Automated Teller Machine (ATM) Monitoring Dashboard System

D.C Pushpakumara
2024

Automated Teller Machine (ATM) Monitoring Dashboard System

A dissertation submitted for the Degree of Master of Information Technology

D.C Pushpakumara
University of Colombo School of Computing
2024

Declaration

Name of the student: Dewa Chinthaka Pushpakumara
Registration number: 2020/MIT/080
Name of the Degree Programme: Degree of Master of Information Technology
Project/Thesis title: Automated Teller Machine (ATM) Monitoring Dashboard System

1. The project/thesis is my original work and has not been submitted previously for a degree at this or any other University/Institute. To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

2. I understand what plagiarism is, the various types of plagiarism, how to avoid it, what my resources are, who can help me if I am unsure about a research or plagiarism issue, as well as what the consequences are at University of Colombo School of Computing (UCSC) for plagiarism.

3. I understand that ignorance is not an excuse for plagiarism and that I am responsible for clarifying, asking questions and utilizing all available resources in order to educate myself and prevent myself from plagiarizing.

4. I am also aware of the dangers of using online plagiarism checkers and sites that offer essays for sale. I understand that if I use these resources, I am solely responsible for the consequences of my actions.

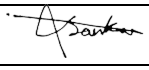
5. I assure that any work I submit with my name on it will reflect my own ideas and effort. I will properly cite all material that is not my own.

6. I understand that there is no acceptable excuse for committing plagiarism and that doing so is a violation of the Student Code of Conduct.

Signature of the Student	**Date (DD/MM/YYYY)**
	28/09/2024

Certified by Supervisor(s)

This is to certify that this project/thesis is based on the work of the above-mentioned student under my/our supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

	Supervisor 1	**Supervisor 2**	**Supervisor 3**
Name	Dr. A.P. Sayakkara		
Signature			
Date	2024-09-30		

Abstract

This document presents the design, development, and implementation of a web based Automated Teller Machine (ATM) Monitoring Dashboard System that provides real-time monitoring of machine status and performance to Bank of Ceylon with the aim of ensuring that the machines are operational and delivering a good user experience.

Bank of Ceylon has a large interconnected network covering citizens all over the country with 571 branches, more than 600 ATMs (Automated Teller Machine), 850 CRMs (Cash Recycling Machine) and 10 CDMs (Cash Deposit Machine). At present, IT unit of the bank use the view functionality of the BASE24 system to check the status of ATM/CDM/CRM. Due to some limitations of BASE24 application a monitoring dashboard system was developed in order to overcome these limitations.

The primary objective of this system is to provides real-time monitoring of machine status and performance of ATM Machines. The system is implemented as a fully web-based application. The front-end is developed using HTML, CSS, and JavaScript, providing a responsive and user-friendly interface that users can access through any preferred browser. The back-end is built using Java with RESTful web services, ensuring a modular and stateless architecture that allows seamless communication between the client and server. The system uses an Oracle database to manage data securely and efficiently. Users access the application through the Bank's internal network, ensuring security by restricting usage to internal personnel only.

To evaluate the effectiveness of the ATM monitoring dashboard, a series of tests and case studies are conducted using a representative sample of ATMs in a simulated environment and testing were carried out by the selected core workers of QA Team and Business Teams.

In conclusion, this document underscores the importance of real-time monitoring in the banking industry and presents a comprehensive ATM monitoring dashboard as a valuable tool for optimizing ATM network operations.

Acknowledgement

I would like to express my deepest appreciation to all those who provided me the possibility to complete this project. I am thankful for their aspiring guidance, invaluable constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

First and foremost I would like to expresses my gratitude to my supervisor Dr.Asanka Sayakkara for the encouragement, guidance, advice, information and motivation to me throughout the project.

I also wish to extend my sincerest thanks to Mr. Ruwan Nanayakkara, Chief Manager IT of Bank of Ceylon and staff members of the Quality Assurance department of Bank of Ceylon for their vital support given to me at the planning, development and evaluation stages of the system. Their support paved the way towards a well-built system.

Further the acknowledgement would be incomplete if I don't mention a word of thanks to my beloved parents and my loving wife whose continuous encouragement had led me to successful completion of the project.

Table of Contents

List of figures

List of Tables

List of Abbreviations

ATM – Automated Teller Machine

BOC – Bank of Ceylon

CDM – Cash Deposit Machine

CRM – Cash Recycling Machine

CSS - Cascading Style Sheet

HTML - Hyper Text Markup Language

HTTP – Hypertext Transfer Protocol

QA – Quality Assurance

UAT – User Acceptance Testing

Chapter 1 – Introduction

1.1 Project Overview

Bank of Ceylon (BOC) considered to be the Sri Lanka's foremost state-owned and major licensed commercial bank which established on 1st of August 1939. The bank has a large interconnected network covering citizens all over the country with 571 branches, more than 600 ATMs (Automated Teller Machine), 850 CRMs (Cash Recycling Machine) and 10 CDMs (Cash Deposit Machine). In addition to the local presence, the bank operates four overseas branches, namely, Chennai, Male, Hulhumale, Seychelles and 2 ATMs and 2 CRMs in Male branch. Moreover, the bank has a fully operated subsidiary in United Kingdom-London. [1]

Altogether the aforementioned ATM/CDM/CRM network contains over 1500 machines for electronic telecommunication, by diversifying bank operation through enabling the customers to perform financial transactions such as cash withdrawals, deposits, fund transfers and account inquiries at any time without the direct interaction with bank staff. Even though, these machines have become a significant touch points in the customer experience journey, in some circumstances this customer service can be unavailable. Main cause behind this unavailability occurred due to numerous faults including but not limited to Network faults, Cash faults and Card reader faults.

However, when focusing on the ATM network, BASE24 application need to be highlighted since it is considered as the heart of this network. BASE 24 is a product of ACI Worldwide company that is specialized for the electronic payments systems used by large financial institutes. All the ATM/CDM/CRM machines in different locations of the country are operated and handled by this application. In other words, any ATM/CDM/CRM in the BOC network are connected to the BASE24 system.

At present, IT unit of the bank use the view functionality of the BASE24 system to check the status of ATM/CDM/CRM. But, that feature only allows the user to view the current status of single machine at given time. Due to that limitation, currently they are unable to examine the entire ATM network to identify the issues of the machines. As a result, the monitoring team of the bank do not attend to these matters, unless those are being informed or inquired by a particular branch employee or by a customer.

Moreover, BASE 24 application can be accessed only from the BOC head office and BOC DR (Disaster Recovery) centre due to network restrictions and regulatory compliances of the bank.

Thus, in order to get the machine status or cash balance of a particular ATM/CDM/CRM, branch officer have to physically visit the machine location or otherwise to contact the Head office/DR centre.

1.2 Motivation

Major problems and issues identified when analysing the existing process that used to monitor the ATM network are listed in below.

1. BASE24 system primarily focuses on transaction processing and related aspects, potentially overlooking other important and critical areas such as physical security, device malfunctions, or network connectivity issues. Thus, a comprehensive monitoring solution is needed to cover the essential aspects of ATM operations.
2. In situation where ATM/CDM/CRM status or cash balance needs to be checked, either branch officer has to physically reach the ATM location which could be in the same location of the branch/ different location or to contact the Head office or Disaster Recovery Centre. In most cases, officers face numerus difficulties when accessing machines which are placed in a different site than their branch location. Inability to address such matters immediately, will build a negative perception in consumers and stakeholders mind, which will at the end directly cause a reputational damage to the bank.
3. As BASE24 system is a dedicated switch which build to cater the transaction handling process mainly, it does not provide the capability to collect and analyse data in a comprehensive manner. Even though, it featured with viewing the status of a particular machine at a given time, that is not capable of providing various reports requested by the bank management occasionally related to cash management and ATM down time for data analytics and auditing purposes.
4. Inability of list down the ATM/CDM/CRM details based on different criteria such as service unavailable machines, cash empty or cassette fault machines, machines with card reader faults and pin-pad faults.
5. Currently, ATM/CDM/CRM machines are being examined for issues and other anomalies during an inspection or periodic checks only. This can cause for such critical issues to undercover for a long period of time. Thus, at the end results in extended periods of machine downtime.
6. Current version of the BASE24 system in the bank, considered to be a legacy application and more likely to function based on commands. Thus, only trained and

experienced bank employees are able to operate it. Furthermore, due to the limited number of licenses available for this system, unable to provide several users for the monitoring purposes.

7. Monitoring multiple ATMs spread across various geo locations can be challenging and time-consuming for manual operators. Moreover, with the future expansion of the ATM network, it will become extremely difficult to maintain consistent and comprehensive monitoring coverage just based on the current features in the BASE24 system.

To overcome these drawbacks, implementing an automated ATM monitoring system can provide real-time monitoring, reduce human error, offer wider coverage, scale with the ATM network, provide data analytics capabilities, and maintain a comprehensive audit trail. Furthermore, automated system can enhance the efficiency, accuracy, and effectiveness of the process of ATM monitoring which at the end leading to improved security, operational efficiency, and customer satisfaction.

1.3 Objectives

The main goal of this project is to design and develop a web-based ATM/CDM/CRM monitoring dashboard to provide real-time monitoring of machine status and performance, with the aim of ensuring that the machines are operational and delivering a good user experience. Below are the major objectives of the suggested ATM/CDM/CRM monitoring dashboard.

- Design and develop a web based “ATM/CDM/CRM monitoring dashboard” that provides real-time monitoring of machine status and performance to BOC Head office and DR premises.
- Design and develop a web-based application to view their branch ATM/CDM/CRMs status and cash position at any time.
- Notify the alerts to the branch users about critical issues such as Network failures, Cassette faults, Pin pad faults, Card reader faults
- Provide ATM downtime reports using historical data in order to identify the problems related to the ATM network

1.4 Background of the Study

Base24 is a real-time transaction processing system used in the financial industry to process high-volume electronic transactions. It is developed and marketed by ACI Worldwide, a global provider of real-time electronic payment and banking solutions. Base24 is designed to handle a wide range of transaction types, including ATM withdrawals, point-of-sale transactions, credit and debit card authorizations, and online banking transactions.

Bank of Ceylon uses two different types of communication protocol for parse messages between ATM/CDM/CRM and BASE24 system.

1. Diebold Direct Connect (DDC)
2. NCR Direct Connect (NDC)

Currently Bank has a significant ATM/CDM network which consist of more than 600 ATMs (Automated Teller Machine), 850 CRMs (Cash Recycling Machine) and 10 CDMs (Cash Deposit Machine). All these machines are connected to the BASE24 application and maintained by a dedicated ATM team in the IT division of the bank.

Since BASE24 system is built to cater the transaction handling process mainly and does not provide the capability to collect and analyse data comprehensively, need for a dedicated monitoring system has been highlighted during the past few years in the bank.

Even though, BASE24 has some functionalities to provide full detail report for a specified ATM/CDM/CRM machine, querying set of data that belongs to more than one machine has become a great challenge for the respective IT team. Below screenshot illustrates the screen of the current system, in which allows to query all the details related to a specific ATM (property named “ATM TERMID”).

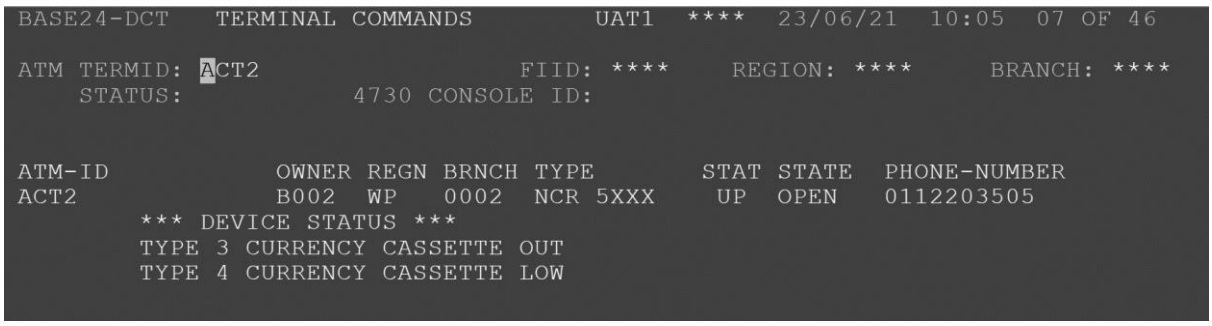


Figure 1- BASE24 ATM Status Screen

In other words, for a given time if need to check all the island wide ATM/CDM/CRMs which are having specified kind of issue (ex: service unavailable machines or cash emptied machines), bank need to purchase a separate system which is specifically built to monitoring of ATMs/CDM/CRMs. At present with the limitation of the current BASE24 system, head office IT team attends to the machine issues, once only those issues have been informed from the branch level employee or through a complain of a customer or any other third party. Because of that at a given time there could be several machines in different branch location which are having issues, but immediately not been fixed due to a lack of proper monitoring solution.

Moreover, there are times when the branch user/employee need to check the cash balances of the ATM machines in their branch. In such situations user can make use of the Admin card provided by the head office to check the necessary details, by directly accessing the branch ATM machine. But, there are some cases where the user has to check the cash balance of an ATM which is assigned to their branch but located in a different location. In such scenarios, the user has to physically visit the ATM to directly access the machine and collect the details. Or otherwise, user can contact the head office IT team via email or over the phone and request for the data. If the user is able to view the cash balance details related to all the ATMs assigned to their branch via a single screen, it would be much effective in many ways.

1.5 Scope of the Study

In actual scenario ATM related data is being extracted from a system file in the Base 24 system. Extracted data is being used and the processed data is presented in the ATM monitoring system. For the project purposes, separate ATM simulator is being created and records of relative data are updated to the database. Which in the end being the input for the proposed ATM monitoring system.

1. Administrative Module
    a. User Profile Management – Allow administrator to add/modify and disable users in the system.
    b. User Privilege Management – Assign user allowed functions only.
    c. ATM Information Management – Allow administrator to add/modify machine related data.

2. ATM/CDM/CRM Current Status Module

This module allows users to monitor the current status of the machine with the below details.

- Machine Name – Bank use different identical names to identify the machines
- Branch Code – One branch can have one or more machines attached to them
- Location – Where does the machine installed
- Cassette Information – Cassette is the container in which the cash is kept inside the machine. Inside the cassettes have sensors to identify the cash level. When it becomes low or empty machine will send a message to the core system. Cash can be jammed inside the cassettes also this happens it will send a message as cash fault.
- Receipt Printer Information - Same as the cassette, receipt printer is indicating receipt printer paper roll is low, empty and fault.
- Pin Pad Information – Pin pad is use to enter the pin number and other information like customer requested amount. When the pin pad is not working properly will indicate to the core system.
- Currency Acceptor (CDM/CRM), Money Drawer (ATM) Information – In the Cash Deposit Machine and Cash Recycling Machines cash acceptance and cash dispenser unit are same its identified as Currency Acceptor. In the Automated Teller Machines this unit have only the cash dispenser unit. This unit can be faulted due to cash jam inside the currency acceptor unit or cash dispenser unit.
- Card Reader Information - Card reader can be faulted due to card stuck inside the card reader.
- Network Status – Machines are connected through the intranet to the Base24 system. When the machine has no network, it indicates as machine is down.
- Machine Status – When the branch users log into the machine, machine is out of service for the customers.
- Cash Level Information – An ATM has 5 types of cassettes. 4 cassettes for denominations of 100,500,1000,5000 and one cassette is to store retracted notes. A CDM/CRM consists of 6 types of cassettes. 4 cassettes for the denominations of 100,500,1000,5000, one cassette is for the All Acceptance cassette (This cassette accept all the types of allowed denominations) and one cassette is for the reject notes.

- Last Sent SMS – Branch users are informed when an error occurs and escalated to the higher-level officers. In the dashboard can view the last send SMS during last two weeks.

3. ATM/CDM/CRM Dash Board Module

   This module includes different types of graphs represent the below described information related to machines.

- Count and Percentages of in service for the customers and out of service machines due to network failure, machine fault or the branch operator log into the machine.
- Breakdown of in-service machines into three categories –
    - No fault machines – With no faults of the machine
    - Minor issues machines - Customers can use the machine functions, but with the minor issues like cassette low, receipt printer low
    - Critical issues machines – Customer can't use the cash related functions of the machine due to faults like all cassette empty, all cassette fault, currency acceptor faults.
- Total active machine count based on the machine type (ATM/CDM/CRM) and machine venders (DMS/Informatics/Hitachi)
- Breakdown of machines grouped by its faults – Cassette empty, cassette faults, receipt printer faults, card reader faults, pin pad faults, currency acceptor / money drawer faults.
- Detailed information of each category.

4. ATM Fault Tracking Module

   This module allows ATM monitoring team to detect and report faults like alerting the relevant parties when a fault is detected.

5. Reporting Module

   This module contained the detailed information (Starting time, End time, Total duration) about below listed fault groups for a selected time period.

- All Cassette Empty
- All Cassette Fault
- Pin Pad Fault
- Card Reader Fault
- Currency Acceptor or Money Drawer Fault
- Network Failure

6. Web based application for Branch Users

   Branch users are allowed to view their branch ATM/CDM/CRM status and cash balance of each cassettes.

7. User Acceptance Testing Module

   In order to get the user acceptance, developed system needs to be tested by user department, IS-Audit department, risk department and compliance department based on different aspects.

1.6 Feasibility Study

A feasibility study is a systematic assessment of the practicality and viability of a proposed project, initiative, or course of action. It evaluates various aspects, including technical, economic, legal, operational, and scheduling considerations, to determine if the project is feasible and worth pursuing. Here are some key elements included in the feasibility study of proposed ATM monitoring system.

1. Technical Feasibility

   Project was developed as a Web-based application using HTML and JavaScript as an open source server-side programming language and Oracle as database management system which are capable of delivering highly reliable solutions. In order for the Web application to be accessible across the whole branch network, it was hosted on a virtual machine server that was located at the bank's internal network. It can be determined that the project was technically feasible when considering the primary technologies utilized for the development of the web application and the hardware used for hosting the application.

2. Economic Feasibility

   Technology used for the Development of web application is open source technology which is available for free. VM server for host the application and the database are provided by the bank. As a result, since the project did not bare any additional costs, it can be said to have been economically feasible.

3. Security Feasibility

   The Web application is only available within the bank's intranet and is hosted on a virtual server there. As a result, users are prohibited from accessing the Web application via the public internet, and only authorized users are permitted to view the features that are permitted. Given the aforementioned information, it can be said that the project is security-wise feasible.

4. Legal Feasibility

   Since the system does not need to collect any customer or financial information relating to any of the projects and it is not integrated with any other banking systems such as the Core Banking System. Therefore, even though the system is only available to Bank officers, no staff member will have access to customer sensitive data through the system. Therefore, it can be concluded that the project is legally feasible.

1.7 Structure of the Dissertation

The remaining chapters of the dissertation are organized as follows;

1. Chapter 1 – Introduction

   In this chapter, it mainly focuses on the introduction of the system which has been developed. It briefly mentions the problems of the current system, the solutions and the scope of the developed system.

2. Chapter 2 – Background

   Second chapter describes the background of the system and the literature review about similar systems and technologies. It describes the analysis of the requirement with the different technologies and the design strategies which were used to develop the project.

3. Chapter 3 – Design Architecture

   In this chapter, it illustrates the requirement analysis and the design of the solution and also it shows all the UML diagrams such as Deployment diagram, Component diagram, Class diagram, Sequence diagram, State Charts, Activity diagrams and the database design using the ER diagrams. Major module structures are comprehensively explained and all the test plans are described in this chapter.

4. Chapter 4 – Methodology

   The Methodology phase of the system is the most important part of the project development. It describes the implementation environment, related technologies and tools, modules of the proposed system and their interactions. Also describes the selected technologies and tools. It can also be used to give clients a clear idea of the project during the development phase

5. Chapter 5 – Testing and Evaluation

   This chapter includes techniques of testing, types of software testing, test cases of the system and user evaluation.

6. Chapter 6 – Conclusion

   This includes the conclusion of the project which highlights the weaknesses/limitations of the used technique and solutions for them. It describes the benefits of developing the project to the Bank.

Chapter 2 – Background

2.1 Introduction

This chapter includes reviews of current systems that are comparable to the developed system and a thorough study of the system's requirements. The chapter mainly focuses on the functionality and the technologies utilized in both systems when comparing the developed system with similar systems. Finally, the technologies and design principles that were employed in the system's development are also discussed.

2.2 Requirement Analysis

In the Requirement analysis phase includes all the needs and expectations of Bank business unit team that were identified and agreed during the discussions. These requirements have then been further developed by identifying actors, identifying functional and non-functional requirements, Prioritizing and analysing requirements by conducting several discussions with the business team.

According to the information gathered through the discussion had with the staff members of the IT division, the overall summarized requirement as follows;

Basically, software requirements can be identified under two main categories: Functional requirements and Non-Functional requirements. Functional requirements describe the required software performance of the system while Non-Functional requirements discuss about the hardware and software requirements in developing, implementing and maintaining the system.

2.2.1 Functional Requirement

- Design and develop a web based “ATM/CDM/CRM monitoring dashboard” that provides real-time monitoring of machine status and performance to BOC Head office and DR premises.
- Design and develop a web-based application to view their branch ATM/CDM/CRMs status and cash position at any time.
- Notify the alerts to the branch users about critical issues such as Network failures, Cassette faults, Pin pad faults, Card reader faults
- Provide ATM downtime reports using historical data in order to identify the problems related to the ATM network
- Users with the administrative privileges should be able to manage other user profiles and user roles.

2.2.2 Non - Functional Requirement

- System should have user friendly interfaces and easy to handle.
- Users should be authenticated and should be role-based access controlled.
- System should be compatible with different web browsers.
- System should be handled multiple uses without affecting the performances.
- System should be available at any time without down time.

2.3 Review of Similar Systems

2.3.1 Diebold Nixdorf's Vynamic View

Diebold Nixdorf's Vynamic View is an advanced ATM monitoring system that provides real-time information and insights to help ATM operators manage their tasks more effectively. The system uses a combination of hardware, software, and cloud-based analytics to collect and analyse data from individual ATMs and across entire networks of machines. [2]

Some of the key features of Diebold Nixdorf's Vynamic View

- Availability Monitoring - Vynamic View Availability Manager provides you with an up-to-date, fast and detailed overview of your system availability in a user-friendly HTML interface. It visualizes in map form, on demand, the current availability status of the entire managed network and lets you manage them remotely from a monitoring centre
- Incident Manager - Vynamic View Incident Manager is an automatic service management system that ensures that when one of your devices goes down, a trouble ticket is generated, dispatch is notified via email, SMS, fax, or voice — and that someone is on their way with the proper parts and skills to fix it fast.
- Log & Journal File Handling - Vynamic View Log & Journal Manager is a software-agnostic solution providing a centralized environment that collects and stores data. Vynamic View Log & Journal Manager can be configured to collect Windows event logs, terminal application software log files and electronic journal files for easy accessibility and simplified processing.

2.3.2 NCR's APTRA Vision

APTRA Vision enables you to remotely monitor, manage, and troubleshoot your ATM network from a centralized dashboard. It also offers real-time alerts and reporting, as well as customizable dashboards and reporting [3] [4]

Some of the key features of APTRA Vision

- Dashboard Analytics - Enables real time comparisons of how transaction volumes are trending now on a given population of ATMs compared to how they did at a benchmark period of time. Users will see visual trending overlays on this dashboard of the different transaction types which will help to indicate which transaction types are being most utilized.
- Multivendor Capability - APTRA Vision's scope of support extends beyond the management of NCR ATMs. APTRA Vision supports assisted or self-service networks with a diverse range of device types, suppliers and network structures.
- High Performance Incident Management - APTRA Vision provides comprehensive incident management capabilities. It provides a visual display of incidents associated with clients.

2.3.3 Fiserv Device Manager - ATM & Self-Service Monitoring and Management

Device Manager is an event-driven incident management system that receives, analyses, responds to and manages critical self-service device information, supporting for all device types, including ATMs, kiosks, self-service devices, Teller Cash devices and POS devices, from any vendor. Device Manager uses real-time diagnostics from any type or make of Device to determine a resolution plan, automatically prioritizing and dispatching to service providers based on configurable business rules. It is designed to make real-time management of large networks of self-service devices easier and simpler. [5] [6]

2.3.4 FEEL View by GRG Banking

FEEL View is a multi-vendor terminal monitoring and comprehensive operation management system. With advanced functions like monitoring, remote distribution, reconciliation, ticket tracking, E-journal, OCR management and cash management, system can provide a centralized platform to manage branch performance and analyse resource profitability. FEEL View can also extend to support other management functions, such as ATM asset management, repairing service management, help desk for CIT company and maintenance personnel. [7]

Some of the key features of FEEL View

- Support ATM, CDM, CRS, STM, VTM, cash sorting machine monitoring.
- Support module monitoring (EPP, journal printer, receipt printer, card reader, dispense module, barcode scanner and etc.)
- Cassette Volume monitoring and Cash Replenishment report
- Customizable notification type and message (voice, video and text)

2.4 Related Technologies

ATM/CDM/CRM monitoring dashboard System was implemented as a fully web-based application where users will be able to access it through the Bank's internal network using any user preferred browser (Microsoft Edge, Chrome and Firefox). The following list of technologies and frameworks utilized to develop the application.

2.4.1 Frontend Development

Frontend development is the process of creating UI and UX of a web application. This process involves designing and creating visualize components of the application that users interact with directly. In general, frontend developments include the HTML, CSS and JavaScript to create and enhance the visual elements of a web application.

1. HTML

   This language provides the structure and content of a web page by using tags to define different elements such as headings, paragraphs, images, links, forms, and more.
2. CSS

   This will define the presentation and styling of web pages. It allows developers to control the layout, colours, fonts, and other visual aspects of HTML elements. This language can be used to create responsive designs for different screen sizes by defining separate styles.
3. JavaScript

   This is a programming language that enhances the interaction and dynamic behaviour of the web application by using features such as animations, form validations, event handling etc.

Bootstrap is a well-known frontend development framework which is used together with the above-mentioned front-end languages. Using such a framework will provide customizable predefined design components and classes which saves development time and effort.

Bootstrap Framework

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. Bootstrap is an HTML, CSS and JS library that focuses on simplifying the development of informative web pages (as opposed to web applications). [8]

2.4.2 Backend Development

This is the process of building server-side components of a web application or software system. It involves the implementation of the logic and data integration capabilities that allows applications to function properly. Java and RESTful web service will be used to develop the backend of this application.

RESTful Web Service

RESTful Web Service is a popular approach to creating web APIs that provides a simple and effective way to access data over HTTP. REST, or Representational State Transfer, is a software architectural style that defines a set of constraints to create scalable, reliable, and maintainable web services. It was designed to manage communication in distributed systems, particularly over the internet. RESTful APIs, or REST APIs, adhere to these architectural principles to provide a standardized and interoperable way for different systems to communicate.

RESTful APIs allow clients to interact with server-side resources using standard HTTP methods like GET, POST, PUT, and DELETE. These APIs use URIs to identify resources and use representations (such as JSON or XML) to communicate data between clients and servers. [9]

2.4.3 Database Management

Oracle database is used as the database for the web application and the database will be hosted in the same web server, which is used to host the web application.

Oracle Database

Oracle Database is a widely used relational database management system (RDBMS) developed by Oracle Corporation. It organizes data into tables consisting of rows and columns and supports SQL for querying and manipulating data. Key features include data integrity constraints, transactions with ACID properties, data security, query optimization, stored procedures, high availability, data compression, and support for various data types.

2.5 Related Design Strategies

2.5.1 RESTful Architecture

RESTful architecture, based on the principles of Representational State Transfer (REST), is a software architectural style used for designing networked applications. It provides a set of constraints and guidelines for building scalable, reliable, and maintainable web services. Here's an overview of the key principles and characteristics of RESTful architecture.

- Client-Server Architecture - RESTful application should have a client-server architecture. A Client is someone who is requesting resources and are not concerned with data storage, which remains internal to each server, and server is someone who holds the resources and are not concerned with the user interface or user state. They can evolve independently. Client doesn’t need to know anything about business logic and server doesn’t need to know anything about frontend UI.
- Statelessness - RESTful services are stateless, meaning that each request from a client to the server must contain all the information necessary to understand and fulfil that request. The server does not store any client state between requests, which improves scalability and reliability.
- Stateless Communication - RESTful services communicate using standard HTTP methods (GET, POST, PUT, DELETE) and status codes. This simplifies integration with existing web infrastructure and enables interoperability between different systems.

Chapter 3 – Design Architecture

3.1 Introduction

This chapter provides a detailed analysis on the technologies, design methods and the best practices which have been used to design a user-friendly ATM Monitoring Dashboard System. It illustrates the requirement analysis and the design of the developed solution using the UML diagrams such as Use Case diagrams, sequence diagrams and class diagrams.

3.2 System Architecture

A web app architecture presents a layout with all the software components (such as databases, applications and middleware) and how they interact with each other.

Typically, a web-based application architecture comprises 3 core components as given in Figure2.

1. **Web Browser**: The browser or the client-side component or the front-end component is the key component that interacts with the user, receives the input and manages the presentation logic while controlling user interactions with the application. User inputs are validated as well, if required.
2. **Web Server:** The web server also known as the backend component or the server-side component handles the business logic and processes the user requests by routing the requests to the right component and managing the entire application operations. It can run and oversee requests from a wide variety of clients.
3. **Database Server**: The database server provides the required data for the application. It handles data-related tasks. [10]

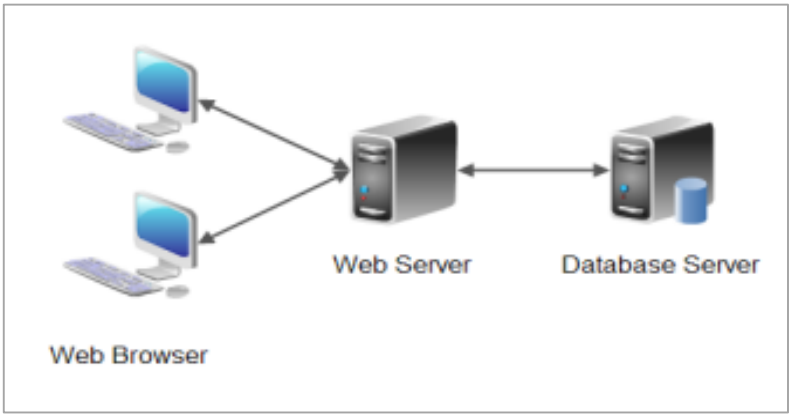


Figure 2- System Architecture Diagram

In the developed application frontend was written using HTML, CSS and JavaScript. Dataflow in a REST API is given in Figure 3. When a user sends an HTTP request to the server. The request contains information such as the HTTP method (GET, POST, PUT, DELETE), URI (Uniform Resource Identifier) of the resource, headers, and optionally a request body for data to be sent to the server. The server receives the HTTP request sent by the client. The server identifies the endpoint to which the request is addressed based on the URI provided in the request. After processing the request and updating the resource state (if applicable), the server generates an HTTP response. The response includes an appropriate status code (e.g., 200 for success, 404 for not found, 500 for server error), headers, and optionally a response body containing data to be sent back to the user. [11]

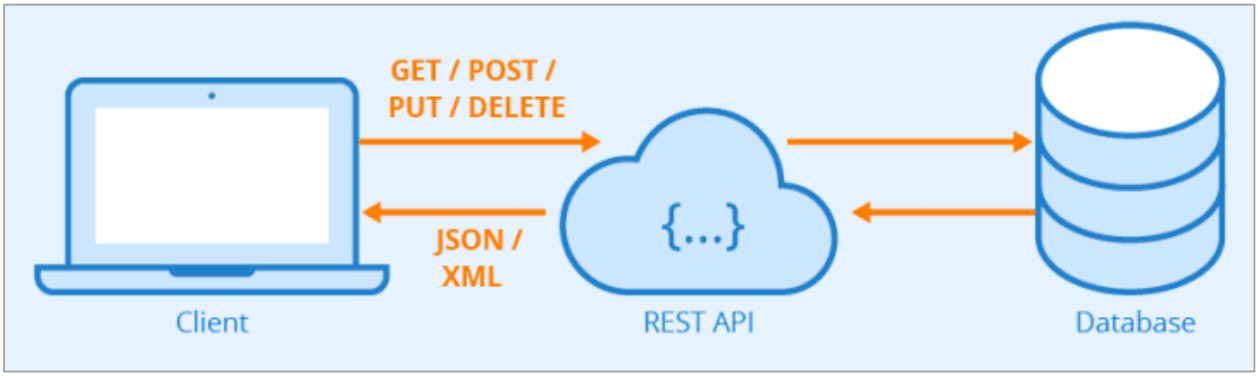


Figure 3- Data flow in REST API

3.3 UML Diagrams

3.3.1 Use Case Diagrams

Use Case Diagram for Administrator

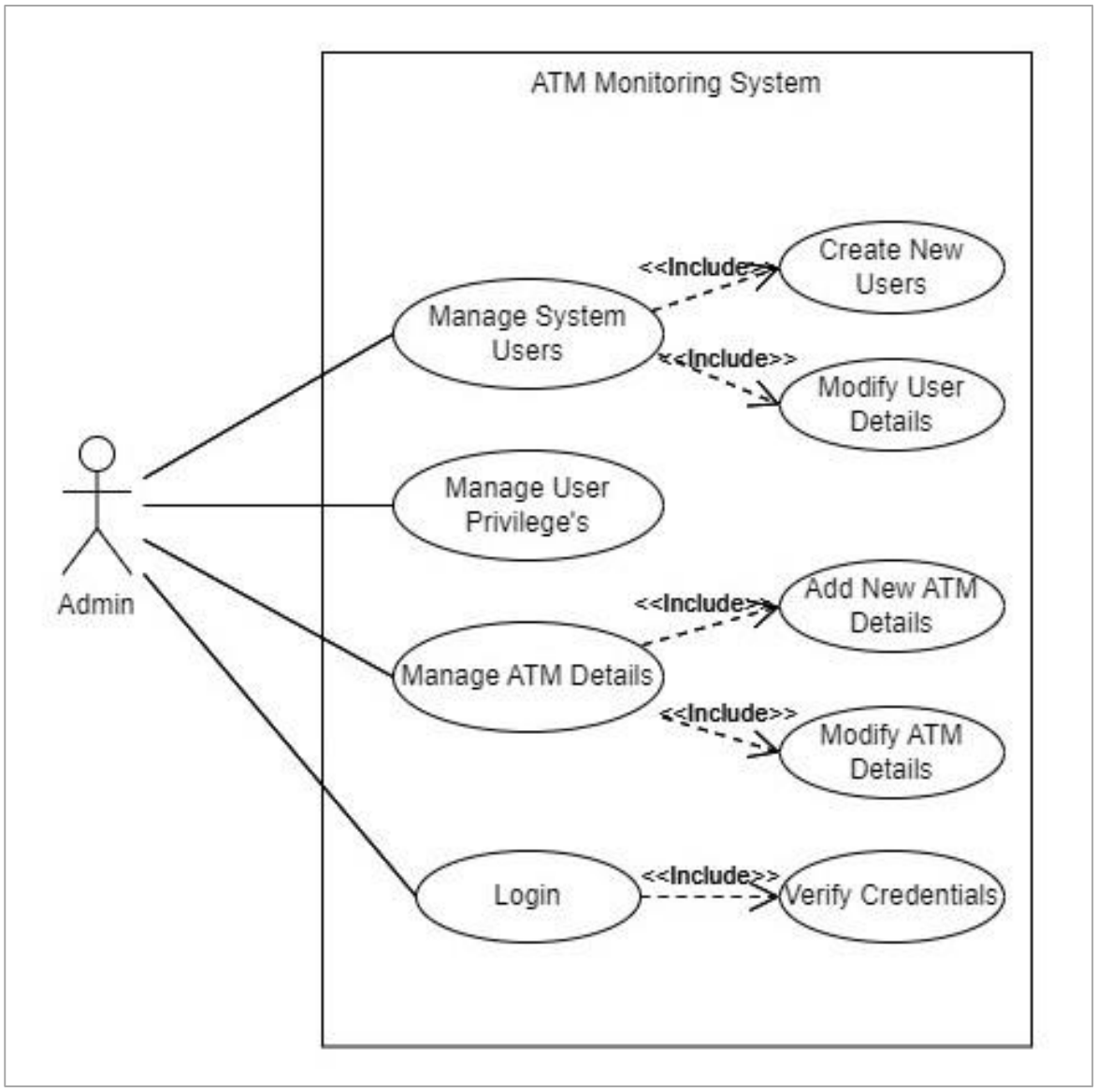


Figure 4- Use Case Diagram for Administrator

Use Case Diagram for Branch User

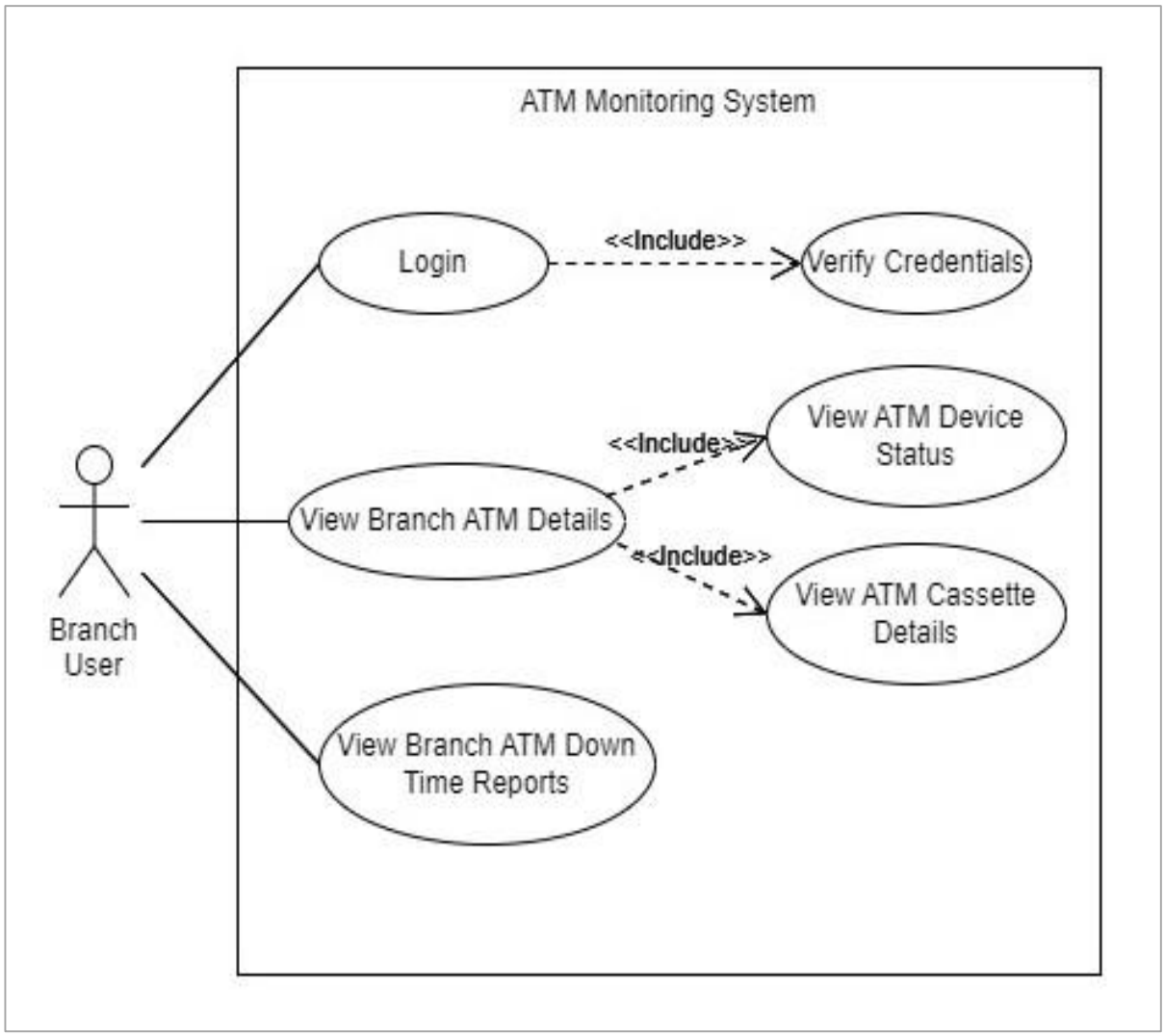


Figure 5- Use Case Diagram for Branch User

Use Case Diagram for Head Office User

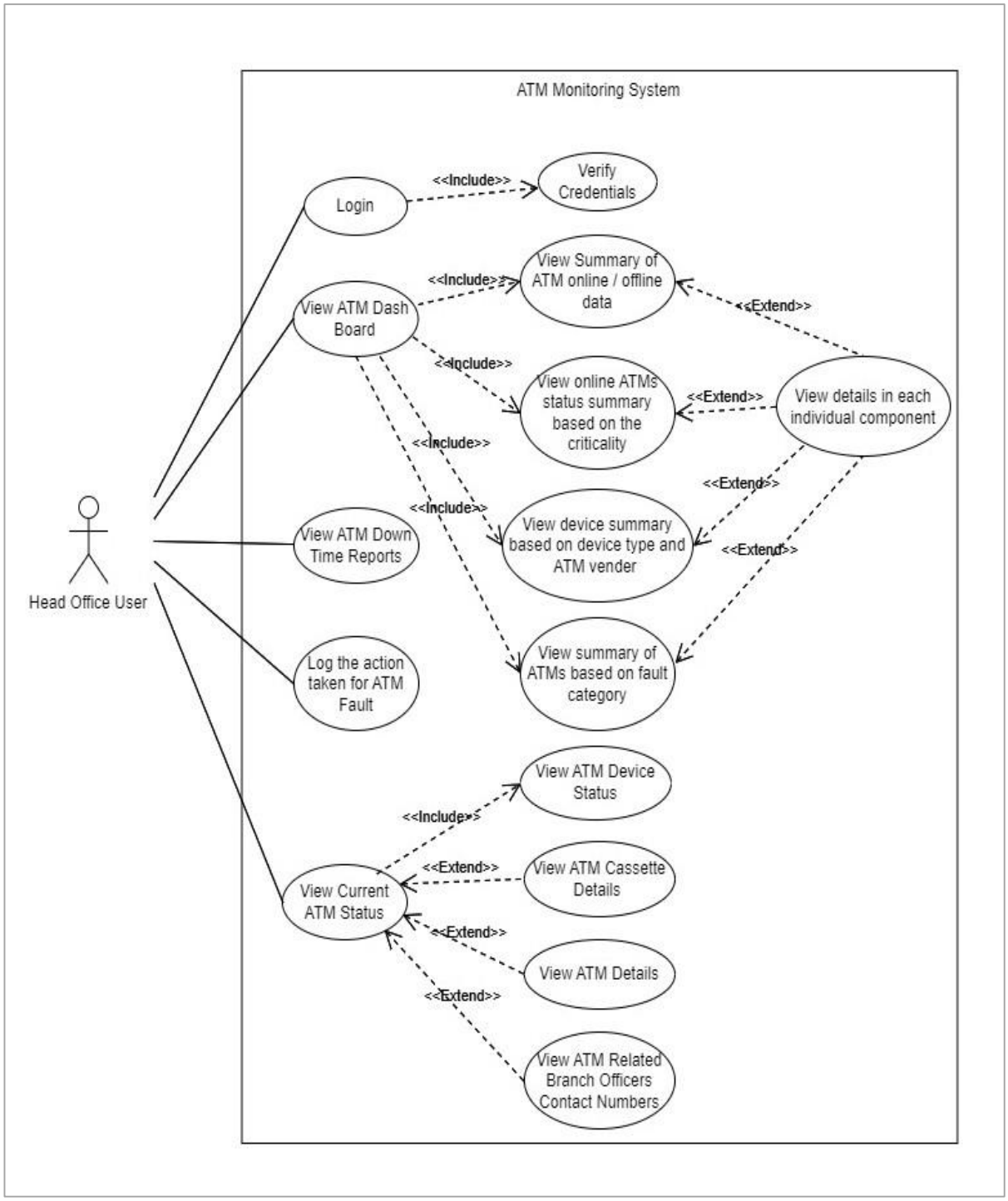


Figure 6- Use Case Diagram for Head Office User

3.3.2 Use Case Narrative

Use Case Narrative for Login

Use Case ID	UC_01
Use Case Name	Login to the system
Priority	High
Primary Actor	All users
Secondary Actors	None
Description	This use case allows user to login into the system by validating the user name and the password to access the relevant functions that they allowed in the system.
Preconditions	User has to have a valid account
Typical Course of Events	1. User enter the username and the password. 2. Click the "Login" button.
Alternative Course	Error message will display for the invalid username or password and redirected to the Login page
Post Condition	User will redirect to the Home page

Table 1- Use case narrative for login

Use Case Narrative for Create New User

Use Case ID	UC_02
Use Case Name	Create new user for the system
Priority	High
Primary Actor	Administrator
Secondary Actors	None
Description	This use case allows administrator to create a new user for the system and grant access to the system
Preconditions	Admin user has the privilege to create new users.
Typical Course of Events	1. User select “Create new system user” option 2. Enter the required data 3. Click on the “Save” button.
Alternative Course	1. If user not entered all the required fields in the form will display an error message 2. Error message will display if the user name is already in the system
Post Condition	Default user name will be users bank provident fund number (Ex – PFXXXXXX) and password will be the same. Users will ask to change the default password in the first login.

Table 2- Use case narrative for create new user

Use Case Narrative for Create Log for ATM Fault

Use Case ID	UC_03
Use Case Name	Create a log for ATM fault
Priority	High
Primary Actor	Head Office User
Secondary Actors	None
Description	This use case allows user to create a new log for a relevant ATM fault. This will create a new log for what is the issue, what are the action taken and what is the result after taking actions for an ATM machine
Preconditions	1. ATM names loaded into the ATM names drop down list 2. User has the access to create a new ATM fault entry
Typical Course of Events	1. User select “Create new ATM fault entry” option 2. Enter the required data 3. Click on the “Create New Entry” button.
Alternative Course	1. If user not entered all the required fields in the form will display an error message 2. After selecting the ATM name from the dropdown recent logs for selected ATM loaded into a separate table
Post Condition	Added request will display on the recent logs table

Table 3- Use case narrative for create log for ATM fault

Use Case Narrative for Search ATM Down Time Report

Use Case ID	UC_04
Use Case Name	Search ATM down time report
Priority	High
Primary Actor	Head Office User / Branch User
Secondary Actors	None
Description	This use case allows user to generate report for a selected date range. This report includes different faults occurring time and end times and how frequently it happens during the selected period.
Preconditions	User has the access to generate the report
Typical Course of Events	1. User select “ATM Down Time Report” option 2. Select the date range from the calendar. 3. Click on the “Search” button.
Alternative Course	1. If the selected date range is invalid, application will display an error message 2. If the data is not available for the selected date range system will give a message to the user.
Post Condition	If the data is available for the selected date range data will loaded in to a table and user has the option to download as in Excel format as well.

Table 4- Use case narrative for search ATM down time report

3.3.3 Sequence Diagrams

Sequence Diagram for Login

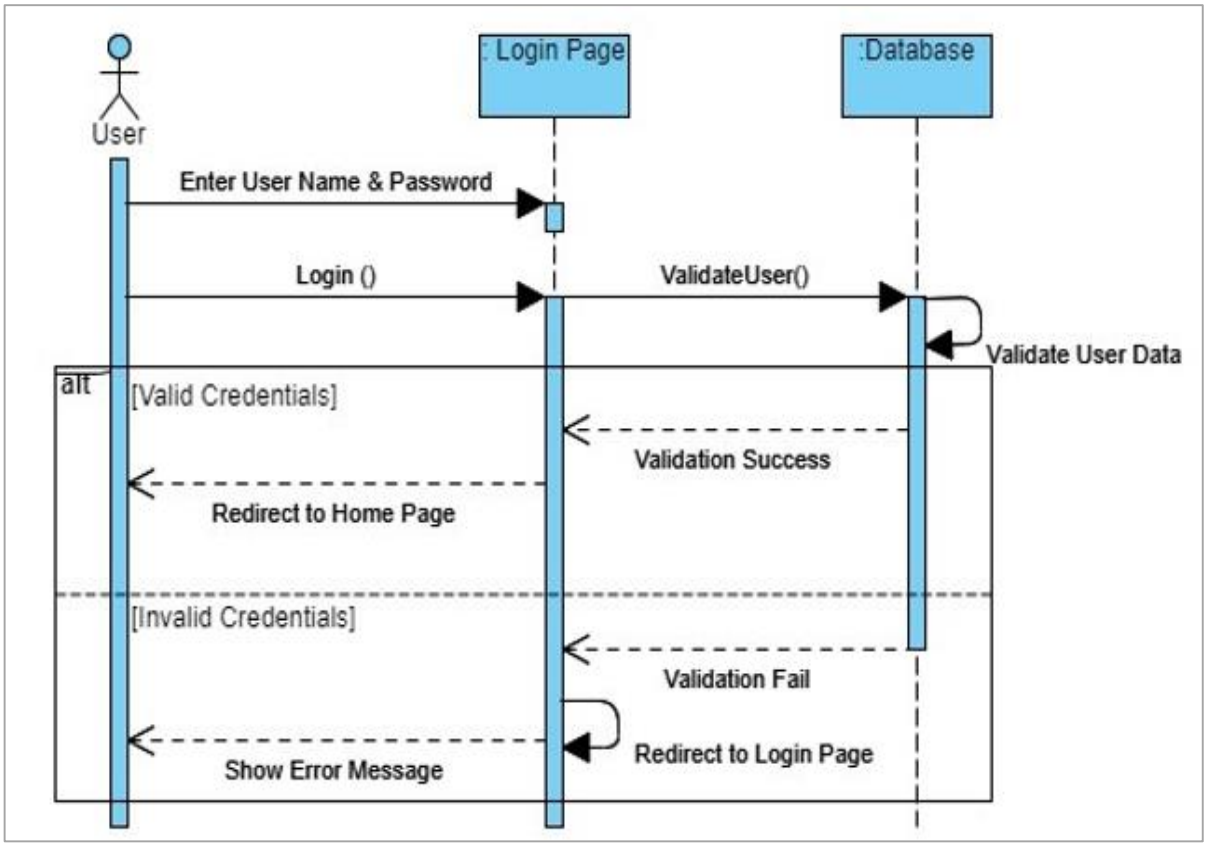


Figure 7- Sequence Diagram for Login

Sequence Diagram for Create New User

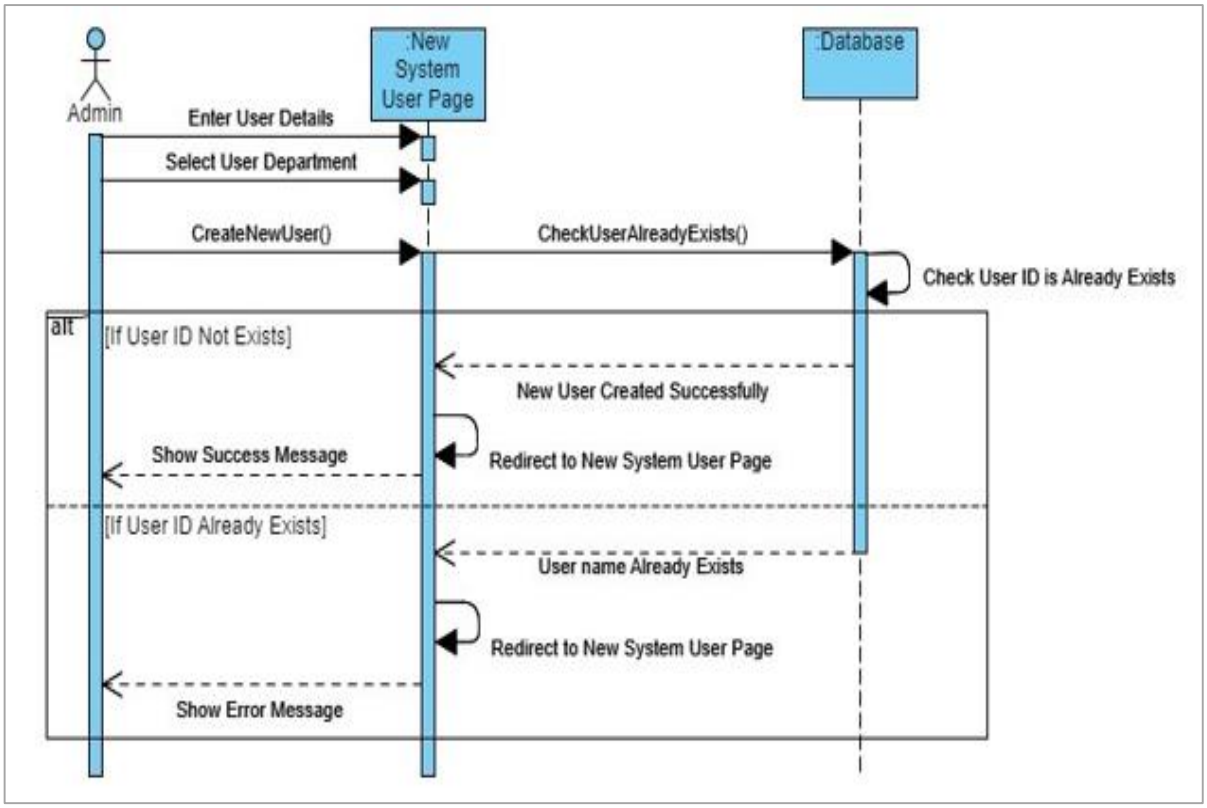


Figure 8- Sequence Diagram for Create New User

Sequence Diagram for Create ATM Fault Log

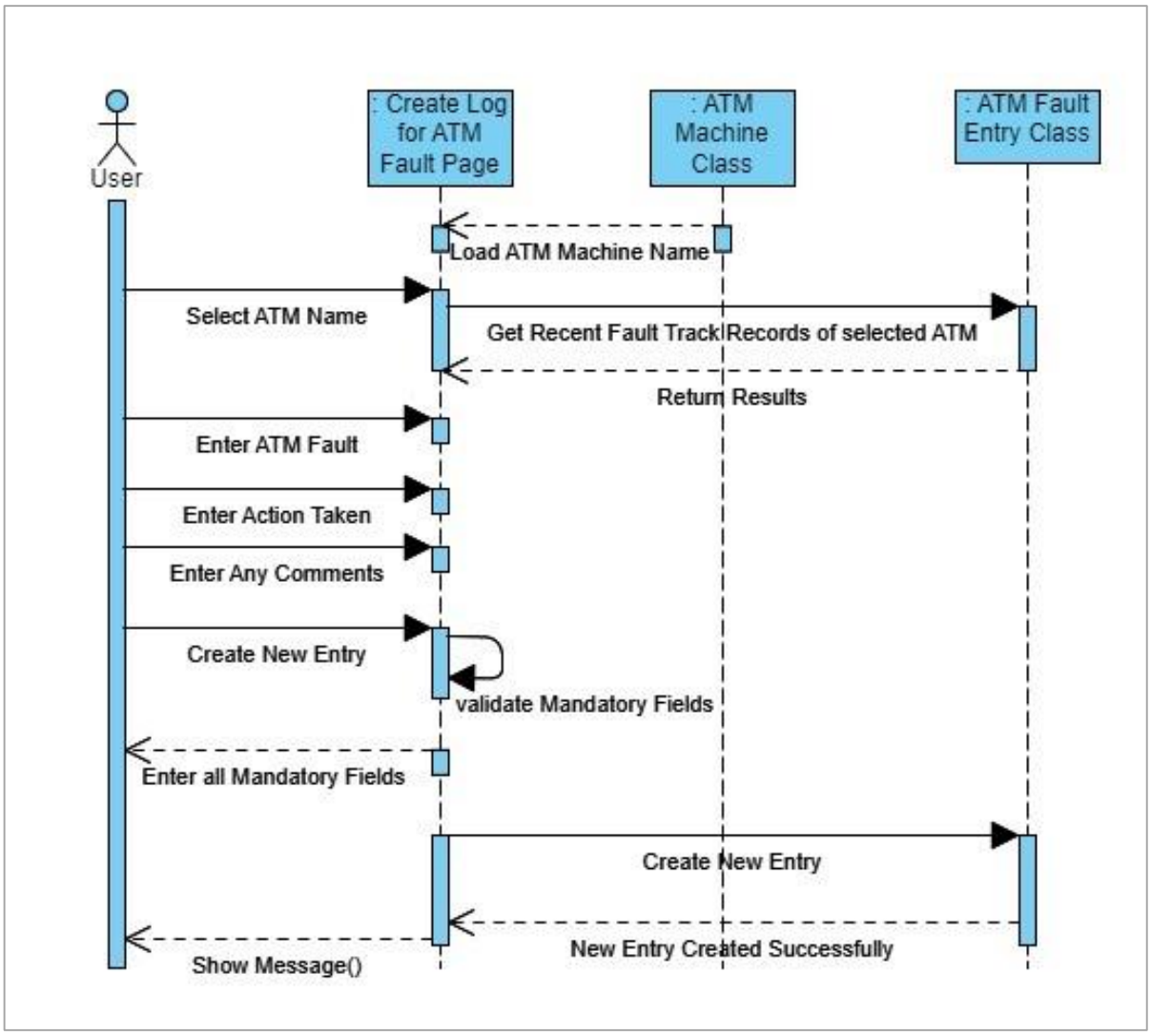


Figure 9- Sequence Diagram for Create ATM Fault Log

Sequence Diagram for Search ATM Downtime Report

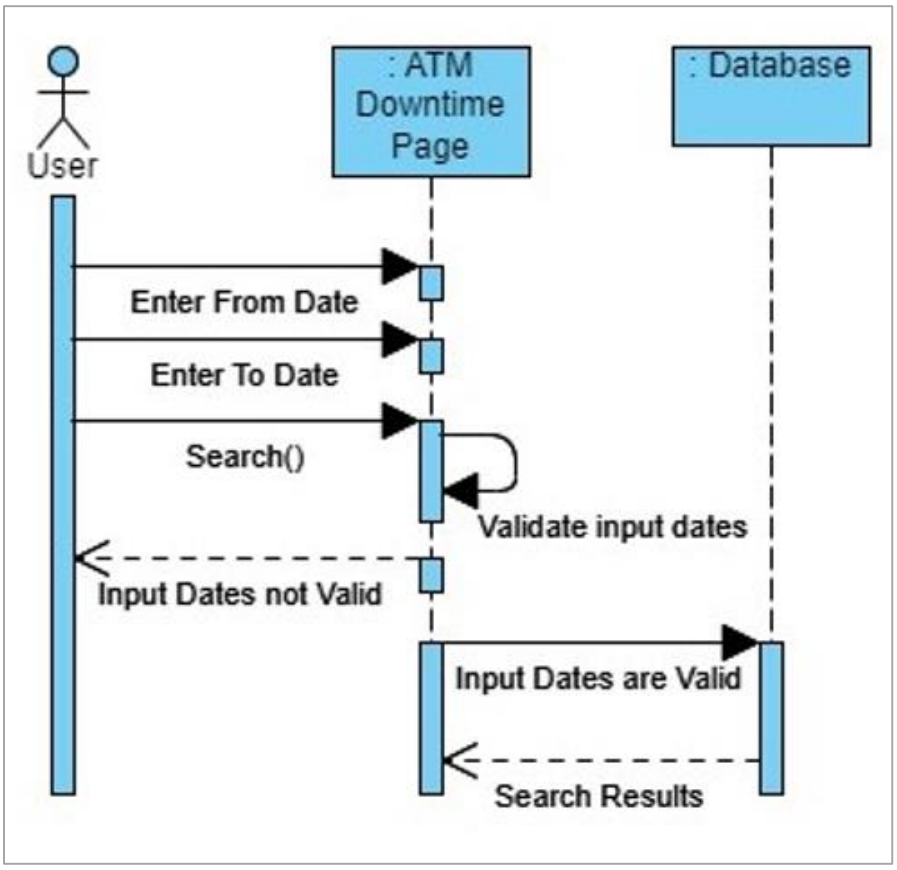


Figure 10- Sequence Diagram for ATM Downtime Report

3.3.4 Class Diagram

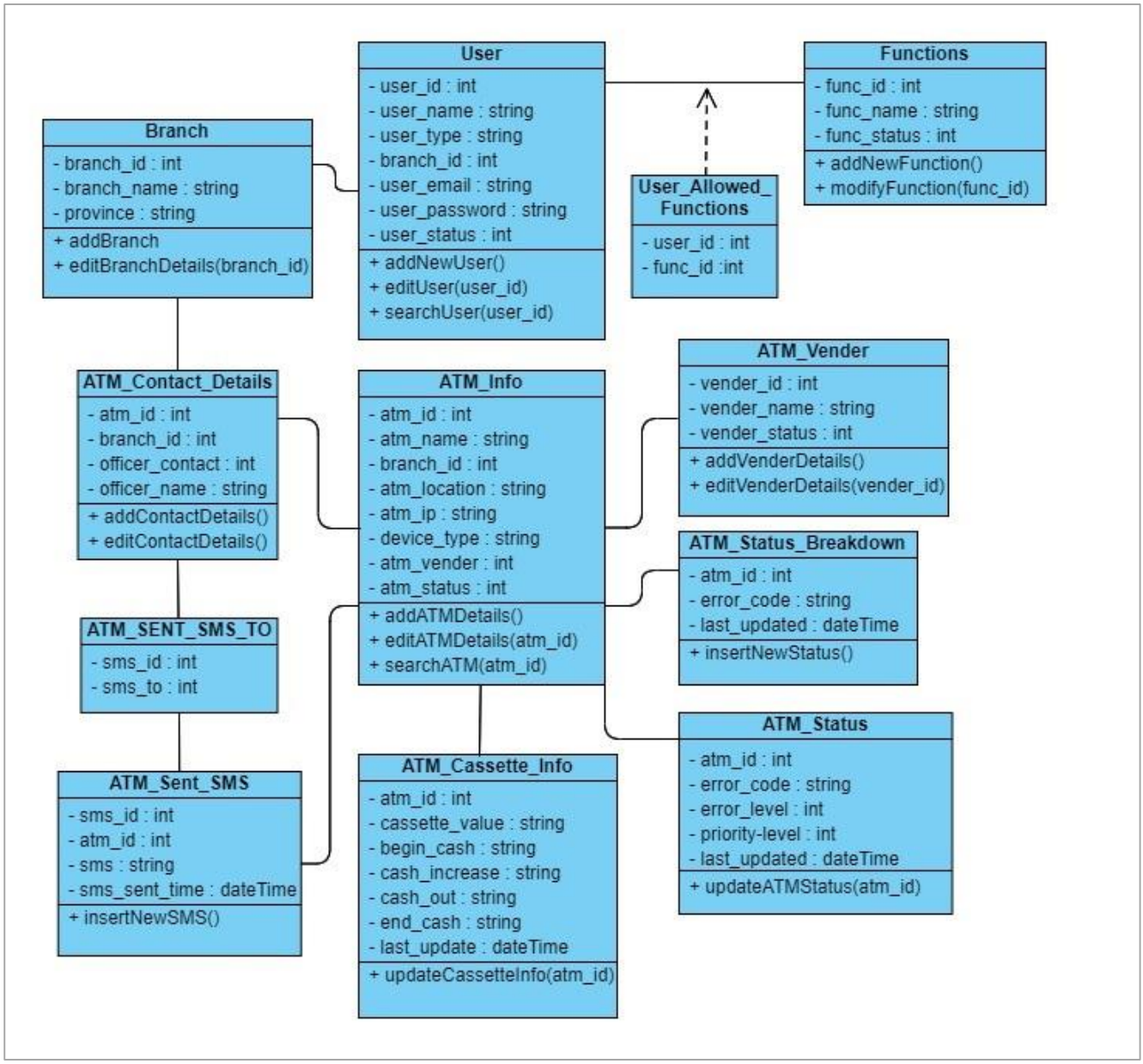


Figure 11- Class Diagram of the System

Chapter 4 – Methodology

The ATM Monitoring Dashboard System was developed using Apache NetBeans IDE 13 for both web application and web service development, providing a comprehensive environment for coding and testing. Additionally, Toad for Oracle was engaged for managing the Oracle database, facilitating tasks such as schema design, query execution, and maintenance. These tools collectively enhanced the development process, ensuring user-friendliness and efficient comprehension of the system's functionalities and implementation details.

4.1 Architecture of Implementation

As discussed in the Design Architecture chapter, frontend was written using HTML, CSS and JavaScript. Java and RESTful web service will be used to develop the backend of this application. In between AJAX is used to exchanging data with the web service.

Directory structure of the above mentioned technologies are shown below.

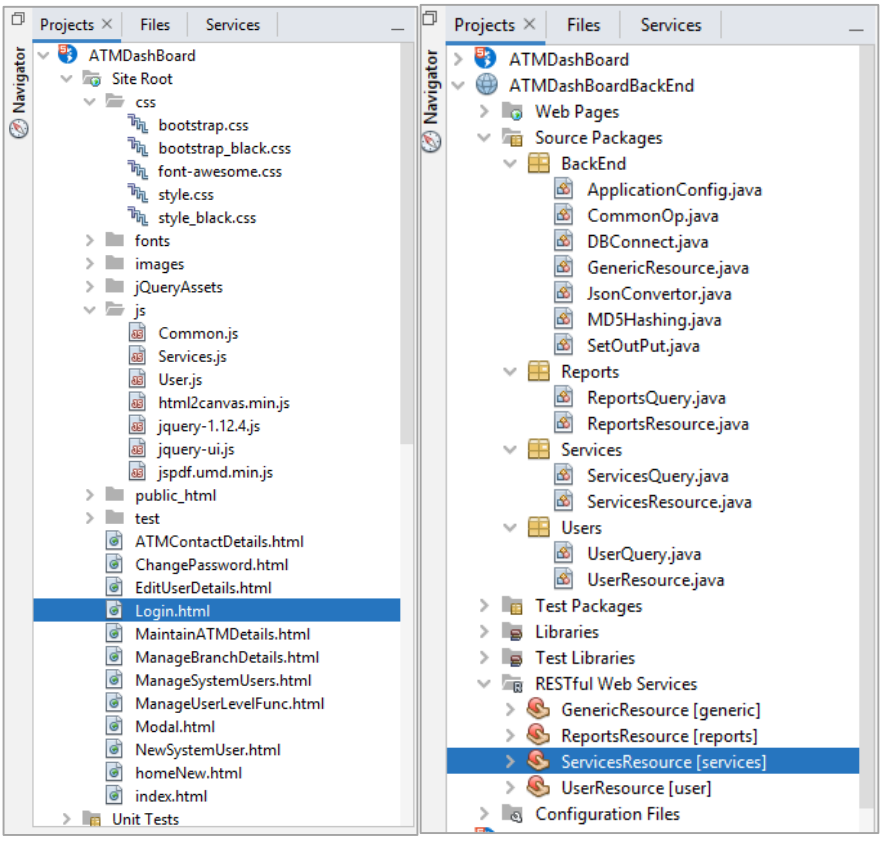


Figure 12- Folder structure of Web application and Web service

Web Application directory includes all the code of the developed web application including all the HTML files. Application folder includes few other folders which are described below.

- css - This folder contains all CSS files used for styling the HTML documents. The primary CSS file, often named styles.css and bootstrap.css are stored here.
- fonts - This folder contains font files used in the web application. Various font formats like TrueType (TTF), Web Open Font Format (WOFF), Embedded OpenType (EOT), and Scalable Vector Graphics (SVG) may be present to ensure compatibility across different browsers.
- images - This folder stores all image files used in the web application. Images referenced in CSS or HTML files are typically stored here.
- jQueryAssets – This folder stores all the external jQuery library folder.
- js - This folder contains both your custom JavaScript files (script.js) and the Ajax JavaScript file (ajax.js).

Below are the crucial folders within the web service project

- Source Package - This folder encapsulates the core Java source code files, organized within a package structure.
- Libraries - The libraries folder stores third-party libraries or dependencies utilized by the web service. These libraries enhance the functionality of the service by providing additional features or capabilities. They are typically added to the project as JAR files and managed within this folder.
- Restful Web Services - This folder include sub-directories or classes specifically dedicated to handling HTTP methods within the RESTful web services. These classes or packages are responsible for implementing the various HTTP methods supported by the RESTful endpoints, such as GET, POST, PUT, DELETE, etc.

```
@Context
private javax.servlet.http.HttpServletRequest reqGetSystemUsers;
@GET
@Path("getSystemUsers/")
@Produces(MediaType.APPLICATION_JSON)
public String GetSystemUsers() {
    sp.setSout();
    String Result = null;
    DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");
    LocalDateTime now = LocalDateTime.now();
    long beforeResp = 0;
    long afterResp = 0;
    UserQuery sqlOp = new UserQuery();
    try {
        beforeResp = System.currentTimeMillis();
        System.out.print("Get System Users - " + reqGetSystemUsers.getRemoteAddr()
                + "(" + dtf.format(now) + ")" + " Response Time - ");
        beforeResp = System.currentTimeMillis();
        Result = "{\"data\":" + sqlOp.getSystemUsers() + "}";
        afterResp = System.currentTimeMillis();
        System.out.println((afterResp - beforeResp) / 1000 + " seconds");
    } catch (Exception e) {
        System.out.println(e.toString());
    }
    sp.closeSout();
    return Result;
}
```

Figure 13- GET request handle in Web service

```
@Context
private javax.servlet.http.HttpServletRequest reqAddNewUserGroup;
@POST
@Path("addNewUserGroup/")
@Produces(MediaType.APPLICATION_JSON)
public String AddNewUserGroup(String ServiceRecordjasonobject) throws JSONException, FileNotFoundException {
    sp.setSout();
    String Result = null;
    DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");
    LocalDateTime now = LocalDateTime.now();
    long beforeResp = 0;
    long afterResp = 0;
    UserQuery sqlOp = new UserQuery();
    try {
        JSONObject jsonObj = new JSONObject(ServiceRecordjasonobject);
        beforeResp = System.currentTimeMillis();
        System.out.print("Create New User Group - " + reqAddNewUserGroup.getRemoteAddr()
                + "(" + jsonObj.getString("UGroup") + ")("
                + dtf.format(now) + ")" + " Response Time - ");
        Result = "{\"data\":" + sqlOp.addNewUserGroup(jsonObj) + "}";
        afterResp = System.currentTimeMillis();
        System.out.println((afterResp - beforeResp) / 1000 + " seconds");

    } catch (Exception e) {
        System.out.println(e.toString());
    }
    sp.closeSout();
    return Result;
}
```

Figure 14- POST request handle in Web service

4.2 AJAX (Asynchronous JavaScript and XML)

As discussed in the early chapters AJAX is used to exchanging data with the front end and the back end. In JavaScript, AJAX (Asynchronous JavaScript and XML) is a technique for exchanging data asynchronously between a web browser and a server without reloading the entire web page. It allows for dynamic updating of web content by sending HTTP requests in the background and processing the server's response without interrupting the user's interaction with the page. AJAX typically using the HTTP methods GET or POST. Upon receiving a response from the server, the client-side JavaScript code can update the HTML content, modify CSS styles, or execute additional JavaScript code based on the retrieved data. AJAX is widely used in web development to create responsive and interactive web applications that provide a seamless user experience by fetching and displaying data dynamically. Additionally, while the name AJAX suggests XML (Extensible Markup Language), modern implementations often use JSON (JavaScript Object Notation) for data interchange due to its lightweight and easy-to-parse nature. [12]

```
var GeneralPath = "http://localhost:8080/ATMDashBoardBackEnd/webresources/user/";
//Check Login
function checkLogin(arrayrecord_data) {
    var jason_data_object;
    try {
        var recordjasonobject = {"UName": arrayrecord_data[0], "Pwd": arrayrecord_data[1]};
        var servicerecordJSON = JSON.stringify(recordjasonobject);
        $.ajax({
            async: false,
            crossDomain: true,
            type: "POST",
            url: GeneralPath + "checkLogin/",
            contentType: "application/json",
            credentials: false,
            data: servicerecordJSON,
            success: function (response) {
                jason_data_object = response;
                var jdset = jason_data_object;
                if (jdset !== null) {
                    reurnLogin(jason_data_object);
                } else
                    window.alert("Error");
            },
            error: function () {
                alert('Internal Server Error!!!');            }
        });
    } catch (e) {
        alert(e);
    }
```

Figure 15- Sample AJAX code

4.3 Look and Feel of the Web Application

In order to maintain the look and feel of the web pages throughout the web application, common custom build template has been used for the application without using a freely available web template. Sample web page used in the application is shown below.

Web page consists of two main sections as header and the content section. Since the header was common for the each and every page in the web application. Only the content section was developed as a separate page and was loaded inside an iframe as shown in the below code segment.

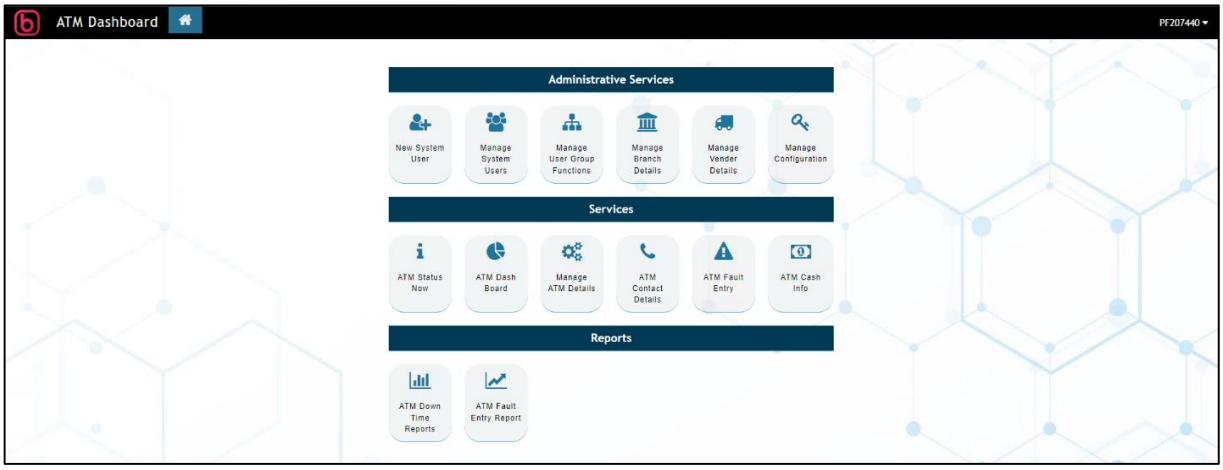


Figure 16- Main Interface

4.3.1 Modal Box

A modal is a dialog box or popup window that appears on top of the current page, typically used to display additional information, gather user input, or prompt for actions without navigating away from the underlying content.

```
<div class="topnav" id="myTopnav">
    <a href="#" style="text-decoration: none"><img src="images/bocweb.png" alt="Home" style="padding-right: 20px;"> ATM Unit </a>

    <a href="homeNew.html" class="active" id="0"
       onclick="document.getElementById('iframeID').src = this.getAttribute('href');
                return false;"><i class="fa fa-home" aria-hidden="true"></i></a>

    <div class="dropdown">
        <button class="dropbtn"><label id="uName"></label>
            <i class="fa fa-caret-down"></i>
        </button>
        <div class="dropdown-content">
            <a href="#" onclick="changePassword()">Change Password</a>
            <a href="#" onclick="logoutFun()">Logout</a>
        </div>
    </div>

    <a href="javascript:void(0);" class="icon" onclick="myFunction()">&#9776;</a>
</div>

<div class="container">
    <iframe id = "iframeID" class="responsive-iframe" src="homeNew.html"></iframe>
</div>
```

Figure 17- iframe code

In the system, Bootstrap modal boxes are employed to effectively manage error messages, confirmation prompts, and successful messages. These modal components ensure clear communication with users by presenting descriptive titles and messages, along with appropriate icons or colours to signify different message types. Error modals alert users about invalid inputs or system errors, confirmation modals prompt users for critical actions, while success modals inform users about completed tasks. Each modal type features actionable buttons tailored to the specific context, enhancing user experience and facilitating interaction within the system.

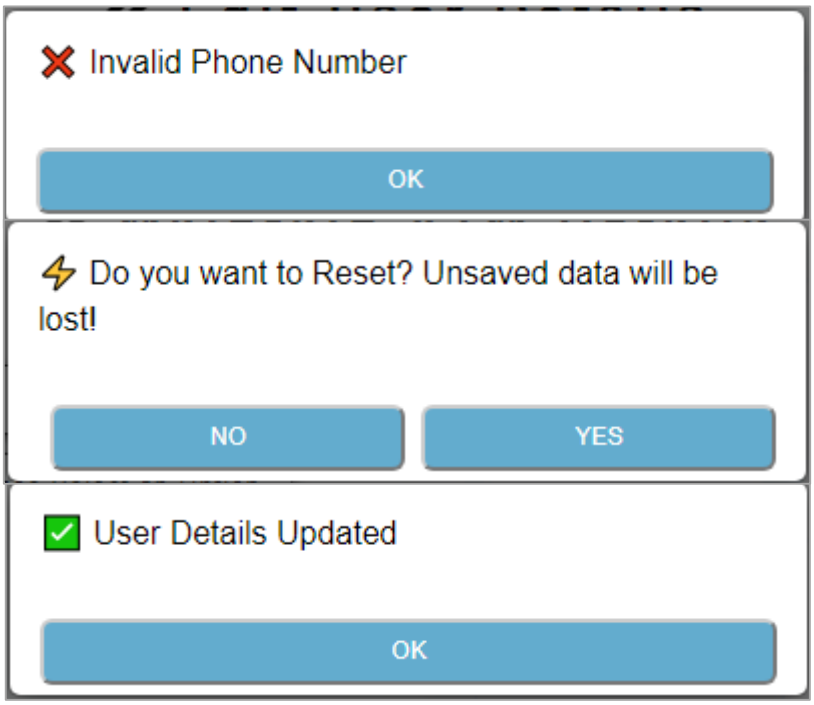


Figure 18- Sample error messages

Below sample code defines an error message model, incorporating a structured approach for handling error messages within the web application

```
<!-- Modal -->
<div class="modal fade" id="myModal" role="dialog" data-backdrop="static" data-keyboard="false">
    <div class="modal-dialog">
        <!-- Modal content-->
        <div class="modal-content" id="modalContent">
            <div class="modal-body">
                <p id="modal_body"></p>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-default" data-dismiss="modal" id="okbtn">OK</button>
            </div>
        </div>
    </div>
</div>
<!-- Modal Confirm -->
<div class="modal fade" id="myModalConfirm" role="dialog" data-backdrop="static" data-keyboard="false">
    <div class="modal-dialog">
        <!-- Modal content-->
        <div class="modal-content" id="modalContentConfirm">
            <div class="modal-body">
                <p id="modal_body_confirm"></p>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-default" data-dismiss="modal">NO</button>
                <button type="button" class="btn btn-default" data-dismiss="modal">YES</button>
            </div>
        </div>
    </div>
</div>
```

Figure 19- Modal Code

4.3.2 Security Controls

To safeguard against unauthorized access and various security threats, robust security controls were implemented throughout the development of the web-based application. These controls encompassed a comprehensive range of measures including authentication mechanisms, access controls, encryption protocols and input validation

- Encryption Protocols

  In the "ATM Monitoring Dashboard System," security measures include encrypting and storing passwords in the database using the MD5 algorithm, a one-way hashing algorithm. This means that once a password is hashed using MD5, it cannot be decrypted to retrieve the original plain text. Instead, MD5 generates a fixed-size hash value (typically a 32-digit hexadecimal string) regardless of the input size. This hash value serves as a unique representation of the password and is stored securely in the database. By employing MD5 hashing, the system enhances password security by protecting sensitive information from unauthorized access or decryption attempts.

- Input Validation

  In the system, comprehensive input validation measures have been implemented to ensure data integrity and security. This includes validation for various scenarios such as empty values, invalid email formats, invalid mobile numbers, and text inputs intended for numeric values only. For empty values, the system checks if the input fields are blank or contain only whitespace characters, preventing submission if the input is incomplete. Additionally, email validation employs regular expressions to verify if the provided email address conforms to a standard format, while mobile number validation ensures that phone numbers adhere to specified formats or conventions. Moreover, text inputs intended for numeric values are restricted to accept only numerical characters, effectively preventing the input of non-numeric data. These validation mechanisms collectively contribute to a robust system architecture, safeguarding against erroneous or malicious data inputs and enhancing overall data reliability and security.

- Access Control

  In the system, Role-Based Access Control (RBAC) has been implemented as a key access control mechanism. RBAC is a widely used approach that assigns permissions to users based on their roles within the organization or system. In the context of the system, RBAC simplifies access management by categorizing users into predefined roles, each with its associated set of permissions. Administrator define roles based on job responsibilities or functional requirements, and users are assigned to appropriate roles. Permissions are then granted to roles rather than individual users, reducing the complexity of access control management and ensuring consistent enforcement of security policies.

4.4 ATM Simulator

In the context of the BOC environment, the essential functionality of the ATM dashboard relies on data acquired from Base 24 system files. Unfortunately, access to the Base 24 file within the live environment at BOC is restricted, presenting a significant constraint during the implementation of this project. As a workaround to address this challenge, separate standalone application to simulate the actual ATM machine, has been developed using Java language. This simulator implements such a way that it simulate/reproduce ATM errors and behaviours, providing a testing environment. With this tool, can refine the dashboard's functionality without needing direct access to Base 24 files. This approach streamlines development and ensures the dashboard is well-prepared for real-world scenarios.

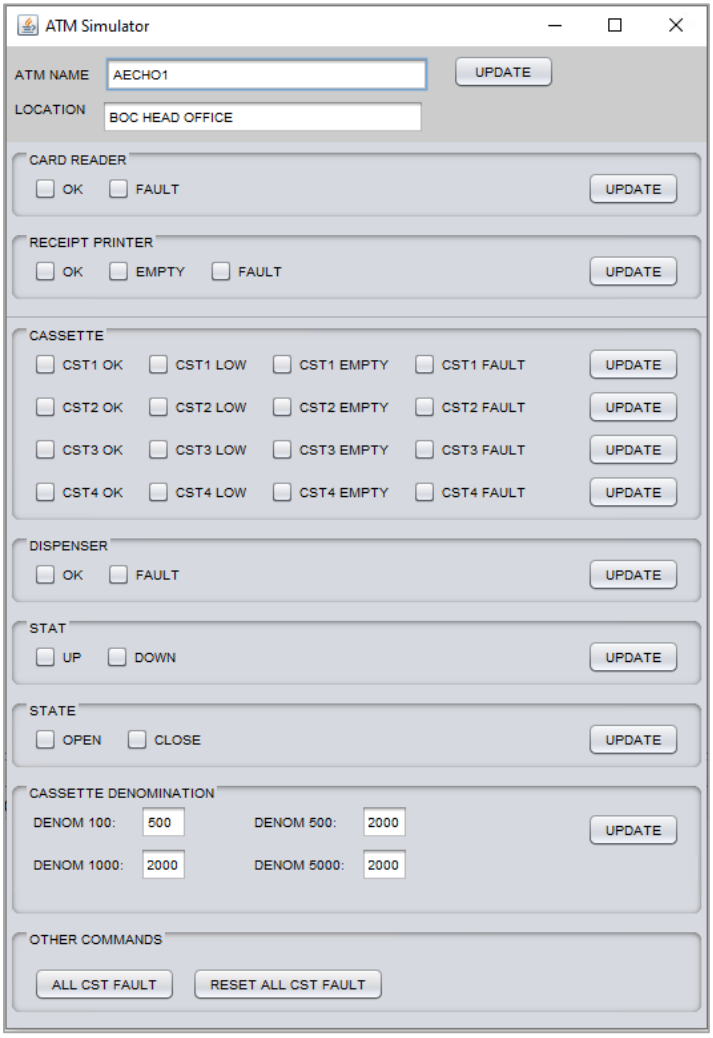


Figure 20- ATM Simulator

Chapter 5 – Testing and Evaluation

5.1 Introduction

Testing is a critical phase aimed at ensuring the developed application meets quality benchmarks and aligns with customer specifications. It involves thorough validation to identify and address any potential issues that could compromise performance or overall quality. This chapter outlines a comprehensive testing plan for the developed web application, concluding in an evaluation process to verify the achievement of project objectives.

5.2 System Testing

For the creation and execution of high-quality software applications, a robust testing plan is vital to identify and rectify issues prior to deployment in a production environment. As part of this strategy, unit testing is undertaken during the development phase, where critical components are methodically isolated and analysed to ensure the accuracy and reliability of their inputs and outputs.

In the testing phase of the waterfall model, integration testing was chosen to bring together individual pieces of the software and check how they function as a whole. This testing focused on ensuring that the interfaces between these parts were well-connected and functioned properly once integrated. It helped identify any issues or inconsistencies that arose when merging these components, allowing for adjustments to be made before moving forward.

Therefore, as per the test plan, started by testing each component separately. Once unit testing is finished, moved on to integration testing. This involves gradually combining the components, one by one, until they're all integrated. This step ensures that the different parts of the system work together smoothly and effectively.

5.2.1 Test Cases

Below are the sample test cases used for evaluating the ATM Monitoring Dashboard System.

- Test Cases for Login

Test ID	Test Case Summary	Pre- Conditions	Test Case Steps	Expected Results
1.1	Test Login function with valid credentials	Login page is opened	1. Enter valid username and password a. User b. Admin	User should login successfully
1.2	Check login with invalid username	Login page is opened	1. Enter invalid username a. Empty b. Incorrect	User shouldn't be able to login
1.3	Check login with invalid password	Login page is opened	1. Enter invalid password a. Empty b. Incorrect	User shouldn't be able to login
1.4	Check logout Function	User is logged into the system	1. Click on Logout button	User should logout successfully
1.5	Change Password	User is logged into the system	1. Select Change Password option 2. Fill the fields as per requirement and click on Change button.	Chang Password screen should be opened. Password should be changed successfully

Table 5- Test cases for login

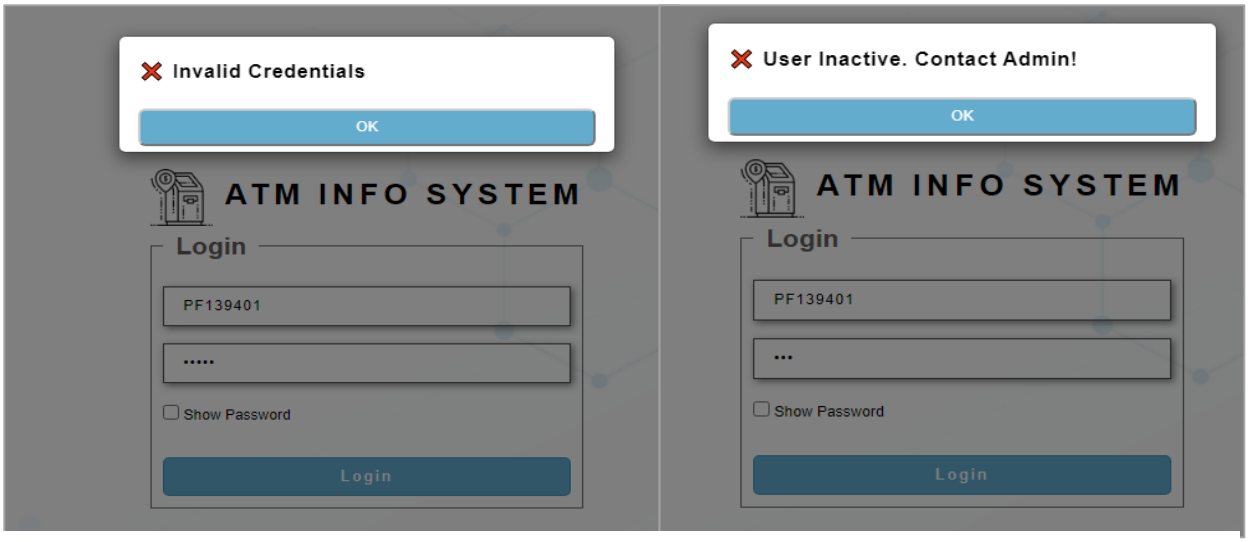


Figure 21- Validation in Login page

- Test Cases for New System User

Test ID	Test Case Summary	Pre- Conditions	Test Case Steps	Expected Results
2.1	Test 'Add New User' function	Create New System User screen is opened	1. Enter all the mandatory fields and no mandatory fields if necessary. 2. Click on Add New button.	New user should be added successfully
2.2	Validate for empty values for mandatory fields	Create New System User screen is opened	1. Try to proceed with empty values for mandatory fields	User should not be able to proceed
2.3	Create new user profile with already registered PF number	Create New System User screen is opened	1. Enter a PF number which is already registered. 2. Enter all the necessary details accurately. 3. Click on Add New button.	Should display an error message
2.4	Validate for invalid input for email address	Create New System User screen is opened	1. Enter all the mandatory fields 2. Enter invalid email address	Should display 'Invalid Email'.

2.5	Validate for invalid input for phone number	Create New System User screen is opened	1. Enter all the mandatory fields 2. Enter invalid phone number	Should display 'Invalid Phone number'

Table 6- Test cases for New system user

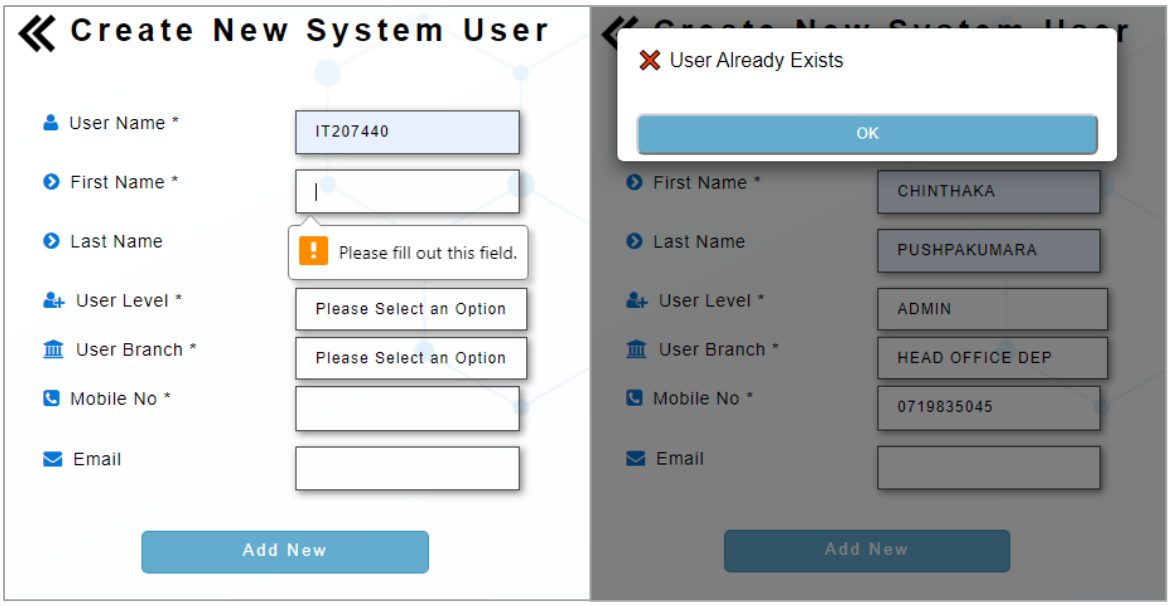


Figure 22- Validation in New system user

5.2.2 Testing Status

Test ID	Description	Pass / Fail
1.1	Test Login function with valid credentials	Pass
1.2	Check login with invalid username	Pass
1.3	Check login with invalid password	Pass
1.4	Check logout Function	Pass
1.5	Change Password	Pass
2.1	Test add new user function	Pass
2.2	Validate for empty values for mandatory fields	Pass
2.3	Create new user profile with already registered PF number	Pass
2.4	Validate for invalid input for email	Pass
2.5	Validate for invalid input for phone number	Pass

Table 7- Status of test cases

Detail test plan is mentioned in the Appendix A with all the other test cases and test results

5.3 System Evaluation

To assess the ATM Monitoring Dashboard System, a simple approach was taken. About 28 staff members from various user groups who tested the system were asked to fill out a questionnaire. The questionnaire focused on two aspects: the system's visual appeal and its functionality. By collecting feedback in this manner, the aim was to gain insights into users' perceptions of the system's appearance and performance, with the ultimate goal of pinpointing areas for improvement to enhance the overall user experience.

A questionnaire was created using Google Forms, containing multiple questions related to the developed web application. Users were asked to rate each question on a scale from Very Poor to Excellent, with five levels corresponding to ratings from 1 to 5. This numerical rating system was chosen to facilitate easy data analysis and evaluation of user feedback. By gathering ratings for each question, the aim was to assess various aspects of the web application's performance and user satisfaction levels effectively.

ATM Monitoring Dashboard Feedback Form

We value your feedback to help us improve our ATM monitoring dashboard. Please take a few moments to share your thoughts with us.

99.dewa@gmail.com Switch account

Not shared

* Indicates required question

Name *

Your answer

User Group *

Choose

Email

Your answer

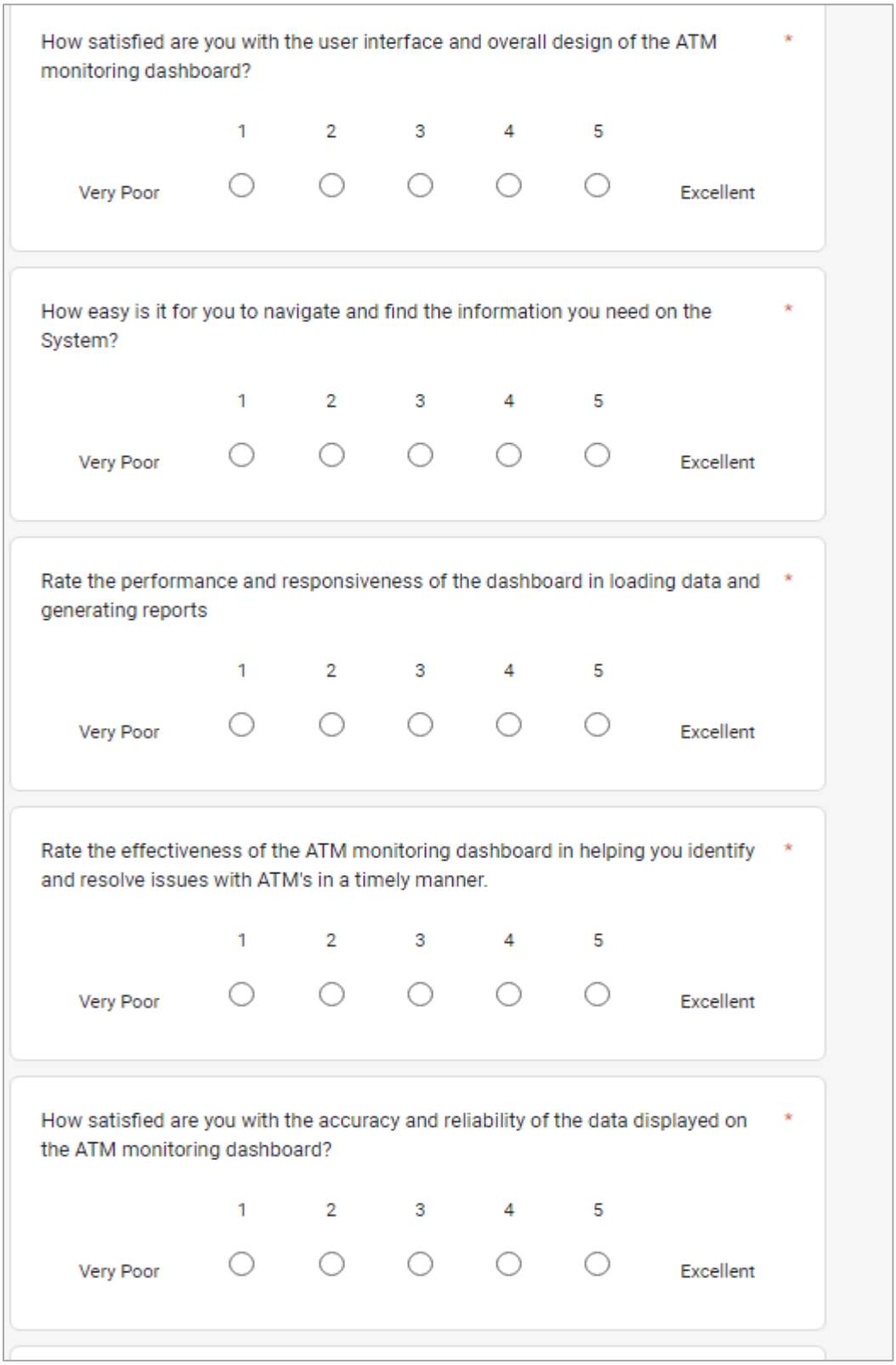
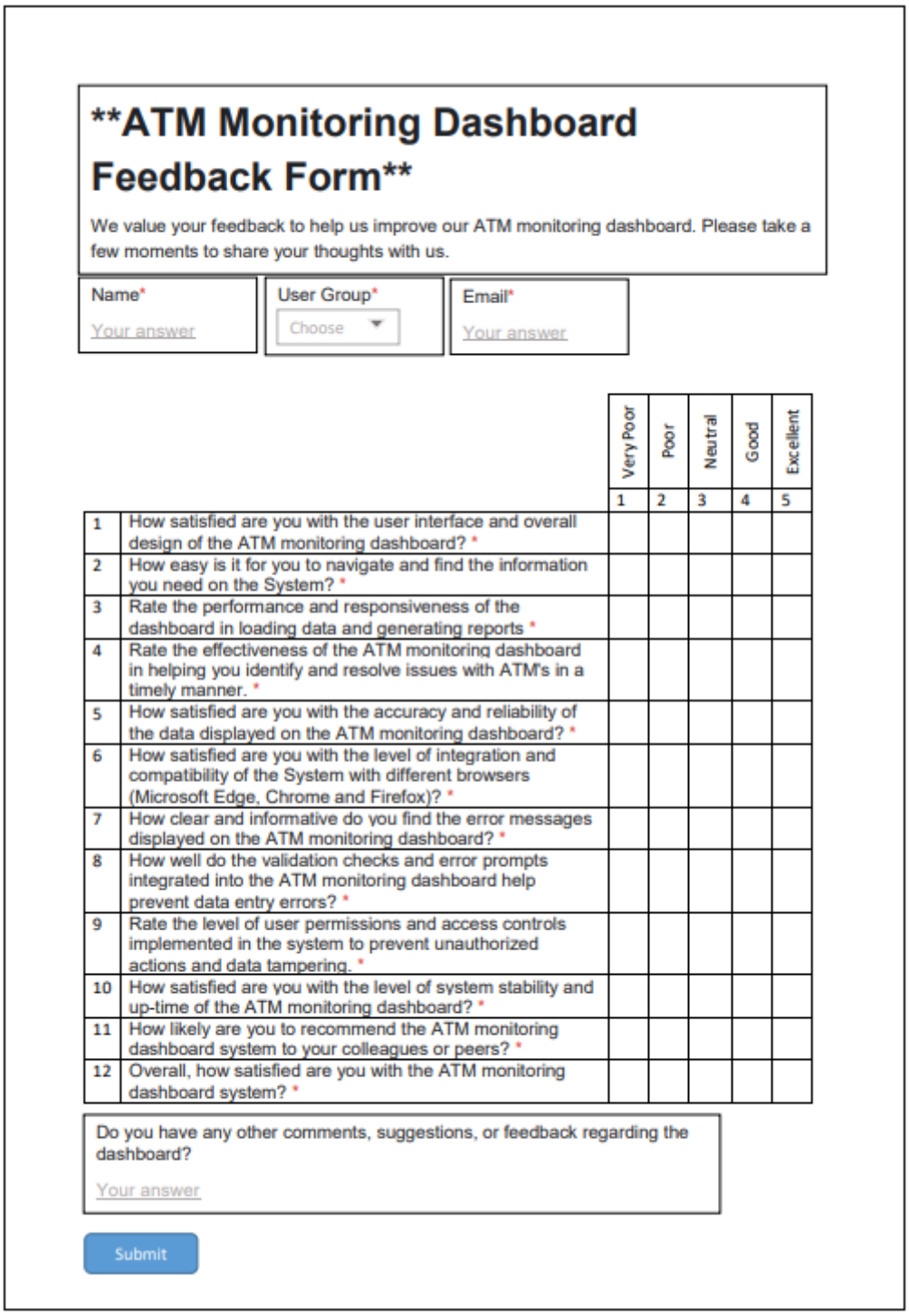


Figure 23- Google Form

These are the questions given in the Google Form, which has been created to collect feedback of the new system.



ATM Monitoring Dashboard Feedback Form

We value your feedback to help us improve our ATM monitoring dashboard. Please take a few moments to share your thoughts with us.

Name* Your answer

User Group* Choose

Email* Your answer

		Very Poor	Poor	Neutral	Good	Excellent
		1	2	3	4	5
1	How satisfied are you with the user interface and overall design of the ATM monitoring dashboard? *					
2	How easy is it for you to navigate and find the information you need on the System? *					
3	Rate the performance and responsiveness of the dashboard in loading data and generating reports *					
4	Rate the effectiveness of the ATM monitoring dashboard in helping you identify and resolve issues with ATM's in a timely manner. *					
5	How satisfied are you with the accuracy and reliability of the data displayed on the ATM monitoring dashboard? *					
6	How satisfied are you with the level of integration and compatibility of the System with different browsers (Microsoft Edge, Chrome and Firefox)? *					
7	How clear and informative do you find the error messages displayed on the ATM monitoring dashboard? *					
8	How well do the validation checks and error prompts integrated into the ATM monitoring dashboard help prevent data entry errors? *					
9	Rate the level of user permissions and access controls implemented in the system to prevent unauthorized actions and data tampering. *					
10	How satisfied are you with the level of system stability and up-time of the ATM monitoring dashboard? *					
11	How likely are you to recommend the ATM monitoring dashboard system to your colleagues or peers? *					
12	Overall, how satisfied are you with the ATM monitoring dashboard system? *					

Do you have any other comments, suggestions, or feedback regarding the dashboard?

Your answer

Submit

Figure 24- Questions given in the Google Form

5.3.1 Analysis of the Results

28 staff members were participated from four distinct user groups to participate in the evaluation process of the ATM Monitoring Dashboard System. This participation includes 11 members from the IT division who used the system, 9 members from the branch users,5 members from computer room operator staff and 3 members from branch operations department.

By involving representatives from each group, the evaluation process ensures a comprehensive assessment, encompassing diverse perspectives and requirements, thereby facilitating the identification of strengths, weaknesses, and areas for improvement to enhance overall system effectiveness and user satisfaction.

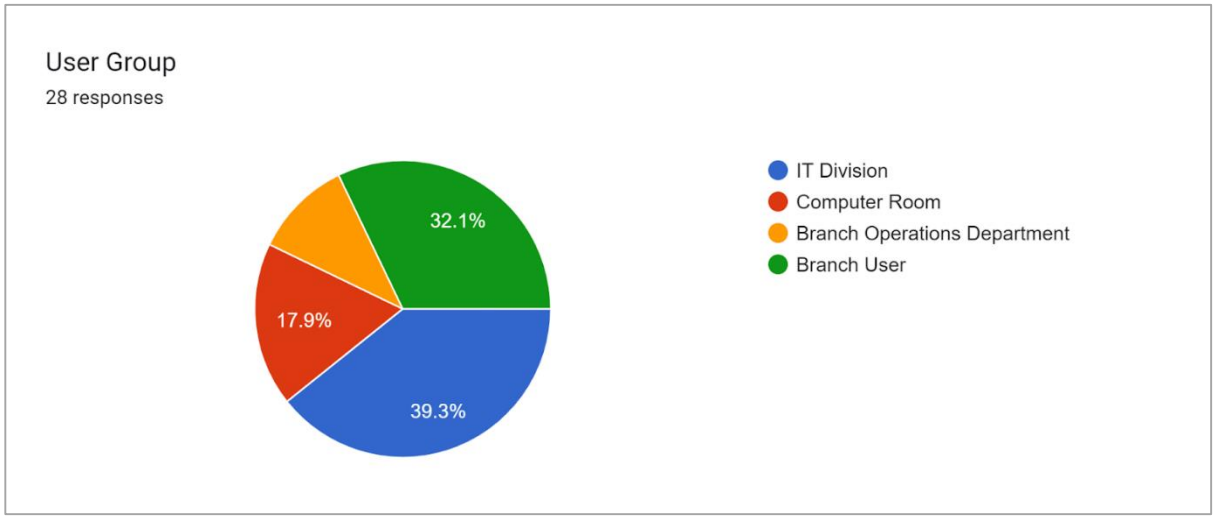


Figure 25- User group wise feedback summary

Below are the responses received for each questions from the users who participated in the evaluation process.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
User1	5	4	3	4	4	3	4	3	3	4	4	4
User2	4	5	4	5	5	3	5	5	4	3	4	5
User3	5	5	5	5	5	4	4	5	4	5	5	5
User4	4	4	3	4	4	3	2	3	4	4	4	4
User5	4	5	5	4	5	4	4	4	5	5	5	5
User6	5	5	5	5	5	5	5	5	5	5	5	5
User7	4	4	4	4	4	4	4	4	4	4	4	4
User8	3	3	3	3	3	3	3	3	3	3	3	3
User9	3	3	4	3	3	3	3	4	4	4	4	3
User10	3	4	3	4	3	5	4	4	3	5	3	3
User11	5	5	5	5	5	5	5	5	5	5	5	5
User12	5	5	5	5	5	5	5	5	5	5	5	5
User13	5	4	5	4	5	4	5	5	4	4	5	4
User14	5	5	5	5	5	5	5	5	5	5	5	5
User15	5	5	5	5	5	5	5	5	5	5	5	5
User16	4	4	4	4	3	3	4	3	3	4	4	4
User17	5	5	4	5	5	4	5	5	5	5	5	5
User18	5	5	5	5	5	5	5	5	5	5	5	5
User19	4	4	5	4	5	5	5	5	5	5	5	5
User20	4	4	4	4	5	5	5	5	5	5	5	5
User21	3	3	3	3	4	3	3	3	3	3	3	3
User22	5	4	5	4	4	5	5	5	5	5	5	5
User23	4	4	5	5	5	5	5	5	5	5	5	5
User24	5	5	4	5	5	4	5	3	5	4	5	5
User25	4	5	3	5	5	4	5	5	4	3	5	4
User26	5	5	5	3	3	5	5	4	4	5	5	5
User27	4	4	4	3	3	4	3	3	4	4	3	4
User28	3	2	3	2	4	3	2	3	3	3	3	3
Average	4.29	4.29	4.21	4.18	4.36	4.14	4.29	4.25	4.25	4.36	4.43	4.39

Table 8- Evaluation of Results

After collecting and organizing the raw data, proceeded to prepare graphs for in-depth analysis and evaluation of the results. Figure 25 illustrates the average rating (on a scale of 1 to 5) obtained for each question, providing a comprehensive overview of user feedback. Meanwhile, Figure 24 presents a detailed breakdown of responses, showcasing the distribution of ratings on the same scale for individual questions. These graphical representations serve as valuable tools for discerning user perspectives, identifying trends, and informing strategic decision-making to optimize system performance and user satisfaction.

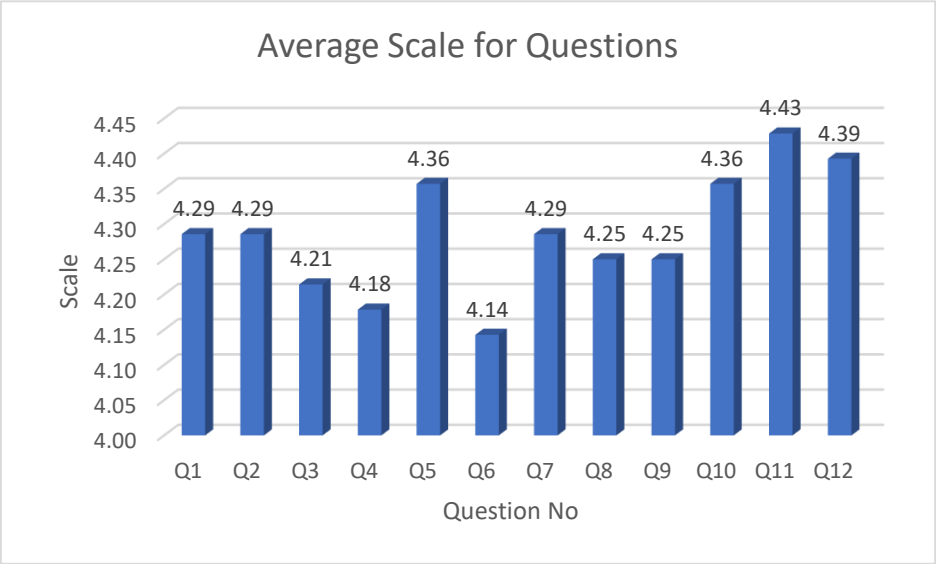
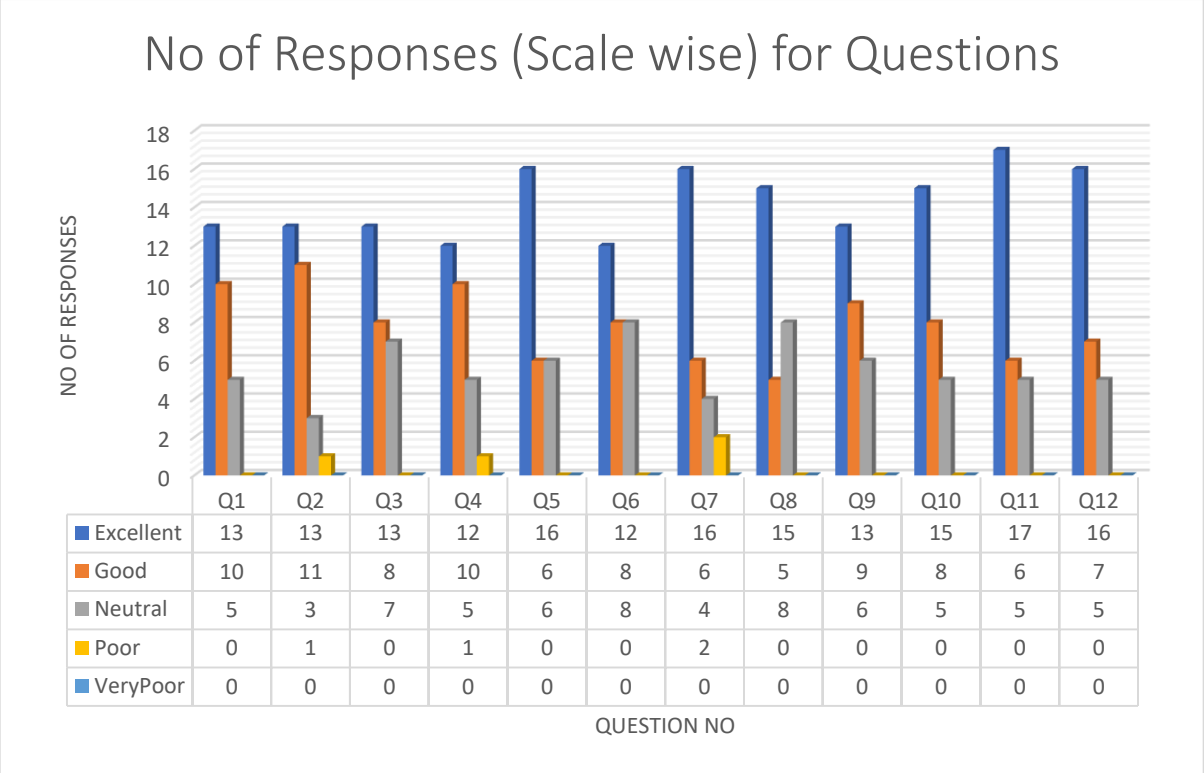


Figure 27- Average scale for questions



	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
Excellent	13	13	13	12	16	12	16	15	13	15	17	16
Good	10	11	8	10	6	8	6	5	9	8	6	7
Neutral	5	3	7	5	6	8	4	8	6	5	5	5
Poor	0	1	0	1	0	0	2	0	0	0	0	0
VeryPoor	0	0	0	0	0	0	0	0	0	0	0	0

Figure 26-Scale wise responses for questions

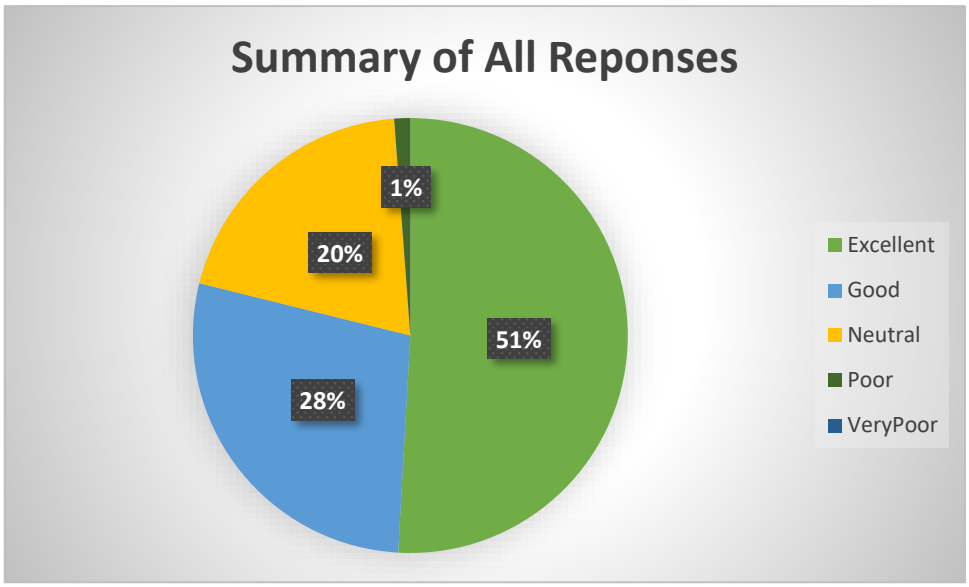


Figure 28- Summary of All Reponses

Based on the analysis of the figures and data, the feedback regarding the overall system indicates that a significant majority of the population, constituting nearly 80%, have rated the system as either good or excellent. Conversely, a minority of users, comprising less than 20%, provided a neutral rating. This minority group expressed concerns primarily related to error messages and response time. It is noteworthy that the evaluation of the system during the assessment period yielded successful outcomes, reflecting the effectiveness and satisfaction levels among the majority of users.

Chapter 6 – Conclusion

As outlined in the initial chapter, the primary aim of the project was to introduce a web-based 'ATM Monitoring Dashboard System' to facilitate real-time monitoring of the status and performance of all ATM/CDM/CRM machines al around the country. Therefore, the success of the system hinges upon its ability to address the limitations inherent in the existing processes. In other words, the effectiveness of the developed solution is gauged by its capacity to overcome the challenges and deficiencies present in the current monitoring methods. By providing a centralized platform for monitoring and management, the system aims to enhance efficiency, accuracy, and accessibility in monitoring ATM-related operations across the country. Hence, the degree of success is measured by the extent to which the system streamlines processes, improves decision-making capabilities, and enhances overall operational effectiveness in the monitoring of ATM infrastructure.

The project presented significant challenges and required considerable time and effort to implement a wide array of features within a constrained timeframe. Despite these obstacles, it has effectively met all Functional and Non-functional Requirements of the system. The system's design, with a focus on scalability, enables seamless integration of additional features in the future. Ultimately, the system is user-friendly and aligns with operational requirements, effectively meeting the needs of users and stakeholders. In summary, it successfully accomplishes its objective of introducing an innovative web-based 'ATM Monitoring Dashboard System'.

6.1 Future Work

While the current version of the application fulfils its intended purpose with all the necessary features, there is ongoing consideration for future development. This includes the addition of new functionalities and enhancements to further improve the user experience and meet evolving needs.

The future roadmap for enhancing this system includes the integration of transaction monitoring capabilities. This significant addition will allow for real-time tracking and analysis of transactions across ATM/CDM/CRM machines. By implementing transaction monitoring, the system will offer comprehensive oversight and analysis, enabling proactive detection of anomalies and potential security threats.

List of References

[1] Bank *of Ceylon - About Us: Bankers to the nation* (no date) *Bank of Ceylon - About Us | Bankers to the Nation*. Available at: https://www.boc.lk/about (Accessed: 20 June 2023).

[2] *VynamicTM view: Diebold Nixdorf* (no date) *Diebold Nixdorf - VynamicTM View (Availability)*. Available at: https://www.dieboldnixdorf.com/en-us/banking/portfolio/software/view/ (Accessed: 21 February 2023).

[3] *NCR APTRA VISION* (2015) *Hansab Global*. Available at: https://www.hansab.com/en/ncr-aptra-vision (Accessed: 21 February 2023).

[4] *NCR APTRA VISION* (2015) *Hansab Global*. Available at: https://www.hansab.com/en/ncr-aptra-vision (Accessed: 21 February 2023).

[5] *ATM Solutions* (no date) *Fiserv*. Available at: https://www.fiserv.com/en/solutions/payments/atm-solutions.html (Accessed: 21 February 2023).

[6] *Fiserv device manager: ATM & Self-service monitoring and management - fiserv: ATM Marketplace* (no date) *Fiserv Device Manager: ATM & Self-Service Monitoring and Management - Fiserv | ATM Marketplace*. Available at: https://www.atmmarketplace.com/companies/showcases/fiserv/products/fiserv-device-manager-atm-self-service-monitoring-and-management/ (Accessed: 14 May 2023).

[7] *Feel view* (no date) *GRGBanking*. Available at: https://global.grgbanking.com/en/ProductDetail_115_145.html (Accessed: 14 May 2023).

[8] *Bootstrap (front-end framework)* (2023) *Wikipedia*. Available at: https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework) (Accessed: 23 June 2023).

[9] (No date) *What is restful API? - restful API explained - AWS*. Available at: https://aws.amazon.com/what-is/restful-api/ (Accessed: 24 June 2023).

[10] *Devops services* (2023) *ClickIT*. Available at: https://www.clickittech.com/ (Accessed: 10 August 2023).

[11] GfG (2023) *REST API architectural constraints*, *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/rest-api-architectural-constraints/ (Accessed: 24 June 2023).

[12] (No date a) *Ajax introduction*. Available at: https://www.w3schools.com/xml/ajax_intro.asp (Accessed: 22 February 2024).

Appendix A – Testing Plan

Below are the additional test cases used for evaluating the ATM Monitoring Dashboard System.

- Test Cases for Manage System Users

Test ID	Test Case Summary	Pre- Conditions	Test Case Steps	Expected Results
3.1	Test Edit User Details	Test user is Admin	1. Click on Edit User icon 2. Edit fields, a. First name b. Last name c. Mobile Number d. Status 3. Click on Update button	Edit User Details screen should open Record should be updated successfully
3.2	Test Reset User Password	Test user is Admin	1. Click on Reset User Password	Confirmation pop up should appear and user should be able to confirm or cancel it. Password should reset successfully for the relevant user only.
3.3	Test search functionality	Test user is Admin	1. Search a record using a. User Name b. First Name	Record should be filtered

Table 9- Test Case for Manage System Users

- Test Cases for Manage ATM Details

Test ID	Test Case Summary	Pre- Conditions	Test Case Steps	Expected Results
4.1	Go to 'Manage ATM Details' view	The URL http://*****/ ATMDashBoard/index.html is up and running	1. Login to the system Select 'Manage ATM Details' under Services	Directs to 'Maintain ATM Details' page
4.2	Search for ATM	'Maintain ATM Details' page is opened	1. Search for value which is not in the database. 2. Search for a value in the database. 3. Compare the data with database.	Displays an error message. Displays the relevant data in the fields. Correct
4.3	Test for field validations	'Maintain ATM Details' page is opened	1. Try to update the record without entering below fields; a. ATM Name b. Branch c. Location d. OS e. Model f. Vender g. Device Type	Displaying 'fill out this field' and not able to add new ATM
4.4	Update existing ATM details	'Maintain ATM Details' page is opened	1. Update and check database.	Added fields updated correctly in the database.
4.5	Add new record	'Maintain ATM Details' page is opened	1. Enter all the mandatory fields and no mandatory fields if necessary. 2. Click on Add/Update button.	Correctly add the record to the db.

4.6	Check reset button	'Maintain ATM Details' page is opened	1. Enter some values for all the fields and click on Reset button 2. Click on 'YES'	Displays a confirmation message. Reset all the fields.

Table 10- Test Cases for Manage ATM Details

- Test Cases for ATM Downtime Report

Test ID	Test Case Summary	Pre- Conditions	Test Case Steps	Expected Results
5.1	Test Technical Faults function	ATM Down Time Report screen is opened	1. Simulate technical fault s and check report	Data displayed should be accurate
5.2	Test Technical Faults function – Download List	ATM Down Time Report screen is opened	1. Click Download	Data displayed should be accurate
5.3	Test invalid date range	ATM Down Time Report screen is opened	1. Enter invalid dates a. From date empty b. To Date Empty c. From Date is greater than To Date	Display an error message

Table 11- Test Cases for ATM Downtime Report

- Testing Status

Test ID	Description	Pass / Fail
3.1	Test Edit User Details	Pass
3.2	Test Reset User Password	Pass
3.3	Test search functionality	Pass
4.1	Go to ‘Manage ATM Details’ view	Pass
4.2	Search for ATM	Pass
4.3	Test for field validations	Pass
4.4	Update existing ATM details	Pass
4.5	Add new record	Pass
4.6	Check reset button	Pass
5.1	Test Technical Faults function	Pass
5.2	Test Technical Faults function – Download List	Pass
5.3	Test invalid date range	Pass

Table 12- Testing Status

Appendix B – User Manual

This section provides a comprehensive overview of the main functionalities offered by the developed ATM Monitoring Dashboard System. Through a combination of screenshots and relevant instructions.

Login

- Access the system using a browser using http://*****/ATMDashBoard/index.html URL. This will direct to the Login page of the system.
- In order to gain access to the system, users are required to input their valid username and password on the login page.
- If the authentication is successful, user will be directed to the system's home page or if authentication fails, an error message will be displayed.

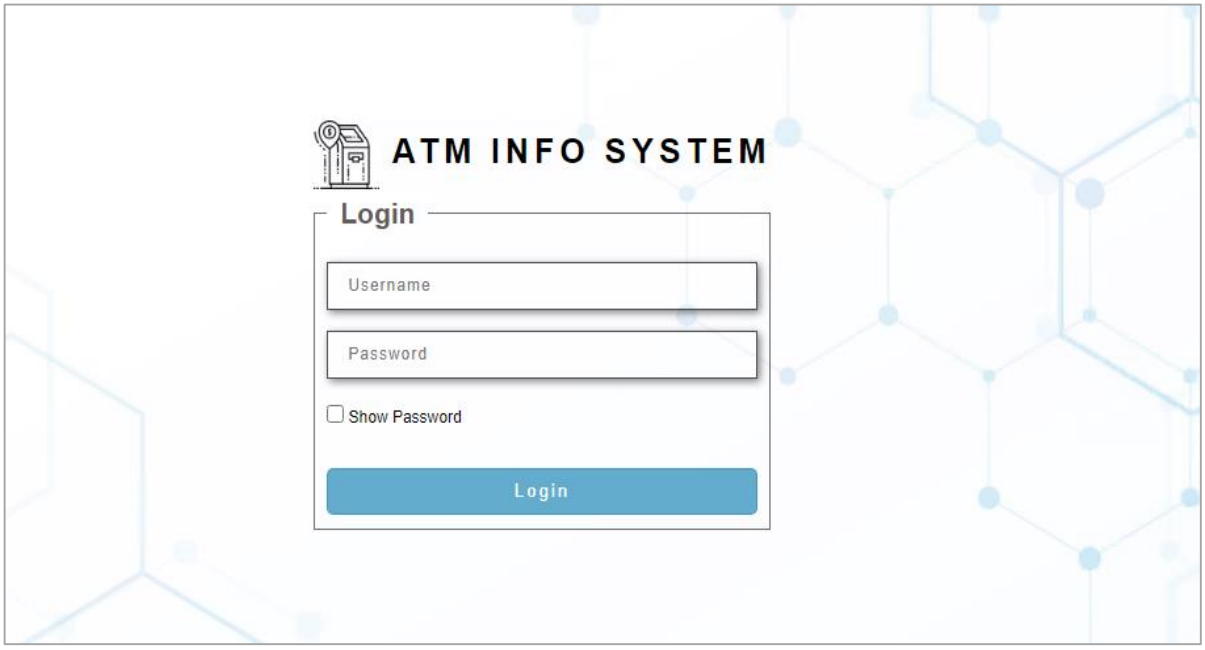


Figure 29- Screenshot of Login Page

Home Page

Once logged in, users will find the home page customized to their assigned user group, showing only the functions they are authorized to access.

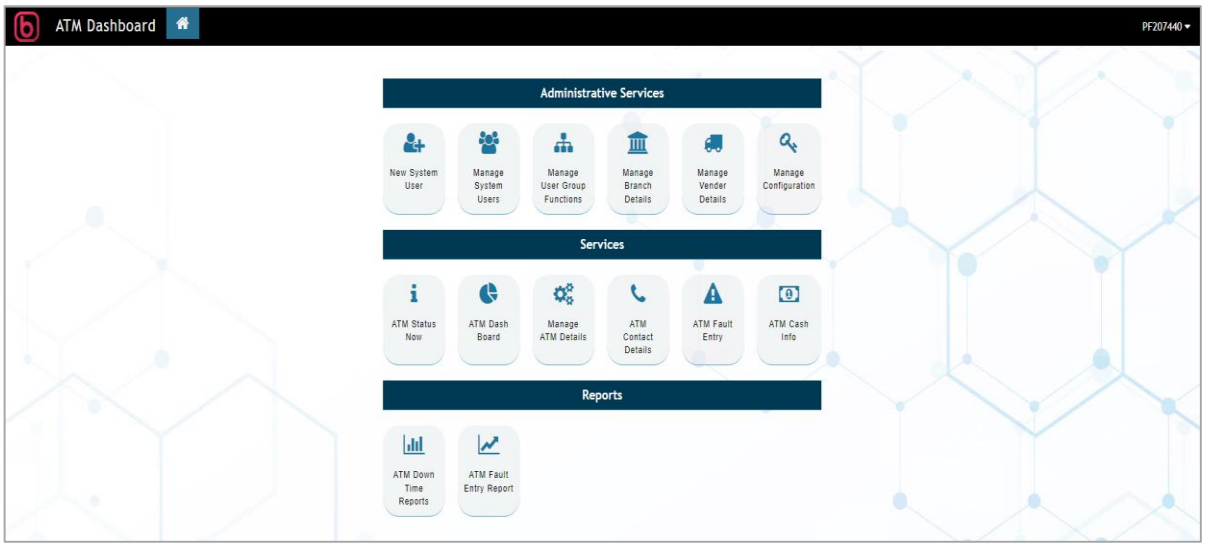


Figure 31- Screenshot of Admin Home Page

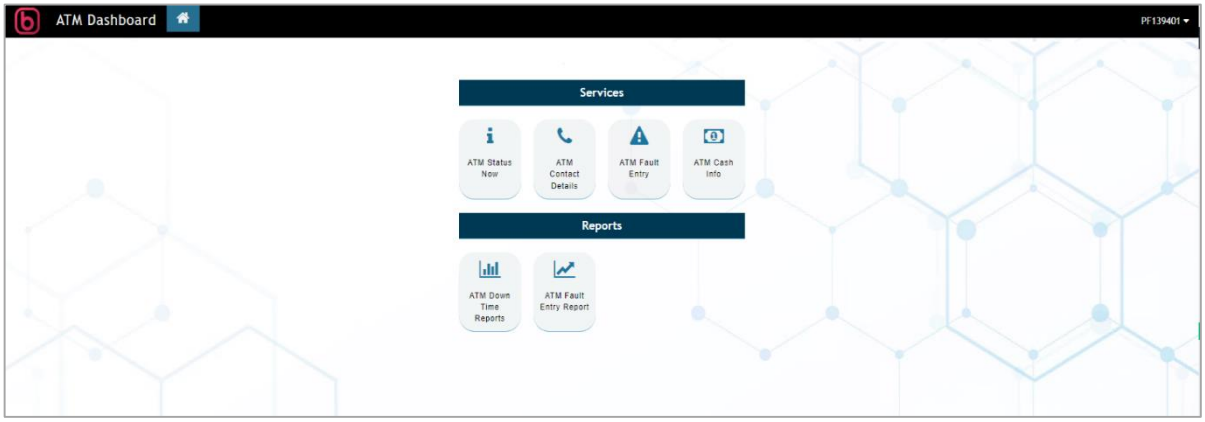


Figure 30- Screenshot of Branch User Home Page

- Figure 30 illustrates the functions assigned to a branch user, while Figure 31 displays all functions allocated to the admin. Similarly, users belonging to other groups will only see functions permitted for their user group.

- In the top right-hand corner, the username of the logged-in user is displayed. Hovering over the username triggers a dropdown menu with options to either change the password or log out.

Figure 32- Screenshot of Change Password/Logout Option

Change Password

- If users wish to change their password, they can select the option shown in the Figure 32.
- If the password policy is disabled, the new password must consist of a minimum of 5 characters.
- If the password policy is enabled, the new password should contain a number, a lowercase letter, an uppercase letter, a special character and must consists of 8 characters.
- Only users with admin privileges have the authority to modify the password policy status.
- Users can't use the old password as the new password.

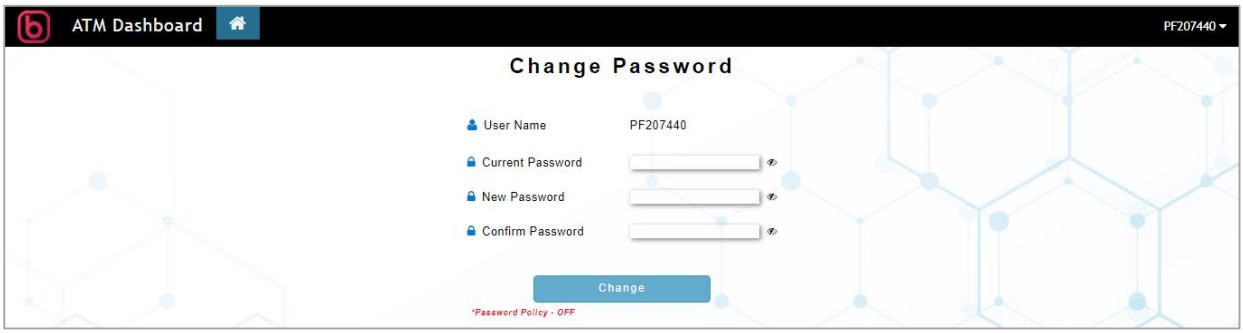


Figure 33- Screenshot of Change Password

Create New System User

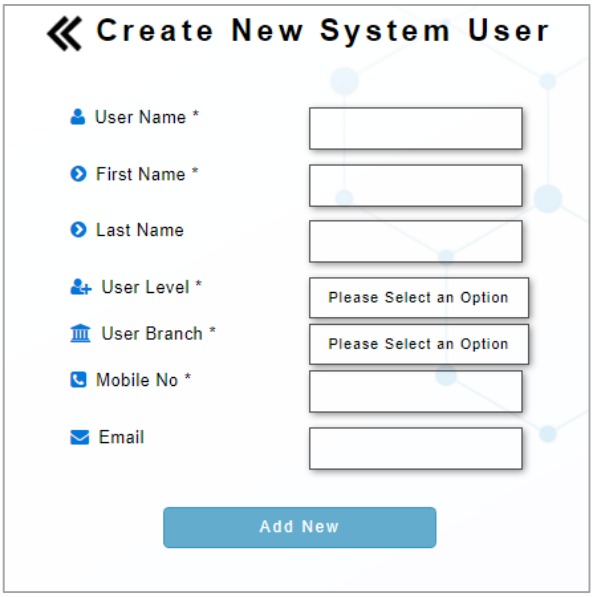


Figure 34- Screenshot of Create New User

- The user name will be the user's bank provident fund number.
- All fields marked with an asterisk (*) are mandatory.
- The mobile number and email fields are validated before submitting. The email field can be left empty.

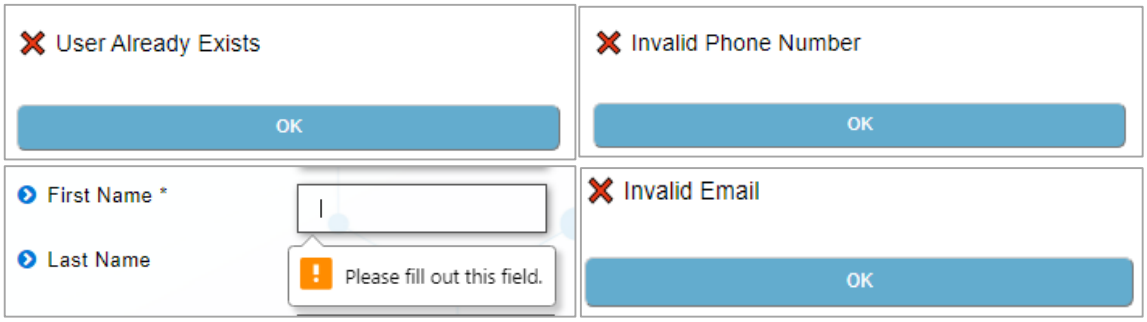


Figure 35- Screenshot of Validation in new system user

Manage System Users

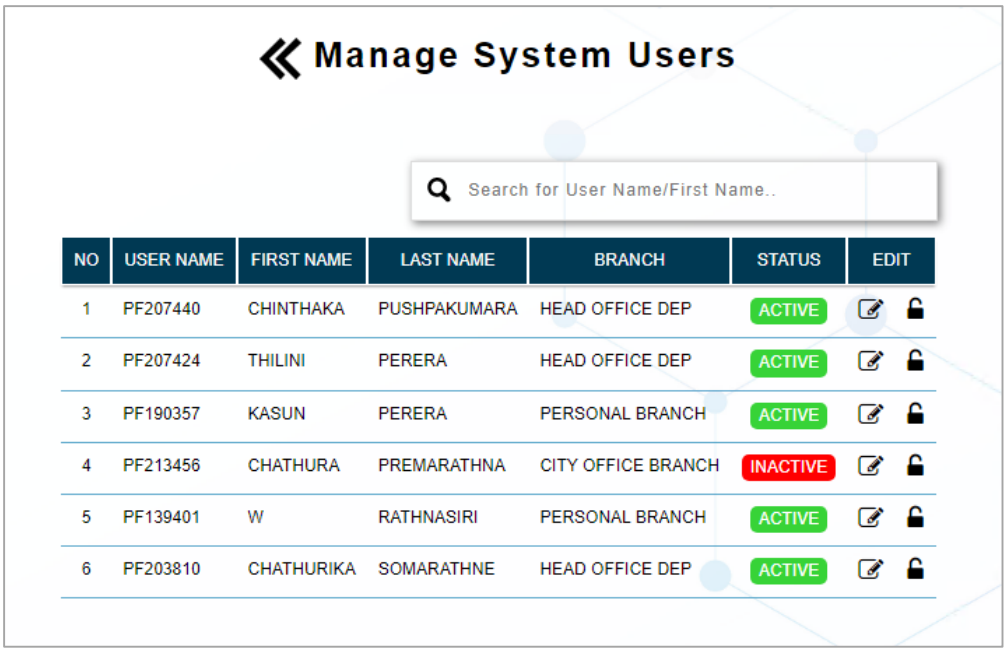


Figure 36- Screenshot of Manage System Users

- Figure 36 illustrates the complete list of registered users, including both active and inactive users.
- Every row includes options to edit the respective user data and reset the user's password.
- Only who have the administrative privileges can edit details or reset the password.
- Users are allowed search and filter by either username or first name.
- Resetting the password will restore the user's password to the default, which is the same as their username

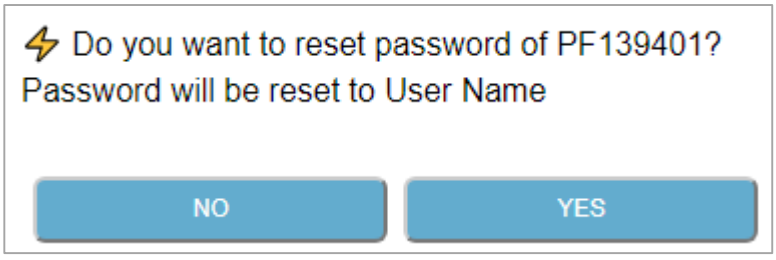


Figure 37- Screenshot of Reset Password

Edit User Details

- Clicking on the edit button image on each row in the Manage User Details form (Figure 36) will redirect users to the Edit User Detail page, displaying the relevant user details.

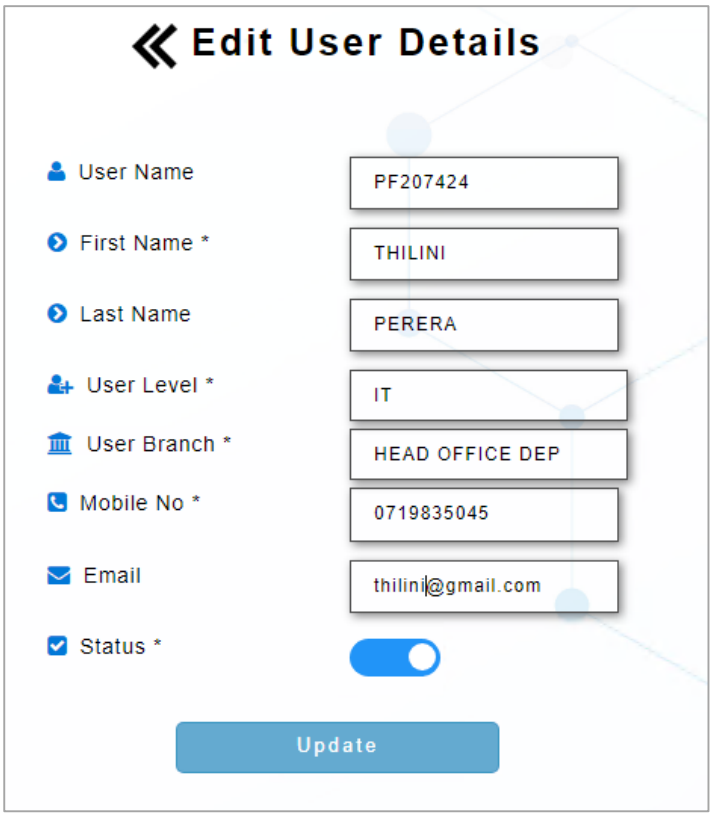


Figure 38- Screenshot of Edit User Details

- Similar to the Add New User form, validation is applied to the required fields, as well as the mobile number and email fields.

Manage User Group Functions

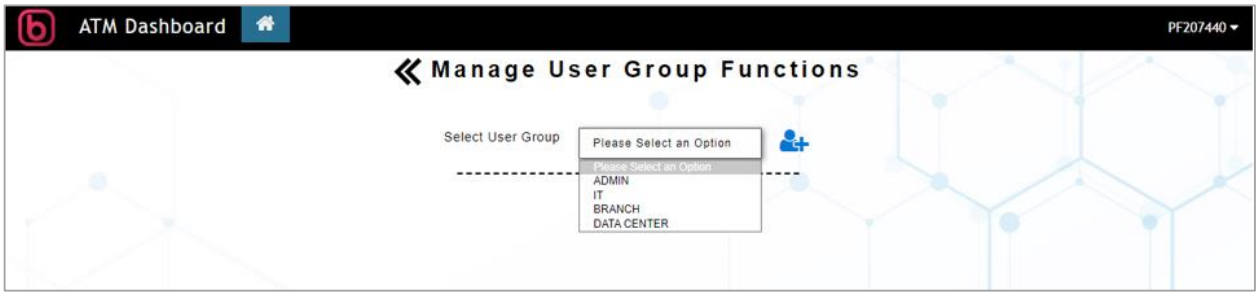


Figure 39- Screenshot of Select User Group

- As shown in the figure 39 administrators can select the user group from the dropdown menu. The dropdown menu loads all active user groups for selection.
- After selecting the user group, the system loads all available functions, with the currently allowed functions pre-selected. Administrators can then customize the allowed functions for the selected user group. Additionally, they have the flexibility to change the user group name if necessary.

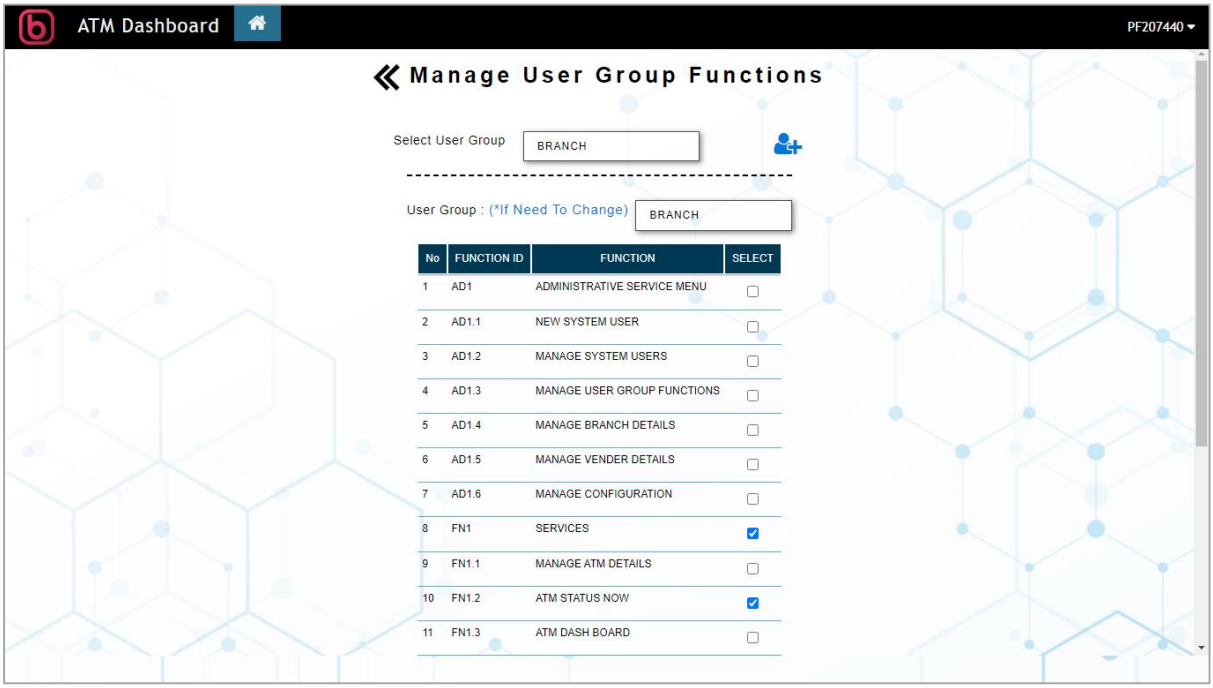


Figure 40- Screenshot of Update User Functions

- Administrators can create a new user group by clicking on the "Add New User Level" button.

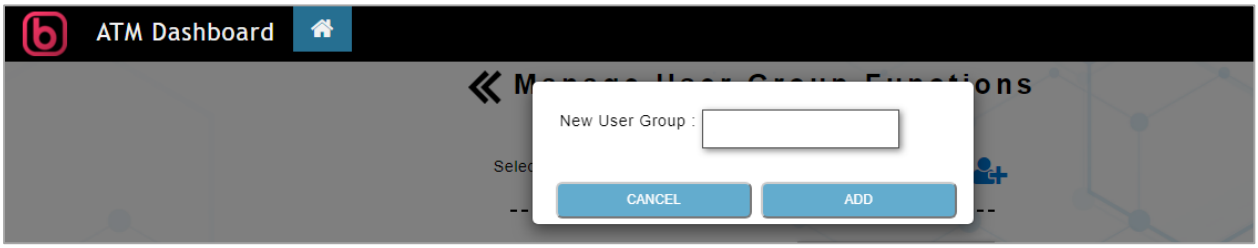


Figure 41- Screenshot of Add New User Level

Manage Branch Details

- All branch details, including branch code, branch name, and province, are loaded when the page loads.

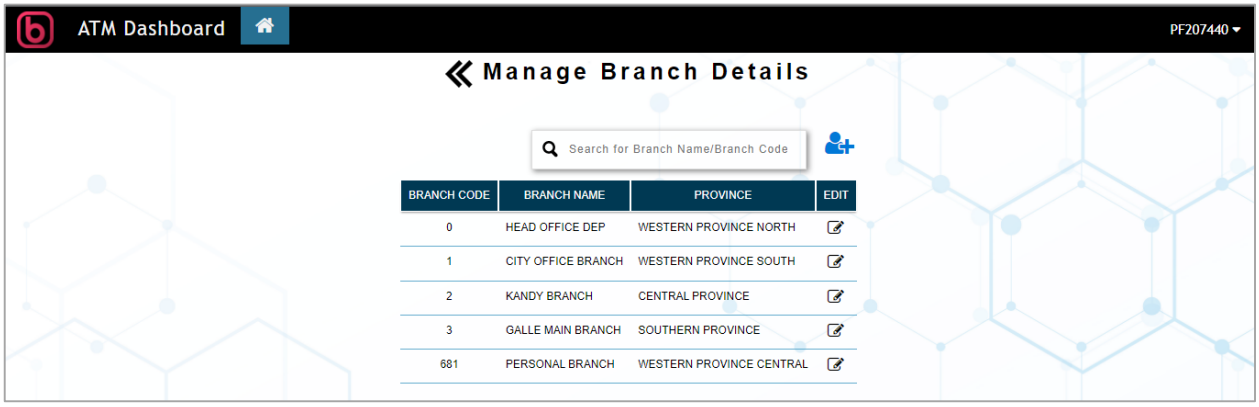


Figure 42- Screenshot of Branch Details

- Users can edit each branch's details by using the "Edit Branch Details" icon.

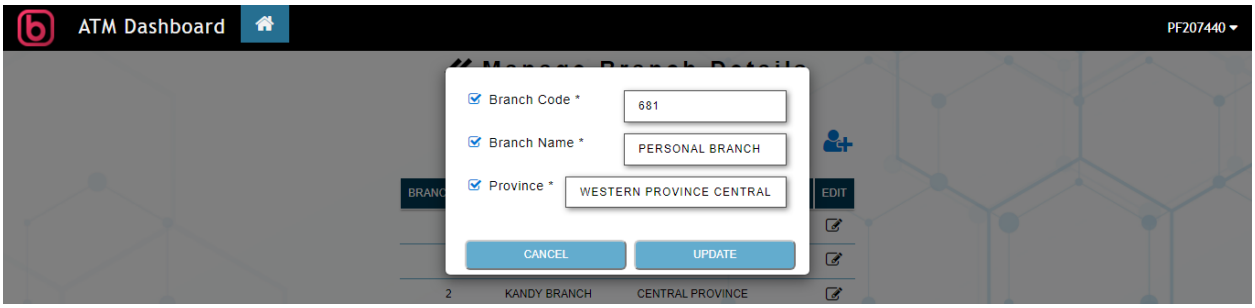


Figure 43- Screenshot of Edit Branch Details

- Users can add a new branch using the “Add New Branch” button

Manage ATM Details

- From this menu, users have the ability to search for ATM details, add new ATMs, and update existing ATM details.
- Branch details, ATM model names, and ATM vendors are pre-loaded into dropdown menus for user selection.
- All fields marked with an asterisk (*) are mandatory.

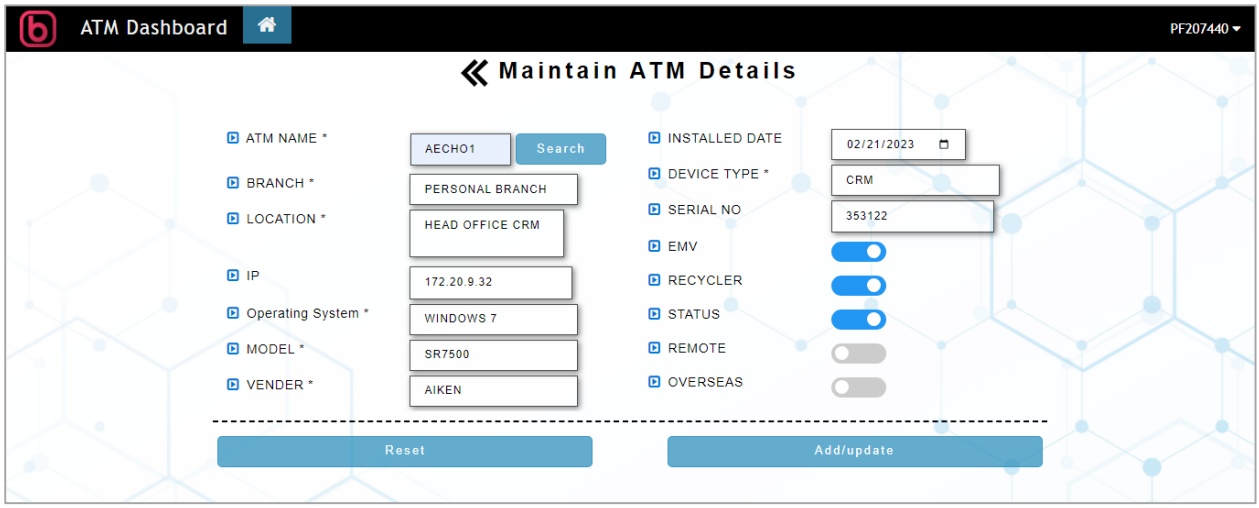


Figure 44- Screenshot of Manage ATM Details

ATM Contact Details

- From this menu, users can add or modify the contact details of the relevant ATM operators in the branch.
- These contact details are utilized by the computer room monitoring team to communicate about ATM faults and send ATM fault messages.
- Up to four contact details can be added.
- For branch users, only the ATMs related to their branch are loaded in the dropdown menu.
- After selecting the ATM from the dropdown menu, information such as ATM model, vendor, ATM location, last updated user, and the user are loaded.

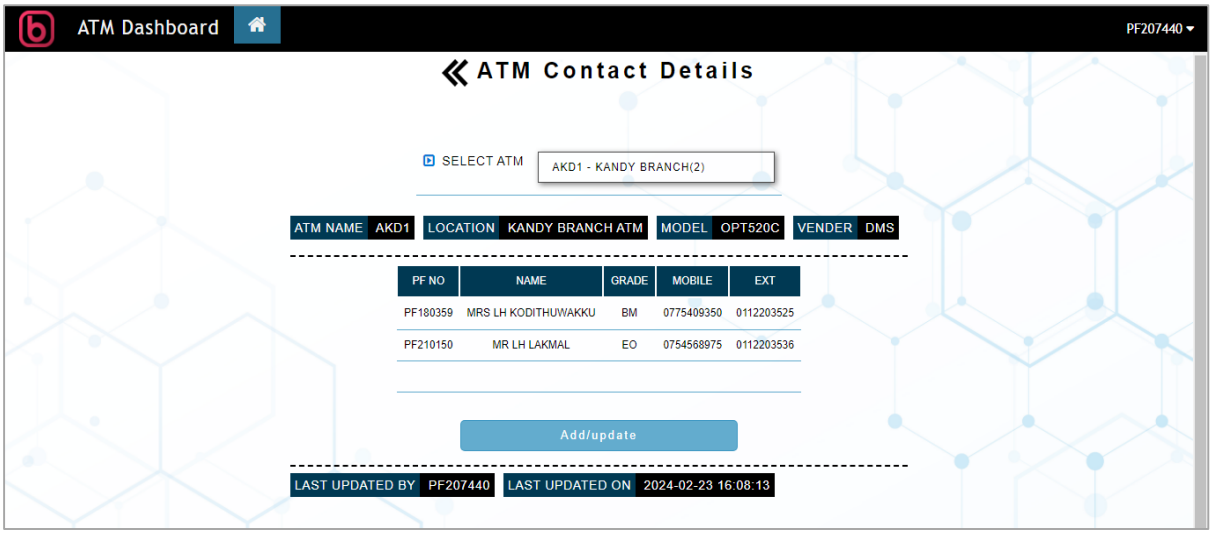


Figure 45- Screenshot of ATM Contact Details

ATM Status Now

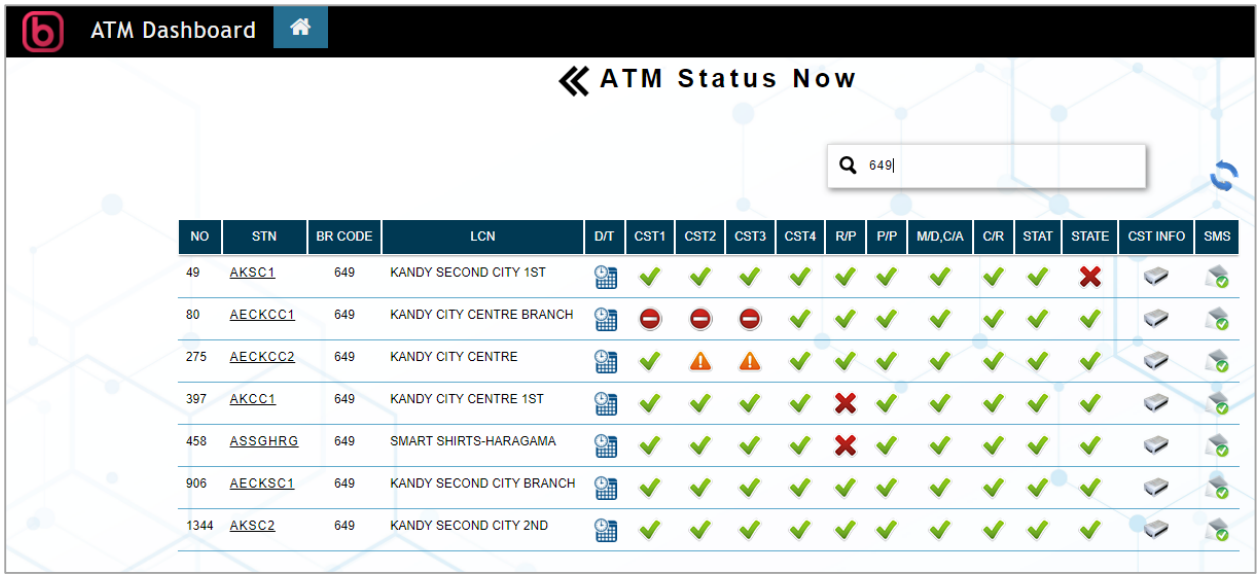


Figure 46- Screenshot of ATM Status

- As shown in the figure 46 current status of all the ATMs are shown on this page.
- Users can utilize ATM name, branch code, or ATM location as filtering options.
- Branch users are only able to view ATMs attached to their branch, while other authorized users have access to view the status of all machines.
- Icons are used instead of words to display the status of ATM devices, as shown in the above figure 46.

- The following details are displayed for each machine:
    - ATM Name
    - Branch Code
    - ATM Location
    - Last Updated Date and Time
    - All Four Cassette Status (Ok, Low, Empty, Fault)
    - Receipt Printer Status (Ok, Low, Empty, Fault)
    - Pin Pad Status (Ok, Fault)
    - Currency Acceptor/ Money Drawer Status (Ok, Fault)
    - Card Reader (Ok, Fault)
    - Communication Status (Up, Down)
    - Machine Status (Open, Close)
    - Cassette Information
    - Last Send Messages
- Clicking on the ATM name will display the ATM details and the corresponding contact detail information.

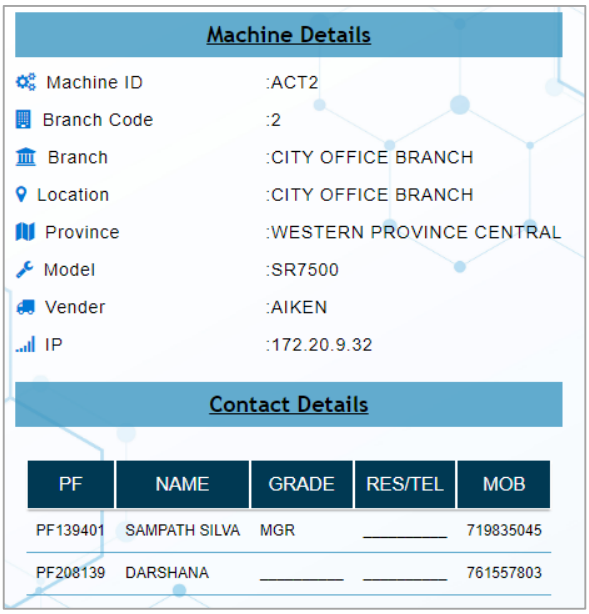
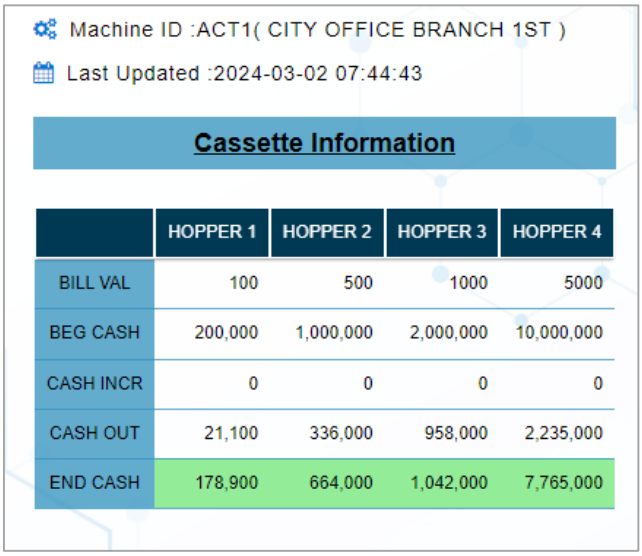


Figure 47- Screenshot of ATM Details

- Clicking on the cassette information icon will display the cassette denomination figures.



	HOPPER 1	HOPPER 2	HOPPER 3	HOPPER 4
BILL VAL	100	500	1000	5000
BEG CASH	200,000	1,000,000	2,000,000	10,000,000
CASH INCR	0	0	0	0
CASH OUT	21,100	336,000	958,000	2,235,000
END CASH	178,900	664,000	1,042,000	7,765,000

Figure 48- Screenshot of ATM Cassette Information

- Clicking on the "Last Sent SMS" icon will display the most recent SMS sent to the officers of the ATM.

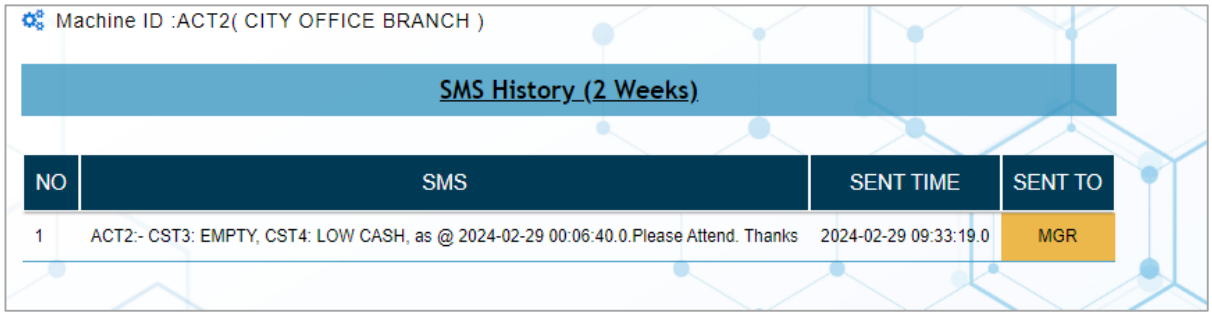


Figure 49- Screenshot of Last Sent SMS

ATM Dashboard

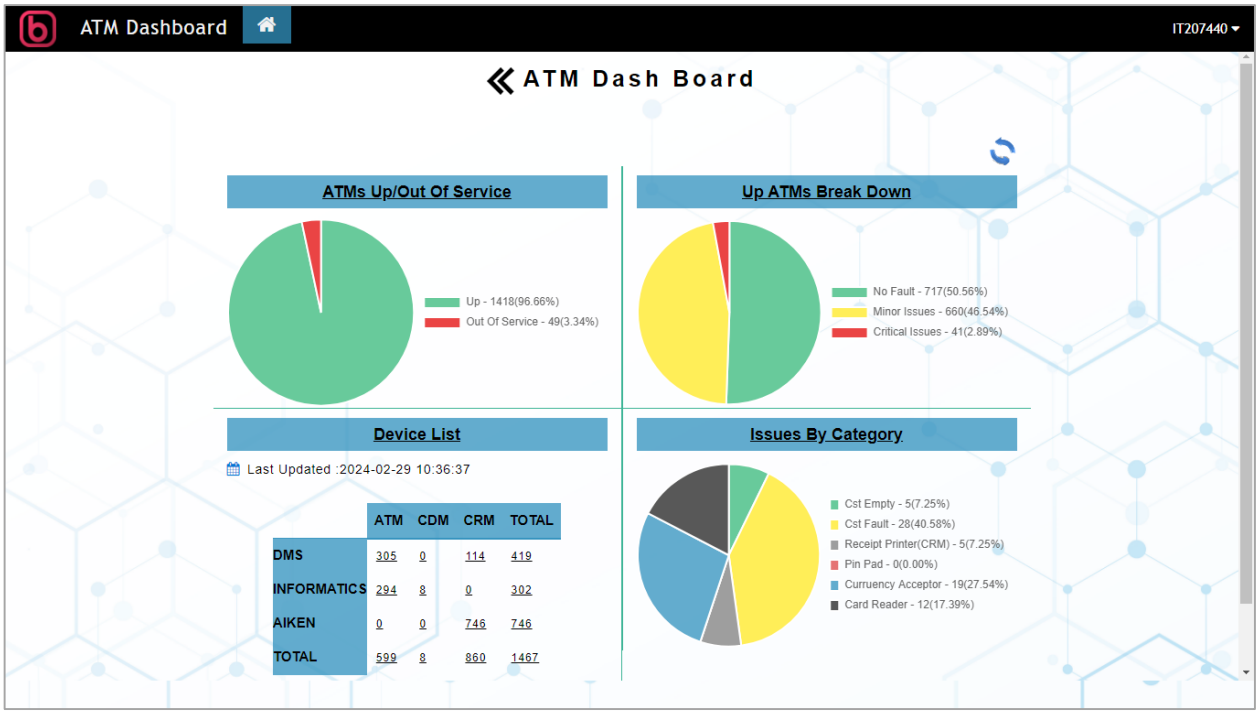


Figure 50- Screenshot of ATM Dashboard

- This interface features various types of graphs representing the information related to machines, as depicted in Figure 50.
    - Graph 1
        - Count and Percentages of in service for the customers and out of service machines due to network failure, machine fault or the branch operator log into the machine.
    - Graph 2
        - No fault machines – With no faults of the machine.
        - Minor issues machines - Customers can use the machine functions, but with the minor issues like cassette low, receipt printer low.
        - Critical issues machines – Customer can't use the cash related functions of the machine due to faults like all cassette empty, all cassette fault, currency acceptor faults.

- Graph 3
    - Total active machine count based on the machine type (ATM/CDM/CRM) and machine venders (DMS/Informatics/Hitachi)
- Graph 4
    - Breakdown of machines grouped by its faults – Cassette Empty, Cassette faults, Receipt printer faults, Card reader faults, Pin pad faults, Currency acceptor / Money drawer faults.

- Users can view detailed information for each category by clicking on the graph legend name. figure 51 shows only the Card Reader fault ATM list.

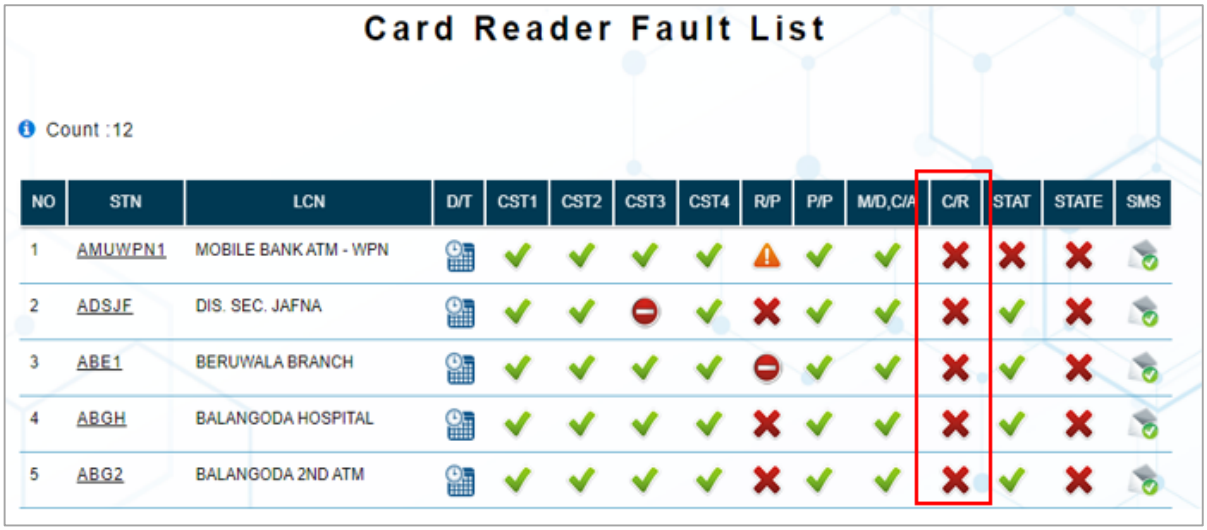
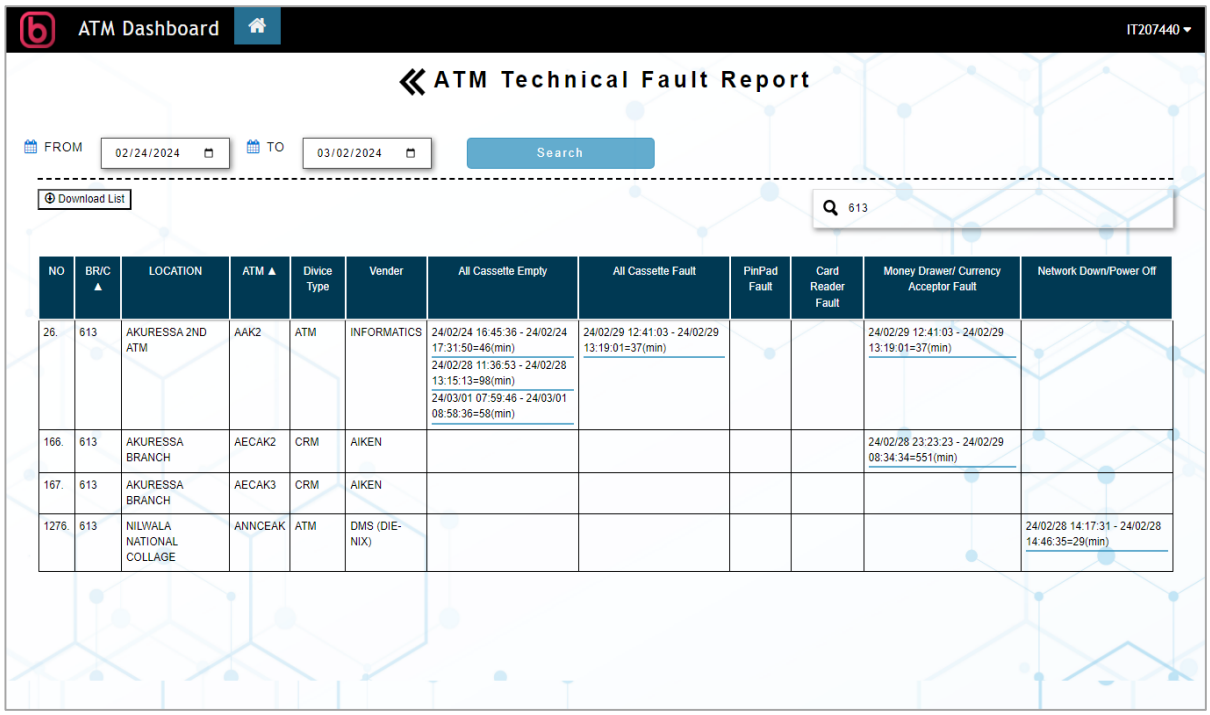


Figure 51- Screenshot of Card Reader Fault ATMs

Appendix C – Reports

This section provides a comprehensive overview of the Reports offered by the developed ATM Monitoring Dashboard System.

ATM Technical Fault Report



NO	BR/C ▲	LOCATION	ATM ▲	Divice Type	Vender	All Cassette Empty	All Cassette Fault	PinPad Fault	Card Reader Fault	Money Drawer/ Currency Acceptor Fault	Network Down/Power Off
26.	613	AKURESSA 2ND ATM	AAK2	ATM	INFORMATICS	24/02/24 16:45:36 - 24/02/24 17:31:50=46(min) 24/02/28 11:36:53 - 24/02/28 13:15:13=98(min) 24/03/01 07:59:46 - 24/03/01 08:58:36=58(min)	24/02/29 12:41:03 - 24/02/29 13:19:01=37(min)			24/02/29 12:41:03 - 24/02/29 13:19:01=37(min)	
166.	613	AKURESSA BRANCH	AECAK2	CRM	AIKEN					24/02/28 23:23:23 - 24/02/29 08:34:34=551(min)	
167.	613	AKURESSA BRANCH	AECAK3	CRM	AIKEN						
1276.	613	NILWALA NATIONAL COLLAGE	ANNCEAK	ATM	DMS (DIE-NIX)						24/02/28 14:17:31 - 24/02/28 14:46:35=29(min)

Figure 52- Screenshot of ATM Technical Fault Report

- As shown in the figure 52 users can view ATM fault report on selected time period.
- This report contained the detailed information (n (Starting time, End time, Total duration) about below listed fault groups for a selected time period.
    - All Cassette Empty
    - All Cassette Fault
    - Pin Pad Fault
    - Card Reader Fault
    - Currency Acceptor or Money Drawer Fault
    - Network Failure / Power Down
- Branch users are only able to view ATMs attached to their branch.
- Users have the option to download the list in Excel format.

- Users can filter by ATM name or branch code and sort the list by ATM name or branch code.

ATM Technical Fault Graph

Similar to the previous report, this report presents a graphical view of ATM technical faults.

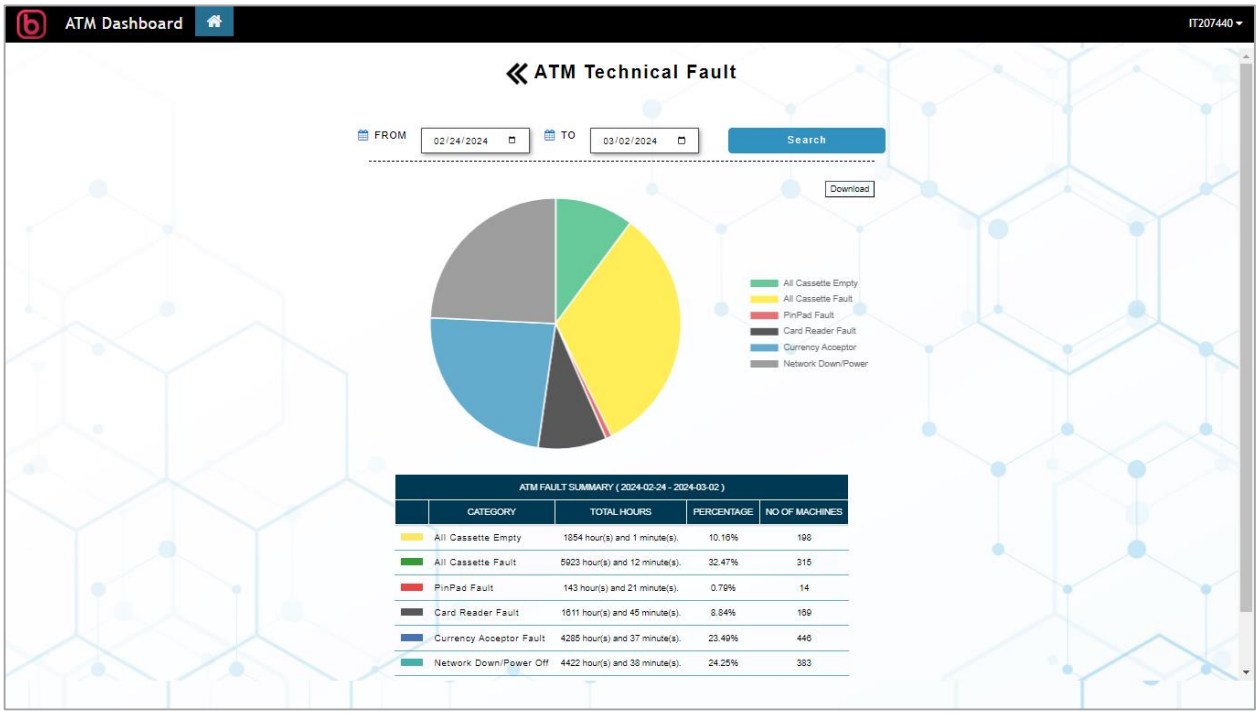
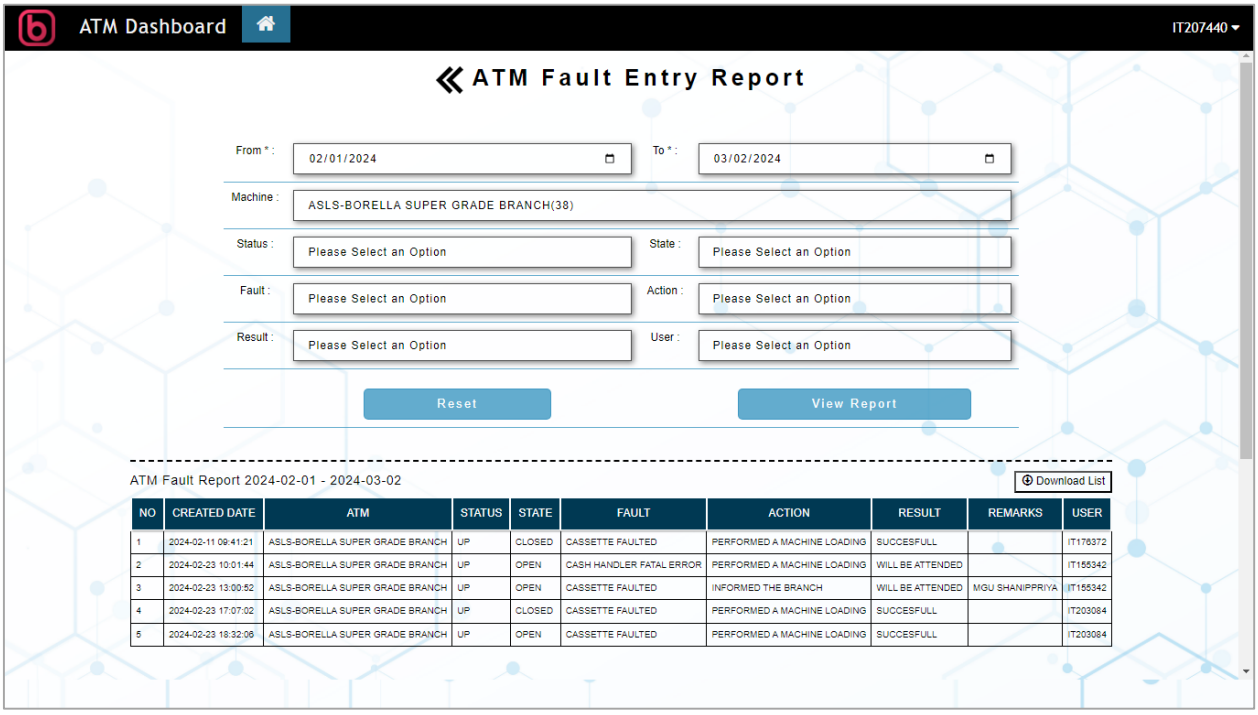


Figure 53- Screenshot of ATM Technical Fault Graph

ATM Fault Entry Report

- This report illustrates the actions taken by relevant users in response to ATM faults, as well as the results of those actions for the selected time period.
- Users have the flexibility to utilize filters to generate specific reports. For instance, they can select a particular ATM name, choose a date range, and generate a report detailing the actions taken for that ATM.
- Users have the option to download the list in Excel format.



NO	CREATED DATE	ATM	STATUS	STATE	FAULT	ACTION	RESULT	REMARKS	USER
1	2024-02-11 09:41:21	ASLS-BORELLA SUPER GRADE BRANCH	UP	CLOSED	CASSETTE FAULTED	PERFORMED A MACHINE LOADING	SUCCESFULL		IT176372
2	2024-02-23 10:01:44	ASLS-BORELLA SUPER GRADE BRANCH	UP	OPEN	CASH HANDLER FATAL ERROR	PERFORMED A MACHINE LOADING	WILL BE ATTENDED		IT155342
3	2024-02-23 13:00:52	ASLS-BORELLA SUPER GRADE BRANCH	UP	OPEN	CASSETTE FAULTED	INFORMED THE BRANCH	WILL BE ATTENDED	MGU SHANIPPRIYA	IT155342
4	2024-02-23 17:07:02	ASLS-BORELLA SUPER GRADE BRANCH	UP	CLOSED	CASSETTE FAULTED	PERFORMED A MACHINE LOADING	SUCCESFULL		IT203084
5	2024-02-23 18:32:06	ASLS-BORELLA SUPER GRADE BRANCH	UP	OPEN	CASSETTE FAULTED	PERFORMED A MACHINE LOADING	SUCCESFULL		IT203084

Figure 54- Screenshot of ATM Fault Entry Report