Software Requirements Specification

for

ShoeBae

**Version 1.0 approved**

**Gage Fulwood - Team Leader**

**Citlali Hernandez Lopez**

**John Mark Taylor**

**Yusuf Sarigul**

**Easton Greenwood**

**Mississippi State University**

**Jan 1, 2023**

# Introduction

## Purpose

The purpose of this document is to provide an informative description of this E-commerce system. While providing our customers with the goals we hope to achieve, the document will also provide information regarding software specifications used in the development of **Shoebae 1.0**. This document will benefit each stakeholder of **Shoebae 1.0** by providing a better understanding of concepts that have been used, such as: product management, seller accounts, customer accounts, order management, payment processing, and security measures taken.

## Document Conventions

This document employs a hierarchical numbering system to organize sections and subsections. Sections are presented in bold, size eighteen font, using the format 'X.SectionName,' where 'X' is the section number. Subsections follow a similar convention, displayed in bold, size fourteen font as 'X.Y.SubsectionName,' where 'Y' represents the subsection number. Subsubsections are presented in bold, size 12 font as ‘X.Y.Z.SubsubsectionName’, where ‘Z’ represents the subsubsection number. Additionally, the sections themselves will be written in regular, size 12 font.

For the referencing of the Shoe Bae 1.0 SRS document, our team will be using MLA formatting, specifically the MLA 9th edition format. This format will allow for in-text citations and a works cited section (1.5 references).

## Intended Audience and Reading

This Software Requirements Specification (SRS) document is crafted to cater to a diverse set of stakeholders involved in the development, implementation, and maintenance of Shoebae 1.0. The primary audience includes:

**Developers:** To understand the technical specifications, coding requirements, and implementation details.

**Project Managers:** To gain insights into the project scope, timelines, and resource requirements.

**Marketing Staff:** To comprehend the features and functionalities that can be leveraged for promotional activities.

**Users:** To understand the capabilities and functionalities of the E-commerce system.

**Testers:** To identify testing scenarios and requirements.

**Documentation Writers:** To assist in the creation of user manuals and system documentation.

Readers are advised to follow a sequence that aligns with their specific interests and responsibilities:

**Overview Sections:**

Begin with sections providing an overview of the E-commerce system (e.g., Purpose, Product Scope).

Understand the goals, benefits, and objectives of Shoebae 1.0.

**Product Scope:**

Delve into the short description of Shoebae 1.0, its purpose, and how it aligns with corporate goals or business strategies.

**Overall Description:**

Explore the product perspective, understanding the context, and origin of Shoebae 1.0.

Identify major components through simple diagrams and understand the system's interactions.

**System Features:**

Examine the major functions and features the E-commerce system must perform.

Follow the functional requirements organized by system features.

**User Classes and Characteristics:**

Learn about different user classes (Unregistered, Registered, Buyer, Seller, Admin) and their respective privileges.

**Operating Environment:**

Understand the hardware platform, operating systems, and other software components necessary for Shoebae 1.0.

**Design and Implementation Constraints:**

Identify any limitations or constraints that may impact development options.

**Other Requirements:**

Check for any additional requirements not covered in other sections.

**Nonfunctional Requirements:**

Review performance, safety, and security requirements.

**Appendices:**

Refer to the Glossary for definitions of key terms.

Explore Analysis Models (if included) for visual representations.

Check the To Be Determined List for pending items.

## Product Scope

The web application ShoeBae provides consumers with a hub that other users use to buy, sell, and trade goods, primarily shoes. The purpose of this application is to offer users a database of goods sold by other consumers, providing a convenient way to shop and a platform to sell goods at a price of their choosing. Keeping up with modern standards, this platform allows users to filter through content using various forms of characterization, such as type, color, and size. It also provides the opportunity for people to sell their goods on our platform, thanks to our user-friendly design that makes setting up listings easy and simple.

This design ensures that the software and its features are as simple as an average consumer scrolling through, finding an item, and adding it to a cart or adding a listing while waiting for an offer. Our platform is a place for goods to be widely viewed by a large audience, facilitating purchases. In more detail, as a user, you will sequentially create an account with a username and password, and then log in. You can search through our platform with tools to assist your search, such as filtering through items by type, color, or size. If you find an item, you can easily add it to your cart and complete the process by checking out.

For sellers, the process is similarly simple, with a few additional features to make listing their item hassle-free. The process prompts users with questions about their product to seamlessly categorize it according to their desired specifications within our system. Once these questions are answered, they can list the item and wait for offers or trades to arrive in their inbox for consideration.

Should users decide to sell their goods, the domain owners will take a small commission. This fee is intended to provide continuous support, keeping the web application operational and ensuring it doesn't discourage users from selling their goods. The fee is kept relatively low to maintain user satisfaction and encourage continued engagement on the platform. To further support the application, ads relevant to users' searches will be implemented, alongside the continuous support of developers and investors, all contributing to the website's quality and reliability.

## References

1. *Judy, Tirian, et al. “Software Requirements Specification for Y.” Canvas, Mississippi State University, 10 Sept. 2023,* [*https://canvas.msstate.edu/courses/121638/files/10040003*](https://canvas.msstate.edu/courses/121638/files/10040003)
2. *Pressman, Roger S., and Bruce R. Maxim. Software Engineering: A Practioner’s Approach. 9th ed., McGraw-Hill, 2020.*

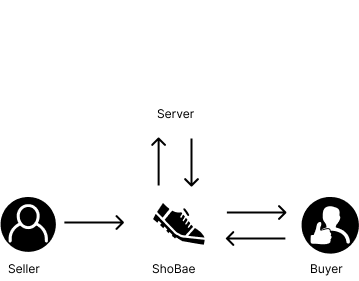
# Overall Description

## Product Perspective

ShoeBae is a website designed to sell to college students seeking quality and affordable footwear. It’s a website that takes inspiration from websites like StockX, Goat, and eBay. ShoeBae will only sell shoes and nothing else as of this moment, but will be able to expand and when the opportunity arises. ShoeBae is designed to fullfil the project credit for the class Introduction to Software Engineering (CSE 4214 / 6214) at MSU.

The usage of ShoeBae are limited to our knowledge and requirements set by the class scope.

Our purpose in creating this website is to acquire the skills necessary to implement various software requirements of an ecommerce website for later use in our workplace. ShoeBae has features like dynamic bidding which engages the user while still ensuring reasonable prices. ShoeBae ensures the authenticity and quality of the products to guarantee our users a standard shopping experience. Prospective sellers will undergo a meticulous vetting process where their character and products will be analyzed in order to make sure they are a perfect fit for our website’s goal — quality products at an affordable price. Ensuring quality goods with sellers with outstanding character allows our users to be able to feel safe in buying our products with high satisfaction.



## Product Functions

Shoebae 1.0 will provide various functions including:

* **Maintain data associated with inventory (shoes)** 
  + each item will be assigned a title, description, and price
  + Shoebae 1.0 will keep a quantity count for each item number

* **Allow access to a limited number of functions for guest, buyers, and sellers**
  + Guest will be limited to search and view functions
  + Buyers will have the ability to login, logout, search items by number, and return items
  + Sellers will have the ability to login, logout, add items, remove items, update pricing on items, add item title, update item title, add item description, update item description, search item number, view item, compare prices on items, update inventory, add item thumbnail, remove item thumbnail, process buyer orders, approve exchange requests, return approval, and refund buyer funds.
  + Admins will have access to all functions for buyers and sellers, ability to flag items, delete buyers, delete sellers, remove items, override exchanges, and override refunds.
* **Shopping cart**
  + Shoebae 1.0 will have a shopping cart function with the purpose of keeping track of items a buyer plans to purchase.
* **Payment Processing**
  + Allows sellers and admins to process payments when provided a 16 digit card number, card holder name, zip code associated with the card, and the correct security code.
  + Shoe bae 1.0 will only allow the use of credit cards.

## User Classes and Characteristics

1. **User**

**Description:** The User class encompasses each user type that will interact with our ecommerce platform. User privileges and engagement capabilities vary depending on registration or administrative status.

* 1. **Unregistered**

**Description:** The Unregistered subclass includes the User class, and is the default user type on the platform. This user can only view and browse product listings, though they have the option to upgrade their account status to registered.

* 1. **Registered**

**Description:** The Registered subclass includes the User class, and encompasses users that have a registered account.

* + 1. **Buyer**

**Description:** The Buyer subclass extends the Registered subclass, and encompasses users that would like to purchase listed products on the platform.

* + 1. **Seller**

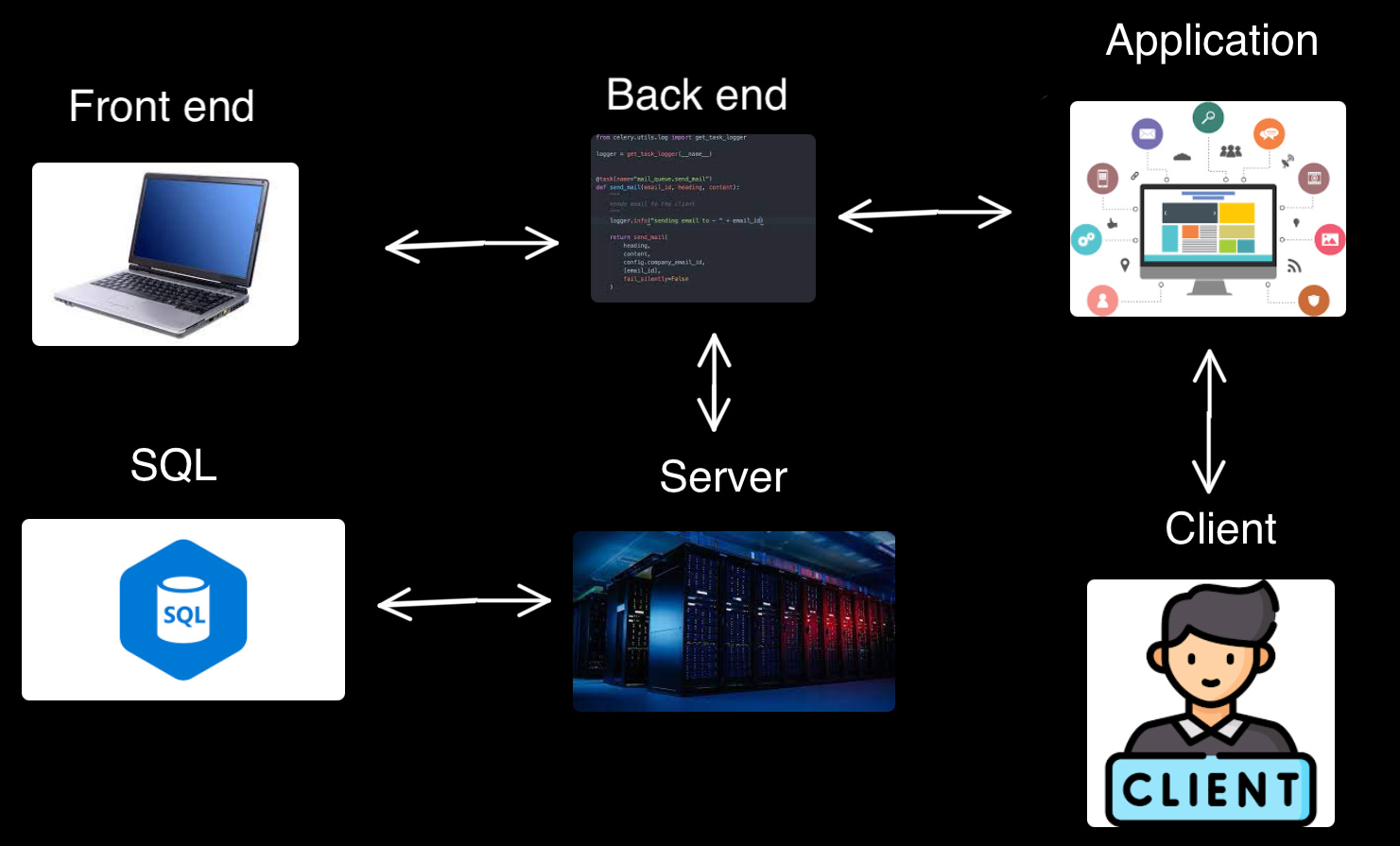
**Description:** The Seller subclass extends the Registered subclass, and encompasses users that would like to list products for sale on the platform.

* 1. **Admin**

**Description:** The Admin subclass extends the User class, and consists entirely of users with administrative privileges to oversee the platform.

## Operating Environment

ShoeBae is poised to be a web platform employing HTML, CSS, and JavaScript for its front-end development, while Python will drive its back-end functionalities. It is designed to be compatible with Windows 10+, MacOS, iOS, and Chrome-based browsers. The database infrastructure will be built using SQL technology.



**Image References**

1. *computer. Encyclopædia Britannica, Encyclopædia Britannica, https://www.britannica.com/technology/computer/Supercomputer#/media/1/130429/231796. Accessed 2 Feb. 2024.*
2. *Frederic. Desktop and its interactions . 17 July 2020. Web Application Development: The BASIC Concepts, Shourai, https://shourai.io/wp-content/uploads/2020/07/kindpng\_1272110.png. Accessed 2 Feb. 2024*
3. *Freepik. client icons. Flaticon, https://www.flaticon.com/free-icon/client\_6009864. Accessed 2 Feb. 2024.*
4. *Hunter, Alex. client icons. 16 Oct. 2020. What Is a Dedicated Server?, Parallels , https://www.parallels.com/blogs/ras/app/uploads/2020/10/dedicated-server-bg.jpg. Accessed 2 Feb. 2024.*
5. *Sharma, Sajal. segment of code in python. 23 Apr. 2020. freeCodeCamp, freeCodeCamp, https://www.freecodecamp.org/news/content/images/2020/04/Screenshot-2020-04-20-at-5.26.51-PM.png. Accessed 2 Feb. 2024.*
6. *SQL icon. Azure SQL Database Monitoring , IBM, https://www.ibm.com/content/dam/adobe-cms/instana/media\_logo/Azure-SQL-Database-Monitoring.component.complex-narrative-xl.ts=1688044945966.png/content/adobe-cms/us/en/products/instana/supported-technologies/azure-sql-database-monitoring/jcr:content/root/table\_of\_contents/body/content\_section\_styled/content-section-body/complex\_narrative/logoimage. Accessed 2 Feb. 2024.*

## Design and Implementation Constraints

**Corporate and Regulatory Policies:**

Data Protection Regulations:

* Compliance with regional and international data protection laws (e.g., GDPR, CCPA) is mandatory.

Security Standards:

* Adherence to company-specific security standards and protocols is required to ensure data integrity and user privacy.

**Hardware Limitations:**

Minimum System Requirements:

* The application must operate within specified minimum hardware requirements to ensure optimal performance for end-users.

**Interfaces to Other Applications:**

Third-Party Integrations:

* Integration with payment gateways, shipping providers, and other third-party services must comply with API specifications and security standards.

**Technologies and Tools:**

Technology Stack:

* Use of a predefined technology stack, including specific programming languages, frameworks, and databases, as outlined in the project guidelines.

Development Tools:

* Utilization of specified development tools and environments to ensure consistency across the development team.

**Security Considerations:**

Authentication Mechanisms:

* Implementation of robust user authentication mechanisms to enhance platform security.

Data Encryption:

* Encryption of sensitive data during transmission and storage to mitigate security risks.

**Design Conventions:**

User Interface Guidelines:

* Adherence to established user interface design conventions for a seamless and user-friendly experience.

**Programming Standards:**

Coding Standards:

* Conformity to coding standards and best practices for maintainability and collaborative development.

**Communication Protocols:**

Network Communication:

* Specification of communication protocols for secure data transmission over networks.

**System Maintenance:**

System Updates:

* Provision for future system updates and enhancements to accommodate changing business requirements.

**Language Requirements:**

Multilingual Support:

* Incorporation of multilingual support to cater to a diverse user base.

**TBD (To Be Determined):**

Any additional constraints yet to be determined will be specified in subsequent project phases.

# 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.>*

# 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.>*