

Enhancing User Experience in Prize Bond Lotteries through Intelligent Notification Systems

by

Ahnaf shahriar Dip

24141152

Shafat Bin Nizam Nibir

22241045

Dewan Golam Mortoza

23141095

MD. Rajwan Ahmed Zehad

24341114

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2024

© 2024. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Ahnaf Shahriar Dip

24141152

Shafat Bin Nizam Nibir

22241045

Dewan Golam Mortoza

23141095

MD. Rajwan Ahmed Zehad

24341114

Approval

The project titled “Enhancing User Experience in Prize Bond Lotteries through Intelligent Notification Systems” submitted by

1. Ahnaf Shahriar Dip (24141152)
2. Shafat Bin Nizam Nibir (22241045)
3. Dewan Golam Mortoza (23141095)
4. MD. Rajwan Ahmed Zehad (24341114)

Of Summer, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 20, 2024.

Examining Committee:

Supervisor:

Mr. Md. Imran Bin Azad

Senior Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:

Sadia Hamid Kazi, PhD

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

This thesis proposes the design and implementation of a modern, user-friendly system for checking prize bond results. It aims to revolutionize the process of checking prize bond results by implementing OCR(Optical Character Recognition) by using pytesseract and PyMuPdf library. The primary objective is to simplify the process of verifying whether a user's prize bond number has won or not .The system will leverage image processing techniques to automatically extract and read the prize bond number from the scanned image and pdf. Later it store the extracted prize bond number and matches with the result database and give us the final winner result.This way we find the winner.Finally the system send notification automatically to the winner if the user win or not .The proposed system offers a seamless and efficient experience for users, allowing them to verify their prize bond status through a simple user friendly system.

Keywords:Optical Character Recognition (OCR); image processing; automation; event-driven programming; range entry; email notifications; in-app notifications; system architecture; user interface design; result verification; PyTesseract; PyMuPDF; text extraction; matching algorithm; database integration; role-based access control.

Acknowledgement

First and foremost, all praise and gratitude belong to Almighty Allah. whose grace and mercy have guided us throughout this journey and enabled the successful completion of our thesis without any significant problems.

We would also like to appreciate our supervisor, Mr. Md. Imran Bin Azad Sir. His helpful advice, intelligent input, and ongoing support have helped us a lot. His patience and willingness to assist whenever we encountered difficulties were tremendously appreciated and we are truly grateful.

Finally, we express our heartfelt gratitude to our beloved parents. Throughout this academic journey, we have been strengthened by their unwavering support, encouragement, and prayers. We are here today as a result of their unending love and sacrifices. We owe them everything.

Table of Contents

Declaration	i
Approval	ii
Abstract	iii
Acknowledgment	iv
Table of Contents	v
List of Figures	vii
1 Introduction	1
1.1 Introduction	1
1.2 Background Information	2
1.3 System Functionality	3
2 Related Work	4
2.1 Related Works	4
2.2 Comparison between related works	5
2.3 Advantages and Disadvantages	5
3 System Development Life Cycle (SDLC)	7
3.1 Requirements	7
3.1.1 Functional Requirements	7
3.1.2 Non-functional Requirements	7
3.2 Design	8
3.2.1 System Architecture	8
3.2.2 User Experience (UX) FLOW	8
3.2.3 Data Storage	8
3.3 Implementation	9
3.3.1 Key Components	9
3.3.2 Notification System	9
3.4 Testing	9
3.4.1 Unit Testing	9
3.4.2 Integration Testing	10
3.4.3 User Acceptance Testing (UAT)	10
3.5 Deployment	10
3.5.1 Environment Setup	10
3.5.2 Continuous Integration/Continuous Deployment (CI/CD) . . .	10

3.6	Maintenance	10
3.6.1	System Monitoring	10
3.6.2	Future Enhancements	11
4	Project Architecture	12
4.1	Frontend Architecture (User Experience)	12
4.1.1	Overview of the User Interface (UI)	12
4.1.2	Flow of User Experience (UX)	12
4.1.3	Tools and Technologies	15
4.1.4	Authentication and Security	15
4.1.5	User input and accessibility	15
4.2	Backend Architecture	16
4.2.1	1. Overview of the Backend	16
4.2.2	2. System Components and Workflow	16
4.2.3	3. Security Points	18
4.2.4	4. Performance Optimization and Scalability	19
4.2.5	5. Logging and Error Handling	19
4.2.6	6. Deployment and DevOps	19
4.3	Use Case and ER Diagram	20
5	Methodology	22
5.1	Overview of Methodology	22
5.2	PDF to Text Extraction	23
5.3	Image to text extraction	23
5.4	Manual Data entry	24
5.5	AJAX Integration and JSON Response	24
5.6	Result Matching, Data Storage, and Presentation	25
5.7	Notification system and user management	27
6	Preliminary Analysis and Visualization	30
6.1	Preliminary Analysis Flow chart	30
6.2	Visualization of project	31
7	Future Work	41
8	Summary	42
	Bibliography	43

List of Figures

4.1	Use Case Diagram	20
4.2	ER diagram	21
5.1	Data Storage and Class Diagram	26
5.2	Core process of the Methodology	29
6.1	Preliminary Analysis flow chart	30

Chapter 1

Introduction

1.1 Introduction

The financial industry has been one of the sectors that has benefited greatly from this technological revolution in the quickly changing world of technology. One example of this technological advancement is our research. It will deal with the venture of manually looking through physical prize bonds or prolonged lists of numbers to identify winners.

Traditionally, the process whereby winners are chosen during prize bond awards is rather laborious and, hence, subject to mistakes. Prize bond-hires have, by physical means, had to go through their respective prize bonds or, rather, a long list of numbers by hand. This process is time-consuming and invites human error. The normal processes differ from one another, and hence the smart use of technology in the process would open up an entire new vista to the perspective of the bondholders, minimize the time and reduce the chances of mistakes.

Keeping these affirmatives in sight, we are looking forward to building a system that automates the identification of winners for the prize bond draws. We aim to develop a technological solution that would save time and avoid mistakes, thereby improving the experience for bondholders in general thereby upending its time and manpower-intensive nature.

Optical Character Recognition is expected to emerge as a cornerstone of the model, allowing us to convert selected document formats into editable and searchable records. It will smooth the way out for prize bond verification, thereby speeding the process along. Automation of this would keep the gut-wrenching work of cross checking manually and assist in reliable winner determination.

Web development techniques form an integral part of our study, which will create an interactive interface for bondholders to store and retrieve their prize bond numbers. This feature not only allows searching for winning numbers but will also provide a timely alert to bondholders updating them about their bonds.

We intend to take mobile App development into our system and improve the bondholder's experience. Hence, users would be able to obtain their information through

their mobile devices, increasing convenience, and get prompt information regarding his/her prize bonds. The essence of our research is to place technological enhancement as a core of further improvements to the prize bond process. Digitizing checks aims to create a more automated and consumer-friendly environment for bondholders.

1.2 Background Information

In several nations, prize bonds have become a well-liked means of investing for people who want to make money while also having the chance to win monetary rewards. The method of selecting winners has traditionally depended on labor-intensive, prone to error manual checks against lengthy lists of bond numbers. This conventional method causes stress and uncertainty for bonds by taxing staff and delaying the announcement of winners.

Technology integration is now crucial for increasing operational effectiveness and boosting customer service as the financial sector develops. Technologies that have shown a great deal of promise for automating data extraction from physical documents include optical character recognition (OCR). By enabling the rapid and precise digitization of bond numbers from images or PDFs, this feature has the potential to drastically improve the prize bond verification process by cutting down on the amount of time needed for manual inspections.

In addition, the emergence of digital platforms has made room for interactive systems that may give consumers real-time updates. It is possible to design a more user-friendly interface for bondholders that makes it easier for them to obtain bond information and receive timely notifications about victories by combining web development and mobile app technologies. The process can be further streamlined by integrating these systems with official winner lists via APIs. This will guarantee that bondholders receive timely updates regarding their status without the delays that come with more conventional means.

Addressing data security concerns is crucial, in addition to improving efficiency and user experience. One of the main priorities should be protecting sensitive data, like bond numbers and personal information, as digital platforms become more and more important for financial operations. Strong security measures will assist maintain system integrity and foster a culture of trust.

In order to improve the efficiency, accuracy, and bondholder experience of the prize bond verification procedure, this research effort intends to investigate the possibilities of utilizing contemporary technology. Through the use of automated winner identification and improved digital channel communication, we can lower the risks involved in manual processing and greatly enhance bondholder experience overall.

1.3 System Functionality

The device will encompass three main additives:

Data Extraction: This portion will recall Optical character recognition (OCR) to extract prize bond numbers from photos or PDFs. The extracted numbers will then be saved in a database on our website.

Winner List Integration: The system can be tailored to combine with official winner lists posted by the issuing company. This integration may involve periodically extracting records from a delegated webpage or imposing an API (application Programming Interface) supplied by means of the organization.

Winner Notification: The extracted prize bond numbers in opposition to the winner list may be automatically in comparison by the system. For every identifying match, the system will cause a notification to the user's cellphone.

Benefits of this automation:

Efficiency: hugely reduces the effort and time required to test prize bonds for winnings.

Accuracy: eliminate the possibility of human blunders during manual checking.

Convenience: supply real-time notifications and signals without delay to customers' telephones.

Technical Considerations:

OCR Accuracy: The accuracy of the OCR engine is vastly vital for dependable prize bond number extraction.

Data Protection: To protect private consumer information, which includes prize bond numbers, the system needs to have robust safety features in vicinity.

Scalability: A widespread amount of data extraction and winner listing comparisons should be supported by way of the device's layout. In order to improve personal experience and expedite the prize claim technique, this research gives a unique method to prize bond winner detection by way of combining computer imaginative and prescient and data processing strategies.

Chapter 2

Related Work

2.1 Related Works

Previously many development teams worked with a similar idea of our research. All the works are mainly android mobile applications where users can get the service by identifying prize bond numbers and providing the result to the users on whether they won the prize or didn't. There are more additional features for different services for users related to prize bonds in different apps.

We found the mobile applications: Prizebond, Prize Bond Checker, Prizebond Wallet, and Prize Bond Monitor. The common relation among all these works is that they show the result of their prize bonds. We collected user reviews from the Google Play Store regarding these apps. Most of these apps had positive reviews based on the performance they promised to provide to the user.

Although many negative reviews were also noticed based on the loading time, hangs, lack of some features, and less clearly explained features. Prizebond and Prizebond checker app has login and signup features. By that user security is ensured and multiple users don't get interrupted. On the other hand, the Prizebond wallet and Prize bond monitor do not safeguard any security as they don't have this feature.

One major issue with the Prize Bond Monitor app is it is based in something other than Bangladesh. It is based in India. But there wasn't any information about this in the Play Store to identify as a User. All the apps have different kinds of additional features like FAQs, Privacy policies, contact details, about us, and other features that explain the basic info of prize bonds.

However, these prize bond applications weren't done with a proper development team and plan. The key perspective wasn't for any research purposes but mainly for business purposes.

2.2 Comparison between related works

The biggest difference between our work and their work is their work isn't research and they haven't come up with new ideas and more user-friendly features unlike us.

Here are some key parts of our work:

1. Identifying prize bond numbers through PDF files and storing them in the database using the PyMuPDF library.
2. Identifying prize bond numbers through image files and storing them in the database using the PyTesseract library.
3. Implementing an automatic notification system to inform users about the results of their prize bonds.
4. Providing a manual input feature for users to enter their prize bond numbers and store them in the database.
5. Sharing the official PDF file from Bangladesh Bank on the homepage, ensuring that users have access to an authentic and clear view of prize bonds.
6. Additional features like contacts, services, and about us are there to support our users to use our website properly.

Except for manually inputting numbers and additional features other parts are unique than others. We are giving the privilege to the users to have their account and make the process more easygoing for them to use our service. Due to individual accounts, there is no chance for overlaps. The whole website is built with proper planning and research. The admin team will update the website with more capability. If there are more requirements from the user reviews then the admin team will add more features as well. Those mobile applications don't have these privileges as they have many issues like high loading time, hangs, lack of some features.

2.3 Advantages and Disadvantages

Advantages:

1. Less time-consuming.
2. Suitable for Dysgraphia and nonliterate people due to having the option not to write.
3. No overlaps among the users.
4. No need to check. Automatic notification will come when the result comes out.

5. Getting authentic and direct information from the Bangladesh Bank through the pdf file at the home page.
6. High performance and secured database.
7. Option for contacting the admin team as contact details are already provided.

Disadvantages:

1. No direct connection with banks, so no solution regarding bank-related issues.
2. No proper result if the information provided by the users is incorrect.
3. No option for buying or selling any price bonds via this website.

Chapter 3

System Development Life Cycle (SDLC)

3.1 Requirements

3.1.1 Functional Requirements

- **Prize Bond Number Extraction:** The system should extract prize bond numbers from various online sources with the help of Optical Character Recognition (OCR) technologies(PyMuPDF and PyTesseract libraries).
- **Database Storage and Management:** The extracted numbers have to be saved in a secure database with a functionality of role-based access control for admin and user of the program.
- **Result Matching:** The system should be able to link up the numbers extracted with the official prize bond outcomes and update the data in parallel with time.
- **User Notifications:** Notify the users in the in-app and email regarding the winning status and updates.

3.1.2 Non-functional Requirements

- **Performance:**The system should provide high levels of performance and recognition while operating in real-time mode and performing OCR's.
- **Scalability:**The system must be capable of supporting higher levels of usage by users, as well as the verification of prize bond information.
- **Security:** Data protection by adopting role based access combined with encryption. Likewise for token based authentication of sensitive information.

3.2 Design

3.2.1 System Architecture

- **Frontend Architecture:** HTML, CSS and JavaScript shall be responsive and user-friendly. The features of this design include: a login panel bond input scan or manual advising
- **Application Layer:** This layer deals with reception of user requests and responses, as well as user interactions.
- **Data Layer:** Closely associated with increasing the degree of audience knowledge about the product. The application contains added entries for prize bonds, data for collecting necessary data, improved information about users, notices as well as logs.
- **Service Layer:** APIs downloaded lists of prize bond numbers along with notification.

3.2.2 User Experience (UX) Flow

- **Home Page:** To log in or to register and to provide information about the prize bonds.
- **Dashboard:** A feature that displays the entered prize bonds by the user and shows an output of that and the output history.
- **Result Notification:** Whenever there is an alteration in the standing of the prize bond; customers are always advised.
- **Service Layer:** APIs downloaded lists of prize bond numbers along with notification.

3.2.3 Data Storage

- **Relational Database:** MySQL will be used for upcoming user profiles, prize bond numbers, tables, and notifications of results. Primarily, Foreign Key constraints will ensure the relational integrity in the table.

3.3 Implementation

3.3.1 Key Components

OCR for Text Extraction

- **PDF Extraction:** This designed web application successfully incorporates PyMuPDF for reading and processing of uploaded PDFs while paying attention to the extraction of prize bond numbers only.
- **Image Extraction:** Concerning extraction of text information within images, the program relies on PyTesseract.

Data Validation:

- Checks data entered against prize bond numbers by using regex format and also limits entries that are out of bounds.

User Management:

- Consists of Admin and user roles and allows administrative as well as users' access to data with limited privileges.

3.3.2 Notification System

- **Email Notification:** Django email functionality will be used to notify users on general update and results of the prize bond.
- **Dashboard Notifications:** By using WebSocket or AJAX poll client updated information about the devices will be delivered in real time.

3.4 Testing

3.4.1 Unit Testing

- **OCR Testing:** Accuracy of OCR functionality in various document types at 90'
- **Database Testing:** CRUD operation for the User and Result Tables and integrity test for Test Data to be written with Primary and Foreign keys.

3.4.2 Integration Testing

- **OCR and Database:** The data extracted is correctly stored and fetched from the database.
- **Result Matching and Notification:** Enable event-based result matching and notify users in real time through their respective dashboards and via email.

3.4.3 User Acceptance Testing (UAT)

Get feedback regarding user-friendliness, timing of notifications, and performance by conducting trials with prospective users to ensure that the interface meets expectations.

3.5 Deployment

3.5.1 Environment Setup

- **Containerization:** Docker containerizes the backend for easy portability between development, testing, and production.
- **Cloud Infrastructure:** Deployment on a cloud service provider ensures redundancy, scalability, and data security.

3.5.2 Continuous Integration/Continuous Deployment (CI/CD)

CI/CD Pipeline: This is the automated test, build, and deploy done in updates; improves the integrity and deployment time for code.

3.6 Maintenance

3.6.1 System Monitoring

- **Real-time Monitoring:** Both Prometheus and Grafana show real time up-time and performance so it was easy for us to quickly spot any faults when they occurred.
- **Logging:** The user activity is logged for security or debug purposes as well as to keep record of all errors.

3.6.2 Future Enhancements

- **Direct Bank Integration:** API incorporation for bank related services usage.
- **User Data Validation:** To do even more so far as data validation is concerned in order to minimize the inaccuracies likely to be given by the users.
- **Prize Bond Marketplace:** The user is also in a position to purchase and sell the prize bonds from the website.

Chapter 4

Project Architecture

4.1 Frontend Architecture (User Experience)

The system's frontend design of prize bond notification is constructed in such a way so that any user from any education level can easily articulate the whole system smoothly and get an immersive experience out of it. We have concentrated more on the easy accessibility of components, tools and made the whole system responsive regardless of any device like windows, mac, android or ios operating system.

4.1.1 Overview of the User Interface (UI)

In order to make our system responsive and easy to access, we have used several combinations of web technologies, tools and frameworks (HTML, CSS, Javascript). The core target was to make the system simple and user friendly so that any user can easily scan their respective prize bond and get their desired notification.

4.1.2 Flow of User Experience (UX)

The final goal was to construct the system in such a way so that users only go through some easy steps. Below are some of the important steps:

A. Landing Page/Home Page

Description: The home page of the system is organized in an easy to understand and simple way whenever the user starts to use the system. A visible button is there to ask users whether they want to input their prize bond number manually or they can enter the number by scanning the prize bond numbers was our main call to action.

Important attributes

- **Sign up or log in:** A simple dashboard can be visible for logging and creating new account.

- **Prize Bond Entry Field:** Multiple options are there to input the prize bond number and several types of scanning system.
- **Notifications Section:** This section gives important information like next draw dates or any kind of essential announcements.

B. Scanning and Entering Prize Bonds

Description: Prize bond number can be entered manually (typing) or scanning the numbers from prize bond.

Important attributes:

- **Prize Bond Scanning:** This section is responsible for scanning the prize bond number using mobile phones camera or can be used any webcam for computers. Our system can automatically read and store number, characters. OCR or optical character recognition has been used.
- **Manual Entry:** To ease user's accessibility manual number input field is there.
- **Verification:** The system will automatically reply if the user has inputted any wrong sets of prize bond numbers or any invalid input.

C. Dashboard

Description: Once any user log in to the system they will be directed to their individual dashboard.

Important attributes:

- **Bond List:** A section is there where users can see their submitted prize bond numbers.
- **Result Status:** A mark for every prize bond number will show whether the result has been announced or any pending results that might be announced later.
- **Winning History:** User's winning history will take place in a table where their previous history will also be shown.

D. Result Notification

Description: Once the result gets announced users will be able to get it from their dashboard.

Important attributes:

- **Real-Time Updates:** The moment a bank updates/releases any results our system will be updated automatically as well.
- **Winning Bond Highlight:** The prize bond number will correspond to the winning number to highlight the results.
- **Detailed Result Breakdown:** A user will get a plethora of information and any details regarding the draw and result breakdown will also be there.

E. Email Notification System

Description: Once the bank publishes the results, an announcement along with the results will be sent to users email.

Important attributes:

- **Notification Trigger:** The email will contain important details about the prize bond and if the user has earned any prize.
- **Dashboard Link:** An attached link will be sent for further information and a link directed to users dashboard will also be sent.
- **Unsubscribe/Manage Preferences:** The accessibility of getting notified by email can be controlled whether an user wants to get notification via email or not. They can turn off notifications.

F. Account and Profile Configurations

Description: To make an user's profile according to their preferences they will be able to make any necessary edit to their profile information and change their settings.

Important attributes:

- **Profile Information:** User will be able to change their credentials (used id, password) and other miscellaneous information.
- **Notification Preferences:** Notification via email can be managed and they would be able to turn on notifications and turn them off as well.
- **Delete Account:** Any user can delete their account anytime if they want. An option will be there for this too.

4.1.3 Tools and Technologies

Frontend Framework: Widely used web technologies like HTML, CSS are used to form the system dynamic and fruitful user experience.

UI/UX Design: To make the system aesthetically pleasing the website design was created by tools like Figma.

Responsiveness: CSS responsive framework is used to make the system responsive regardless of any device.

Real-Time Updates: The dashboard can update the results and send notification to the user through web sockets.

4.1.4 Authentication and Security

User authentication: The system is well designed to protect the user information like prize bond number and their any other credentials. To implement this the system is constructed using authentication protocols. All the information is encrypted prior to being sent for the authentication and backend processing. This includes users email, password, prizebond number and other major information.

4.1.5 User input and accessibility

Accessible Design: The platform was designed so that any level of technically sound people can use it. It was to make sure to use user interface compliance with industry standards.

Error Handling and Feedback: The system automatically generates a prompt to notify any wrong inputs that have been created by the user. It shows important prompts to guide them to enter proper information. Users can manage their prizes in an effective manner because of our proper frontend architecture.

By taking help of frontend architecture users can manage their prizebond and get efficient and faster notification in an effective manner.

4.2 Backend Architecture

The backend architecture of the prize bond notification system is created to manage important tasks like user authentication, data storage, notification management, result retrieval, and constructing a proper dashboard. To ensure a reliable connection between the frontend and the backend architecture, the system has been designed efficiently. Moreover, it has focused on scalability and other dependability.

4.2.1 1. Overview of the Backend

Multi-layer architecture with the help of necessary elements of the system's backend is constructed.

Application Layer: It manages any kind of user request and any sort of communication with remote service.

Data Layer: Oversees all storage functions, including notification logs, reward bond numbers, and user information.

Service Layer: Incorporates third-party services like email and APIs for retrieving results. MySQL, Python with Django, and external APIs for retrieving prize bond results are frequently utilized technologies for backend implementation in this architecture.

4.2.2 2. System Components and Workflow

A. Authentication and User Management

Description: This part manages user registration, login, and profile administration, guaranteeing safe system access.

- **Technologies:** OAuth2 or JWT (JSON Web Tokens) are used by the system for authentication.
- **Signup and Login:** Users provide their email address and password during registration, and these are safely stored.
- **Token-based authentication:** Users can stay authorized without having to enter their credentials again because a JWT token is created and utilized for all future requests after a successful login.

- **Password management:** Offers email-based password reset and recovery capabilities.

B. Data Management for Prize Bonds

Description: This part keeps track of, controls, and handles the users' prize bond numbers.

- **Structure of a Database:**

- **User Table:** Holds user data like password, email, and preferences.
- **Prize Bond Table:** Retains the prize bond numbers that each user has input and links them to the user using foreign keys. The result table contains the prize bond results as they are provided by the bank or prize bond authority.
- **Notification Table:** Keeps track of the time, state, and notification type (dashboard or email) that users receive.

C. Retrieval of Prize Bond Results

Description: Getting the most recent prize bond results from the bank or other trustworthy sources is the responsibility of our service.

- **Approach:**

- **Integration of APIs:** The backend regularly connects to the bank's or other organizations' public or private API using cron tasks to retrieve the most recent prize bond results.
- **Web scraping (in the absence of an API):** In the absence of an API, the system may use web scraping techniques to gather data from the bank's public results page. The scraping service processes the website, extracts the winners and prize bond numbers, and stores the data in the database.
- **Validation of Results:** The backend compares the system-stored prize bond numbers with the results when they have been retrieved.

D. Email Notification System

Description: Notifies users automatically when their prize bond results are available.

- **Triggering Mechanism:** After the results are obtained, a backend job looks for prize bonds that have won and sends out email alerts to users whose bond numbers correspond to the winning outcomes.
- **Content of the Email:** The user's bond number, the draw's result, and a link to their dashboard for more details are among the personalized details included in the email.
- **Retry Mechanism:** If an email delivery attempt fails, the system records the unsuccessful attempt and tries again using a message queuing system.

E. Notification on the Dashboard System

Description: The user dashboard is updated in real-time when the numbers of the winners and reward bonds match.

- **Technologies:** WebSocket or AJAX polling is used to deliver real-time updates, ensuring that users may see results as soon as they are made available.
- **Real-Time Updates:** Every time a new result is uploaded, the backend sends out a WebSocket event, which updates the user's dashboard with the winning details.
- **Database querying:** To update the user's dashboard with the most recent prize bond statuses (such as "Won" or "Not a Winner"), the system executes database queries.
- **Result Caching:** To reduce database load and enhance performance, caching technologies such as Redis are used to cache frequently retrieved results.

4.2.3 3. Security Points

Encryption: SSL/TLS is used to encrypt all sensitive data, including prize bond numbers and user passwords, both in transit and at rest.

Database Security: Role-based access control is used to safeguard database access, guaranteeing that only authorized users and services are able to interact with the data.

API Rate limitation: To prevent DDoS attacks, the system has rate limitation in place to make sure that external services (such as email notifications and result retrieval) are not overloaded.

4.2.4 4. Performance Optimization and Scalability

Scaling: The backend's ability to grow horizontally allows for the addition of more servers to accommodate growing user traffic and processing capacity.

Load Balancing: Using a load balancer, incoming traffic is dispersed evenly among several backend instances.

Optimization of Databases: In order to improve performance when retrieving prize bond data, indexing and query optimization techniques are employed. Furthermore, a caching layer that caches frequently asked queries lowers demand.

4.2.5 5. Logging and Error Handling

Error Logging: To record system errors, problems retrieving prize bond results, and notification failures, the backend makes use of a logging framework.

Monitoring: Prometheus and Grafana are two linked tools for tracking system uptime and performance in real-time.

Fallback Mechanism: The system provides fallback techniques, such as queueing unsuccessful notifications for retry or showing error warnings to users on the frontend, in case external services (like email or result APIs) fail.

4.2.6 6. Deployment and DevOps

Containerization: Docker containers are used to deploy the backend, which facilitates system management and scalability across development, staging, and production environments.

CI/CD Pipeline: To automate testing and deployment procedures, a continuous integration/continuous deployment (CI/CD) pipeline is established.

Cloud Infrastructure: To provide scalability, redundancy, and security features, the backend is housed on cloud infrastructure.

This backend architecture aims to ensure that the prize bond notification system functions well, can handle large user populations, and can handle the complexity of automating notifications while maintaining data security and integrity.

4.3 Use Case and ER Diagram

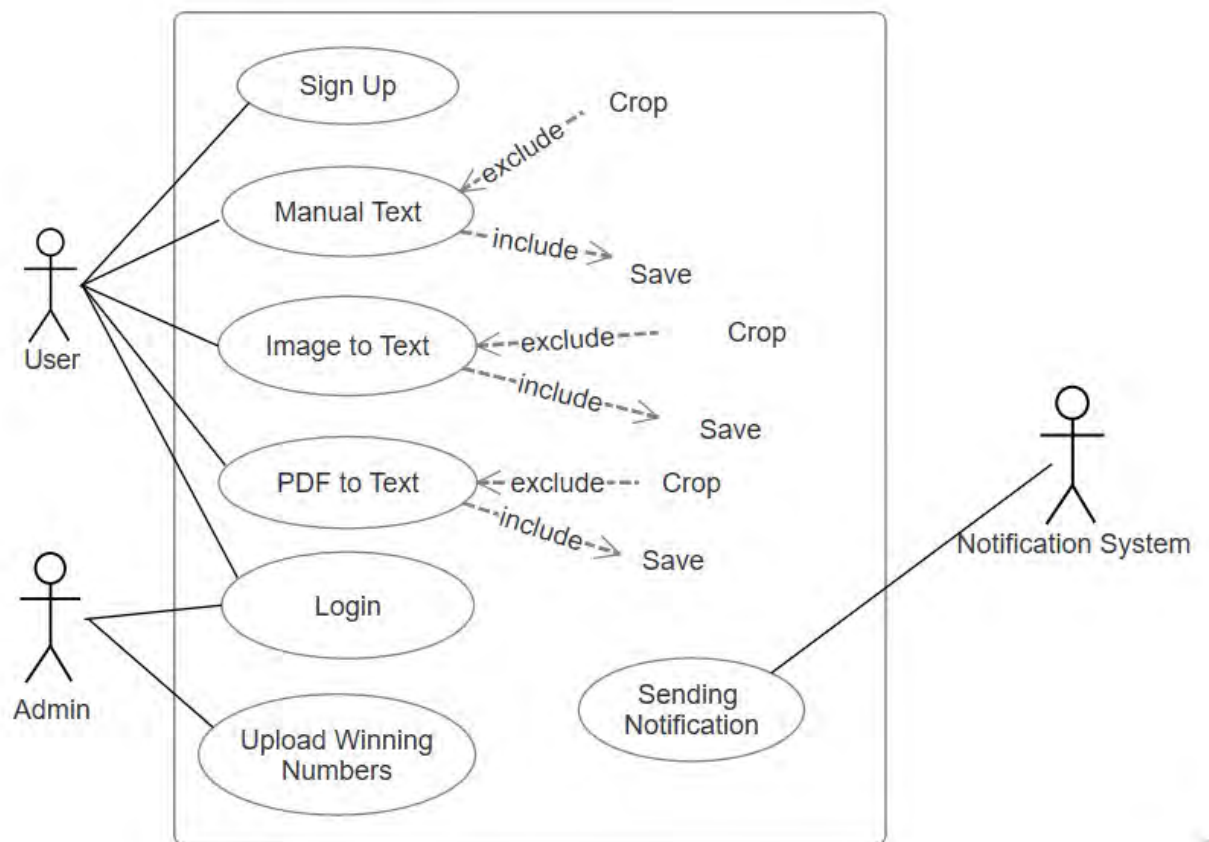


Figure 4.1: Use Case Diagram

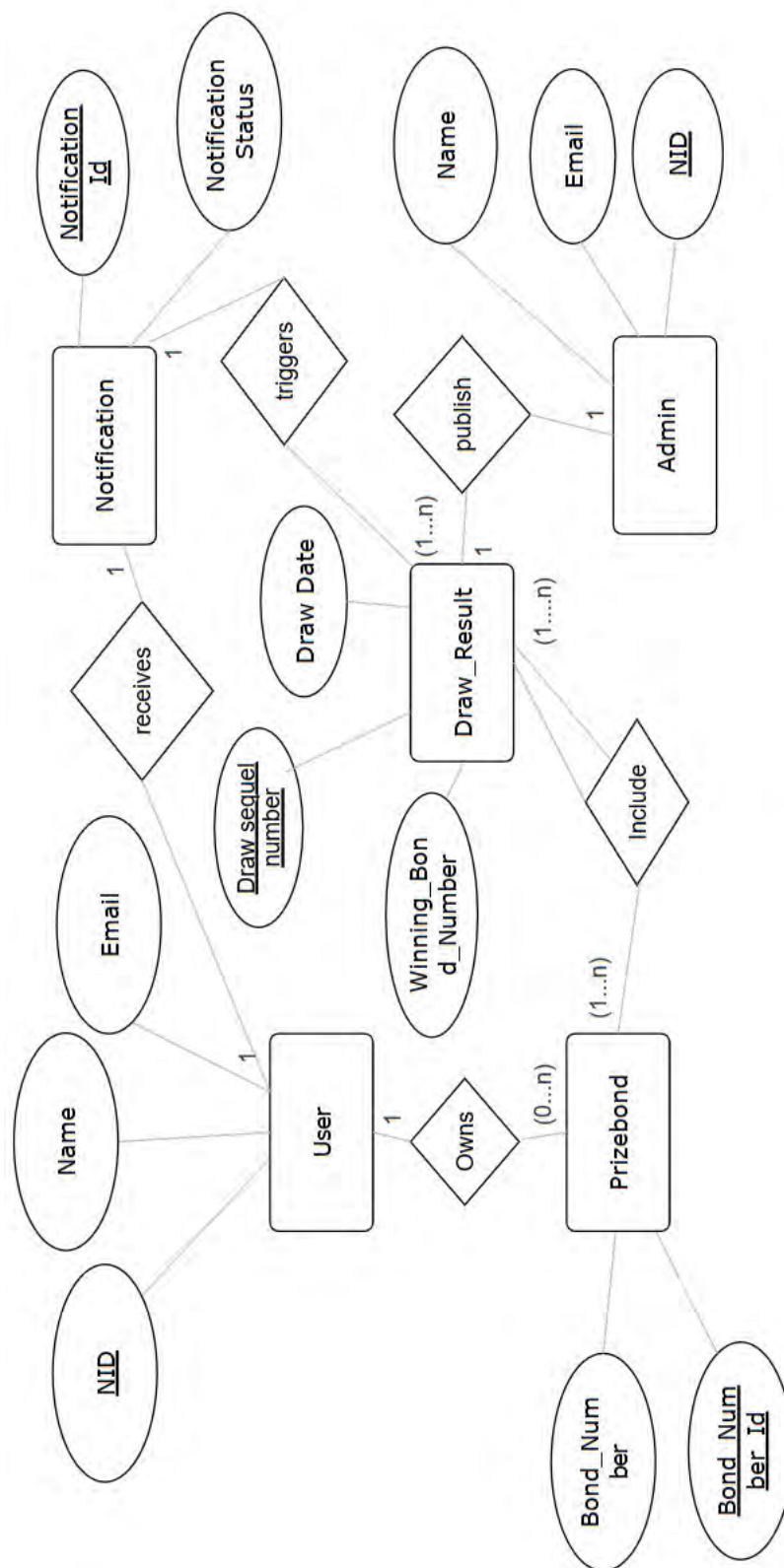


Figure 4.2: ER diagram

Chapter 5

Methodology

5.1 Overview of Methodology

We have implemented the Optical Character Recognition(OCR) which was used for Data Extraction. This implementation is used to gear the user experience in the prize bond lotteries by extracting the bond number from image and pdf. This approach ensured about 90 percent accurate data extraction in various scenarios (Inbasekaran et al., 2021). Moreover, we have implemented Options where user can submit prizebond data as a Manual input (single entry) and Multiple entry at a time(Range of input) which gives user more consistency.

Also, the OCR data from PDF, image and Manual entry which is specifically built for focusing on extracting only prize bond numbers by using validation checks for bond number formats and Regex. Duplicate entries are prevented and input limitation is added for range of input and appropriate error or success messages are displayed to the user.

For superuser or admin we have take additional privileges, such as specifying a bond position and managing data in the `result_data` class. Which are available where for users and it is stored into the `ocr_data` class. We have also utilized MySQL as the database for storing and managing the extracted data.

Most importantly, Data flow or synchronization mechanism between different models is implemented. where we used event-driven programming. User management is also implemented. Lastly we are sending in app notification and email notification to the all users who wins .

Here it was the overall overview of the whole methodology. The detailed methodology of **PDF to Text Extraction , Image to text extraction, (Manual Data entry / Range of input), AJAXIntegration, (Result Matching, Data Storage and Presentation), Notification system and user mangement is Described Below:**

5.2 PDF to Text Extraction

The PDF to Text Extraction process is implemented by using the PyMuPDF library to extract text from uploaded PDF files(Guedes and Da Silva, 2021). The system handles PDF files which are uploaded by the users. Then it reads the pdf file and extracts text content by using PyMuPDF library. The extracted text was then parsed using regular expressions (regex) to identify both Bengali and English numbers only.

The regex pattern is designed to match consecutive 7-digit numbers which ensures that only prize bond numbers are captured from the text.

The system provides a unified graphical user interface (GUI) for both admins and regular users. However, there are some key differences in functionality based on the user's role:

- **Admin Privileges:** When the user is an admin then the extracted data is stored in the `result_data` class. Admins are also granted an additional feature to assign and store a prize bond position along with the data.
- **User Data:** For regular users, the extracted data is stored in the `ocr_data` class. Here for user they don't have the option to specify a prize bond position.

This design efficiently uses the same function and GUI for both admins and regular users, while distinguishing data storage and features based on the user's role. It ensures seamless data handling and role-specific functionality.

5.3 Image to text extraction

For image files we used pytesseract library which is much faster and efficient for ocr technology and its is a popular and advanced OCR tool. This library converts text from images into digital format (Inbasekaran et al., 2021). Users will upload image files containing prize bond numbers, and pytesseract performs OCR on the images. Which converts the visual text into machine-readable text.

The system used regular expressions(regex) to identify both Bengali and English numbers only from the image for matching consecutive 7-digit numbers. Here it ensures precise extraction of prize bond numbers from image.

Much like the PDF-to-text extraction process here the system distinguishes between admin and regular user privileges while maintaining a unified graphical user interface (GUI) for both.

- **Admin Privileges:** For Admins, extracted data is stored in the `result_data` class. They have the additional functionality to assign prize bond positions.
- **User Data:** For regular users, data is stored in the `ocr_data` class. Here they don't have the ability to specify a prize bond position.

5.4 Manual Data entry

For the Manual Data Entry here users manually enter prize bond numbers into the system through a web form. The system allows for two types of inputs **individual bond number** and **range of bond numbers**.

To validate these inputs we have implemented regular expression (regex) here. Which ensures the bond numbers follow the format of two letters followed by seven digits. This validation guarantees that only valid prize bond numbers are accepted.

In the case of range input this system verifies that the range is logical (i.e., the starting number is less than the ending number) and that the total number of bond entries does not exceed 100 at a time. After validated, the system processes the bond numbers in sequence and store them in the appropriate database.

Just like the other functionalities, its uses a unified graphical user interface (GUI) for both admins and regular users.

- **Admin Privileges:** For Admins, extracted data is stored in the `result_data` class. They have the additional functionality to assign prize bond positions.
- **User Data:** For regular users, data is stored in the `ocr_data` class. Here they don't have the ability to specify a prize bond position.

Here the overall functionality provides seamless manual entry while enforcing strict validation and preventing duplicate entries.

5.5 AJAX Integration and JSON Response

The integration of AJAX (Asynchronous JavaScript and XML) alongside JSON (JavaScript Object Notation) played a crucial role in handling file uploads (image and PDF) and real-time validation for manual or range inputs (Paulson, 2005). AJAX here allows the system to perform text extraction processes asynchronously. Also AJAX enables the sorting of result data in real-time. Finally, This ensures that users can interact with the system without having to reload the entire page. This provides a smooth and responsive experience (Paulson, 2005).

In our system, the JSON format is used for transmitting data between the client and the server. After an AJAX request is made then the server processes the data (such as extracted bond numbers) and returns the results in JSON format. The structured nature of JSON makes it easy to handle and update the front end with real-time results from uploaded files. Finally, This includes displaying extracted text from PDFs and images immediately after processing. So we don't need to refresh the page (Zakas, 2010).

Additionally, users can zoom and crop images for better accuracy before submitting the file. which enhances the user experience further and improves ocr accuracy. Also, If any discrepancies are found in the extracted bond numbers then the users

can edit the data directly. After satisfied with the correct results users can save the final prize bond numbers. Finally, it sent back to the server using AJAX and the data is stored or processed further in JSON format(Zakas, 2010).

This combination of AJAX and JSON ensures efficient and real-time communication between the client and the server. This allow the dynamic updates and provides a highly interactive and user-friendly system.

5.6 Result Matching, Data Storage, and Presentation

Result Matching:

For result matching here, we utilize automated and event-driven programming through Django signals. This approach ensures real-time synchronization of data between users(`ocr_data`) and admins(`result_data`). By leveraging Django's built-in signals, we can automate the process of updating relevant data models whenever changes are made, ensuring accurate and up-to-date information across the system (Holovaty and Kaplan-Moss, 2009). Specifically, if a user uploads their `ocr_data` before the corresponding result is uploaded by the admin, the system automatically matches the bond numbers and inserts the data into the `winner_data` class. Similarly, if the admin uploads `result_data` first and the user submits matching `ocr_data` later then the data is also inserted into `winner_data`. This matching and filtering process is handled using Django's built-in functions, keeping in mind that ensuring accuracy.

Data Storage:

We use a MySQL relational database to manage and store the data. We are ensuring strong data relationships and consistency through primary key and foreign key relations. These relationships guarantee data integrity across multiple classes such as `ocr_data` , `result_data` , and `winner_data`.

- If a user deletes their account then all their related data including `ocr_data` and `winner_data` will be automatically removed from the system by maintaining data hygiene. Similarly, if a user deletes specific `ocr_data` entries then the corresponding `winner_data` will be also deleted.
- For admins, deleting single or multiple entries from `result_data` will like wise trigger the automatic deletion of related `winner_data` . This seamless management of data ensures that no orphan records exist and enhances the overall efficiency of the database.
- Moreover, to prevent overloading the system we have implemented that users and admins are restricted to entering a maximum of 100 bond numbers at a time. This feature works when using the range input feature. It ensures load balancing and optimal system performance.

Here the overall functionality provides seamless manual entry while enforcing strict validation and preventing duplicate entries.

Below the figure 4.1 shows the Design of the Data storage and class Diagram:

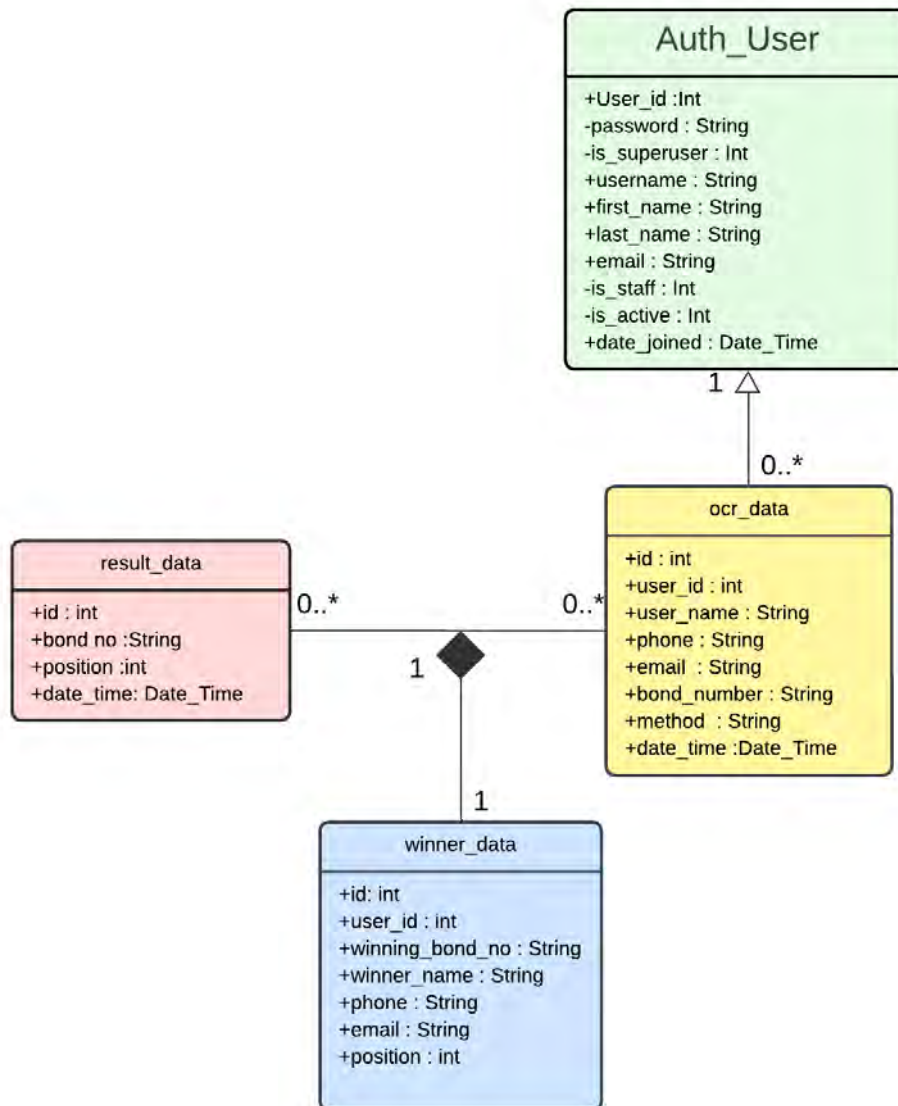


Figure 5.1: Data Storage and Class Diagram

Presentation of Data:

We have handled data presentation differently for admins and users to maintain privacy and control over the data.

- **Admin View:** Here, admins can only view, update, and delete the `result_data` they have uploaded only. They do not have access to the users `ocr_data`. So, it ensures user privacy and data security.
- **User View:** Each user can only view, edit, or delete their own `ocr_data`. No user can access or modify another user's data. So that guarantees data protection and personalization. Additionally, users have access to a feature where they can view a complete list of all winning bond numbers and their respective positions.

So, a layered approach to data presentation ensures privacy and role-based access. Moreover, it provides a clear and streamlined interface for both users and admins. This enhances user engagement while maintaining strict data access controls based on user roles.

5.7 Notification system and user management

Notification system:

In our system notification system plays an important role. This feature is implemented using Django's email functionality. This system sends automated email notifications to users when they win a prize bond. The system collects and groups bond data for each winner. After that, it ensures that if a user has multiple winning bonds. If yes then all bond numbers and their respective positions are included in a single email.

The process works as follows:

- **Winner Data Collection:** Our system collects all winning bond numbers associated with each user by querying the `winner_data` model. Then, each user's email is mapped to their list of winning bonds and ensures that all relevant data is included in one notification.
- **Email Notifications:** Once the data is prepared then a personalized email is sent to each winner about the list of winning bonds and positions. This ensures that winners receive timely and accurate information about their prize bonds without needing to manually check the system.

User Management:

User management is handled by an interface where admins can view and modify user data. This section includes role-based access controls which allow admins to manage user's roles and permissions dynamically.

- **User List:** Admins can view a list of all registered users, and display key details such as first name, last name, username, and their status as staff or admin.
- **Role Management:** Admins have the ability to toggle user roles. Can be assigned a user as user/ Admin. Admins can promote or demote users to or from admin roles, giving them full access to the system's administrative features.
- **User Deletion:** If a user is deleted then all related data including their `ocr_data` and any associated prize bond data will be automatically removed from the database. This ensures that no orphaned and duplicate data remains and maintains the integrity of the system.

This unified notification and user management system ensures efficient handling of prize bond results and user permissions. In the meantime, it maintains a high level of automation and user experience.

Below the figure 4.1 shows a visual representation of The core process of the Methodology.

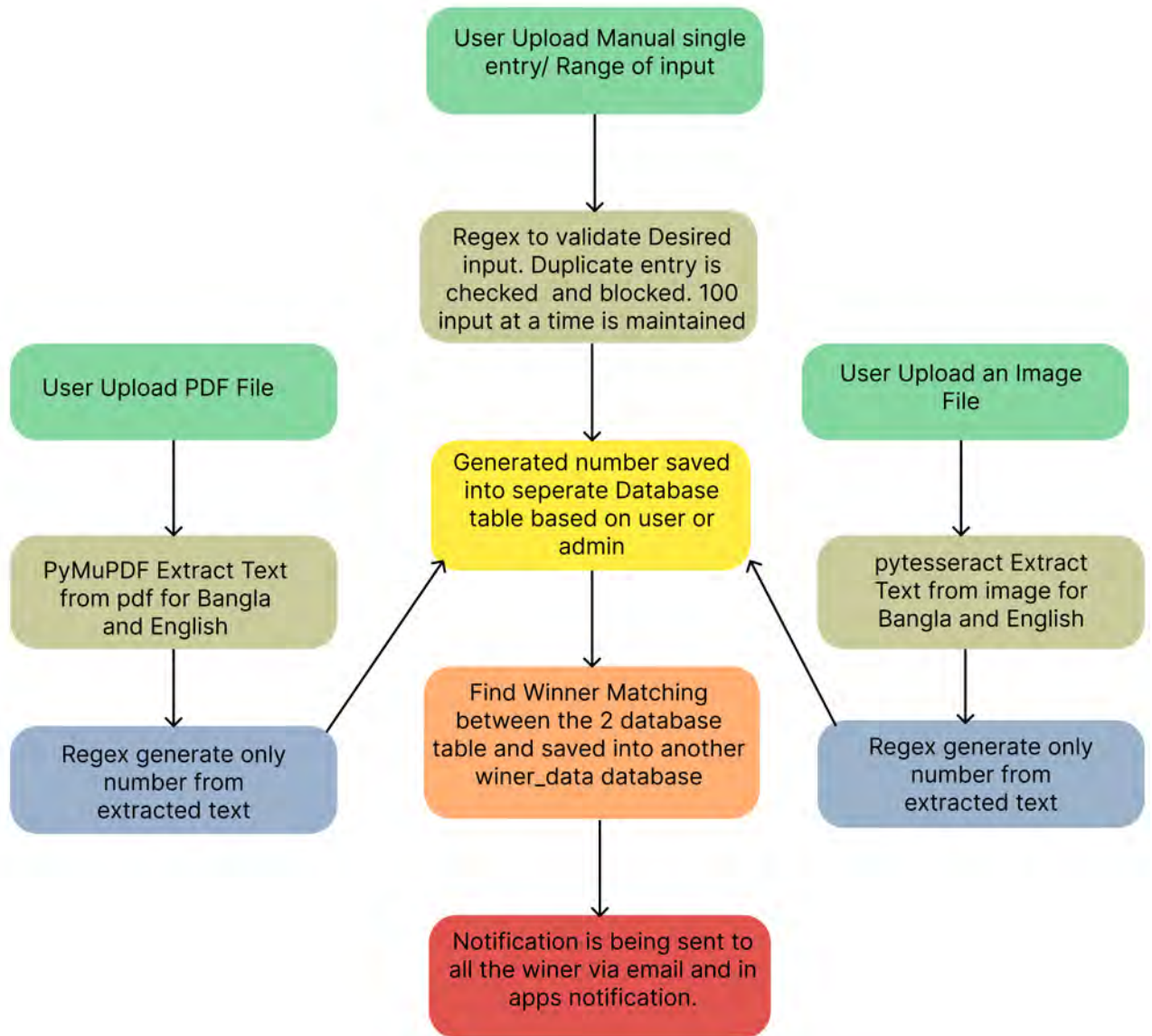


Figure 5.2: Core process of the Methodology

Chapter 6

Preliminary Analysis and Visualization

6.1 Preliminary Analysis Flow chart

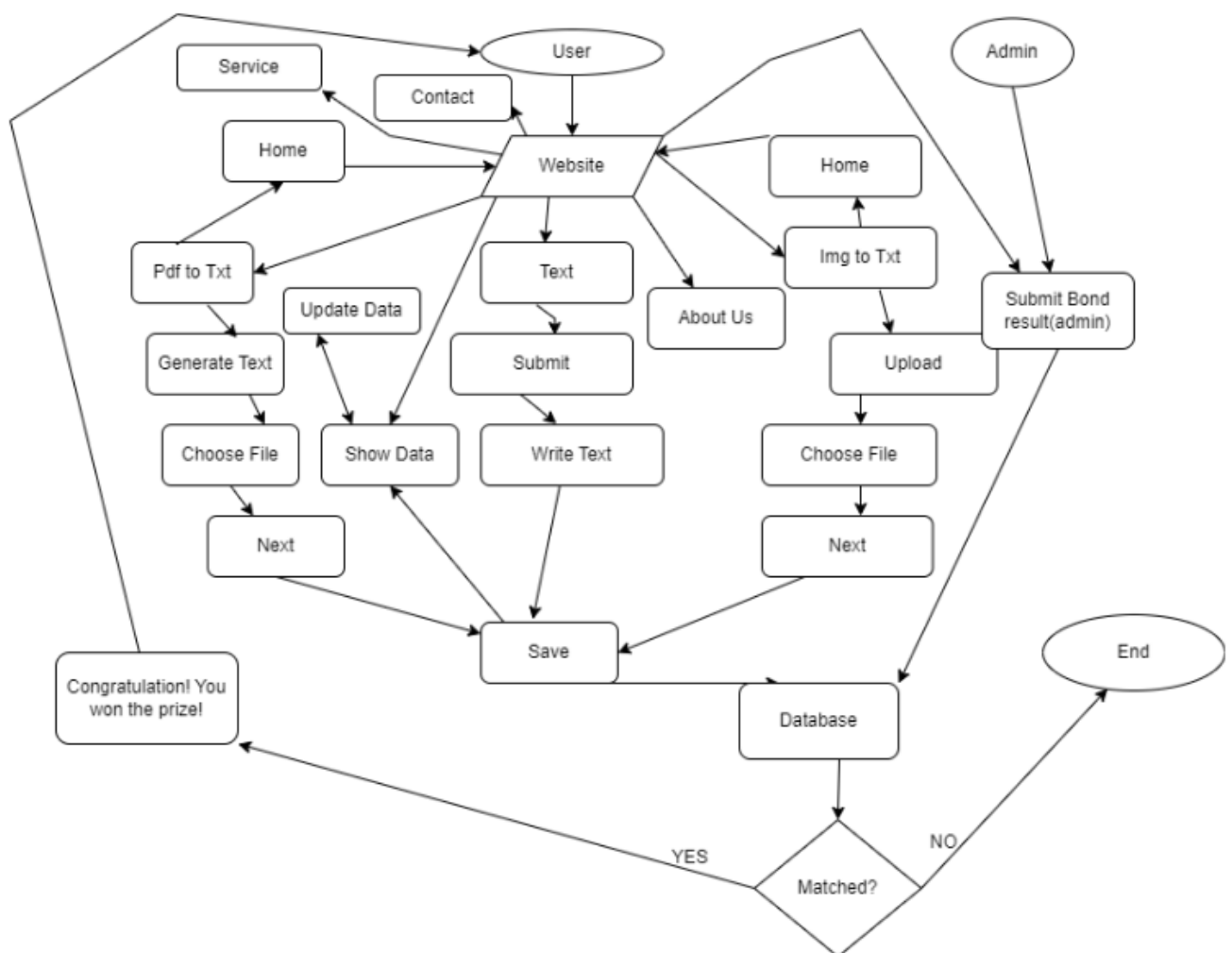
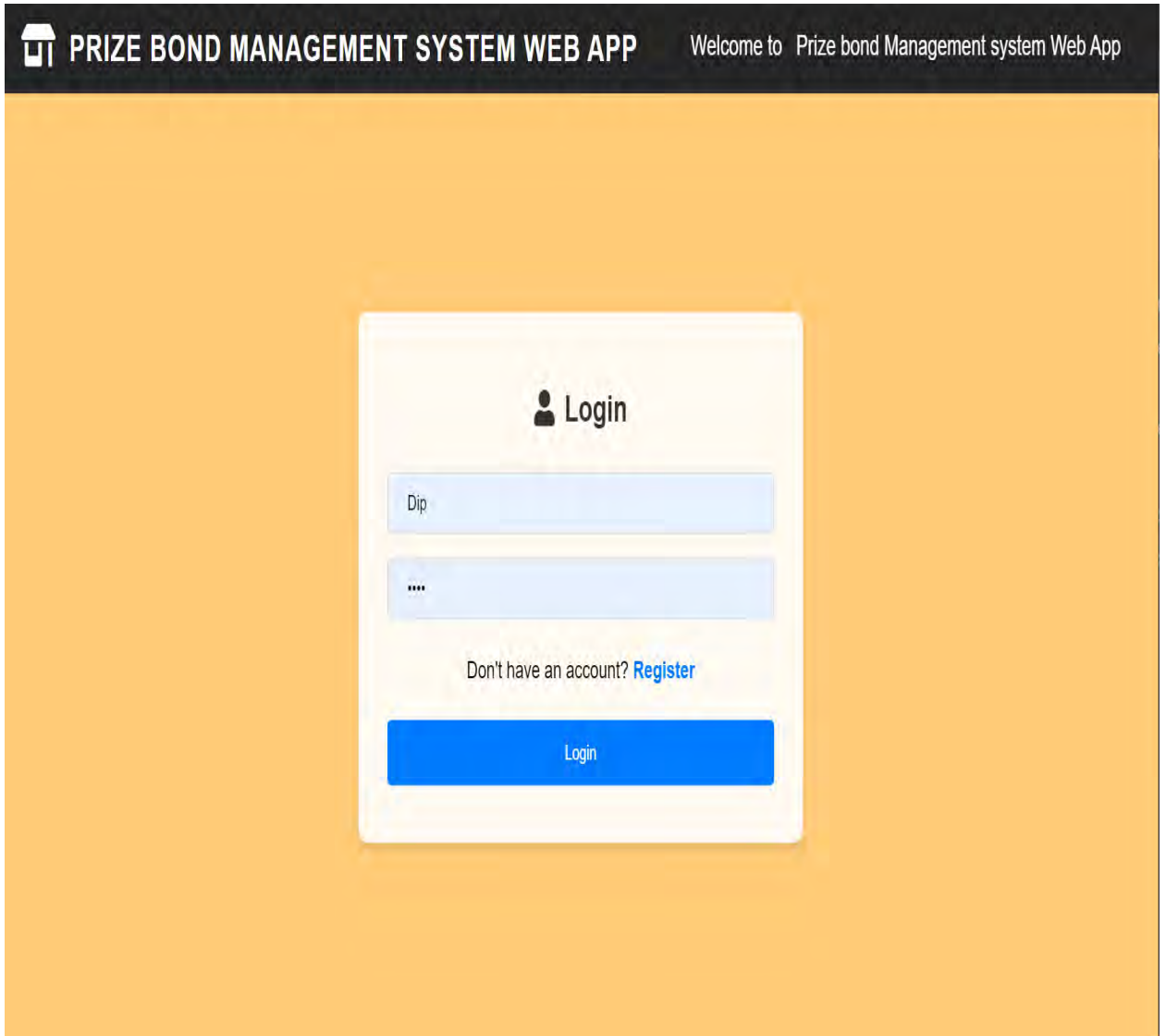


Figure 6.1: Preliminary Analysis flow chart

6.2 Visualization of project

Login Page:

This is our Login Page.



The screenshot displays the login interface of the Prize Bond Management System Web App. At the top, a dark header bar contains the app's name and a welcome message. The main content area has a solid orange background. Centered on this background is a white login card. The card features a user icon and the word 'Login'. Below this are two input fields: the first is labeled 'Dip' and the second is for a password, indicated by four dots. A link to 'Register' is provided for users without an account. At the bottom of the card is a prominent blue button labeled 'Login'.

PRIZE BOND MANAGEMENT SYSTEM WEB APP Welcome to Prize bond Management system Web App

 **Login**

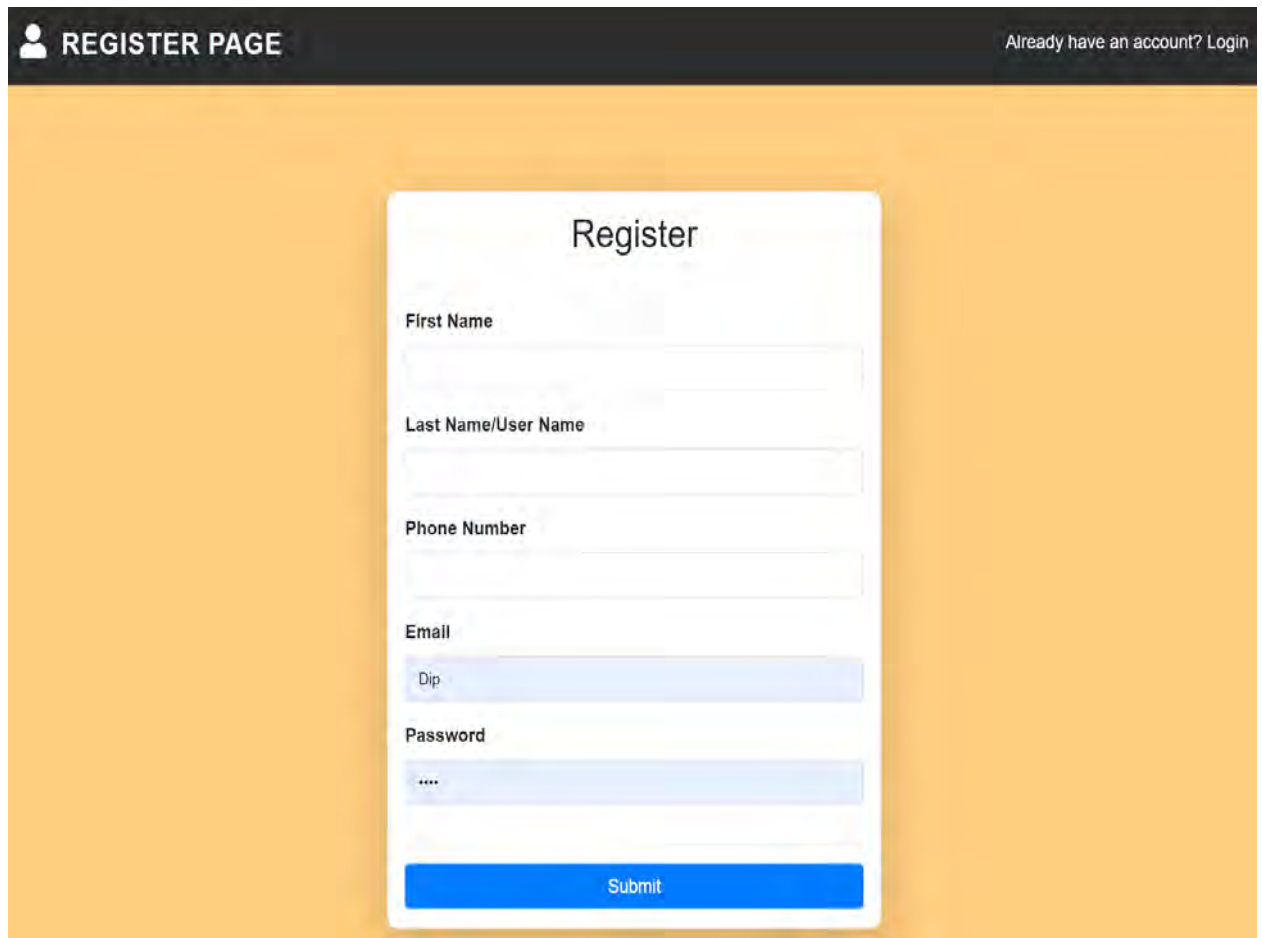
Dip

Don't have an account? [Register](#)

Login

Register Page:

This is our Register Page.



The image shows a user registration form titled "Register" centered on a solid orange background. The form is a white card with a light gray border. At the top of the page, there is a dark gray header bar. On the left side of this bar is a white user icon followed by the text "REGISTER PAGE" in white. On the right side of the header bar is the text "Already have an account? Login" in white. The registration form itself has the title "Register" at the top. Below the title are five input fields, each with a label to its left: "First Name", "Last Name/User Name", "Phone Number", "Email", and "Password". The "Email" and "Password" fields have a light blue background. The "Email" field contains the text "Dip" and the "Password" field contains four dots. At the bottom of the form is a blue button with the text "Submit" in white.

REGISTER PAGE Already have an account? Login

Register

First Name

Last Name/User Name

Phone Number

Email

Dip

Password

....

Submit

pdf to Text: We are successfully extracting only numbers from this pdf. We can also edit the number if any wrong number is found. Also Here Position field will appear when login as admin.

Upload a PDF file

Choose File

Choose FileCSE 470 As... CFG (1).pdf

Uploaded PDF:

1 / 4

−

+

Question 1 [10 Marks]

A program that finds all Pythagorean triples (a, b, c) such that $a^2 + b^2 = c^2$ and all numbers are less than or equal to n. If a, b, or c is divisible by 5, print "Divisible by 5".

```
Public void find_pythagorean_triples( int n)
{
    for (int a = 1 ; a < n+1 ; a++)
    {
        for ( int b = a ; b < n+1 ; b++)
        {
            for( int c = b ; c < n+1 ; c++)
            {
                if ( ((a * a) + (b * b)) == (c * c))
                {
                    if ( a % 5 == 0 || b % 5 == 0 || c % 5 == 0)
                    {
                        System.out.println(" a "+a + " b "+ b+ " c " + c + " is divisible by 5 ");
                    }
                    else
                    {
                        Svsstem.out.println( "Pythagorean Triple: a "+ a + "b "+b + "c "+c);
                    }
                }
            }
        }
    }
}
```

Generate Text

Extracted Text:

2354101

Save

Extracted Text:

2354101

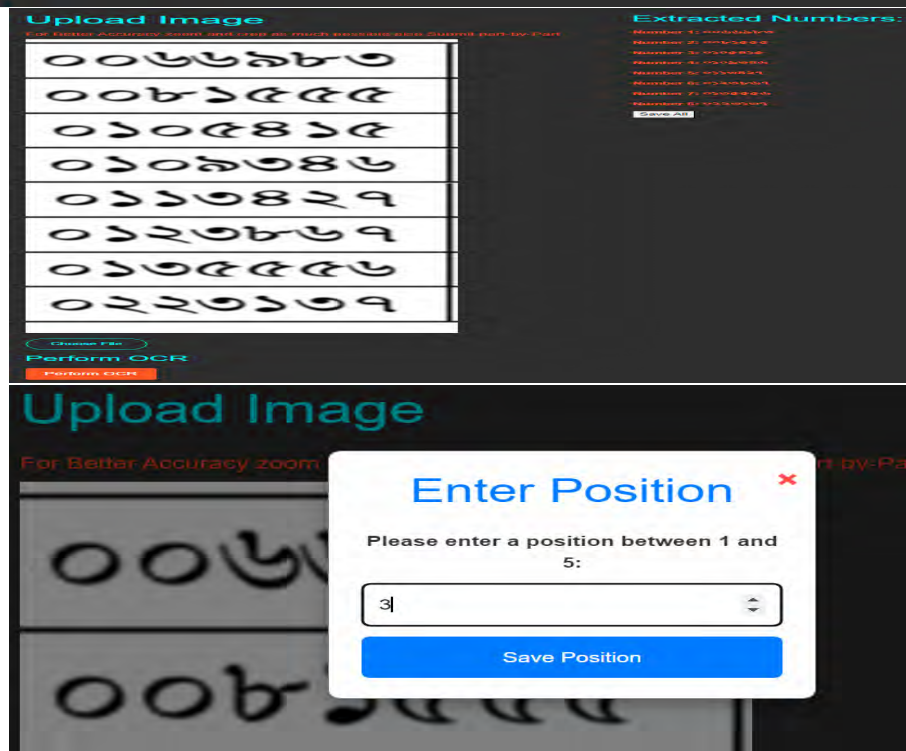
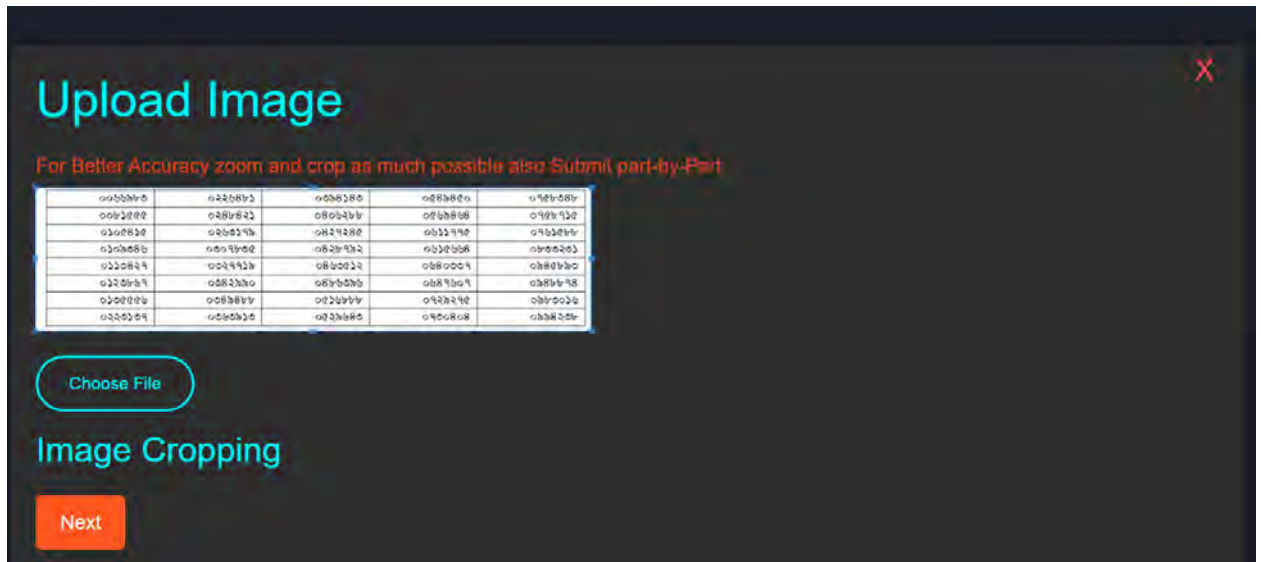
Position:

1

Save

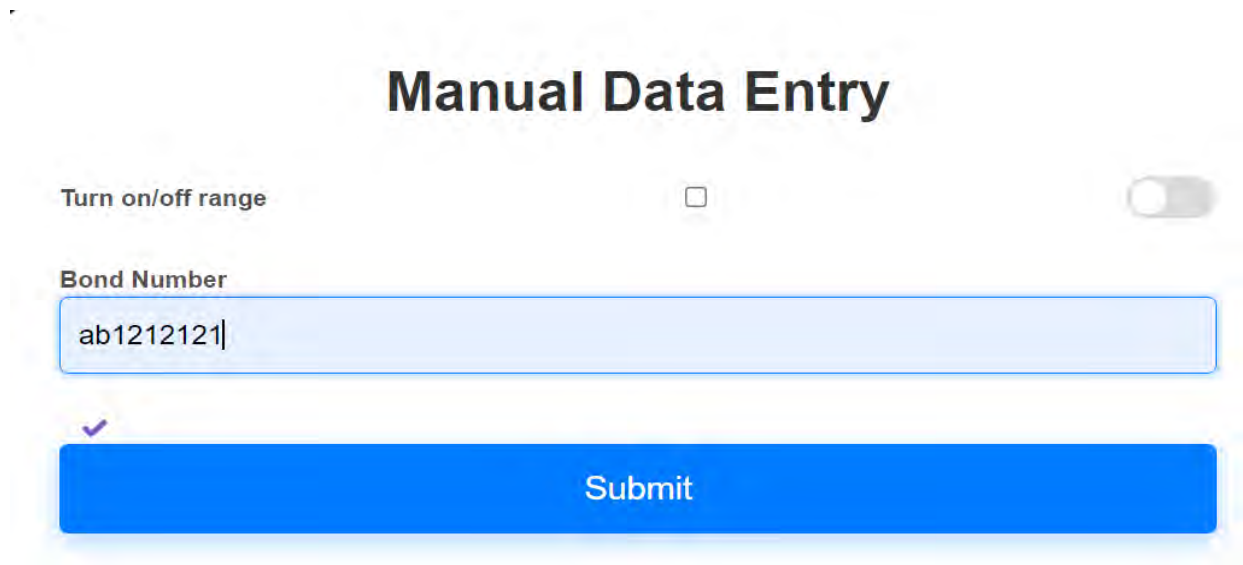
Image to Text:

We have implemented the feature where user can zoom in and crop the image for better user experience and accurate result. Next we are performing OCR and extracting the text and saving it into the database. Here we are getting 100% accurate data when cropping and performing OCR. Enter Position modal appears when Admin Use it.

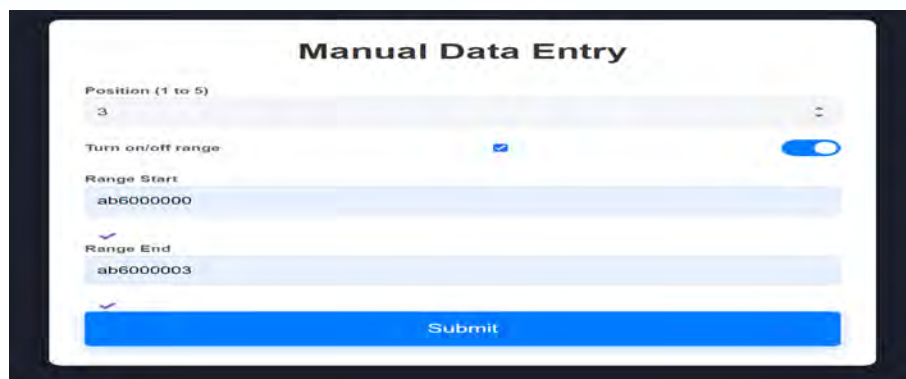


Manual Data Entry:

We have implemented the feature where user can enter manual single and range of numbers entry. Also position field appears when admin uses the functionality and for admin data is saved into separate database.



The image shows a web form titled "Manual Data Entry". At the top, there is a label "Turn on/off range" followed by a checkbox and a toggle switch, both of which are currently turned off. Below this is a label "Bond Number" followed by a text input field containing the value "ab1212121". A small purple checkmark icon is visible to the left of the input field. At the bottom of the form is a large blue button labeled "Submit".



The image shows a web form titled "Manual Data Entry" with a dark border. It contains several fields: "Position (1 to 5)" with a value of "3", "Turn on/off range" with a checked checkbox and a blue toggle switch, "Range Start" with a value of "ab6000000", and "Range End" with a value of "ab6000003". Each of the last three fields has a small purple checkmark icon to its left. At the bottom is a blue button labeled "Submit".

Show submitted Data user:

All the price bond numbers are automatically saved into the database into ocr data model for user. We can see the method we used to extract the price bond number and Delete.

The screenshot shows a web application interface for 'Prize Bond Lotteries'. The user is logged in as 'Dihan'. The left sidebar contains navigation links: PDF to Text, Image To Text (OCR), Manual Data Entry, Show Data (highlighted), Check Winner, and Logout. The main content area displays the title 'All Price Bond Data for User: Dihan' above a table of submitted data.

Unique User ID	Bond Number	Method	Actions
18	2354101	pdf	Delete
18	ab6000000	Manual_Entry	Delete
18	ab6000001	Manual_Entry	Delete
18	ab6000002	Manual_Entry	Delete
18	ab6000003	Manual_Entry	Delete

[Delete All My Data](#)

Show submitted Data by Admin:

All the price bond numbers are automatically saved into the database result data model for Admin. Admin can only see and Delete it.

PDF to Text

Image To Text (OCR)

Manual Data Entry

Manage Results

Winners

User Management

Logout

Prize Bond Lotteries

Home

About

Contact

Logged in as Admin: Dip

Manage Result Data

ID	Bond Number	Position	Date & Time	Actions
852108	00000000	3	2024-10-17 14:52:50	Delete
852116	ab6000000	3	2024-10-17 02:55:02	Delete
852117	ab6000001	3	2024-10-17 02:55:02	Delete
852118	ab6000002	3	2024-10-17 02:55:02	Delete
852119	ab6000003	3	2024-10-17 02:55:02	Delete

Delete All Result Data

Check Winner by user:

All the winning bond number along with bond number will be shown. Only a user can see his winner data .Others user data is protected

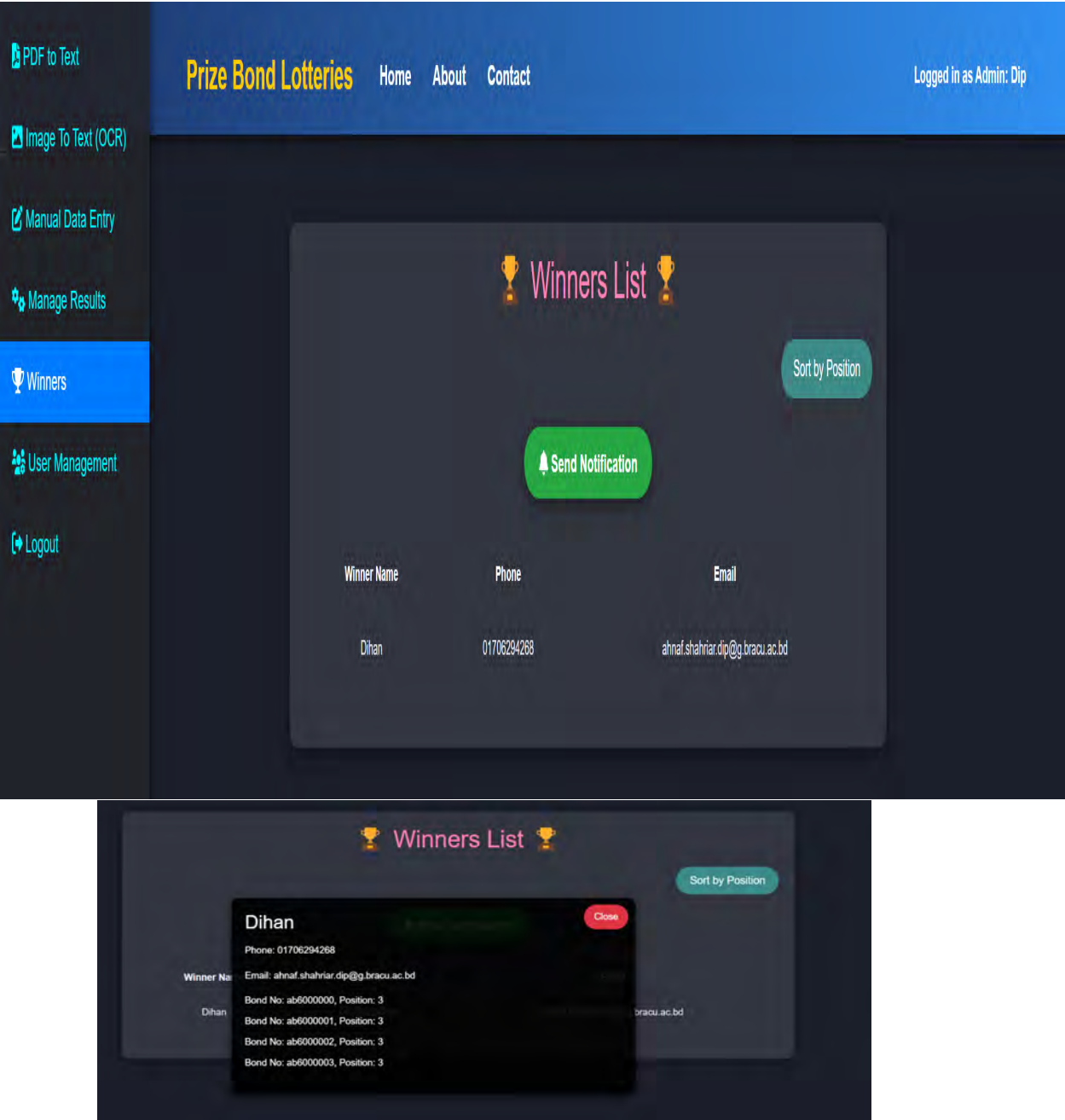
The screenshot shows a web application interface for 'Prize Bond Lotteries'. The user is logged in as 'Dihan'. The 'Check Winner' option is selected in the sidebar. The main content area displays a congratulatory message and a list of winning bond numbers and positions.

Bond Number	Position
ab600000	3
ab600001	3
ab600002	3
ab600003	3

Back to Home

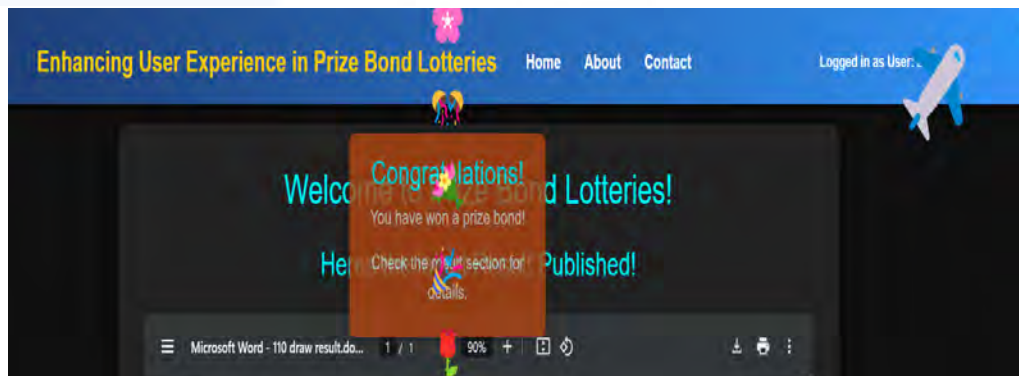
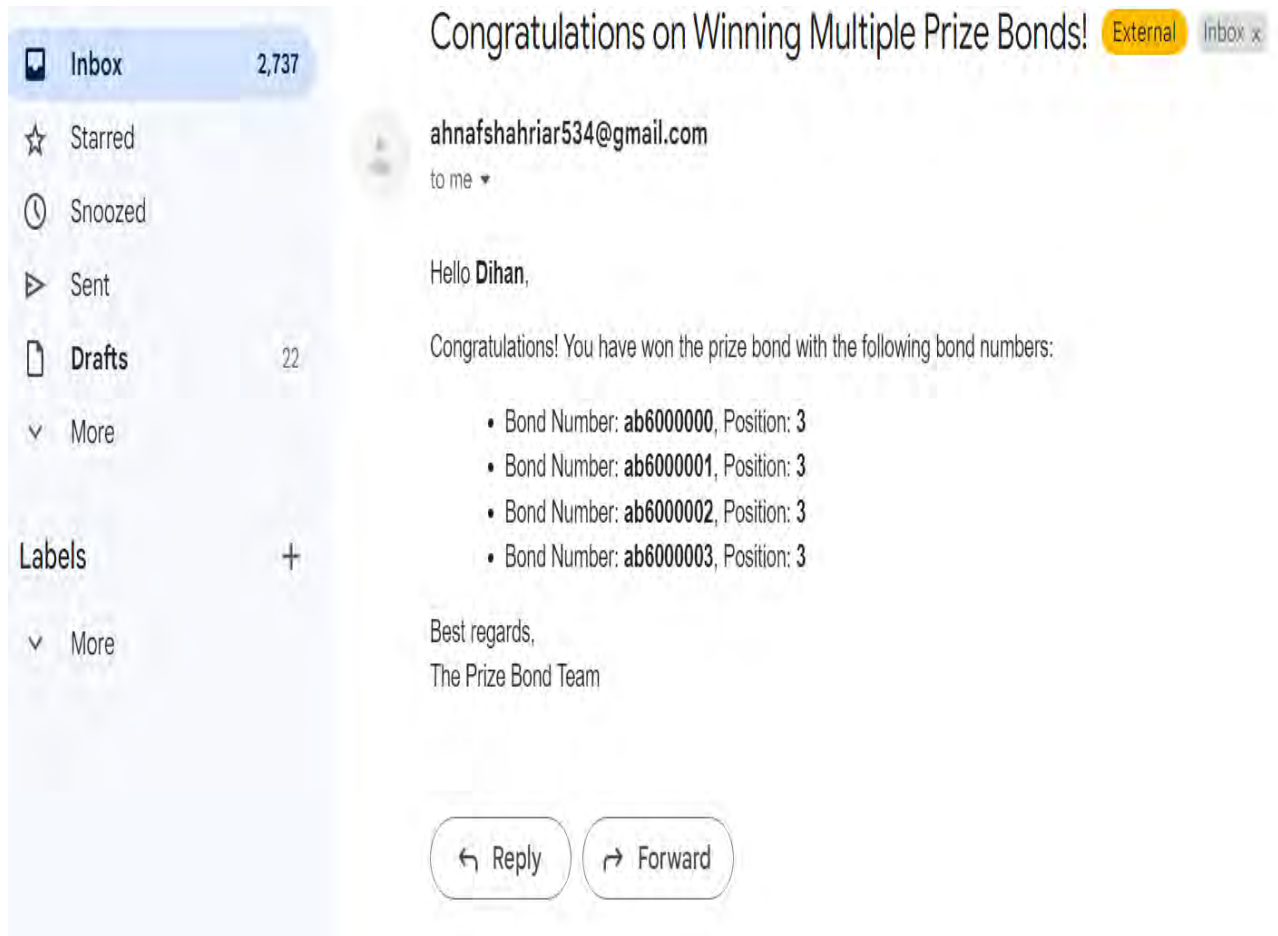
Check Winner by Admin:

Admin can see list of winner. And clicking into the name a modal open with details of winner's bond number and positions. Also admin can send notification to all user by clicking send notification button



Notifications:

There are 2 types of notification system. 1. notification via email 2. When User login they will receive in apps notification.



Chapter 7

Future Work

While our current system has successfully automated the process of prize bond management but there are several areas that remain open for future improvements. One of key limitation is the lack of direct integration with banks. It means users cannot resolve bank-related issues or queries through the platform. Because of that, the system depends on the accuracy of the user-provided data. If incorrect information is entered by the user we have no mechanism to validate or correct it because we don't have all the published bond no. So it may lead to inaccurate results. Furthermore, there are currently no options to buy or sell prize bonds via the website, which could be a useful feature for users. Future developments could focus on addressing these areas such as potentially by integrating bank APIs for direct communication and implementing data validation mechanisms to ensure the accuracy of user submissions. This could be maintained by checking with all released bonds by banks. Lastly, adding a marketplace feature for buying and selling prize bonds. These enhancements would significantly improve the user experience and extend the platform's functionality.

Chapter 8

Summary

Although the growing popularity of prize bonds as an investment option, the traditional method of manually picking winners is time-consuming and subject to human error. To address this, our system leverages Optical Character Recognition (OCR) technology to automate the extraction of prize bond numbers from images and PDFs. Which significantly reduces the time and effort needed for manual verification. By integrating this system with modern web technologies users will be able to store, retrieve, and verify their bond numbers efficiently while receiving real-time updates and notifications about winning bonds through both web and mobile platforms.

The system emphasizes OCR accuracy, robust data security, and scalability. Overall, this ensures precise data extraction, protecting sensitive user information, and managing large volumes of data with ease. Our Key features include automated notifications to winners, real-time data synchronization, and API integration with official winner lists and all are designed to streamline the prize bond management process. Lastly, Future improvements will focus on addressing bank-related queries, enhancing data validation, and adding a marketplace for buying and selling prize bonds. This may further enrich the user experience and expand the system's functionality.

Bibliography

Inbasekaran, A., Gnanasekaran, R. K., and Marciano, R. (2021). Using Transfer Learning to contextually Optimize Optical Character Recognition (OCR) output and perform new Feature Extraction on a digitized cultural and historical dataset. 2021 IEEE International Conference on Big Data (Big Data).

<https://doi.org/10.1109/bigdata52589.2021.9671586>

Guedes, G. B., and Da Silva, A. E. A. (2021). Supervised Learning Approach for Section Title Detection in PDF scientific articles. In Lecture notes in computer science (pp. 4454). <https://doi.org/10.1007/978-3-030-89817-5-3>

Paulson, L. (2005). Building rich web applications with Ajax. Computer, 38(10), 14–17. <https://doi.org/10.1109/mc.2005.330>

Zakas, N. C. (2010). Professional Ajax (2nd ed.). Wrox Press.

Holovaty, A., & Kaplan-Moss, J. (2009). The Definitive Guide to Django: Web Development Done Right. Apress. <https://doi.org/10.1007/978-1-4302-1938-5>

Chen, T., Chen, X., Ma, X., Wang, Y., and Wang, Z. (2022, June 9). Coarsening the granularity: Towards structurally sparse lottery tickets. arXiv.org. <https://arxiv.org/abs/2202.04736>

Katz, R. (2000, September 27). US20070129144A1 - Methods and apparatus for enhanced play in lottery and gaming environments - Google Patents. <https://patents.google.com/patent/US20070129144A1/en>

How- Lottery ticket scanners work — VA Lottery blog. (2020). <https://www.valottery.com/aboutus/blog/how-lottery-ticket-scanners-work>

Forrester, J. (2023, October 26). Machine learning - Challenging The Algorithm Of The Lottery. Embryo.<https://embryo.com/blog/machine-learning-and-the-lottery/>

Mehrotra, A. K., and Musolesi, M. (2020). Intelligent Notification Systems. Synthesis Lectures on Mobile and Pervasive Computing, 11(1), 1–75. <https://doi.org/10.2200/s00965ed1v01y201911mpc014>

Augusto, J. C. (2022). Contexts and Context-awareness Revisited from an Intelligent Environments Perspective. Applied Artificial Intelligence, 36(1). <https://doi.org/10.1080/08839514.2021.2008644>

Oh, Hyungik and Jal, Laleh and Jain, Ramesh. (2015). An intelligent notification system using context from real-time personal activity monitoring. 10.1109/ICME.2015.7177508.

Visuri, A., Van Berkel, N., Okoshi, T., Gonçalves, J., and Kostakos, V. (2019). Understanding smartphone notifications' user interactions and content importance.

International Journal of Human-Computer Studies, 128, 72–85.
<https://doi.org/10.1016/j.ijhcs.2019.03.001>

Bd, P.C.B.P.O.P.B.C.I.(n.d.). Prizebond Checker BD - Prizebond - Online Prize Bond Checker in bd. Prizebond Checker BD-Prizebond - Online Prize Bond Checker in Bd. <https://prizebondcheckerbd.com/>

Rahman,S.H.a.A.(2022, March 16). Prize Bonds: Tk 35 crore winnings remain unclaimed. The Daily Star.
<https://www.thedailystar.net/business/economy/banks/news/tk-35cr-winnings-remain-unclaimed-2982411>

Bangladeshi Prize Bond Checker's Play Store App Ranking and Store Data from data.ai. (n.d.)
<https://www.data.ai/apps/google-play/app/com.technobd.prizebond/Bangladesh> Bank. (n.d.). <https://www.bb.org.bd/en/index.php/Investfacility/prizebond>