<Team G>

grade: 55
comment: a failing report. the use-case diagram is wrong and incomplete, failing to reflect the overall picture of the system. this is an unacceptable report 1, view this grade as a serious warning to the team. hope the team can shape up in future reports/system building.

<Easy Shop>
Software Requirements Specification
For <Subsystem or Feature>

Version <1.0>

[Note: The following template is provided for use with the Rational Unified Process. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

Revision History

Date	Version	Description	Author
<17/10/2024>	<1.1>	< Completed the first report which included sections 1 - 4>	<Florian,Edison Enmanuel, Padilla,Emmanuelle, Omit,Md Mazharul Islam, Rodriguez,Carlos A, Perez,Misael>

Table of Contents

1. Introduction 4
   - 1.1 Purpose 4
   - 1.2 Scope 4
   - 1.3 Definitions, Acronyms, and Abbreviations 4
   - 1.4 References 4
   - 1.5 Overview 4

2. Overall Description 4
   - 2.1 Use-Case Model Survey 5
   - 2.2 Assumptions and Dependencies 5

3. Specific Requirements 5
   - 3.1 Use-Case Reports 5
   - 3.2 Supplementary Requirements 5

4. Supporting Information 5

Software Requirements Specification

1. Introduction

*[The introduction of the **Software Requirements Specification (SRS)** provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the **SRS**.]*

sloppy: should delete these and fill in your own introduction.

*[Note: The **SRS** captures the complete software requirements for the system, or a portion of the system. Following is a typical **SRS** outline for a project **using use-case modeling**. This artifact consists of a package containing use cases of the use-case model and applicable Supplementary Specifications and other supporting information. For a template of an **SRS not** using use-case modeling, which captures all requirements in a single document, with applicable sections inserted from the Supplementary Specifications (which would no longer be needed), see the file titled rup_srs.dot.]*

*[Many different arrangements of an **SRS** are possible. Refer to [IEEE93] for further elaboration of these explanations, as well as other options for an **SRS** organization.]*

1.1 Purpose(Misael

*[Specify the purpose of this **Software Requirements Specification**. The **SRS** fully describes the external behavior of the application or subsystem identified. It also describes nonfunctional requirements, design constraints, and other factors necessary to provide a complete and comprehensive description of the requirements for the software.]*

your own purpose

Login/Account:

The purpose of the SRS document is to explain the capabilities of the application. Our application is ebidding, which is a parody of the ebay website. Our application will be a website and a mobile application. The software will allow the application to make an account with their email address, and passwords. For security purposes, the website/mobile app will ask for your phone number.

After making your account, it will ask you to login again. Only five attempts are allowed before a text will be sent to the user about the attempts.

Bidding:

There will be a search bar to search for a specific item or something similar. On the side of the screen, there will be an option to click on the type of item you want. For example, “electronics” then an option for different types of electronics. The bidding starts when you have an item that you have found, and are trying to start bidding. First, you have to have your payment information, which is a credit card, debit card, or pay pal. If you don’t already have a payment method in your account, it will ask you to type in your information. Once that is done, you are ready to start bidding. There is a option to view all the pervious bids of other users, and the option to bid. We will come up with more features later.

Non-Functional requirements:

- Encrypted passwords
- Search time will be 1 second.
- Loading time 1 second

Design Constraints:

- At least 8 GB of RAM
- At least 15 GB of Storage
- Using Python, Javascript
- GDPR regulations will be implemented

Assumptions:

We assume that the user will be an adult of at least 18 years of age. They have full access to the internet and have an internet connection. This adult will have at least a credit card or debit card to be able to purchase items on the website/mobile app.

1.2 Scope(Misael

*[A brief description of the software application that the **Software Requirements Specification** applies to, the feature or other subsystem grouping, what Use-case model(s) it is associated with, and anything else that is affected or influenced by this document.]*

The main features that we will be focusing on are managing the user's account, encrypted password information, and managing the payment process. We will be managing a sum of users for our software. In addition, we will protect the user's information. Lastly, we will handle the payment process using the user's credit/debit card information.

Use-Case Model:

A user, which has made an account with us, will log in to view a specific item they wants to purchase. They will see the bidding price and see the number of bidders on the item. They will then enter the amount of money they are willing to bid.

1.3 Definitions, Acronyms, and Abbreviations (Md)

*[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the **Software Requirements Specification**. This information may be provided by reference to the project's Glossary.]*

SRS: Software Requirements Specification

UI: User Interface

1.4 References

*[This subsection provides a complete list of all documents referenced elsewhere in the **Software Requirements Specification**. Identify each document by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]*

- invaluable.com

1.5 Overview

*[This subsection describes what the rest of the **Software Requirements Specification** contains and explains how the document is organized.]*

The platform is designed to provide buyers and sellers with a seamless experience for purchasing and selling products. It specifically targets individuals interested in acquiring products through an online bidding process.

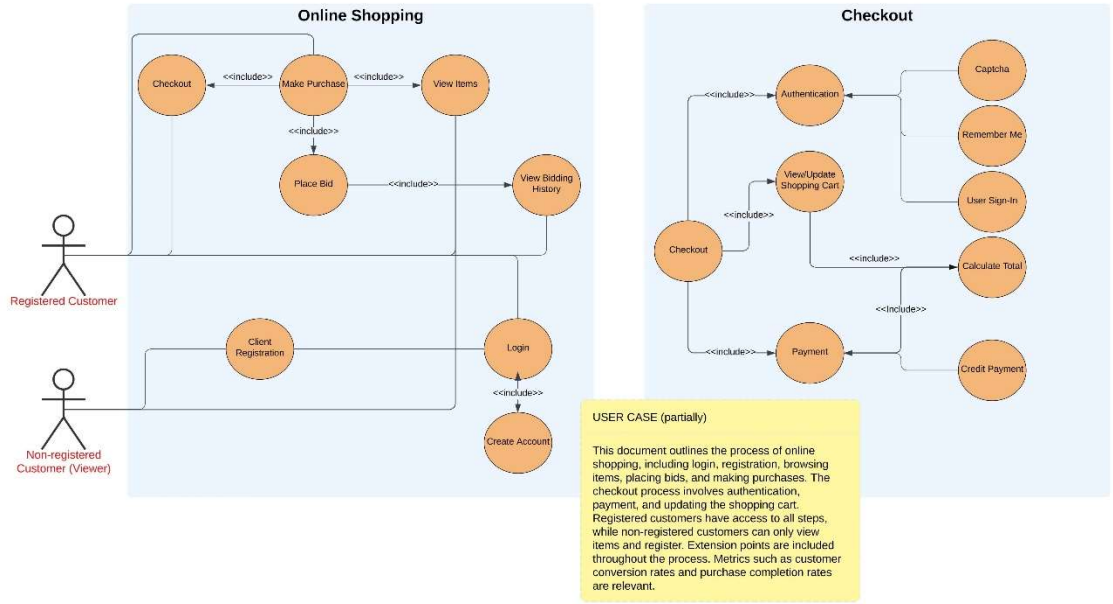
2. Overall Description

*[This section of the **Software Requirements Specification** 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 Section 3, and makes them easier to understand. Include such items as product perspective, product functions, user characteristics, constraints, assumptions and dependencies, and requirements subsets.]*

2.1 Use-Case Model Survey

*[If using use-case modeling, this section contains an overview of the use-case model or the subset of the use-case model that is applicable for this subsystem or feature. This includes a list of names and brief descriptions of all use cases and actors, along with applicable diagrams and relationships. Refer to the **Use-Case-Model Survey Report**, which may be used as an enclosure at this point.]*

more like a flow-chart than a use-case diagram



Actors:

Registered customers have full access to features such as browsing, bidding, managing their shopping cart, and completing purchases. Non-registered customers can only view items and must register to bid or buy. If they try restricted actions, they are prompted to register.

Registered users log in to access features, with authentication and an optional "Remember Me" feature. New users register by providing details and verifying their identity through CAPTCHA, ensuring human interaction.

All users can browse items, but non-registered users must register to place bids or make purchases. Registered users can place bids, with their bidding history automatically updated for tracking. They can also view and update their shopping cart.

Checkout:

The checkout process involves authentication and payment, with the system calculating the total and handling exceptions like payment verification. Once the checkout is complete, the system updates inventory and confirms the order. Registered users can also track their bids through their bidding history.

2.2 Assumptions and Dependencies (Md)

*[This section describes any key technical feasibility, subsystem or component availability, or other project related assumptions on which the viability of the software described by this **Software Requirements Specification** may be based.]*

Assumptions

Internet Connectivity*: The user has a stable internet connection.*

Device Compatibility*: The user should have an android device.*

Product*: The product is up to date and reflective of actual offerings.*

Payment security*: Food processing will start after the payment.*

Feedback and reviews*: The buyer gives reviews of the food after receiving the food.*

Dependencies:

Front-End Frameworks/Libraries*: React Native*

Back-End Frameworks*: Django, a python-based web framework running on a web server.*

Database Systems*: NoSQL database MongoDB*

Payment Processing Services*: PayPal*

User Authentication Services*: OAuth*

Mapping and Location Services*: Google Maps API for location-based features.*

Push Notification Services*: React Native component (FCM on android and APN on apple).*

Code Versioning Tools*: Git for version control*

Dependency Management Tools*: npm*

3. Specific Requirements

*[This section of the **Software Requirements Specification** contains all software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements and testers to test that the system satisfies those requirements. When using use-case modeling, these requirements are captured in the use cases and the applicable supplementary specifications. If use-case modeling is not used, the outline for supplementary specifications may be inserted directly into this section.]*

3.1 Use-Case Reports (Md)

[In use-case modeling, the use cases often define the majority of the functional requirements of the system, along with some non-functional requirements. For each use case in the above use-case model, or subset thereof, refer to, or enclose, the use-case report in this section. Make sure that each requirement is clearly labeled.]

- User account: User has to create an account to enter the application. Additionally, the user account has a user name, email and phone number
- Visitor account: Visitor can browse the listing of items.
- Super-users: Super-user has the ability to upgrade a visitor to a user.
- Search Functionality: The user can search the food.

3.2 Supplementary Requirements (Md)

[Supplementary Specifications capture requirements that are not included in the use cases. The specific requirements from the Supplementary Specifications, which are applicable to this subsystem or feature, should be included here and refined to the necessary level of detail to describe this subsystem or feature. These may be captured directly in this document or referred to as separate Supplementary Specifications, which may be used as an enclosure at this point. Make sure that each requirement is clearly labeled.]

1. System Compatibility: The app must be compatible with the latest versions of iOS and Android operating systems to ensure broad accessibility.
2. Data Privacy: The app must comply with relevant data protection regulations to safeguard user privacy.

3. Scalability: The backend infrastructure must be designed to scale horizontally to handle loads during special events or promotions.

4. Supporting Information

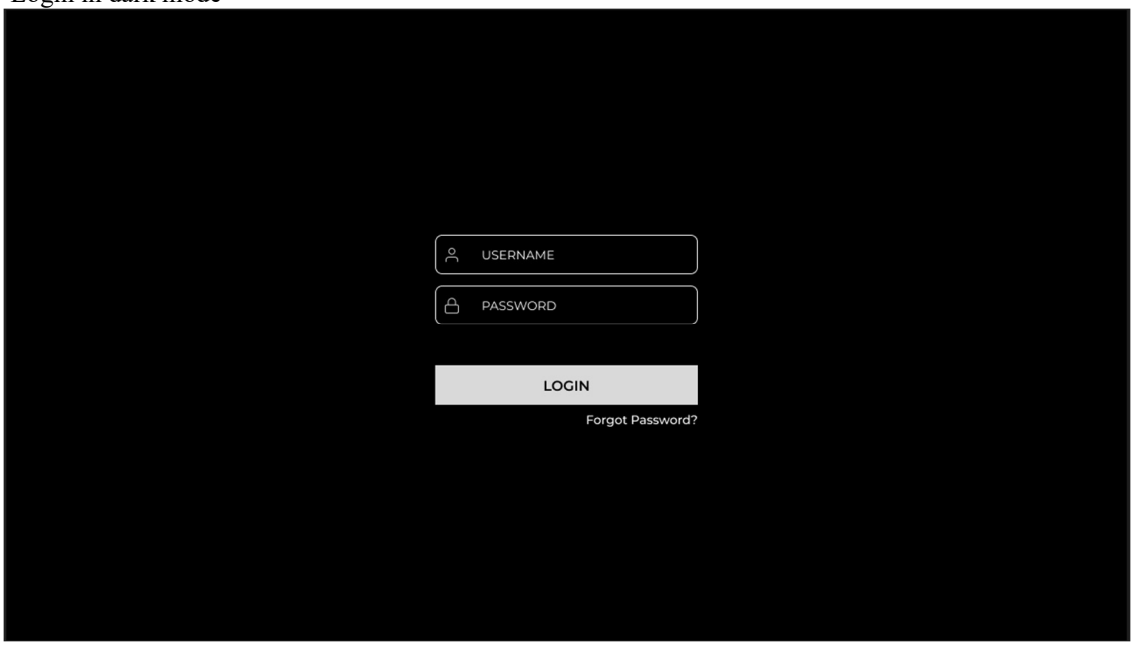
*[The supporting information makes the **Software Requirements Specification** easier to use. It includes:*

- *Index*
- *Appendices*

*These may include use-case storyboards or user-interface prototypes. When appendices are included, the **Software Requirements Specification** should explicitly state whether or not the appendices are to be considered part of the requirements.]*

USER-INTERFACE PROTOTYPES:
(These wireframes were created using Figma)

Login in dark mode



Homepage for e-bidding website

The user interface will feature a prominent display of the main item at the top of the page, accompanied by a high-quality image. Users will have the ability to cycle through multiple images of the item to view different angles and variations. Additionally, there will be an option for users to like the item, indicating their interest, and a clearly labeled button to add the item to their shopping cart for easy access during checkout.