# SOFTWARE REQUIREMENTS SPECIFICATION

for

# Online Sales Portal

**Version 1.0**

**Animesh Jha (19CS10070)**
**Nisarg Updhayaya (19CS30031)**
**Pranav Rajput (19CS30036)**

**Indian Institute of Technology, Kharagpur**

**March 20, 2021**

# Revision History

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

# Contents

# 1. Introduction

## 1.1. Purpose

This Software Requirements Specification (SRS) document describes the software, non software and functional requirements of the Online Sales Portal (OSP). A user on the portal can buy and sell goods, managers can audit, ensure product quality and help in negotiations. The primary purpose of the OSP is to create a common platform for managers and a one-stop store for both buyers and sellers.

## 1.2. Document Convention

**bold** used for topics, no other explicit convention has been used otherwise.

## 1.3. Intended Audience and Reading Suggestions

This SRS is for developers, project managers, users and testers. Further the discussion will provide all internal, external, functional and non-functional information about the OSP.

## 1.4. Project Scope

The software aims to provide a convenient interface to the users to streamline purchases. It saves time for both the buyers and sellers as a one stop place to browse and review offers. To achieve this goal it has the following features.

- Maintains a database of Manager, Customer and Item profiles.
- Presents a login interface, where a user can login in as a manager or a customer. First time users can apply for a profile
- Each party will be able to access their profiles and use the functionalities available to them
- Managers would be able to categorize, review and reject items
- Customers who are Sellers can upload items along with their relevant details
- Customers who are Buyers can search for specific items and put in a request to buy an item

- Buyer, Seller and Managers can receive contact information to engage in negotiations items
- OSP would provide a buyer and seller their payment details, after a payment is completed they can log whether they payed or received said payment.
- Managers can perform an audit of purchases made on the OSP.

## 1.5. References

- Online-Sales Structural Model
- Online-Sales Behavioral Model

# 2. Overall Description

## 2.1. Product Perspective

This software will be built as a streamlined one stop online store. OSP would create a common platform for buyers and sellers where they can search, compare and sort items. Anyone can use OSP to become a buyer or a seller. It would also have managers for monitoring and auditing items, and to help in seller buyer negotiations.

## 2.2. Product Functions

The OSP has the following major functions

- The portal is moderated by managers and used by customers who can buy and sell goods
- Items are categorized into different types, managers can add/remove categories of items in the portal
- Each item has its photo, age, manufacturer, city, price and other data available. Item data is provided by the seller
- Potential buyers can search for items, negotiate on the price and buy the item via an online transaction.
- Managers can manage buyers and sellers, help in price negotiations between a buyer and seller, and can perform audits of sold items, review and reject items of poor quality

## 2.3. User Classes and Characteristics

The OSP has 3 types of users.

- **Manager**: A manager can do the following:
  - Manage buyers/sellers
  - Manage categories; change categories of items if needed
  - Review items and rejects items of poor quality
  - Help negotiations - talk to (over email) buyer and seller both to help bridge small gaps or to assist with delivery logistics

  - Perform audit of matched buy-sell of items

- **Customer**: There are 2 types of customers
  - **Buyer**: A buyer can do the following:
    * Search for an item and find the list of the sellers who have uploaded that particular item for sale
    * Raise a request to buy an item. Offers a price. Negotiate on the selling price. If the seller agrees with a certain price then only the buyer can buy it
    * Cannot buy a heavy item like a refrigerator if it's seller location (city) is not the same
  - **Seller**: A seller can do the following:
    * Upload an item along with its information (Category of the item, Photo of the item, price, The age of the item (if not new), Name of the manufacturing company, and City)
    * Negotiate the selling prices with a buyer
    * Can approve and reject the offers received by different buyers on an item
    * Cannot sell a heavy item like a refrigerator if it's buyer location (city) is not the same

## 2.4. Operating Environment

The software can be used by any user with access to a device with internet. OSP would have a server where it will perform all information retrieval and a database which would store the manager, customer, item and transaction details. The user end would be a GUI which would fetch information from the back-end and render it on the user's screen.

- Database: MongoDB
- Backend: Would be hosted on Heroku, the codebase would use Flask, a robust Python framework to create a backend for the web application.The codebase would reside on GitHub, the Heroku deployment would use this as its source.
- Frontend: Vue.js, a Progressive JavaScript Framework along with HTML5, CSS and Bootstrap framework.

## 2.5. Design and Implementation Constraints

The OSP uses free tier MongoDB, so it is limited by 512 MB database space. At this stage we aren't providing a mechanism to actual conduct online transaction because of security concerns, however the portal is being designed in a way to allow easy integration of this service via UPI. OSP is further limited by the free tier of hosting platform Heroku.

## 2.6. User Documentation

All user documentations, including basic tutorials would be made available via the README of the repository. The same information would also be available on the deployed application

## 2.7. Assumptions and Dependencies

- **Assumptions**:
  - All data about items, managers and customers is provided by the users.
  - Buyer Seller negotiations are not conducted on the platform but via the Email ID or the phone number given by the users.
  - Online transactions are not conducted on the platform, it is the sellers responsibility to update the transaction status, the buyer can give a proof of payment to the seller who can then update the payment status.
- **Dependencies**:
  - Flask : Framework for backend
  - Vue.js : Front end framework
  - MongoDB : Database host
  - Bootstrap and CSS : Front end
  - Heroku : Web deployment host
  - HTML5 : Front end
  - GitHub : Hosts the codebase

# 3. External Interface Requirements

## 3.1. User Interfaces

There will exist a Graphical User Interface available via the web deployment. The UI deployed would be made as intuitive as possible, all design would be based on a base template to ensure uniformity. The GUI would be made as sleek as possible to reduce loading times.

## 3.2. Hardware Interfaces

OSP would be accessible via any device with an internet connection, keeping this in mind all design would be browser optimised to improve user experience. The free version of heroku has RAM limit of 512MB.

## 3.3. Software Interfaces

There is any software requirement for a user. The user should just have access to a modern internet browser.

## 3.4. Communication Interfaces

- **User - Front End**: User can access the front-end UI through a modern web-browser.
- **Front End - Back End**: This communication happens via Vue.js and Flask
- **Back End - Database**: The backend communicates with the database using the python pyMongo library
- **User - User**: Users can communicate via the Email IDs and the phone numbers provided on the OSP

# 4. System Features

*Online Sales Portal* is an online marketplace. So it provides features which streamlines the process of buying and selling goods.

## 4.1. User Profiles

1. Manager : They can manage the portal. They can change the categories of items, manage buyers and sellers, review the items available for sale on the portal, perform sales audit, and help with negotiations and logistics.
2. Customer : They can use the portal to buy and sell goods. They can act as
   a) Buyer
   b) Seller

## 4.2. Categorization of items

The portal provides functionality to categorise the items on sale. The manager can add/remove categories from the portal. Some examples of categories are :

1. Electronics
2. Real Estate
3. Books
4. Automobile
5. Education & Learning
6. Home & Lifestyle
7. ...

## 4.3. Upload item

This section of the portal allows the customer to upload an item for sale on the portal by specifying its category, price, age, manufacturing company, city from where it will be shipped. The Customer can also upload an image of the product and provide any other specific information regarding it.

**Response Sequence** :

- The customer proceeds to add a new item to the database by entering the various data-fields related to it such as the item name, price, age, manufacturing company and city of shipping as input to the front-end program.

**Functionalities** :

- Helps in expanding the existing item database of the online-sales portal.

## 4.4. Search for items

This section of the portal allows the user to search for the presence of any particular item of their choosing in the existing database of items. The search for items can be carried out by either of the 2 parameters : item name and the category of the item.

**Response Sequence** :

- The User selects whether the search is to be carried out by name or category.
- User provides the necessary input, that is the item or category name (based on search type)
- The front-end program makes a fetch request to the back-end for an item search by name/category.
- The back-end program returns a list of items to the front-end program with the closest/exact matches to the search request.
- The front-end program then displays the results (the item list returned by the back-end program) that are an exact/closest match in the database to the search request.

**Functionality** :

- Helps customers, buyers and sellers to look for items easily whose details are then used to carry out further operations.

## 4.5. Request to buy item

This part of the portal allows the customer to put in a request to buy an item of their choice and offering a price for the item in consideration.

**Response Sequence** :

- The customer selects an item of their interest from the existing database of items in the portal.

- The customer then puts a purchase request for the item along with a price they are willing to pay.
- (optional: user-dependent) The customer can carry out negotiations with the seller of the product to lower the price of the item they want to buy.
- The purchase request put in by the customer gets accepted/rejected by the seller on his discretion.
- If the purchase request is accepted, the number item of items available is decreased by one.

**Functionality** :

- Allows the customers to make purchases for the items of their choosing for an acceptable price. Helps in making purchases go smoothly.

## 4.6. Negotiation

This section of the portal provides the customer to interact with the seller of an item of their interest and provides them with the means to conduct negotiations with the seller to lower the price of the item of interest.

**Response Sequence** :

- The customer retrieves the seller (of the item they want to buy) contact information that is present on the portal.
- The customer then proceeds to negotiate the item price with the seller.
- Additionally, the manager can also be asked to intervene in the negotiation process.

**Functionality** :

- Provides the customer the option to negotiate with the seller in order to lower the price of the item of their interest. Helps in providing customer-seller interaction.

## 4.7. Payment and check delivery constraints

This section of the portal allows the buyer to get the payment information of the seller directly from the portal. For products listed as heavy the buyer will only be able to place a buy request if the seller and buyer city is same.

**Response Sequence** :

- Upon accepting a sale request the seller provides the payment information to the buyer.

- Upon receiving the payment the seller updates the payment status to complete for the sale request.

**Functionalities** :

- Provides an easy method for the retrieval of seller contact information.

## 4.8. Review and reject items

This section of the portal allows the manager to view all the items listed for sale on the portal and remove them if they feel they are not up to the standard.

**Response Sequence** :

- The Manager can view all the items put on sale by the customers(sellers).
- The item selling requests are approved/rejected based on the manager's discretion.

**Functionality** :

- Faulty selling requests or requests from sources that seem unreliable can be removed by the manager.

## 4.9. Audit

This section of the portal allows the manager to perform sales audits. The portal provides the functionality to view all successfully matched sale items and the price at which the sale took place.

**Response Sequence** :

- The Manager can view all the successfully completed sales requests on the website.

**Functionality** :

- The managers can get an idea of the successful sales, the categories in which those were made, the sellers and the buyers involved, etc.

# 5. Other Nonfunctional Requirements

## 5.1. Performance Requirements

Should throttle requests to make sure that API limit is not exceeded

## 5.2. Safety Requirements

- There is no risk of any safety threat that can be issued by the system.
- OSP is a web application so there can always be chances of internet based threats, to combat this we will have the following two measures
  - Password protected login interface
  - All media uploads would be scanned for viruses before saving to the database

## 5.3. Security Requirements

- The portal system's back-end servers will never display a customer's password.
- The portal system's back-end servers will only be accessible to authenticated administrators.
- The portal system's back-end databases will be encrypted (third-party dependencies)
- NOTE : Data privacy(encryption) is not being handled by us as we have used mongoDB, a third party database storage service which may have some security issues.
- NOTE : All third-party APIs have been assumed to be secure and bug-free in the ideal case but this may not hold true everywhere and thus may cause security and technical issues.

## 5.4. Software Quality Attributes

- Object Oriented Design principles ensure that the application can easily be modified for future extensions.
- The application stores all the data in a cloud database and is thus, reliable. Hence users don't need to worry about data loss in case the app shuts down.

- The front-end is made as lightweight as possible to ensure faster loading times and a smoother experience.
- The codebase would be open sourced and therefore needs to be highly readable and documented to allow developers to contribute easily

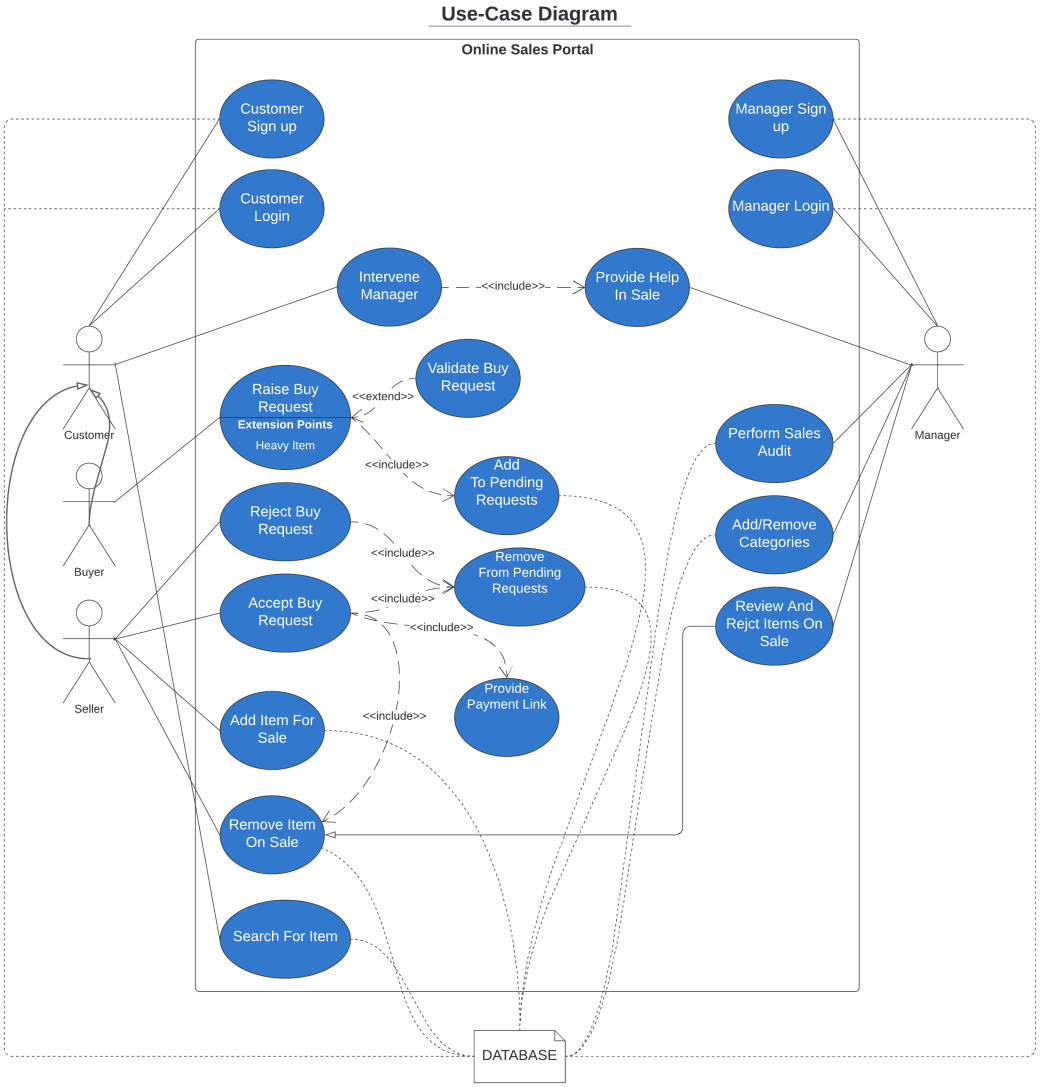
# 6. Other Requirements

GUI should be effective, interactive and intuitive to increase user experience. The total application should be made as sleek as possible to decrease loading time and to improve user experience. All designed should be made as mobile friendly as possible as bulk of the internet traffic originates from mobile phones.

# A. Glossary

- SRS : Software Requirements Specification, A document that completely describes all of the functions of a proposed system (i.e. OSP) and the constraints under which it must operate. For example, this document.
- OSP : Online Sales Portal; the name of the project in consideration.
- Negotiations : The functionality to get the contact details of a seller of a product in order to negotiate the price.
- Manager : One of the final users of the software, who uses it for managing the portal which includes managing the products, the various users and performs audits. Has the highest level of access.
- Customers : One of the final users of the software, who can use it for buying or selling various products.
- Buyer : One of the final users of the software, who can use it for buying various products
- Seller : One of the final users of the software, who can use it for selling various products
- Heavy items : An item with weight greater than a pre-set maximum limit which imposes restrictions on the delivery and transportation of the item.
- GUI : The graphical user interface; It is a form of user interface that allows users to interact with electronic devices through graphical icons and audio indicators as the primary notation.

# B. Use Case Diagram



# C. Class Diagram