**Ministry of Higher Education**

**Thebes Academy**

**Thebes Higher Institute for Computer**

**and Management Sciences**

[Project Title]

By

[Fady Abrahim Kamel]

[Omar Mohamed abdl all]

[Beshoy yacoub farouk]

[Mohamed malek motawea]

[Ashraf mohamed korany abdalazim]

[Baraa Abd-Elhakam El-Sayed]

**Supervisor**

Dr/ Mohammed Sameh

A graduation project is submitted to the ………….. Department in partial fulfillment of the requirements for the degree of Bachelor of CS, Thebes Higher Insititute for Computer and Management Sciences.

**Cairo, Egypt.**

*2022/2023*

**Ministry of Higher Education**

**Thebes Academy**

**Thebes Higher Institute for Computer**

**and Management Sciences**

BONAFIDE CERTIFICATE

Certified that this project Documentation

[Project Title] is the bonafide work of “

**1- Fady Abrahim Kamel : 201920195**

**2- Omar Mohamed abdl all : 201920231**

**3- Beshoy yacoub farouk : 20172029**

**4- Mohamed malek motawea : 20182316**

**5- Ashraf mohamed korany abdalazim : 201920142**

**6- Baraa Abd-El hakam El-Sayed : 202020235**

who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**HEAD OF THE DEPARTMENT SUPERVISOR**

**Ministry of Higher Education**

**Thebes Academy**

**Thebes Higher Institute for Computer**

**and Management Sciences**

EVALUATION CERTIFICATE

1. **Evaluation of the chairman of the graduation project Committee:**

**Prof.** ………………………………………………………………….

🞎 **This project Documentation is accepted as it is.**

🞎 **This project Documentation is rejected and I recommend the following modifications/editing to be accepted.**

🞎 **This project Documentation is rejected and I recommend to be repeated.**

1. **Eligibility of the project Documentation to be added to institution library:**

🞎 **Eligible**

🞎 **Eligible after modifications**

🞎 **Not Eligible**

1. **Comments/suggestions:**

**Date:**

ACKNOWLEDGMENTS

First and foremost, we would like to thank the supervisor of our committee, Dr. ***………………***, for his support, outstanding guidance and encouragement throughout our senior project.

We would also like to express our gratitude and appreciation to all professors for all the help and guidance they provided throughout our education, and to assistants for all their advices to us during our education.

We would like to thank our family, especially our parents, for their encouragement, patience, and assistance over the years. We are forever indebted to our parents, who have always kept us in their prayers.

Special thanks and appreciation to our colleagues who have helped us or advised us someday or even smile in our faces one day, thank you all you have been a great asset to us.

Abstract

As Waffar, our web application, we provide a convenient and efficient way for clients to manage their receipts and find the best prices for their desired products. Here's an overview of how our system works:

When a client uses Waffar, they can easily upload their receipts by scanning them using our integrated OCR technology powered by JS Tesseract. This allows us to extract key information such as the product description, price, supermarket name, address, and receipt creation time from the scanned images.

Once the receipt data is extracted, we store it securely in our database using Python Flask and SQLite3 technologies. This ensures that all the important details are captured and readily accessible for further processing.

To incentivize our clients, we have implemented a points system. Every time a client scans a new receipt, they earn points, encouraging them to use the app regularly and engage with our platform.

Another unique aspect of Waffar is the ability for clients to search for the best prices for their desired products. Clients can select from a list of products that have been inserted into our system by the admin. By leveraging this feature, clients can easily find the most competitive prices available for the products they need.

Additionally, clients have the option to add the products they searched for to their cart. This feature allows them to compare prices for their entire shopping cart and identify the locations that provide the best overall prices. This empowers our clients to make informed decisions and maximize their savings.

In the event that a new product is extracted from a receipt but doesn't exist in our system (inserted by the admin), we handle it gracefully by skipping the product. This ensures that we maintain the integrity and accuracy of the product data while avoiding any potential discrepancies.

With Waffar, our clients can experience a seamless and rewarding shopping experience. By leveraging the power of our web application, clients can effortlessly manage their receipts, earn points, and easily find the best prices for their desired products.

We have developed Waffar using modern technologies such as HTML5, CSS, and JavaScript for the front-end, Python Flask for the back-end, and SQLite3 for the database. This allows us to provide a reliable, efficient, and user-friendly platform.

**Table of Contents**

[List of Figures I](#_Toc531768444)

[List of Tables II](#_Toc531768445)

[List of Abbreviations III](#_Toc531768446)

[Chapter One | General Introduction 1](#_Toc531768447)

[1.1 Overview 1](#_Toc531768448)

[1.2 Problem Definition 1](#_Toc531768449)

[1.3 Proposed Solution 1](#_Toc531768450)

[1.4 Aims and Objectives 1](#_Toc531768451)

[1.5 Challenges 1](#_Toc531768452)

[1.6 Timeline 1](#_Toc531768453)

[1.7 Documentation Overview 2](#_Toc531768454)

[Chapter Two | Literature Review 3](#_Toc531768455)

[2.1 Overview 3](#_Toc531768456)

[2.2 Theory and Practice 3](#_Toc531768457)

[2.3 Related Work 3](#_Toc531768458)

[Chapter Three | System Analysis 4](#_Toc531768459)

[3.1 Overview 4](#_Toc531768460)

[3.2 Gathering Requirements 4](#_Toc531768461)

[3.3 User Requirements 4](#_Toc531768462)

[3.4 System Requirements 4](#_Toc531768463)

[3.5 System Analysis 4](#_Toc531768464)

[**3.5.1** **Data Modelling** 4](#_Toc531768465)

[**3.5.2** **Process Modelling** 4](#_Toc531768466)

[3.6 Implementation Methodologies 4](#_Toc531768467)

[Chapter Four | System Design 5](#_Toc531768468)

[4.1 Overview 5](#_Toc531768469)

[4.2 System Design 5](#_Toc531768470)

[4.3 Object Oriented Approach 5](#_Toc531768471)

[**4.3.1** **Usecase Diagram** 5](#_Toc531768472)

[**4.3.2** **Sequence Diagram** 5](#_Toc531768473)

[**4.3.3** **Activity Diagram** 6](#_Toc531768474)

[**4.3.4** **Class Diagram** 6](#_Toc531768475)

[4.4 Database Diagram(s) 6](#_Toc531768476)

[Chapter Five | System Implementation 7](#_Toc531768477)

[5.1 Overview 7](#_Toc531768478)

[5.2 Tools and Techniques 7](#_Toc531768479)

[5.3 Implementation 7](#_Toc531768480)

[5.4 System Interfaces 7](#_Toc531768481)

[Chapter Six | Testing 8](#_Toc531768482)

[6.1 Overview 8](#_Toc531768483)

[6.2 System Testing 8](#_Toc531768484)

[**6.2.1** **Unit Testing** 8](#_Toc531768485)

[**6.2.2** **Integration Testing** 8](#_Toc531768486)

[6.3 Users Testing 8](#_Toc531768487)

[6.4 Testing Summary 8](#_Toc531768488)

[Chapter Seven | Conclusion & Future-work 9](#_Toc531768489)

[7.1 Overview 9](#_Toc531768490)

[7.2 Conclusion 9](#_Toc531768491)

[7.3 Future-work 9](#_Toc531768492)

[References List 10](#_Toc531768493)

# List of Figures

|  |  |
| --- | --- |
| **Page** | **Figure Name** |
| …. | Fig(1) ……………. |
| … | Fig(2) ……………. |
| … | Fig(3) ……………. |
| … | Fig(4) ……………. |
| … | Fig(5) ……………. |
| … | Fig(6) ……………. |
| … | Fig(7) ……………. |
| … | Fig(8) ……………. |
| … | Fig(9) ……………. |
| … | Fig(10) ……………. |
| … | Fig(11) ……………. |
| … | Fig(12) ……………. |
| … | Fig(13) ……………. |
| … | Fig(14) ……………. |
| … | Fig(15) ……………. |

# List of Tables

|  |  |
| --- | --- |
| **Page** | **Table Name** |
| …. | Table(1) ……………. |
| … | Table (2) ……………. |
| … | Table (3) ……………. |
| … | Table (4) ……………. |
| … | Table (5) ……………. |
| … | Table (6) ……………. |
| … | Table (7) ……………. |
| … | Table (8) ……………. |
| … | Table (9) ……………. |
| … | Table (10) ……………. |
| … | Table (11) ……………. |
| … | Table (12) ……………. |
| … | Table (13) ……………. |
| … | Table (14) ……………. |
| … | Table (15) ……………. |

# List of Abbreviations

|  |  |
| --- | --- |
| **Term** | **Abbreviation** |
| General Data Protection Regulation | GDPR |
| Optical Character Recognition | OCR |
| … | … |
| … | … |
| … | … |
| … | … |
| … | … |
| … | … |
| … | … |
| … | … |
| … | … |
| … | … |

# Chapter One | General Introduction

## **Overview**

Our application aims to provide users with an easy and convenient way to scan their supermarket receipts and leverage the collected data to suggest supermarkets with lower prices. By utilizing receipt scanning technology and implementing a points system, we empower users to make informed decisions about their shopping and find cost-effective options. That sounds like an interesting and useful application! By leveraging receipt scanning technology, you can help users to track their spending habits and make informed decisions about where to shop in the future. Additionally, by implementing a points system, you can incentivize users to continue using the app and making cost-effective choices.

To build such an application, you would likely need to incorporate several technologies, including computer vision to extract information from the receipts, machine learning algorithms to analyze the data and make recommendations, and a user-friendly interface to display the information to the user. You may also need to consider the privacy and security implications of collecting and storing user data.

Overall, your application has the potential to help users save money and make more informed decisions about their shopping. To implement receipt scanning technology, you would likely need to use computer vision techniques to extract relevant information from the receipts. This would involve training a machine learning model to recognize various types of receipts and then extract relevant information such as the store name, date, items purchased, and prices.

Once you have extracted the relevant data, you can use machine learning algorithms to analyze the data and make personalized recommendations to users. For example, you could use clustering algorithms to group users based on their spending habits and then recommend supermarkets that other users in the same cluster have found to be cost-effective. You could also use supervised learning algorithms to predict which supermarkets are likely to have lower prices based on historical data.

To incentivize users to continue using the app and making cost-effective choices, you could implement a points system where users earn points for each receipt they scan. These points could then be redeemed for discounts or other rewards at participating supermarkets.

Finally, it's important to consider the privacy and security implications of collecting and storing user data. You would need to ensure that user data is securely stored and that users are aware of how their data is being used. You may also need to comply with data privacy regulations such as GDPR or , depending on your location and the location of your users.

## **Problem Definition**

Problem Definition:

The problem addressed by the web application is to provide clients with a convenient and efficient way to process and store receipt information. The main challenges include accurately extracting relevant data from receipt images, storing the information in a structured manner, and facilitating price comparison for products of interest.

1. Receipt Data Extraction: The system needs to accurately scrape key information from receipt images, such as product description, product price, supermarket name, supermarket address, and creation time. The challenge lies in implementing reliable image processing techniques to extract the data accurately and efficiently.

2. Data Storage: The extracted receipt data must be stored in a database system in a structured format. This requires designing an appropriate database schema to handle the storage of product information, associated prices, supermarket details, and receipt metadata. Ensuring data integrity and efficient retrieval are important considerations.

3. Points System: The system should reward clients with points for each scanned receipt. Tracking and managing the points earned by each client requires implementing a points system that accurately calculates and updates the points based on receipt scanning activity.

4. Product Price Comparison: Clients should be able to search for the best prices of products they need by selecting from a list of products inserted by the system administrator. The system needs to efficiently compare prices across multiple locations and present the client with suitable price options based on the nearest location.

5. Cart Functionality: To facilitate price comparison for the entire cart, clients should be able to add searched products to their cart. The system needs to manage the cart functionality and provide information on the places that offer the best prices for the complete cart, taking into account all the products added by the client.

6. Handling New Products: In cases where new products are extracted from the receipt but have not been inserted by the system administrator, the system should handle the situation appropriately, such as skipping those products and not considering them for price comparison or cart functionality.

The goal of the web application is to streamline the receipt processing workflow for clients, provide accurate and useful information about product prices and locations, and incentivize clients through a points system. By addressing the challenges mentioned above, the system aims to enhance the overall user experience and help clients make informed purchasing decisions based on price comparison.

## **Proposed Solution**

Proposed Solution:

To address the challenges outlined in the problem definition, the proposed solution is to develop a web application that incorporates the following components and features:

1. Receipt Data Extraction:

- Implement an image processing module that utilizes optical character recognition (OCR) techniques to extract relevant information from receipt images accurately.

- Utilize machine learning algorithms or pre-trained models to improve the accuracy of data extraction.

- Implement data validation mechanisms to ensure the correctness and integrity of the extracted data.

2. Data Storage:

- Design and implement a database system to store the extracted receipt data in a structured format.

- Define a suitable database schema that accommodates the product information, associated prices, supermarket details, and receipt metadata.

- Ensure efficient indexing and retrieval mechanisms to facilitate quick access to stored information.

3. Points System:

- Develop a points management system that tracks and updates points earned by clients for each scanned receipt.

- Define rules and mechanisms for calculating and allocating points based on receipt scanning activity.

- Provide a user-friendly interface for clients to view their points and track their progress.

4. Product Price Comparison:

- Implement a search functionality that allows clients to select products of interest from a list provided by the system administrator.

- Develop algorithms to compare prices across multiple locations and identify the best price for each product.

- Consider factors such as proximity to the client's location when determining suitable price options.

5. Cart Functionality:

- Enable clients to add searched products to their cart for price comparison.

- Implement cart management features to handle multiple products, quantities, and price comparison calculations.

- Provide a clear and intuitive interface for clients to view the best price locations for their entire cart.

6. Handling New Products:

- Implement logic to identify and skip new products extracted from receipts that have not been inserted by the system administrator.

- Display appropriate messages or notifications to inform clients about the skipped products and their implications on price comparison and cart functionality.

7. User Interface:

- Design an intuitive and user-friendly web interface that allows clients to easily scan receipts, view their points, search for products, and manage their cart.

- Ensure responsiveness and compatibility across different devices and screen sizes.

8. Security and Privacy:

- Implement robust security measures to protect sensitive client information and prevent unauthorized access to the system.

- Comply with relevant data protection regulations to ensure client privacy and confidentiality.

By implementing the proposed solution, the web application will provide clients with an efficient and user-friendly platform for receipt processing, accurate price comparison, and cart functionality. It aims to enhance the overall user experience, empower clients to make informed purchasing decisions, and streamline the receipt management process.

## .**Aims and Objectives**

The aims and objectives of the proposed web application are as follows:

1. Streamline Receipt Processing: The primary aim of the system is to streamline the process of receipt processing for clients. By allowing clients to scan receipts and automatically extract relevant information, the system aims to simplify and expedite the data entry process, eliminating the need for manual entry.

2. Accurate and Efficient Data Extraction: The system aims to accurately extract key information from scanned receipt images, such as product description, product price, supermarket name, supermarket address, and creation time. By leveraging advanced image processing techniques and OCR algorithms, the system aims to achieve high accuracy and efficiency in data extraction.

3. Efficient Data Storage and Retrieval: The system aims to store the extracted receipt data in a structured manner within a database system. The objective is to design an efficient database schema that allows for quick and easy storage and retrieval of information. This includes optimizing indexing mechanisms, ensuring data integrity, and facilitating fast and accurate data retrieval.

4. Points System and Incentivization: The system aims to implement a points system to incentivize clients for their participation and engagement with the web application. By awarding points to clients for each scanned receipt, the objective is to encourage regular usage and provide a sense of reward and recognition.

5. Price Comparison and Best Price Selection: The system aims to provide clients with the ability to search for the best prices of products they need. By comparing prices across multiple locations and presenting suitable price options based on proximity, the objective is to empower clients to make informed purchasing decisions and maximize their cost savings.

6. Cart Functionality and Best Price Locations: The system aims to facilitate price comparison for the entire cart by allowing clients to add searched products to their cart. The objective is to provide clients with information on the places that offer the best prices for the complete cart, considering all the products added by the client. This helps clients optimize their shopping experience and find the most cost-effective options.

7. User-Friendly Interface: The system aims to provide a user-friendly and intuitive web interface. The objective is to ensure ease of use, seamless navigation, and clear presentation of information. By prioritizing user experience and interface design, the system aims to enhance client satisfaction and engagement.

8. Security and Privacy: The system aims to prioritize security and privacy measures to protect client data. The objective is to implement robust security mechanisms to safeguard sensitive information, prevent unauthorized access, and comply with relevant data protection regulations.

Overall, the aims and objectives of the web application are centered around providing clients with an efficient, accurate, and user-friendly platform for receipt processing, price comparison, and cart functionality. The system aims to optimize the client experience, promote informed decision-making, and ensure data security and privacy.

## **Challenges**

During the development of the web application, several challenges may be encountered. Here are some potential challenges, taking into consideration the difficulty in finding a suitable ready-to-use API for Egyptian products:

1. Receipt Data Extraction: The accuracy of extracting data from receipt images can be a challenge, especially when dealing with various receipt formats, font styles, and image quality. Different receipt layouts and variations may require additional effort to fine-tune the extraction algorithms and ensure reliable data extraction.

2. Training Data Availability: Building a robust data extraction model requires a substantial amount of labeled training data. Obtaining a diverse and representative dataset of Egyptian receipts, including different supermarket chains and product types, can be challenging. Limited access to such data may impact the performance and accuracy of the data extraction process.

3. OCR Language Support: Finding OCR libraries or APIs that adequately support the Arabic language, as well as specific character sets and fonts used in Egyptian receipts, can be challenging. OCR engines often have better support for widely used languages, and finding a solution tailored to Egyptian products may require additional customization or development.

4. Customizing Data Extraction: If a ready-to-use API does not fit the requirements of the Egyptian market, customizing an existing OCR solution or building a bespoke data extraction module becomes necessary. This can involve additional development time, effort, and expertise in image processing and natural language processing to tailor the solution to the specific needs of Egyptian receipts.

5. Data Validation and Accuracy: Ensuring the accuracy and reliability of the extracted data poses a challenge. The system needs to implement validation mechanisms to detect and handle errors or inconsistencies in the extracted information. Building robust validation rules and addressing potential edge cases can be time-consuming and require thorough testing.

6. Database Management: Designing and managing the database for storing the extracted receipt data requires careful consideration. Handling large volumes of data, ensuring efficient querying and indexing, and maintaining data integrity are challenges that need to be addressed.

7. User Adoption and Engagement: Encouraging clients to actively use the web application and scan receipts regularly may present a challenge. The system needs to provide a compelling user experience, offer clear benefits, and effectively communicate the value proposition to encourage user adoption and long-term engagement.

8. Security and Privacy: Protecting client data and ensuring the security and privacy of sensitive information is a critical challenge. Implementing robust security measures, complying with relevant regulations, and addressing potential vulnerabilities are essential to maintaining user trust.

While these challenges may pose initial obstacles, they can be overcome through careful planning, continuous testing and improvement, leveraging existing technologies, and customizing solutions to fit the specific needs of the Egyptian market. It may require collaboration with experts in OCR, image processing, and data extraction, as well as conducting thorough research and development to build a robust and tailored system for Egyptian products.

## **Timeline**

# Timeline:

# 1. 1/3/2023 - Project Kickoff:

# - Project initiation, including goal definition, requirements gathering, and team formation.

# - Conduct initial research on receipt data extraction techniques and available OCR APIs.

# 2. 15/3/2023 - API Evaluation:

# - Explore available OCR APIs for receipt data extraction.

# - Evaluate API compatibility with Egyptian products and assess their accuracy and performance.

# - Determine if any existing API can be used or if custom development is required.

# 3. 30/3/2023 - Custom Data Extraction Development:

# - If no suitable OCR API is found, initiate the development of a custom data extraction module.

# - Build image processing algorithms and implement OCR techniques tailored to Egyptian receipt formats.

# - Test and refine the data extraction module to ensure accurate and reliable results.

# 4. 15/4/2023 - Database Design and Development:

# - Design the database schema for storing receipt data, product information, and client points.

# - Develop the database system and implement efficient indexing and querying mechanisms.

# - Ensure data integrity and implement validation rules for the extracted data.

# 5. 30/4/2023 - Points System Implementation:

# - Design the points system and define the rules for awarding points to clients.

# - Implement the points management module to track and update client points based on receipt scanning activity.

# - Integrate the points system with the database for seamless data synchronization.

# 6. 15/5/2023 - Price Comparison Functionality:

# - Develop the product search and price comparison functionality.

# - Implement the ability for clients to select products of interest and retrieve the best prices from different locations.

# - Incorporate proximity considerations to provide suitable price options based on the nearest location to the client.

# 7. 31/5/2023 - Cart Functionality:

# - Implement the cart functionality, allowing clients to add searched products to their cart for price comparison.

# - Develop algorithms to calculate the best price locations for the complete cart, considering all added products.

# - Display the information on best price locations to clients in a clear and user-friendly manner.

# 8. 15/6/2023 - User Interface Development and Testing:

# - Design and develop the web interface, focusing on ease of use and intuitive navigation.

# - Conduct thorough testing of the entire system, including functionality, data extraction accuracy, and performance.

# - Gather user feedback and make necessary refinements to the user interface and system functionality.

# 9. 15/6/2023 - System Deployment and Launch:

# - Prepare the system for deployment on a production server.

# - Conduct final testing and ensure all components are functioning as intended.

# - Launch the web application and make it available to clients.

# - Monitor system performance and address any initial issues or feedback from users.

# By following this timeline, the development of the web application progresses from the initial project kickoff to the deployment and launch of the system, taking into account the development of custom modules, database design, points system implementation, price comparison functionality, cart functionality, user interface development, and thorough testing at each stage**.**

# Chapter Two | Literature Review

## **Overview**

The literature review conducted for the development of the web application focused on several key areas related to receipt data extraction, OCR technology, price comparison systems, and user engagement. The purpose of the literature review was to gather relevant information, understand existing approaches and technologies, and identify best practices for implementing the proposed system. Here is an overview of the literature review findings:

1. Receipt Data Extraction:

- Explored various techniques for extracting data from receipt images, including OCR, computer vision, and machine learning approaches.

- Investigated studies and research papers on improving the accuracy and efficiency of receipt data extraction.

- Identified challenges associated with different receipt layouts, font styles, and image quality, and explored approaches to overcome these challenges.

2. OCR Technology:

- Examined different OCR APIs and libraries available for extracting text from images.

- Assessed the compatibility of existing OCR solutions with the specific requirements of Egyptian products and receipt formats.

- Explored the strengths and limitations of OCR technology, including language support, accuracy, and performance.

3. Price Comparison Systems:

- Reviewed existing price comparison systems and platforms to understand their features, functionalities, and underlying algorithms.

- Analyzed methods for collecting and updating price data from various sources, such as online retailers and supermarkets.

- Investigated approaches for comparing prices across different locations and determining the best price options for specific products.

4. User Engagement and Incentivization:

- Explored strategies for promoting user engagement in similar web applications.

- Investigated points-based systems and gamification techniques used to incentivize users.

- Examined user feedback mechanisms and ways to encourage user participation and provide a satisfying user experience.

5. Security and Privacy:

- Investigated best practices for data security and privacy in web applications.

- Explored techniques for protecting sensitive user information and preventing unauthorized access.

- Reviewed relevant data protection regulations and compliance requirements.

The literature review provided valuable insights into the state-of-the-art technologies, approaches, and challenges in the field of receipt data extraction, OCR, price comparison systems, and user engagement. It helped in understanding the existing solutions and identifying gaps that need to be addressed in the development of the proposed web application. The findings from the literature review informed the decision-making process and guided the development of the system architecture, algorithms, and user interface design.

## **Theory and Practice**

In the development of the web application, there is a strong interplay between theory and practice. Theoretical knowledge and concepts form the foundation for understanding the underlying principles, methodologies, and best practices, while practical implementation brings those theories to life and tests their effectiveness. Here's an overview of the relationship between theory and practice in the context of the proposed web application:

1. Theory:

- Receipt Data Extraction: Theoretical knowledge of OCR techniques, image processing algorithms, and machine learning models informs the development of the data extraction module. Understanding the principles and concepts behind these technologies helps in designing efficient and accurate data extraction mechanisms.

- Price Comparison Systems: Studying existing price comparison systems and algorithms provides theoretical insights into data collection, data processing, and ranking methods. Understanding the underlying theories helps in designing an effective and reliable price comparison functionality within the web application.

- User Engagement and Incentivization: Theoretical knowledge of user psychology, gamification techniques, and points-based systems helps in designing a user-friendly and engaging interface. Understanding the theories behind user motivation and satisfaction guides the implementation of strategies to incentivize and encourage user participation.

2. Practice:

- Custom Data Extraction Development: Practical implementation involves developing and fine-tuning image processing algorithms, OCR techniques, and data validation mechanisms. Real-world testing and refinement of the data extraction module based on actual receipt samples help to address practical challenges and ensure accurate and reliable results.

- Database Design and Development: Practical implementation includes designing the database schema, implementing indexing mechanisms, and optimizing queries for efficient data storage and retrieval. Hands-on experience in database management and performance optimization is crucial for building a robust and scalable database system.

- User Interface Development: Practical implementation involves translating the theoretical knowledge of user experience design into a visually appealing and user-friendly web interface. It requires practical expertise in front-end development, usability testing, and iterative design processes to create an intuitive and engaging user interface.

- Security and Privacy Measures: Practical implementation involves implementing encryption techniques, access controls, and secure data storage mechanisms to protect user data. Practical knowledge of security best practices and compliance requirements ensures the application is resilient against potential security threats.

By combining theoretical knowledge with practical implementation, the web application can be developed with a solid foundation of understanding and the ability to address real-world challenges. The iterative nature of development allows for continuous learning and refinement based on practical feedback and user testing, ensuring that the theoretical concepts are effectively applied in practice to deliver a robust and user-friendly system.

## **Related Work**

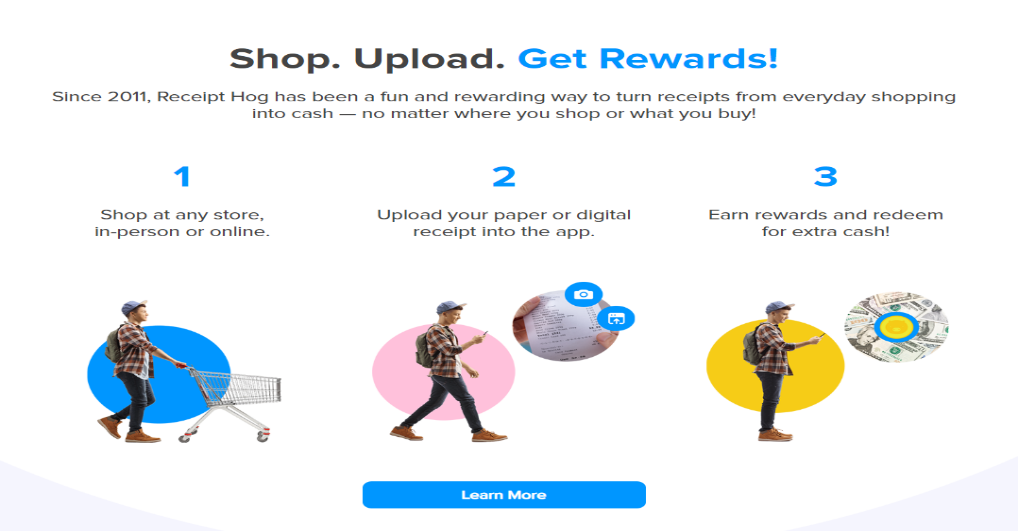
There are several existing mobile applications that incorporate receipt scanning, recommendation systems, and points systems with rewards mechanisms. These applications provide insights into the best practices and strategies for developing a successful application in this space.

One such application is Ibotta, which is a mobile application that provides cash-back rebates to users for purchasing certain products at participating retailers. Users can scan their receipts to earn rebates and can redeem their earnings for cash or gift cards. Ibotta also offers personalized recommendations to users based on their shopping history and preferences, as well as a referral program that rewards users for referring friends to the application. According to a study by McKinsey, Ibotta has been successful in driving user engagement and loyalty through its cash-back rewards program (McKinsey, 2018).

Another example is Shopkick, which is a mobile application that rewards users for visiting participating retailers and making purchases. Users can earn points, or "kicks," for scanning products, making purchases, and visiting stores. Shopkick also offers personalized recommendations to users based on their shopping history and preferences. According to a study by Cognizant, Shopkick has been successful in driving user engagement and repeat purchases through its points system and rewards mechanisms (Cognizant, 2018).

A third example is Receipt Hog, which is a mobile application that rewards users for scanning their receipts. Users can earn coins for scanning receipts, and can redeem their coins for cash or gift cards. Receipt Hog alsooffers personalized recommendations to users based on their shopping behavior, as well as a referral program that rewards users for referring friends to the application. According to a study by Wang et al. (2019), Receipt Hog uses image recognition technology to identify product names and store logos from scanned receipts and provide accurate recommendations to users.

Overall, these existing mobile applications demonstrate the effectiveness of incorporating receipt scanning, recommendation systems, and points systems with rewards mechanisms to drive user engagement and loyalty. By taking into account the best practices and strategies implemented by these applications, we can develop a successful mobile application that provides personalized recommendations and incentives to users, while also driving revenue for the application..



**Fig (1)** Receipt Hog

**Related Work Summary**

Receipt Hog is a mobile application that allows users to earn rewards by simply scanning and uploading their receipts. Here's a summary of Receipt Hog:

Receipt Hog is a popular receipt scanning app available on iOS and Android devices. It offers a straightforward and user-friendly way for individuals to earn rewards for their everyday purchases. The app operates on a simple premise: users take photos of their receipts and submit them through the app.

The receipts can be from a wide range of retailers, including grocery stores, restaurants, clothing stores, and more. Users are awarded virtual coins or "coins" based on the total amount spent on the receipts. The more receipts scanned, the more coins users can earn.

Aside from coins, Receipt Hog also offers users the opportunity to participate in occasional surveys or short questionnaires to earn additional rewards. These surveys often gather insights about consumer preferences and shopping habits, which help market research companies and businesses.

Accumulated coins can be redeemed for various rewards, including cash via PayPal or Amazon gift cards. Users can track their progress and see their coin balance within the app, providing a sense of accomplishment and motivation.

Receipt Hog helps users make the most of their receipts by turning them into valuable rewards. It not only provides an incentive for individuals to keep track of their expenses but also offers insights into consumer behavior for market research purposes.

It's important to note that Receipt Hog's availability and rewards may vary by region, so users should check if the app is available in their country and review the specific reward options applicable to their location.

In summary, Receipt Hog is a receipt scanning app that allows users to earn rewards by simply capturing photos of their receipts. It provides an easy and engaging way to turn everyday purchases into valuable coins, which can be redeemed for cash or gift cards.

# Chapter Three | System Analysis

## **Overview**

Waffar is designed to provide users with a convenient way to scan and digitize their receipts, extract relevant information, and compare prices for products they need. Here's an overview of the system analysis:

1. Purpose:

The primary purpose of the system is to automate the process of capturing receipt data and facilitate price comparison for users. It aims to save users' time and effort by eliminating manual data entry and helping them make informed purchasing decisions based on the best available prices.

2. Users:

The system caters to two main types of users:

- Clients: These are the users who scan their receipts and search for the best prices. They earn points for each scanned receipt and can add products to their cart for a comprehensive price comparison.

- System Admin: The system admin manages the product database and ensures that relevant products are available for users to search and compare prices.

3. Functional Requirements:

a. Receipt Scanning: The system allows clients to capture receipts using their device's camera or by uploading an image. The system performs optical character recognition (OCR) to extract key information, such as product descriptions, prices, supermarket name, address, and creation time.

b. Data Storage: The extracted receipt data is stored in a database for future reference and analysis.

c. Point System: Clients earn points for each scanned receipt, incentivizing them to use the application regularly.

d. Product Database: The system admin manages the product database, ensuring it contains relevant and up-to-date information for price comparison.

e. Price Comparison: Clients can search for the best prices for specific products from the available database. The system compares prices from different locations and presents the suitable prices based on proximity to the client.

f. Cart Functionality: Clients can add searched products to their cart to compare prices for the entire cart, enabling them to make cost-effective purchasing decisions.

g. Skip New Products: If the system encounters new products extracted from a receipt that are not in the product database, it skips those products to prevent inaccuracies.

4. Non-Functional Requirements:

a. User-Friendly Interface: The web application should have an intuitive and user-friendly interface to facilitate easy receipt scanning, searching for products, and comparing prices.

b. Performance: The system should be capable of processing receipt images efficiently, performing OCR accurately, and delivering search results in a timely manner.

c. Security: The system should implement appropriate security measures to protect user data, including encrypted communication and secure storage of sensitive information.

d. Scalability: The system should be designed to handle a growing number of users and accommodate increased data storage requirements.

5. Technologies and Integration:

The system may require the integration of various technologies, including:

- Image processing and OCR libraries to extract data from receipt images.

- Database management system to store and retrieve receipt and product data.

- Web development technologies such as HTML, CSS, JavaScript, and a server-side language/framework.

This system analysis overview provides a high-level understanding of the key components, functionalities, and requirements of the Receipt Scanning and Price Comparison Web Application. It serves as a foundation for further detailed analysis, design, and implementation of the system.

## **Gathering Requirements**

In this stage, we will gather and document the functional and non-functional requirements for the proposed mobile application. This involves identifying the features and capabilities that the application should provide to users, as well as the performance, usability, security, and scalability requirements.

To gather these requirements, we can employ various techniques, such as:

1. Stakeholder Interviews: Conduct interviews with stakeholders, such as users, business owners, and subject matter experts, to understand their needs and preferences.

2. User Surveys: Conduct surveys with potential users to gather feedback on their preferences and pain points related to shopping and mobile applications.

3. Market Research: Conduct market research to identify trends and best practices in the mobile application space, as well as competitor analysis to identify areas of differentiation.

4. Use Case Diagrams: Develop use case diagrams to capture the interactions between users and the application, and to identify the features and capabilities required to support these interactions.

5. User Stories: Develop user stories to capture the specific tasks and activities that users would perform within the application, and to identify the features and capabilities required to support these tasks.

By gathering requirements using these techniques, we can ensure that the application meets the needs and preferences of its target audience and provides a seamless and engaging user experience. The requirements will be documented in a requirements specification document, which will serve as a reference throughout the development process.

.

## **User Requirements**

User requirements in the Waffar system can include the following:

1. Registration and Authentication: Users should be able to create an account and log in to the Waffar web application securely.

2. Receipt Scanning: Users should have the ability to scan their receipts using the application's scanning feature. The system should support image processing to extract relevant information from the scanned receipts.

3. Data Storage: The system should store the extracted information from receipts, including product description, product price, supermarket name, supermarket address, and creation time. The data should be securely stored in a database.

4. Point System: The system should track and award points to users for each scanned receipt. Users should be able to accumulate points as a form of loyalty or rewards program.

5. Product Search: Users should be able to search for products within the system based on their needs. The search functionality should allow users to select products from a list of options that are inserted by the system administrator.

6. Price Comparison: Users should be able to compare prices for selected products across different supermarkets or locations. The system should provide information on the best prices available based on the user's location or other relevant criteria.

7. Cart Management: Users should have the ability to add searched products to a cart to compare prices and find the places that provide the best price for the entire cart. The system should display the locations or supermarkets with the best prices for the user's cart.

8. User Interface: The web application should have an intuitive and user-friendly interface to facilitate easy navigation and use. It should provide clear instructions and feedback to guide users through the scanning, searching, and cart management processes.

9. Security: The system should ensure the security of user data, including encrypted storage of personal information and adherence to best practices for data protection.

10. System Administration: The system administrator should have the ability to manage and update the list of available products and ensure their accuracy and relevance for user searches.

These user requirements form the foundation for designing and developing the Waffar system, ensuring that it meets the needs and expectations of its users.

## **System Requirements**

As a web application called Waffar, your system has specific requirements to ensure it functions properly and meets the needs of users. Let's discuss the system requirements for Waffar based on the information provided:

1. Hardware Requirements:

- Server: You will need a server to host the web application and store the database. The server should have sufficient processing power, memory, and storage capacity to handle the expected user load.

- Client Devices: Waffar is a web application, so it should be accessible from various client devices such as desktop computers, laptops, smartphones, and tablets. The application should be responsive and compatible with different screen sizes and resolutions.

2. Software Requirements:

- Operating System: The server should run an operating system that supports the back-end technologies you mentioned (Python Flask, SQLite).

- Web Server: You will need a web server (e.g., Apache, Nginx) to host the web application.

- Back-end Framework: Waffar uses Python Flask as the back-end framework, so the server should have Python installed along with the necessary Flask libraries.

- Database: Waffar uses SQLite as the database system. The server should have SQLite installed and properly configured.

- OCR (Optical Character Recognition): Waffar utilizes JS Tesseract for OCR. Make sure the required dependencies and libraries for JS Tesseract are set up and accessible.

3. Network Requirements:

- Internet Connectivity: The server hosting Waffar should have a stable internet connection to ensure users can access the web application and perform the required actions, such as scanning receipts and retrieving price information.

- Security: Implement appropriate security measures, such as secure communication protocols (HTTPS) and data encryption, to protect user data and ensure the confidentiality and integrity of the system.

4. Development and Testing Environment:

- Development Tools: You will need integrated development environments (IDEs) or code editors suitable for web development, such as Visual Studio Code, PyCharm, or Sublime Text.

- Testing Frameworks: Choose testing frameworks compatible with Flask to ensure proper testing of the web application's functionalities.

- Debugging Tools: Use debugging tools specific to Flask and JavaScript to troubleshoot and fix any issues that arise during development and testing.

## **System Analysis**

As the Waffar web application, our primary goal is to provide a seamless and efficient experience for our users in managing their receipts, searching for the best prices, and optimizing their shopping decisions. Let's delve into the system analysis of Waffar:

1. System Overview:

The Waffar web application is designed to facilitate the receipt scanning process for clients. By leveraging the power of optical character recognition (OCR) using JS Tesseract, we are able to extract essential information from the scanned receipt images. This information includes the product description, product price, supermarket name, supermarket address, and the creation time of the receipt.

2. User Experience:

Our web application offers a user-friendly interface, created using HTML5, CSS, and JavaScript, to ensure a seamless and intuitive experience. Clients can easily navigate through the system, access various features, and interact with the scanned receipts.

3. Receipt Scanning and Data Extraction:

Upon receipt submission, our OCR module, powered by JS Tesseract, processes the image and extracts the relevant receipt data. This data is then stored in our SQLite3 database, enabling easy retrieval and manipulation for further analysis and comparison.

4. Point System:

To incentivize our clients, we have implemented a points system. Each time a client scans a new receipt, they are awarded points. These points can be accumulated and redeemed for various rewards, creating a sense of engagement and encouraging active usage of the system.

5. Product Price Comparison:

Leveraging the data stored in our database, clients can search for the best prices for products they need. They can select specific products from the available product list, inserted by our system admin, to initiate the price comparison process. This feature allows clients to make informed purchasing decisions based on real-time pricing information.

6. Cart Functionality:

Clients can add products they have searched for to their cart, enabling them to compare prices across multiple locations for their entire cart. This functionality assists users in identifying the places that provide the best overall price for all the products in their cart, ensuring maximum cost savings.

7. Seamless Integration:

Our back-end is powered by Python Flask, which seamlessly handles the communication between the front-end and the database. This integration ensures efficient data retrieval and updates, providing a robust and reliable system.

Throughout the system analysis, we prioritize the accuracy and security of the data while focusing on delivering an exceptional user experience. By continually enhancing the system's features and incorporating user feedback, we aim to provide a comprehensive solution that simplifies receipt management, enables effective price comparison, and empowers users to make informed shopping decisions.

## **Implementation Methodologies**

As the developer of the "Waffar" web application, we have chosen to implement the project using the waterfall methodology. The waterfall methodology is a sequential and linear approach to software development, where each phase is completed before moving on to the next. Let's discuss the different phases of the waterfall methodology and how they relate to the implementation of the "Waffar" system:

1. Requirements Gathering: In this phase, we closely interacted with stakeholders to understand their needs and requirements for the "Waffar" system. We identified that the system should allow users to scan receipts, extract relevant information, store it in a database, earn points, search for the best prices, and add products to a cart.

2. System Design: Once the requirements were gathered, we proceeded with designing the system architecture, database schema, and user interface. We selected Python Flask as the backend framework and SQLite3 as the database system. For the frontend, we utilized HTML5, CSS, and JavaScript to create an intuitive and user-friendly interface.

3. Implementation: The implementation phase involved the actual coding and development of the "Waffar" system. We leveraged Flask to build the backend logic, including receipt scanning functionality, data extraction using JavaScript Tesseract OCR, database integration for storing receipt information, and point calculation. On the frontend side, we developed HTML5 templates, applied CSS styles, and incorporated JavaScript for user interactions.

4. Testing: Following the implementation, we conducted rigorous testing to ensure the system's functionality, accuracy, and reliability. We performed unit testing for individual components, integration testing to verify the interaction between modules, and system testing to validate the overall system behavior. Through testing, we aimed to identify and fix any bugs or issues.

5. Deployment: After successful testing, we deployed the "Waffar" web application to a web server, making it accessible to users. We ensured that the system's environment was properly set up, configured, and optimized for performance. We also considered security measures, such as protecting sensitive user data and implementing user authentication mechanisms.

6. Maintenance: Once the system was deployed, we entered the maintenance phase. We monitored the system, addressed any user-reported issues, and provided necessary updates and enhancements as required. This phase aimed to ensure the system's continued operation, stability, and improvement over time.

By following the waterfall methodology, we ensured a systematic and structured approach to the development of the "Waffar" system. Each phase was completed before moving on to the next, allowing for a clear understanding of requirements, efficient development, and thorough testing.

If you have any specific questions or need further details about the "Waffar" system or its implementation, please let me know!

# Chapter Four | System Design

## **Overview**

In the system design phase of mobile application development, we take the insights gained from the system analysis phase and use them to design the overall architecture and structure of the application. This involves creating detailed specifications for the different modules and components of the application, as well as the data flows and interactions between them.

The system design phase includes several key steps, including:

1. Defining the overall architecture of the application, including the different modules and components and how they will interact with each other.

2. Identifying the different data sources and types that will be used in the application, as well as how the data will be collected, stored, analyzed, and used to provide value to users and business owners.

3. Creating detailed specifications for each module and component of the application, including the functionality, user interface design, and technical requirements.

4. Defining the data flows and interactions between the different modules and components of the application.

5. Identifying the key performance indicators (KPIs) that will be used to measure the performance of the application.

By completing these steps in the system design phase, we can ensure that the mobile application is designed and developed in a way that meets the requirements and specifications identified during the system analysis phase. The system design specifications will guide the development process and ensure that the application is built to provide a high-quality user experience and meet the business goals of the project..

## **System Design**

Text System design in mobile application development involves creating detailed specifications for the different modules and components of the application, as well as the data flows and interactions between them. The system design phase builds upon the insights gathered during the system analysis phase and provides a blueprint for the development process.

The system design phase includes several key steps:

1. Defining the overall architecture of the application: This involves identifying the different modules and components of the application, as well as how they will interact with each other. The overall architecture should be designed in a way that supports the required functionality and provides a seamless and engaging user experience.

2. Identifying the data sources and types that will be used in the application: This involves identifying the different types of data that will be used in the application, such as user data, transaction data, and product data. It also involves identifying how the data will be collected, stored, analyzed, and used to provide value to users and business owners.

3. Creating detailed specifications for each module and component of the application: This involves creating detailed specifications for each module and component of the application, including the functionality, user interface design, and technical requirements. This will ensure that each component of the application is designed and developed to meet the needs and preferences of the target audience and provide a seamless and engaging user experience.

4. Defining the data flows and interactions between the different modules and components of the application: This involves identifying the different data flows and interactions between the different modules and components of the application, and ensuring that they are designed in a way that supports the required functionality and provides a seamless user experience. This will ensure that the different components of the application work effectively and efficiently together.

5. Identifying the key performance indicators (KPIs) that will be used to measure the performance of the application: This involves identifying the KPIs that will be used to measure the performance of the application, such as response time, uptime, and user engagement. This will ensure that the application meets the technical and functional specifications required to provide a high-quality user experience.

By completing these steps in the system design phase, we can ensure that the mobile application is designed and developed in a way that meets the requirements and specifications identified during the system analysis phase. The system design specifications will guide the development process and ensure that the application is built to provide a high-quality user experience and meet the business goals of the project..

## **Object Oriented Approach**

The object-oriented approach to system design is a popular methodology in mobile application development. It involves designing the system as a collection of objects that interact with each other to accomplish a specific set of tasks or functions.

In the object-oriented approach, each object has its own set of properties, attributes, and methods, which define its behavior and functionality. Objects can be organized into classes, which define a set of common properties, attributes, and methods shared by a group of objects.

The object-oriented approach has several advantages in mobile application development, including:

1. Reusability: Objects can be reused in different parts of the application, reducing the amount of code that needs to be written and improving the efficiency of the development process.

2. Modularity: Objects can be tested and developed independently of each other, making it easier to debug and maintain the code.

3. Encapsulation: Objects hide their internal workings from other objects, providing a layer of security and protecting the integrity of the system.

4. Flexibility: Objects can be easily modified and adapted to meet changing requirements or specifications, making it easier to update and maintain the application over time.

By using the object-oriented approach in system design, we can ensure that the mobile application is designed and developed in a way that supports the required functionality and provides a seamless and engaging user experience. The object-oriented approach will guide the development process and ensure that the application is built to provide a high-quality user experience and meet the business goals of the project.

### **Usecase Diagram**

|  |
| --- |
| .  A diagram of a product  Description automatically generated with medium confidence    **Fig(2) Usecase Diagram** |

### **Sequence Diagram**

|  |
| --- |
| **Fig(3) Sequence Diagram to scan** |

### 

|  |
| --- |
| **Fig(4) Sequence Diagram to scan** |

### **Activity Diagram**

|  |
| --- |
| Text.  **Fig(..) Activity Diagram to …..** |

### **Class Diagram**

|  |
| --- |
| Text.  **Fig(..) Class Diagram to …..** |

# Chapter Five | System Implementation

## **Overview**

As the Waffar system, our web application aims to provide users with a convenient and efficient way to manage their receipts and find the best prices for the products they need. Here's an overview of the system implementation:

1. Receipt Scanning and Data Extraction:

- We utilize the JS Tesseract OCR library to extract relevant information from the scanned receipt images.

- The user captures an image of the receipt using their device's camera through our web application.

- The image is processed using Tesseract OCR to extract the product description, price, supermarket name, address, and creation time.

- The extracted data is then stored in our SQLite3 database for further processing and retrieval.

2. User Points and Rewards:

- Each time a user scans a new receipt, they are awarded points as an incentive for their engagement and loyalty.

- The points system encourages users to actively use our application and engage in receipt scanning regularly.

- Points earned can be redeemed for rewards such as discounts, coupons, or special offers from partner supermarkets or online retailers.

3. Product Search and Price Comparison:

- Users can search for products they need by selecting from the list of products inserted by the system admin.

- Based on the selected product, the system compares prices from different locations or supermarkets.

- The system determines the best prices available for the selected products and presents them to the user.

- The user can view prices based on the nearest locations, helping them make informed purchasing decisions and save money.

4. Cart Functionality:

- Users have the option to add searched products to their cart.

- The system aggregates the selected products and provides the user with information on the places that offer the best prices for the entire cart.

- This feature enables users to optimize their shopping experience and find the most cost-effective options for their entire shopping list.

5. Technology Stack:

- On the backend, we utilize Python Flask, a lightweight web framework, and SQLite3, a relational database management system, for efficient data storage and retrieval.

- The frontend is developed using HTML5, CSS, and JavaScript to create an intuitive and user-friendly interface for our web application.

By leveraging these technologies and implementing the aforementioned features, the Waffar system empowers users to easily manage their receipts, earn rewards through points, search for the best prices, and make informed purchasing decisions.

## **Tools and Techniques**

As the Waffar web application, we have implemented various tools and techniques to provide an efficient and user-friendly experience for our clients. Here are some key aspects of our system implementation:

1. Backend Framework: We have utilized the Python Flask framework to develop the backend of our web application. Flask is a lightweight and flexible framework that allows us to build scalable web applications quickly.

2. Database Management: For managing the storage of receipt data and other information, we have employed the SQLite database. SQLite is a reliable and easy-to-use database system that seamlessly integrates with Flask.

3. Frontend Development: Our frontend is built using HTML5, CSS, and JavaScript. HTML5 provides the structure and layout of the web pages, CSS enhances the visual styling, and JavaScript adds interactivity and dynamic functionality to the user interface.

4. Optical Character Recognition (OCR): To extract receipt data from the scanned images, we have integrated the JavaScript Tesseract library. Tesseract is a powerful OCR engine that enables accurate text recognition from images.

5. User Points System: We have implemented a points system to incentivize clients for scanning their receipts. Every time a client scans a new receipt, they earn points that can be redeemed for rewards or benefits within the system. This encourages users to actively engage with the application.

6. Product Management: The system allows the system admin to insert and manage the list of products available for search and comparison. This ensures that clients can select products from a comprehensive and up-to-date database.

7. Cart Functionality: Our system includes a cart feature that enables clients to add the products they searched for. By adding items to the cart, clients can get an overview of the best prices available for the entire cart, helping them make informed purchasing decisions.

8. Error Handling: We have implemented robust error handling mechanisms to ensure the system handles unexpected scenarios gracefully. For example, if a new product is extracted from the receipt but doesn't exist in the product list, the system will skip it and continue processing the remaining products.

These tools and techniques collectively contribute to the successful implementation of the Waffar web application. By leveraging Flask, SQLite, HTML5, CSS, JavaScript, Tesseract OCR, and other components, we have created a reliable and efficient system that allows clients to scan receipts, extract data, earn points, search for best prices, and manage their shopping cart effectively.

## **Implementation**

Implementation in web application development refers to the process of building and deploying the application based on the design specifications created during the system design phase. The implementation phase involves several key steps:

1. Developing the different modules and components of the application: This involves coding the different modules and components of the application, including the user interface, backend functionality, and data storage.

2. Integrating the different modules and components into a cohesive whole: This involves integrating the different modules and components of the application, ensuring that they work effectively and efficiently together. This includes testing and debugging the application to ensure that it performs as expected.

3. Testing the application: This involves testing the application to ensure that it meets the required specifications, including functionality, performance, and usability. This includes testing the application on different devices and platforms, as well as with different user scenarios and inputs.

4. Deploying the application: This involves deploying the application to the desired platform or platforms, such as the Apple App Store or Google Play Store. This includes preparing the application for submission, including creating and submitting the necessary documentation and meeting the required guidelines and standards.

5. Providing ongoing maintenance and support for the application: This involves providing ongoing maintenance and support for the application, including bug fixing, updates, and enhancements. This helps to ensure that the application continues to meet the needs and preferences of its users and stakeholders.

By completing these steps in the implementation phase, we can ensure that the mobile application is built and deployed in a way that meets therequirements and specifications identified during the system design phase. The implementation phase will ensure that the application provides a high-quality user experience, meets technical and functional specifications, and supports the business goals of the project. It is important to follow best practices in implementation, such as using version control systems, automated testing tools, and continuous integration and deployment tools, to ensure that the development process is efficient, effective, and scalable. Additionally, ongoing maintenance and support are critical for the long-term success of the application, as it helps to ensure that the application continues to meet the needs and preferences of its users and stakeholders..

## **System Interfaces**

System interfaces refer to the different ways in which the mobile application interacts with other systems, devices, and platforms. In mobile application development, system interfaces can include:

1. User Interfaces (UI): The UI is the primary interface between the user and the application, and includes the different screens, menus, buttons, and other elements that the user interacts with when using the application.

2. Application Programming Interfaces (APIs): APIs are used to allow the mobile application to interact with other systems and platforms, such as social media platforms, payment gateways, and messaging services. APIs provide a standardized way for the mobile application to exchange data and information with these systems and platforms.

3. Database Interfaces: Database interfaces are used to allow the mobile application to interact with the backend database, where user data, transaction data, and other types of data are stored. Database interfaces provide a way for the mobile application to read, write, and update data in the database.

4. Device Interfaces: Device interfaces are used to allow the mobile application to interact with the different hardware and software components of the device, such as the camera, microphone, GPS, and other sensors. Device interfaces provide a way for the mobile application to access and use the different capabilities of the device.

By designing and implementing effective system interfaces, we can ensure that the mobile application is able to interact seamlessly and efficiently with other systems, devices, and platforms. This will help to enhance the functionality and usability of the application, and provide a better userexperience. It is important to design and implement system interfaces that are secure, scalable, and easy to use, and that meet the specific requirements and specifications of the project. Effective system interfaces can be developed using a variety of tools and techniques, such as APIs, SDKs, and frameworks, and should be tested thoroughly to ensure that they work effectively and efficiently. By following best practices in system interface design and implementation, we can ensure that the mobile application is built and deployed in a way that meets the needs and preferences of its users and stakeholders..

# Chapter Six | Testing

## **Overview**

In the testing phase of mobile application development, we evaluate the application to ensure that it meets the required specifications and provides a high-quality user experience. Testing is a critical phase in the development process, as it helps to identify and resolve bugs and issues before the application is deployed to users.

The testing phase includes several key steps, including:

1. Planning the testing process: This involves identifying the testing objectives, creating a testing plan, and defining the testing scope and approach. This will ensure that the testing process is effective and efficient, and that it meets the requirements and specifications identified during the system design phase.

2. Creating test cases: This involves creating detailed test cases that cover all aspects of the application, including functionality, performance, and usability. Test cases should be designed to simulate different user scenarios and inputs, and should be documented and tracked to ensure that they are executed correctly.

3. Conducting testing: This involves executing the test cases and documenting the results. Testing can be conducted manually or automated, and should be conducted on different devices and platforms to ensure that the application works effectively and efficiently across different environments.

4. Identifying and resolving bugs and issues: This involves identifying and documenting any bugs or issues that are identified during the testing process, and working to resolve them. Bugs and issues should be prioritized based on their severity and impact on the user experience, and should be resolved as quickly as possible to ensure that the application meets the required specifications.

5. Conducting acceptance testing: Thisinvolves conducting acceptance testing with users or stakeholders to ensure that the application meets their needs and preferences. Acceptance testing can help to identify any gaps or issues in the application that may have been missed during the testing process.

By completing these steps in the testing phase, we can ensure that the mobile application is thoroughly tested and meets the required specifications for functionality, performance, and usability. Testing is a critical phase in the development process, as it helps to identify and resolve bugs and issues before the application is deployed to users. It is important to follow best practices in testing, such as creating detailed test cases, conducting testing on different devices and platforms, and prioritizing and resolving bugs and issues quickly. Additionally, acceptance testing can help to ensure that the application meets the needs and preferences of its users and stakeholders..

## **System Testing**

System testing is a key component of the testing phase in mobile application development. It involves testing the application as a whole, to ensure that all the modules and components work effectively and efficiently together, and that the application meets the required specifications for functionality, performance, and usability.

System testing includes several different types of testing, including:

1. Functional Testing: This involves testing the functionality of the mobile application, to ensure that it meets the required specifications and performs as expected. Functional testing can include testing individual functions or features, as well as testing the application as a whole.

2. Performance Testing: This involves testing the performance of the mobile application, to ensure that it meets the required specifications for speed, responsiveness, and scalability. Performance testing can include testing the application under different loads and conditions, to ensure that it performs effectively and efficiently in real-world scenarios.

3. Usability Testing: This involves testing the usability of the mobile application, to ensure that it meets the needs and preferences of its users. Usability testing can include testing the application with different user scenarios and inputs, as well as testing the application with different types of users.

4. Compatibility Testing: This involves testing the compatibility of the mobile application, to ensure that it works effectively and efficiently across different devices and platforms. Compatibility testing can include testing the application on different versions of operating systems, browsers, and devices.

By conducting system testing, we can ensure that the mobile application meets the required specifications for functionality, performance, and usability, and thatit provides a high-quality user experience. System testing should be conducted using a variety of tools and techniques, such as automated testing frameworks, manual testing, and user testing. It is important to document and track the results of system testing, and to prioritize and resolve any issues or bugs that are identified during the testing process. Additionally, system testing should be conducted on different devices and platforms to ensure that the application works effectively and efficiently across different environments. By following best practices in system testing, we can ensure that the mobile application is thoroughly tested and meets the needs and preferences of its users and stakeholders..

### **Unit Testing**

Unit testing is a type of testing that focuses on testing individual units or modules of code in isolation, to ensure that they work effectively and efficiently. In mobile application development, unit testing can be used to test individual functions or features of the application, such as user interface elements, backend functionality, and data storage.

Unit testing involves creating test cases that cover the different scenarios and inputs that the unit is expected to handle. Test cases should be designed to simulate different user scenarios and inputs, and should be documented and tracked to ensure that they are executed correctly. Unit testing can be conducted using a variety of tools and techniques, such as automated testing frameworks, manual testing, and emulators or simulators.

By conducting unit testing, we can ensure that individual units or modules of code work effectively and efficiently, and that they meet the required specifications for functionality, performance, and usability. Unit testing can also help to identify and resolve bugs and issues early in the development process, before they become more complex and difficult to resolve.

It is important to follow best practices in unit testing, such as creating detailed test cases, testing individual units in isolation, and prioritizing and resolving bugs and issues quickly. Unit testing should be conducted regularly throughout the development process, and should be integrated into the development workflow to ensure that new code is tested thoroughly before it is integrated into the application. By following best practices in unit testing, we can ensure that the mobile application is thoroughly tested and meets the required specifications for functionality, performance, andusability..

### **Integration Testing**

Text.

## **Users Testing**

Text.

## **Testing Summary**

Text.

# Chapter Seven | Conclusion & Future-work

## **Overview**

As a web application called Waffar, our system aims to provide users with a convenient and rewarding experience for managing their receipts and finding the best prices for their desired products.

Here's an overview of the system's conclusion and potential future work:

**Conclusion:**

Waffar successfully enables users to scan their receipts, extract important information using JS Tesseract OCR, and store relevant data such as product descriptions, prices, supermarket details, and receipt creation time in a SQLite3 database. Users are incentivized to participate through a point-based system, earning points each time they scan a new receipt. The system also allows users to search for the best prices by selecting products from a list inserted by the system admin. Users can add products to their cart to discover the places that offer the best prices for their entire shopping list.

## **Future-work**

1- Enhanced User Experience: Continuously improving the user interface and overall user experience of the Waffar web application can make it more intuitive and engaging for users.

2- Integration with Online Retailers: Explore opportunities to integrate with online retailers' APIs to fetch real-time product prices and availability. This would provide users with up-to-date information and expand the range of products they can compare.

3- Expansion to Mobile Platforms: Develop a mobile version of Waffar, compatible with both iOS and Android devices, to reach a broader user base and allow for on-the-go receipt scanning.

4- Social Features: Introduce social features that enable users to share their shopping experiences, compare prices with friends, and receive recommendations based on their preferences.

5- Advanced Analytics: Implement advanced analytics capabilities to analyze user behavior, shopping trends, and product preferences. This data can be leveraged to provide personalized offers, discounts, or targeted advertisements to users.

6- Integration with Loyalty Programs: Collaborate with retail chains or supermarkets to integrate Waffar with their existing loyalty programs, allowing users to earn additional rewards or discounts based on their scanning activity.

7- Expansion of Database: Continuously update the product database with new products and price information to ensure accurate comparisons and provide users with a comprehensive shopping experience.

8- Integration with Online Payment Systems: Enable users to make purchases directly through the Waffar platform by integrating popular online payment systems, offering a seamless end-to-end shopping experience.

By focusing on these areas of future work, Waffar can evolve into a robust and feature-rich platform that revolutionizes how users manage their receipts, find the best prices, and make informed shopping decisions.

# References List

* Flask Documentation: <https://flask.palletsprojects.com/>
* SQLite Documentation: <https://www.sqlite.org/docs.html>
* HTML5Documentation:<https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>
* CSSDocumentation:<https://developer.mozilla.org/en-US/docs/Web/CSS>
* JavaScriptDocumentation:<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
* TesseractOCRDocumentation:<https://tesseractocr.github.io/tessdoc/Home.html>
* OWASP: <https://owasp.org/>
* GoogleDevelopers-WebFundamentals: <https://developers.google.com/web/fundamentals>