COUNTERFEIT CURRENCY NOTE SOURCING SYSTEM

A DISSERTATION

Submitted by

AHMED FAYAZ W

2018239001

in partial fulfillment for the award of the degree of

MASTER OF SCIENCE (INTEGRATED)

in

COMPUTER SCIENCE

FACULTY OF SCIENCE AND HUMANITIES
ANNA UNIVERSITY
CHENNAI 600025

January 2022

ANNA UNIVERSITY
CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this dissertation titled **"COUNTERFEIT CURRENCY NOTE SOURCING SYSTEM"** is the Bonafide work of **Mr. AHMED FAYAZ W** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other dissertation based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

(Dr. L. JONES TARCIUS DOSS)
Professor and Head
Department of Mathematics
Anna University
Chennai 600025

(Dr. G. SUMATHI)
Supervisor
Teaching Fellow
Department of Mathematics
Anna University
Chennai 600025

(**Mr. K. KAMALA KANNAN**)
Project Coordinator
Teaching Fellow
Department of Mathematics
Anna University
Chennai 600025

ABSTRACT

The main objective of this project is to find the Source of the Counterfeit Currency Notes, from the already existing database including a comparison with similar results. To begin with, this application is used to digitalize all the fake notes from all the existing cases and the upcoming cases. Furthermore, this application is used to find the top matching results of a new Counterfeit note which has been caught in a new case. Since all the fake note details are already stored in the database, the new note is compared to find the exact matches, or similar matches if an exact match is not available. On top of that this application also provides the function to compare any two Currency Notes to find the similarities between them. This application is made offline as it is a private government project. The application is only accessible by the members of the Forensic Department. This application works by extracting the features of the currency note like Security Thread and 25 extra features to improve accuracy. The features are in turn stored in the database separately. When a new note is inputted, the features of the note are extracted and compared separately with each feature, and the results are returned.

திட்டப்பணிச்சுருக்கம்

இந்த திட்டத்தின் முக்கிய நோக்கம், ஏற்கனவே இருக்கும் தரவுத்தளத்தில் இருந்து போலி நாணயத் தாள்களின் மூலத்தைக் கண்டறிவதே ஆகும் . இந்தப் பயன்பாடு தொடங்குவதற்கு, தற்போதுள்ள அனைத்து வழக்குகள் மற்றும் வரவிருக்கும் வழக்குகளில் இருந்து அனைத்து போலி குறிப்புகளையும் டிஜிட்டல் மயமாக்க பயன்படுகிறது. மேலும், புதிய வழக்கில் சிக்கிய புதிய கள்ள நோட்டின் சிறந்த பொருத்த முடிவுகளைக் கண்டறிய இந்தப் பயன்பாடு பயன்படுத்தப்படுகிறது . அனைத்து போலி குறிப்பு விவரங்களும் ஏற்கனவே தரவுத்தளத்தில் சேமிக்கப்பட்டிருப்பதால் , புதிய குறிப்பு சரியான பொருத்தங்கள் அல்லது சரியான பொருத்தம் கிடைக்கவில்லை என்றால் ஒத்த பொருத்தங்களைக் கண்டறிய ஒப்பிடப்படுகிறது. அதற்கு மேல் இந்தப் பயன்பாடு இரண்டு நாணயத் தாள்களை ஒப்பிட்டு அவற்றுக்கிடையே உள்ள ஒற்றுமைகளைக் கண்டறியும் செயல்பாட்டையும் வழங்குகிறது . இது ஒரு தனியார் அரசாங்கத் திட்டம் என்பதால் இந்த விண்ணப்பம் ஆஃப்லைனில் செய்யப்படுகிறது . விண்ணப்பத்தை தடயவியல் துறை உறுப்பினர்கள் மட்டுமே அணுக முடியும். செக்யூரிட்டி த்ரெட் போன்ற கரன்சி நோட்டின் அம்சங்களையும் துல்லியத்தை மேம்படுத்த **25** கூடுதல் அம்சங்களையும் பிரித்தெடுப்பதன் மூலம் இந்தப் பயன்பாடு செயல்படுகிறது . அம்சங்கள் தரவுத்தளத்தில் தனித்தனியாக சேமிக்கப்படும் . ஒரு புதிய நோட்டை உள்ளிடும்போது, குறிப்பின் அம்சங்கள் பிரித்தெடுக்கப்பட்டு , ஒவ்வொரு அம்சங்களுடனும் தனித்தனியாக ஒப்பிடப்பட்டு முடிவுகள் வழங்கப்படும்.

ACKNOWLEDGMENT

First and foremost, I would like to thank **Dr. L. JONES TARCIUS DOSS**, Professor and Head, Department of Mathematics, Anna University, Chennai for his constant motivation throughout the course.

I am most grateful to **Dr. G. SUMATHI**, Teaching Fellow, Department of Mathematics, Anna University, Chennai, for her inspirational words and guidance throughout this project. I also thank my faculty advisor **Dr.V.VIJAYALAKSHMI,** Assistant Professor, Department of Mathematics, Anna University, Chennai**,** for her moral support.

I would like to extend my sincere thanks to the review committee members **Dr. Y. SARADA**, Professor, **Dr. S. SADIQBASHA**, Associate Professor, **Mr. K. KAMALA KANNAN**, Project Coordinator, Teaching fellow, Department of Mathematics, Anna University, Chennai, for their suggestions, improvement and constructive criticism during this project.

I am indebted to **Dr. B. BABU MADHAVAN**, Company Director, Sakura Geoinformatics Software Research Company, Chennai, **Mr. LINGESHWARAN**, Managing Director, Sakura Geoinformatics Software Research Company, Chennai, **Ms. K. SAUNDHARYA**, Business Analyst, Sakura Geoinformatics Software Research Company, Chennai, for his guidance. I am also grateful to my parents, friends, and colleagues for their timely support and constant words of encouragement. Last but not the least, I extend my thanks to the Almighty.

Chennai-25

January 2022 **(AHMED FAYAZ W)**

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	Application Programming Interface
CPU	Central Processing Unit
DB	Database
DFD	Data Flow Diagram
DIP	Digital Image Processing
GSR	Geoinfomatics Software Research
IDE	Integrated Development Environment
OpenCV	Open Source Computer Vision Library
SAR	Synthetic Aperture Radar
UI	User Interface
UML	Unified Modeling Language

CHAPTER 1

INTRODUCTION

This chapter commences with a brief introduction about the organization in which the project was carried out and the services provided by the company. It also contains an overview of the project.

1.1 ORGANIZATION PROFILE

Sakura GSR is a Geoinformatics Software Development Company mainly focused on Research, Applications, Image Processing, SAR Data, and Remote Sensing Data. Sakura provides services such as DIP Research, 3D Modeling Research, Mapping using Aerial Vehicles. Sakura also provides services for Private Organizations. This Project is Confidential, about the Currency Note. Thus Codes and other Confidential details about this Project are not shared much.

1.2 OBJECTIVE OF THE PROJECT

The project aims to find the source of fake Indian currency notes to eradicate the production at the initial stage. To design the application in a user-friendly way. As the application takes counterfeit currency notes as inputs to perform certain functions that provide the source of origin of the given counterfeit notes. Provide services to extract serial numbers and signature features of the currency notes and inputs the extracted data into a database. This application should provide an easier and faster approach to identify the source of origin with the help of the data present in the database.

1.3 PROJECT OVERVIEW

The main objective of this project is to find the Source of the Counterfeit Currency Notes, from the already existing database including a comparison with similar results. To begin with, this application is used to digitize all the fake notes from all the existing cases and the upcoming cases.

Furthermore, this application is used to find the top matching results of a new Counterfeit note which has been caught in a new case. Since all the fake note details are already stored in the database, the new note is compared to find the exact matches, or similar matches if an exact match is not available. On top of that this application also provides the function to compare any two Currency Notes to find the similarities between them. This application is made offline as it is a private government project.

The application is only accessible by the members of the Forensic Department. This application works by extracting the features of the currency note like Security Thread and 25 extra features to improve accuracy. The features are in turn stored in the database separately. When a new note is input, the features of the note are extracted and compared separately with each feature, and the results are returned.

To implement this into an application, version 5 of PyQt is used in the front-end part, Python in the back-end part, and MySQL as Database. This application is divided into several modules which will be as follows:

- Login
  - ✓ End-users can login using the given account.
- Client Details
  - ✓ Enter Client Details.
  - ✓ Stores Client Details.
- Available Platforms
  - ✓ Only on Windows, Only in one computer.
- Extract Serial Number
  - ✓ Extract Serial Number from Currency Note.
- Update Counterfeit Currency Note
  - ✓ Extract features and update the details in Database
- Extract Top Match
  - ✓ Compare and return top matching result.
- Compare Notes
  - ✓ Compare two notes and return similarity percentage.

1.4 ORGANIZATION OF THE REPORT

Chapter 1: This chapter describes a profile of Sakura where the project has been done. Also, it gives a brief overview of the project.

Chapter 2: This chapter gives details about the system that was expected to be developed. It provides the system requirements and explains in detail the various tools and technologies involved in the development of the system.

Chapter 3 : This chapter discusses the design phase of the system.

Chapter 4: This chapter discusses the implementation of the system and explains about the project in detail. It also explains the testing done with the developed test cases.

Chapter 5: This chapter summarizes the whole project and points out the assets and limitations of the project and also includes the areas of future enhancements

CHAPTER 2

SYSTEM ANALYSIS

System Analysis is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. It is conducted to study a system or its parts to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

2.1 EXISTING SYSTEM

In the existing system, the work was done by human labor. The employees would be performing tasks manually, comparing notes in-person, and coming up with a top match.

2.2 PROPOSED SYSTEM

To overcome the disadvantages of the existing system, a new system was proposed that bridges the gap. The project consists of 4 modules. To begin with, the serial number of the note should be extracted. Furthermore, the features of the notes are detected. Last but not least the features should be searched and return a top match, and compare features between two notes. All these details are stored in the database.

2.3 PROJECT INITIATION

The application was split into corresponding modules and the functionality for each module was finalized. Later the modules were prioritized and based on the priority front-end development was started followed by back-end development.

2.4 SYSTEM REQUIREMENTS

System requirements are the required specifications a device must have in order to use certain hardware or software.

2.4.1 Hardware Specifications

Table 2.1 – Hardware Specifications

Processor	Intel Xeon
System Type	64-bit Operating System
Processor Speed	2.20 GHz
Random Access Memory	16 GB
Solid State Drive	1 TB

Table 2.1 discusses the system resource settings allotted for the specific device i.e. lists the hardware specification as part of the configuration of the system used to develop the tool. These hardware specs were already installed in the system and are much powerful than what was needed.

2.4.2 Software Specifications

Table 2.2 - Software Specifications

Operating System	Windows 10
Front-end	PyQt5
Back-end	Python
Database	MySQL
IDE	Pycharm Community Edition

Table 2.2 lists the software specification i.e. the software side of the configuration of the system used to develop the application. The operating system, languages used, databases used and IDEs used come under software specification.

2.5 TOOLS AND TECHNOLOGIES USED

Counterfeit Currency Note Sourcing System is built using the following software technologies. This section gives an overview of these tools and technologies.

2.5.1 Python

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

The entire back-end development was carried out using the Python language. The functionalities involved in the application also been developed using the available packages in python.

2.5.2 Python – OpenCV

OpenCV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free for use under the open-source Apache 2 License. Starting with 2011, OpenCV features GPU acceleration for real-time operations.

OpenCV is a library for python which specializes in Digital Image Processing. This library is the most used for all of the image manipulations of the Currency Note.

2.5.3 PyQt

PyQt is a Python binding of the cross-platform GUI toolkit Qt, implemented as a Python plug-in. PyQt is free software developed by the British firm Riverbank Computing. It is available under similar terms to Qt versions older than 4.5; this means a variety of licenses including GNU General Public License (GPL) and commercial license, but not the GNU Lesser General Public License (LGPL). PyQt supports Microsoft Windows as well as various flavours of UNIX, including Linux.

Since this is a private project, the application is made for a single user. Thus version 5 of PyQt is used for creating desktop GUI.

2.5.4 MySQL

MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database.

MySQL is currently the most popular database management system software used for managing the relational database. It is an open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with Python for creating powerful and dynamic server-side or web-based enterprise applications.

It is developed, marketed, and supported by MySQL AB, a Swedish company, and written in C programming language and C++ programming language.

2.5.5 Pycharm Community Edition

Pycharm is an IDE used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with Version Control Systems (VCSes), and supports web development with Django as well as data science with Anaconda.

This IDE is used as it is most Comfortable with python. Since all the technologies are libraries of python, this IDE is the most suited for this project.

CHAPTER 3

SYSTEM DESIGN

System Design is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy specific requirements. Before planning, you need to understand the old system thoroughly and determine how computers can best be used to operate efficiently. System Design focuses on how to accomplish the objective of the system.

3.1 SYSTEM ARCHITECTURE

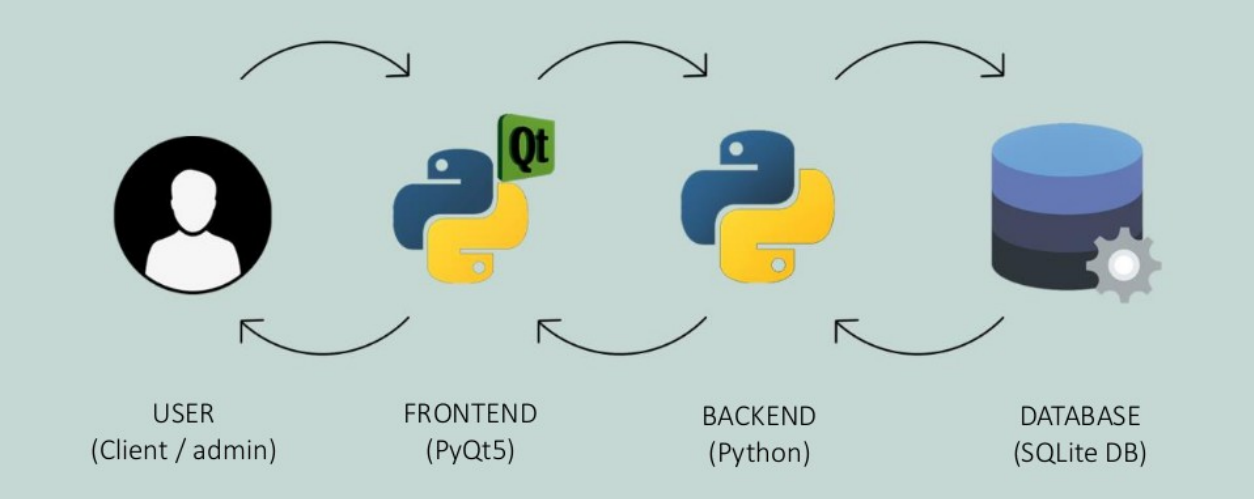

Figure 3.1 – System Architecture

Figure 3.1 shows the system architecture of the application which consists of a platform, GUI, database. The database is accessed using queries and the response from the database are returned.

3.2 UML DIAGRAMS

A Unified Modeling Language diagram is a diagram based on the UML to visually represent a system along with its main actors, roles, actions, artifacts or classes, to better understand, alter, maintain, or document information about the system.

3.2.1 Use Case Diagram

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions that some system or systems should or can perform in collaboration with one or more external users of the system.

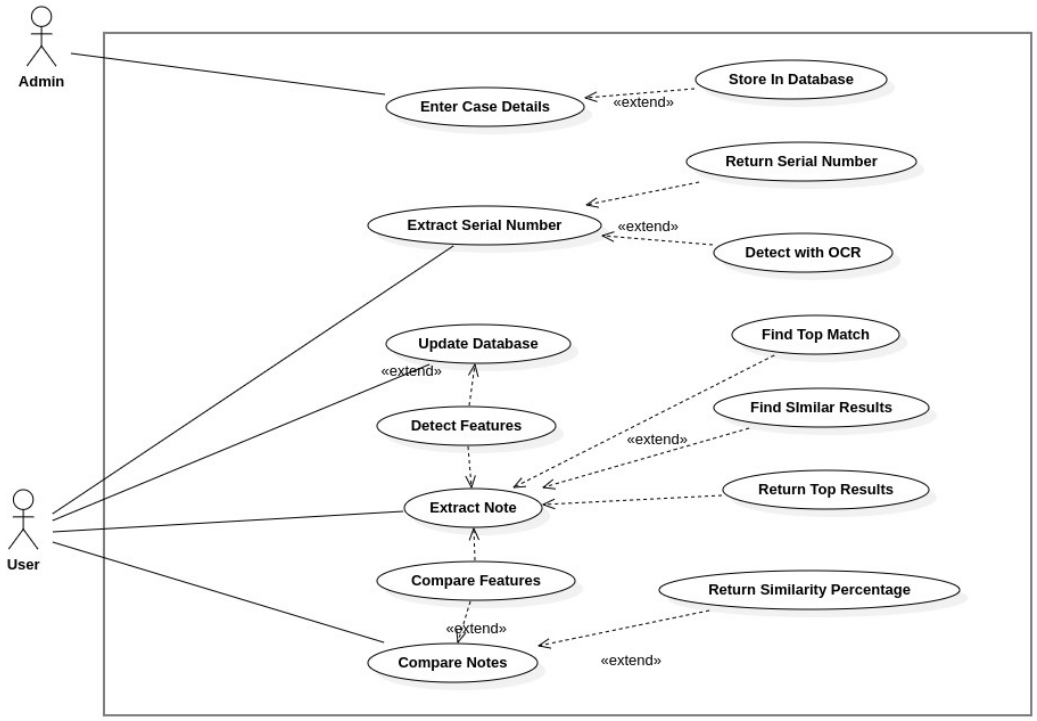

Figure 3.2 – Use Case Diagram

Figure 3.2 shows the Use case diagram of Desktop Application that includes an actor – Admin and actions performed by the actors are Enter case details, extract serial number, update database, extract note and compare notes.

3.2.2 Class Diagram

A class diagram describes the types of objects in the system and the different types of relationships that exist among them.

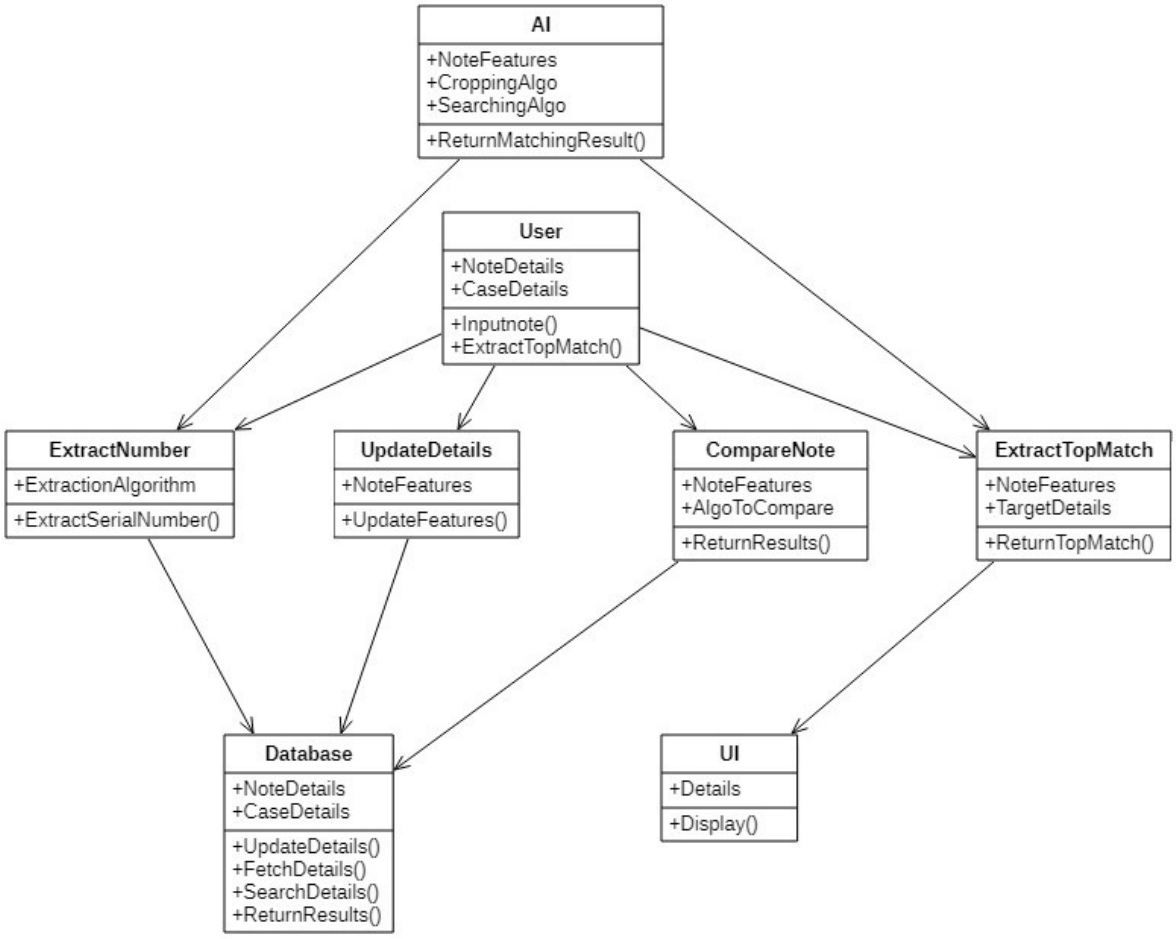

Figure 3.3 – Class Diagram

Figure 3.3 shows the class diagram for Desktop Application which contains Eight classes – AI, User, Extract Number, Update Details, Compare Notes, Extract Top Match, Database, UI. The class contains attributes and operations to be performed by the class.

3.2.3 Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system.

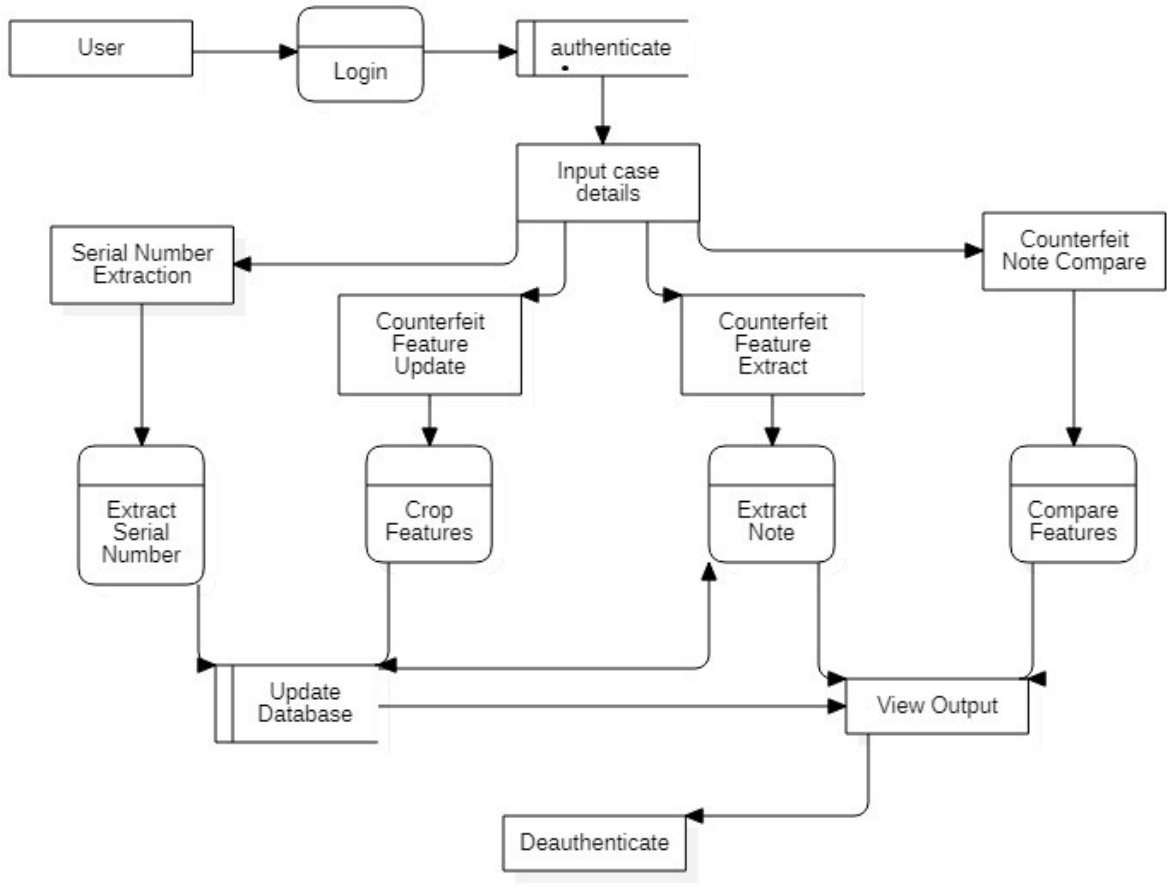

Figure 3.4 – Data Flow Diagram

Figure 3.4 shows the data flow diagram for Counterfeit Currency Note Sourcing System which involves four entities – Extract Number, Update Details, Compare Notes, Extract Top Match, describes the data flow throughout the system

CHAPTER 4

SYSTEM IMPLEMENTATION AND TESTING

System Implementation is the process of converting the manual or old computerized system with the newly developed system & making it operational, without disturbing the functioning of the organization. System Implementation comprises creating computer-compatible files, Training the people who are going to operate the system, Installing the necessary hardware, terminals & network, Installing the proposed software, testing. The main purpose of this implementation process is to design and create a system element conforming to that element's design properties and requirements. The element is constructed employing appropriate technologies and industry practices. This process bridges the system definition processes and the integration process.

4.1 SYSTEM MODULES

A module is a software component or part of a program that contains one or more routines. One or more independently developed modules make up a program. The project involves the following modules:

- Login
- Client Details
- Serial Number Extraction
- Update Counterfeit Note
- Extract Counterfeit Note
- Compare Counterfeit Note

4.2 MODULES

4.2.1 Login Module

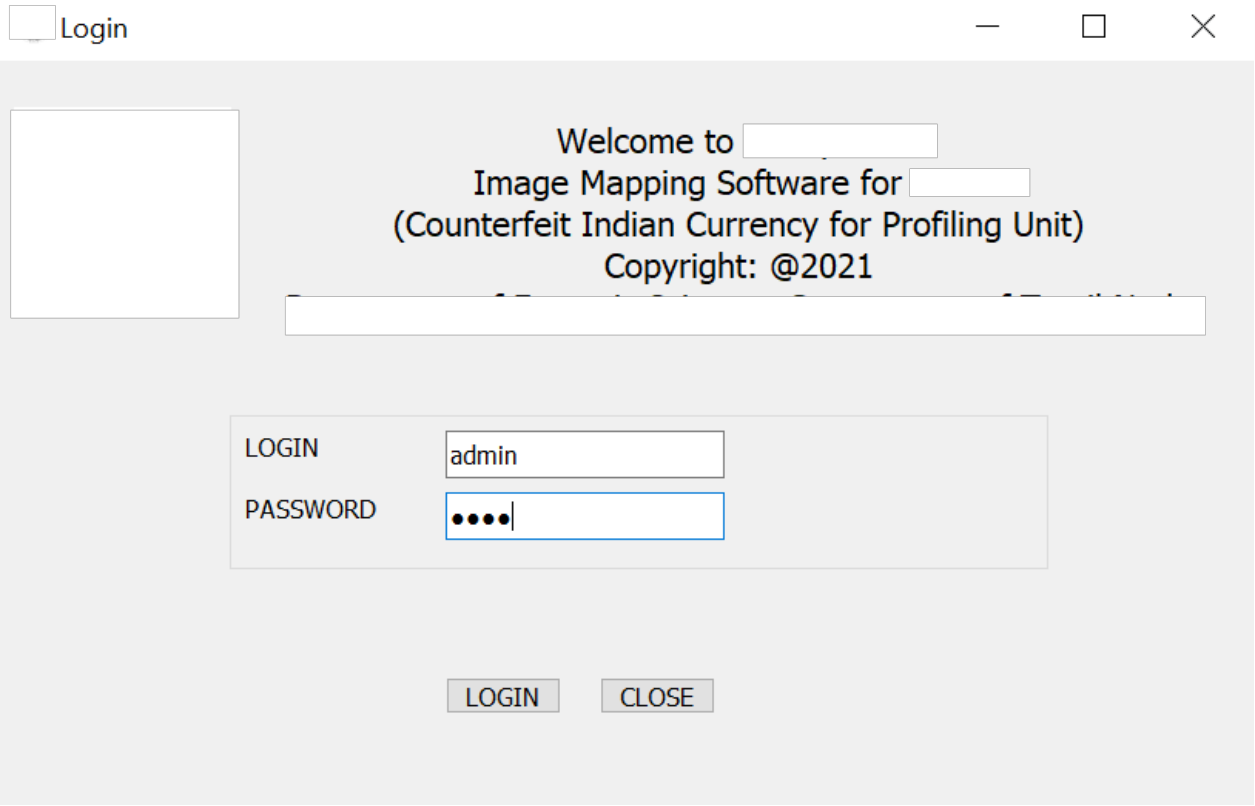

Figure 4.1 – Login Module

The Login module is developed to login into the registered account. As this application is designed for only one user, A new User cannot be created.

4.2.2 Client Details Module

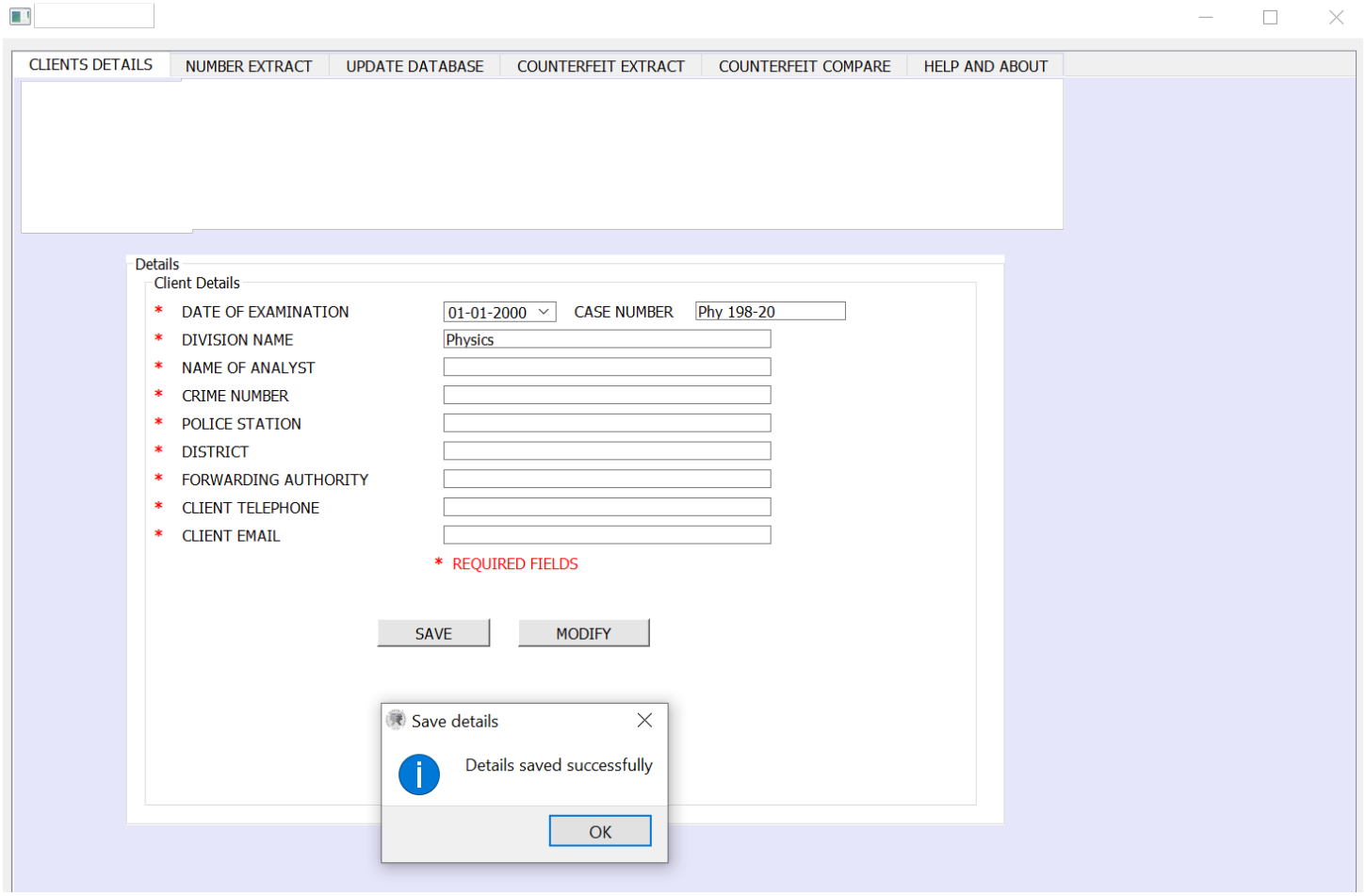

Figure 4.2 – Client Details Module

The details of the currency notes such as date, Division Name, Crime number, Police Station, District, Client details are all stored here. The details are stored in the database for future use.

4.2.3 Serial Number Extraction Module

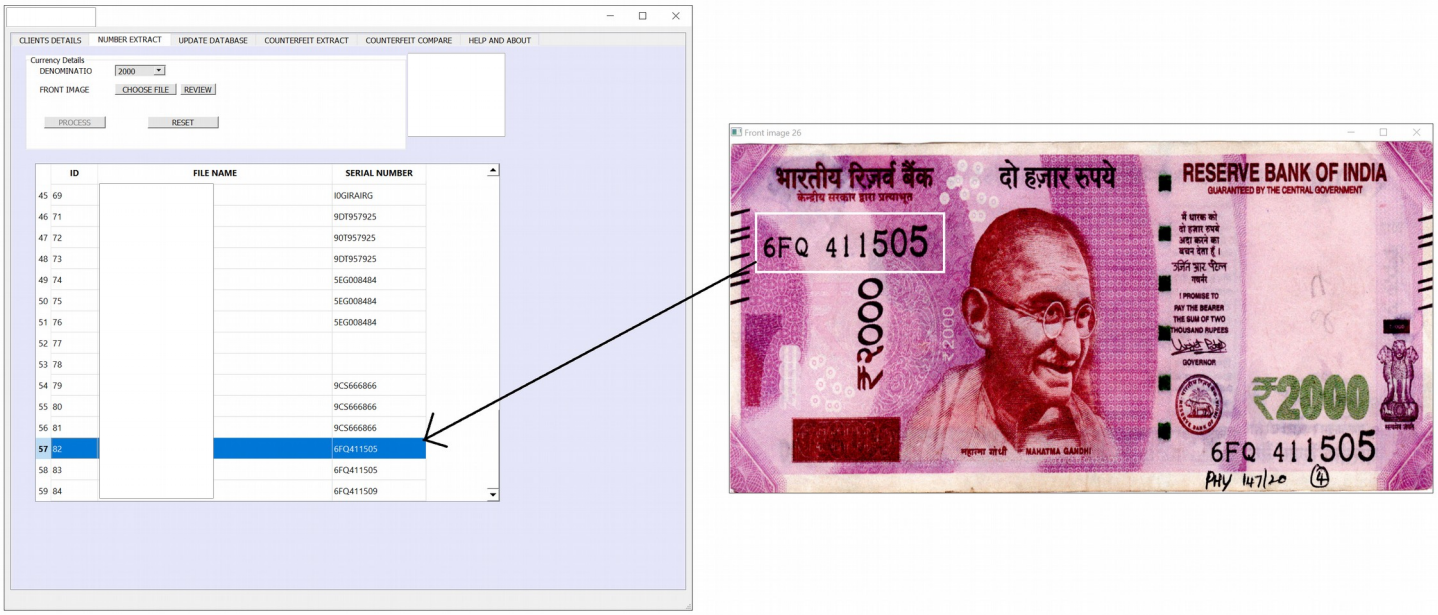

Figure 4.3 – Serial Number Extraction Module

The Serial Number of Currency notes are extracted in this module. The case details have been blurred as the project is confidential. Here, the note is fed as input to this module. Using the algorithm we have created, the AI automatically recognizes the Serial Number and, using an Algorithm called Easy-OCR, the serial number is detected and stored in the database with the note details.

4.2.4 Update Counterfeit Note Module

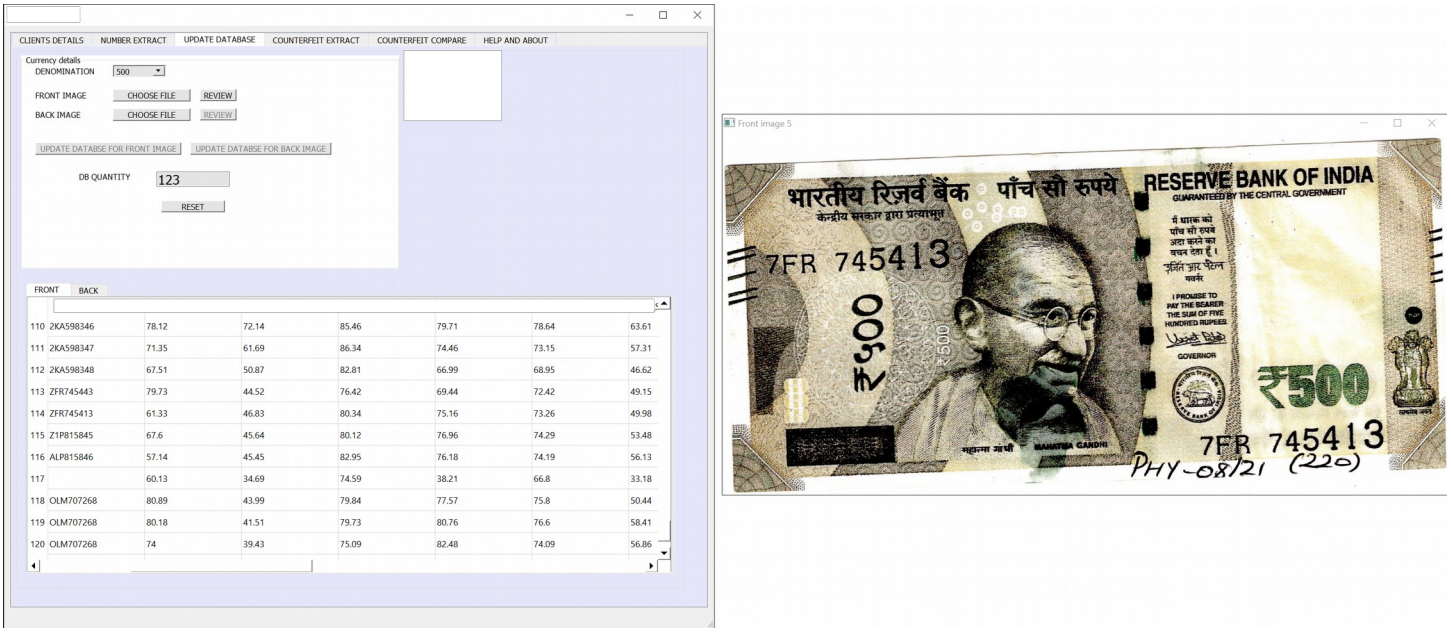

Figure 4.4 – Update Counterfeit Note Module

In this module, the counterfeit note is fed as input. There are 16 and 11 features in the front and back of the note respectively. The features are blurred as it is confidential. The algorithm searches for all these features and stores them in the database in numeral form. In the future, these numbers are said to be replaced by Image Database directly.

4.2.5 Extract Counterfeit Note Module

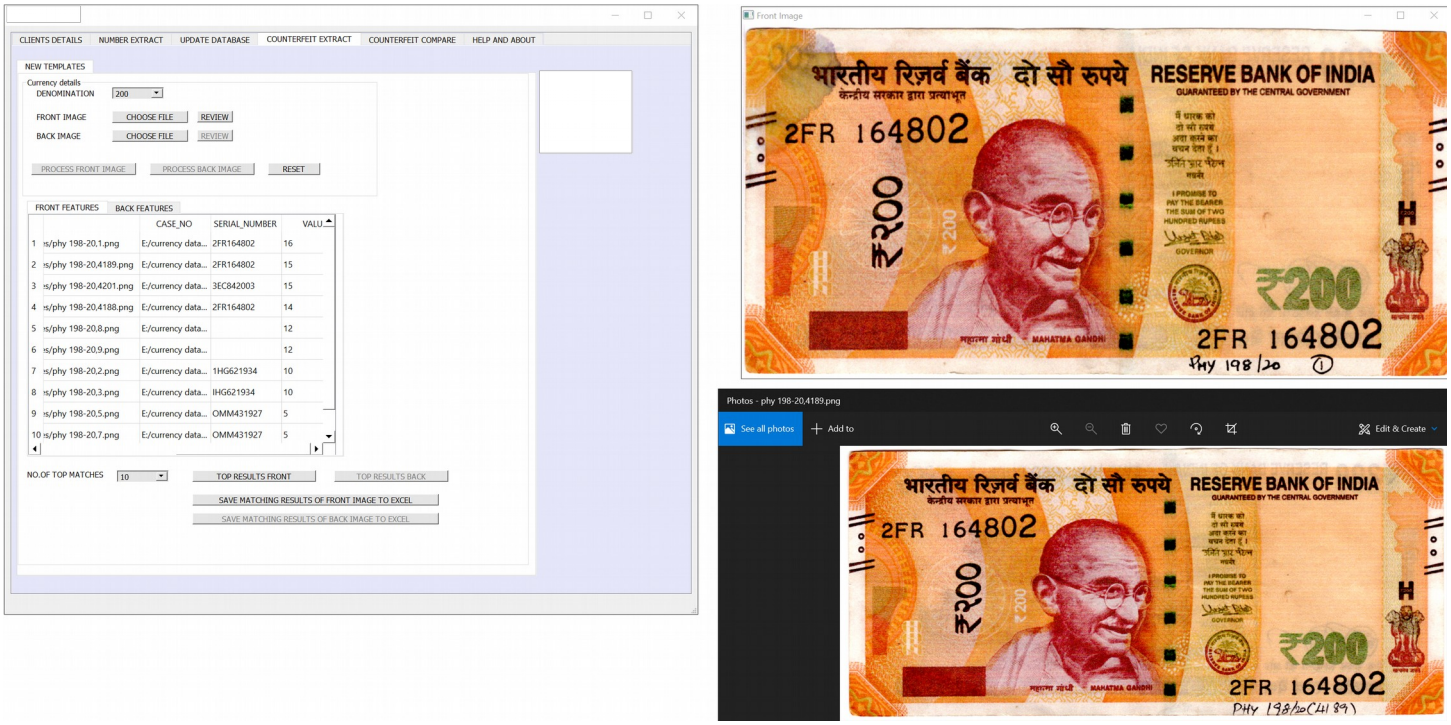

Figure 4.5 – Extract Counterfeit Note Module

The desired top match of a note required is fed as input to the module. Here the first note at the top is fed as input. The algorithm searches for the features in the note which is fed as input and recognizes it as a source. Using this source it searches with the already existing DB and tries to return a top match or a similar match if the top match is not found. Here the second note is the top match found as the details are listed in Figure 4.5.

4.2.6 Compare Counterfeit Note Module

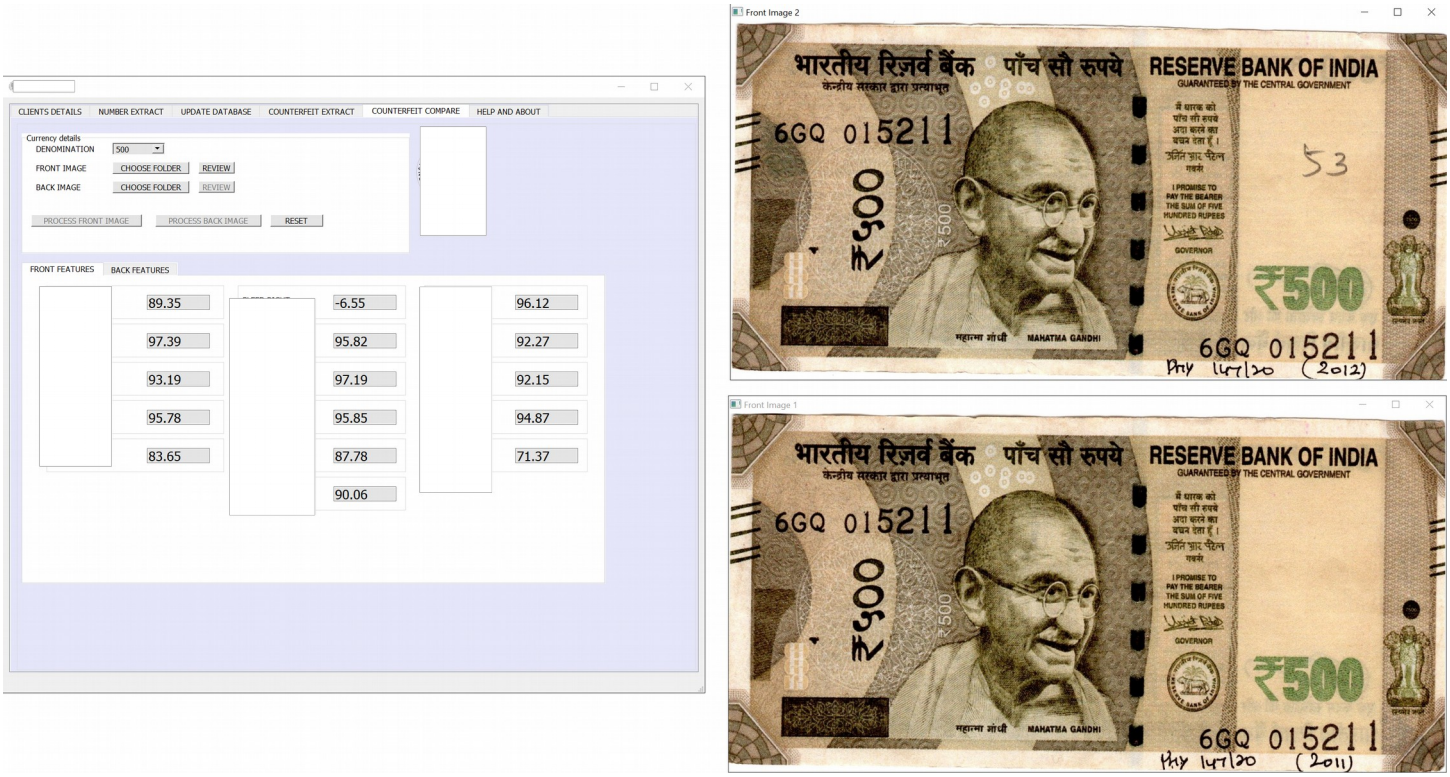

Figure 4.6 – Compare Counterfeit Note Module

This is a unique module as all the previous modules are linked. Here there are two notes fed as inputs. The algorithm identifies all the features in these two notes and then compares them. Then the application results in the similarity percentage between them as shown in Figure 4.6.

4.3 TESTING

Software testing is a process of identifying the correctness of software by considering its all attributes (Reliability, Portability, Re-usability, Usability) and evaluating the execution of software components to find the software bugs or errors, or defects.

Software testing as a whole considers two things – validation and verification. Validation is a dynamic mechanism of validating and testing the actual product. Verification is a static practice of verifying documents, design, code, and programs. The following are the testing techniques used in the OTT Web Application.

4.3.1 Unit Testing

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with the functional correctness of the standalone modules.

Each module in this project is unit tested to verify and validate its functionality like testing the login page with valid and invalid credentials, registering end-users with valid and invalid details, etc.,

4.3.2 Integration Testing

Integration testing is the second level of the software testing process that comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

Once all the components or modules are working independently, then we need to check the data flow between the dependent modules.

After integrating each unit tested module with the other modules in the project valid functionality testing is done. Malfunctioning of the application after integrating each unit was identified and corrected before proceeding to the other unit.

4.3.3 System Testing

System Testing includes testing of a fully integrated software system. Generally, a computer system is made with the integration of software (any software is only a single element of a computer system). The software is developed in units and then interfaced with other software and hardware to create a complete computer system.

After integration testing is done, (i.e. after testing all the modules are working properly with the specified functionalities), system testing is done for the overall application. The bugs and defects raised during system testing have also been fixed and the overall system is working properly based upon the specified requirements.

4.3.4 UI Automation Testing

UI automation testing is a technique where these testing processes are performed using an automation tool. Instead of having testers click through the application to verify data and action flows visually, test scripts are written for each test case. A series of steps to follow when the verifying data is then added.

PyUnit unit testing framework, dubbed 'PyUnit' by convention, is a Python language version of JUnit. JUnit was written by smart cookies Kent Beck and Erich Gamma, and is, in turn, a Java version of Kent's Smalltalk testing framework.

Each is the de facto standard unit testing framework for its respective language, and therefore both are a strong basis for an effective and elegant Python framework.

4.4 TEST CASES

A test case is a document, which has a set of test data, preconditions, expected results, and postconditions, developed for a particular test scenario to verify compliance against a specific requirement. A test scenario measuring functionality across a set of actions or conditions to verify the expected result. They apply to any software application, can use manual testing or an automated test, and can make use of test case management tools. Test cases can measure many different aspects of code. The steps involved may also be intended to induce a Fail result as opposed to a positive expected result such as when a user inputs the wrong password on a login screen. Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some endpoint or also known as execution postcondition.

Below are few major reasons to write test cases:

- Test cases, if written as per industry standard, can be helpful to different teams to gain knowledge about application.
- Test cases can be used by automation engineers for their reference purpose to automate application behavior.
- Test cases are proof of concept for 'What's tested and What's not tested' during application release cycle.
- The purpose of a test case is to determine if different features within a system are performing as expected and to confirm that the system satisfies all related standards, guidelines and customer requirements.

- The process of writing a test case can also help reveal errors or defects within the system.

Table 4.1 – Test Cases

Module	Test Case Summary	Expected Result	Actual Result	Status
Login	Enter The Valid Credentials	Registration Successful	Registration Successful	Pass
	Enter Invalid Credentials	Registration Failed	Registration Failed	Pass
Enter Case Details	Enter The Case Details	Update Database Successfully	Update Database Successfully	Pass
Detect Serial Number	Upload Counterfeit Currency Note, Serial Number of the note should be detected.	Detect Serial number and return successfully.	Serial number detected and return successfully.	Pass
Update Database	Upload Currency Note with denomination, Features should be detected and stored in the database successfully.	Detect Features and store the data in Database successfully	Features detected and stores successfully	Pass
	Upload note with Different Denomination.	Detect and return “Different Denomination”	Detected different denomination and returned “Different Denomination” Successfully	Pass

Table 4.1 – Test Cases(Continued)

Module	Test Case Summary	Expected Result	Actual Result	Status
Extract Note	Upload Currency Note with Denomination, Return Top Matches.	Detect Features, Compare it with Database, return Exact Match and Similar Results	Features Detected, and returned Exact Match and Similar Results.	Pass
Extract Note	Upload Currency Note with different Denomination.	Detect and return “Different Denomination”	Detected different denomination and returned “Different Denomination” Successfully	Pass
Compare Note	Upload 2 Currency Note, Return the Similarity percentage of both the notes.	Detect, compare, return the similarity Percentage of Notes.	Detected, compared, returned the similarity Percentage of Notes.	Pass
	Upload 2 Currency Note with different Denomination.	Detect and return “Different Denomination”	Detected different denomination and returned “Different Denomination” Successfully	Pass

CHAPTER 5

CONCLUSION

This chapter encapsulates the overall view of the project and the areas of future enhancements.

Counterfeit Currency Note Sourcing System is primarily developed to decrease human labor. The front-end part of the application is built using PyQt5 using Python as the programming language. The back-end part of the application is developed using Python, MySQL is used to store data, access, and update the data.

5.1 LIMITATIONS

Below are some of the limitations for Counterfeit Currency Note Sourcing System:

- The Serial Number Extraction sometimes detect unknown characters as the scanned copy may not be clear enough. It is then corrected by the User itself
- Some features are not detected due to fading of notes.

5.2 FUTURE ENHANCEMENTS

- Numeral data to be enhanced directly to Image data. As numeral data is not accurate enough.
- Detect Unique features in a Counterfeit Note for even more accuracy.
- Extracting Serial Numbers at 100% accuracy.
- Implement Machine learning for detecting unique features in Counterfeit Note.

REFERENCES

1. Martin C. Brown, “Python: The Complete Reference”, McGraw Hill Education, 2018.
2. Sandipan Dey, “Hands-On Image Processing with Python”, Packt, 2018.
3. Rafael C. Gonza Lez & Richard E. Woods, “Digital Image Processing”, Pearson, 2018.
4. Dr. Panchanand Jha, “OpenCV with Python”, Notion Press, 2020.
5. Vinicius M. Grippa, “Learning MySQL”, O’Reilly, 2021.
6. Mark Summerfield, “Rapid GUI Programming with Python and Qt”, Pearson, 2015.
7. Sommerville, I., “Software Engineering”, Pearson, 2011.
8. Abraham Silberschatz, Henry F. Korth, “Database System Concepts”, McGraw Hill, 2011.
9. Rafael C.Gonzalez and Richard E.Woods, “Digital Image Processing”, Pearson, 2003.