SOFTWARE REQUIREMENTS SPECIFICATION

FOR

ONLINE ORDERING SYSTEM

Prepared By

DIGII

İREM YUVALI

İREM BOLAT

DUYGU ADA DİVLE

GİZEM ERDOĞAN

IVAN KUZNETSOV

November, 2021

CHANGE HISTORY

This document is the first version of *ONLINE ORDERING SYSTEM*, Software Requirements Specification that was released on November, 2021. The subsequent changes will be mentioned in this part of the Software Requirements Specification.

PREFACE

The document contains the Software Requirements Specifications of *ONLINE ORDERING SYSTEM*. The mission of the project is to develop a web-based ordering system for CEN421-System Analysis and Design Course given by Havva Esin Ünal.

The goal of the Software Requirements Specification is to describe the requirements of the *ONLINE ORDERING SYSTEM*. This system is prepared according to IEEE standard **[1,2,3,4]**. The software requirements specification is in content compliance with the IEEE standard 1058-1998 in which the contents of this standard are rearranged and a mapping is provided. That is, the content compliant Software Requirements Specifications is mapped into various clauses and subclauses of the IEEE standard 1058-1998.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION (SECTION 1 OF THE SRS)

1.1 Purpose of the SRS

The purpose of the Software Requirements Specification (SRS) is

- to identify what the *Online Ordering System* is supposed to do,
- to cite explicitly all functions, interfaces, and performance requirements. This document is the basis upon which all design, coding, and testing will be based. It outlines any constraints and design issues that would affect the product's performance and reliability.
- to provide a basis for controlling the evolution of the Online Ordering System and system verification in a correct, complete, unambiguous and verifiable manner.

The intended audience is the instructor of the *CEN421-System Analysis and Design* course and the subcontractor group named Dream Squad.

1.2 Scope of the Product

The software product to be produced is the Online Ordering System, which aims to provide a web-based ordering system for CEN421-System Analysis and Design Course.

This document presents the detailed software requirements analysis for the Online Ordering System. The scope includes the functional, performance and operational requirements of the Online Ordering System.

The software requirements analysis is based on the Initial Plan for Online Ordering System **[5]**.

The objectives of the Online Ordering System are

- Saving time for both the customer and the restaurant,
- Eliminating communication difficulties in a way that appeals to everyone,
- Keeping the level of hygiene and cleanliness high.

The scope of the Online Ordering System is:

- The Online Ordering System will be a web-based platform that users can access easily.
- The Online Ordering System will be used as a separate system. It will not be integrated with any system. Importing data from other sources about available restaurants is out of scope.
- In scope of developing Online Ordering System will be developing and testing both server and client sides of application. Promoting, advertisement, platform trainings are out of scope.

1.3 Definitions, Acronyms and Abbreviations

DFD: Data Flow Diagram.

DIGII: Software development and project team of Online Ordering System.

Dream Squad: Subcontractor of DIGII.

ER Diagram: Entity Relationship Diagram.

HTTPS: Hyper Text Transfer Protocol Secure

IEEE: Institute of Electrics & Electronics Engineering.

Online Ordering System: A web-based project developed by DIGII for CEN421-System Analysis and Design Course.

SRS: Software Requirements Specification

TCP/IP: Transmission Control Protocol/Internet Protocol

UI: User Interface

URL: Uniform Resource Locator

Users: The people who make use of the system e.g., customers, managers and meal deliverers.

1.4 References

[1] IEEE Std 1058-1998, IEEE Standard for Software Management Plans.

[2] IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications

[3] IEEE Std 1016-1998, IEEE Recommended Practice for Software Design Descriptions

[4] IEEE Std 1063-1998, IEEE Standard for Software User Documentation

[5] Initial Plan for Online Ordering System, 1st November, 2021

1.5 Overview

The SRS is organized in three sections:

Introduction Section (Section 1 of the SRS): in which purpose, scope, definitions, references and an overview of the SRS are given.

Overall Description Section (Section 2 of the SRS): describes the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in specific requirements part (Section 3 of the SRS), and makes them easier to understand.

Specific Requirements Section (Section 3 of the SRS): contains all of the software requirements to a level of detail sufficient to enable DIGII to design the Online Ordering System to satisfy those requirements, and testing that the system satisfies those requirements.

2 OVERALL DESCRIPTION (SECTION 2 OF THE SRS)

2.1 Product Perspective

2.1.1 System Interfaces

Online ordering system should be standalone application with no extra dependencies on system level. One thing that is acceptable is to use open-source libraries inside code to make it easy to maintain and develop. All basic information will be included into system must be inserted my developers after checking its correctness and relevance.

2.1.2 User Interfaces

- UI should be done primarily for web usage
- UI must be done using one style guideline
- UI should be adoptable for both mobile and laptop usage
- UI must be user friendly and easy to understand
- There will be 6 types of UI for the following tasks:
    - Registration
    - Login
    - Profile settings / information
    - Main search window (search for restaurant)
    - Restaurant information (basic information: location/phone/rating/description + list of meals)
    - Order confirmation

2.1.3 Hardware Interfaces

There is no defined hardware interface for the Online Ordering System.

2.1.4 Software Interfaces

Software interfaces of the Online Ordering System Project are given in table below.

Name	Mnemonic	Version Number	Source	Purpose
Python Language Reference, version 3.6	Python	3.6	python.org	Runtime for backend part of application.
PostgreSQL, version 14.0	PostgreSQL	14.0	postgresql.org	Maintain databases (users, restaurants, etc.) for application.
Django, version 3.2.9	Django	3.2.9	djangoproject.com	Developing back-end side of project. Implementing servitor with ports and URLs
HTML **CSS** **JavaScript**	HTML CSS JS	HTML5 CSS JS		Developing frontend side of the project.

Table 1: Software Interfaces

2.1.5 Communication Interfaces

Due to developing web-based application communication between server and client sides of project will be done using TCP/IP protocols over HTTPS. URL address will be provided to access application through the Internet.

Client and server sides of application will communicate through port number 80 and different URLs for different purposes.

2.1.6 Memory Constraints

There need to be some requirements for developing the Online Ordering System Project. These are given below:

- RAM - 512MB (To run Python + Django)
- HARD DRIVE - 5 GB (To Store Python + Databases)

2.1.7 Operations

- System Administrator / Developers
    - Remove inappropriate content on site
    - Add data to the database which cannot be uploaded automatically
    - Answer questions from users about platform
- Manager (Owner of restaurant)
    - Add restaurant
    - Add item to menu of restaurant
    - Edit item of menu in restaurant
    - Delete item of menu in restaurant
    - Delete restaurant
- Customer
    - Search for restaurant
    - Create order at restaurant
    - Submit feedback if order has been made
    - Edit personal information
- Meal Deliverer
    - See current order to deliver
    - Edit personal information
    - Provide feedback for their employer

2.1.8 Site Adaptation Requirements

Platform should run independently on different systems. Service initialization require installation of python development toolkit with addition of Django framework. Configuration of the service side will be provided as an extra file. Before plication will be accessible for each user there are several steps to do:

- Initialize databases for users, restaurants and menus
- Deploy server side of application to given host with registration its address in DNS

2.2 Product Functions

1. ***User authentication function:*** give users the ability to register a new account of two options as a customer and as a restaurant manager. Provide ability to log in to users account.
2. ***User update function:*** give users the ability to update their account information.
3. ***User delete function:*** give users the ability to delete their accounts.
4. ***Add restaurant function*****:** Provide the owner of the restaurant ability to add his restaurant to the Ordering System. He could provide such information as: location, description, price of regular meal, phone number to order food by phone. As a result, his restaurant will be available for customers in search window.
5. ***Delete restaurant function:*** provide manager of restaurant ability to delete his restaurant in case it's closing or something happened with it. As a result of this function restaurant will not be shown in the list for users searchers.
6. ***Manage menu of restaurant function:*** give restaurant manager utilities too at some positions to menu of restaurant or delete some positions, change prices and description of positions. As a result, menu will be changed in the databases so each user from that point we'll see only updated menu.
7. ***Search for a restaurant function:*** provide ability for users to search restaurants through name matching and what type of food is made in this restaurant.
8. ***Order a meal function:*** user will have ability to add different positions of menu he or she liked into basket and after this create an order. After confirmation order information will be delivered to user.
9. ***Give restaurant feedback function:*** if user ordered something from the restaurant he or she cooled provide feedback for this restaurant with rating pros and cons.
10. ***Help request form function:*** give users form we do they can feel to ask for help you they have struggled with platform.

2.3 User Characteristics

Administrator (Developers): These are the people who are responsible for creating, editing and deleting the users and also, they can view the entire system. Besides that, the administrators maintain the Online Ordering System.

Manager (Restaurant): the people who manage their products on the system. They can create, edit and delete the items on the menu.

Customer: The people who use the system for place an order. They can search restaurants, view the products, create an account and order. Also, they get feedback about their order.

Meal Deliverer: The people who are responsible for delivering the orders.

2.4 Constraints

- The Online Ordering System will be standalone application.
- The Online Ordering System will be web-based application, Android and iOS applications are out of scope. There are no hardware limitations.
- The Online Ordering System consists of four kinds of users, which are administrators, managers(restaurants), meal deliverers and customers.

2.5 Assumptions and Dependencies

- All the users are responsible for the truth of their information.
- It is the responsibility of the restaurants to make the updates of meals/products so that there must always has the up-to-date information.

2.6 Apportioning of Requirements

The help request and feedback functions/modules will not be developed in first version of the Online Ordering System.

Payment and credit card:

The credit card information is taken from the customer, but no credit card verification will be done. In the future versions of the Online Ordering System trust intermediaries will be used for the purpose of validating the credit card information. Since the credit card information taken from the customer cannot be used, money cannot be drawn from the banks, online. In the future versions of the Online Ordering System, banks will be used and the money transactions will be handled. Future versions of the Online Ordering System will not be developed within the scope of this course.

3 SPECIFIC REQUIREMENTS (SECTION 3 OF THE SRS)

This section of this SRS describes all specific software requirements in detail. Requirements include the descriptions of all inputs into and outputs from the software, in addition to the descriptions of functions in response to the inputs.

3.1 External Interfaces

The designing representation of these interfaces are given in Appendix C.

i. Main Page

- **Name:** Online Ordering System Main Page

- **Purpose:** It is used to make the login process; the customer can login from this page, and go to the related page (restaurant view page, deliverer view page, customer view page) according to the login user and password. If the visitor doesn't have any user name and password, then he/she would be registered to our system by following the new user or register link.

- **Inputs/Types:**
  Login Name = string
  Password = string

- **Outputs/Types:**
  If login is valid, to related page (restaurant view page, deliverer view page, customer view page) else invalid try again.

- **Screen Formats:** there are two sections in main page. The left side of the page is the registration part and the right side of the page login part. Users can access and view the two parts at the same time.

- **Execution Time:** This interface is the "Home Page" of our system.

- **Relationship to other interfaces:** according to the user type, users will be directed to the relevant page (restaurant view page, deliverer view page, customer view page).

ii. Restaurant View Page

- **Name:** Restaurant View Page

- **Purpose:** It's designed for the restaurants can access and view their orders and the list of products. Also, they can edit the menu (adding/deleting some contents)

- **Inputs/Types:**
  product name: string
  price: string
  ingredients: string

- **Outputs/Types**: Feedback (changes saved successfully)

- **Screen Formats:** There are two sections in restaurant view page. the Orders can be viewed on the left side of the page. The list of the products can be viewed on the right side of the page.

- **Execution Time:** Manager

- **Relationship to other interfaces:** ----

iii. Deliverer View Page

- **Name:** deliverer view page

- **Purpose:** the deliverer only can view the orders and give feedback about delivering status.

- **Inputs/Types:**
  deliverer choice: boolean (1: "delivered" or 0: "not delivered")

- **Outputs/Types**: "delivered" or "not delivered"

- **Screen Formats:** the deliverer only can view the orders and give feedback about delivering status.

- **Execution Time:** the interface is seen like "Meal Deliverer" when meal deliverer login to the system for meal deliverer user.

- **Relationship to other interfaces:** From this interface, logout and main page can be reached.

iv. Customer View Page

- **Name:** Customer View Page
- **Purpose:** It's designed for customers make a decision about their order types.
- **Inputs/Types:** -----
- **Outputs/Types:** ----
- **Screen Formats:** there are two sections in customer view page. Customers need to choose an order type (to eat at home or to eat at the restaurant)
- **Execution Time:** The interface is seen when customer, meal deliverer or manager login to the system.
- **Relationship to other interfaces:** if it's chosen to eat at home, it will be directed to the restaurant choice page. if it's chosen to eat at a restaurant, it will be directed to the restaurant code-pin page.

v. Restaurant Choice Page

- **Name:** Restaurant Choice Page
- **Purpose:** The restaurants on the system are listed in this page. Therefore, customers can pick the restaurant that they wanted to eat from.
- **Inputs/Types:** ------
- **Outputs/Types:** -------
- **Screen Formats:** The restaurants on the system are listed in this page.
- **Execution Time:** From this interface, firstly customer has to choose a restaurant for the eat then you can see the menu.
- **Relationship to other interfaces:** according to the chosen restaurant, customers will be directed to the product list of the restaurant.

vi. Meal Choice Page

- **Name:** meal choice page
- **Purpose:** The product list of the chosen restaurants are listed in this page. The customer can put a tick into checkboxes for products that they wanted to buy.
- **Inputs/Types:** checkboxes: boolean
- **Outputs/Types:** “The orders added to your bag* will be shown on the screen.
- **Screen Formats:** The product list of the chosen restaurants is listed in this page.
- **Execution Time:** The interface is seen when customers choose a restaurant.
- **Relationship to other interfaces:** The orders will be directed to the restaurant view and deliverer view pages. Also, customers can turn back to the customer view page again to create another order.

vii. Restaurant Code/Pin Page

- **Name:** Restaurant code-pin page
- **Purpose:** It is used for the separated from the home order. It will be given a randomly generated code for each restaurant on the system, and customers will use this code to reach the restaurant's menu page.
- **Inputs/Types:** pin: string
- **Outputs/Types:** --
- **Screen Formats:** By means of the code given by the restaurant, that code will be entered in a small area reserved on the order screen.
- **Execution Time:** The interface is seen when customers choose an option like restaurant ordering. Every restaurant will have a code and if the customers choose to eat at the restaurant, they will enter this code.
- **Relationship to other interfaces:** It will be directed to the customers view page and restaurant view page.

viii. Meal Choice Page

- **Name:** meal choice page
- **Purpose:** The product list of the chosen restaurants are listed in this page. The customer can put a tick into checkboxes for products that they wanted to buy.
- **Inputs/Types:** checkboxes: boolean
- **Outputs/Types:** "The orders added to your *bag* will be shown on the screen.
- **Screen Formats:** The product list of the chosen restaurants is listed in this page.
- **Execution Time:** The interface is seen when customers choose a restaurant.
- **Relationship to other interfaces:** The orders will be directed to the restaurant view and deliverer view pages. Also, customers can turn back to the customer view page again to create another order.

3.2 Functions

The functional requirements of the Online Ordering System are represented by DFD diagrams in Appendix B.

3.3 Performance Requirements

Performance requirements have not been determined yet.

3.4 Logical Database Requirements

The database requirements used by the Online Ordering System are given in the ER diagram in Appendix A.

3.5 Design Constraints

3.5.1 Standards Compliance

- Documentations of the Online Ordering System is prepared according to IEEE standards **[1,2,3,4]**.

3.6 Software System Attributes

3.6.1 Reliability

The Online Ordering System will be 100% operational for 90% of the calendar time during its first year of operation.

3.6.2 Availability

- The system will be available to serve customers and restaurants for 7 days and 24 hours.

3.6.3 Security

- It will be required to log in a username and password.
- Passwords will be in plain format.
- Communication will not be encrypted between the clients and the server.

3.6.4 Maintainability

- The Online Ordering System website will be developed using an object-oriented programming approach and using Python development language.
- Each and every module will be explicitly defined and documented.
- Documents for each phase of the software development process and source codes will be delivered with the product so that internal maintenance will take place.

3.6.5 Portability

The Online Ordering System will be a platform independent system, provided that, all of the software interfaces specified in *subsection 2.1.4* are provided for both the server site and the client site.

APPENDIX A

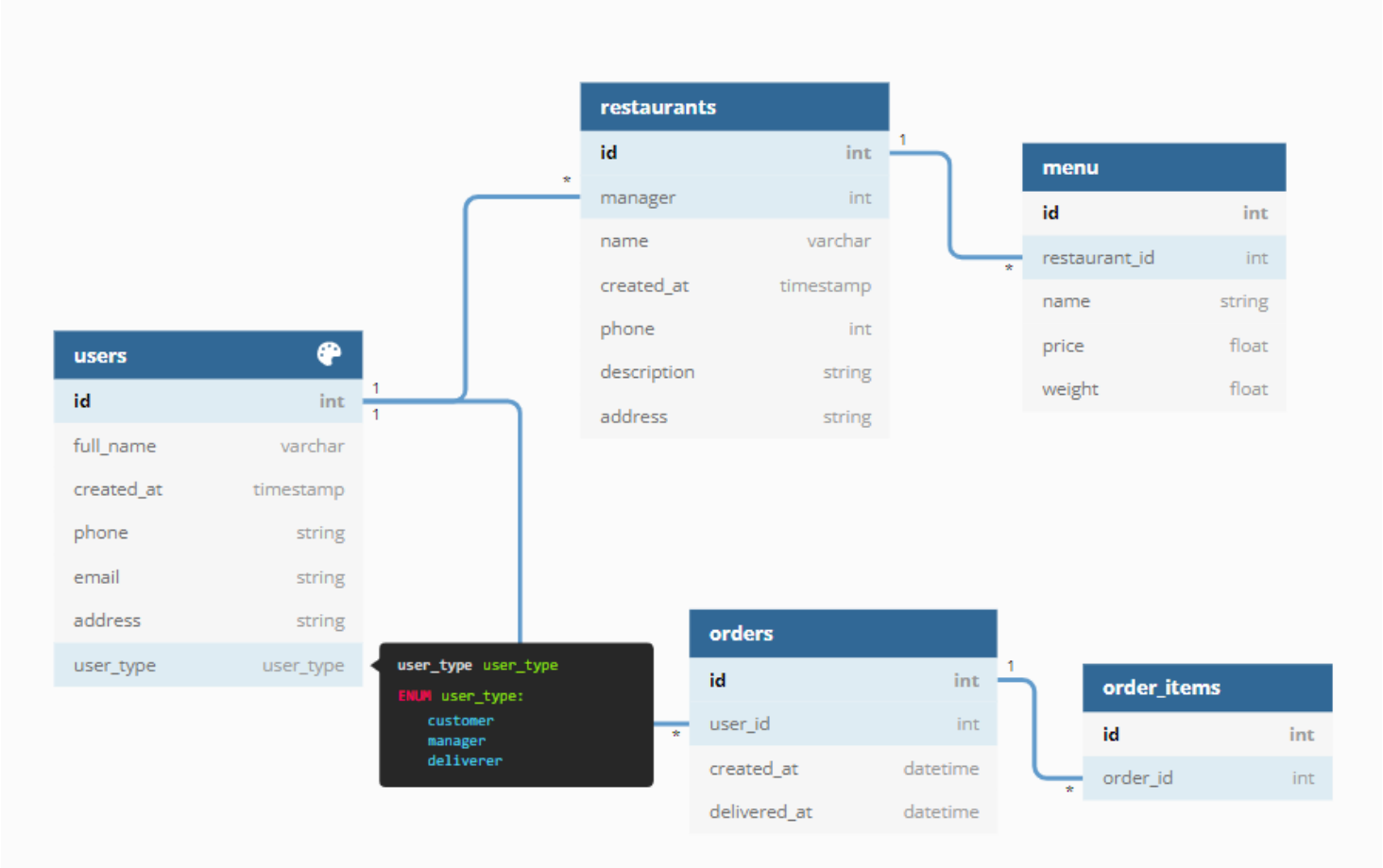

Figure 1: ER Diagram of the Online Ordering System

APPENDIX B

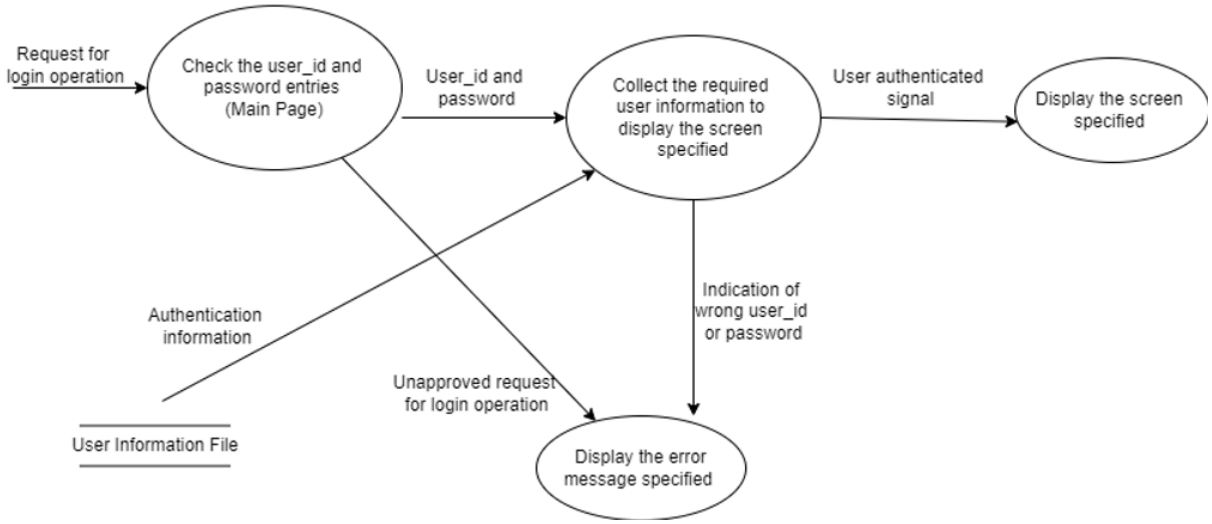

Figure 2: DFD for User Authentication Function

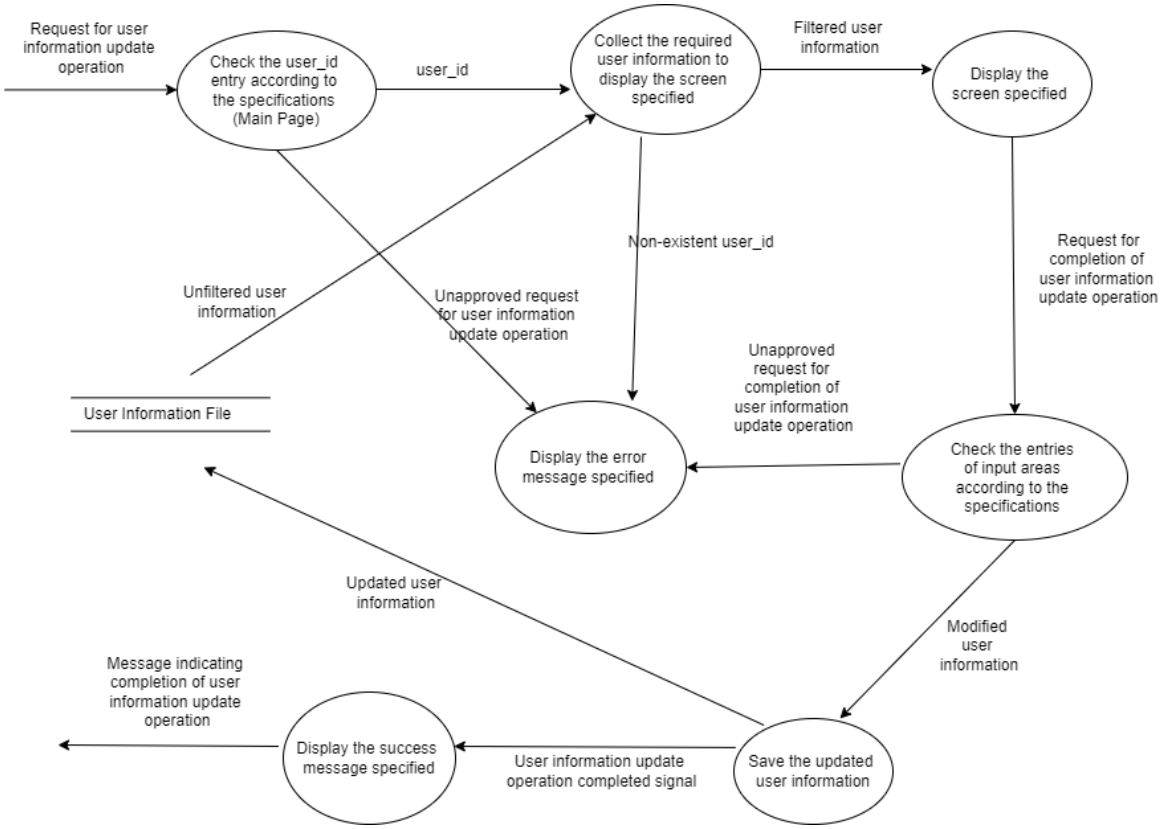

Figure 3: DFD for User Update Function

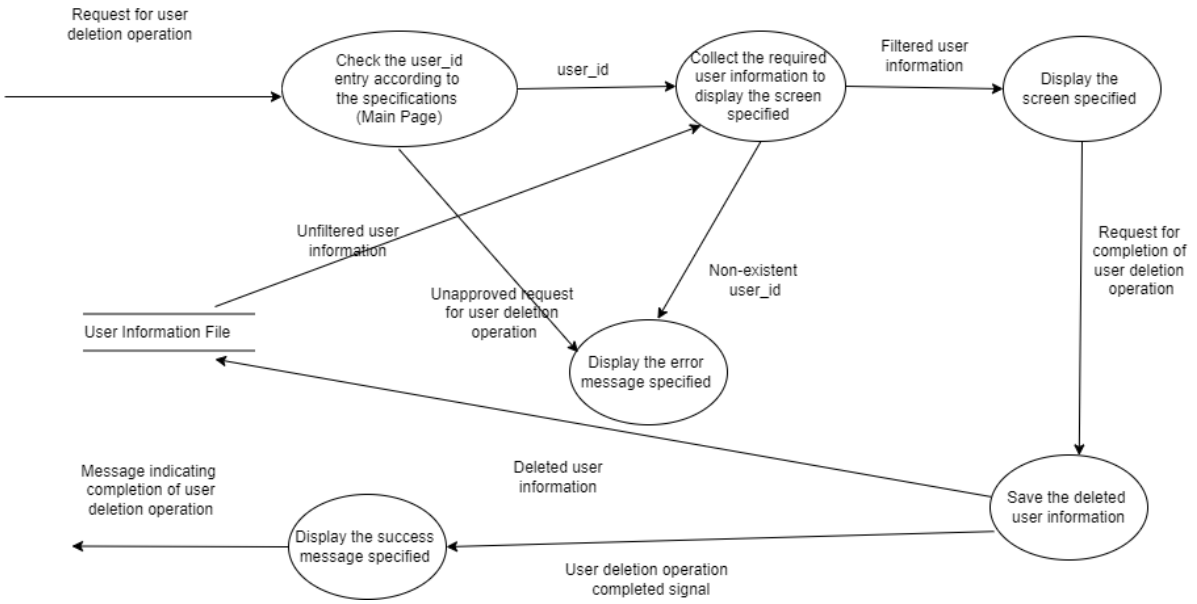

Figure 4: DFD for User Delete Function

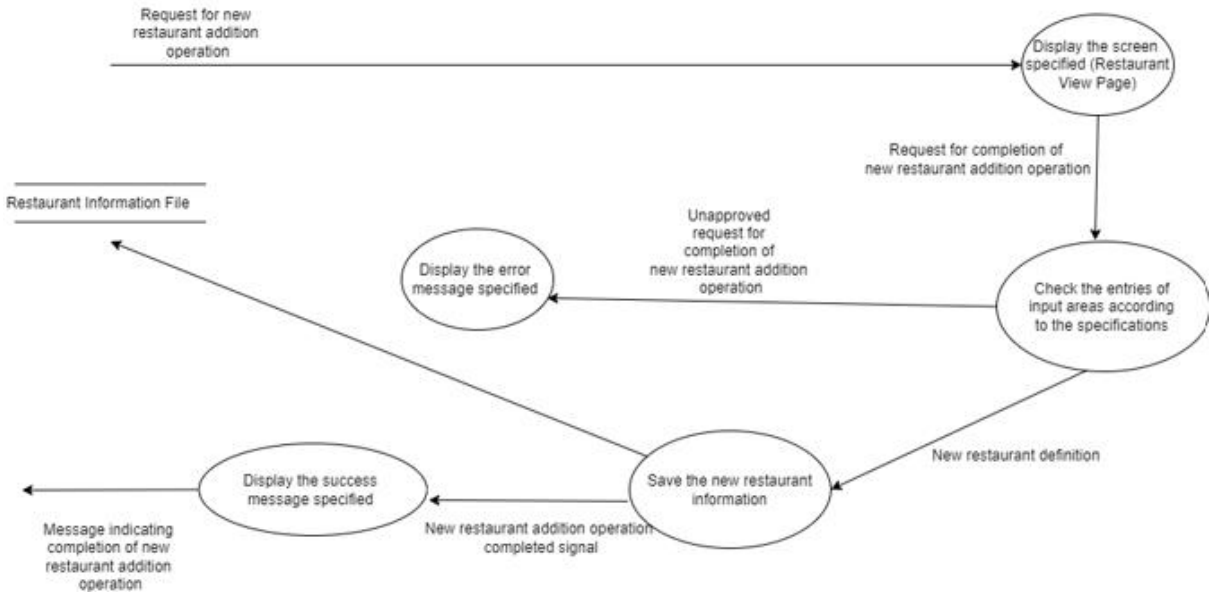

Figure 5: DFD for Add Restaurant Function

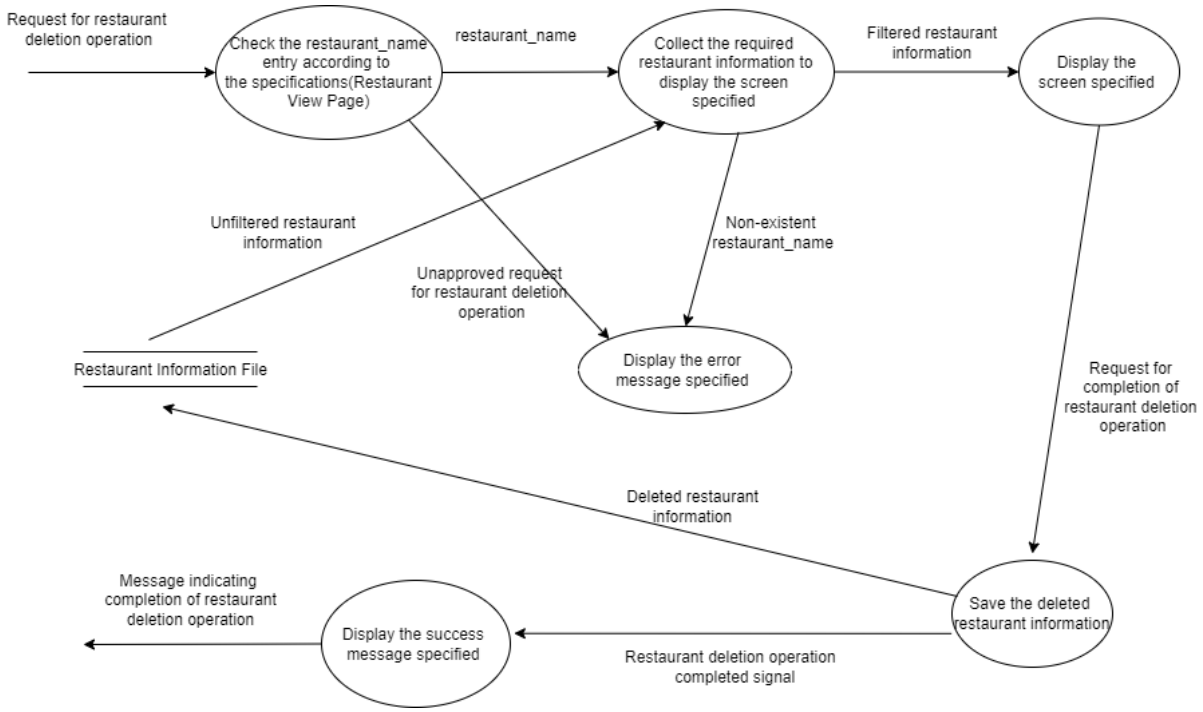

Figure 6: DFD for Delete Restaurant Function

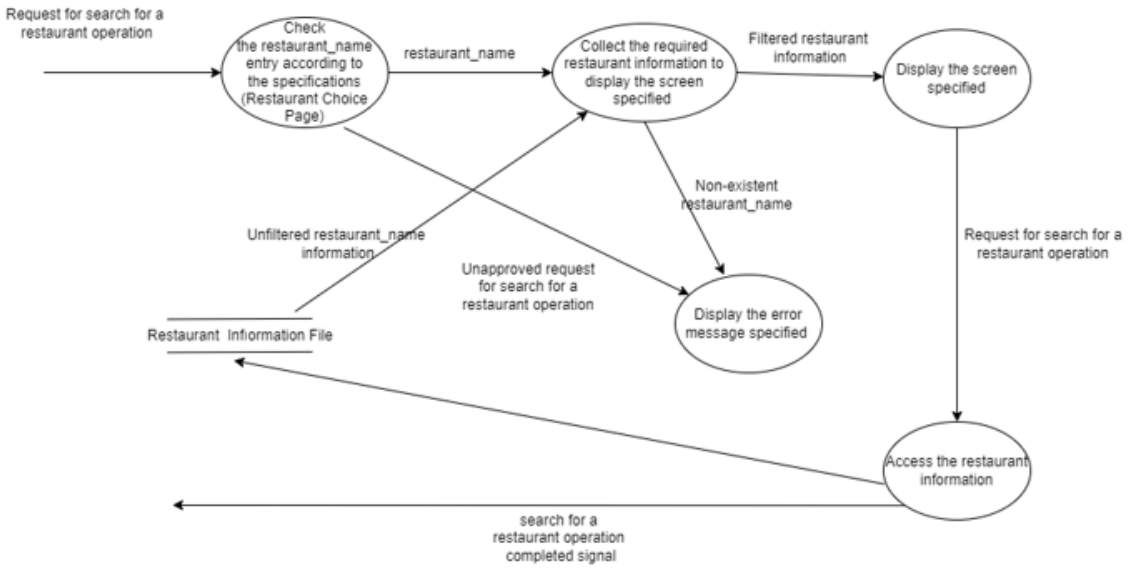

Figure 7: DFD for Search for a Restaurant Function

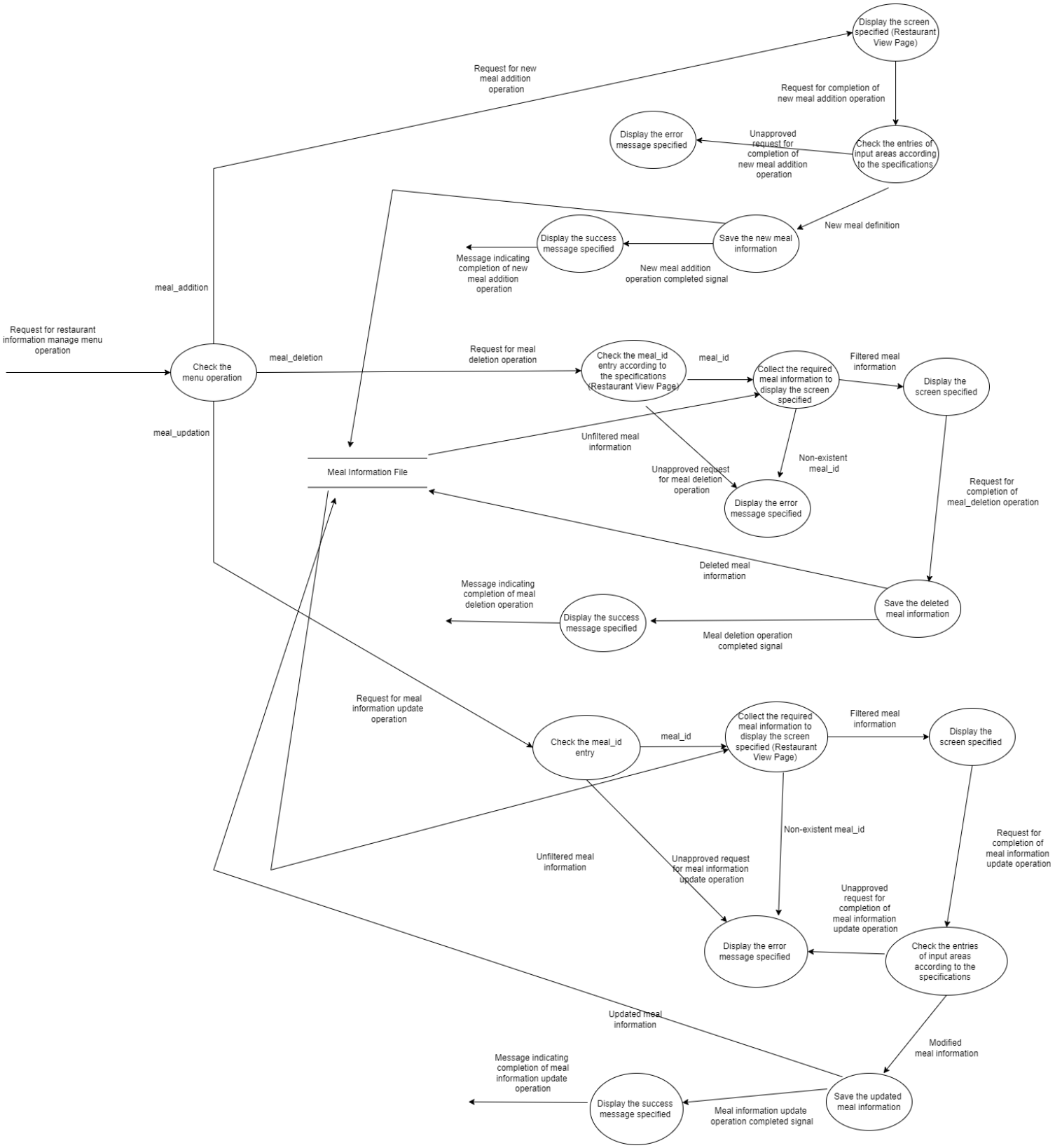

Figure 8: DFD for Manage Menu of Restaurant Function

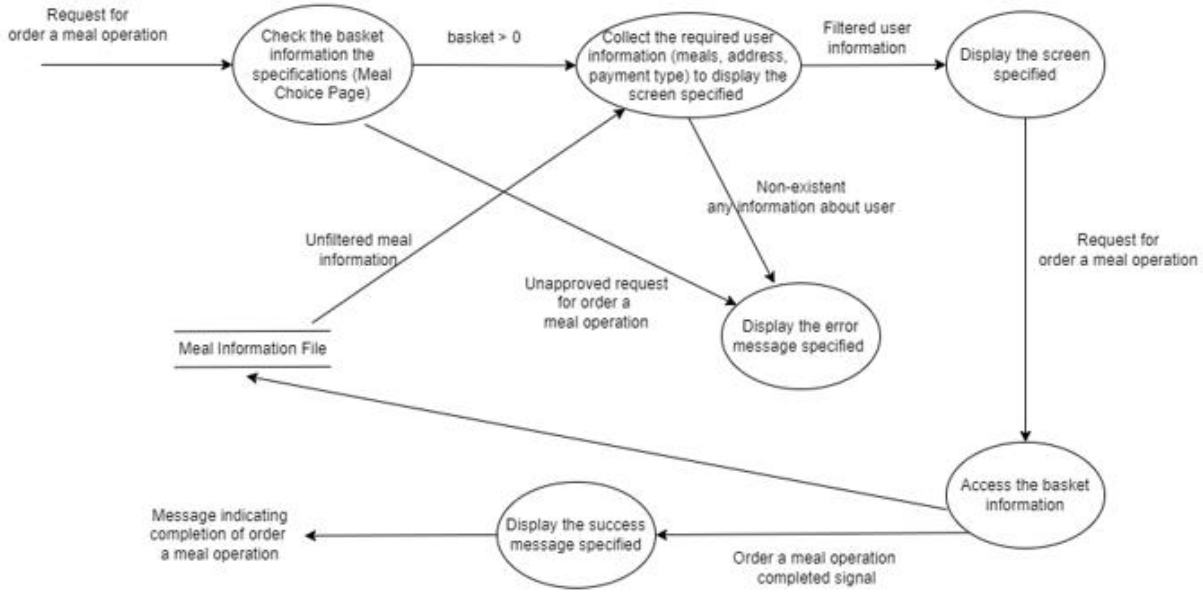

Figure 9: DFD for Order a Meal Function

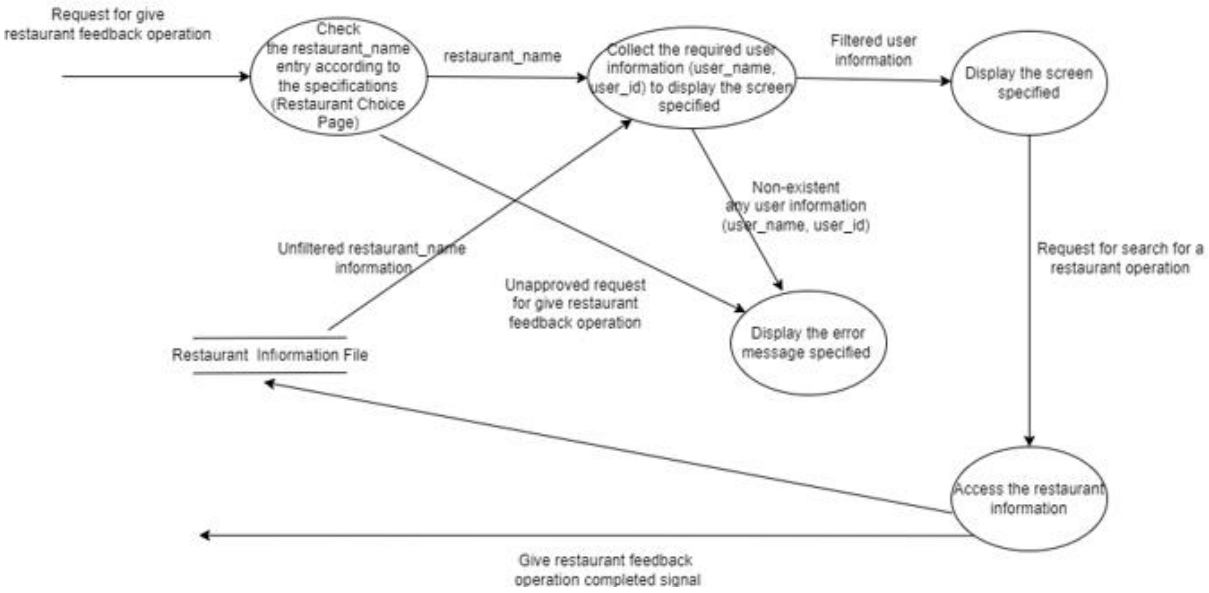

Figure 10: DFD for Give Restaurant Feedback Function

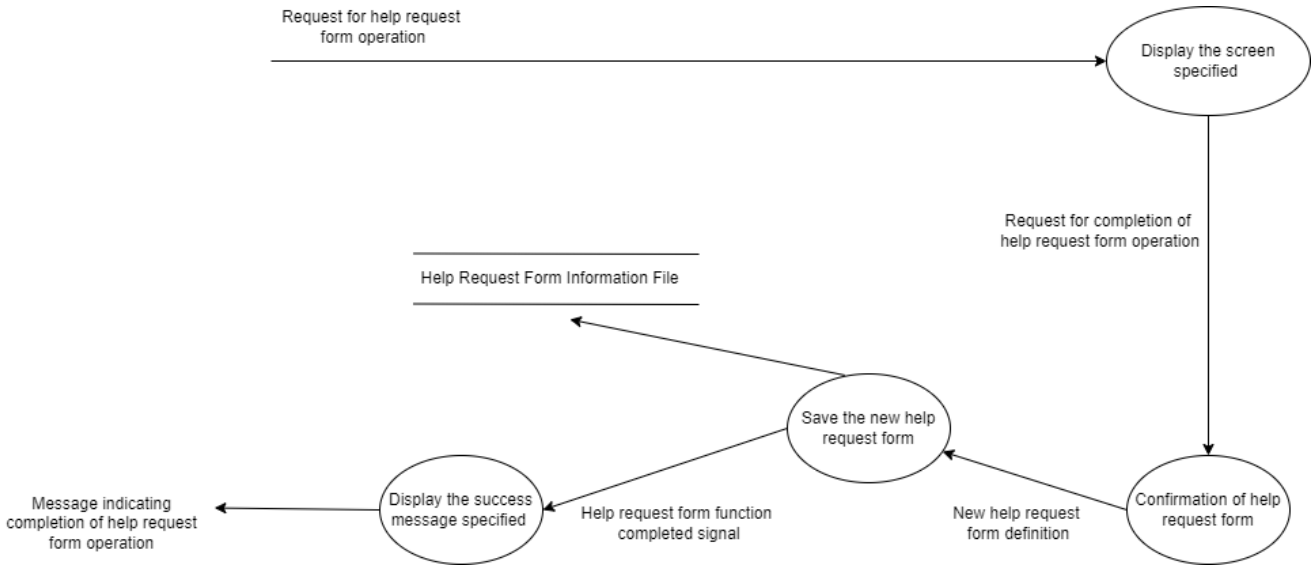

Figure 11: DFD for Help Request Function

APPENDIX C

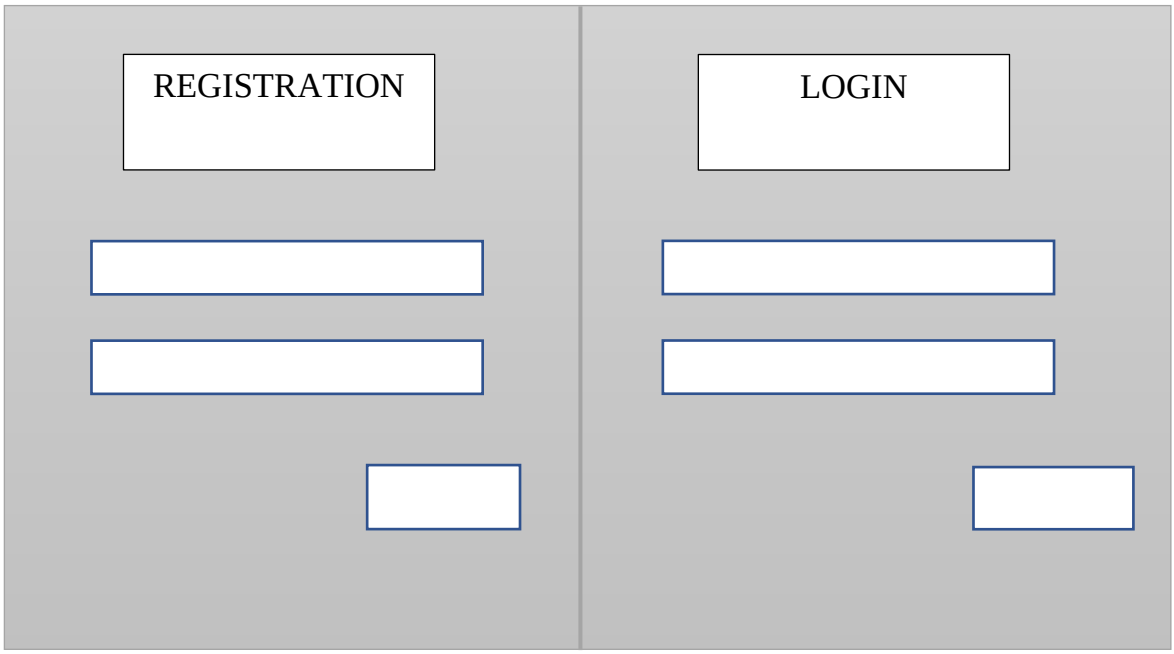

Figure 12: Main Page

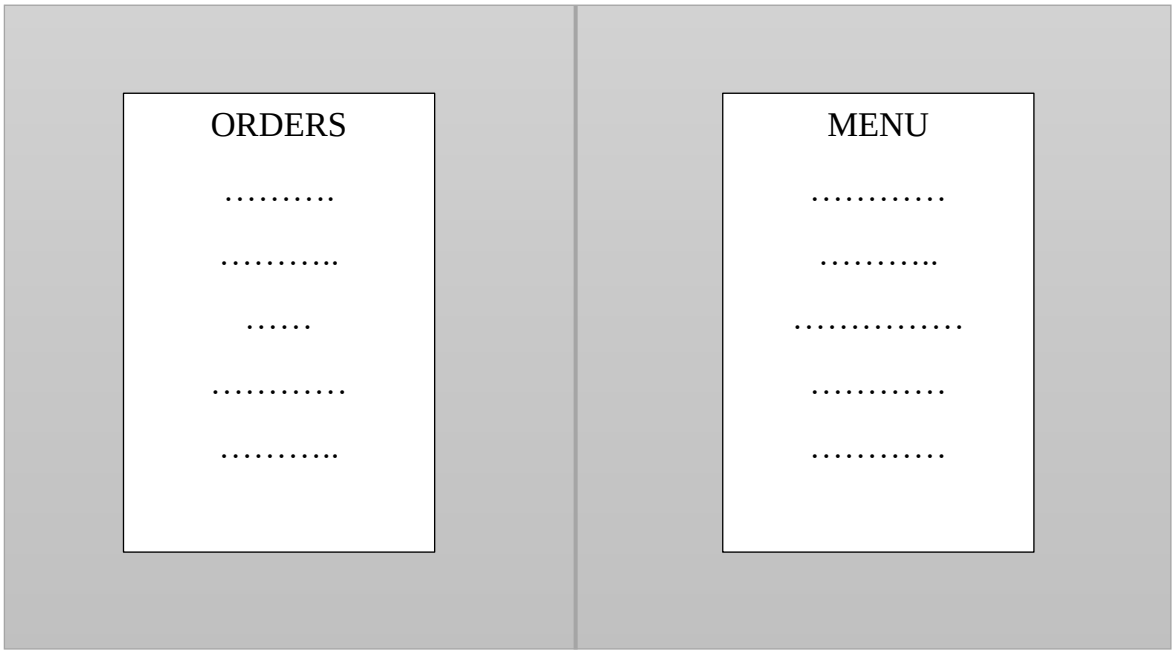

Figure 13: Restaurant's View Page

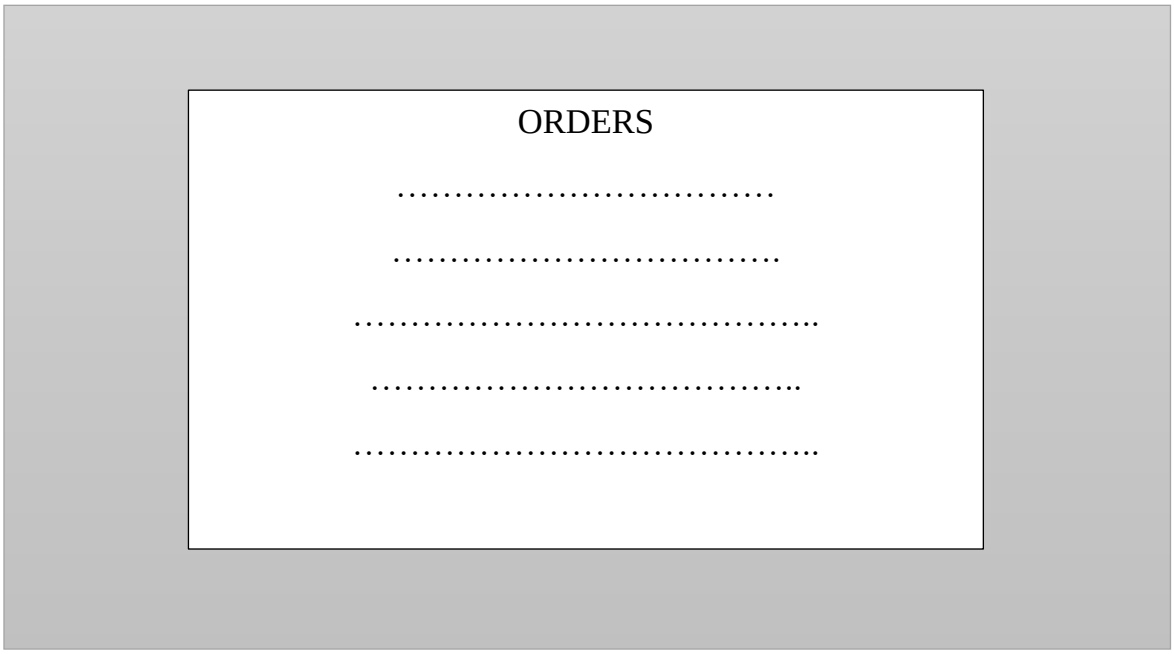

Figure 14: Deliverer's View Page

Figure 15: Customers View Page

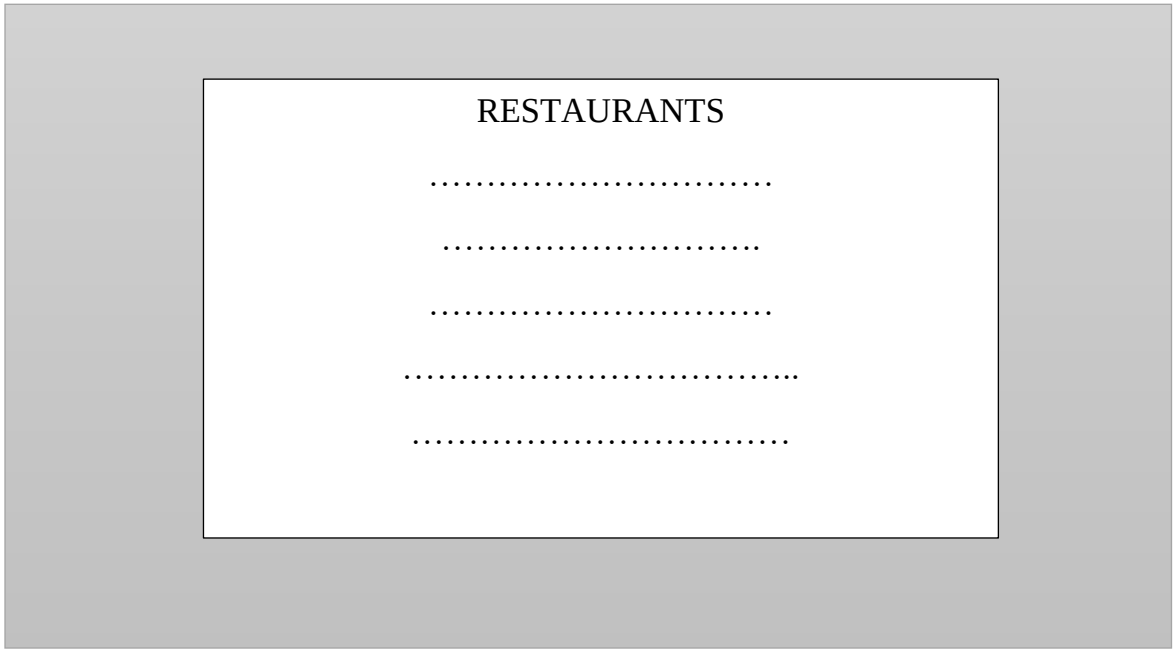

Figure 16: Restaurant Choice Page

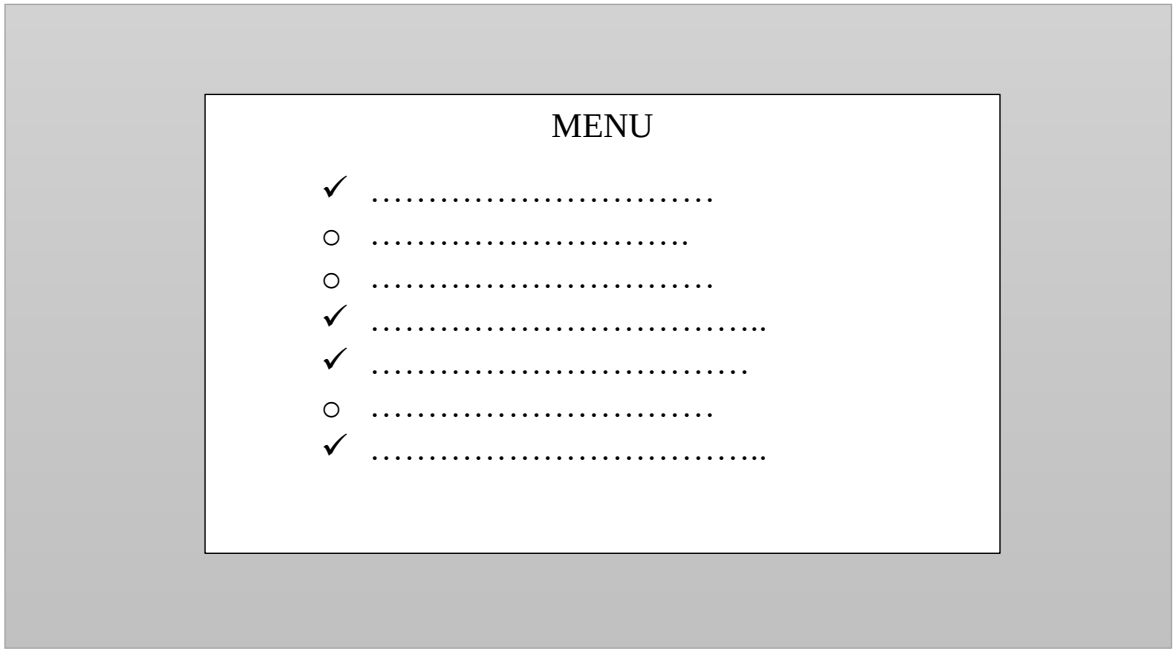

Figure 17: Meal Choice Page

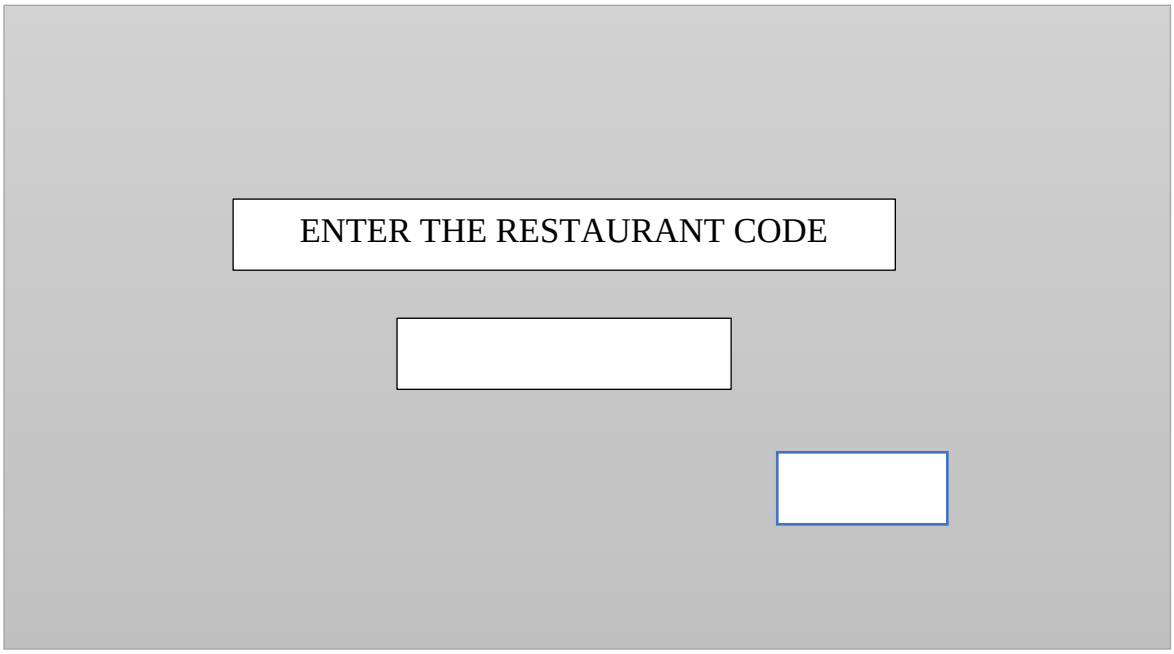

Figure 18: Restaurant Code/Pin Page

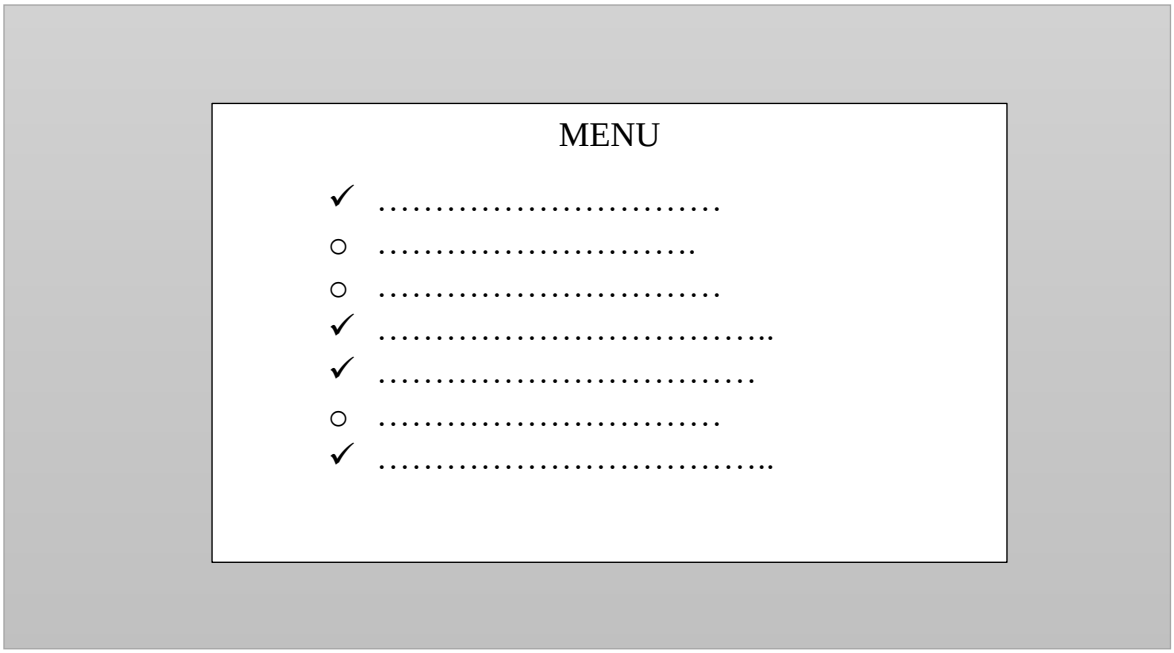

Figure 19: Meal Choice Page