SOFTWARE REQUIREMENTS SPECIFICATION

for

Medicine Inventory System

Version 1.0

Prepared by

Group 8

Aditya Jain
Aditya Semwal
Akash Kumar Panda

Course : CS6103D Software Systems Lab
Submission date: 11.12.2020

Contents

1. Introduction

Everyone in this world, somewhere in the world and in time one has to face some kind of disease or injuries. The disease to cure we need some doctor to concern. As we go to the doctor he analyzes the problem with us and tell us the reason of problem and prescribe us some sort of medicines. These medicines are available in medical stores. one needs a prescription to the store which the staff takes care and use that to see which medicines they need according to the name given to the medicine. now, this becomes a bit tedious to locate the particular medicine. Now here come up with such a system which capable of recording that from where the medicines need to take out according to the name, brand, price in the racks, which are marked with the no . of them. Here the system helps the staff to manage the medicines in the racks and also useful in reinstalling the stock in the racks and easy to find.

1.1. Purpose

The main objective of the this Project on Medicine Inventory System is to manage the details of Sells, Medicines, Stocks, Company,Inventory. It manages all the information about Sells, Medical Shop, Inventory.The project is totally built at administrative end and thus only the staff registered is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Sells, Medicines, Medical Shop, Stocks. It tracks all the details about the Stocks, Company, Inventory.

1.2. Intended Audience and Reading Suggestions

This particular document is merely meant to be read by anyone but the reader concerned should be related to owning a medical store or running a pharmaceutical agency. Moreover it would also benefit the marketing staff as they do not have to be hindered with matters related to availability of medicines. The SRS contains brief details about the product,the functionalities it provides and UML diagrams that would give a comprehensive overview.

1.3. Project Scope

As this is generic software it can be used by a wide variety of outlets (Retailers and Wholesalers) to automate the process of manually maintaining the records related to the subject of maintaining the stock and cash flows.This project is basically updating the

manual chemist inventory System To Automated inventory system, So that organization can manage their record in efficient and organized form.
This software helps you to track all the products of medical shop moreover it's a medicine inventory system software. it is a flexible and adaptive software suited to medical shops or stores or pharmacies of any size.

1.4. Definitions, Acronyms and Abbreviations

Some of the abbreviations used in our SRS are as follows :

1. inv–Inventory
2. med–Medicine
3. mesr–Measures
4. UML–Unified Modelling Language

1.5. Document Conventions

This Document follows the Traditional English writing pattern with each topic being a Header and its sub topics being sections within the Header.The header is called as 'Chapter' and it starts with numeric value'1',the subtopics called as 'Sections' are thus numbered following the index extended value such as '1.1'.

1.6. References and Acknowledgments

The references that provided a helping hand in building such an application are as follows:

- http://guides.rubyonrails.org
- https://www.javatpoint.com/ruby-on-rails-tutorial
- https://www.youtube.com/

2. Overall Description

2.1. Product Overview

This product "Medicine Inventory System" is a self contained product which would help us to create the registration of the staff members that work in that particular pharmaceutical agency. Added to it ,it would also enable us to have members listed for each survey and maintain list of medicine stocks available with us that gets updated time to time based on the usage. Only the authenticated staffs would be able to modify the contents and thus that would enhancing be helpful to who ever gets to use it.

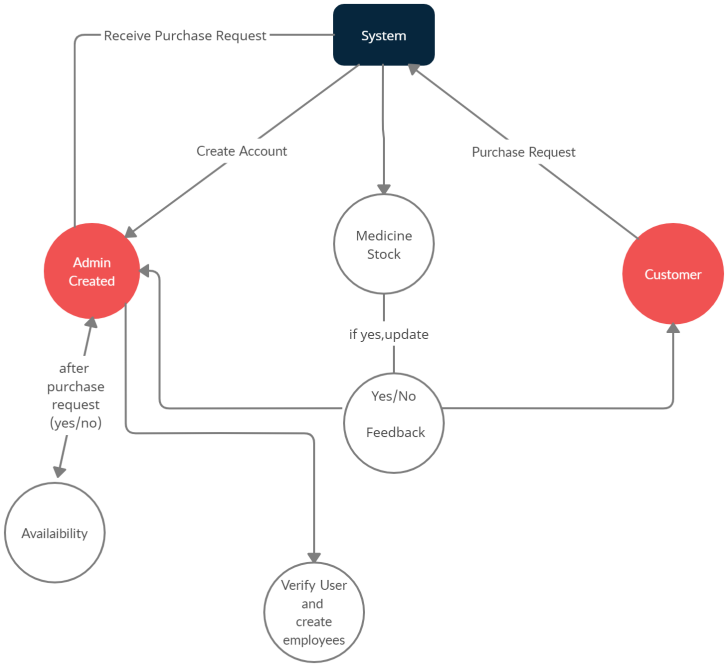

Figure 2.1.: overview of product

2.2. Product Functionality

Various functions are to be implemented here and among them a brief overview of functions can be listed as :

- Employee Credentials
- Managing Employees
- Stock Details
- Secure Purchase
- Feedback
- Update Stock Details
- View Details
- Account Handling

2.3. Design and Implementation Constraints

The particular product is being developed under Ruby on Rails Working on it the main problem that we encounter is the run-time Speed and moreover Boot Speed is one other major concern that we face here For our product although its not using any specific tools other than Rails ,still we need to use the database "SQLite3".In terms of security perspective it would only allow the authenticated staffs to do necessary modifications since they have to register by mail with their own log in credentials. Basic hardware requirements can be cited as :

- 256 MB RAM
- 30 GB Hard Disk space
- Active Internet Connection

2.4. Operating Environment

The website will be operate in any Operating Environment - Mac, Windows, Linux etc.

3. System Analysis

3.1. System Design:

3.1.1. Users:

The major functionality of this product is divided into three categories.

1. Administrative User Functions.
2. Staff Member Functions.
3. Normal User functions.

3.1.2. Administrative User Functions

Administrators can perform the following tasks in this system:

1. Create new users
2. Change the password
3. Add/Update the details of Staff
4. Add the information about the Medicine Stock
5. Can view the information about the Inwards
6. Can view the information about the Orders
7. Can view the information about the Returns

3.1.3. Staff Member Function:

Staff member can perform the following tasks in this system:

1. Can approve item purchase
2. Can approve item returns
3. Can update details about the items and warehouse database
4. Can add new product

3.1.4. User Function:

User can perform the following tasks in this system:

1. Create purchase list
2. Create return list

3.2. UML Diagrams

3.2.1. UML (Unified Modelling Language)

The unified modeling language is a standard language for specifying, Visualizing, Constructing and Documenting the software system and its components. It is a graphical language which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand ,design , configure, maintain and control information about the systems.

3.2.2. Visualising

Through UML we see or visualize an existing system and ultimately we visualize how the system is going to be after implementation. Unless we think we cannot implement. UML helps to visualize how the components of the system communicate and interact with each other.

3.2.3. Specifying

Specifying means building models that are precise, unambiguous and complete UML addresses the specification of all the important Analysis Design, Implementation decisions that must be made in developing and deploying a software system. Constructing :UML's models can be directly connected to a variety of programming language through mapping a model from UML to a programming language like Java or C++ or Ruby on Rails.
Forward Engineering and Reverse Engineering is possible through UML.

3.2.4. Documenting

The deliverable of a project apart from coding are some artifacts which are critical in controlling, measuring and communicating about a system during its development viz. Requirements, Architecture, Design, Source code, Project plans, Tests, Prototypes, Releases etc.

3.2.5. Diagrams in UML

Diagrams are graphical presentation of set of elements . Diagrams projects a system, or visualize a system from different angles and perspectives.
The UML has certain diagrams which can help in better understanding of the project. Some of the diagrams we have used are:

1. Class diagrams.
2. E-R diagrams.
3. Use case diagram.
4. Activity diagrams.

3.3. Modules

The System after careful analysis has been identified to present with the following modules:

3.3.1. Employee Information Module:

This module maintains all the information which belongs to the employees who are working for the company. It allows the administrator to add an employee record to the database very easily and it allows to view the list of employees in tabular format out of which he can edit a particular employee.It makes all the above can be done very flexibly.

3.3.2. Inwards Module:

This module maintains all the information related to manage inwards done in the medicine stock. All the inwards are recorded to database and can be viewed as detailed list about the medicines that are presently available. Moreover the stocks can be piled up with the requirements that are needed and the same gets updated in the database as well.

3.3.3. Purchase Module:

This module deals with major and crucial part which includes purchase of medicine items after placing the purchase request by customers and thus further checking if the particulars are available in the stock and thus updating the stock.

3.3.4. Returns Module:

This is one of the other important module that deals after the feedback is received from the user.If the user is not satisfied with the particular product then he can place a return request which would then be verified by the staff and thus be granted the access to return that item.

3.3.5. Administrator Module:

This module is used to manage the details of users of the application. Users are divided into two categories.

1. Admin
2. Staff Members
3. Normal Customers

It allows administrator to add a new employee, view the list of employees and delete a employee from the database.

4. Specific Requirements

4.1. External Interface Requirements

4.1.1. User Interfaces

Describe the logical characteristics of each interface between the software product and the users.
TO DO: Provide the GUI of the app and state how users are expected to interact with it.

4.1.2. 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. You are not required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.
TO DO: Since we do not intend to use sensors or other typical hardware, we may not need this. But if the mess management project people would like to have an interface with the finger print scanners that needs to be mentioned here.

4.1.3. Software Interfaces

Describe the connections between this product and other specific software components. If yours is a standalone app there is nothing to include here.

4.2. Functional Requirements

There are various views in the system, admin, staff and general user. The admin reserves all rights to sites and hence can modify details, add staff and prioritise other things. The various functionalities of sites, views and database management is as follows:

1. The System holds all the details of the all the employees who are working in the organization.

2. It allows admin to manage all types of users, hold their details, authenticate these users at the time of login and accordingly provide different options.

3. It allows admin to create staff credentials and users to create login credentials.

4. It sends created credentials automatically to email of that staff member or user.
5. It allows staff members to login using their credentials and make changes in warehouse holdings in case of inward entries, purchase or return of items.
6. The system holds the details of all product in and out of stock.
7. The system provide security from unauthorised purchase by making buyers to sign up for purchase, users can see all items that can be purchased with selling price and quantity.
8. It allow user to generate purchase and return request of item with complaint regarding that product.
9. It allow staff to update selling information and this information is automatically reflected in selling page of that stock.
10. It allow staff members to increment stock number whenever inward entries are there.
11. Whenever a purchase entry is requested then accordingly the stock number will be automatically decremented if approved by staff.
12. Whenever a return entry is requested then accordingly the stock number will be automatically incremented if approved by the staff.
13. The system allows staff to log into the system and see all purchase and return entries by various users.
14. It allow users and staff to change their password for future security.
15. It allows the admin to view the list of registered users and staff, purchase details, return details, in-stock items details.
16. It allows any user to logout when he wants to come out from the system.

4.3. Use Case Model

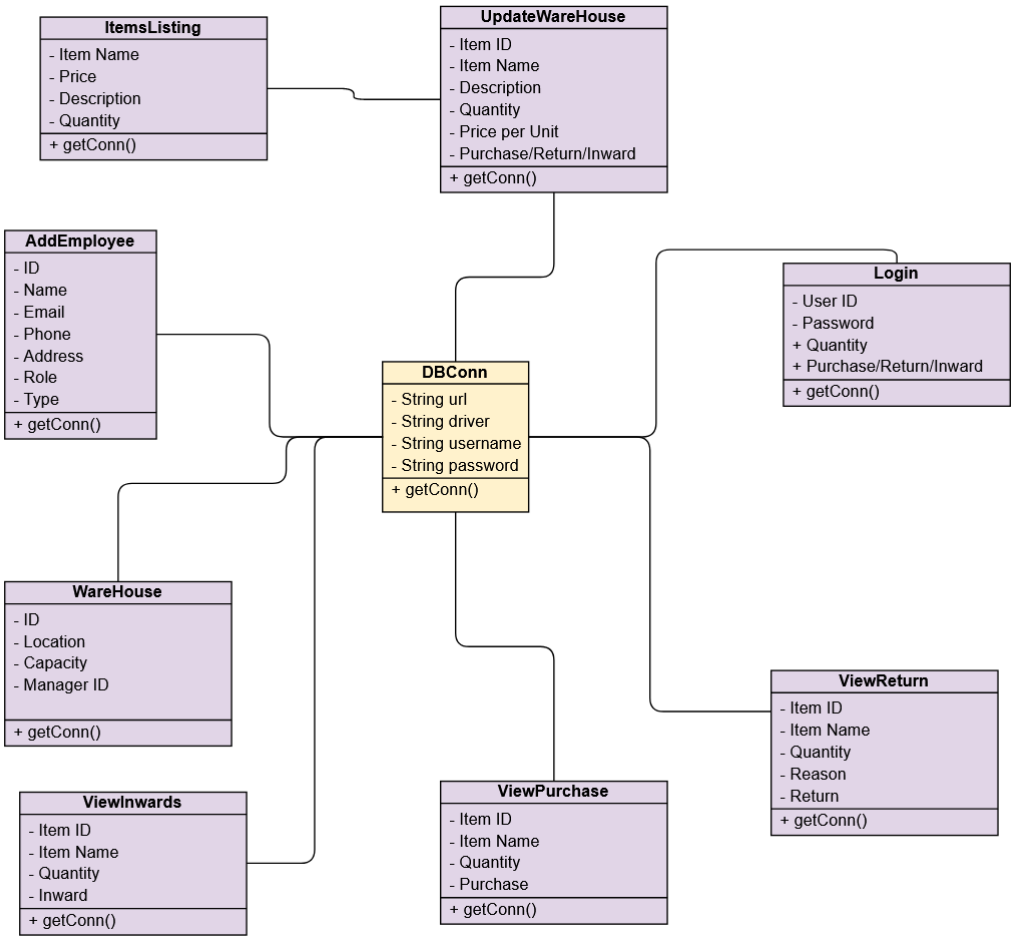

Figure 4.1.: Class Diagram

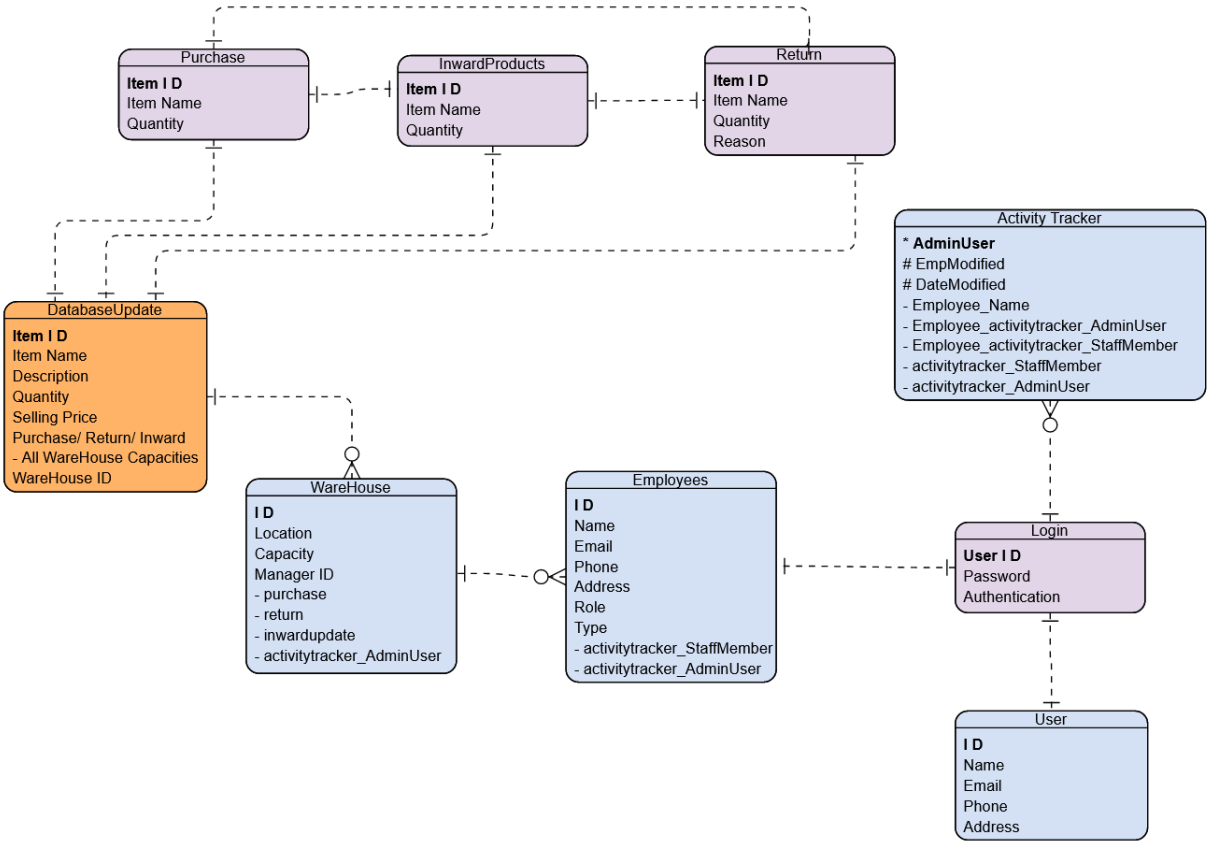

Figure 4.2.: ER Diagram

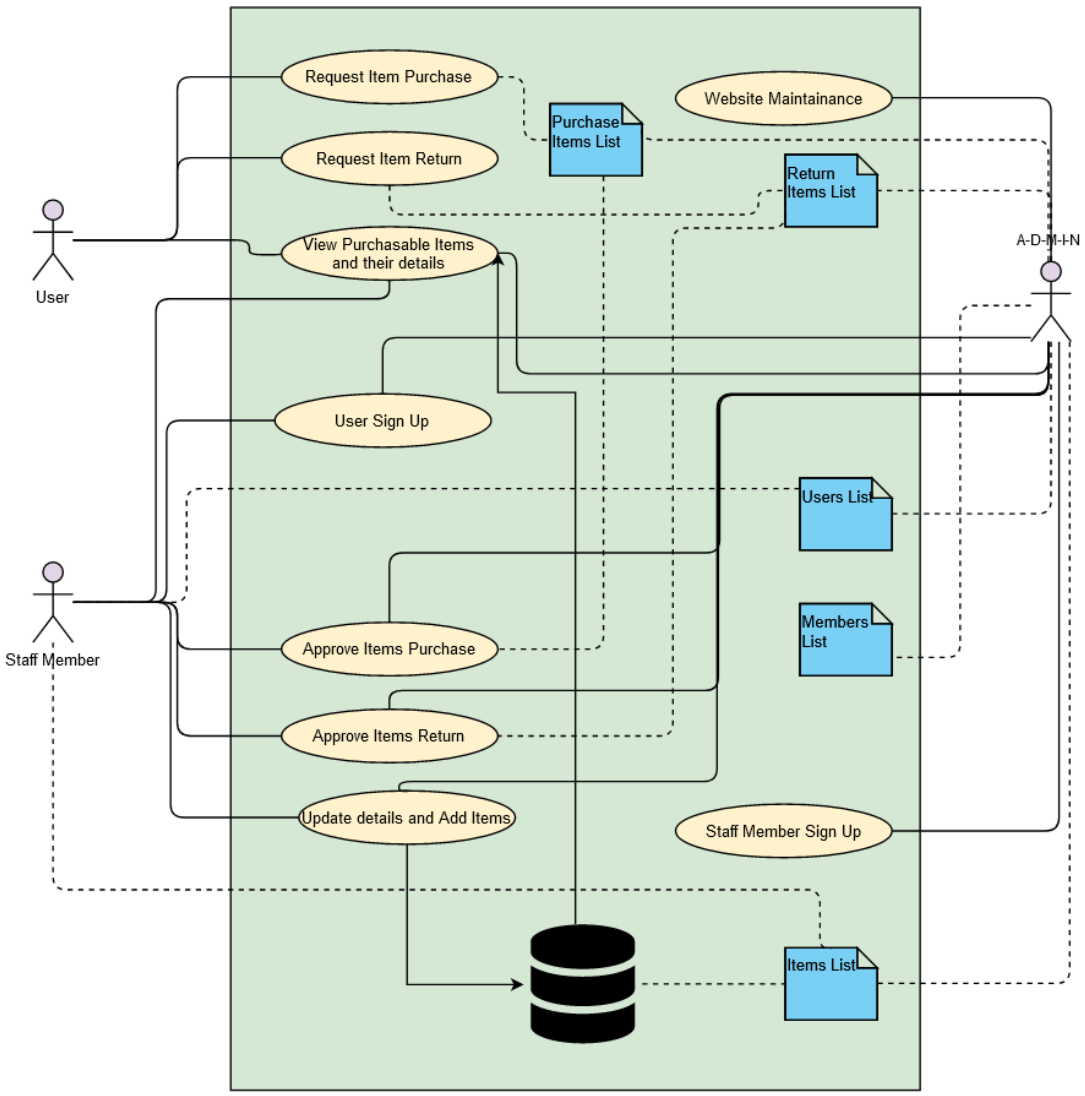

Figure 4.3.: Use Case Diagram

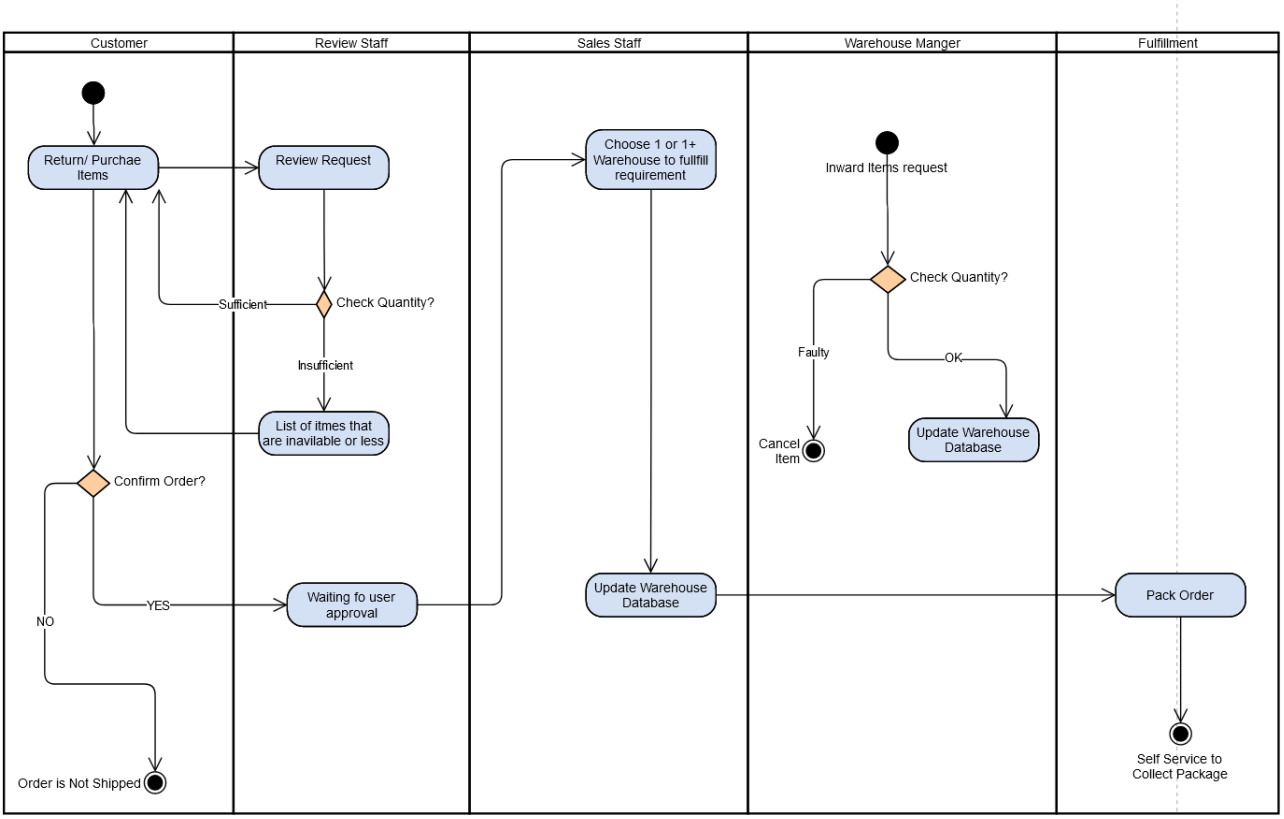

Figure 4.4.: Activity Flow Diagram

5. Other Nonfunctional Requirements

The non-functional requirements consist of:

1. Analysis, Design Data requirements (Use-case diagrams, textual analysis, sequence diagrams, data dictionary etc.)
2. Constraints.
3. Software Quality Attributes.

5.1. Analysis, Design Data requirements

The Analysis Design of the system yield Use Case diagrams, textual analysis, Sequence Diagrams, Class diagrams Data Dictionary. Data dictionary consists of process statements showing how data is flowing from starting point to end point.
Al this data provide a overview about how to start building project in case of any fault who is responsible, various types of views provided etc. How much data is taken from user and how much to analyse for further use. Schema of database management in par to future requirements.

5.2. Constraints

These are the requirements that are not directly related to the functionality of the system but these should be considered as mandatory when the system is developed.

1. The system should be available over the intranet so that the Users like the admin and staff members can use the system from their respective locations which could be anywhere in the organization.
2. For gaining entry into the system the admin should register staff member info and the user should be able use login passwords for gaining access to the system by credentials received through auto-generated mail.
3. The system should be easy to understand and organized in a structured way. The users should also receive feedback about any errors that occur.
4. There should be no limitation about the hardware platform that is to be used to run the system.
5. Data integrity should be maintained if an error occurs or the whole system comes down.

6. Any update to the database items should reflect automatically in the product display and their respective description pages.

5.3. Software Quality Attributes

Recommendations to further enhance the usability of the system.

1. The system should display a user friendly menu for users to choose from.
2. The system should provide multiple warehouse management and provide ID of warehouse and item to be choose from the popup list in the forms.
3. Services of the system should be available 24 hours a day.
4. The system should be designed in such a way that it is easy to enhance it with more functionality in near and far future. It should be scalable easily maintainable.

A. Activity Log

A.1. Aditya Jain

Content: Chapter 1 and Section 3.1 and 3.2
Diagram: Class Diagram

A.2. Aditya Semwal

Content: Chapter 5 and Section 4.2
Diagram: Use Case Diagram and Activity Flow Diagram

A.3. Akash Kumar Panda

Content: Chapter 2, Section 3.3 and 4.1
Diagram: ER Diagram and Overview of Product