# CONSIGNMENT MANAGEMENT IN HOSPITAL INVENTORY

A Project as a Course requirement for

**Integrated Master of Computer Applications**

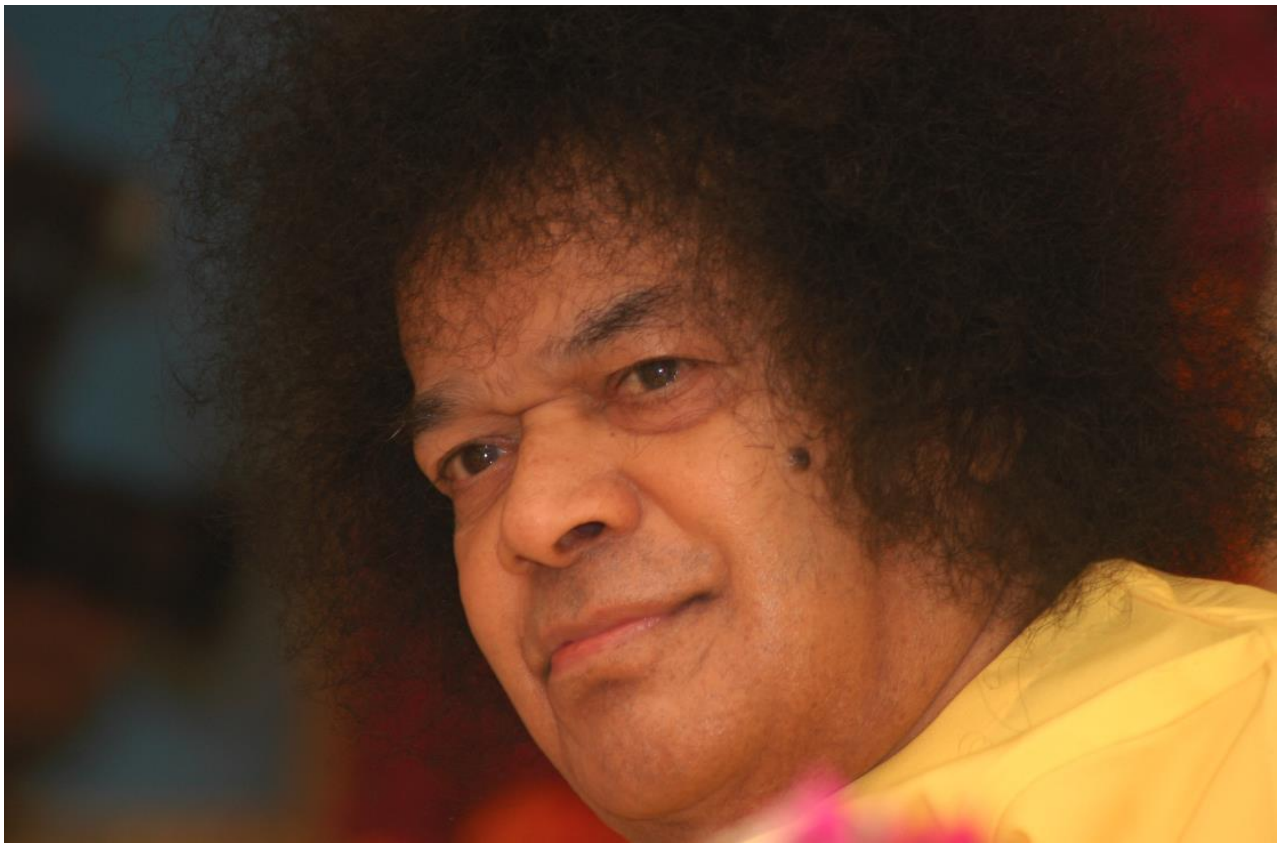
**Bangarugiri Sateesh**
134401


SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING
(Deemed to be University)

Department of Mathematics and Computer Science
Muddenahalli Campus
March - 2018

# Dedicated To

# Bhagwan Sri Sathya Sai Baba



SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING
(Deemed to be University)

# *CERTIFICATE*

This is to certify that this Project titled **Consignment Management in Hospital Inventory** submitted by Bangarugiri Sateesh, 134401, Department of Mathematics and Computer Science, Muddenahalli Campus is a bonafide record of the original work done under my supervision as a Course requirement for the Degree of Integrated Master of Computer Applications.

……..
Sri P Sunil Kumar
Project Supervisor

Countersigned by

Place: Muddenahalli

Dr. Pallav Kumar Baruah
Head of the Department

Date: 15th March 2018

# DECLARATION

The Project titled **Consignment Management in Hospital Inventory** was carried out by me under the supervision of Sri P Sunil Kumar, Department of Mathematics and Computer Science, Muddenahalli Campus as a Course requirement for the Degree of Integrated Master of Computer Applications and has not formed the basis for the award of any degree, diploma or any other such title by this or any other University.

……..

Place: Muddenahalli

Bangarugiri Sateesh
134401
Integrated Master of Computer Applications
Muddenahalli Campus

Date: 15th March 2018

# Acknowledgement

First and Foremost, I shall express my heartfelt gratitude to my guru **Bhagawan Sri Sathya Sai Baba** who is the founder chancellor of our institute Sri Sathya Sai Institute of Higher Learning.

I fail in my duty if I do not extend my thanks to the Institute and the Department for providing me with state-of-the art facilities without which I would not have done this project. I take this opportunity to thank the consecutive Heads of the Department Dr. Pallav Kumar Baruah and the Deputy Director Incharge of our college Sri K Sayee Manohar for kindly providing all necessary facilities and permission to use them.

I am grateful to my guides Sri P Sunil Kumar and Sri Subramaniyan M. They have been extremely supportive, encouraging during the progress of the project.

I express my deepest sense of gratitude to all who have directly or indirectly helped me in doing the project.

# Special Thanks

## To

My Mentor

## Subramaniyan M

Senior Manager,
Hospital Health Information System & Telemedicine,
Sri Sathya Sai Institute of Higher Medical Sciences,
Whitefield, Bangalore.

# ABSTRACT

Consignment Management involves managing the pharmacy items by recording details of each item in the hospital. A study of workflow was done in the Sri Sathya Sai Institute of Higher Medical Sciences, Whitefield.

The process begins when items are received from the vendors and the provisional receipt notes are generated for the same. These receipts are to be authorized by Inventory Assistant before issuing the items to the departments in the hospital. After authorization process, the required items have to be issued to the requested departments. Items consumption details have to be recorded by patient wise by the department nurse-incharge. Verification of consumptions with respect to the issues of items for each department has to be checked. The requested items can be issued based on the current stock of the requested department.

Consignment Inventory is inventory that is in the possession of the customer, but is still owned by the supplier. In other words, the supplier places some of his inventory in his customer's possession (in their store or warehouse) and allows them to sell or consume directly from his stock. The customer purchases the inventory only after he has resold or consumed it.

The key benefit to the customer should be obvious; he does not have to tie up his capital in inventory. This does not mean that there are no inventory carrying costs for the customer; he does still incur costs related to storing and managing the inventory. So what's in it for the supplier? This is where the benefits may not be so obvious—or may not even exist.

Consignment Management system that will integrate with the existing HMIS and INVENTORY MANAGEMENT SYSTEM to facilitate the all above mentioned services and it replaces the existing visual basic based software.

# Contents

**ABSTRACT** ..... vii

**1 INTRODUCTION** ..... 1

**1.1 What is Consignment Inventory?** ..... 1

1.2 How is consignment inventory used? ..... 1

1.3 What are the compliance implications? ..... 1

**1.4 Where Consignment Works Best.** ..... 2

**2 REQUIREMENT ANALYSIS** ..... 3

**2.1 Current Scenario and Needs** ..... 3

**2.2 Process Modeling and Flowchart** ..... 4

**2.3 Use Case Modeling Concepts** ..... 5

2.3.1 What is a use case? ..... 5

2.3.2 What is the use case model? ..... 5

**2.4 Use Case Modeling of the Project** ..... 6

2.4.1 Provisional Consignment Receipt Note (CRN) ..... 6

2.4.2 Authorized Consignment Receipt Note (CRN) ..... 7

2.4.3 Consignment Issues to Sub store ..... 8

2.4.4 Consignment Consumption ..... 9

2.4.5 Consignment Consolidation ..... 9

2.4.6 Verification of Consignment Consumption ..... 10

2.4.7 Acceptance of Consignment Consumption ..... 11

**2.5 Sequence Diagram** ..... 12

**2.6 Literature Review** ..... 19

**2.6.1 Spring Framework** ..... 19

**2.6.2 Hibernate** ..... 19

**3 DEVELOPMENT PLATFORM** ..... 20

**4 SYSTEM DESIGN** ..... 21

**4.1 Architecture** ..... 21

**4. 2 Single Page Application Execution** ..... 22

**5 DATABASE DESIGN** ..... 25

**6 DETAIL DESIGN** ..... 26

**6.1 UML Diagrams and Details** ..... 26

**6.2 Code Snippets** ..... 28

**6.3 Database Tables** ..... 30

**6.3.1 Consignment Receipt Header (for generating crhid for items)** ..... 30

**6.3.2 Consignment Transfer (for saving the item transfers)** ..... 30

**6.3.3 Consignment Consumption (for saving the consumption of the items)** ..... 31

**6.3.4 Consignment Receipt Line** ..... 32
**6.3.5 Consignment Stock** ..... 32
**7 TESTING AND IMPLEMENTATION** ..... 33
**8 SCREENSHOTS OF THE APPLICATION** ..... 34
**8.1 Login Page** ..... 34
**8.2 Home Page** ..... 34
**8.3 Provisional Consignment Receipt Note** ..... 35
**8.4 Authorization of Consignment Receipt Note** ..... 35
**8.5 Consignment Issues to Sub Store** ..... 36
**8.6 Consignment Consumption** ..... 36
**8.7 Consignment Consolidation** ..... 37
**8.8 Verification of Consignment Consumptions** ..... 37
**8.9 Acceptance of Consignment Consumptions** ..... 38
**8.11 Future Scope** ..... 38
**9 REFERENCES** ..... 39

# List of Figures

Figure 1 Flowchart ..... 4
Figure 2 Sequence diagram for PCRN ..... 12
Figure 3 Sequence diagram for ACRN ..... 13
Figure 4 Sequence diagram for Issues ..... 14
Figure 5 Sequence diagram for consumptions ..... 15
Figure 6 Sequence diagram for Consolidations ..... 16
Figure 7 Sequence diagram for Verification ..... 17
Figure 8 Sequence diagram for Acceptance ..... 18
Figure 9 Hibernate ..... 20
Figure 10 Architecture ..... 21
Figure 11 SPA Execution ..... 22
Figure 12 Servlets Architecture ..... 23
Figure 13 ER Diagram ..... 25
Figure 14 UML for PCRN ..... 26
Figure 15 UML for ACRN ..... 26
Figure 16 UML for Issues ..... 27
Figure 17 UML for Verification ..... 27
Figure 18 UML for Acceptance ..... 27

# List of Tables

Table 1 Consignment Receipt Header ..... 30
Table 2 Consignment Transfer ..... 31
Table 3 Consignment Consumption ..... 32
Table 4 Consignment Receipt Line ..... 32
Table 5 Consignment Stock ..... 33

# 1 INTRODUCTION

## 1.1 What is Consignment Inventory?

**Consignment inventory**, a practice seen across many types of supply chains, is commonly employed in hospital settings and sometimes by pharmacies. Consignment inventory allows a business to take ownership of product at a flexible time, independent of when it takes possession. Sometimes taking ownership is tied to a procedure or actual dispensation, but it can also be triggered by inventory management. It generally gives the dispensing organization the flexibility to take ownership based on when they actually need the medication. Until that point - and sometimes even briefly afterwards - a wholesale distributor still owns the product.

## 1.2 How is consignment inventory used?

Consignment inventory is often part of the managed services a wholesale distributor provides. They will ship product to a pharmacy - hospital or independent - and manage inventory, monitoring product levels and restocking as necessary. They retain ownership of the drugs until they are used. Offering this level of service can be a competitive advantage for the wholesaler.

Another common use for consignment inventory is in hospital settings, where drugs are dispensed during procedures and the exact amount of a medication that will be needed cannot be predicted in advance. Wholesale distributor representatives will sometimes attend these procedures to determine how much product was used and finalize the change of ownership.

## 1.3 What are the compliance implications?

If you work with consignment inventory, here are several things to consider as you develop compliance procedures:

- Consignment inventory may be managed by personnel who do not normally participate in DSCSA processes. For instance, an operating room nurse may pull compliance inventory off the shelf for a procedure, triggering the change of ownership. How will you train these staff to recognize that a change of ownership has occurred and follow up?
- Once you have oriented all personnel to the DSCSA implications, will you give them access to your compliance system, or train them to communicate the

change of ownership to your staff that are regularly responsible for DSCSA compliance issues?

- In situations where consignment product is used in procedures, hospitals should think through how they can minimize the time between when the product is dispensed to a patient and when the change of ownership occurs and the appropriate compliance documentation is received. Theoretically, they may want to receive the T3 documentation and verify it before they dispense the drugs to their patient. Their relationship with their wholesaler and their level of established trust will likely come into play.
- On the returns side, consignment inventory actually simplifies procedures. A supplier can take back product without a paperwork exchange if the dispenser realizes they will not be using it.

## 1.4 Where Consignment Works Best.

A supplier has a product or group of products that he believes will sell if he can get them in front of end-users. The trick is that getting them in front of end-users means getting them stocked in retail establishments. Retailers are hesitant to stock the product because they do not have the same level of confidence in it as the supplier, they do not want to invest the money and risk getting stuck with something that may not sell. Because the supplier realizes in-store exposure is critical to getting his products sold, he offers to stock his product in their stores. This creates a condition of shared risk whereby the supplier risks the capital investment associated with the inventory while the customer risks dedicating retail space to the product. This also creates a condition of shared benefit because neither the supplier nor the customer will benefit until the product is sold to an end-user. This shared-risk/shared-benefit condition will often be enough to convince a customer to stock the product.

For a more specific example, consider a bicycle manufacturer that produces a wide range of bicycles ranging in price from a couple hundred dollars to several thousand dollars. He has customers (local independent bicycle shops) that stock his low-to-mid-priced models but are hesitant to stock the more expensive bikes because they do not have the confidence that their customers are willing to pay that much for a bike. And, if they do get a customer that wants a high-end bike, they could always special order it for them. The bicycle manufacturer strongly believes that getting his high-end bikes in the shops where customers can see and touch them is critical in driving up sales for these models as well as helping to promote his brand which ultimately drives up sales for the lower cost models. The solution? Well I think you can take it from here.

I consider this the classic consignment model because it is the best-case scenario for applying the consignment inventory model. It works well for:

- New and unproven products
- The introduction of existing product lines into new sales channels.
- Very expensive products where sales are questionable.

The key to all these examples is the combination of a high-degree of demand uncertainty from the customer's point of view, and a high degree of confidence in the sales potential from the supplier's point of view.

The consignment inventory model can also be effective with service parts for critical equipment where the customer would not stock certain service parts due to budget constraints or demand uncertainty. In this situation, consignment inventory allows the supplier to provide a higher service level (by having the parts immediately available), save expedited freight costs, and ensure the customer does not procure a replacement part from a competitor.

# 2 REQUIREMENT ANALYSIS

## 2.1 Current Scenario and Needs

The hospital Sri Sathya Sai Institute of Higher Medical Sciences, Whitefield aims to provide free medical care to the sick and ailing with the dedication, commitment, love and the best of skills, so that they will be cured in body, mind and spirit. The Mission is to provide high quality medical care free of charge to all irrespective of caste, creed, religion, and financial status in an atmosphere of love and care.

The facilities provided by the hospital are:-

- Cardiology
- Cardiac surgery
- Neurology
- Neuro surgery
- Radiology
- Anaesthesiology
- Lab and Blood Bank

This hospital already has the Information System in place to carry out the related work in each departments. The information system in place are HMIS AND INVENTORY MANAGEMENT SYSTEM.

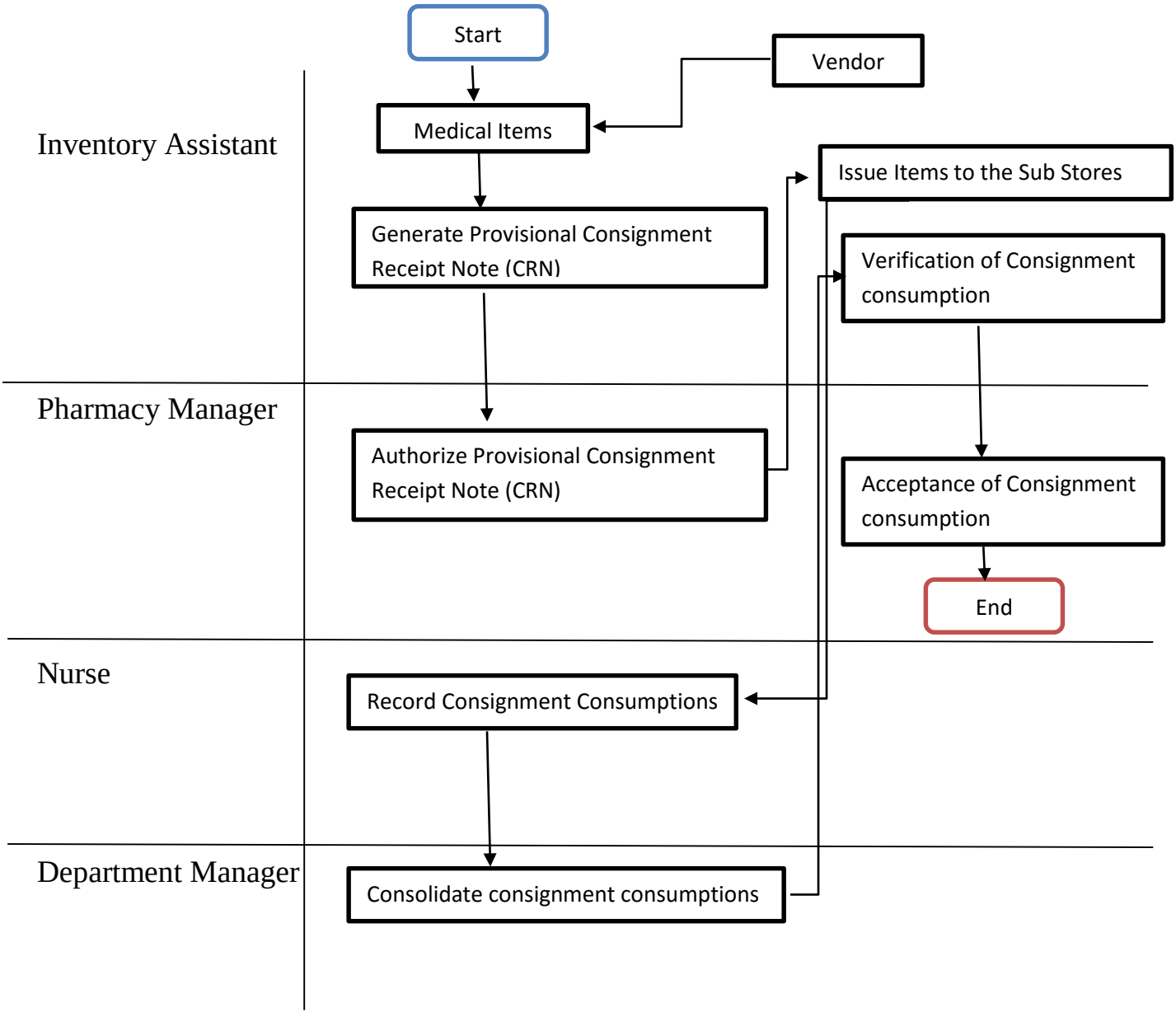
The currently existing system for consignment has written in Visual Basic (VB) Language. Visual basic is a proprietary programming language written by Microsoft, so programs written in Visual basic cannot, easily, be transferred to other operating systems.

Java is platform independent. If you write a Java application, it is easy to get it to run on Macs, Linux, and Windows. VB is tied to windows.

## 2.2 Process Modeling and Flowchart

Medical Items are delivered to the Pharmacy Store (Main Store) by the Vendor after which a provisional consignment receipt note (CRN) has been generated for the items being received. Authorization has to be done for the provisional CRN's by the Pharmacy Manager. Items can be issued to the Sub Stores and consumption of the items can be recorded for each patient. Consolidation of the consignments has been done for the consumption. At last, Verification and Acceptance of the consignment consumption will happen.

*Figure 1 Flowchart*



## 2.3 Use Case Modeling Concepts

### 2.3.1 What is a use case?

A use case is a written description of how users will perform tasks on GUI. It outlines, from a user's point of view, a system's behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled.

### 2.3.2 What is the use case model?

In software and systems engineering, a use case is a list of steps, typically defining interactions between a role which is called as Actor and a system, to achieve a goal. The actor can be a human, an external system, or time.

A use case diagram is a behavioral diagram, which aims to present a graphical overview of the functionalities provided by the system. It consists of a set of actions (use cases) that the concerned system can perform, one or more actors, and dependencies among them.

- **Use cases.** A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.
- **Actors.** An actor is a person, organization, or external system that plays a role in one or more interactions with your system. Actors are drawn as stick figures.
- **Associations.** Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case. Associations are modeled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicating the direction of the initial invocation of the relationship or to indicate the primary actor within the use case. The arrowheads are typically confused with data flow and as a result I avoid their use.
- **System boundary boxes (optional).** You can draw a rectangle around the use cases, called the system boundary box, to indicate the scope of the system. Anything within the box represents functionality that is in scope and anything outside the box is not. System boundary boxes are rarely used, although on occasion I have used them to identify which use cases will be delivered in each major release of a system.
- **Packages (optional).** Packages are UML constructs that enable you to organize model elements (such as use cases) into groups. Packages are depicted as file folders and can be used on any of the UML diagrams, including both use case diagrams and class diagrams. I use packages only when my diagrams become unwieldy, which generally implies they cannot be printed on a single page, to organize a large diagram into smaller ones.

## 2.4 Use Case Modeling of the Project

### 2.4.1 Provisional Consignment Receipt Note (CRN)

| | | | |
|---|---|---|---|
| Use Case ID: | CM01 | | |
| Use Case Name: | Provisional Consignment Receipt Note(CRN) | | |
| Created By: | Sateesh B | Last Updated By: | Sateesh B |
| Date Created: | 26-Oct-2017 | Date Last Updated: | 27-Oct-2017 |
| Actors: | Inventory Assistant | | |
| Goal | To enable Documentation for the Items Received i in Pharmacy | | |
| Scope | To create a Consignment Receipt Note for each Supplier from whom the items are received | | |
| Description: | Inventory Assistant should able to make CRNs for all the items received as one Consignment. | | |
| Preconditions: | 1. User should have logged into the system.<br>2. User should have privilege to make CRN in the system.<br>3. Relevant Conditions & Item list must have been created. | | |
| Postconditions: | The CRN Number may be generated for all the items received. | | |
| Trigger: | User chooses to make CRNs for the items received form the supplier | | |
| Normal Flow: | **1.0 Selecting Supplier and Receiver**<br>1. User chooses supplier name and Receiving location (Suppplied to).<br>2. User clicks ‘Add’ button for editing.<br>3. Receipt Date, Material Receipt Register (MRR) Number, DC Number should be entered and Backdate should be validated for the Receipt Date..<br>4. Screen Contains table header with S.NO., Brand Name, UOM, ID Type, ID No., Expiry Date, Shipped Qty, Quote Rate, Amount, Remarks and Cancel.<br>**1.1 Receipt through Main store**<br>1.2On choosing the receipt through Main store.<br>1.2Receipt Date should be current date.<br>1.2User search for item by clicking ‘Enter’ in BrandName for which the CRN can be generated and popup will come showing search box where user can search.<br>1.2User clicks on required items from the search results and it will be displayed in the current screen.<br>1.2User should able to ‘Add’ multiple rows for each item by clicking ‘LineAdd’ button.<br>1.2User should able to add multiple items by clicking ‘ItemAdd’ button.<br>1.2User should able to ‘Modify’ ID No., Expiry Date, Shipped Qty , Quote Rate by selecting existing CRN number by clicking ‘Modify’ button.<br>1.2User should able to cancel/remove the item by clicking ‘Cancel’ button.<br>1.2User clicks ‘Save’ button for CRN generation.<br>**1.2 Receipt through Sub store**<br>1. On choosing the receipt through Sub store.<br>2. Receipt Date is selected from ‘Dropdown Box’ where it shows consumption dates. | | |

| | |
|---|---|
| | 3. User selects CRN number as New Receipt and after clicking 'Add' button, the edit mode will be enabled and user enters MRR number DC Number and DC Date.<br>4. After selecting receipt date (which is Consumption date here), the consumed items will be displayed.<br>5. User cannot modify any fields of displayed items except Quotes rates and Remarks.<br>6. User should able to cancel/remove the item by clicking 'Cancel' button<br>7. User clicks 'Save' button for the CRN generation. |

## 2.4.2 Authorized Consignment Receipt Note (CRN)

| | | | |
|---|---|---|---|
| Use Case ID: | CM02 | | |
| Use Case Name: | Authorized Consignment Receipt Note(CRN) | | |
| Created By: | Sateesh B | Last Updated By: | Sateesh B |
| Date Created: | 26-Oct-2017 | Date Last Updated: | 27-Oct-2017 |
| Actors: | Pharmacy Manager | | |
| Goal | To enable the checking of the generated CRNs and authorize the same. | | |
| Scope | To check and make required modification, if any, to the generated CRNs for the items received and authorize the same. | | |
| Description: | Pharmacy Manager should able to check the CRNs and authorize the same. | | |
| Preconditions: | 4. User should have logged into the system.<br>5. User should have privilege to make CRN in the system.<br>6. Relevant Conditions & CRNs must have been generated. | | |
| Postconditions: | All the items should be authorized in respective CRN. | | |
| Trigger: | User chooses to authorize each item received in respective CRN. | | |
| Normal Flow: | 2.1 **Selecting Supplier and Receiver**<br>5. User chooses supplier name and Receiving location (Suppplied to).<br>6. User selects CRN number from 'Dropdown Box'.<br>7. After selecting CRN number, the items documented under that CRN number will be displayed.<br>8. Receipt Date, Material Receipt Register (MRR) Number, DC Number should also be displayed.<br>9. Screen Contains table header with S.NO, Brand Name, UOM, ID Type, ID No., Expiry Date, Shipped Qty, Quote Rate, Amount, Remarks, Authorize and Cancel.<br>10. User is allowed to authorize the provisional CRNs and not allowed to authorize items already authorized.<br>11. User can only modify the CRN and can't be added any item to the CRN by the user.<br>12. User clicks on 'Checkbox' for the authorization which is under 'Authorize' Column for each item.<br>13. User clicks 'Save' button for saving the authorized CRNs. | | |

## 2.4.3 Consignment Issues to Sub store

| | | | |
|---|---|---|---|
| Use Case ID: | CM03 | | |
| Use Case Name: | Consignment Issues to Sub store | | |
| Created By: | Sateesh B | Last Updated By: | Sateesh B |
| Date Created: | 26-Oct-2017 | Date Last Updated: | 27-Oct-2017 |
| Actors: | Pharmacy Manager | | |
| Goal | To enable the issuing of the required items to the Sub stores. | | |
| Scope | To issue required items to all the Sub stores (Departments). | | |
| Description: | Pharmacy Manager should able to issue the required items to the Sub stores. | | |
| Preconditions: | 7. User should have logged into the system.<br>8. User should have privilege to issue the items in the system.<br>9. Relevant Conditions & Item list must have been created. | | |
| Postconditions: | All the items required by the Sub store should be issued to it. | | |
| Trigger: | User chooses to issue the items required by the Sub store. | | |
| Normal Flow: | **3.0. Selecting My Location and To Location**<br>**4.0.** User chooses My Location from where the items are to be transferred and To Location to where the items are to be transferred.<br>**5.0.** Screen Contains table header with S.NO, Brand Name, Vendor, UOM, ID Type, ID No., Expiry Date, Stock, Issue Qty, Remarks.<br>**3.1. New Transfer**<br>1. User clicks 'Add' button and a popup will come showing search box where the user can search the required items.<br>2. User selects required items from the search results by clicking on it.<br>3. Selected item row will be displayed in the current screen and user can edit only 'Quantity' and 'Remarks'<br>4. User issues items by specifying quantity of it and which should be less than the quantity of the stock.<br>5. User clicks 'Save' button for saving the issue.<br><br>**3.2. Posted Transfer**<br>1. On choosing posted transfer by clicking 'Posted Transfer' radio button.<br>2. User should able to reduce the stock quantity by making 'Zero' under stock column for each item.<br>3. User clicks 'Receipt' button for transferring the issued items.<br>4. After clicking 'Receipt' button, a popup will come showing username and password to be filled for accepting the items.<br>5. After filling 'UserName' and 'Password', the 'Submit' button will be clicked.<br>6. After submission a new column 'Accept' will be added to already existing table in the screen.<br>7. User clicks on 'Check Box' under 'Accept' column for each item telling the item is accepted by the Sub store.<br>8. User clicks 'Save' button for saving the transaction. | | |

## 2.4.4 Consignment Consumption

| | | | |
|---|---|---|---|
| Use Case ID: | CM04 | | |
| Use Case Name: | Consignment Consumption | | |
| Created By: | Sateesh B | Last Updated By: | Sateesh B |
| Date Created: | 26-Oct-2017 | Date Last Updated: | 27-Oct-2017 |
| Actors: | Nurse | | |
| Goal | To enable the documentation of consumption of items by the patients. | | |
| Scope | To record the consumption details of all the patient in all the Department. | | |
| Description: | Nurse should able to capture the consumption details of respective Department. | | |
| Preconditions: | 10. User should have logged into the system.<br>11. User should have privilege to record the consumptions in the system.<br>12. Relevant Conditions & Item list must have been created. | | |
| Postconditions: | All the items consumed should be recorded. | | |
| Trigger: | User chooses to record the consumed items for each patient in the respective Department. | | |
| Normal Flow: | **4.0. Selecting Sub store and To Consumption Date**<br>**5.0.** User chooses Sub store and Consumption Date.<br>**6.0.** User enters 'Patient ID' in the patient ID 'Input Box' and Clicks 'Get' button for getting his/her details.<br>**7.0.** 'Patient ID' is validated, if it is valid then the related data (here Name, Gender, Age, Admission no.,) will be fetched and displayed in the respective 'InputBox' .<br>**8.0.** Surgery Name and Surgeon Name will also be displayed in their respective 'InputBox'.<br>**9.0.** Screen Contains table header with S.NO, ItemName, Vendor, UOM, ID Type, ID No., Cons UOM, Cons Qty, New Qty, Stock UOM.<br>**10.0.** User clicks 'Add' button after which a popup will come showing a search box where user search for the required item and clicks on required item from the search results.<br>**11.0.** Selected item will be displayed in the current screen and User enters the Consumption Quantity for the selected item.<br>**12.0.** The background color of the selected item row will change depending on the location where it is received.<br>**13.0.** User clicks 'Save' button and a new tab will be added to the screen with patient ID as name.<br>**14.0.** User clicks 'Exit' button for coming out of the screen. | | |

## 2.4.5 Consignment Consolidation

| | | | |
|---|---|---|---|
| Use Case ID: | CM05 | | |
| Use Case Name: | Consignment Consolidation | | |
| Created By: | Sateesh B | Last Updated By: | Sateesh B |

| | | | |
|---|---|---|---|
| Date Created: | 26-Oct-2017 | Date Last Updated: | 27-Oct-2017 |
| Actors: | Department Manager | | |
| Goal | To enable the consolidation of consumed items by the patients. | | |
| Scope | To verify the Item Consumption detail of all patients and consolidate it. | | |
| Description: | Department Manager should able to verify and consolidate the consumption details of all the patients. | | |
| Preconditions: | 13. User should have logged into the system.<br>14. User should have privilege to verify and consolidate the consumptions in the system.<br>15. Relevant Conditions &Item Consumptions must have been recorded. | | |
| Postconditions: | All the items consumed should be verified and consolidated. | | |
| Trigger: | User chooses to consolidate the consumed items. | | |
| Normal Flow: | **5.0 Selecting Substore and Consolidated Date**<br>**15.0.** User chooses Substore, Consumption Date and Consumption Date.<br>**16.0.** After choosing, user clicks 'ShowPatient' button after which patient details and items consumed by these patients are displayed.<br>**17.0.** Screen contains patient table and Item table.<br>**18.0.** Patient table has S.No, Patient ID, Patient Name, Surgery Name, Doctor fields. Item table has S.No., Item Name, Id/Batch, Stock, Qty, Consol fields.<br>**19.0.** User clicks on 'Check Box' under 'Consol' column for each item for consolidation to complete for that item.<br>**20.0.** User clicks 'Save' button for consolidation to complete.<br>**21.0.** User clicks 'Exit' button for coming out of the screen. | | |

## 2.4.6 Verification of Consignment Consumption

| | | | |
|---|---|---|---|
| Use Case ID: | CM06 | | |
| Use Case Name: | Verification of Consignment Consumption | | |
| Created By: | Sateesh B | Last Updated By: | Sateesh B |
| Date Created: | 26-Oct-2017 | Date Last Updated: | 27-Oct-2017 |
| Actors: | Pharmacy Manager | | |
| Goal | To enable the verification of Consolidated items. | | |
| Scope | To verify all the consolidated consumed items. | | |
| Description: | Pharmacy Manager should able to verify the consolidated items. | | |
| Preconditions: | 16. User should have logged into the system.<br>17. User should have privilege to verify the consolidated consumptions in the system.<br>18. Relevant Conditions &Consumed Items must have been consolidated. | | |
| Postconditions: | All the consolidated should be verified. | | |

| | |
|---|---|
| Trigger: | User chooses to verify the consolidated items. |
| Normal Flow: | **6.0 Selecting Sub store**<br>**22.0.** User chooses Sub store from the 'Dropdown Box' and clicks 'Get' button.<br>**23.0.** After clicking, it will show all the items and patient details who consumed it in a date wise.<br>**24.0.** Items should be in one of the consumed, verified, accepted and outstanding states.<br>**25.0.** Only items which are in consumed state can be verified .<br>**26.0.** User clicks 'Add' button for editing and clicks on 'CheckBox' for state to change to 'Verified'.<br>**27.0.** User clicks 'Save' button for saving the verifications.<br>**28.0.** User clicks 'Exit' button for coming out the screen. |

## 2.4.7 Acceptance of Consignment Consumption

| | | | |
|---|---|---|---|
| Use Case ID: | CM07 | | |
| Use Case Name: | Acceptance of Consignment Consumption | | |
| Created By: | Sateesh B | Last Updated By: | Sateesh B |
| Date Created: | 26-Oct-2017 | Date Last Updated: | 27-Oct-2017 |
| Actors: | Pharmacy Manager | | |
| Goal | To enable the Acceptance of Verified items. | | |
| Scope | To accept all the verified Consumed items. | | |
| Description: | Pharmacy Manager should able to accept the verified items. | | |
| Preconditions: | 19. User should have logged into the system.<br>20. User should have privilege to accept the verified consumptions in the system.<br>21. Relevant Conditions &Consumed Items must have been verified.. | | |
| Postconditions: | All the verified items should be accepted. | | |
| Trigger: | User chooses to accept the verified items. | | |
| Normal Flow: | **7.0 Selecting Sub store**<br>**29.0.** User chooses Sub store from the 'Dropdown Box' and clicks 'Get' button.<br>**30.0.** After clicking, it will show all the items and patient details who consumed it in a date wise.<br>**31.0.** Items should be in one of the consumed, verified, accepted and outstanding states.<br>**32.0.** Only items which are in verified state can be accepted.<br>**33.0.** User clicks 'Add' button for editing and clicks on 'Checkbox' for state to change to 'accepted'.<br>**34.0.** User clicks 'Save' button for saving the changes.<br>**35.0.** User clicks 'Exit' button for coming out the screen. | | |

## 2.5 Sequence Diagram

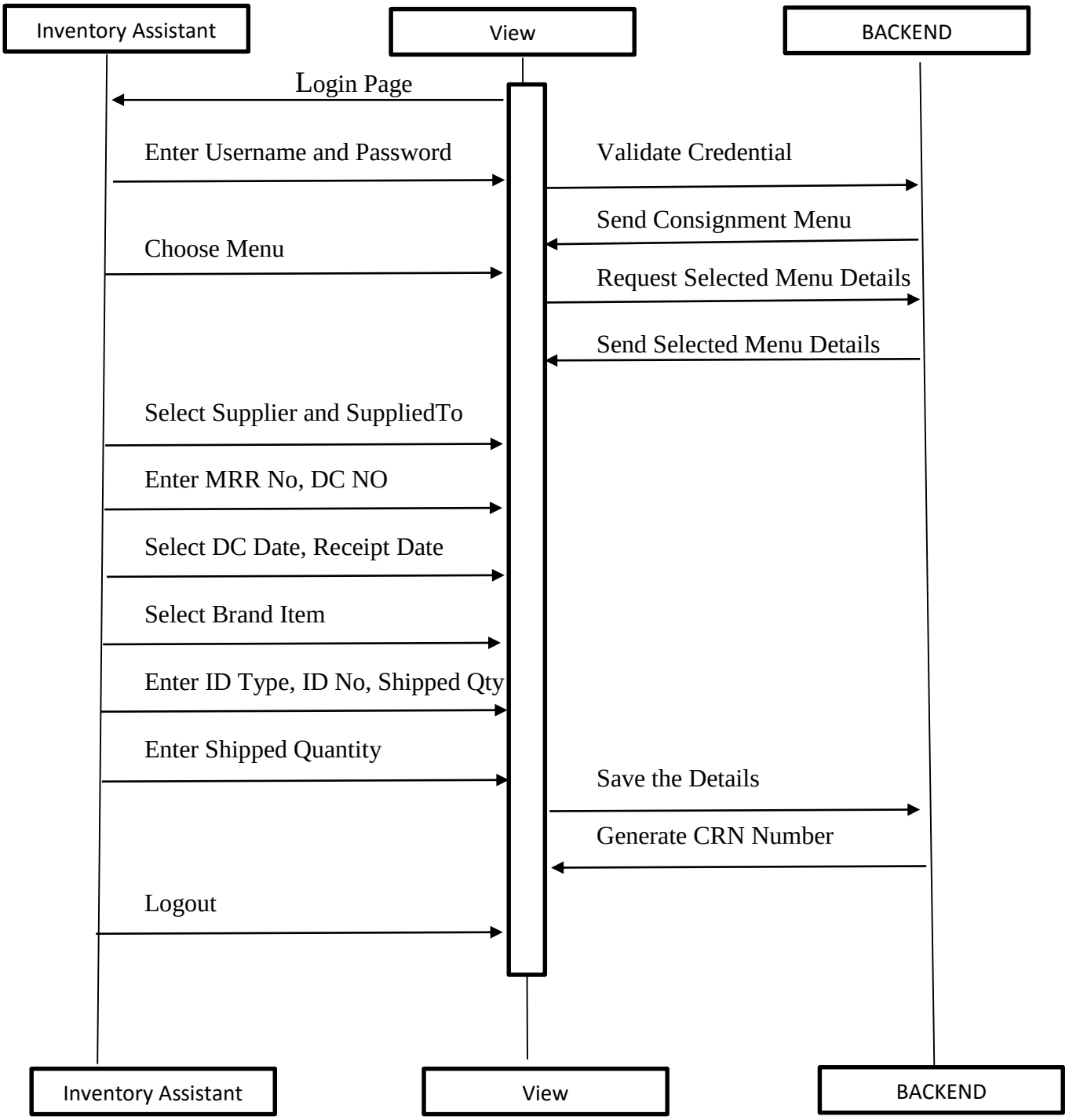



*Figure 2 Sequence diagram for PCRN*

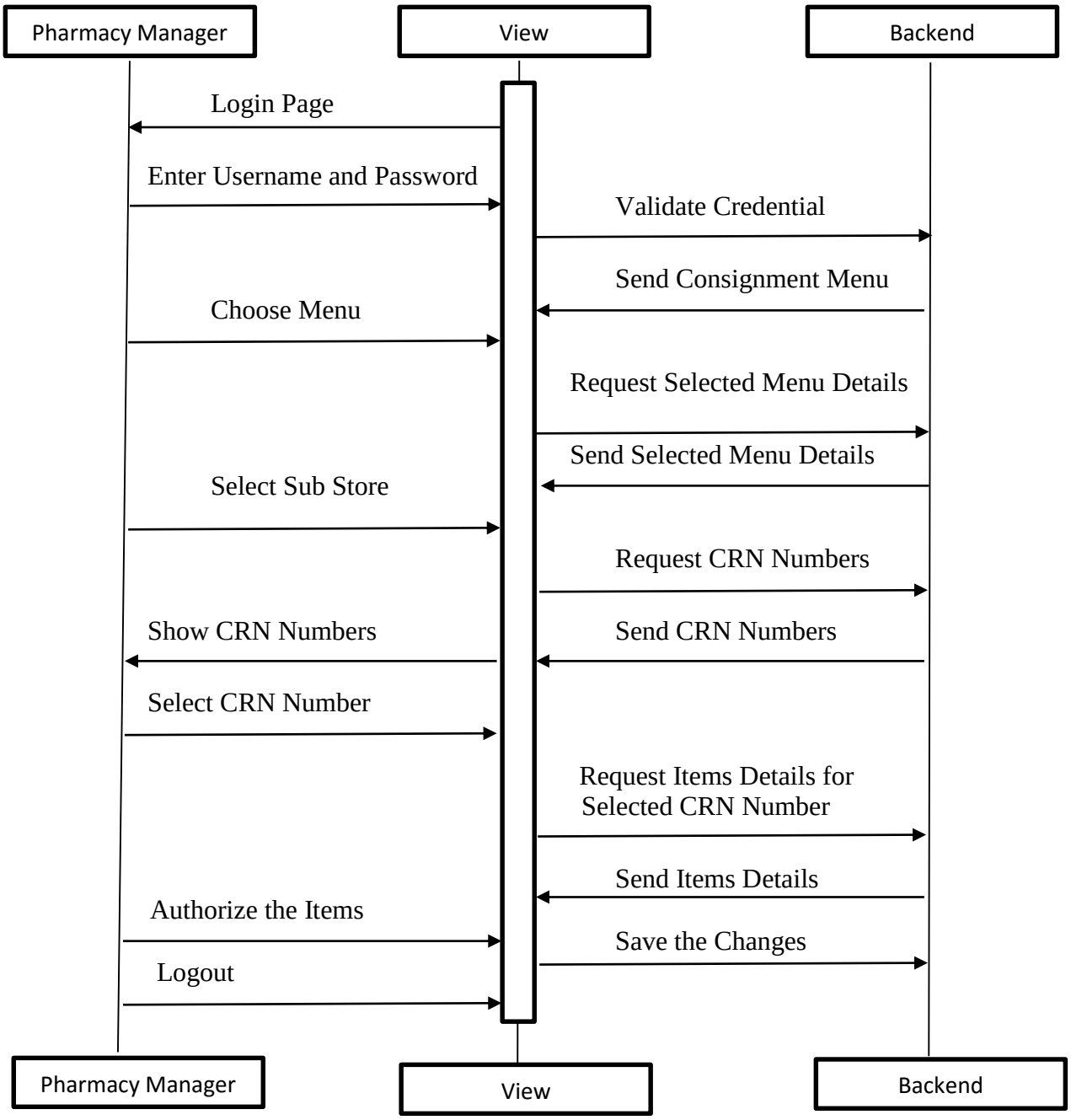



*Figure 3 Sequence diagram for ACRN*

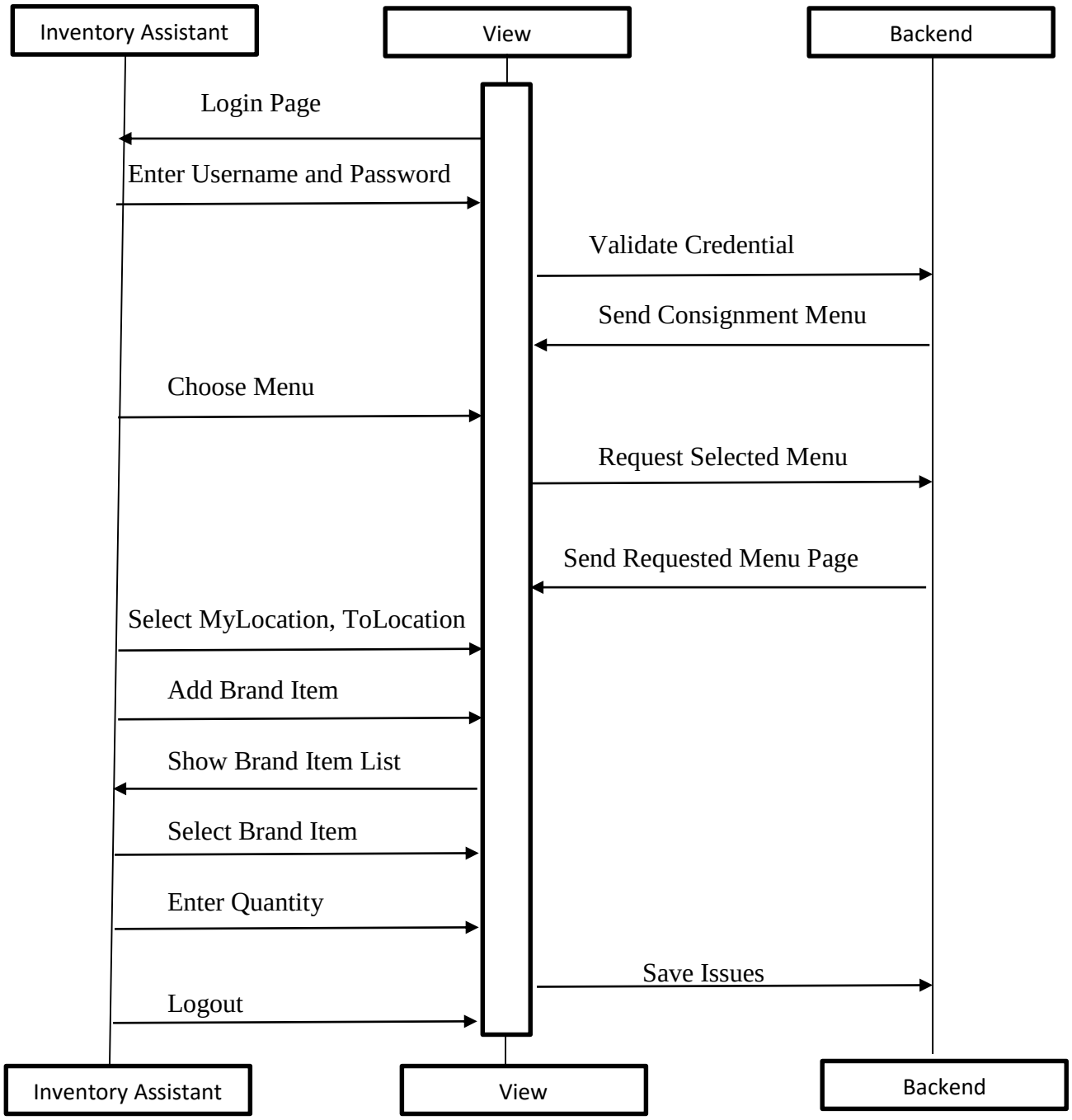



*Figure 4 Sequence diagram for Issues*

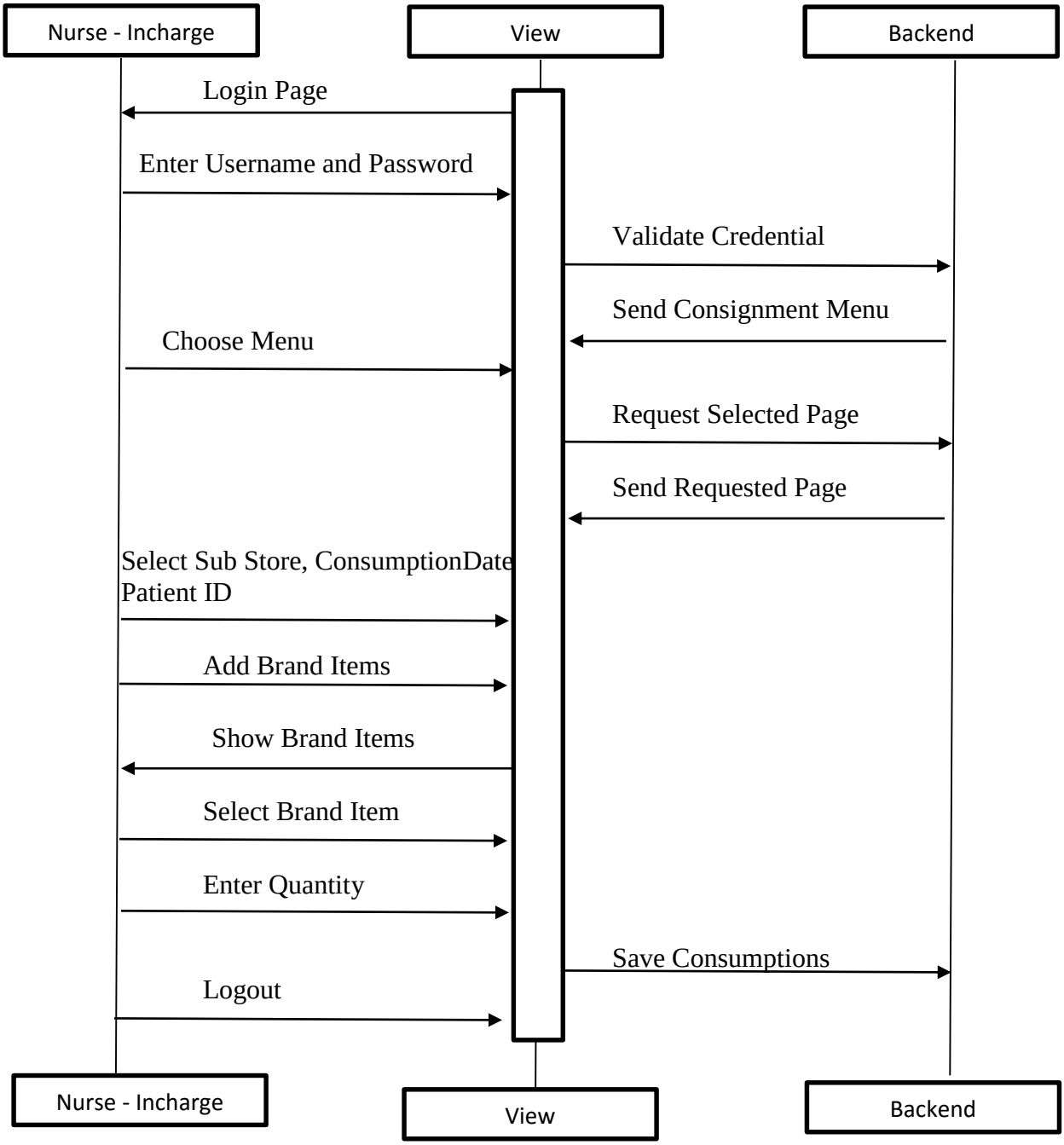



*Figure 5 Sequence diagram for consumptions*

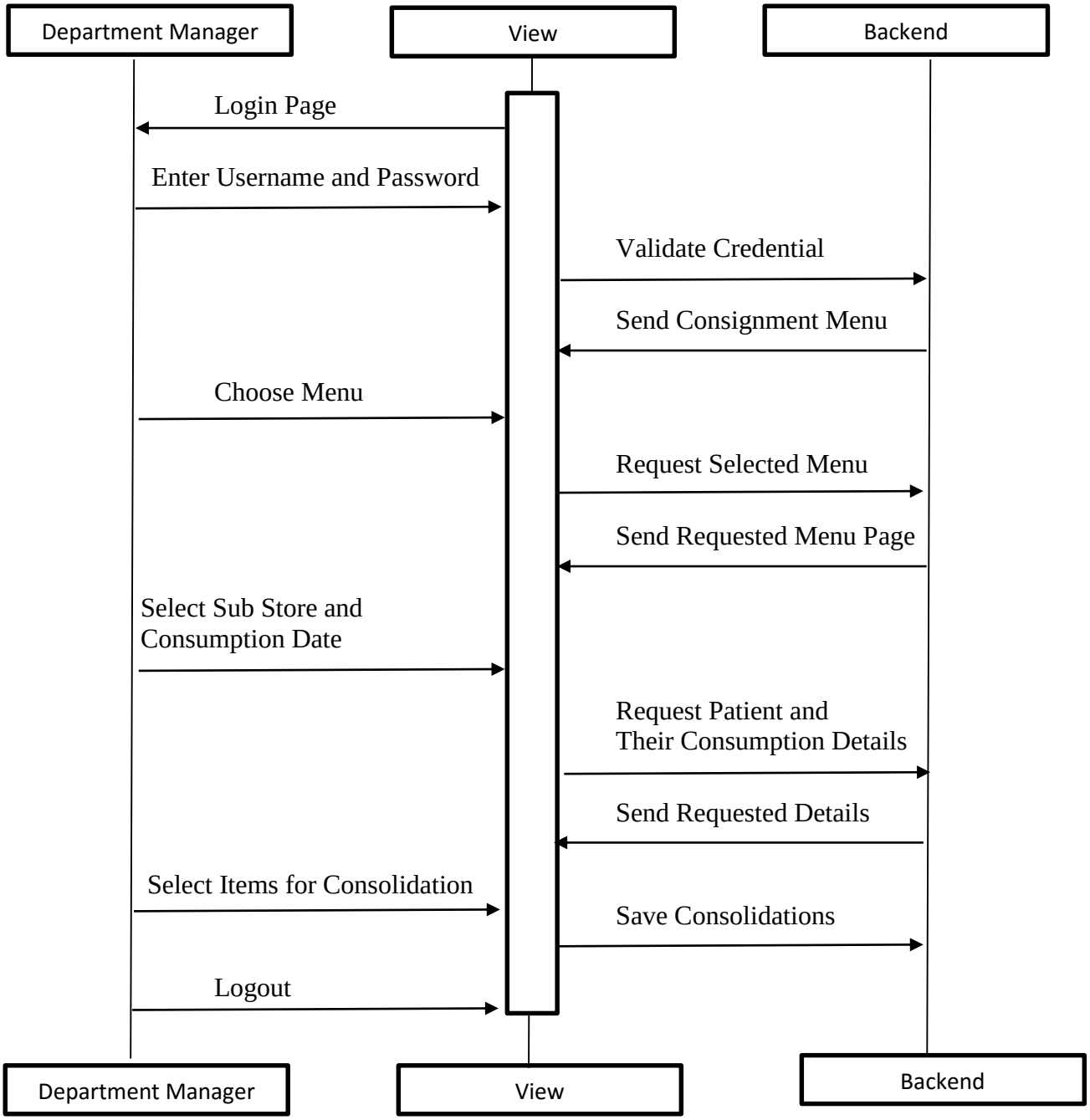



*Figure 6 Sequence diagram for Consolidations*

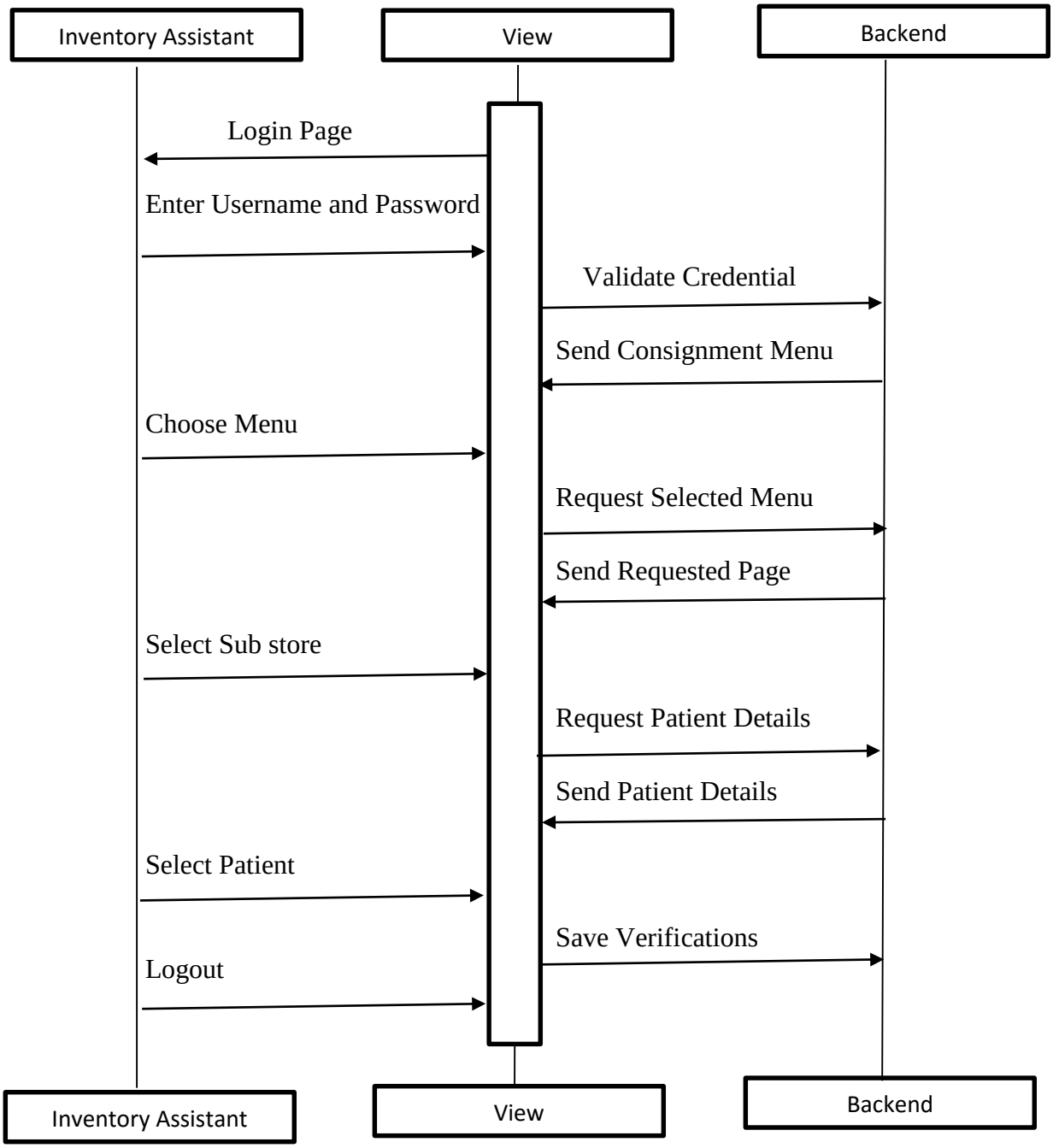



*Figure 7 Sequence diagram for Verification*

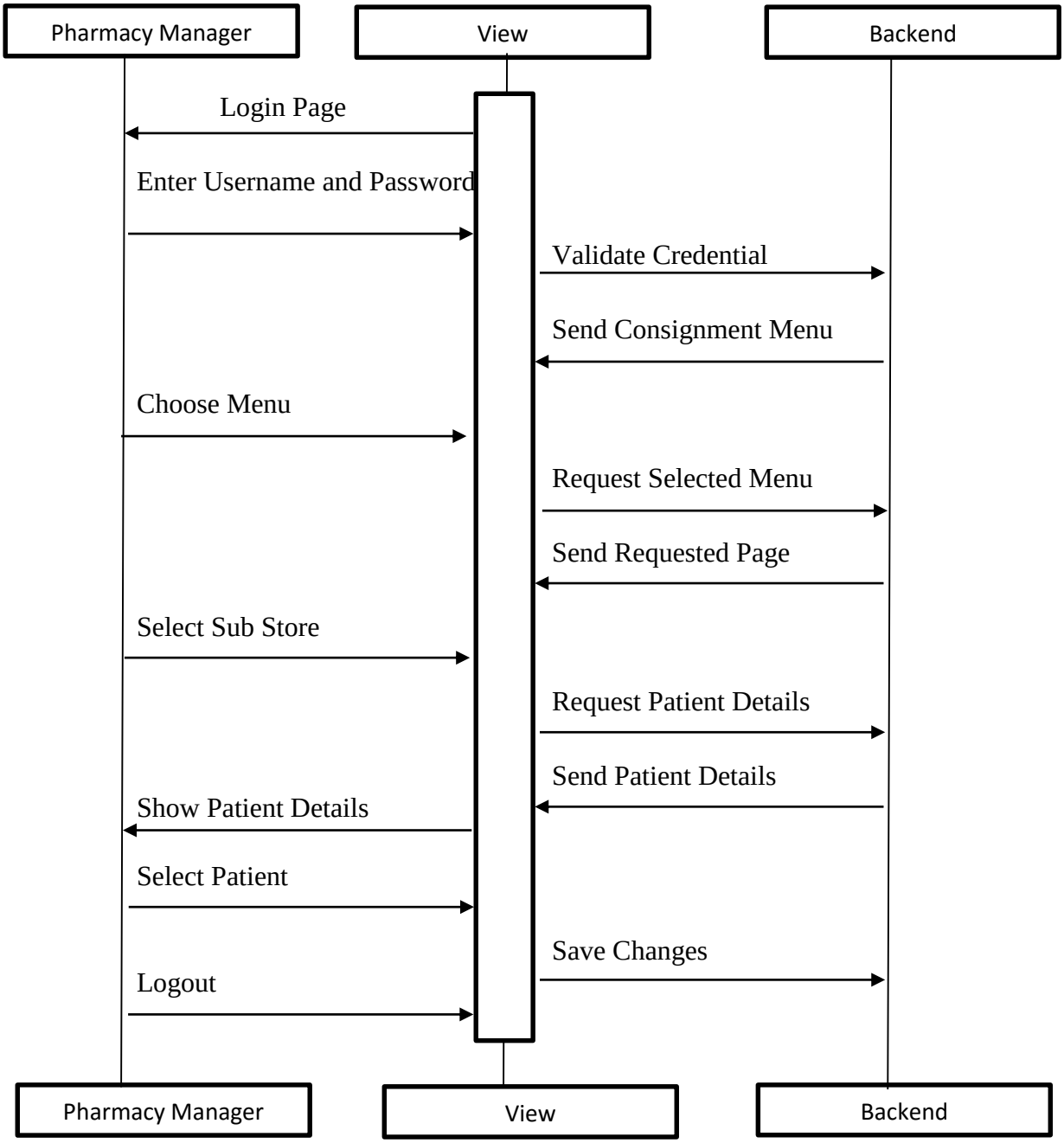



*Figure 8 Sequence diagram for Acceptance*

## 2.6 Literature Review

### 2.6.1 Spring Framework

Spring is the most popular application development framework for enterprise Java. Millions of developers around the world use Spring Framework to create high performing, easily testable, and reusable code. Spring framework is an open source Java platform. It was initially written by Rod Johnson and was first released under the Apache 2.0 license in June 2003.

The core features of the Spring Framework can be used in developing any Java application, but there are extensions for building web applications on top of the Java EE platform. Spring framework targets to make J2EE development easier to use and promotes good programming practices by enabling a POJO-based programming model.

**Dependency Injection (DI)**

The technology that spring is most identified with is the Dependency Injection (DI) flavor of Inversion of Control. The Inversion of Control (IoC) is a general concept, and it can be expressed in many different ways. Dependency Injection is merely one concrete example of Inversion of Control.

When writing a complex Java application, application classes should be as independent as possible of other Java classes to increase the possibility to reuse these classes and to test them independently of other classes while unit testing. Dependency Injection helps in gluing these classes together and at the same time keeping them independent.

**Benefits of Using the Spring Framework Following is the list of few of the great benefits of using Spring Framework:**

- Spring enables developers to develop enterprise-class applications using POJOs. The benefit of using only POJOs is that you do not need an EJB container product such as an application server but you have the option of using only a robust servlet container such as Tomcat or some commercial product.
- Spring is organized in a modular fashion. Even though the number of packages and classes are substantial, you have to worry only about the ones you need and ignore the rest.
- Spring does not reinvent the wheel, instead it truly makes use of some of the existing technologies like several ORM frameworks, logging frameworks, JEE, Quartz and JDK timers, and other view technologies.
- Testing an application written with spring is simple because environment dependent code is moved into this framework. Furthermore, by using JavaBean style POJOs, it becomes easier to use dependency injection for injecting test data.

### 2.6.2 Hibernate

Hibernate is an Object-Relational Mapping (ORM) solution for JAVA. It is an open source persistent framework created by Gavin King in 2001. It is a powerful, high performance Object-Relational Persistence and Query service for any Java Application.
Hibernate maps Java classes to database tables and from Java data types to SQL data types and relieves the developer from 95% of common data persistence related programming tasks.
Hibernate sits between traditional Java objects and database server to handle all the works in persisting those objects based on the appropriate O/R mechanisms and patterns.

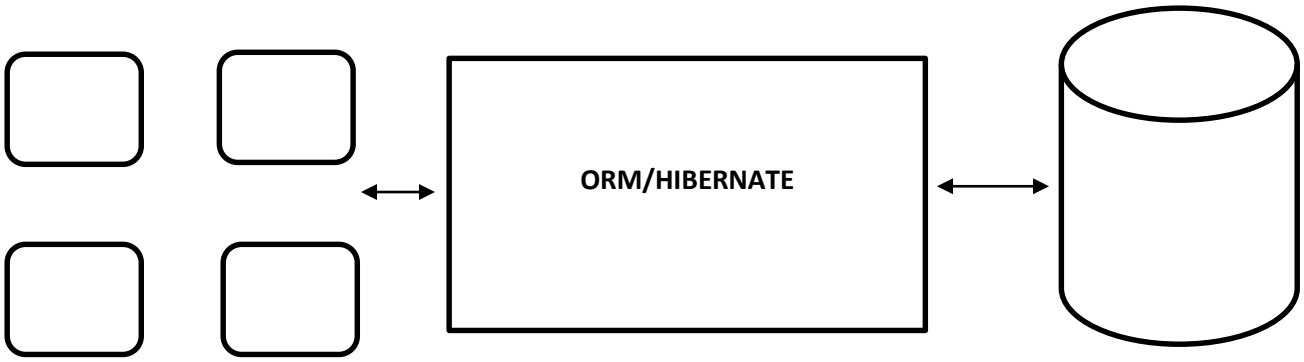



*Figure 9 Hibernate*

## Hibernate Advantages

- Hibernate takes care of mapping Java classes to database tables using XML files and without writing any line of code.
- Provides simple APIs for storing and retrieving Java objects directly to and from the database.
- If there is change in the database or in any table, then you need to change the XML file properties only.
- Abstracts away the unfamiliar SQL types and provides a way to work around familiar Java Objects.
- Hibernate does not require an application server to operate.
- Manipulates Complex associations of objects of your database.
- Minimizes database access with smart fetching strategies.
- Provides simple querying of data.

# 3 DEVELOPMENT PLATFORM

- **Hardware:**
  - Intel® Core (TM) i5-4590 CPU@3.30GHz
  - 8 GB RAM
  - 1 TB Hard Disk

- **Operating System:** Linux ( Ubuntu 16.04 LTS )
- **Software:**
    - Eclipse Neon
    - Java Servlets
    - JSP
    - AngularJS
    - Apache Tomcat
    - HTML, CSS and Bootstrap
    - Oracle DB 11g Express Edition

# 4 SYSTEM DESIGN

## 4.1 Architecture

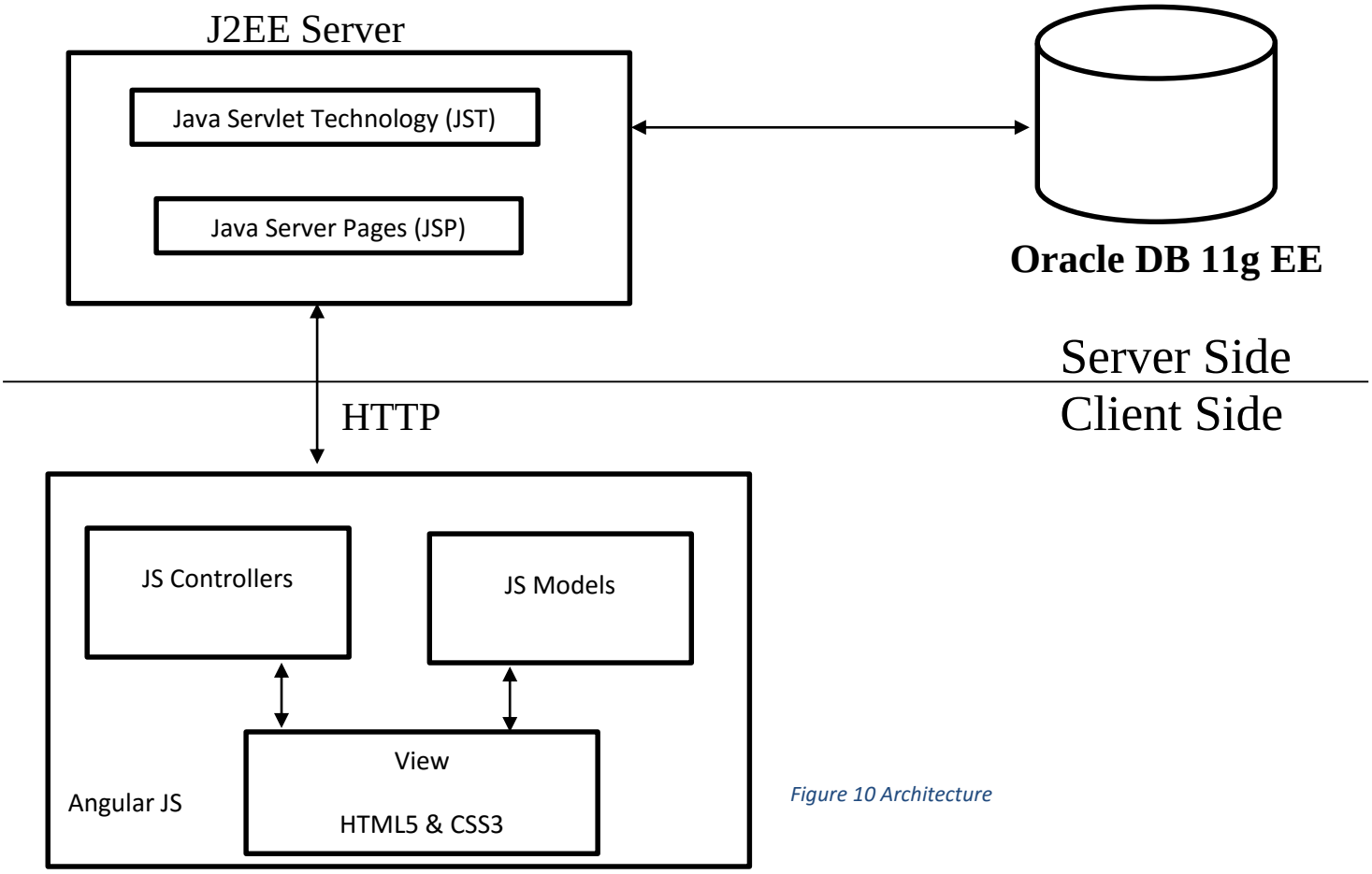



*Figure 10 Architecture*

## 4. 2 Single Page Application Execution

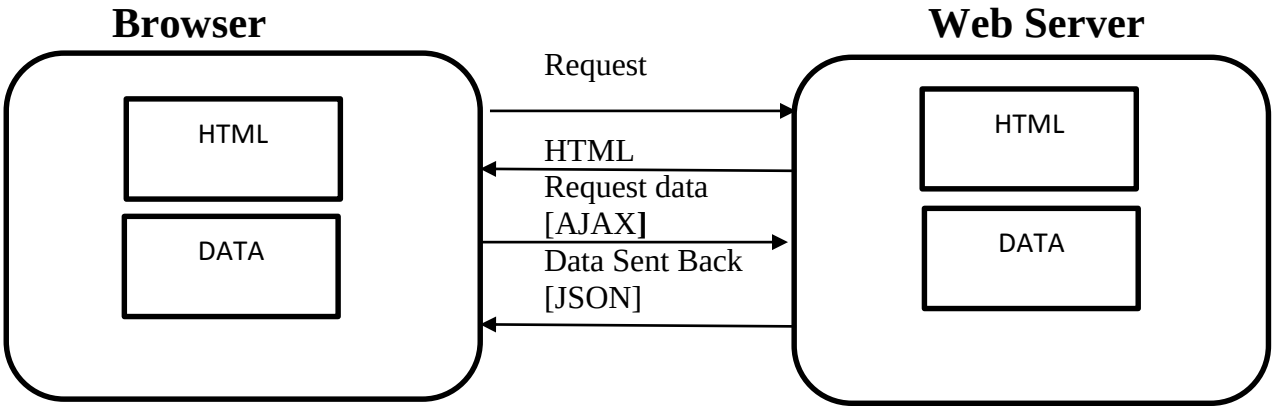



*Figure 11 SPA Execution*

## What is SPA (Single Page Application)?

It is web application in which the majority of interactions are handled on the client without the need to reach a server, with a goal of providing a more fluid user experience.

## SPA using AngularJS

AngularJS is the most popular and dominant JavaScript framework for professional web development. It is well suited for small, large and any sized web app and web application.

- ✓ It follows MVC pattern which helps to organize your web apps or web application properly.
- ✓ It extends HTML by attaching directives to your HTML markup with new attributes or tags and expressions in order to define very powerful templates.
- ✓ It also allows you to create your own directives.
- ✓ AngularJS supports two-way data binding, any change in model will update the view and vice versa without any DOM manipulation.
- ✓ AngularJS uses plain old JavaScript objects (POJO).
- ✓ AngularJS can be easily integrate with any other java script library.

**Routing Support:** Single Page Apps (SPAs) are everywhere nowadays. With the advent of HTML5 and its related APIs, we don't want to redirect our users to a new page every time they click on something. Instead, we want to load the content asynchronously on the same page and just change the URL in the browser to reflect

it. Lots of other popular websites are already doing this, such as Twitter and the Chrome app store.

It makes the user feel as if they are interacting with a desktop app. With AngularJS we can implement a Single Page App very easily with minimum effort. In fact, AngularJS was built with these things in mind; you can basically create different views for different URLs. AngularJS will then load the appropriate view in the main page when a specific URL is requested.

The routing feature also fosters maintainability. This is because we are logically dividing our app into different parts and thereby making it more maintainable.

## Servlets Architecture

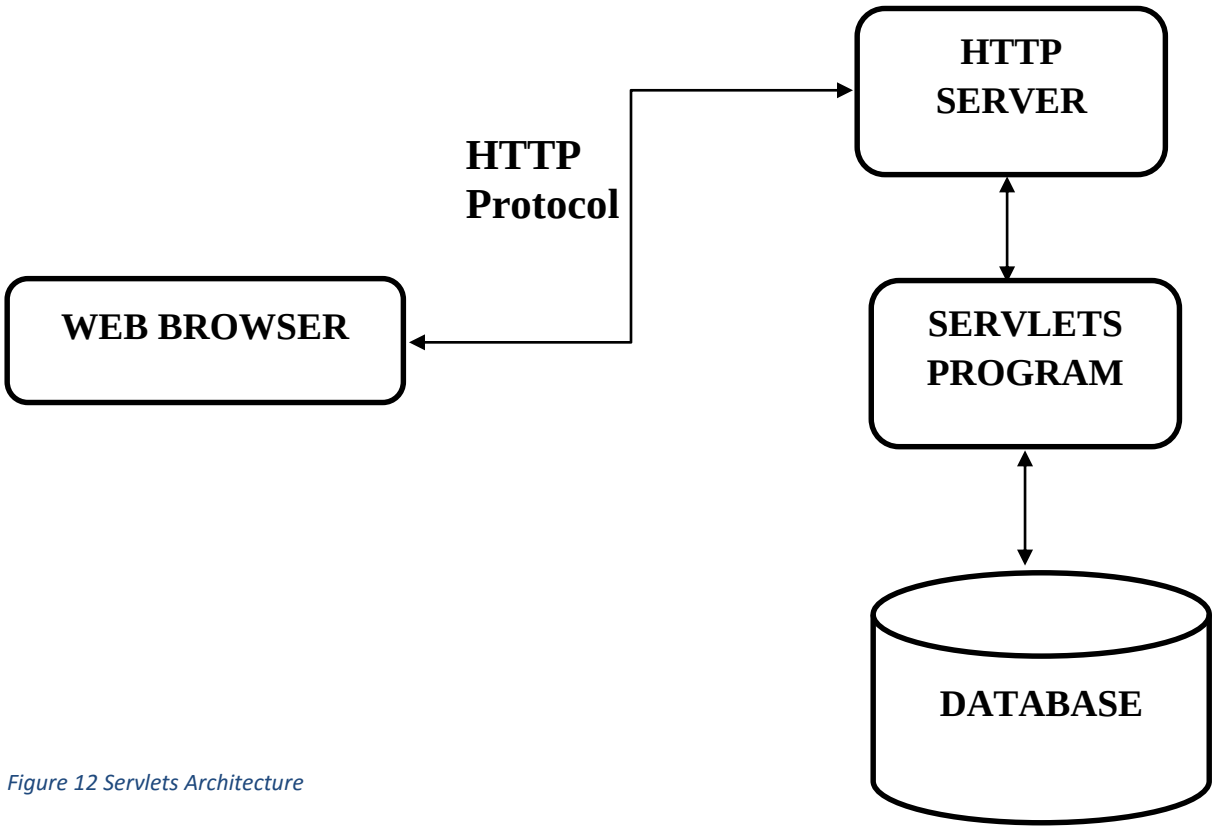



*Figure 12 Servlets Architecture*

### What are Servlets?

Java Servlets are programs that run on a Web or Application server and act as a middle layer between a requests coming from a Web browser or other HTTP client and databases or applications on the HTTP server.

Using Servlets, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

Java Servlets often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But Servlets offer several advantages in comparison with the CGI.

- Performance is significantly better.
- Servlets execute within the address space of a Web server. It is not necessary to create a separate process to handle each client request.
- Servlets are platform-independent because they are written in Java.
- Java security manager on the server enforces a set of restrictions to protect the resources on a server machine. So servlets are trusted.
- The full functionality of the Java class libraries is available to a servlet. It can communicate with applets, databases, or other software via the sockets and RMI mechanisms that you have seen already.

**A servlet life cycle can be defined as the entire process from its creation till the destruction. The following are the paths followed by a servlet**

- The servlet is initialized by calling the init () method.
- The servlet calls service() method to process a client's request.
- The servlet is terminated by calling the destroy() method.
- Finally, servlet is garbage collected by the garbage collector of the JVM.

# 5 DATABASE DESIGN

## ER-Diagram

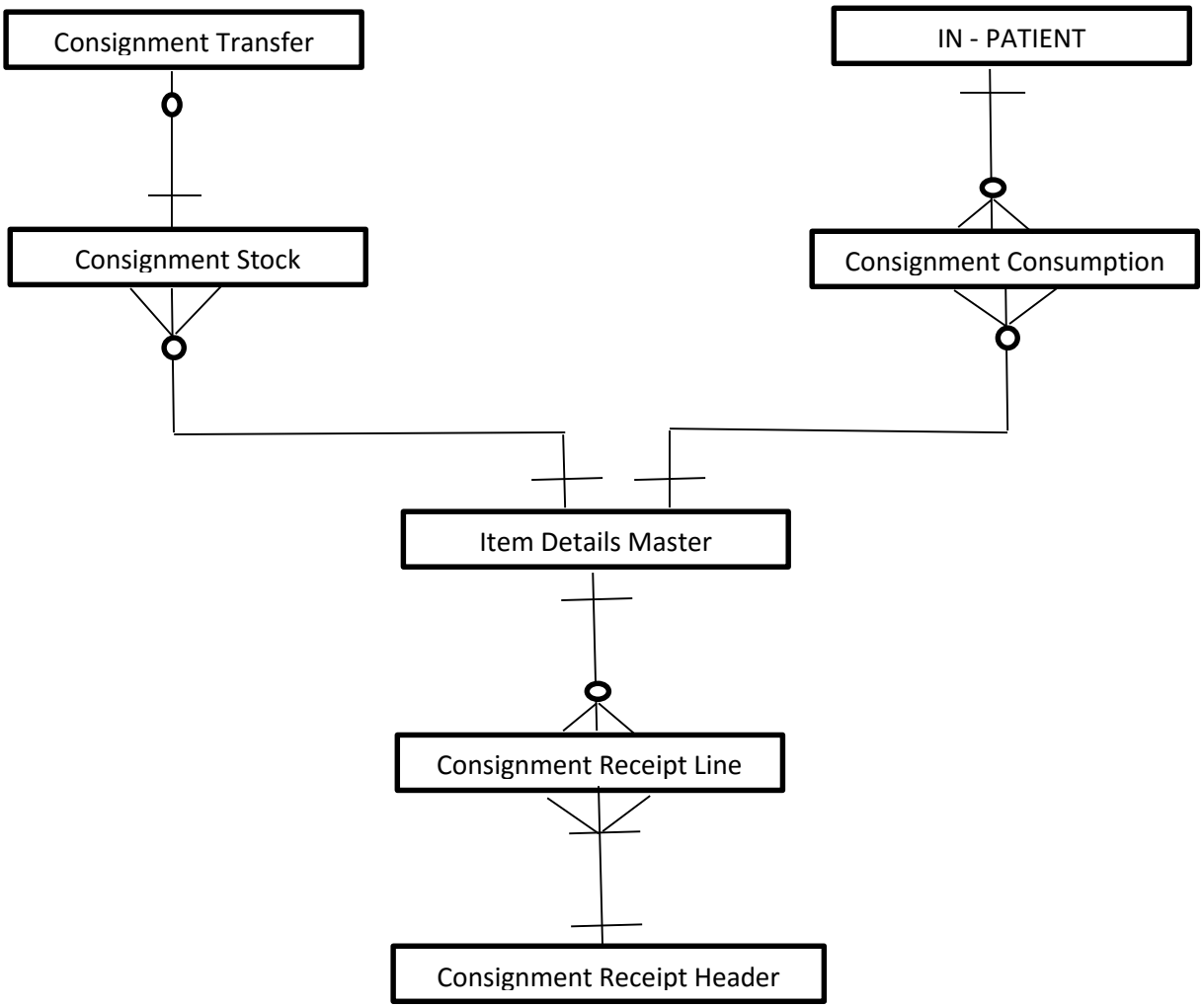



*Figure 13 ER Diagram*

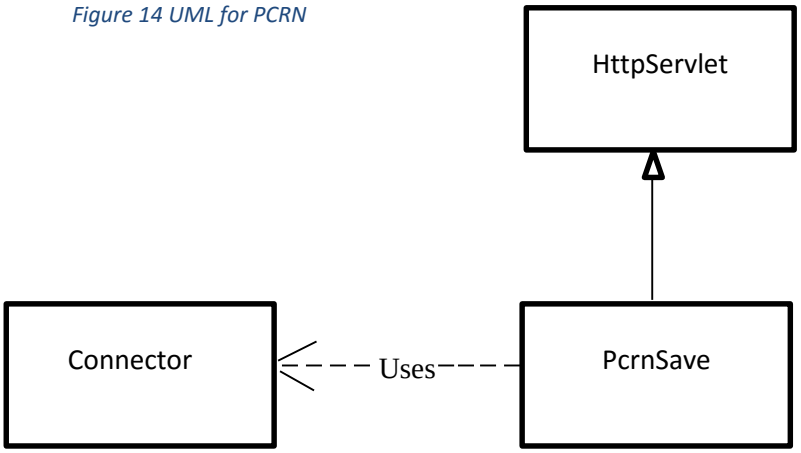
# 6 DETAIL DESIGN

**The Project has following modules:**

- Provisional Consignment Receipt Note
- Authorized Consignment Receipt Note
- Consignment Issues to Sub Store
- Consignment Consumption
- Consignment Consolidation
- Verification of Consignment Consumptions
- Acceptance of Consignment Consumptions

## 6.1 UML Diagrams and Details

*Figure 14 UML for PCRN*



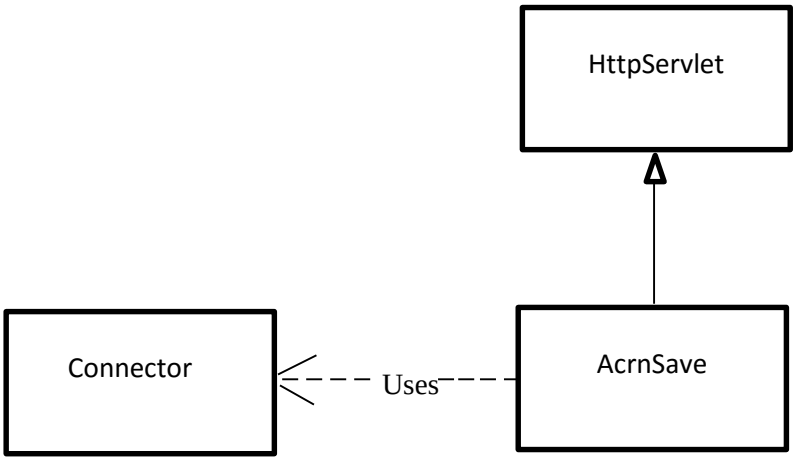




*Figure 15 UML for ACRN*

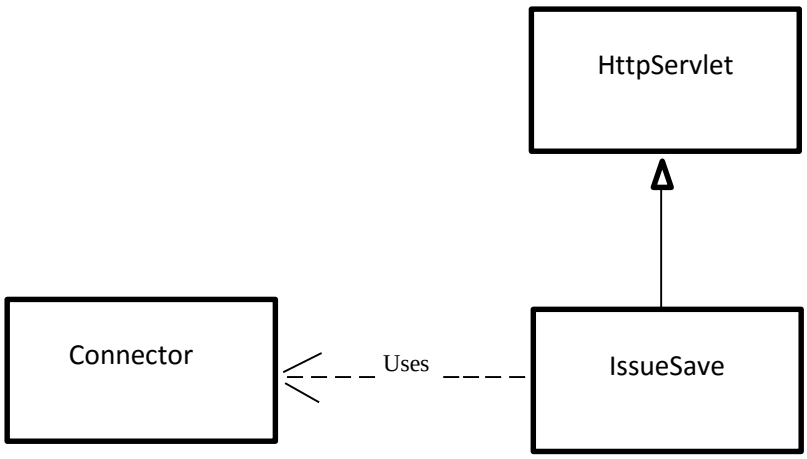



*Figure 16 UML for Issues*

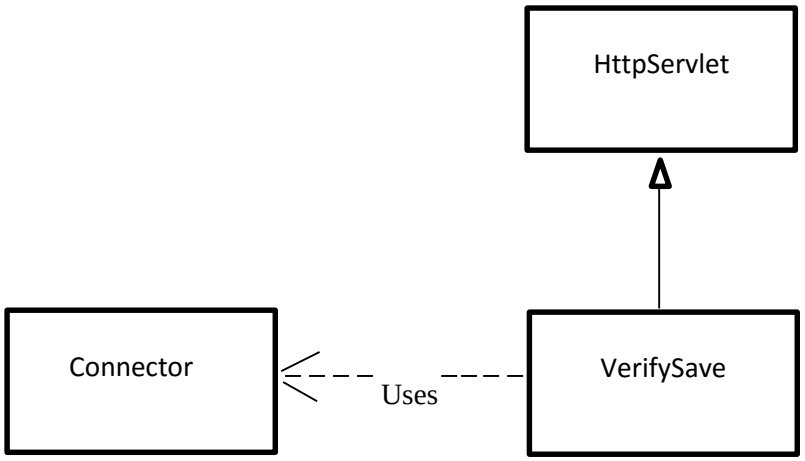



*Figure 17 UML for Verification*

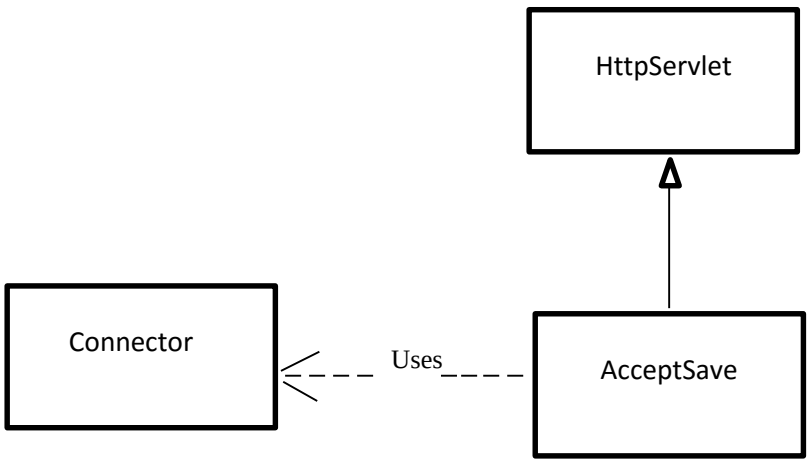



*Figure 18 UML for Acceptance*

**Connector:** It is a Java class used for connecting to the database and has getConnection() method which is static. All the modules classes uses this class for establishing the connection with the database.

**HttpServlet:** The HttpServlet is an abstract class that is suitable for processing HTTP requests. HTTP request is encapsulated as HttpServletRequest object and HTTP response is encapsulated as HttpServletResponse object. All other classes of the modules extends this class for using its functionalities.

## 6.2 Code Snippets

**Connector.java**

```
public class Connector {

   private static Connection conn;
   private static String url;

   /**
    * creates a Connection to the database if already a Connection exists.
    * @return Connection the connection to the database.
    * @throws java.lang.ClassNotFoundException
    */

   public static Connection getConnection () throws ClassNotFoundException {
         System.out.println("Connection is "+conn);
      try {
                        if(conn == null || conn.isClosed()) {
                                    url= "jdbc:oracle:thin:@192.168.1.123:1521:SHTST";
                                    Class.forName("oracle.jdbc.driver.OracleDriver");
                                    conn= DriverManager.getConnection(url, "acctmed",
"sairam");
                                    System.out.println("Connection is "+conn);
//            url= "jdbc:mysql://localhost/project";
//            Class.forName("com.mysql.jdbc.Driver");
//            conn= DriverManager.getConnection(url,"root","sairam");
                                    //System.out.println("connection is "+conn);
                        }
                  } catch (SQLException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                  }
      return conn;
   }
}
```

**LoginServlet.java**

```
public LoginServlet extends HttpServlet{
      public LoginServlet()
      {
            super();
```

```
        }

        @SuppressWarnings("deprecation")
    protected void doPost(HttpServletRequest request,
      HttpServletResponse response) throws ServletException, IOException {
     PrintWriter out = response.getWriter();
     String user= request.getParameter("un");
     String passwd= request.getParameter("pw");
     System.out.println("userdetails "+user + passwd);

    Connection con= null;

     String reply= null;

                try {
                        con = Connector.getConnection();
                } catch (ClassNotFoundException e1) {
                        // TODO Auto-generated catch block
                        e1.printStackTrace();
                }
                if(con == null)
          {
                        System.out.println("im in null");
                        request.setAttribute("error","Check your Connection");
                        //RequestDispatcher rd=request.getRequestDispatcher("#");

                        response.sendRedirect("#/");
                        //rd.include(request, response);

            //response.sendRedirect("#/");
            // return false;
          }

            CallableStatement statement;

                if(con != null){
                 try {
                        statement = con.prepareCall("{call
REPORTS_MODULE.PR_AUTHENTICATE_USER(?,?,?,?)}");
                        statement.setString(1, user);
                statement.setString(2, passwd);
                         statement.registerOutParameter(3,Types.VARCHAR);
                         statement.registerOutParameter(4,OracleTypes.CURSOR);
                         statement.execute();

                 con.close();
                        } catch (SQLException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                        }
                        if (reply.equals("N"))
```

```
                                {
                                /*session.getTransaction().commit();
                                session.close();
                                sf.close();*/
                                // con.close();
                                        out.println("<font color=red>Either user name or
password is wrong.</font>");
                                        response.sendRedirect("#/");

                                }
                                else{

                                        ////////////////////////////
"+session.getAttribute("userid")+":"+session.getAttribute("pwid"));
                                //con.close();
                                        response.sendRedirect("#/main");
                                }
                        }
}
```

## 6.3 Database Tables

### 6.3.1 Consignment Receipt Header (for generating crhid for items)

| Data Item | Type | Description |
|---|---|---|
| CRH_ID | NUMBER | Unique id for consignment receipt |
| CRH_STM_ID | NUMBER | Id for the store master |
| CRH_VDM_ID | NUMBER | Id for the vendor master |
| CRH_RECEIVED_DATE | DATE | Date of the items received |
| CRH_MRR_NO | VARCHAR2(25) | Material Receipt Register number |
| CRH_DC_NO | VARCHAR2(25) | DC number for the items |
| CRH_DC_DATE | DATE | DC date |
| CRH_REMARKS | VARCHAR(500) | Remarks for the delivery |
| CRH_RECEIPT_TYPE | VARCHAR2(20) | Type of the receipt |
| CRH_STATUS | VARCHAR2(30) | Status of the items |
| CRH_ADDED_BY | VARCHAR2(30) | The one who added the items |
| CRH_ADDED_DATE | DATE | Date of the pcrn generation |
| CRH_AUTHORISED_BY | VARCHAR2(30) | One who authorizes the pcrn |
| CRH_AUTHORISED_DATE | DATE | Date of the authorization |
| CRH_USER_ID | VARCHAR2(30) | The logged in user |
| CRH_TERMINAL_NAME | VARCHAR2(30) | Name of the system in which user generates pcrn |
| CRH_DATE | DATE | Date of the crh |

*Table 1 Consignment Receipt Header*

### 6.3.2 Consignment Transfer (for saving the item transfers)

| Data Item | Type | Description |
|---|---|---|
| CNT_ID | NUMBER | Unique id for the transfer |
| CNT_FROM_STM_ID | NUMBER | Id of the store master(My Loc) |

| | | |
|---|---|---|
| CNT_TO_STM_ID | NUMBER | Id of the store master(To Loc) |
| CNT_VDM_ID | NUMBER | Id of the vendor |
| CNT_IDM_ID | NUMBER | Id of the item |
| CNT_BATCH_ISN_FLAG | VARCHAR2(1) | Id type |
| CNT_BATCH_ISN_NO | VARCHAR2(25) | Id no |
| CNT_QTY | NUMBER | Quantity being transferred |
| CNT_REMARKS | VARCHAR2(100) | Remarks for the transfer if any |
| CNT_STATUS | VARCHAR2(15) | Status of the transfer |
| CNT_ISSUED_BY | VARCHAR2(30) | Transfer issued by whom |
| CNT_ISSUED_DATE | DATE | Transfer issue date |
| CNT_RECEIVED_BY | VARCHAR2(30) | One who received the transfer |
| CNT_RECEIVED_DATE | DATE | Transfer received date |
| CNT_USER_ID | VARCHAR2(30) | Logged in user id |
| CNT_TERMINAL_NAME | VARCHAR2(30) | Name of the system in which user interacts |
| CNT_DATE | DATE | Transfer date |

*Table 2 Consignment Transfer*

## 6.3.3 Consignment Consumption (for saving the consumption of the items)

| Data Item | Type | Description |
|---|---|---|
| CCN_ID | NUMBER | Id of the consumption |
| CCN_STM_ID | NUMBER | Id of the store master |
| CCN_CONSUMPTION_DATE | DATE | Date of the consumption |
| CCN_PATIENT_ID | VARCHAR2(10) | Id of the patient |
| CCN_PATIENT_NAME | VARCHAR2(50) | Name of the patient |
| CCN_ADMISSION_NO | NUMBER | Patient's admission number |
| CCN_SURGERY | VARCHAR2(100) | Surgery name |
| CCN_SURGEON | VARCHAR2(50) | Surgeon name |
| CCN_IDM_ID | NUMBER | Id of the item |
| CCN_VDM_ID | NUMBER | Id of the vendor |
| CCN_BATCH_ISN_FLAG | VARCHAR2(1) | Id type |
| CCN_BATCH_ISN_NO | VARCHAR2(25) | Id number |
| CCN_CONS_UOM_ID | NUMBER | Consumption's UOM id |
| CCN_CONS_CATEGORY | VARCHAR2(10) | Consumption category |
| CCN_CONS_FACTOR | NUMBER | Factor of the consumption |
| CCN_CONS_QTY | NUMBER | Consumption quantity |
| CCN_FRESH_QTY | NUMBER | New quantity |
| CCN_VIL_ID | NUMBER | Id of the vil |
| CCN_POL_ID | NUMBER | Id of the pol |
| CCN_NET_RATE | NUMBER | Rate |
| CCN_STOCK_TYPE | VARCHAR2(1) | Type of the stock in consumption |
| CCN_STATUS | VARCHAR2(15) | Status of the consumption |
| CCN_ADDED_BY | VARCHAR2(30) | One who added the consumption |
| CCN_ADDED_DATE | DATE | Date of the consumption |
| CCN_VERIFIED_BY | VARCHAR2(30) | One who verified |
| CCN_VERIFIED_DATE | DATE | Date of the verification |
| CCN_ACCEPTED_BY | VARCHAR2(30) | One who accepted the consumption |
| CCN_ACCEPTED_DATE | DATE | Acceptance Date |
| CCN_USER_ID | VARCHAR2(30) | Logged in user Id |
| CCN_TERMINAL_NAME | VARCHAR2(30) | System name used for consumption updation |
| CCN_DATE | DATE | Consumption date |

| | | |
|---|---|---|
| CCN_CONSOLIDATED_BY | VARCHAR2(30) | One who consolidates the items |
| CCN_CONSOLIDATED_DATE | DATE | Date of consolidation |

*Table 3 Consignment Consumption*

## 6.3.4 Consignment Receipt Line

| Data Item | Type | Description |
|---|---|---|
| CRL_ID | NUMBER | Unique Id of the consignment receipt line |
| CRL_CRH_ID | NUMBER | Id of the_CRH |
| CRL_IDM_ID | NUMBER | Id of the item |
| CRL_ITEM_DESCRIPTION | VARCHAR2(125) | Item name |
| CRL_UOM_ID | NUMBER | Id of the unit of measure |
| CRL_BATCH_ISN_FLAG | VARCHAR2(1) | Id type |
| CRL_BATCH_ISN_NO | VARCHAR2(25) | Id number |
| CRL_EXPIRY_DATE | DATE | Expiry date of the item |
| CRL_RECEIVED_QTY | NUMBER | Received qty |
| CRL_RATE | NUMBER | Rate of the item |
| CRL_VIL_ID | NUMBER | Id of the vil |
| CRL_REMARKS | VARCHAR2(100) | Remarks if any |
| CRL_STATUS | VARCHAR2(15) | Status of the item received |
| CRL_ADDED_BY | VARCHAR2(30) | One who added pcrn |
| CRL_ADDED_DATE | DATE | Date of the pcrn generation |
| CRL_AUTHORISED_BY | VARCHAR2(30) | One who authorizes pcrn |
| CRL_AUTHORISED_DATE | DATE | Date of the authorizes |
| CRL_USER_ID | VARCHAR2(30) | Id of the logged in user |
| CRL_TERMINAL_NAME | VARCHAR2(30) | System name in which it is added |
| CRL_DATE | DATE | Date of the pcrn generation |

*Table 4 Consignment Receipt Line*

## 6.3.5 Consignment Stock

| Data Item | Type | Description |
|---|---|---|
| CNS_ID | NUMBER | Unique id for each transaction of the stock |
| CNS_STM_ID | NUMBER | Id of the store master |
| CNS_VDM_ID | NUMBER | Id of the vendor |
| CNS_IDM_ID | NUMBER | Id of the item |
| CNS_BATCH_ISN_FLAG | VARCHAR2(1) | DC type |
| CNS_BATCH_ISN_NO | VARCHAR2(25) | Dc number |

| | | |
|---|---|---|
| CNS_EXPIRY_DATE | DATE | Date of expiry of the item |
| CNS_STOCK_QTY | NUMBER | Current stock of the item |
| CNS_STATUS | VARCHAR2(1) | Status of the stock for each item |
| CNS_ADDED_DATE | DATE | Transaction added date |
| CNS_ADDED_BY | VARCHAR2(30) | One who made the transaction |
| CNS_USER_ID | VARCHAR2(30) | Id of the logged in user |
| CNS_TERMINAL_NAME | VARCHAR2(30) | System name in which transaction has made |
| CNS_DATE | DATE | Date of the transaction |

*Table 5 Consignment Stock*

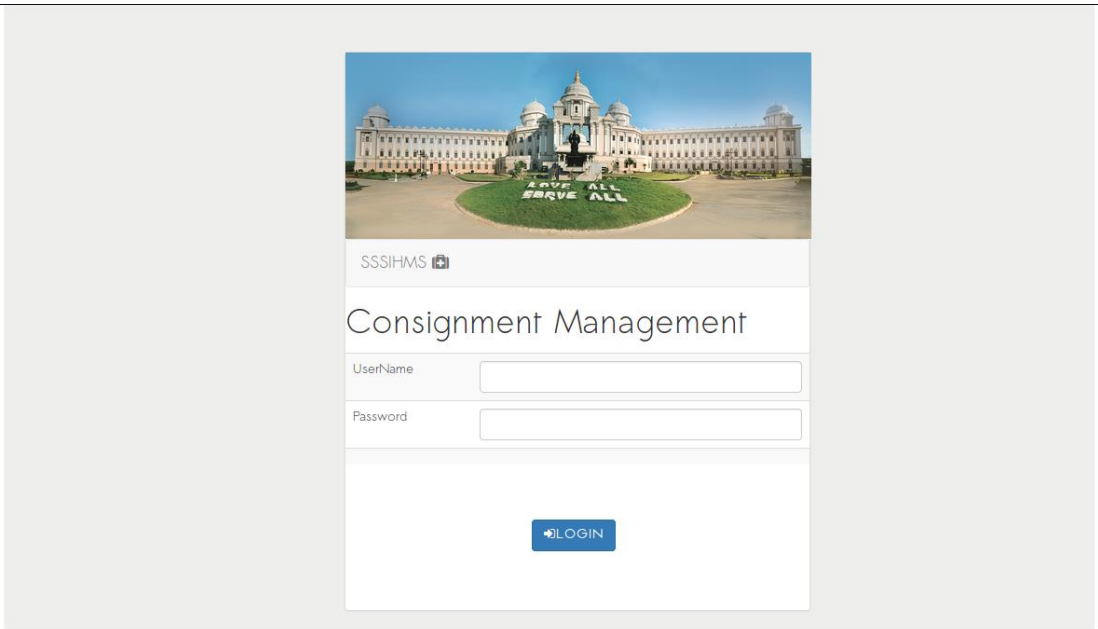
# 7 TESTING AND IMPLEMENTATION

The application is tested in the live environment by connecting with live database in HMIS in the hospital. It has been run on the Windows Operating System and has positive result and all features are working as expected.

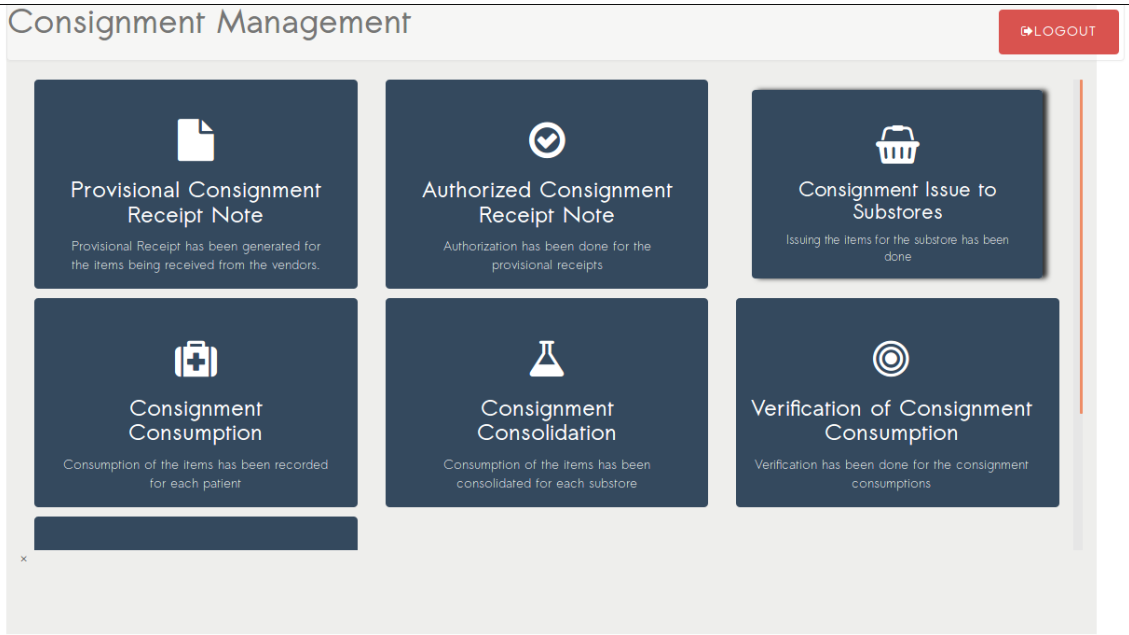
The differences being found has been corrected and all the cases are tested for the desired output in the HMIS in SSSIHMS. The differences related to browser issues which are simple to correct

# 8 SCREENSHOTS OF THE APPLICATION

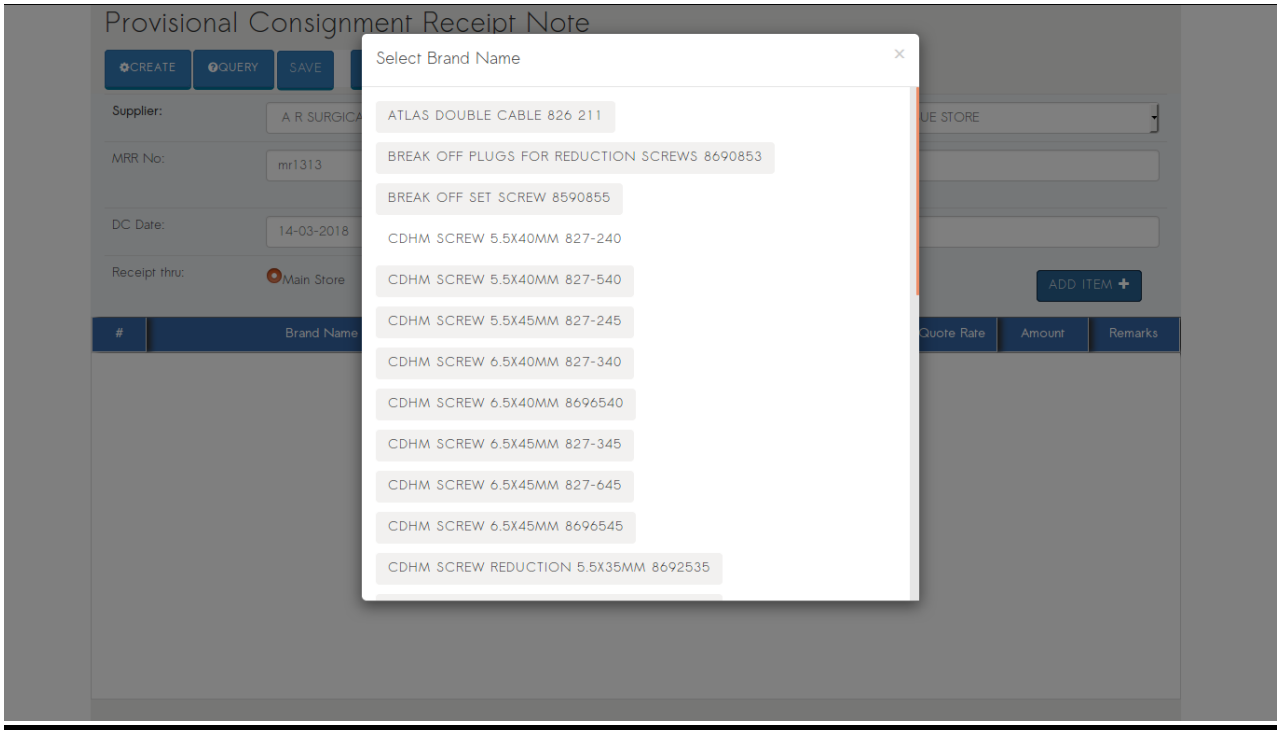
## 8.1 Login Page



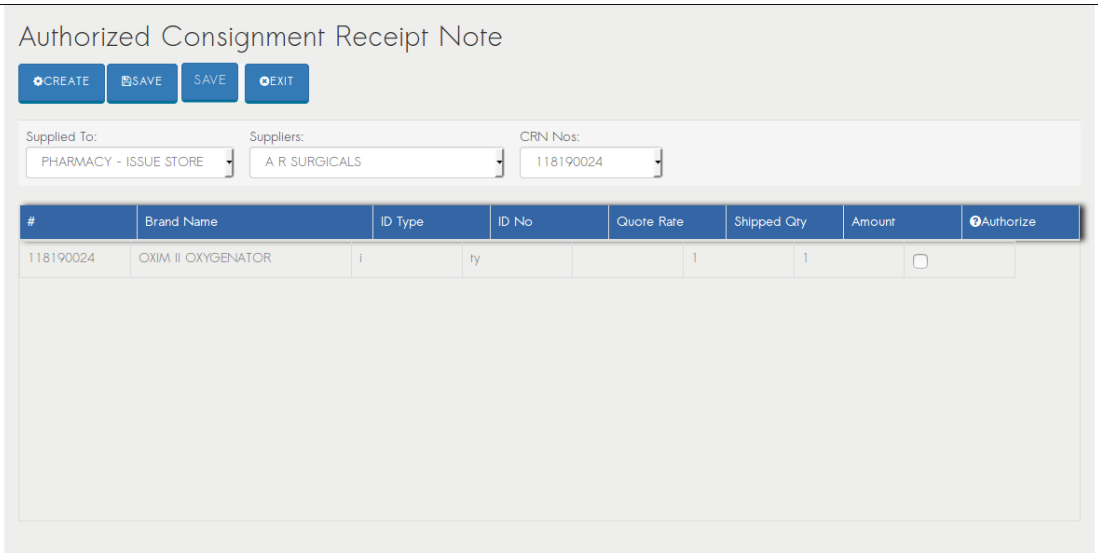

## 8.2 Home Page



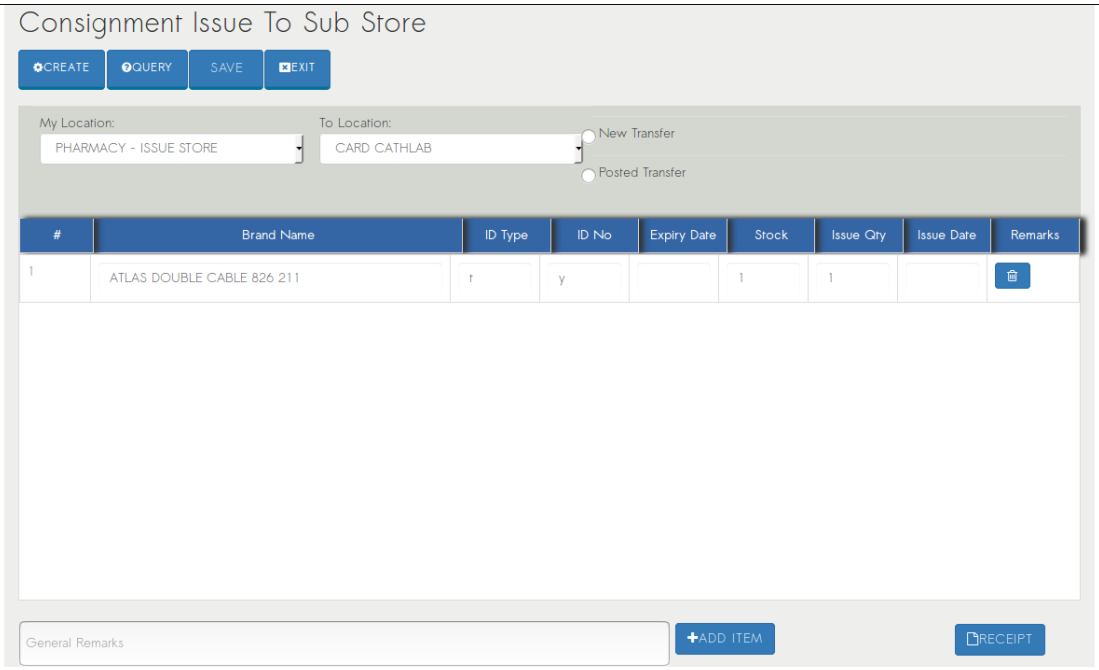
## 8.3 Provisional Consignment Receipt Note



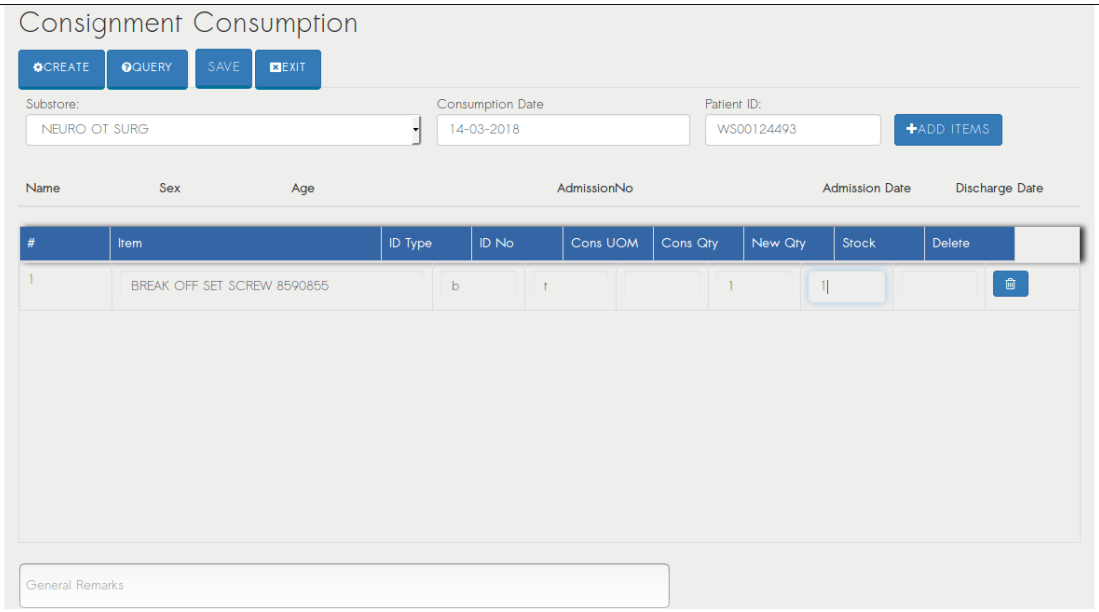

## 8.4 Authorization of Consignment Receipt Note



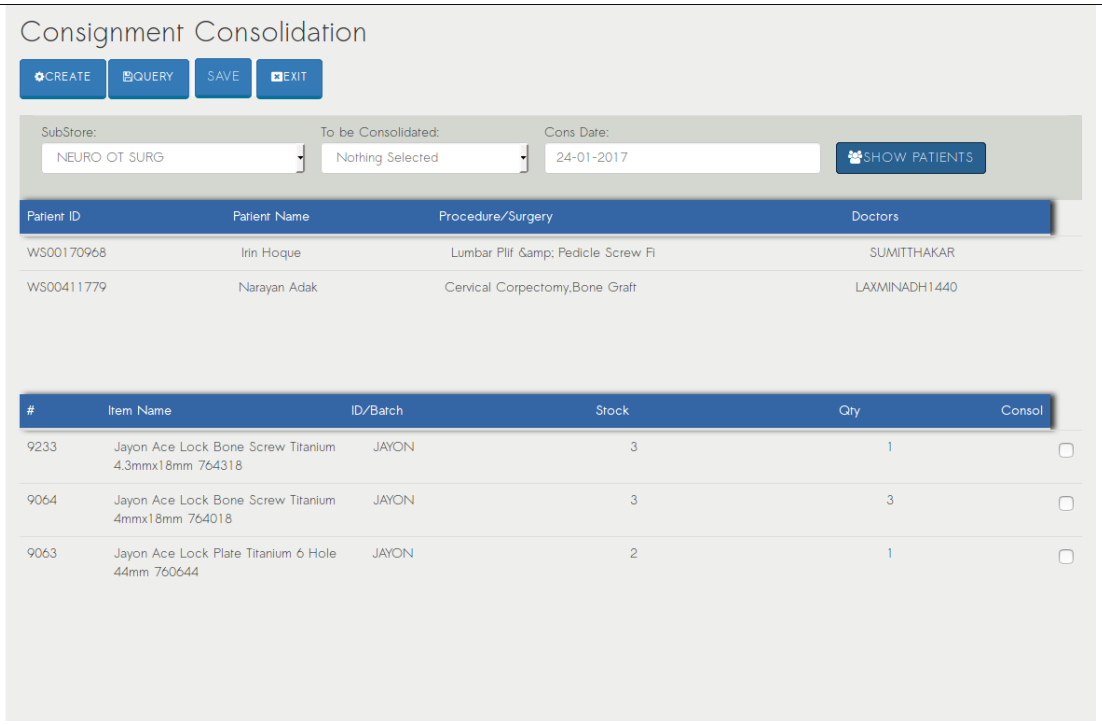
## 8.5 Consignment Issues to Sub Store



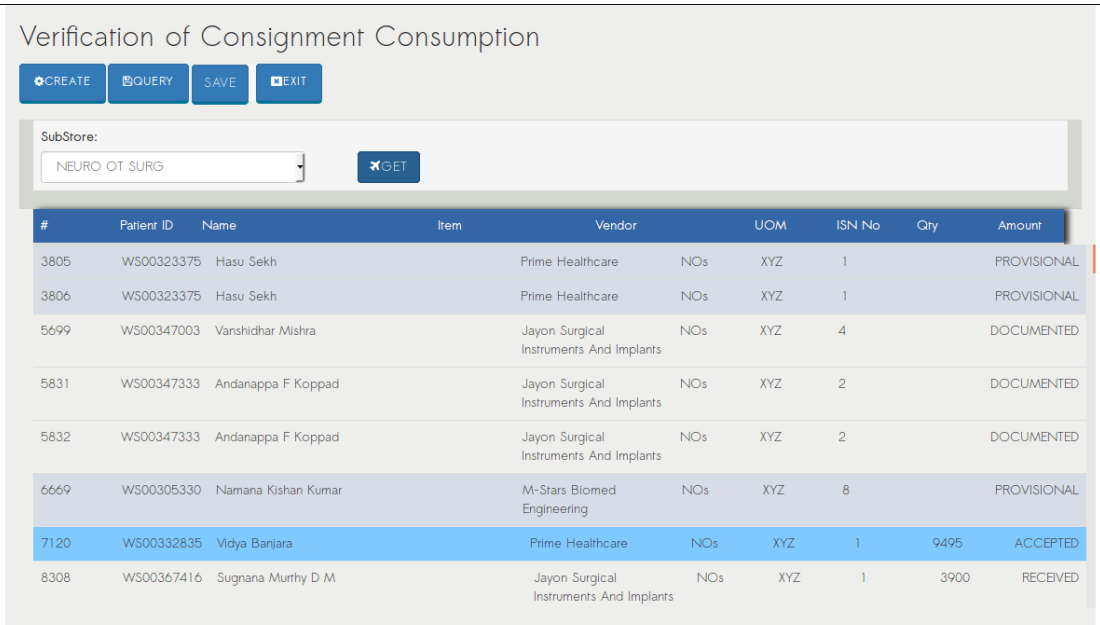

## 8.6 Consignment Consumption



## *8.7 Consignment Consolidation*




## 8.8 Verification of Consignment Consumptions



## 8.9 Acceptance of Consignment Consumptions


## 8.11 Future Scope

- The application can be built using spring framework and Hibernate technologies.
- The front end can be made much more attractive using Angular Material and Material Design Bootstrap.
- REST API can used in the application for communication purpose.

# 9 REFERENCES


1. www.w3schools.com
2. www.stackoverflow.com
3. www.tutorialspoint.com
4. www.javapoint.com
5. www.dineshonjava.com
6. www.angularjs.org
7. www.github.com
8. InventoryOps.com
9. Bootstrap.com
10. www.codepenio.com
11. www.datatables.com