INSTITUT D'ENSEIGNEMENT SUPÉRIEUR DE RUHENGERI

Accredited by Ministerial Order N° 005/2010/Mineduc of 16 June 2010

Scientia et Lux

FACULTY OF SCIENCES AND INFORMATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

OPTION OF SOFTWARE ENGINEERING

CARD TRACK CROSS-PLATFORM MOBILE APP

CASE STUDY: RWANDA

A dissertation submitted in partial fulfillment of requirements for an Award of a Bachelor's Degree of Science in Computer Science Option of Software Engineering

By: SHEMA Eric
Reg Nº: 22/17690

Supervisor: Aline Valerie SHIMIRWA

Co-Supervisor: Adelaide TUYIZERE

Musanze, September 2024

B.P. 155
Ruhengeri
Rwanda

T : +250 788 90 30 30
: +250 788 90 30 32
E:**info@ines.ac.rw**
W: **www.ines.ac.rw**

DECLARATIONS

I do hereby declare that the work presented in this dissertation is my contribution to the best of my knowledge. The same work has never been submitted to any other University or institution. I, therefore, declare that this work is my own for the partial fulfillment of the award of a Bachelor’s degree of Science in Computer Science, with honors in SOFTWARE ENGINEERING at INES-Ruhengeri.

Name of the candidate: **SHEMA Eric**

Signature of the candidate: ....

Date of submission: ....

APPROVAL

This is to certify that this dissertation work entitled **“**Card Track Cross-platform Mobile App” is an original study conducted by **SHEMA Eric** under supervision and guidance.

The supervisor’s name is: **Mrs. SHIMIRWA Aline Valerie**

Signature of the supervisor: ....

Date of submission: ....

DEDICATIONS

To almighty God

To my beloved parents

To my beloved brothers and sisters

To my fellow students

To my supervisor

ACKNOWLEDGMENTS

This research paper is made through the support of different people to carry out this study. With gratitude, I would like to thank my supervisor **SHIMIRWA Aline Valerie** who gave me a piece of good advice and all guidance during the process of developing the system. I am also thankful and appreciative to all my good friends for providing ideas, advice, and other contributions to accomplish my task.

It is my pleasure to thank all of my colleagues in the department of Computer Science for their useful academic interaction.

For that, all glory and praise be to the Almighty God for has provided me the strength, courage, and health to accomplish my studies at INES-Ruhengeri.

Special thanks to my parents and my family who gives me all they have to help me in my education since my childhood until now to carry out this work.

Finally, I submit my profound gratitude, respect, and love to all those who provided me with moral support and constant encouragement throughout the entire study

ABSTRACT

This project details the development of a comprehensive mobile application aimed at streamlining the recovery process for lost ID cards by leveraging the advanced capabilities of modern smartphone technology. The application, developed using the .NET MAUI framework, employs XAML for the front-end interface design, C# for the backend logic, and SQLite for efficient database management. This cross-platform solution offers a user-friendly interface where individuals can easily register, report, and track lost ID cards, facilitating rapid recovery. Key functionalities of the app include ID card registration, the ability to report lost cards, and the integration of emergency contact features that allow for real-time notifications to relevant parties, ensuring rapid communication and action. The application also supports various communication channels, such as SMS and email, to connect card finders with the rightful owners directly. To protect user data, robust security measures are embedded, including secure authentication protocols, ensuring that personal information is safeguarded throughout the recovery process. The app's architecture is designed to be both reliable and scalable, addressing the potential for future enhancements and a growing user base. The development process follows the Waterfall model, though it incorporates iterative and incremental improvements to adapt to user feedback and technical challenges. The project employed a rigorous testing phase to ensure both functionality and security before deployment. The primary programming and scripting languages utilized are XAML and C#, with the SQLite database providing a lightweight yet powerful solution for managing user data efficiently. By connecting the power of smartphones and the flexibility of the MAUI framework, this mobile application simplifies and accelerates the retrieval process for lost ID cards, ultimately reducing the associated stress, costs, and time for both users and card-recovery agencies. The app is designed to improve overall efficiency in lost card recovery efforts, providing a vital service to communities that often struggle with time-consuming and costly traditional methods.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

.NET MAUI: .NET Multi-platform App User Interface.

.Net: Network Enabled Technologies.

API: Application Programming Interface.

C#: C-Sharp.

DBMS: Database Management System.

DFD: Data Flow Diagram.

ER: Entity-Relationship.

FTC: Federal Trade Commission.

GPS: Global Positioning System.

ID: Identification.

IDE: Integrated Development Environment.

INES: Institut d'Enseignement Supérieur.

MAUI: Multi-platform App User Interface.

NIDA: National Identification Agency.

RNP: Rwanda National Police.

SQL: Structured Query Language.

TV: Television.

UI: User Interface.

UX: User Experience.

XAML: eXtensible Application Markup Language.

LIST OF APPENDICES

CHAPTER 1: GENERAL INTRODUCTION

1.1 Background of the project

With the increasing prevalence of lost personal belongings, particularly ID cards which includes; National ID, Student ID, Employee ID, Health insurance ID, Driver's License, and Membership cards, there is a growing need for efficient and effective methods of retrieval. In Rwanda, like also in other different countries, traditional ID tracking processes often involve physical locations searching, giving radio/TV announcements ((FTC), 2024). Which can be complicated, expensive and time-consuming. A mobile app that utilizes ID card registration offers a promising solution by leveraging smartphone technology to modernize the process of lost item recovery.

In Rwanda, similar to many other countries, the traditional methods for tracking and retrieving lost IDs are outdated and inefficient (Police, 2021). Common practices involve: searching physical locations and announcements via radio and TV. These methods are often irritating, costly, and time-consuming.

The advent of smartphone technology presents an opportunity to revolutionize the process of lost item recovery. A mobile application that registers ID cards can offer a modernized, efficient, and effective solution for this problem. By utilizing the capabilities of smartphones, such as: Easy access and use, Digital storage and retrieval and Real-time notifications. This mobile app aims to streamline and enhance the process of recovering lost IDs, providing a more practical and user-friendly alternative to traditional methods (Moe, 2020).

1.2 Problem statement

Despite significant advancements in technology, the process of retrieving lost items, particularly ID cards, remains inefficient and fragmented. Traditional methods of recovery are largely outdated and rely on several cumbersome and unreliable approaches:

Chance encounters: Many lost items are found and returned purely by chance. For example, a person may find a lost ID card and happen to know the owner or know how to contact them. This method is highly unreliable and depends on sheer luck, leading to a low success rate in item recovery.

Delays: the process of physically returning and retrieving lost items is slow. Owners may experience long waits before their items are found and returned, if they are returned at all. This can be particularly problematic for important IDs that are needed for daily activities, such as driver's licenses or health insurance cards.

Uncertainty: Both the owner and the finder face uncertainty in this process. Owners are unsure if their lost items will ever be returned, and finders are uncertain about the correct procedure for returning items. This uncertainty reduces the likelihood of successful item recovery.

1.3 Research objectives

To develop and evaluate a mobile application designed to improve the efficiency and effectiveness of the retrieval process for lost ID cards.

1.3.1 General objective

The aim of this research is to create a mobile application that revolutionizes the process of recovering lost ID cards by leveraging modern technology.

1.3.2 Specific Objectives

To design and implement a mobile application that allows users to register and manage their ID cards efficiently. This application aims to:

i. To enable users to easily register different types of ID cards (e.g., National ID, Student ID, Employee ID, Health Insurance ID, Driver’s License, and Membership Cards) within the app.

ii. To facilitate real-time notifications to users and relevant authorities when an ID card is reported lost or found.

iii. To provide a user-friendly interface for both reporting lost IDs and retrieving them, ensuring easy access and effective digital storage of ID information.

1.4 Research questions

The research questions are designed to address the specific objectives and guide the exploration of how Card Track mobile app is facilitating the retrieval of lost ID cards. Each question corresponds to one of the specific research objectives:

i. What are the main challenges faced in the implementation and adoption of the app?
ii. How effective is the Card Track mobile app in facilitating the retrieval of lost ID cards?
iii. What are users' thoughts on the app's user-friendliness and performance?

1.5 Research hypotheses

This hypothesis posits that the implementation of the Card Track mobile application result in a higher success rate for retrieving lost ID cards. Traditional methods, such as physically returning IDs to designated locations or relying on chance encounters, are often inefficient and fragmented. The Card Track app aims to centralize and streamline the process by providing a digital platform for reporting and tracking lost IDs. This centralized system is expected to reduce the time taken to find and return lost cards, increase the number of cards successfully returned to their owners, and improve overall efficiency in the lost-and-found process. This project explore the following hypothesis regarding card tracking system;

i. **Challenges in Implementation and Adoption**: Main challenges include technological barriers, public awareness and trust, user compliance, and the need for integration with existing systems like national databases. Addressing these challenges is crucial for successful implementation and widespread adoption of the app.
ii. **Effectiveness of Card Track Mobile App:** The app effectively streamlines the retrieval of lost ID cards by leveraging smartphone technology. It modernizes the recovery process, reducing reliance on outdated methods like physical searches and radio announcements, providing faster and more reliable results.
iii. **User Feedback on App User-Friendliness and Performance**: Users appreciate the app's simplicity, ease of navigation, and real-time notifications, which enhance the recovery process. While overall feedback is positive, there

may be suggestions for additional features or improved performance under heavy usage.

1.6 Choice of Study

This study was chosen due to the increasing need for efficient and modern solutions to the problem of lost ID card retrieval in Rwanda (Police, 2021). Traditional methods, such as searching physical locations or broadcasting announcements through radio and television, have proven to be cumbersome, time-consuming, and often ineffective. The rapid rise of smartphone use offers a promising avenue for solving this issue through digital means. The selection of this study is further motivated by the opportunity to explore how a mobile app like Card Track can significantly improve the efficiency of lost item recovery. By leveraging digital storage, real-time notifications, and ease of use, the study aims to demonstrate the app's potential to replace outdated methods and simplify the process for both ID owners and finders. The focus on Rwanda's context also makes the study relevant, as it explores an innovative solution to a prevalent problem in a local setting, with broader applications for other regions facing similar challenges.

1.7 Delimitation of the Study

The delimitation of the study focuses on the design, development, and implementation of a mobile application for retrieving lost ID cards specifically within Rwanda. It is limited to smartphone-based solutions, excluding other digital or non-digital methods, and involves users who own smartphones and are open to using a mobile app for lost ID recovery. The study concentrates on personal identification cards, such as National IDs, Student IDs, Employee IDs, Health Insurance IDs, Driver's Licenses, and Membership cards, and does not address other lost belongings like credit cards or passports. This delimitation allows for a focused exploration of mobile technology's impact on improving the lost ID retrieval process in Rwanda.

1.8 Significance of the study

The findings of this study have practical implications for individuals who frequently lose their ID cards and for organizations responsible for managing retrieval. By providing the understandings into the effectiveness and usability of the app, this research can inform the design and implementation of similar solutions in other contexts. People are interested in the Card Track mobile app for several reasons including; Convenience, Efficiency, Real-Time Notifications, User-Friendly Interface, Community Engagement, Security, Accessibility, and Cost-Effectiveness. And the institution's interest in the Card Track mobile app are Reputation Building, Community Engagement, and Generating revenues.

1.9 Research methodology

The development of Card Track mobile app project follows a mixed-methods approach in four phases. Phase 1 involves requirement analysis through literature review and stakeholder consultations, followed by detailed planning. In Phase 2, the app is designed, developed using agile methodology, and tested for functionality and usability. Phase 3 covers pilot testing, refinement based on feedback, and final deployment. Phase 4 evaluates the app's effectiveness and user satisfaction through data collection and analysis, with findings compiled into a comprehensive report. Development tools include Microsoft Visual Studio with .Net Maui framework.

1.10. Organization of the study

This chapter outlines the structural framework for the project. The project is divided into five distinct chapters, which includes; the introduction, literature review, methodology, Design and implementation, and conclusion. Each contributing to a comprehensive exploration of the research problem, methodology, findings, and implications. The project details organization is as follows: The first chapter is the Introduction, this chapter provides an overview of the research problem, objectives, and significance of the study. It introduces the context within which the research is conducted and outlines the structure of the hypothesis. The second chapter is the Literature Review, this chapter critically reviews existing literature relevant to the research topic. It synthesizes key findings, identifies gaps in current knowledge, and lays the theoretical foundation for the research. The third chapter is the project methodology, this chapter details the research methods and procedures used to address the research questions or hypotheses. It discusses data collection techniques, data analysis methods, and ethical considerations. The fourth chapter is the design and Implementation this chapter describes the development or implementation process of the research intervention, system, or model. It discusses technical details, challenges encountered, and solutions implemented during the design and implementation phases. The fifth chapter includes Conclusion and Recommendations**, t**he final chapter summarizes the findings of the study, discusses implications for theory and practice, and offers recommendations for future research or application. It concludes with reflections on the overall research process and its contribution to the field. Each chapter contributes to a comprehensive understanding of the research objectives, methodologies employed, and the outcomes derived from the study.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

In the development of a card tracking mobile app, this chapter provides a comprehensive overview of existing research and practices related to the retrieval of lost ID cards and the use of digital solutions to address this issue. This chapter aims to establish the context for the development of the Card Track mobile app by examining the limitations of traditional methods, exploring the potential of mobile technology, and identifying best practices from similar digital initiatives.

2.2 Related Works

In this section, the previous studies and existing solutions related to the retrieval of lost ID cards and the development of mobile applications are examined for this purpose. Understanding related works helps to identify successful strategies, common challenges, and areas where improvements can be made (Kesh, 2017).

2.2.1 Tile App

Tile is a popular item-tracking platform that uses Bluetooth technology to help users locate lost items. Users attach Tile devices to their belongings, and the Tile app helps track these items if they go missing. The app provides features like community find, where other Tile users can help locate lost items, and smart alerts for when an item goes out of range (Tiles, 2022).This app functions effectively but lacks the capability to track additional personal belongings, such as ID cards.

2.2.2 Apple’s “find my” App

Apple's Find My app integrates seamlessly with the Apple ecosystem, allowing users to locate their Apple devices and other compatible items. It uses a combination of Bluetooth, Wi-Fi, and GPS to provide accurate location tracking (Apple, 2023). As Tile App, this app functions effectively but lacks the capability to track additional personal belongings, such as ID cards.

2.3 Definition of key terms

In defining the concepts, we have to talk about each key concept that has contribution in the development of a functional card tracking mobile app.

2.3.1 Mobile app development

Mobile app development refers to the process of creating software applications that run on mobile devices such as smartphones and tablets. This process encompasses several stages, each involving specific activities and considerations to ensure the successful creation and deployment of a mobile application. (Tao, 1991)

2.3.2 Cross-platform

Cross-platform refers to the ability of software or applications to function across multiple operating systems or environments using a unified codebase. This means that a cross-platform application can run on different operating systems like Windows, mac OS, Linux, and mobile platforms such as iOS and Android without requiring significant modifications. Achieving cross-platform compatibility typically involves using frameworks or tools that abstract platform-specific details, allowing developers to write code once and deploy it on various systems. This approach enhances development efficiency, reduces costs, and ensures a consistent user experience across different devices. (SQLite, 2023)

2.3.3 Front-end Development

Front-end development refers to the practice of creating the visual and interactive aspects of a software application that users interact with directly. For mobile apps, this involves designing and implementing the user interface (UI) and ensuring a smooth user experience (UX). Front-end development is crucial for the Card Track mobile app, as it directly impacts user engagement, satisfaction, and overall effectiveness. (Haverbeke, 2018)

2.3.4 Back-end Development

Back-end development refers to the server-side logic and infrastructure that power the functionalities of a software application. For the Card Track mobile app, back-end development is essential for handling data processing, storage, and security, enabling the app to perform tasks such as user authentication, ID card registration, and real-time notifications efficiently and securely (Kleppmann, 2023).

2.3.5 Database

Database refers to a structured collection of data that is stored and managed in a way that allows for efficient retrieval, insertion, updating, and deletion of data. Databases are a fundamental component of modern software applications, providing a backbone for storing and accessing persistent data. These systems use structured query language (SQL) for defining and manipulating data. They organize data into tables with predefined schemas. Examples include; MySQL, PostgreSQL, and Microsoft SQL Server (Apple, 2023).

2.3.6 SQLite

SQLite is a lightweight, self-contained, and serverless relational database management system that is embedded into applications. In C#, SQLite is often used to provide a simple yet powerful database solution for local storage needs, especially in desktop and mobile applications. It is favored for its simplicity and ease of integration, requiring minimal setup and maintenance compared to traditional database systems. C# applications interact with SQLite using libraries such as System.Data.SQLite or Microsoft.Data.Sqlite, which offer APIs for executing SQL commands, managing connections, and handling data operations directly within the application's code. SQLite's compact footprint and full-featured SQL support make it ideal for scenarios where a small, efficient, and transactional database is needed without the overhead of a full-fledged database server (Bond, 2023).

2.3.7 XAML

Extensible Application Markup Language (XAML) is responsible for designing the user interface (UI) of the Card Track app. It enabled the creation of a responsive, user-friendly, and visually appealing interface that made it easy for users to navigate and interact with the app. XAML's declarative nature allowed for a clean and structured approach to UI design, ensuring that the app’s interface was both functional and attractive. (Edmondson, 2018)

2.3.8 C#

C# (pronounced "C-sharp") is a modern, object-oriented programming language developed by Microsoft as part of its .NET initiative. It is designed to be a simple, powerful, and versatile language that can be used to build a wide range of applications, from desktop and web applications to mobile and cloud-based services (Microsoft, 2023).

2.3.9 Visual Studio Community (Development Environment)

Visual Studio Community served as the primary integrated development environment (IDE) for the project. It facilitated the coding, debugging, and testing processes, providing a comprehensive platform where all aspects of the app could be developed efficiently. The IDE's robust features ensured seamless integration of various technologies, making the development process more streamlined and productive (Jansen, 1998).

2.3.10 .NET MAUI Framework (Cross-Platform Development Framework)

Network Enabled Technology Multi-platform App User Interface (.NET MAUI) played a crucial role by enabling the development of a single codebase that could run on multiple platforms, specifically Android and iOS. This framework ensured that the Card Track app could reach a broader audience without the need for separate development efforts for different operating systems, thus saving time and resources while ensuring consistency across platforms (Xamarin, 2023).

CHAPTER 3: PROJECT METHODOLOGY

3.1 Introduction

Project Methodology is the systematic and scientific approach used to conduct research, gather data, and interpret findings to address general objective and specific objectives. This chapter outlines the systematic approach that was employed to achieve the objectives of the study. It provides a detailed plan for data collection, analysis, and interpretation, ensuring that the research is conducted carefully and transparently. This chapter is crucial as it establishes the credibility and reliability of the study by explaining the methods used to gather and analyze data, thereby allowing others to replicate the study or build upon its findings.

3.2 Case study

In the context of Rwanda, the cross-platform mobile application project is highly relevant as it addresses specific challenges related to lost ID card recovery in the country. Rwanda, like many other regions, faces inefficiencies with traditional methods such as physical searches and media announcements, which can be cumbersome and time-consuming. By implementing a cross-platform mobile app tailored to the Rwandan context, the project aims to modernize and streamline the process of recovering lost IDs such as national IDs, student IDs, and driver’s licenses by leveraging smartphone technology. This approach not only aligns with Rwanda’s increasing digital transformation but also provides a scalable and user-friendly solution that can be easily adopted across diverse communities and regions within the country (Police, 2021).

3.3 Data Collection Techniques

To understand the needs of our target users, we employ various data collection techniques personalized to the card tracking mobile app. The effective data collection is essential for understanding the current state of ID card loss and recovery methods, as well as for designing and implementing a mobile app that addresses these issues. In this project, three primary data collection techniques were employed: observations, media analysis, and documentation review. Each of these techniques provided valuable insights and helped shape the development of the mobile app.

3.3.1 Observations

Observations involved directly monitoring the current methods used for tracking and retrieving lost ID cards. This included visits to physical locations such as police stations, lost and found offices, and other relevant institutions (Aleryani, 2016). Observations helped to identify the frequency and types of ID cards reported lost, the procedures and challenges involved in the traditional retrieval process, the interactions between individuals and authorities during the retrieval process. Through these observations, we gained firsthand knowledge of the inefficiencies and frustrations faced by individuals trying to recover their lost ID cards. This information was critical in highlighting the need for a more streamlined and user-friendly solution.

3.3.2 Media

Media analysis focused on understanding how lost ID cards are reported and recovered through various media channels. This included; reviewing radio and TV announcements about lost ID cards, Analyzing social media posts and community bulletin boards, Examining online forums and websites dedicated to lost and found items. By analyzing media content, we were able to assess the reach and effectiveness of these traditional methods. This analysis revealed the limitations of relying on media for lost ID card recovery, such as limited coverage, delayed information dissemination, and the lack of interactivity and immediate feedback (NIDA, 2019).

3.3.3 Documentation

Documentation review involved examining existing records and reports related to lost ID card incidents. This included: reviewing police reports and logs of lost and found ID cards, analyzing institutional records from places like universities, hospitals, and workplaces, examining governmental and non-governmental reports on ID card management and recovery. The documentation review provided a comprehensive understanding of the scale and nature of the problem. It helped to quantify the frequency of lost ID card incidents, identify common patterns and trends, and understand the existing protocols for handling such cases (Police, 2021).

3.4 Data Analysis

This section involves systematically examining and interpreting the data collected from various sources to derive meaningful insights and conclusions relevant to the project's objectives. This process includes organizing the data to identify patterns, trends, and correlations, and evaluating the results to address research questions or test hypotheses. For the project at hand, data analysis would focus on assessing the effectiveness of the mobile application in improving lost ID recovery processes, including metrics such as user engagement, recovery rates, and user satisfaction. The goal is to ensure that the application meets its intended objectives and to identify any areas for further improvement or optimization (Shores, 2023).

3.5 System Requirements

System requirements are critical to defining what the new system needs to accomplish and the constraints under which it must operate. These requirements provide a clear understanding of the system's functionality, performance, and operational environment. They are divided into two main categories: functional requirements and non-functional requirements. Functional requirements specify the core functionalities the system must provide, while non-functional requirements define the quality attributes and constraints of the system.

3.5.1 Functional Requirements

These specify what a system should do. They describe the specific behaviors, functions, and interactions that the system must perform. They are concerned with the "what" of the system – what actions the system must perform and what services it should provide (Edmondson, 2018). They are as follow:

i. **User Registration:** The app allows users to register their profiles, including personal details and ID card information.

ii. **ID card Registration:** Users is able to register their various ID cards (e.g., National ID, Student ID, Employee ID, Health Insurance ID, Driver’s License, Membership cards) with the app.

iii. **Lost ID Reporting:** Users is able to report lost ID cards through the app, triggering the tracking process.

iv. **ID card Search:** The app provides a search functionality to help users locate their lost ID cards.

v. **Notification System:** Users receives notifications when their lost ID cards are found or when there is a status update.

3.5.2 Non-Functional Requirements

These specify how a system should perform its functions. They describe the quality attributes, performance criteria, and constraints of the system. They are concerned with the "how" of the system – how well the system performs its functions and under what conditions (Edmondson, 2018). They are as follow:

i. **Usability:** The app have an intuitive and user-friendly interface to ensure ease of use.

ii. **Performance:** The app perform efficiently, with quick response times for searches and notifications.

iii. **Scalability:** The app is scalable to handle a growing number of users and ID card registrations.

iv. **Reliability:** The app is reliable, with minimal downtime and robust error handling.

v. **Compatibility:** The app is compatible with various mobile operating systems (iOS, Android, Windows).

vi. **Cost-Effectiveness:** The development and maintenance of the app is cost-effective to ensure affordability for users.

1.6 Technical Requirements

These involve the technologies and platforms that the software must support, such as;

i. **Mobile Operating Systems (Windows OS Support)**: The application is designed to run on Windows OS, making use of .NET MAUI's capabilities to build cross-platform applications that can be deployed on Windows, as well as other platforms such as Android and iOS if expanded.

ii. **Development Frameworks (.NET MAUI Framework):** The .NET Multi-platform App UI (MAUI) framework is utilized to create a cross-platform mobile application with a single codebase. It allows for the development of applications that can run on various platforms, including iOS, Android, and Windows, using C#.

iii. **Programming Languages (C# and XAML):** The primary programming language used for coding the application. C# is integral to .NET MAUI and is employed to write the logic and functionality of the app. XAML (Extensible Application Markup Language) is a declarative markup language used primarily for designing user interfaces in applications. It's commonly associated with Microsoft's technologies, especially in the context of .NET. XAML is used in various frameworks and platforms.

iv. **Database Management (SQLite)**: Chosen for local database management, SQLite is used to store and manage data directly on the user's device, such as user profiles and ID card information. Its lightweight and efficient nature makes it suitable for local storage needs.

3.7 System Development Methodology

The software development methodology chosen for this project is crucial in ensuring that the mobile app meets user requirements, is delivered on time, and functions efficiently. There exist plenty methodology, example; Waterfall, V-Model, Incremental, Iterative, Agile, Scrum, Kanban, Extreme Programming, Rapid Application Development. For the development of the ID card registration and retrieval mobile app, the agile methodology was adopted. Agile is known for its flexibility, iterative approach, and focus on customer feedback, making it well-suited for this project.

3.7.1 Agile Methodology

Agile methodology emphasizes iterative development, where the project is divided into small, manageable units called sprints. Each sprint involves a cycle of planning, designing, development, testing, releasing and feedback. This approach allows for continuous improvement and adaptation to changing requirements. Key principles of Agile include: Customer Collaboration, incremental progress, adaptability

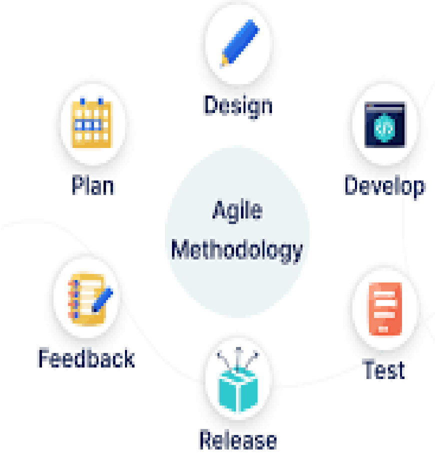


Figure 1: Steps in agile SDLC model.

Planning: During the Planning phase, the team focused on identifying and prioritizing features necessary for the system, based on user stories and stakeholder input. Key activities included: Resource Allocation and project planning.

Design: In the Design phase, we focused on creating the foundational elements of the software, ensuring a user-friendly interface and effective digital storage. Key activities included: Architecture Design and User Interface Prototypes.

Development: The Development phase involved coding the software incrementally, delivering functional pieces of the application in iterative cycles: Code Reviews

Testing: Testing was an ongoing process aimed at ensuring software quality and meeting user requirements. This process was done insuring these testing types;

i. Performance testing, to check the performance, scalability, and stability of the application under load.
ii. Usability testing, to assess how user-friendly the application is.
iii. Continuous testing, to test at every stage of the development cycle in an automated, ongoing manner.
iv. Unit testing, to verify that individual components or pieces of code (like functions or methods) work correctly.

Release: The release includes deployment of the completed software or features to users, including any necessary training and support.

Maintenance: Reviewing the project or iteration to identify what went well, what didn’t, and how to improve processes for future work. This Provide ongoing support, fix bugs, and make improvements based on user feedback and evolving requirements.

CHAPTER 4: DESIGN AND IMPLEMENTATION

4.1 Introduction

This chapter examines the design and implementation of the card track mobile app. In response to the increasing incidence of lost personal identification cards including National IDs, Student IDs, Employee IDs, Health Insurance IDs, Driver's Licenses, and Membership cards this chapter explores the modern solution developed to address the challenges associated with traditional recovery methods in Rwanda and similar contexts.

The chapter details the tools and technologies that were fundamental to the development of the Card Track app. The use of robust development environments such as Visual Studio facilitated efficient coding and testing. The app was developed using C# and XAML, which provided a powerful combination for creating a responsive and user-friendly interface. For data storage, SQLite was utilized to manage and store user data securely, ensuring efficient handling of ID card information (Microsoft, 2023).

This comprehensive overview provides insight into the systematic approach taken to design and implement a mobile app that modernizes the process of recovering lost IDs, addressing the inefficiencies of traditional methods such as physical searches and media announcements. Through the integration of smartphone technology, the Card Track app aims to offer a more effective, convenient, and timely solution for ID card retrieval (SQLite, 2023).

4.2 System Design

The system structure acts as a blueprint for the entire project, outlining the architecture and helping in the planning and design phases. It ensures that all components are identified, and their interactions are well-defined (Davis, 1989).

4.2.1 Use case diagram

A use case diagram is a visual representation that shows the interactions between different actors (users or external entities) and the system, capturing the various use cases (functionalities or actions) that the system provides. In the context of a Card track mobile app. A use case diagram illustrates how different users and external entities interact with the app's features. (Aleryani, 2016)

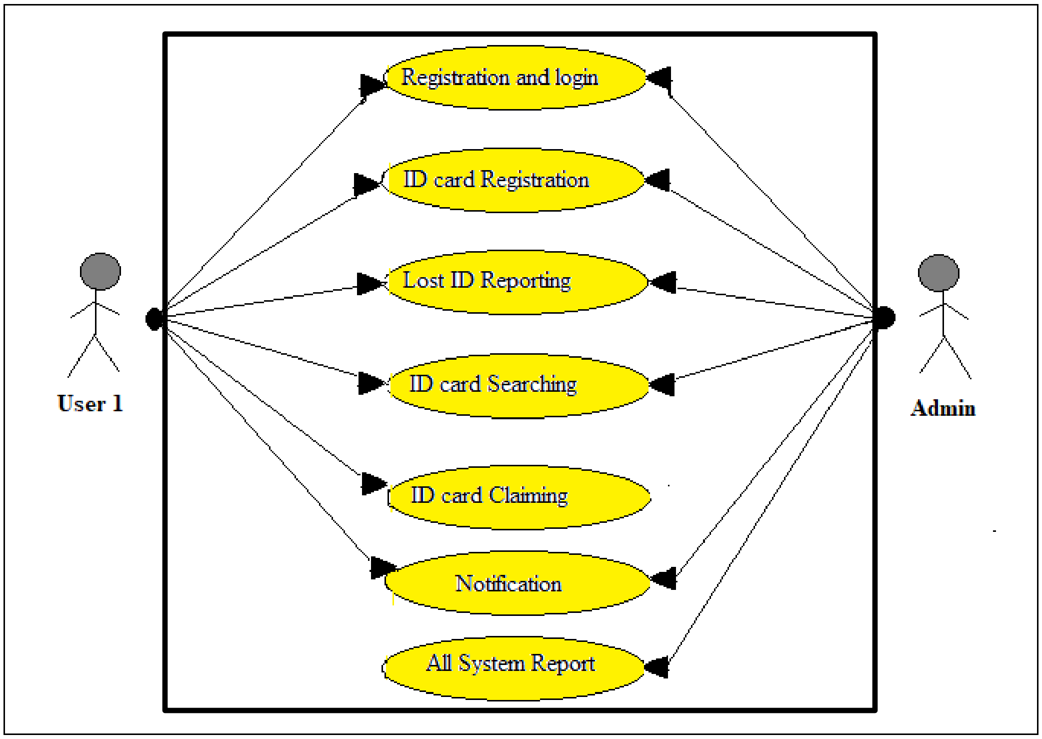


Figure2: The use case diagram of the card track mobile app.

4.2.2 Entity-relationship design

Entity-Relationship (ER) design is a method used in database design to model the relationships between different entities (objects) in a system. In the context of a Card track mobile app, ER design involves defining the entities involved in the app and their relationships. (Kesh, 2017)

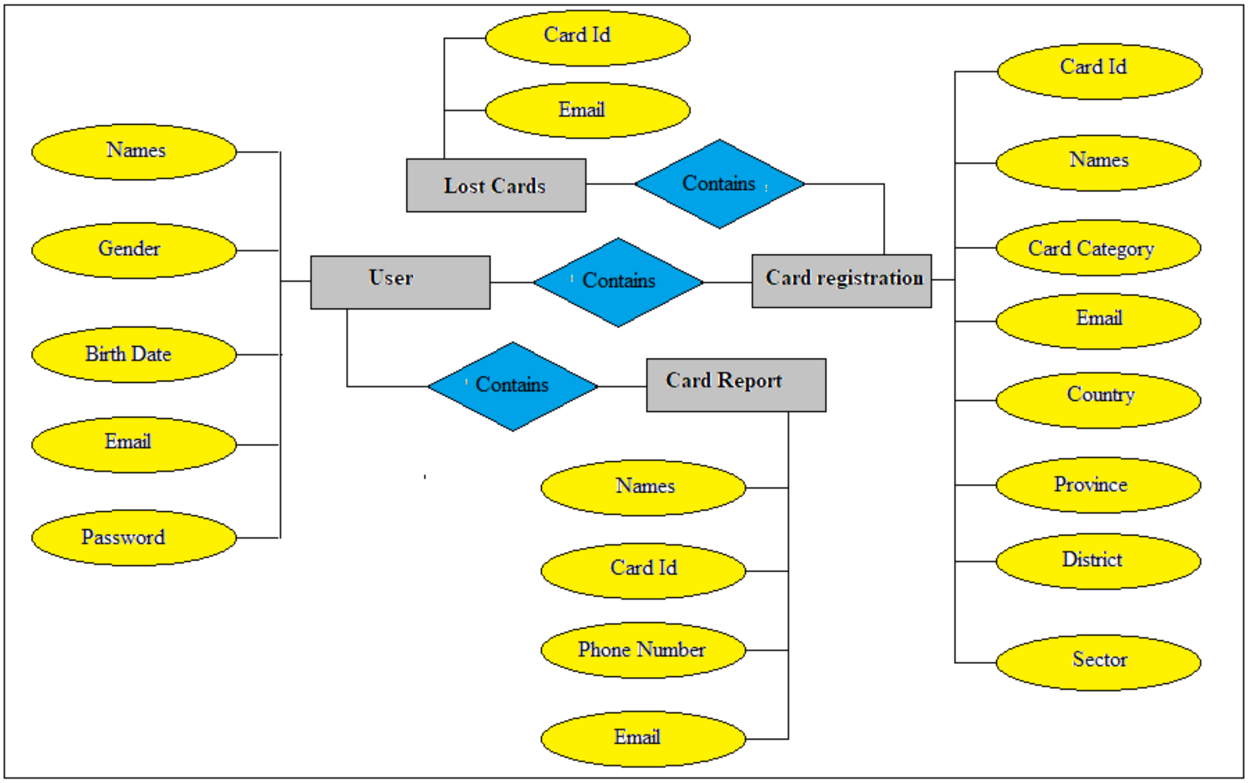


Figure 3*:* The entity relationship diagram.

4.2.3 Data flow diagram

A Data Flow Diagram (DFD) is a visual representation that depicts how data flows within a system or process. In the context of a Card track mobile app, a DFD illustrates the movement of data between different components, such as users, processes, and data stores. (Tao, 1991)

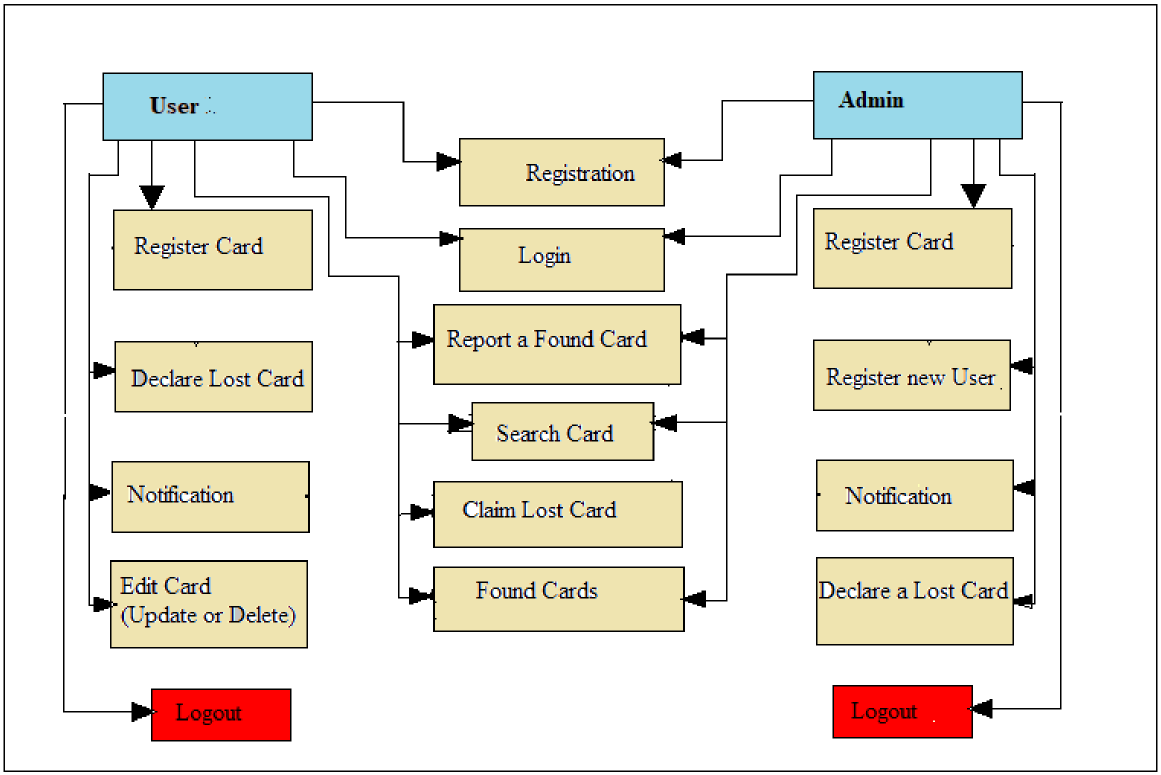


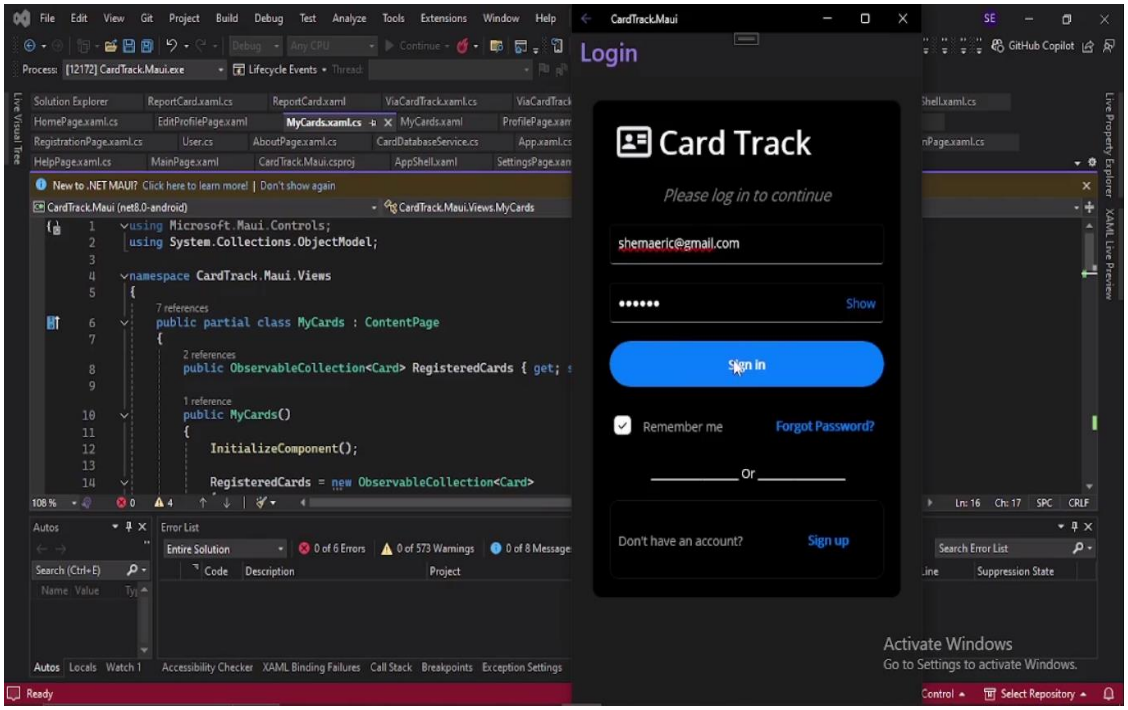
Figure 4: The data flow diagram.

4.3 System Implementation

Screenshots are used to create step-by-step guides, tutorials, or user manuals. They help users understand how to navigate the software, perform tasks, or troubleshoot issues by providing visual references.

4.3.1 Main Page (Login Page)

The main page serves as the login interface for the Card Track app. Users are required to enter their credentials (email and password) to access their accounts. The page validates the input against stored data in the SQLite database to ensure secure authentication. If the login details are incorrect, the user is denied access.



```
using Microsoft.Maui.Controls;
using System.Collections.ObjectModel;

namespace CardTrack.Maui.Views
{
    7 references
    public partial class MyCards : ContentPage
    {
        2 references
        public ObservableCollection<Card> RegisteredCards { get; s
        
        1 reference
        public MyCards()
        {
            InitializeComponent();

            RegisteredCards = new ObservableCollection<Card>
```

Login

Card Track

Please log in to continue

shemaeric@gmail.com

•••••• Show

Sign in

☑ Remember me Forgot Password?

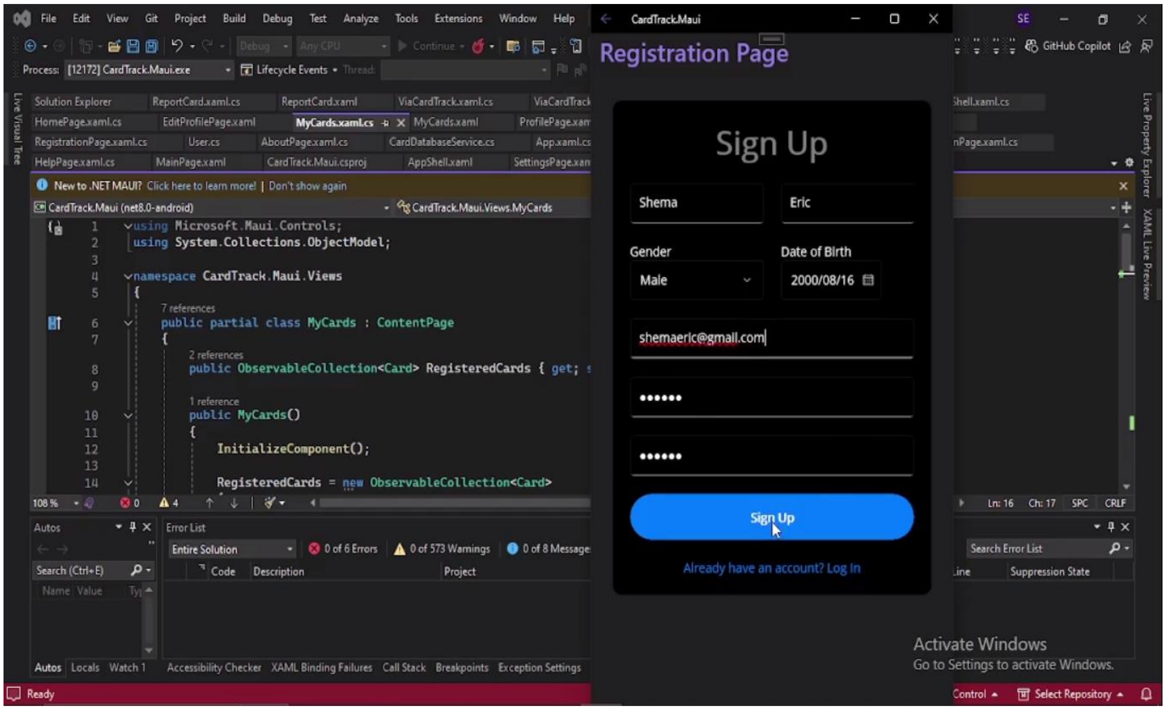
Or

Don't have an account? Sign up

Figure 5: Main page (login page)

4.3.2 Registration Page

The registration page allows new users to create an account by providing personal details such as first name, last name, email, password, and other relevant information. Once the form is filled and submitted, the user data is securely stored in the SQLite database. This page ensures that all necessary information is collected to facilitate the retrieval of lost ID cards.



```
using Microsoft.Maui.Controls;
using System.Collections.ObjectModel;

namespace CardTrack.Maui.Views
{
    7 references
    public partial class MyCards : ContentPage
    {
        2 references
        public ObservableCollection<Card> RegisteredCards { get;

        1 reference
        public MyCards()
        {
            InitializeComponent();

            RegisteredCards = new ObservableCollection<Card>
```

CardTrack.Maui

Registration Page

Sign Up

Shema

Eric

Gender: Male

Date of Birth: 2000/08/16

shemaeric@gmail.com

••••••

••••••

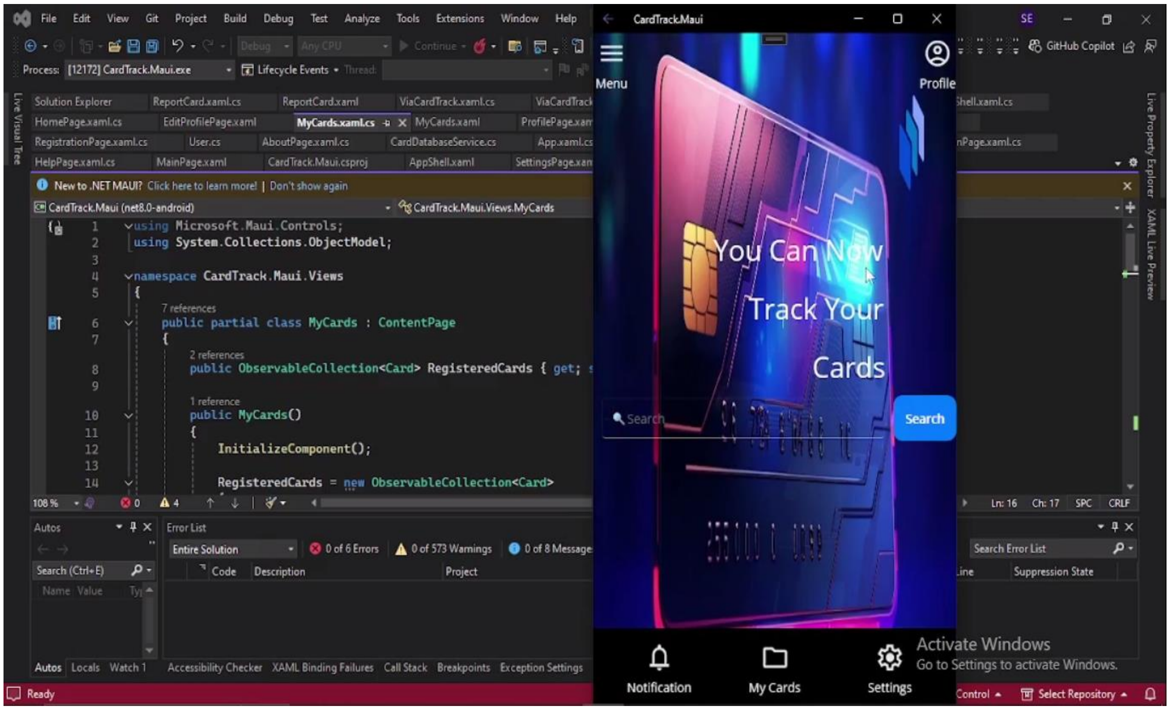
Sign Up

Already have an account? Log In

Figure 6: Registration page.

4.3.3 Home Page

The home page acts as the central navigation hub of the Card Track app. It contains image buttons that direct users to various key features of the app, including the Profile Page, Notification Page, My Cards List, Menu Modal Page, and Settings Page. This page provides easy access to all primary functionalities of the app in a user-friendly interface.

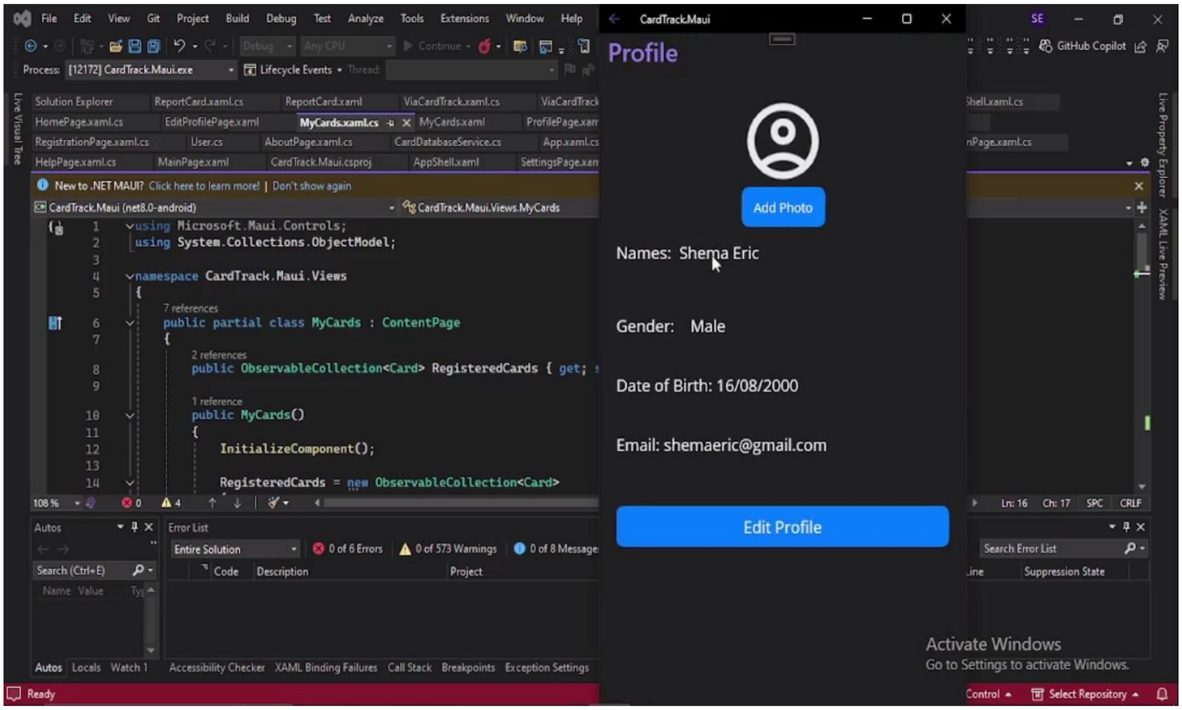


```
using Microsoft.Maui.Controls;
using System.Collections.ObjectModel;

namespace CardTrack.Maui.Views
{
    public partial class MyCards : ContentPage
    {
        public ObservableCollection<Card> RegisteredCards { get; s

        public MyCards()
        {
            InitializeComponent();

            RegisteredCards = new ObservableCollection<Card>
```

Figure 7: Home page

4.3.4 Profile Page

The profile page displays the user’s personal information, such as their name, email, gender, date of birth, and location details. Users can view and edit their profile details as needed. This page helps in managing the user’s account information, which is crucial for personalized services within the app.



```
using Microsoft.Maui.Controls;
using System.Collections.ObjectModel;

namespace CardTrack.Maui.Views
{
    public partial class MyCards : ContentPage
    {
        public ObservableCollection<Card> RegisteredCards { get; s

        public MyCards()
        {
            InitializeComponent();

            RegisteredCards = new ObservableCollection<Card>
```

Profile

Names: Shema Eric

Gender: Male

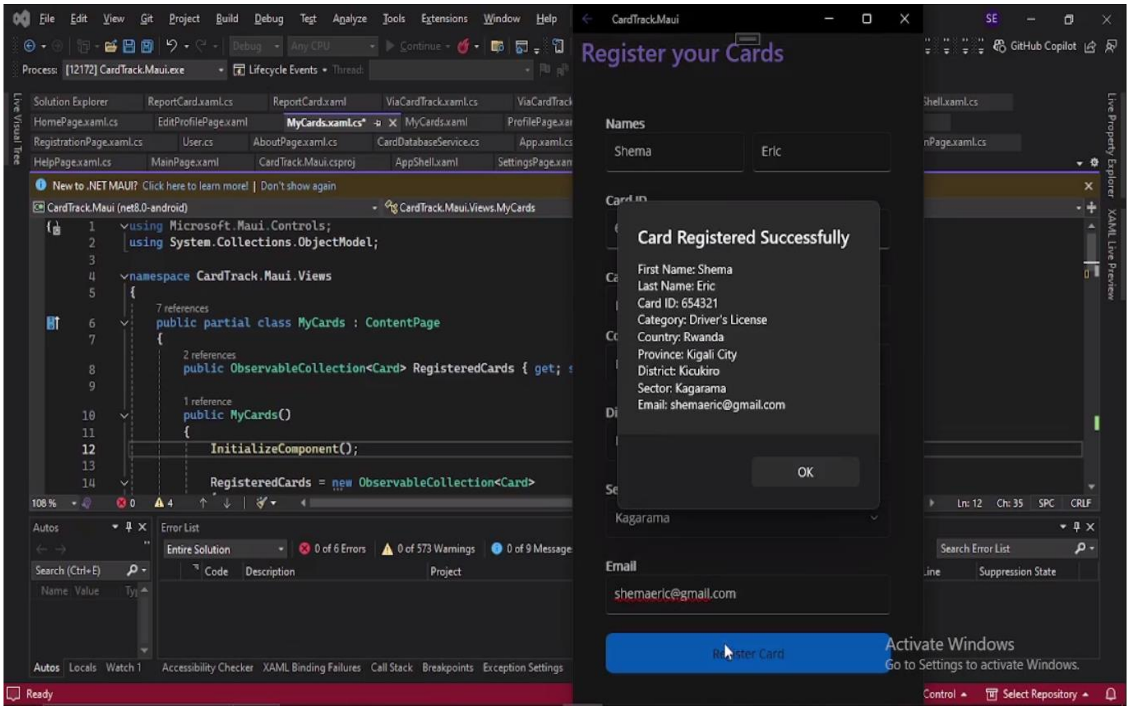
Date of Birth: 16/08/2000

Email: shemaeric@gmail.com

Figure 8: Profile page.

4.3.5 Card registration page

The card registration page, the user registers all card details that are needed for the retrieval in terms it is lost. This page helps in putting the information in the database in order to retrieve them in terms it is needed.



```
using Microsoft.Maui.Controls;
using System.Collections.ObjectModel;

namespace CardTrack.Maui.Views
{
    public partial class MyCards : ContentPage
    {
        public ObservableCollection<Card> RegisteredCards { get;

        public MyCards()
        {
            InitializeComponent();

            RegisteredCards = new ObservableCollection<Card>
```

Register your Cards

Names: Shema | Eric

Email: shemaeric@gmail.com

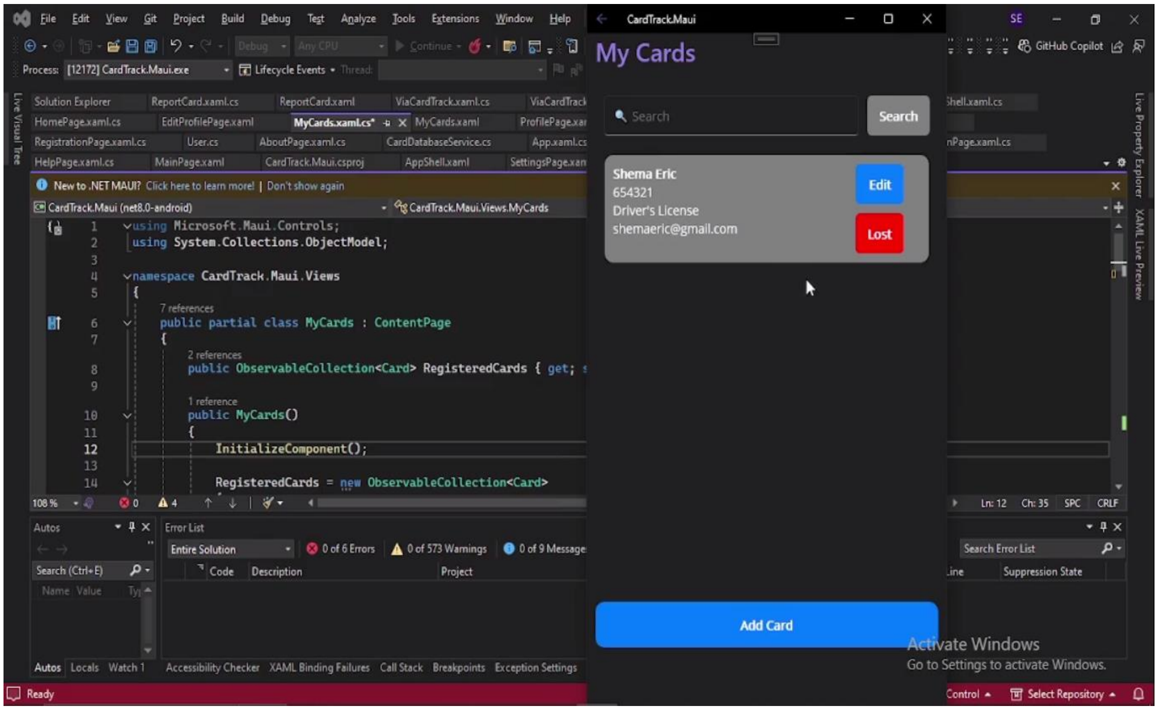
Card Registered Successfully

First Name: Shema
Last Name: Eric
Card ID: 654321
Category: Driver's License
Country: Rwanda
Province: Kigali City
District: Kicukiro
Sector: Kagarama
Email: shemaeric@gmail.com

Figure 9: Card registration page.

4.3.6 My Cards List Page

The cards list page lists all the ID cards that the user has registered within the app. It provides a summary view of each card, including card type and card ID. Users can easily manage their cards, add new cards, or remove outdated ones. This page is essential for tracking and managing all registered identification cards.



New to .NET MAUI? Click here to learn more! | Don't show again

```
using Microsoft.Maui.Controls;
using System.Collections.ObjectModel;

namespace CardTrack.Maui.Views
{
    7 references
    public partial class MyCards : ContentPage
    {
        2 references
        public ObservableCollection<Card> RegisteredCards { get; s

        1 reference
        public MyCards()
        {
            InitializeComponent();

            RegisteredCards = new ObservableCollection<Card>
```

108 % 0 4

0 of 6 Errors 0 of 573 Warnings 0 of 9 Message

Ln: 12 Ch: 35 SPC CRLF

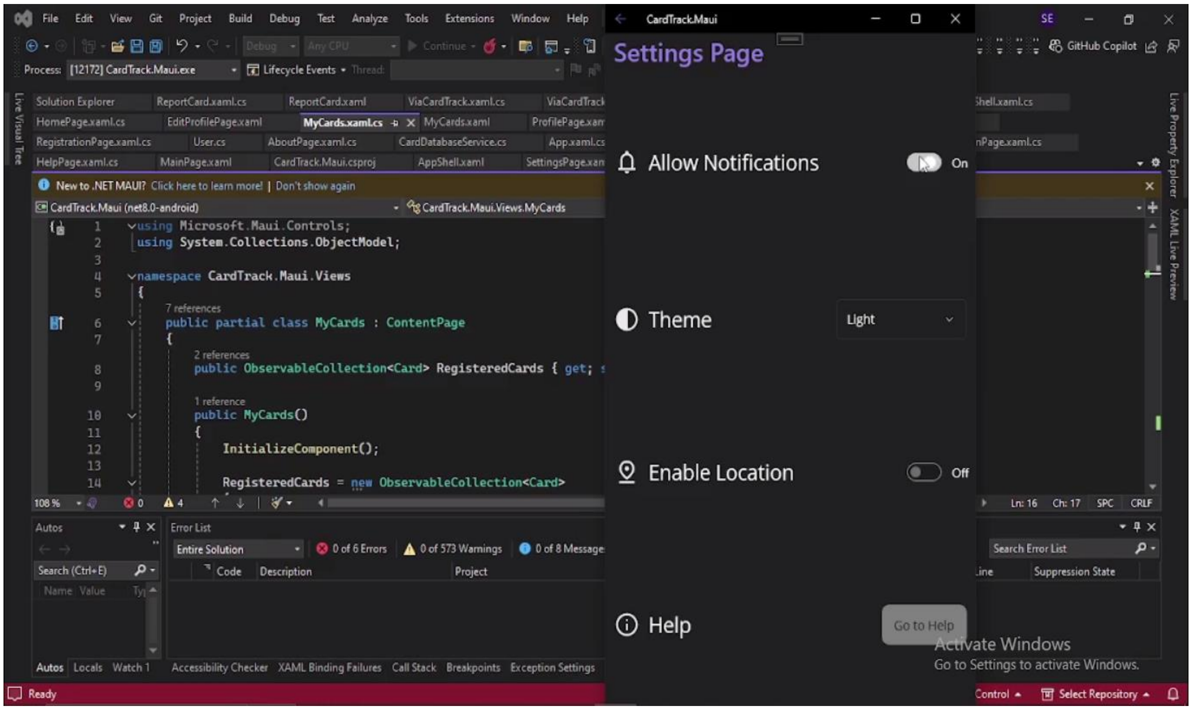
My Cards

Shema Eric
654321
Driver's License
shemaeric@gmail.com

Figure 10: Card list after being registered.

4.3.7 Settings Page

The settings page allows users to configure various app settings according to their preferences. Users can change app behavior, manage notifications, and adjust other settings. This page ensures that users can customize their experience within the Card Track app.



```
using Microsoft.Maui.Controls;
using System.Collections.ObjectModel;

namespace CardTrack.Maui.Views
{
    7 references
    public partial class MyCards : ContentPage
    {
        2 references
        public ObservableCollection<Card> RegisteredCards { get; s
        
        1 reference
        public MyCards()
        {
            InitializeComponent();
            
            RegisteredCards = new ObservableCollection<Card>
```

108 % ⊗ 0 ⚠ 4

Error List: Entire Solution ⊗ 0 of 6 Errors ⚠ 0 of 573 Warnings ⓘ 0 of 8 Messages

Code	Description	Project

Autos | Locals | Watch 1 | Accessibility Checker | XAML Binding Failures | Call Stack | Breakpoints | Exception Settings

Ln: 16 Ch: 17 SPC CRLF

CardTrack.Maui

Settings Page

Allow Notifications On

Theme Light

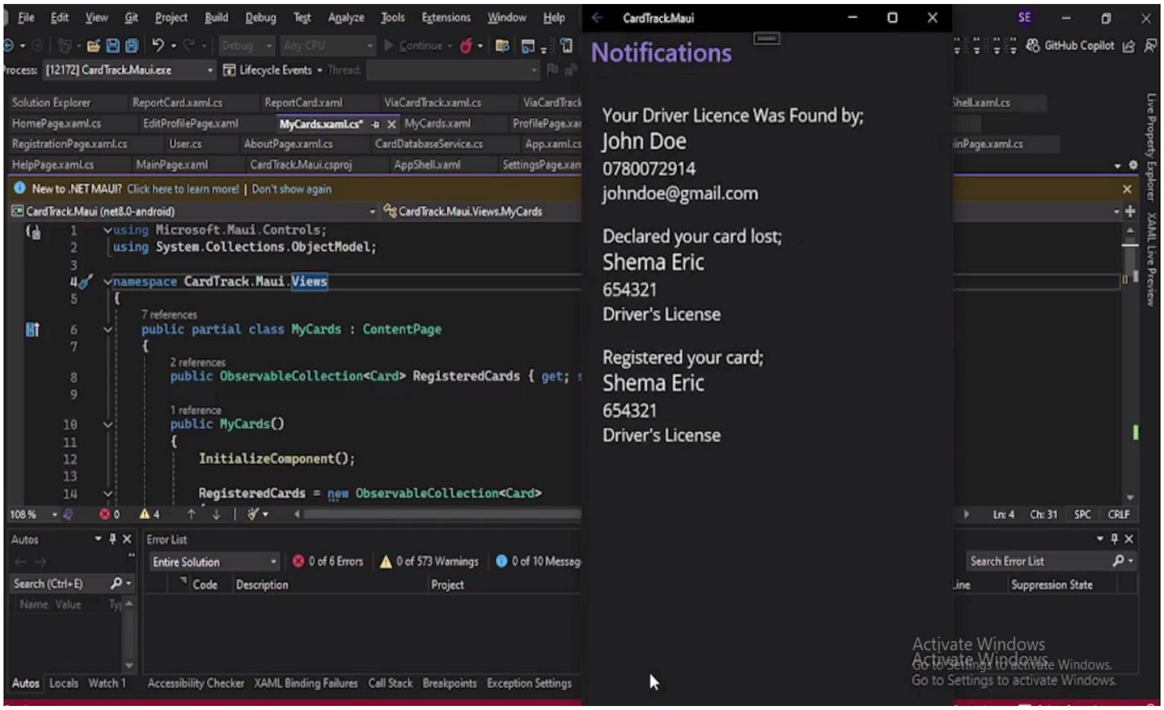
Enable Location Off

Help Go to Help

Figure 11: Settings page.

4.3.8 Notification Page

The notification page displays all notifications related to the user’s ID cards, such as updates about found cards, reminders, and other important alerts. This page ensures that users stay informed about the status of their registered ID cards and any actions they might need to take.



```
using Microsoft.Maui.Controls;
using System.Collections.ObjectModel;

namespace CardTrack.Maui.Views
{
    7 references
    public partial class MyCards : ContentPage
    {
        2 references
        public ObservableCollection<Card> RegisteredCards { get; s

        1 reference
        public MyCards()
        {
            InitializeComponent();

            RegisteredCards = new ObservableCollection<Card>
```

Notifications

Your Driver Licence Was Found by;
John Doe
0780072914
johndoe@gmail.com

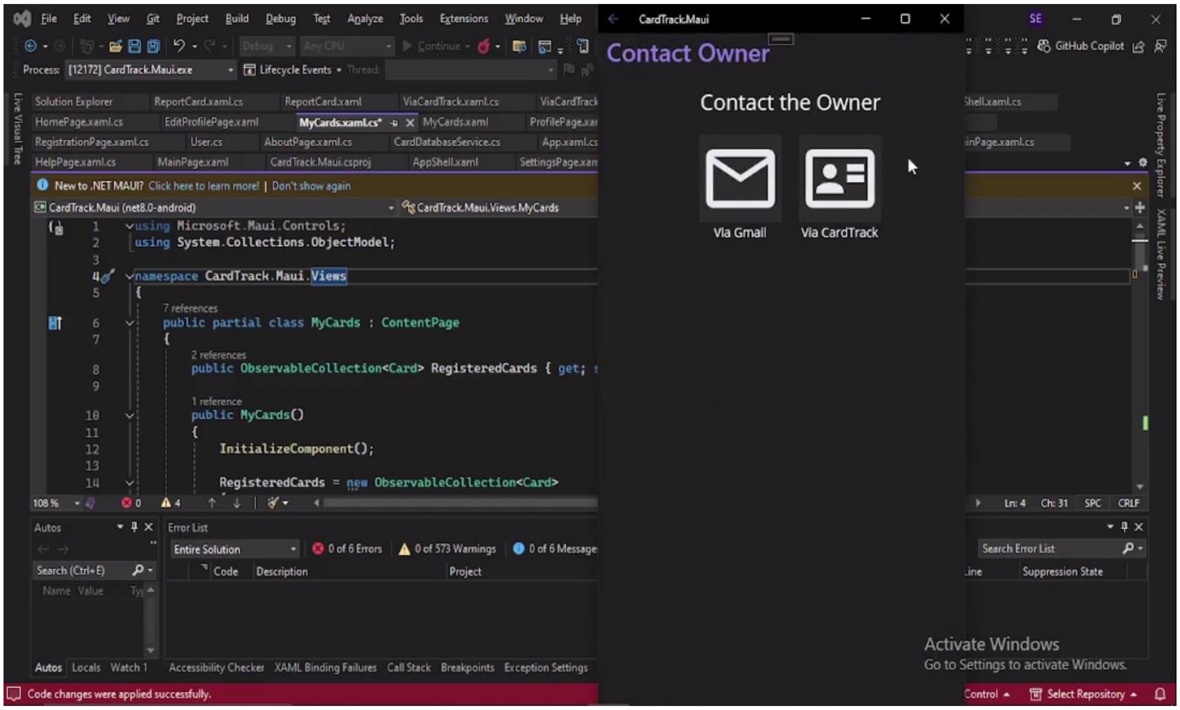
Declared your card lost;
Shema Eric
654321
Driver's License

Registered your card;
Shema Eric
654321
Driver's License

Figure 12: Notification page.

4.3.9 Contact the owner Page

The Contact the owner page allows users to report a lost or found ID card using the Card Track app’s built-in email functionality. Users can fill out a form with details about the card, and the report is sent to the relevant parties. This page streamlines the process of reporting lost or found ID cards, making it more efficient and accessible.



New to .NET MAUI? Click here to learn more! | Don't show again

```
using Microsoft.Maui.Controls;
using System.Collections.ObjectModel;

namespace CardTrack.Maui.Views
{
    public partial class MyCards : ContentPage
    {
        public ObservableCollection<Card> RegisteredCards { get; s

        public MyCards()
        {
            InitializeComponent();

            RegisteredCards = new ObservableCollection<Card>
```

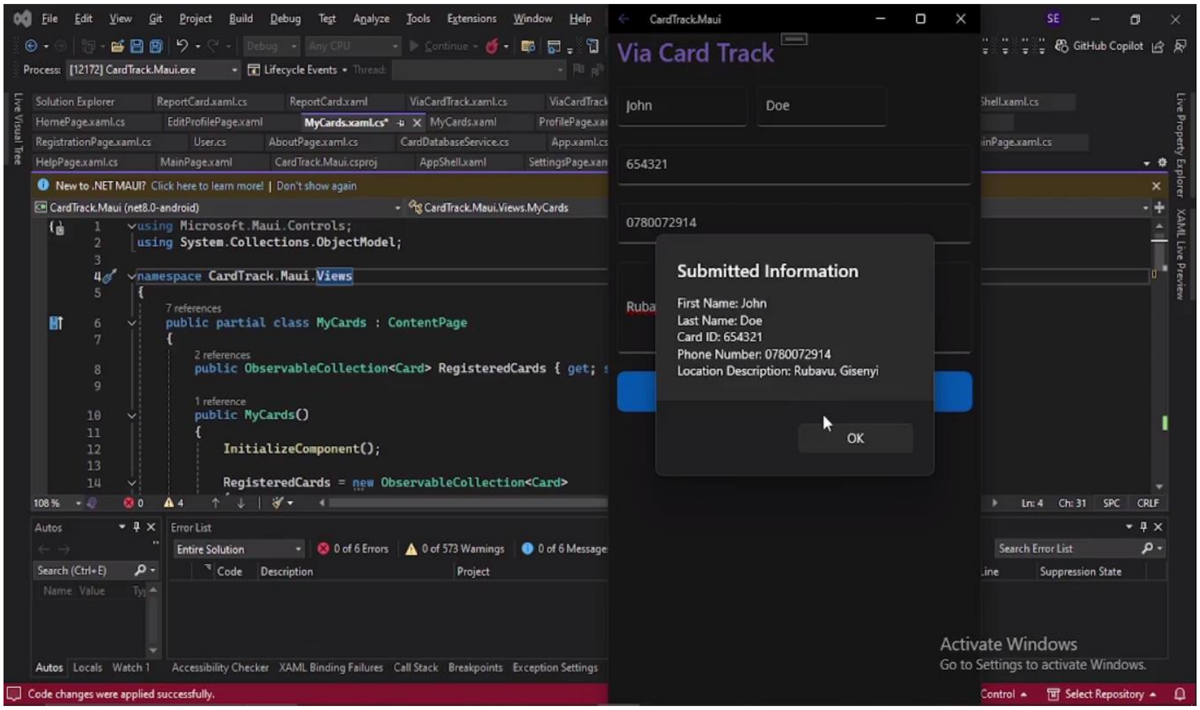
Activate Windows
Go to Settings to activate Windows.

Code changes were applied successfully.

Figure 13: Contact the owner page.

4.3.10 Report via Card Track App Page

This page provides a direct way for users to report lost or found cards within the app itself, bypassing email. Users can fill out the necessary details and submit the report directly through the app. This functionality ensures that users have a quick and integrated method to report card issues without needing external communication tools.



```
using Microsoft.Maui.Controls;
using System.Collections.ObjectModel;

namespace CardTrack.Maui.Views
{
    7 references
    public partial class MyCards : ContentPage
    {
        2 references
        public ObservableCollection<Card> RegisteredCards { get; s

        1 reference
        public MyCards()
        {
            InitializeComponent();

            RegisteredCards = new ObservableCollection<Card>
```

Via Card Track

John

Doe

654321

0780072914

Ruba

Submitted Information

First Name: John
Last Name: Doe
Card ID: 654321
Phone Number: 0780072914
Location Description: Rubavu, Gisenyi

Code changes were applied successfully.

Figure 14: Report via Card Track App Page.

CHAPTER 5: CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The development of the Card Track mobile app has addressed a critical need for a more effective and efficient solution for recovering lost identification cards in Rwanda. Traditional methods, such as physical searches and media announcements, have proven to be time-consuming, costly, and often ineffective. By leveraging modern technology, including .NET MAUI for cross-platform development, SQLite for secure data management, and a user-friendly interface designed with XAML, the Card Track app offers a streamlined process for reporting and retrieving lost IDs. The app's various features, including registration, secure login, profile management, card tracking, and communication tools, all contribute to a comprehensive system that simplifies the recovery process. Ultimately, this project represents a significant step forward in modernizing ID card recovery in Rwanda, making it more accessible and efficient for all users.

5.2 Recommendations

The successful development and implementation of the Card Track mobile app represent a significant advancement in the process of recovering lost identification cards. However, to maximize the app's potential and ensure its widespread adoption, several key stakeholders can play a crucial role. The following recommendations are directed at specific organizations and future researchers, aiming to enhance the app's functionality, promote its use, and ensure that it meets the needs of both users and institutions in Rwanda;

5.2.1 INES-Ruhengeri

It is recommended that INES-Ruhengeri consider incorporating the Card Track app into its administrative systems, particularly for managing student and staff identification cards. The app could be a valuable tool for reducing the incidence of lost IDs on campus and ensuring that lost cards are promptly returned to their rightful owners. Additionally, INES Ruhengeri could provide support for further development and refinement of the app, potentially offering it as a service to other institutions.

5.2.2 Rwanda National Police (RNP)

The Rwanda National Police (RNP) could benefit from adopting the Card Track app as part of its efforts to combat the issue of lost identification cards. The app’s secure database and communication features could be integrated into RNP’s existing systems, allowing for quicker and more efficient recovery of lost IDs. RNP could also collaborate with the app’s developers to enhance security measures and ensure that the app meets national standards for data protection.

5.2.3 National ID Agency (NIDA)

The National ID Agency (NIDA) is encouraged to explore partnerships with the developers of the Card Track app to integrate its functionalities into the national ID management system. By doing so, NIDA could offer a modernized service that helps citizens recover lost National IDs more efficiently. This collaboration could also include the development of additional features tailored to the needs of NIDA, such as automated notifications for recovered IDs and streamlined reporting processes.

5.2.4 Next Researchers

Future researchers are recommended to build upon the foundation left by the Card Track app by exploring additional features and expanding its scope. Potential areas of investigation could include the integration of biometric verification for added security, the development of a web-based platform to complement the mobile app, and the extension of the app’s services to include other types of lost property. Researchers could also study the app’s impact on ID recovery rates and user satisfaction, providing valuable data to inform further improvements.

REFERENCES

A Modern Introduction to Programming. (2018). Marijn Haverbeke.

(FTC), F. T. (2024). *Lost or Stolen Identification.*

Agency, N. I. (2019). *National Identification System in Rwanda.* Kigali, Rwanda: NIDA. Retrieved from https://www.nida.gov.rw/publications/

Aleryani, A. Y. (2016). Comparative study between data flow diagram and use case diagram. *International Journal of Scientific and Research Publications, 6*(3), 124-126.

Consortium, S. (2023). *SQLite Documentation.* Duns, Scotland, UK: SQLite.

Davis, F. (1989). *Technollogy Acceptance Model.*

Designing Data-Intensive Applications. (2023). Martin Kleppmann.

Edmondson, A. (2018). The fearless organization: Creating psychological safety in the workplace for learning. *innovation, and growth John Wiley & Sons*.

Find My App Features. (2023). Apple Inc.

FTC. (2024). *Lost or Stolen Identification.* NYC: Federal Trade Commission. Retrieved from https://www.consumer.ftc.gov/articles/0246-lost-or-stolen-identification

Group, W. B. (2019). *Digital Identification Systems: A Path to Enhanced Service Delivery.* Washington, D.C., USA: World Bank. Retrieved from https://www.worldbank.org/en/topic/digitaldevelopment/brief/digital-identification-systems

How Tile Works. (2022). Tile Inc.

inc., A. (2023). *iOS App Development Documentation.* Cupertino, California, USA: Apple Inc.

Jansen, B. J. (1998). The graphical user interface. *ACM SIGCHI Bulletin, 30*(2), 22-26.

Kesh, S. (1995). Evaluating the quality of entity relationship models. *Information and Software Technology, 37*(12), 681-689.

Microsoft. (2023). *.NET MAUI Documentation.* New York. Retrieved from https://docs.microsoft.com/en-us/dotnet/maui/

Moe. (2020). *Revolutionizing Lost Item Recovery with Mobile Technology*. Journal of Mobile Computing and Applications. Retrieved from https://doi.org/10.1016/j.jmca.2020.03.012

Police, R. N. (2021). *Annual Report on Crime Prevention and Investigation.* Kigali, Rwanda: RNP. Retrieved from https://www.police.gov.rw/reports/

Albert, D . (2024, February 9th). Retrieved March 20th, 2024, from Ask Herts: https://ask.herts.ac.uk/replacement-id-cards-lost-damaged-stolen

Ruhengeri, I. (2020). *ICT Integration in Education: Case Study of INES Ruhengeri*. Musanze, Rwanda: INES Ruhengeri. Retrieved from https://ines.ac.rw/publications/

Shores, R. (2023). *Java SE Documentation. Oracle Corporation.* California, USA: Oracle. Retrieved from https://docs.oracle.com/javase/8/docs/

Standardization, I. O. (2017). *Information Security Management Systems.* Geneva, Switzerland: ISO. Retrieved from https://www.iso.org/isoiec-27001-information-security.html

Tao, Y. (1991). Formal definition and verification of data flow diagrams. *Journal of Systems and Software*, *16*(1), 29-36.

Technology, a. I. (2013). *Guidelines for Secure Data Storage.* Gaithersburg, Maryland, USA: NIST. Retrieved from https://csrc.nist.gov/publications/detail/sp/800-88/rev-1/final

What should I do if I lose my student ID card? (2023, October 20th). Retrieved March 26th, 2024, from Bond Univesisty Library: https://bond.libanswers.com/faq/274353

Xamarin. (2023). *Introduction to XAML.* San Francisco, California, USA: Xamarin. Retrieved from https://docs.microsoft.com/en-us/xamarin/xamarin-forms/xaml/

APPENDICES

Appendix A: Source codes

Login page

```
using CardTrack.Maui.Models;
using CardTrack.Maui.Services;
using CardTrack.Maui.Views;
using Microsoft.Maui.Controls;
using System;
using System.Threading.Tasks;

namespace CardTrack.Maui
{
    public partial class MainPage : ContentPage
    {
        private bool _isPasswordVisible = false;
        private DatabaseService _databaseService;

        public MainPage()
        {
            InitializeComponent();


            _databaseService = App.Database;


            var tapGestureRecognizer = new TapGestureRecognizer();
            tapGestureRecognizer.Tapped += OnShowHideLabelTapped;


            showHideLabel.GestureRecognizers.Add(tapGestureRecognizer);
        }

        private void OnShowHideLabelTapped(object sender, EventArgs e)
        {
```

```
    if (passwordEntry == null || showHideLabel == null)
    {
        DisplayAlert("Error", "Password entry or Show/Hide label is not initialized", "OK");
        return;
    }

    _isPasswordVisible = !_isPasswordVisible;
    passwordEntry.IsPassword = !_isPasswordVisible;
    showHideLabel.Text = _isPasswordVisible ? "Hide" : "Show";
}

private async void OnLoginButtonClicked(object sender, EventArgs e)
{
    if (emailEntry == null || passwordEntry == null)
    {
        await DisplayAlert("Error", "Email or Password entry is not initialized", "OK");
        return;
    }

    string email = emailEntry.Text;
    string password = passwordEntry.Text;

    if (await IsAuthenticated(email, password))
    {
        await Navigation.PushAsync(new HomePage());
    }
    else
    {
        await DisplayAlert("Error", "Invalid email or password", "OK");
    }
}

private async Task<bool> IsAuthenticated(string email, string password)
{
```

```
            var user = await _databaseService.GetUserAsync(email);
            return user != null && user.Password == password;
        }

        private async void OnForgotPasswordButtonClicked(object sender, EventArgs e)
        {
            await Navigation.PushAsync(new PasswordRecoveryPage());
        }

        private async void OnCreateAccountButtonClicked(object sender, EventArgs e)
        {
            await Navigation.PushAsync(new RegistrationPage());
        }
    }
}
```

Registration page

```
using CardTrack.Maui.Services;
using CardTrack.Maui.Models;
using System;
using System.Text.RegularExpressions;

namespace CardTrack.Maui.Views
{
    public partial class RegistrationPage : ContentPage
    {
        public RegistrationPage()
        {
            InitializeComponent();
        }

        private async void OnSignUpButtonClicked(object sender, EventArgs e)
        {
            string firstName = firstNameEntry.Text;
            string lastName = lastNameEntry.Text;
string email = emailEntry.Text;
string password = passwordEntry.Text;
string confirmPassword = confirmPasswordEntry.Text;
string gender = genderPicker.SelectedItem?.ToString();
DateTime? dateOfBirth = dateOfBirthPicker.Date;

if (string.IsNullOrWhiteSpace(firstName) ||
    string.IsNullOrWhiteSpace(lastName) ||
    string.IsNullOrWhiteSpace(email) ||
    string.IsNullOrWhiteSpace(password) ||
    string.IsNullOrWhiteSpace(confirmPassword) ||
    string.IsNullOrWhiteSpace(gender) ||
    dateOfBirth == null)
{
    await DisplayAlert("Error", "All fields are required.", "OK");
    return;
}

if (password != confirmPassword)
{
    await DisplayAlert("Error", "Passwords do not match.", "OK");
    return;
}

if (!IsValidEmail(email))
{
    await DisplayAlert("Error", "Please enter a valid email address.", "OK");
    return;
}


var existingUser = await App.Database.GetUserAsync(email);
if (existingUser != null)
{
```

```
            await DisplayAlert("Error", "Email already in use.", "OK");
            return;
        }

        var newUser = new User
        {
            FirstName = firstName,
            LastName = lastName,
            Email = email,
            Password = password,
            Gender = gender,
            DateOfBirth = dateOfBirth.Value
        };

        await App.Database.SaveUserAsync(newUser);

        await DisplayAlert("Success", "Account created successfully!", "OK");



        await Navigation.PopAsync();
    }

    private bool IsValidEmail(string email)
    {
        if (string.IsNullOrWhiteSpace(email))
            return false;

        try
        {
            return Regex.IsMatch(email,
                @"^[^@\s]+@[^@\s]+\.[^@\s]+$",
                RegexOptions.IgnoreCase, TimeSpan.FromMilliseconds(250));
        }
        catch (RegexMatchTimeoutException)
```

```
            {
                return false;
            }
        }

        private async void OnLoginLabelTapped(object sender, EventArgs e)
        {

            await Navigation.PushAsync(new MainPage());
        }
    }
}
```

Card registration page

```
using Microsoft.Maui.Controls;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text.RegularExpressions;

namespace CardTrack.Maui.Views
{
    public partial class CardRegistrationPage : ContentPage
    {
        private Dictionary<string, List<string>> provincesByCountry = new Dictionary<string,
List<string>>
        {
            { "Rwanda", new List<string> { "Kigali City", "Northern Province", "Southern Province", "Eastern
Province", "Western Province" } }
        };

        private Dictionary<string, List<string>> districtsByProvince = new Dictionary<string, List<string>>
        {
            { "Kigali City", new List<string> { "Gasabo", "Kicukiro", "Nyarugenge" } },
            { "Northern Province", new List<string> { "Burera", "Gakenke", "Gicumbi", "Musanze", "Rulindo" }
},
            { "Southern Province", new List<string> { "Gisagara", "Huye", "Kamonyi", "Muhanga",
"Nyamagabe", "Nyanza", "Nyaruguru", "Ruhango" } },
            { "Eastern Province", new List<string> { "Bugesera", "Gatsibo", "Kayonza", "Kirehe", "Ngoma",
"Nyagatare", "Rwamagana" } },
            { "Western Province", new List<string> { "Karongi", "Ngororero", "Nyabihu", "Nyamasheke",
"Rubavu", "Rusizi", "Rutsiro" } }
        };

        private Dictionary<string, List<string>> sectorsByDistrict = new Dictionary<string, List<string>>
        {
            { "Gasabo", new List<string> { "Bumbogo", "Gacuriro", "Gikambura", "Gikondo", "Gisozi", "Jali",
"Kabuye", "Kinyinya", "Kivugiza", "Ndera", "Remera", "Rusororo", "Rukiri I", "Rukiri II", "Saruyange" } },
            { "Kicukiro", new List<string> { "Bibare", "Gatenga", "Gikambura", "Gisozi", "Kagarama",
"Kanombe", "Kicukiro", "Kinyinya", "Niboye", "Nyakabanda", "Ruhuha", "Zindiro" } },
            { "Nyarugenge", new List<string> { "Akumunigo", "Biryogo", "Gatenga", "Gitega", "Kabuye",
"Kicukiro", "Kimisagara", "Nyarugenge", "Rwandex", "Ruzindana", "Taba" } },
            { "Burera", new List<string> { "Bugarama", "Cyeru", "Gahunga", "Gakenke", "Gatunda", "Gicumbi",
"Kinyababa", "Mparo", "Muhoza", "Rugarama", "Rumburwa" } },
            { "Gakenke", new List<string> { "Busengo", "Gakenke", "Gathiga", "Gicaca", "Karambo", "Kivuye",
"Muniga", "Muhondo", "Rutare", "Rugarama" } },
            { "Gicumbi", new List<string> { "Bwisige", "Byumba", "Cyumba", "Giti", "Kaniga", "Kageyo",
"Ngarama", "Rwabicuma", "Rutare", "Shingiro" } },
            { "Musanze", new List<string> { "Bugarama", "Gacaca", "Gahinga", "Kabeza", "Kivuye", "Musanze",
"Nyange", "Nyamabuye", "Rwaza" } },
            { "Rulindo", new List<string> { "Base", "Bumbogo", "Cyinzuzi", "Gakenke", "Kinigi", "Kivuye",
"Muniga", "Nyange", "Nyamabuye", "Rugarama" } },
            { "Gisagara", new List<string> { "Gakoma", "Gikonko", "Gishubi", "Gishamvu", "Kansi", "Kibirizi",
"Kivumu", "Nyamiyaga", "Nyarabizi", "Zaza" } },
            { "Huye", new List<string> { "Bujumbura", "Gitovu", "Huye", "Maraba", "Muniga", "Rusatira",
"Tare", "Save", "Ndiza", "Ndora" } },
            { "Kamonyi", new List<string> { "Byimana", "Gacurabwenge", "Kamonyi", "Kivumu", "Nyamiyaga",
"Nyarubaka", "Rugalika", "Ruhango", "Rutunga" } },
    { "Muhanga", new List<string> { "Birambo", "Cyeza", "Gashora", "Gitovu", "Kabacuzi", "Kabuye",
"Kamonyi", "Muhanga", "Nyamiyaga", "Shyogwe" } },
    { "Nyamagabe", new List<string> { "Buruhukiro", "Gasaka", "Kaduha", "Kansi", "Muniga",
"Nyamiyaga", "Nyaruguru", "Rukamba", "Rwankuba" } },
    { "Nyanza", new List<string> { "Busasamana", "Cyabakamyi", "Nyanza", "Rwabusoro",
"Rwabusoro", "Ruhango", "Kibilizi", "Rukamba", "Nyarubaka" } },
    { "Nyaruguru", new List<string> { "Bugarama", "Cyahinda", "Kibeho", "Muniga", "Nkomero",
"Nyabimata", "Rukondo", "Rwabusoro", "Shyogwe" } },
    { "Ruhango", new List<string> { "Byimana", "Gashora", "Gitovu", "Kamonyi", "Ruhango", "Runda",
"Shyogwe", "Mbuye", "Nyamiyaga" } },
    { "Bugesera", new List<string> { "Berenge", "Gashora", "Gatenga", "Kamabuye", "Nyarugenge",
"Rweru", "Rukara", "Shyogwe", "Zaza" } },
    { "Gatsibo", new List<string> { "Gasange", "Gicumbi", "Kageyo", "Kiziguro", "Muvumba",
"Nyagatare", "Rwimiyaga", "Rugarama", "Rulindo", "Zaza" } },
    { "Kayonza", new List<string> { "Bwisige", "Kayonza", "Kibungo", "Kirehe", "Kivuye", "Muhazi",
"Nyamirama", "Rukara", "Rurenge", "Zaza" } },
    { "Kirehe", new List<string> { "Gatore", "Gikomero", "Kirehe", "Muhura", "Muniga", "Nyamabuye",
"Rukara", "Rukiga", "Zaza" } },
    { "Ngoma", new List<string> { "Birembo", "Bugesera", "Cyamuganga", "Gashanda", "Kazo",
"Kayonza", "Ngoma", "Rukara", "Zaza" } },
    { "Nyagatare", new List<string> { "Gatunda", "Gatsibo", "Kiyombe", "Matimba", "Nyagatare",
"Rwimiyaga", "Rugarama", "Rutare", "Zaza" } },
    { "Rwamagana", new List<string> { "Bwisige", "Kayonza", "Kigarama", "Muhazi", "Nyakabanda",
"Rwamagana", "Zaza" } },
    { "Karongi", new List<string> { "Bwishyura", "Gihombo", "Gitesi", "Karongi", "Kibuye", "Muringa",
"Nyundo", "Ruganda", "Zaza" } },
    { "Ngororero", new List<string> { "Bwishyura", "Gakoma", "Gikoro", "Gitesi", "Karongi", "Kibirizi",
"Ngororero", "Nyundo", "Ruganda" } },
    { "Nyabihu", new List<string> { "Bugarama", "Gihombo", "Gitesi", "Karongi", "Kibuye", "Kivuye",
"Nyabihu", "Nyundo", "Ruganda" } },
    { "Nyamasheke", new List<string> { "Bugarama", "Gihombo", "Gitesi", "Karongi", "Kibuye",
"Nyamasheke", "Nyundo", "Ruganda" } },
    { "Rubavu", new List<string> { "Bugeshi", "Cyanzarwe", "Gisenyi", "Ngoma", "Kibuye", "Kivuye",
"Nyundo", "Rutsiro", "Zaza" } },
    { "Rusizi", new List<string> { "Bugarama", "Cyangugu", "Gihombo", "Karongi", "Kibuye",
"Nyundo", "Ruganda", "Rusizi", "Zaza" } },
    { "Rutsiro", new List<string> { "Bugarama", "Gihombo", "Gitesi", "Karongi", "Kibuye", "Kivuye",
"Nyundo", "Rutsiro", "Zaza" } }
        };

        public CardRegistrationPage()
        {
    InitializeComponent();
    InitializeCountryPicker();
        }

        private void InitializeCountryPicker()
        {
    CountryPicker.ItemsSource = provincesByCountry.Keys.ToList();
        }

        private void OnCountrySelected(object sender, EventArgs e)
        {
    if (CountryPicker.SelectedItem is string selectedCountry &&
provincesByCountry.ContainsKey(selectedCountry))
    {
        ProvincePicker.ItemsSource = provincesByCountry[selectedCountry];
        ProvincePicker.SelectedItem = null;
        DistrictPicker.ItemsSource = null;
        SectorPicker.ItemsSource = null;
    }
        }

        private void OnProvinceSelected(object sender, EventArgs e)
        {
    if (ProvincePicker.SelectedItem is string selectedProvince &&
districtsByProvince.ContainsKey(selectedProvince))
    {
        DistrictPicker.ItemsSource = districtsByProvince[selectedProvince];
        DistrictPicker.SelectedItem = null;
        SectorPicker.ItemsSource = null;
    }
        }

        private void OnDistrictSelected(object sender, EventArgs e)
        {

    if (DistrictPicker.SelectedItem is string selectedDistrict &&
sectorsByDistrict.ContainsKey(selectedDistrict))
    {
        SectorPicker.ItemsSource = sectorsByDistrict[selectedDistrict];
    }
        }



        private async void OnRegisterCardButtonClicked(object sender, EventArgs e)
        {
    string firstName = FirstNameEntry.Text;
    string lastName = LastNameEntry.Text;
    string cardId = CardIdEntry.Text;
    string category = CategoryPicker.SelectedItem?.ToString();
    string country = CountryPicker.SelectedItem?.ToString();
    string province = ProvincePicker.SelectedItem?.ToString();
    string district = DistrictPicker.SelectedItem?.ToString();
    string sector = SectorPicker.SelectedItem?.ToString();
    string email = EmailEntry.Text;

    if (!IsValidEmail(email))
    {
        await DisplayAlert("Error", "Please enter a valid email address.", "OK");
        return;
```

```
}

bool IsValidEmail(string email)
{
    if (string.IsNullOrWhiteSpace(email))
        return false;

    try
    {
        return Regex.IsMatch(email,
            @"^[^@\s]+@[^@\s]+\.[^@\s]+$",
            RegexOptions.IgnoreCase, TimeSpan.FromMilliseconds(250));
    }
    catch (RegexMatchTimeoutException)
    {
        return false;
    }
}

if (string.IsNullOrEmpty(firstName) ||
    string.IsNullOrEmpty(lastName) ||
    string.IsNullOrEmpty(cardId) ||
    string.IsNullOrEmpty(category) ||
    string.IsNullOrEmpty(country) ||
    string.IsNullOrEmpty(province) ||
    string.IsNullOrEmpty(district) ||
    string.IsNullOrEmpty(sector) ||
    string.IsNullOrEmpty(email))
{
    await DisplayAlert("Error", "All fields are required.", "OK");
    return;
}

string message = $"First Name: {firstName}\n" +
                            $"Last Name: {lastName}\n" +
                            $"Card ID: {cardId}\n" +
                            $"Category: {category}\n" +
                            $"Country: {country}\n" +
                            $"Province: {province}\n" +
                            $"District: {district}\n" +
                            $"Sector: {sector}\n" +
                            $"Email: {email}";

            await DisplayAlert("Card Registered Successfully", message, "OK");
            await Navigation.PopAsync();
        }



    }
}
```

Profile page

```
using Microsoft.Maui.Controls;
using CardTrack.Maui.Services;
using CardTrack.Maui.Models;
using System.Threading.Tasks;

namespace CardTrack.Maui.Views
{
    public partial class ProfilePage : ContentPage
    {
        public ProfilePage()
        {
            InitializeComponent();
            LoadUserData();
        }

        private async void LoadUserData()
        {
            var user = await App.Database.GetCurrentUserAsync();
            if (user != null)
            {

                namesLabel.Text = $"{user.FirstName} {user.LastName}";
                emailLabel.Text = user.Email;
                genderLabel.Text = user.Gender;
                dateOfBirthLabel.Text = user.DateOfBirth.ToString("yyyy/MM/dd");

                profileImage.Source = user.ProfileImagePath ?? "default_profile_photo.png";
            }
        }

        private async void OnEditProfileButtonClicked(object sender, EventArgs e)
        {

            await Navigation.PushAsync(new EditProfilePage());
        }

        private async void OnEditPhotoButtonClicked(object sender, EventArgs e)
        {

        }
    }
}
```