

Foodies Web App

Software Requirements Specification

V1.0

19/2/2023

Revision History

Date	Description	Author	Comments
19/2/2023	Initial version	Abdullah	

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Name	Title	Date
<Your Name>	Lead Software Eng.	

V1.0	1
19/2/2023	1
1. Introduction	4
1.1 Purpose	5
1.2 Scope	5
Functional Requirements	6
The Web Ordering System	6
Menu Management System	7
Order Retrieval System	8
User Interface Specifications	8
Web Ordering System	8
Menu Management System	9
Order Retrieval System	9
System Evolution	10
3.1 Specific Requirements	10
3.2 Functional Requirements	11
3.3 Technical Requirements	12
3.4 Non-functional Requirements	13
• Performance	13
• Reliability	13
• Availability	14
• Security	14
• Maintainability	14
• Portability	14
4. Analysis Models	14
4.1 Data Flow Diagrams (DFD)	14
4.2 State-Transition Diagrams (STD)	15
5. Change Management Process	17

1. Introduction

In today's age of fast food and take-out, many restaurants have chosen to focus on quick preparation and speedy delivery of orders rather than offering a rich dining experience. Until very recently, all of these delivery orders were placed over the phone, but there are many disadvantages to this system, including the inconvenience of the User needing to have a physical copy of the menu, lack of a visual confirmation that the order was placed correctly, and the necessity for the restaurant to have an employee answering the phone and taking orders.

What I propose is an online ordering system, originally designed for use in college cafeterias, but just as applicable in any food delivery industry. The main advantage of my system is that it greatly simplifies the ordering process for both the User and the restaurant. When the User visits the ordering webpage, they are presented with an interactive and up-to-date menu, complete with all available options and dynamically adjusting prices based on the selected options. After making a selection, the item is then added to their order, which the User can review the details of at any time before checking out. This provides instant visual confirmation of what was selected and ensures that items in the order are, in fact, what was intended.

This system also greatly lightens the load on the restaurant's end, as the entire process of taking orders is automated. Once an order is placed on the webpage, it is entered into the database and then retrieved, in pretty much real-time, by a desktop application on the restaurant's end. Within this application, all items in the order are displayed, along with their corresponding options and delivery details, in a concise and easy to read manner. This allows restaurant employees to quickly go through the orders as they are placed and produce the necessary items with minimal delay and confusion.

1.1 Purpose

The Purpose of this document is to outline the requirements for the Foodies web app to be developed for the User. This document will be used by all stakeholders including developers and testers.

1.2 Scope

** Functional features:*

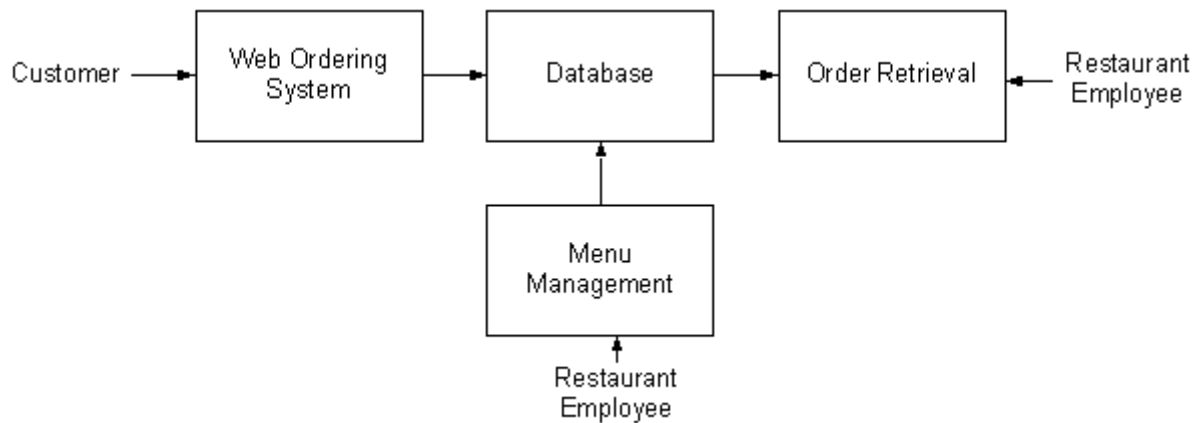
Sign up – login – searching – loyalty points – change password

** Deliverables:*

Development and testing work products that are relevant to the User

1.3 System Model

The structure of the system can be divided into three main logical components. The first component must provide some form of menu management, allowing the restaurant to control what can be ordered by Users. The second component is the web ordering system and provides the functionality for Users to place their order and supply all necessary details. The third and final logical component is the order retrieval system. Used by the restaurant to keep track of all orders which have been placed, this component takes care of retrieving and displaying order information, as well as updating orders which have already been processed.



1.4 Definitions, Acronyms, and Abbreviations

Abbreviation	Word
U	User
A	Admin

2. Specific Requirements

As can be seen in the system model diagramed above, each of the three system components essentially provides a layer of isolation between the end user and the database. The motivation behind this isolation is twofold. Firstly, allowing the end user to interact with the system through a rich interface provides a much more enjoyable user experience, particularly for the non-technical users which will account for the majority of the system's users. In addition, this isolation layer also protects the integrity of the database by preventing users from taking any action outside those which the system is designed to handle. Because of this design pattern, it is essential to enumerate exactly which functions a user will be presented with and these functions are outlined below, grouped by component.

2.1 The Web Ordering System

Users of the web ordering system, namely restaurant Users, must be provided the following functionality:

Create an account.
Manage their account.
Log in to the system.
Navigate the restaurant's menu.

Select an item from the menu.
Customise options for a selected item.
Add an item to their current order.
Review their current order.
Remove an item/remove all items from their current order.
Provide delivery and payment details.
Place an order.
Receive confirmation in the form of an order number.

As the goal of the system is to make the process of placing an order as simple as possible for the User, the functionality provided through the web ordering system is restricted to that which most pertinent to accomplish the desired task. All of the functions outlined above, with the exceptions of account creation and management, will be used every time a User places an order. By not including extraneous functions, I am moving towards my goal of simplifying the ordering process.

2.2 Menu Management System

The menu management system will be available only to restaurant employees and will, as the name suggests, allow them to manage the menu that is displayed to users of the web ordering system. The functions afforded by the menu management system provide user with the ability to, using a graphical interface:

New User
Edit User
Delete User
New Account
Edit Account
Delete Account
Change Password
Add a new/update/delete vendor to/from the menu.
Add a new/update/delete vendor to/from the menu.
Add a new/update/delete vendor to/from the menu

Add a new/update/delete food category to/from the menu.
Receive confirmation in the form of an order number.
Add a new/update/delete food category to/from the menu.
Add a new/update/delete option for a given food item.
Update price for a given food item.
Update default options for a given food item.
Update additional information (description, photo, etc.) for a given food item.

It is anticipated that the functionality provided by this component will be one of the first things noted by the restaurant user, as they will have to go through it to configure their menu, etc. before beginning to actually take orders. Once everything is initially configured, however, this component will likely be the least used, as menu updates generally do not occur with great frequency.

2.3 Order Retrieval System

Of the three components, the order retrieval system is functionally the simplest. Like the menu management system, it is designed to be used only by restaurant employees, and provides the following functions:

- Retrieve new orders from the database.
- Display the orders in an easily readable, graphical way.
- Mark an order as having been processed and remove it from the list of active orders.

2.4 User Interface Specifications

NIII

2.5 Web Ordering System

Users of the web ordering system will interact with the application through a series of simple forms. Each category of food has its own form associated with it which presents a drop down menu for choosing which specific item from the category should be added to the order, and a series of check boxes and radio buttons for selecting which options are to be included. Adding an item to the order is accomplished by a single button click. Users select which category of food they would like to order, and therefore which form should be displayed, by navigating a menu bar, an approach which should be familiar to most users.

Entering delivery and payment details is done in a similar manner. The user is presented with a form and must complete the required fields, which include both drop down and text boxes, before checking out and receiving a confirmation number. One thing worth noting here is that whenever possible drop down boxes and buttons were used over freeform input in order to both simplify the ordering process and reduce the possibility of an SQL injection attempt.

2.6 Menu Management System

User interaction with the menu management system is similar to that with the web ordering system. Users navigate a tree structure to find the vendor, category, or specific food item that they would like to modify and after making their selection they are presented with a form which displays all of the current fields and values associated with that item, all of which can be modified or removed. The form also presents buttons which allow the addition of new fields and values. Unlike the web ordering system, however, most of the input here will be freeform, specifically in the form of text boxes, since there is no finite set of fields which could be added. This does not raise a major concern though, as input sanitation will be performed, and the user, who is assumed to be a restaurant employee, is less likely to be malicious than a web user.

2.7 Order Retrieval System

User interaction with the order retrieval will be very simple. The application will automatically fetch new orders from the database at regular intervals and display the order numbers, along with delivery time, in a panel on the left hand side of the application. To view the details of an order, the user must simply click on that order number, which will populate the right-hand panel with the details, displayed in an easy to read and navigate tree structure. This structure can intuitively be expanded and collapsed to display only the desired information. Finally, once an order is processed, the user clicks a single button, labelled “Processed”, to remove it from the list of active orders.

2.8 System Evolution

As mentioned in the system model, at the heart of the entire ordering system is the database. In fact, the system could be completely operational using nothing but the database and an appropriate shell utility, assuming that all users are well-versed in SQL and enjoy using it to order food. While this would be a bit extreme, it does illustrate the point that the one part of the system which will stay relatively constant is the database. On the other hand, it is very probable that the other components will continue to evolve with time.

For example, with the booming popularity of mobile applications, I would really like to make the web interface available as a phone application as well. Also it may make sense to at some point migrate the menu management and order retrieval systems to web, or even mobile, applications as well, as some users may prefer to use them as such.

I am also certain that if this system goes into actual use, many requests will arise for additional features which I had not previously considered, but would be useful to have. For this reason, I feel as though the application can be constantly evolving, which I consider a very good thing.

3.1 Functional Requirements

This will be the largest and most important section of the SRS. The User requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project's software design, implementation, and testing.

Each requirement in this section should be:

- Correct
- Traceable (both forward and backward to prior/future artifacts)
- Unambiguous
- Verifiable (i.e., testable)
- Prioritised (with respect to importance and/or stability)
- Complete
- Consistent
- Uniquely identifiable (usually via numbering like 3.4.5.6)

Attention should be paid to carefully organize the requirements presented in this section so that they may be easily accessed and understood. Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.

3.2 Functional Requirements

This section describes specific features of the software project. The requirements are specified in the user story format and listed below.

USER STORY ID	As a	I want to	so that I can
1	User	Create my account using email and a strong password.	Login to the system
2	User	Login to the website using my email or id and password	Access to the website and use it's features
3	User	Find all nearby restaurants	Set an order
4	User	Be able to use search box in home page	Find my desired restaurant
5	User	Find a promotion and offer in home page	Use them
6	User	Have loyalty points	Get vouchers and offers

3.3 Technical Requirements

User story #1 (Sign up)

- T1** Email - Email is required
- T2** Email - Email must not be blank
- T3** Email - Email format is valid
- T4** Email - Email could be (commercial - organisational)
- T5** Password - Password is required
- T5** Password - Must not be blank
- T6** Password - Enter at-least one numeric value
- T7** Password - Enter at-least one uppercase letter
- T8** Password - Enter at-least one special character
- T9** Password - Enter at-least 8 characters
- T10** password - must be hidden(appears as black dots)
- T10.1** Password - user can show his hidden password through (show password button)
- T11** Confirm Password - Confirm Password must not be blank
- T12** Confirm Password - Passwords do Match
- T13** confirm password - must be hidden(appears as black dots)
- T13.1** Confirm Password - user can show his hidden password through (show password button)
- T14** confirm password - User can't paste any data in this field
- T15** Unique ID - ID is required
- T16** Unique ID- ID must not be blank
- T17** Unique ID - First character cannot have space
- T18** Unique ID - Special characters are not allowed

User story #2 (Login)

- T19** Email or ID - Email or ID is required
- T20** Email or ID - Email or ID must not be blank
- T21** Email or ID - Email or ID must be valid (registered)
- T22** Password - Password is required
- T23** Password - Must not be blank
- T24** Password - Must be matched with its registered Email
- T25** Password - must be hidden (appears as black dots)

- T26** Password - user can show his hidden password through (show password button)
- T27** forget password - user can access to forget password page through (forget password button) below logins field
- T28** forget password - Old Password must not be blank
- T29** forget password - New Password must not be blank
- T30** forget password - Enter at-least one numeric value
- T31** forget password - Enter at-least one special character
- T32** forget password - Choose a difficult Password
- T33** forget password - Confirm Password must not be blank
- T34** forget password - Passwords do not Match

User story #3 (Using website)

- T35** Search box - search box is displayed in top right corner , clickable and accepts input data
- T36** Search box - search box has (X) button to clear inserted data
- T37** Search box - when the user clicks on enter button or search icon the system should redirect user to the result page
- T38** Search box - if the user entered wrong data in search box the system should pop up an error message (no results are found)
- T39** Menu bar - menu bar search box is displayed in top left corner and clickable
- T40** Menu bar - when user click on menu it displays a list with options
- T41** Menu bar - when user click on any option from menu list the system should direct the user to the suitable page
- T40** Nearby restaurants- Nearby restaurants button is displayed and clickable in the header
- T41** Nearby restaurants - when the user clicks on the button the system directs him to all Nearby restaurants page
- T41** Nearby restaurants - the user can select any restaurant from listed Nearby restaurants
- T42** offers - Nearby restaurants button is displayed and clickable in the header
- T43** offers - when the user clicks on the button the system directs him to all offers page
- T44** offers - the user can select any offer from listed offers

3.4 Non-functional Requirements

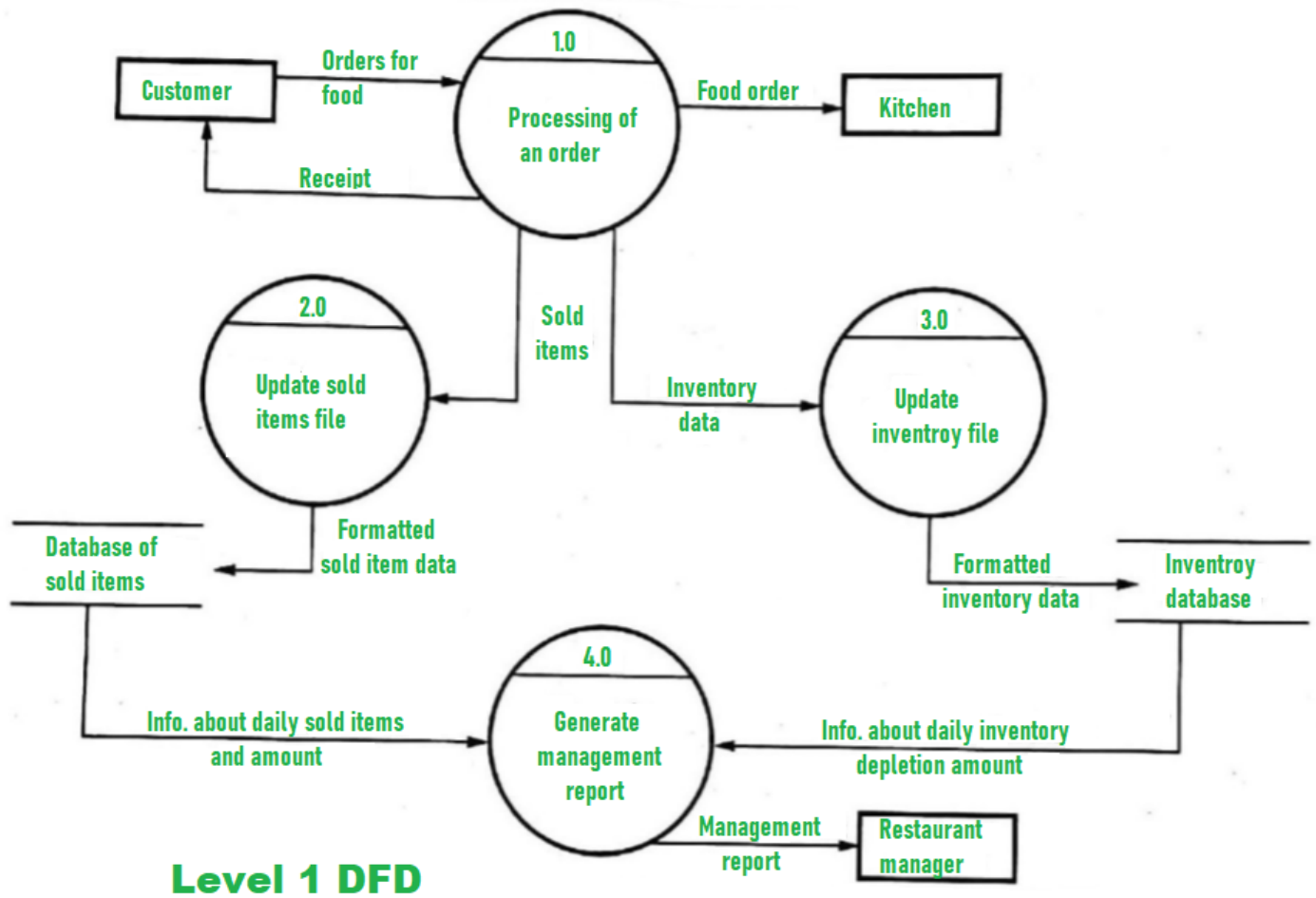
Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transactions shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).

- Performance
- Reliability
- Availability
- Security
- Maintainability
- Portability

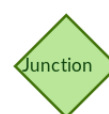
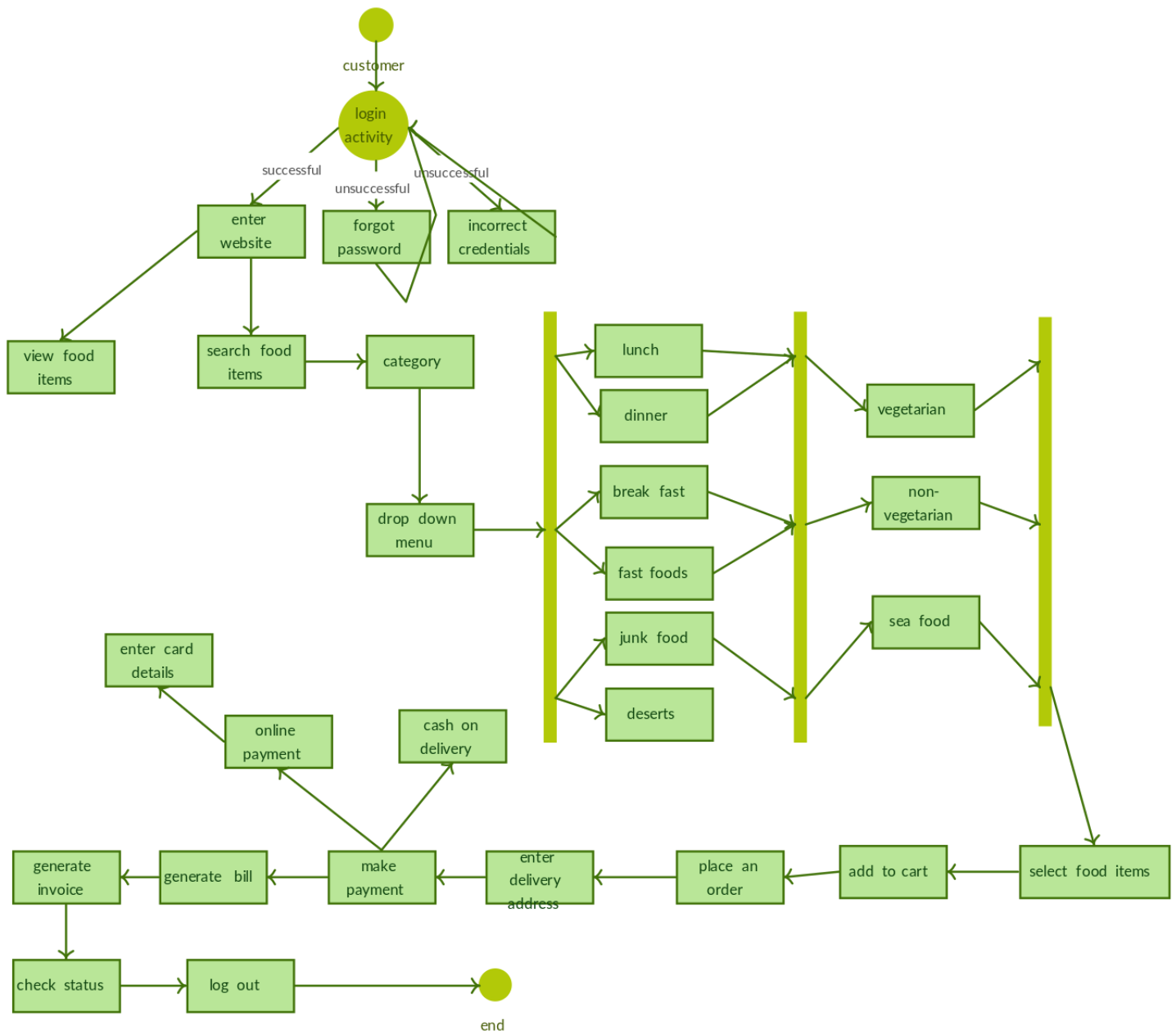
4. Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS's requirements.

4.1 Data Flow Diagrams (DFD)



4.2 State-Transition Diagrams (STD)



5. Change Management Process

Changes to the SRS either from the development, testing team or the client side will be communicated to the project sponsor Mr Omar Bakr.

Any change made to the SRS will require a sign off from the Development lead , QA lead and the client.

Once approved changed will be made to the SRS and the new SRS will be circulated to all stakeholders