**Software Requirements Specification**

**for**

**CSwap**

**Version 1.0 approved**

**Prepared by Caccese, Fahey, Thibodeau, Massari, Bourgoin**

**College Swap**

**2/14/2022**

**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** |
| --- | --- | --- | --- |
| Caccese, Fahey, Thibodeau, Massari, Bourgoin | 2/14/2022 | Initial Document Setup | 1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **Introduction**

## **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.>*

The application CSwap is an online local area marketplace designed to allow college students to have a centralized place to buy and sell college related items. The main objective of this application is to provide college students a way to buy and sell items relevant to their student life. The following document will describe the software requirements needed to create such an application.

## **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 was created based on the IEEE template for System Requirement

Specification Documents.

The following conventions in the document were used as follows:

| **Convention** | **Description** |
| --- | --- |
| DB | Database |
|  |  |
|  |  |
|  |  |
|  |  |

## **Intended Audience and Reading Suggestions**

*<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.>*

This document will be made for developers, project managers, finance staff, users, testers, marketing staff, and document writers.  The rest of this SRS contains Product Scope, An overall description of the product, External Interface Requirements, System Features, Nonfunctional Requirements, and Other Requirements.

## **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.>*

The Software will be a Backend as a service (Baas, a NoSQL database program). We are specifically using Firebase to create the application. It’s purpose is to act as an online local market place for students attending college. The software promotes the corporate goal of loyal customers. The average student is taking 5 years to graduate from college. The software is designed to appeal to students going from one semester/school year to another. The makes it easy to create loyal customers.

## **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.>*

CSwap Git Hub Link: [*https://github.com/JacquesLJT/COS420\_Project*](https://github.com/JacquesLJT/COS420_Project)

# **Overall Description**

## **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.>*

CSwap is being developed for everybody who is attending college. Any student that has books, electronics, or a lease that they don’t want can use it more specifically. CSwap is also being made for any student that is looking for these items. There is also a section for users that want to buy and sell leases. It is a new product being created with open source code.

## **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.>*

Account Creation

* Input Username: Allows the user to put in a username to log in.
* Input Password: Allows the user to put in a password to log in.
* Login with Google: Allows the user to log in with an existing Google account.
* Login with Facebook: Allows the user to log in with an existing Facebook account.

Sorting Posts

* Input Zip Code: Input the zip code of where you are selling the item.
* Select Category: Filters listings by type so only one type is visible.

Post Manipulation

* Post Listing: Allows the user to post a listing.
* Remove Listing: Allows user to remove a listing.

Class Manipulation

* Authorize user: Allows unauthorized users to become authorized users
* View Listing: Allows the user to view a listing’s details.

## **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.>*

## **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.>*

*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.>*

## **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**

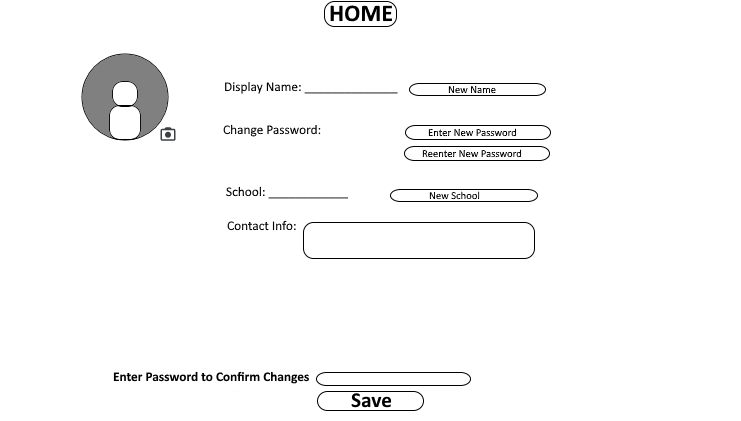
*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions 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).>*

CSwap is developed in HTML, CSS, and Java Script. We are assuming users are not blocking Java Script, if they are out application will not work on their device.

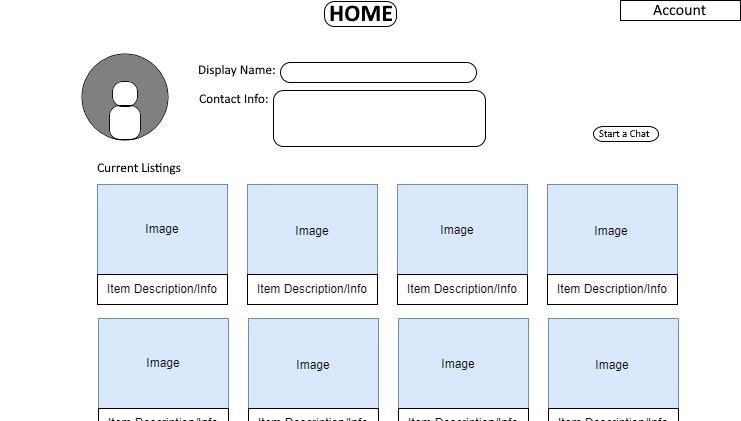
# **External Interface Requirements**

## **User Interfaces**

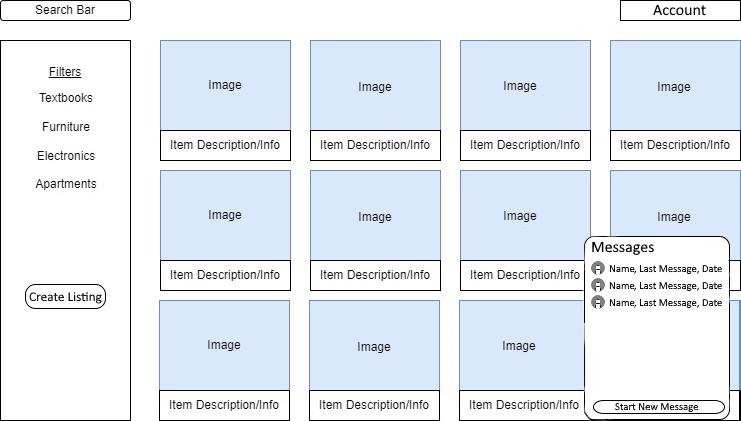
This is an example of what our front page would look like. It has a button (Account) that will open a dropdown menu that will have options to view your account page, edit your account information and log out. The search bar will allow you to search by the names of items or the ISBN for books. The Filters will allow you to filter by our four categories Textbooks, Furniture, Electronics, and Apartments. The Create Listing button will allow people looking to put create a listing to do just that. All our UI will be designed to work on Desktop and Mobile on the major web browsers.



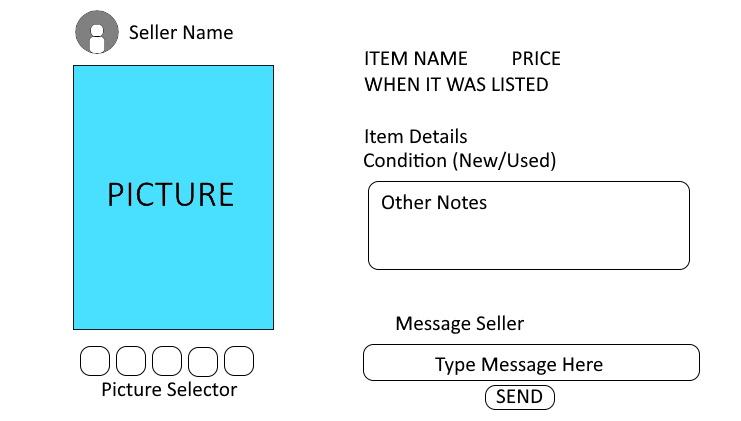
This is a basic look for what people's account pages will look like when they edit them. It will allow you to change your profile picture, display name, password, school, and contact information. To confirm changes you need to enter your current password otherwise it won't save for security purposes.



This is a basic look for what people's profiles will look like when people view them.



Additionally, on the bottom right corner of most pages, we will have a chat button which when clicked will open all chats that the logged-in user currently has. This will allow them to access their chats from most pages and keep up with their chats.



When someone clicks on an item to further look at it this is a rough look at what it would look like. You can see the seller's name at the top and clicking on that will link to their profile. Right below that, you will see (a) picture(s) of the item. To the right is a list of details including the name of the item, the price, when it was listed on CSwap, the condition of the item, and a section for other notes that the seller has put on the item. Below is a quick way to start a message with the seller.

## **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**

## **Functional requirements**

REQ 1: The system shall allow the user to log in using an email and password

REQ 2: The system shall allow the user to log in through google

REQ 3: The system shall allow the user to log in through Facebook

REQ 4: The system shall allow the user to log out of their account

REQ 5: The system shall ask the user for their location via zip code

REQ 6: The system shall allow the user to change zip code

REQ 7: The system shall allow the user to change password

REQ 8: The system shall display the provided picture of the item for sale

REQ 9: The system shall display items from the corresponding categories

REQ 10: The system shall allow user to click on the item’s image to see more information about the item

REQ 11: The system shall allow the user to search by title to find the item

REQ 12: The system shall allow the user to select a category to view items from

REQ 13: The system shall allow the user to find items by adding search filters.

REQ 14: The system shall allow the user to change the area they will see items listed for sale

REQ 15: The system shall allow the user to add an item for sale

REQ 16: The system shall require a title for the item a seller wants to sell

REQ 17: The system shall require a picture of the item the seller is trying to sell

REQ 18: The system shall require a description of the item a seller wants to sell

# **Other Nonfunctional Requirements**

## **Performance Requirements**

The CSwap application will offer a fast and quick to use marketplace for college students to buy and sell goods in their local area. The application will also offer a quick to use marketplace which allows landlords in the area to list units for rent. The following Nonfunctional requirements are to describe the performance of the application

*The system shall be available for 24hrs a day, 7 days a week to all users, 99% of the time*

*The system shall not take more than one second to respond 99% of the time*

*The system shall not require longer than 5 minutes to make an account, 99% of the time*

*The system shall allow 10,000 messages to be sent between users at one time*

## **Security Requirements**

The CSwap application will be using Firebase to store and handle user data instead of using first party servers which drastically improves the security of the application. The following Nonfunctional requirements are regarding the applications security requirements.

*The system shall store user account information using Firebase*

*The system shall use Firebase policies for protecting user data*

*The system shall keep all user data on Firebase and away from non-required third parties*

## **Privacy Requirements**

The CSwap application will be collecting user data in terms of the user’s approximate location using their zip code, name of university, and radius of location to supply the user with items in their approximate location. The following Nonfunctional requirements are meant to maintain privacy regarding the user’s location.

*The system shall not share user location data with third parties.*

*The system shall collect user location data only to find items in the vicinity.*

*The system shall only use the users zip code to find items in vicinity*

*The system shall only use the user’s radius of location to find items in vicinity*

*The system shall handle user data in compliancy with all local privacy regulations*

## **Software Quality Attributes**

*The system shall be portable across devices such as laptops to mobile*

*The system shall allow all possible users to use the application at once*

*The system shall be relied on by users to work as expected whenever a request to the application is made*

## **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.>*

*Product should be self sustaining, only requiring security updates / bug fixes & maintenance. Feature updates are optional.*

# **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.>*