Software Requirements Specification

for

TopShelf

Version 1.0 approved

Prepared by Jackson Dumas, Jacob Damon, Zachary Ellis, Nathaniel Barrett, and Alex Kokkosoulis

CS3773-002 Group 8

10/10/2019

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this document is to give a detailed description of the TopShelf (version 1.0) software customer portal system, where a customer can place orders for grocery items and have them delivered to their home address. It will also illustrate both the software (internal or external) and hardware requirements for the system. This document will also provide the necessary information to allow for the client to approve of design of the software. Additionally, it will offer the developers a reference should there be any confusion about the implementation of the system.

## Document Conventions

This document’s description of functional priorities is to be interpreted as inherited. Specifically, a higher-level requirement is dependent on the requirements of its parts or details. Additionally, this document is to contain several naming conventions and typographical standards to enhance the understanding of both the client and the developers. The conventions and standards are as follows:

|  |  |
| --- | --- |
| **Bold font** | Bold font signifies a section in the document. Bolded text indicates a change in section as well as a title for its respective section. |
| Italic Text | Italic text indicates information that was not originally written by the authors of this document. |
| Underlined text | Underlined text indicates a title or header for a piece of information. This is to separate the significance of bold font (used for sections) and the title of a sub-section. |
| Highlighted text | highlighted text indicates information that is essential to the section which it contained. |
| User | a User is defined as a person that interacts with the mobile application described in this document. |
| Admin | a person that is given special permission to control or manage the system in some way. |
| Product | The physical product that is being received by the customer. In this instance, “Product” refers to grocery items. |
| Mobile App | the customer portal sub-system that this document describes. The software that is operated on a mobile phone or other mobile device. |
| Shopping Cart | The list of currently selected products to be purchased and delivered. |
| Checkout | The digital representation of a transaction. Checkout indicates the process of purchasing the products contained in the cart. |
| Delivery Window | The time frame in which the groceries are to be delivered. This is to be determined during the checkout process by the user. |
| ID | The identification tag for this requirement or element. This tag is separate from the details of the requirement, as it is to be used as a convenient identifier. |
| TITLE | The title of a piece of information. Indicates a description and purpose, as described below. |
| DESC | The description of this information. Furthered by the purpose and dependency, as described below. |
| PURP | The reason, or rational for this element. The purpose indicates the intent of this piece of information. |
| DEP | A dependency of the described information. A requirement or piece of information that is either extremely pertinent or required for this element. |
| ACTORS | Used in the context of a use case entry. This tag describes the entities that will be interacting with the application |

## Intended Audience and Reading Suggestions

This document is intended for several parties. These parties include but are not limited to: the client, the developers, and the project managers. The information presented by this document will be pertinent to different parties, depending on the section, as different sections can contain either broad concepts or detailed technical descriptions. It would benefit the reader to read the table of contents, followed by the overview sections (1.1 – 2.2), and then any sections identified in the table of contents that contain value to the reader.

## Product Scope

This software or mobile-app is to be used by a customer in order to place orders for grocery delivery. The user will be able to select items from their preferred grocery store, add those items to their cart, and purchase the products, providing an address and time for the products to be delivered. The system will allow a complete bypass of the typical process of obtaining weekly groceries. Where a user may have had to partition time to go to the grocery store, they can now have them delivered to their home whenever convenient. For a more detailed description of the user actions described above, see section 2.2.

## References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

# Overall Description

## Product Perspective

This app is new, self-contained product designed to give users the best experience when shopping through various grocery stores.

Total ACTORS: Visitor -> Customer, Database, Cart, Communicator (for submitting orders)

* ID: UC-1

TITLE: Create Profile

DESC: Upon loading the app, there needs to be 2 options, login, or need a new account. If the user does not already have an account, they need to create one, the fields in the account need to include, name, address, credit card info, prompt for username and password

ACTORS: Visitor, Database

* ID: UC-2

TITLE: Login

DESC: Users login credentials should be verified, upon verification, they should be taken to the “Home” screen of the shopping app

ACTORS: Customer, Database

* ID: UC-3

TITLE: Select Store

DESC: Shoppers need to be able to select which store they want to shop from, or browse through

ACTORS: Customer, Database

* ID: UC-4

TITLE: Search

DESC: Shoppers should be able to search for an item by name, or type of item, through any store

ACTORS: Customer, Database

* ID: UC-5

TITLE: Add to Cart

DESC: Once a shopper finds an item they wish to purchase, they should be able to click a button and add it to cart

ACTORS: Customer, Cart

* ID: UC-6

TITLE: View Cart

DESC: Customers should be able to view their current cart at any time

ACTORS: Customer, Cart

* ID: UC-7

TITLE: Manage Cart

DESC: Customers should be able to remove items from their cart if they no longer wish to purchase them, this should be done from either from the cart screen or shopping screen

ACTORS: Customer, Cart

* ID: UC-8

TITLE: Checkout

DESC: The user should be able to navigate to their cart and hit the checkout button, charging their card and submitting a ticket for delivery

ACTORS: Customer, Communicator

* ID: UC-9

TITLE: Add Credit Card

DESC: There should be a method to manage the card on file, users should be able to add or remove cards linked to their account

ACTORS: Customer, Database

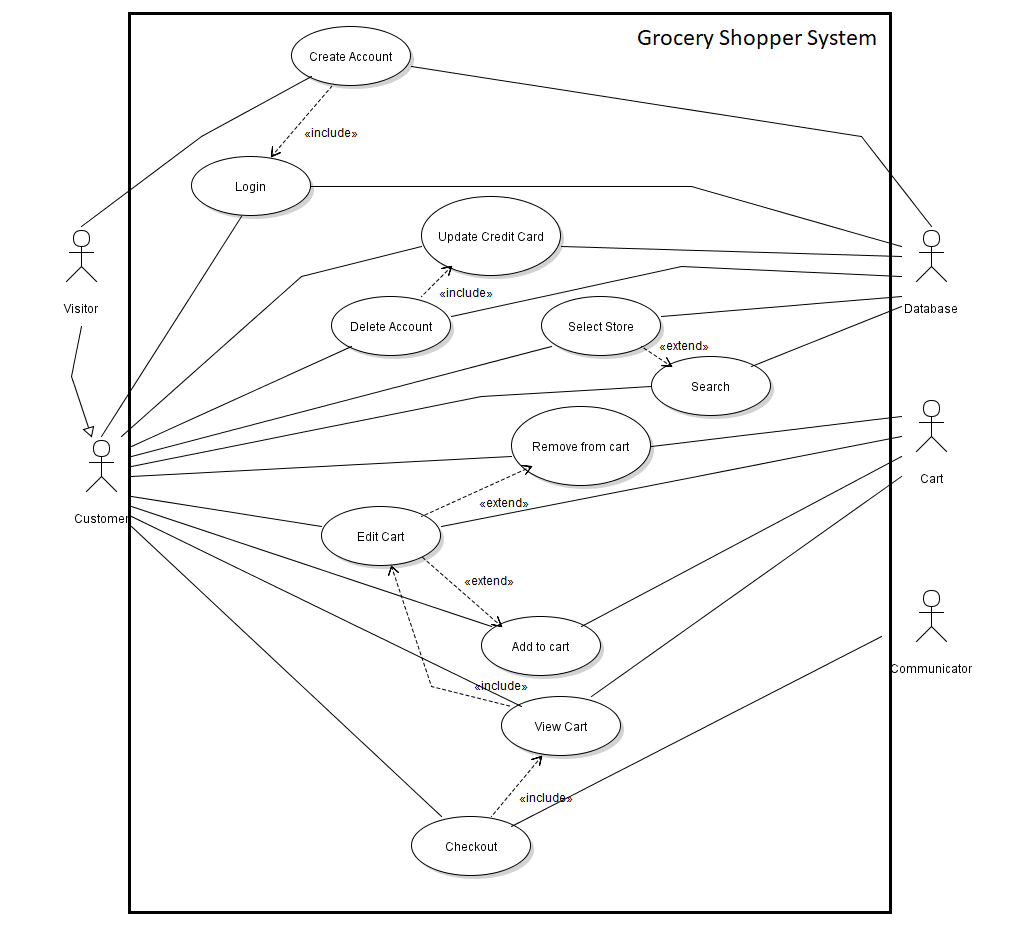
* ID: UC-10

TITLE: Delete Account

DESC: Users should have the option to delete their account

ACTORS: Customer, Database

**Use Case Diagram:**



## Product Functions

* ID: HLr1

TITLE: Choose Grocery Store

DESC: The mobile-app will be allow the user to place orders from a specific grocery store, with the ability to view that store’s specific inventory.

PURP: Offer the user a greater influence over their product selection

DEP: None

* ID: HLr2

TITLE: Personal Account

DESC: The mobile-app will allow the user to view their personal account information as well as modify it easily.

PURP: To offer the user the ability to customize their experience

DEP: Database Access

* ID: HLr3

TITLE: Persistent Order History

DESC: The user will be able to view their order history, including the payment information, the address that was used, and the products ordered.

PURP: To allow the user to maintain their trust of the application.

DEP: Database Access

* ID: HLr4

TITLE: Database Access

DESC: The user will be able to save their data and access store/product information via a database.

PURP: To offer the user their information on any device, and to lighten the application.

DEP: An internet connection.

* ID: HLr5

TITLE: Driver Tracking

DESC: The user will be able to view the current status of their order, including the current location of their delivery driver.

PURP: To inform the user of the status of their order

DEP: None

* ID: HLr6

TITLE: Personal Offers

DESC: The user will be able to log into their personalized account, including accessing personalized discounts and offers.

PURP: To offer the user a customized, personal experience

DEP: None

* ID: HLr7

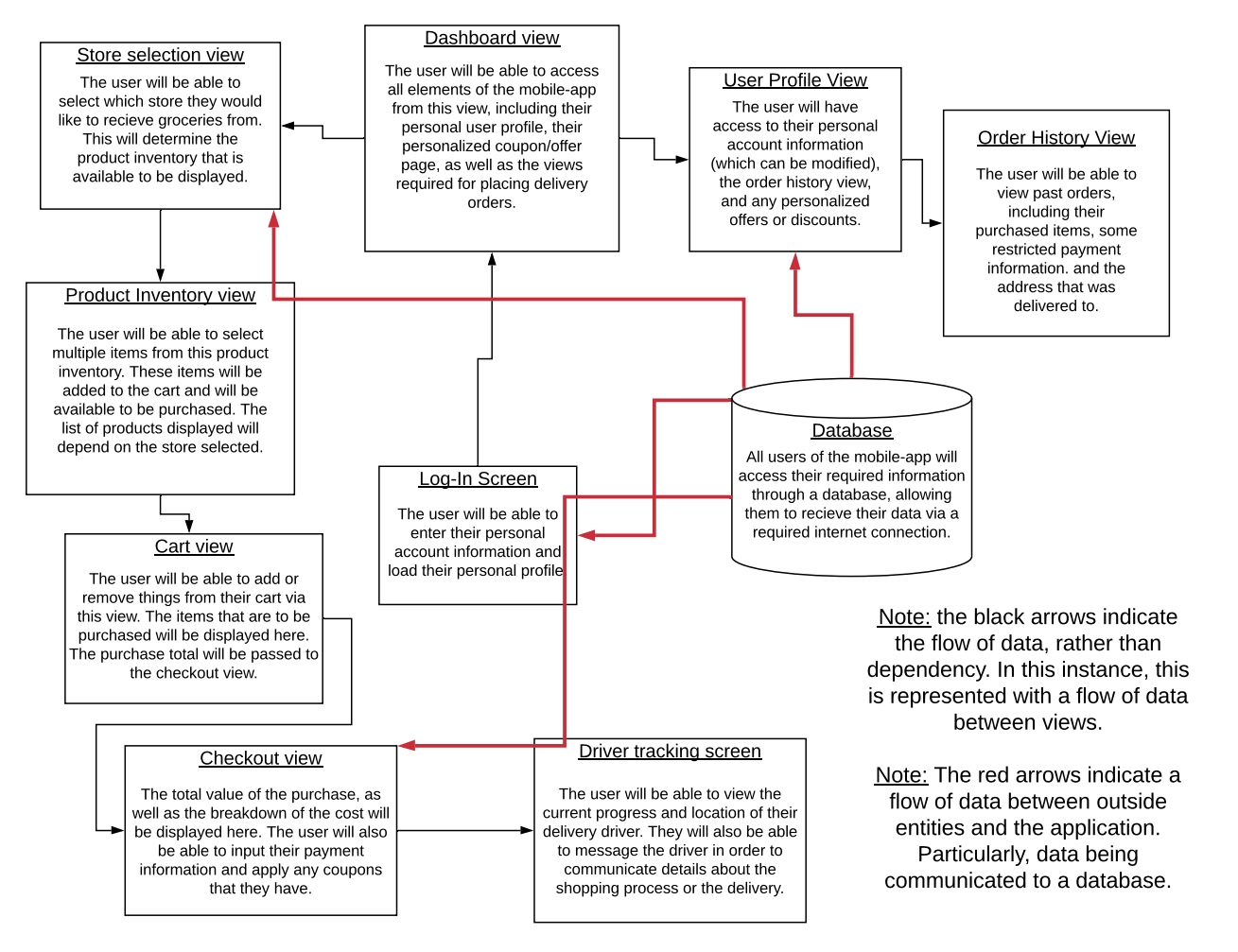
TITLE: Payment

DESC: The user will be able to input their payment information, as well as save it for convenience.

PURP: To allow the user to pay for their orders and customize their payment.

DEP: Database access

**High Level Data Flow Diagram:**



## User Classes and Characteristics

**General User** – this user-class can be described as a customer, or any person that does not possess administrative authority. Anyone that is placing delivery orders and purchasing products is likely a general user. The intention of this mobile-app is not to restrict users to any education level or experience. General users are designated as the most important user-class, although they possess the least administrative authority. The health of the mobile-app is positively correlated with the happiness of this user-class.

**Store Representative** – a store representative is person that possesses authority pertinent to a grocery retailer. This may include releasing coupons on a specified date, adding/removing products from the store inventory, and publish advertisements. Additionally, there will be some necessary experience in order to achieve this level of authority. This may include training or verification of some kind.

**Admin** – admins are responsible for maintaining the system itself. More specifically, they are responsible for ensuring the health of the user-base by allowing user refunds as well as removing false or suspicious accounts. They are likely the most frequent user and require some technical experience and education in order to maintain the functionality of other user-classes. Not all admins will have the same administrative power. Some admins will have access to functionality that others do not.

## Operating Environment

TopShelf is designed to be operated by shoppers on mobile devices, allowing users to shop while on the go. The supported platforms with which this application will operate include: Android(4.4/API 19 or later), as well as iOS (v9 or later). These system requirements are set by the Xamarin framework, which will be used for the development of the application. Additionally, TopShelf will require a touchscreen, as well as a functional GPS system. Supported devices include but are not limited to: iPhone(v5 or later), iPad(v2 or later), Samsung Galaxy(S4 or later), and Google Nexus(v5 or later).

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## System Feature 1

<Don’t really say “System Feature 1.” State the feature name in just a few words.>

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## System Feature 2 (and so on)

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

## Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>