Software Requirements Specification

for

<Project>

Version 1.0 approved

Prepared by Hudson Hargrove, Matthew Crosno,

Andrea Ambrose, Name

Department of Computer Science, Mississippi State University

26 August 2024

1. Introduction

1.1 Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>
This Software Specifications and Requirements (SRS) document provides the requirements specification for all implemented systems featured in the release of [NAME]Version 1.0, an online shopping platform built on principles of minimalism and simplicity. The purpose of this document is to define all software features, functions, and nonfunctional requirements associated with Version 1.0, developed in accordance with the stakeholders' preliminary high-level requirements as defined in Reference [2]. The document pertains only to systems and subsystems for Version 1.0, any subsequent releases will include updated SRS documents.

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

This document uses a tiered section format, with primary section titles written in bold, size eighteen font with a numerical format that follows the scheme 'X. Title', where X represents the section number, and Title represents the title of the section. Subsections are conveyed using the 'X.Y Title' scheme, where X represents the section number, Y represents the subsection index, and Title represents the subsection title. Unless otherwise stated, all detailed requirements inherit the priority of the encompassing high-level requirements. All acronyms are defined in Appendix A: Glossary.

1.3 Intended Audience and Reading

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

The SRS is intended to serve as a guide for developers, users, testers, and documentation writers. While each subgroup of the intended audience may find unique value in the
information contained herein, it is suggested that all readers begin by becoming familiar with 'Section2: Overall Description', which provides a high-level overview of the software release and its functions,classes, and constraints. Users, and documentation writers will also find the information

contained in ‘Section 3: System Features’ to be helpful for understanding the capabilities and features available in [NAME], Version 1.0. Developers should consult Sections 3, 4, and 5 as a guide to workflow formation, as these sections provide insight into the specified system features and requirements. Those tasked with testing the software should read Sections 3, 4, and 5 to develop test strategies that ensure all requirements are satisfied for each feature.

1.4 Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

[NAME] is an online shopping app designed around the principles of minimalism and user-centric simplicity. Unlike sprawling e-commerce platforms that overwhelm users with ads, endless widgets, and relentless upselling, [NAME] offers a refreshing alternative focused on ease of use and genuine engagement.

The app features a clean, intuitive interface that makes navigation effortless. Instead of presenting users with an overwhelming array of choices, [NAME] provides a curated selection of high-quality products, streamlining the decision-making process. This curated approach ensures that users find what they need quickly and confidently.

In line with its minimalist ethos, [NAME] delivers an ad-free experience, eliminating distractions and keeping the focus on the products themselves. Personalized recommendations are based on user preferences, avoiding manipulative tactics often seen in other platforms.

Privacy is a cornerstone of [NAME]'s design. The app limits data collection and maintains transparency about data usage, empowering users with control over their information. Additionally, [NAME] emphasizes ethical shopping by featuring products from responsible brands, allowing users to make conscientious choices.

By prioritizing a minimalist, user-first approach, [NAME] sets itself apart from current e-commerce giants. It represents a thoughtful evolution in online shopping, focusing on user satisfaction and well-being rather than engagement metrics and commercialism.

1.5 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.>

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5 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).>

3. 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.>

3.1 Account Management

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 Description and Priority

Implement a login window that contains a box for a user to input both a username or email as well as a password. Also implement functionality for the user to create a new account if one does not exist.

Priority: High

4.1.2 Stimulus/Response Sequences

Trigger: User inputs account information

<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:

3.2 System Feature 2 (and so on)

4. Other Nonfunctional Requirements

4.1 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.>

4.2 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.>

4.3 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.>

4.4 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.>

5. 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.>

SRS
- System Requirements and Specifications - Document that defines all system features, functions,nonfunctional requirements, and classes in order to convey the objectives of the software and the manner in which the software will satisfy those objectives.

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.>

Project Name
Project Name