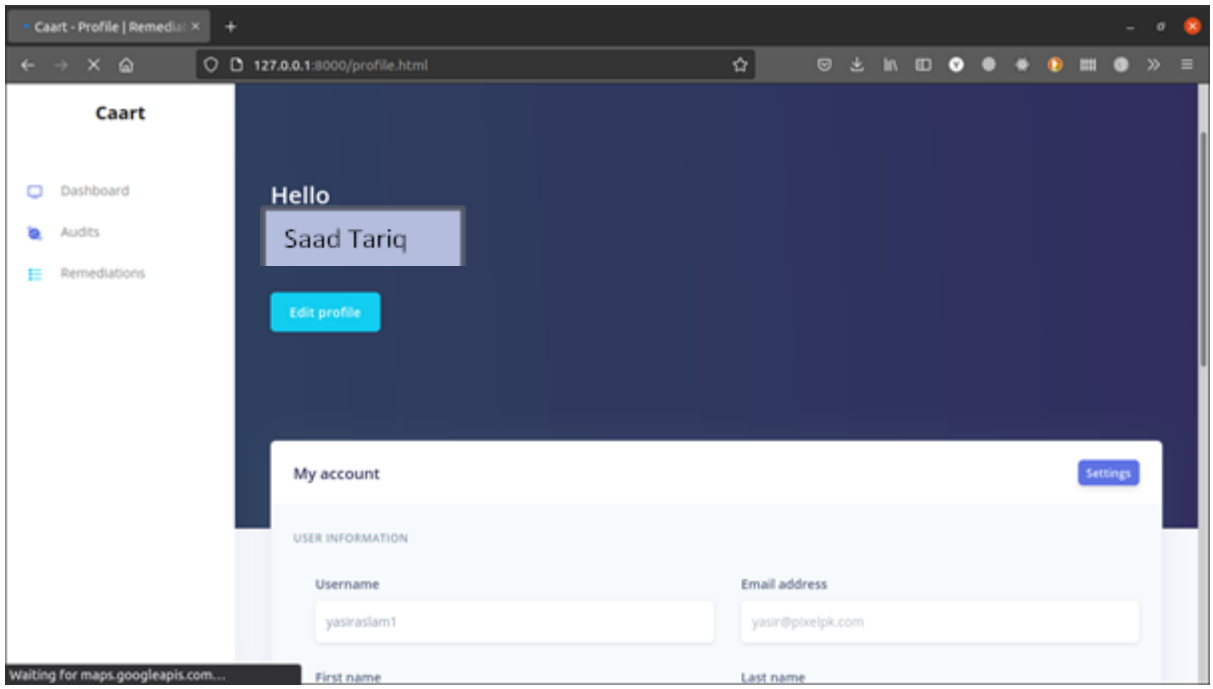


Figure 18: Profile Screen

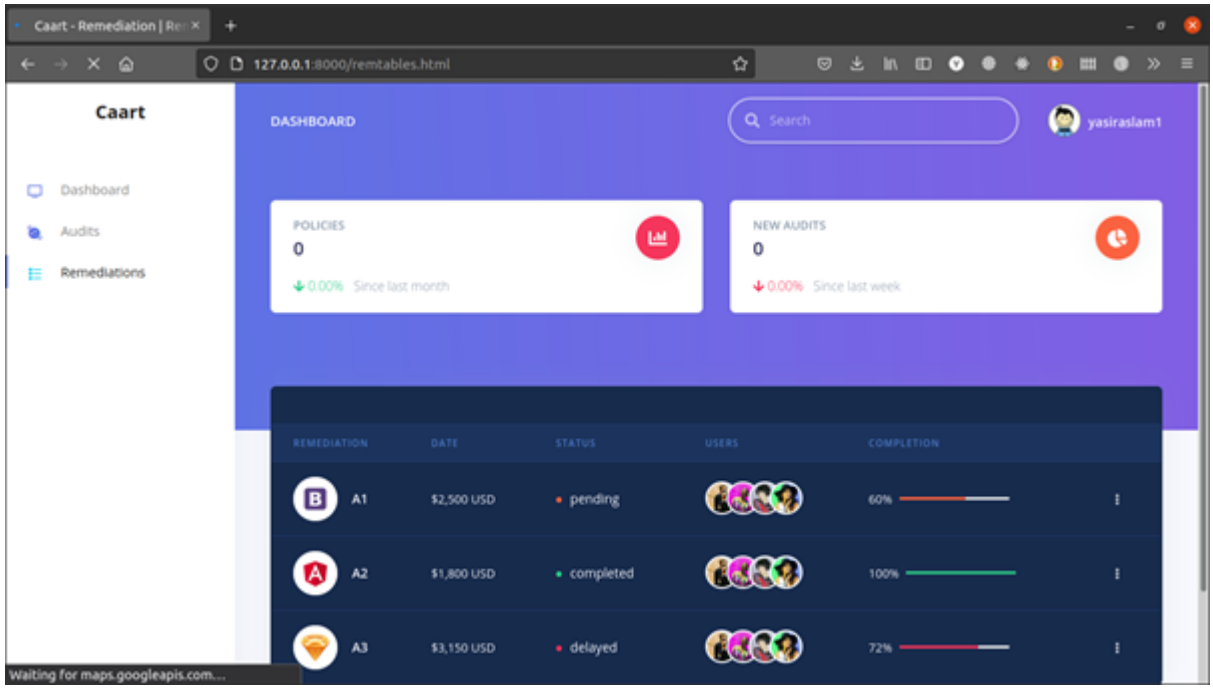


Figure 19: Remediations

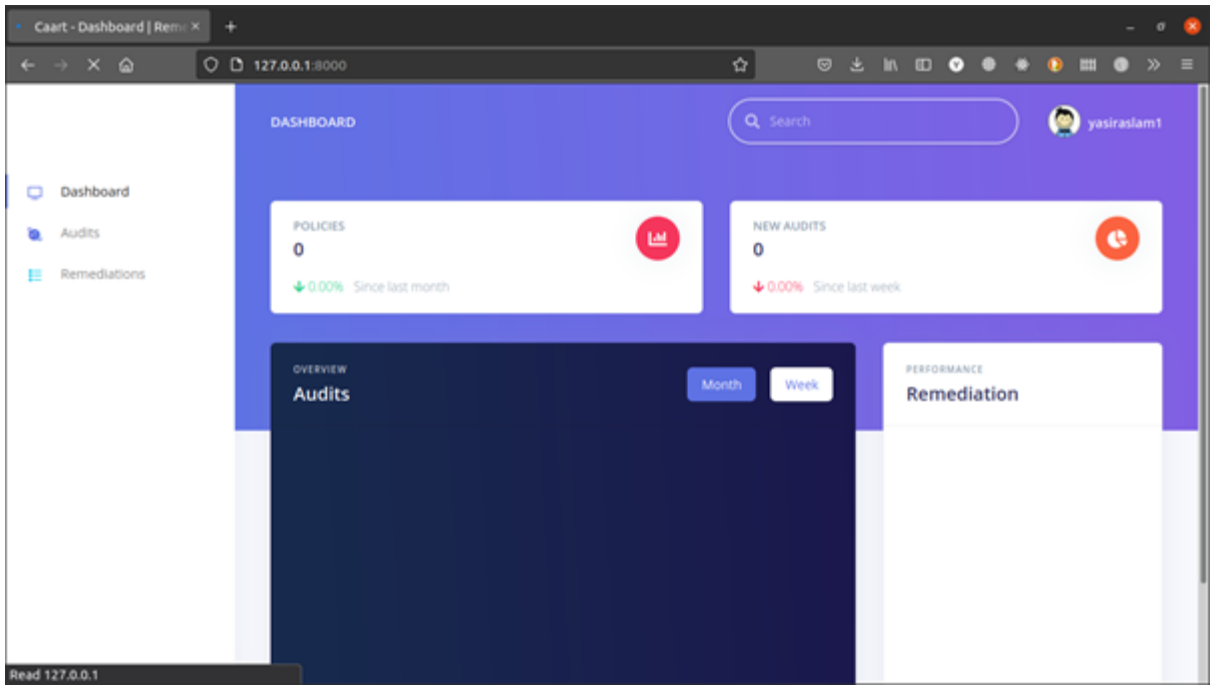


Figure 20: Dashboard

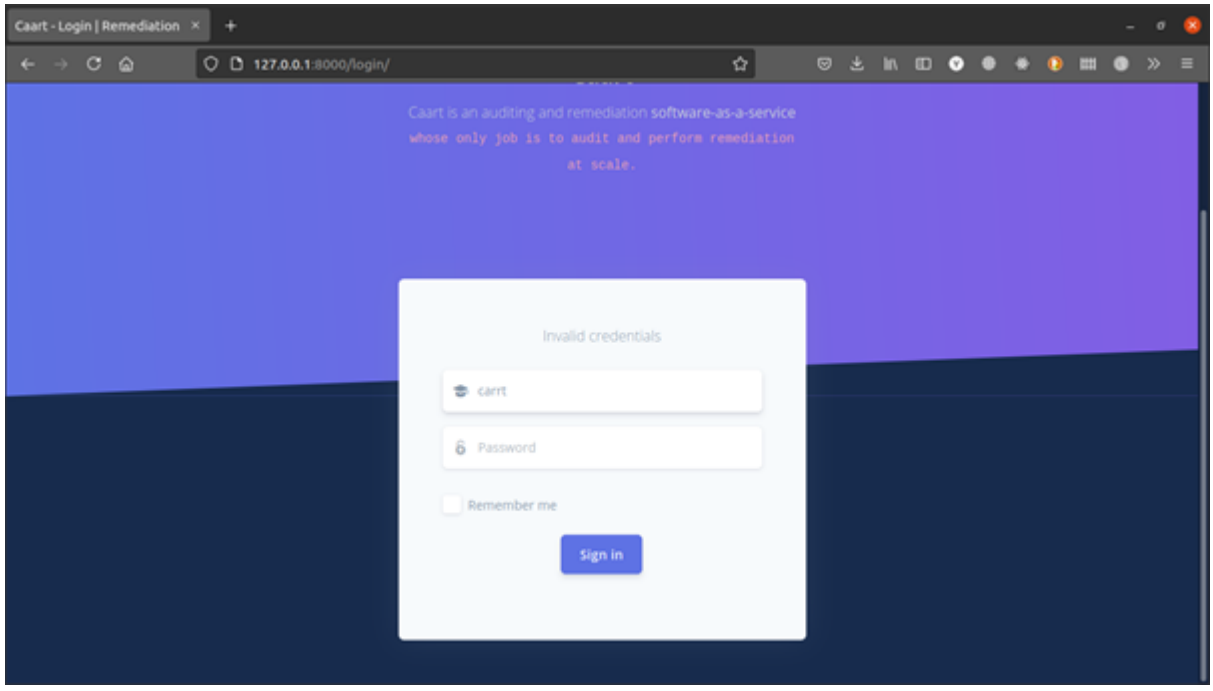


Figure 21: Sign in Screen Laptop design

7. Chapter 7: REFERENCES

1. *Perminov, P., Kosachenko, T., Konev, A., & Shelupanov, A. (2020). Automation of information security audit in the Information System on the example of a standard "CIS Palo Alto 8 Firewall Benchmark". International Journal, 9(2).*

2. *PONGSRISOMCHAI, S., & NGAMSURIYAROJ, S. (2019, February). Automated IT Audit of Windows Server Access Control. In 2019 21st International Conference on Advanced Communication Technology (ICACT) (pp. 539-544). IEEE.*

3. *Spindel, B. (2002). Benchmarking System Security: A new assessment tool can help auditors measure computer security against established benchmarks. (Computers & Auditing). Internal Auditor, 59(1), 23-26.*

4. *Souppaya, M., Harris, A., McLarnon, M., & Selimis, N. (2002). Systems Administration Guidance for Securing Microsoft Windows 2000 Professional System. NIST Special Publication, 800, 43.*

CAaRT - Final

ORIGINALITY REPORT

10%	5%	1%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	docslide.in Internet Source	2%
2	Submitted to Kuala Lumpur Infrastructure University College Student Paper	1%
3	Submitted to Napier University Student Paper	1%
4	Submitted to Higher Education Commission Pakistan Student Paper	1%
5	Submitted to Asia Pacific International College Student Paper	1%
6	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	1%
7	Submitted to UNITEC Institute of Technology Student Paper	1%
8	en.wikipedia.org Internet Source	1%

9	Submitted to Sonora High School Student Paper	1 %
10	Submitted to Queen Mary and Westfield College Student Paper	<1 %
11	findarticles.com Internet Source	<1 %
12	Submitted to University of Maryland, University College Student Paper	<1 %
13	Submitted to Kuwait University Student Paper	<1 %
14	Submitted to University of Greenwich Student Paper	<1 %
15	www.slideshare.net Internet Source	<1 %
16	Fiedelholtz. "The Cyber Security Network Guide", Springer Science and Business Media LLC, 2021 Publication	<1 %
17	Submitted to INTI International University Student Paper	<1 %
18	www.irjet.net Internet Source	<1 %
19	Internet Source	<1 %