The application for software in an ATM(Automated Teller Mashine)

Mihail Konstantinov, Janika Ahonen , Sara Vehviläinen
TVT24KMO Group_13
Oulu University of Applied Scienses
Information Technology, Software Engineering

Introduction

This project is dedicated to the development of ATM software. The main goal was to create a reliable, secure, and functional application that supports basic banking operations.

In previous courses, we covered the fundamentals of object-oriented programming, databases, and APIs. The objective of this project was to apply those theoretical concepts in practice.

Figure 1 presents a state diagram of the system, illustrating the different states of the ATM and their transitions.

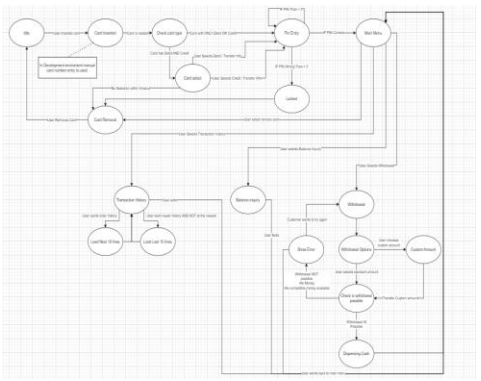

FIGURE 1. State Diagram

Objectives

Our aim was to develop a secure and efficient ATM system that includes the following functionalities: balance inquiry, transaction history, and cash withdrawal. Users log in using their card number and PIN code.(In our system, when a user logs in, their personal image is shown.) The application is secured with an encrypted PIN code and utilizes user-specific tokens to ensure that no one can access information that does not belong to them. Additionally, in our system, customers can have debit or credit cards, as well as a dual card that combines both features.

Methods

To achieve our goals, we divided the project into three main components: database, backend and frontend.

First, we designed a MySQL/MariaDB database to store user information, account balances, and transaction history. Then, we developed a REST API using Node.js and Express.js, which connected the database to the client application. Finally, we built the client-side application using Qt, allowing users to interact with the system via a desktop interface.

At the final stage, one of our team members also set up a reverse proxy, which managed all connections in the system.

As mentioned earlier, we implemented bcrypt encryption for PIN protection and used tokens for authentication. Additionally, we ensured extensive testing to verify that all system components functioned seamlessly together.

Figure 2 presents the system's ER diagram.

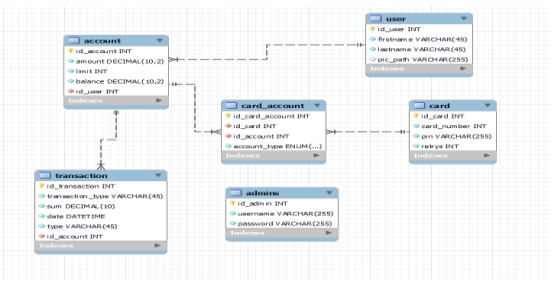

FIGURE 2. The ER Diagram

Results

We successfully developed a fully functional ATM system that supports balance inquiries, transaction history viewing, and cash withdrawals using debit, credit, and dual cards. PIN authentication ensures secure access to accounts.

For security reasons, we implemented a card lock feature that activates after three incorrect PIN attempts. We also designed an intuitive button-based user interface for ease of use. Figure 3 displays an example of the application interface, where the user inserts their card and enters their card number and PIN.

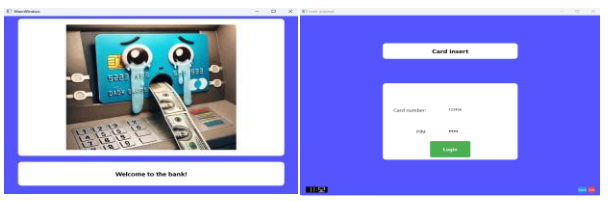

FIGURE 3. Application

Conclusions

Our ATM project met all objectives and even exceeded expectations in some aspects. We created admin users, who have the ability to perform CRUD operations via the REST API, while regular users cannot access others' data due to the individualized token system. Additionally, we ensured that most security-related functions were handled on the backend, keeping the Qt client as simple as possible. This approach contributed to a secure and efficient banking application.

Through this project, we gained valuable experience in client-server architecture, database management, and security measures.

Advanced features such as dual-function cards, image integration, setting up our own server, and a strong focus on security demonstrated our ability to develop complex software solutions, which supports us in achieving the highest grade.

References

MySQL/MariaDB Documentation
Node.js and Express.js Official Guides
Qt Framework Documentation
Course Materials on Programming and Databases

Software Development Application Project

ECTS credits: 15

Date of publication: 16.02.2025

Instructors: Pekka Alaluukas, Teemu Korpela, Johanna Talvensaari