

# FLEXIHOUSE

BS Computer Sciences, Batch 2013

Project Supervisor

**Dr. Husnain Mansoor Ali**

Associate Professor  
BS Computer Sciences

Submitted by:

Talha Amin	1312191
Amirul Sunesara	1312169



Department of Computer Science  
Shaheed Zulfikar Ali Bhutto Institute of Science & Technology Karachi  
May 2017

# 1 Project Detail

Type (Nature of project)	<input checked="" type="checkbox"/> <b>Software</b> Based <input type="checkbox"/> <b>Hardware</b> Based			
Area of specialization	Optimization			
Project Group Members				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	1312191	Talha Amin	talhaamin94@gmail.com	
(ii)	1312169	Amirul Sunesara	amirulsunesara@gmail.com	

\*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

## 2 Plagiarism Free Certificate

This is to certify that, I am Talha Amin S/D/o Amin Mustafa, group leader of FYP under registration no 1312191 at Computer Science Department, SZABIST. Karachi. I declare that my FYP proposal is checked by my supervisor.

Date: 02/11/16 Name of Group Leader: Talha Amin

Signature: \_\_\_\_\_

Name of Supervisor: Dr. Husnain Mansoor Ali Designation: Program Manager

Signature: \_\_\_\_\_

Co-Supervisor (if any): \_\_\_\_\_ Designation: \_\_\_\_\_

Signature: \_\_\_\_\_

## **Acknowledgement**

We would like to express our deepest appreciation to Dr. Husnain Mansoor Ali who supervised the development of this project. His suggestions and encouragement were indispensable elements in the course of developing this project.

We would also like to express gratitude to the faculty who over the 4 years have ingrained in us the skills and ability to develop this project and for future endeavours we may undertake.

Thank you,

Talha Amin

Amirul Sunesara

## Document Version History

Version ID	Changes Made	Date
<b>v1</b>	Requirements added	2-10-2016
<b>v2</b>	Design for semester 7 milestones added	9-10-2016
<b>v3</b>	Testing added, made changes to data dictionary, screenshots added, Code added in glossary	8-1-2017
<b>v3.1</b>	Traceability Matrix added	8-1-2017
<b>V4</b>	2 Use cases added, Changes made to Use Case Diagram, DFD, ERD and Data Dictionary. New Sequence Diagram added and new test case added. Screen shots updated	10-4-2017
<b>V4.1</b>	Data Dictionary Updated, Screen shots updated, use case updated, test cases added	15-05-2017
<b>V4.2</b>	Added item details table to data dictionary. Completed test cases for all use cases.	29-05-2017

## Project Task Distribution

Project Task	Member Responsible
Model Designer	Amirul Sunesara
Model Routine	Talha Amin
User Module Design	Amirul Sunesara
User Module Routines and Validation	Talha Amin
Slotting Design	Amirul Sunesara
Slotting Routine	Talha Amin
Detailed Shelf View	Amirul Sunesara
Dispatcher Routine	Talha Amin

## Table of Contents

1	Project Detail.....	1
2	Plagiarism Free Certificate .....	3
	Document Version History .....	5
	Project Task Distribution.....	6
1	Introduction .....	1
1.1	Problem Statement .....	1
1.2	Objectives .....	1
1.3	Assumptions & constraints.....	2
1.4	Project Feasibility Analysis .....	3
1.4.1	Economic Feasibility .....	3
1.4.2	Technical Feasibility .....	3
1.4.3	Operational Feasibility .....	3
1.4.4	Schedule Feasibility .....	4
1.4.5	Conclusion of Feasibility Analysis .....	4
1.5	Project scope .....	4
1.5.1	Features Included .....	4
1.5.2	Scope Limitations .....	4
2	Requirements Analysis.....	5
2.1	Existing System Study .....	5
2.2	Stakeholders list.....	5
2.3	Requirements.....	5
2.3.1	Functional Requirements.....	5

2.4	Non- Functional Requirements .....	6
2.4.1	Interface requirements .....	6
2.4.2	Availability Requirements .....	7
2.4.3	Security Requirements .....	7
2.4.4	Performance Requirements .....	7
2.4.5	User Documentation .....	7
2.5	Use case Design .....	8
2.6	Use case Descriptions .....	9
2.7	Software Development Life Cycle Model .....	35
3	System Design and Implementation .....	36
3.1	System Architecture .....	36
3.1.1	User Module .....	39
3.1.2	Modeller Module .....	39
3.1.3	Slotting Module .....	39
3.1.4	Dispatcher Module .....	39
3.2	Data Flow Diagram .....	40
3.3	Database Design .....	41
3.3.1	Database Diagram .....	41
3.3.2	Data Dictionary .....	42
3.3.3	Database Normalization .....	51
3.4	Sequence Diagrams .....	52
3.4.1	Signup Sequence .....	52
3.4.2	Login Sequence .....	53
3.4.3	Modeller Sequence .....	54



3.4.4	Slotting Sequence.....	55
3.4.5	View Information Sequence.....	56
3.4.6	Place Order Sequence .....	57
3.4.7	Dispatch Items Sequence.....	58
3.5	Requirements Traceability Matrix .....	59
4	Testing.....	60
4.1	Scope of Testing .....	60
4.2	Test Plan Strategy .....	60
4.2.1	Unit Testing.....	60
4.2.2	Integration Testing .....	61
4.2.3	System Testing .....	61
4.3	Test Environment.....	62
4.4	Schedule .....	62
4.5	Test Case Design and Description .....	63
4.6	Status Reporting .....	72
4.7	Risks and Assumptions.....	72
4.8	Major Deliverables .....	72
4.9	Exit Criteria.....	72
5	Conclusion .....	73
5.1	Lessons Learnt.....	73
5.2	Future Enhancements.....	73
5.2.1	Allow for more pallet sizes .....	73
5.2.2	Improve Warehouse Modelling .....	73
5.2.3	Special items storage Conditions/ Designated area slotting .....	73

6	Appendix .....	74
6.1	Glossary .....	74
6.2	Screenshots .....	75

## List of Tables

Table 1: User Sign up use case .....	9
Table 2: User Sign-in use case .....	11
Table 3: Warehouse Modeller Use Case .....	12
Table 4: Warehouse view model Use Case .....	14
Table 5: Edit Warehouse Model Use Case .....	17
Table 6: Add items use case .....	19
Table 7: View All Items Use Case .....	21
Table 8: Order items Use Case .....	23
Table 9: Add Worker use case .....	25
Table 10: View Workers List use case .....	27
Table 11: Detailed shelf view use case .....	29
Table 12: Receive Notifications use case .....	31
Table 13: Issue Order for Dispatch Use Case .....	33
Table 14: User Table .....	42
Table 15: Warehouse Table .....	43
Table 16: Shelf Table .....	44
Table 17: Item Table .....	45
Table 18: Item Details Table .....	46
Table 19: Consignment Table .....	47
Table 20: item_consignment Table .....	48
Table 21: Order Table .....	49
Table 22: Order Items Table .....	50
Table 23: Workers Table .....	51
Table 24: Requirement Traceability Matrix .....	59
Table 25 Signup Test Case .....	63
Table 26: Login Test Case .....	63
Table 27: Design Warehouse Test Case .....	65

Table 28: View Warehouse Test Case .....	65
Table 29: Add Items Test Case .....	66
Table 30: View Items Test Case.....	67
Table 31: Place Order Test Case .....	67
Table 32: Add Worker Test Case .....	68
Table 33: Order Items Test Case .....	69
Table 34: View workers test case .....	70
Table 35: Detailed Shelf View test case .....	70
Table 36: Receive notifications test case.....	71
Table 37: Issue order for dispatch test case .....	71
Table 38: Key Terms .....	74

## List of Figures

Figure 1: Pallet Dimensions.....	2
Figure 2: Pallet Dimensions with alternate views.....	3
Figure 3: Use Case Diagram.....	8
Figure 4 Flexihouse Context Diagram .....	36
Figure 5: ASP.NET MVC Request Pipeline .....	37
Figure 6: Product Breakdown Structure.....	38
Figure 7: Data Flow Diagram.....	40
Figure 8: FlexiHouse ERD.....	41
Figure 9: Flexihouse Signup Sequence Diagram.....	52
Figure 10: Flexihouse Login Sequence Diagram .....	53
Figure 11: Flexihouse Modeller Sequence Diagram .....	54
Figure 12: Flexihouse Slotting Sequence Diagram.....	55
Figure 13: Flexihouse View Information Sequence Diagram .....	56
Figure 14: Place Order Sequence Diagram.....	57
Figure 15: Dispatch Items Sequence Diagram .....	58
Figure 16: SS1 - Landing Page .....	75
Figure 17: SS2 - Warehouse Design .....	76
Figure 18: SS3 – Warehouse Dimensions.....	76
Figure 19: SS4 – Detailed Shelf View.....	77
Figure 20: SS5 - View Warehouse .....	77
Figure 21: SS5 – Add Consignment .....	78
Figure 22: SS6 – View Consignments .....	78
Figure 23: SS7 – View Items.....	79
Figure 24: SS8 – Create Worker Account.....	79
Figure 25: SS9 – View Registered Workers .....	80
Figure 26: SS10 – View Orders.....	80
Figure 27: SS11 – Order Instructions .....	81

Figure 28: SS12 – Order step 1 .....	81
Figure 29: SS13 – Order step 2.....	82
Figure 30: SS14 – Order step 3.....	82
Figure 31: SS15 – Edit Profile .....	83
Figure 32: SS16 – Worker - New Consignments .....	83
Figure 33: SS17 – Consignment Instruction Page .....	84
Figure 34: SS18 – New Orders .....	84

# 1 Introduction

Warehouses have been around for centuries, the first remnants of a warehouse in history are found in the Roman Era (123 BC). However, there was a large increase in the number and types of warehouses after the industrial revolution. Now warehouses have become an integral part of manufacturing and retail industry. Since the growth of e-commerce, warehouses have become more and more important as stores are now online and they require their items to be stored in a warehouse until they are needed. Because of all this, the efficiency of a warehouse in storage and dispatch have become increasingly important.

## 1.1 Problem Statement

Storage and dispatch of warehouse items is an essential process in the working of a warehouse which needs to be managed as best as possible. It is very important for warehouse owners to store as much as possible in the limited space that they have and keep a record of where all the items are stored. Also when the time comes for items to be dispatched, the order has to be divided among the various workers assigned to the respective zones as it is not possible for one person to carry all the items in the warehouse due to time constraints.

## 1.2 Objectives

The goal of this system is to improve the efficiency and workflow of a warehouse as well as provide warehouse owners to manage their warehouse without the need for expensive tailor made solutions.

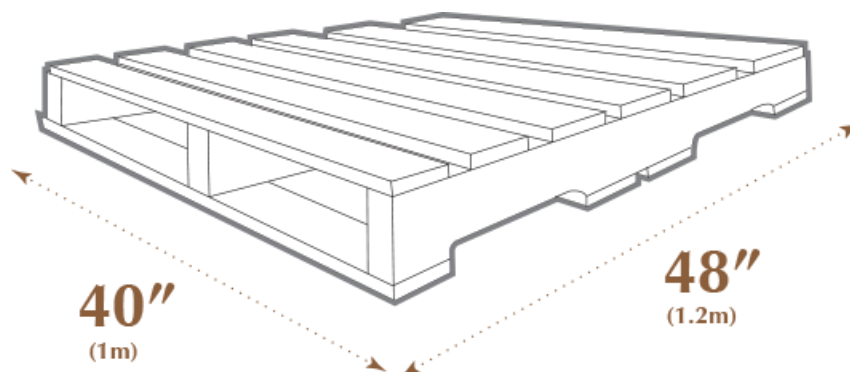
The system will achieve this goal by:

- Providing a means to the user to deliver us with the structure and layout of their warehouse
- Allowing the user to maintain detailed information regarding where the items are stored and what items are stored
- Providing an efficient solution to slotting in terms of time taken as well as shelf space optimization
- Providing a means for the warehouse to receive customer orders
- Creating an efficient dispatch policy to reduce time take for dispatch

### 1.3 Assumptions & constraints

The assumptions that the project team will make and base their design upon:

- All systems and database components being developed are unique to this system.
- The web application is simple to use and a user will have prior knowledge of how websites are generally laid out and used.
- All users must have access to an internet capable device.
- All users use a modern web browser. Google Chrome is the recommended choice.
- Users will require access to the system 24 hours a day.
- Users may or may not have technical knowledge and the final solution should satisfy both varieties.
- All users have an idea of the workings of a warehouse.
- The warehouse user is familiar with the dimensions and sizes of all the items placed in their warehouse.
- The warehouse is a company operated warehouse and not an independent warehouse where spaces are bought and sold.
- The warehouse pallet sizes will not be variable, only the ISO standard (40"x48") will be used (diagram show below)
- The order picking forklift of the warehouse is a standard order picker capable of lifting upto 3000 lbs. (maximum of 24 pallets) assuming each pallet weighs 125 lbs.



*Figure 1: Pallet Dimensions*



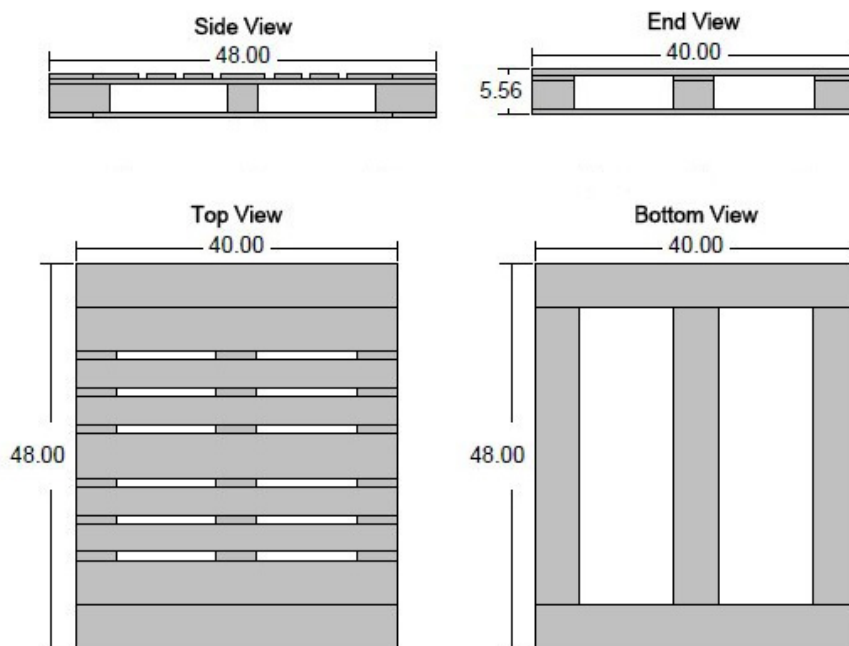


Figure 2: Pallet Dimensions with alternate views

## 1.4 Project Feasibility Analysis

### 1.4.1 Economic Feasibility

Not applicable since the developers are not being paid.

### 1.4.2 Technical Feasibility

We have several options for the technical implementation of the project. Since it is a web application we can use ASP.NET MVC, MEAN stack or Java Spring. The team members had experience with all these technologies.

After a thorough analysis online and much deliberation with the development team we recommend to use ASP.NET MVC for the backend development and SQL Server database based on the team's experience with the technologies and market preference.

### 1.4.3 Operational Feasibility

Since most of the work in local warehouses is done manually, this application will be a huge help to these prospective clients as they will not only have a system to store all their warehouse records on but it will also optimize the workings of their warehouse. We want to make this application available to as many people as we can and this is why we decided to make the system a web application.

#### **1.4.4 Schedule Feasibility**

The allotted time for this project is more than sufficient for the developers to successfully engineer and deliver the system. The time to complete the project was estimated based on experience.

#### **1.4.5 Conclusion of Feasibility Analysis**

Overall with the time restrictions and the technical expertise of the team we can say that the project is feasible and can be implemented successfully, there are no requirements in the scope that cannot be achieved using the recommended technologies.

### **1.5 Project scope**

#### **1.5.1 Features Included**

The solution being proposed is a web application. The warehouse user after going through the signup operation and gaining proper authorization will be able to manage their warehouse operations using this web application. They will first create a model/floor plan of their warehouse which will give the system various information regarding the warehouse and through that information, it will decide how best to optimize the workings of the warehouse. The system will also help the warehouse manager/owner by managing their consignments. When a consignment will arrive, the system will decide where to place the items which will be most efficient in terms of access time as well as shelf space.

When the warehouse will receive an order from the customer, it will also dispatch the items in the most efficient manner possible in terms of speed. The system will provide some other features such as notifications and item tracking.

#### **1.5.2 Scope Limitations**

There are several aspects that will not be covered by this solution:

- In the shelves will be placed only in 2 orientations, vertical or horizontal
- The system will not put a limitation on aisle space, it can be as small or as large as possible.
- The shelf size for a particular warehouse will be fixed for a particular warehouse, as this is the practice in real life warehouses.
- There will be only one depot (I/O point) in the warehouse, our solution will not handle multiple depots.
- Our solution will not provide the feature of order tracking for the customer.
- We are not handling special item storage conditions like refrigeration.

## 2 Requirements Analysis

### 2.1 Existing System Study

Although there are many systems that allow warehouse owners to automatically manage their warehouse, the problem is that these solutions are tailor-made for a particular warehouse and doesn't provide a generalized solution. Also, these systems are highly expensive and not affordable by small scale warehouses. The systems also don't have any utility for the warehouse owner to design a model of their warehouse, so any design changes in the warehouse will have to be communicated to the solution providers manually.

Our solution will allow warehouse owners to control and manage their warehouse in an optimized way without the need for expensive tailor made solutions.

### 2.2 Stakeholders list

Following is the stakeholders list for this system:

- Warehouse Owner/ Manager
- Warehouse Workers
- Customers
- Companies that supply the warehouse with the items

### 2.3 Requirements

#### 2.3.1 Functional Requirements

The list includes the set of functional requirements that the system will satisfy. All functional requirements will be prefixed with "FS" followed by the number in the list when used elsewhere in this document. E.g. **FS-10**

##### 2.3.1.1 Information and Authorization Requirements

1. Warehouse Manager should be able to register for the service
2. Warehouse Manager and Worker should be able to login
3. Customer should be able to order items from the warehouse
4. Warehouse Manager can keep a record of all the warehouse customers
5. Warehouse Manager can view all his workers
6. Warehouse Manager can add new workers to their warehouse

### **2.3.1.2 Modeller Requirements**

7. The application will have a utility for designing the warehouse, it will contain all the things that are generally found in a floor plan of a warehouse, e.g. shelves, offices etc.
8. Be able to edit the layout of the warehouse
9. View the model at any time
10. Adjust scale of the model according to user requirement

### **2.3.1.3 Slotter Requirements**

11. The application will break down the warehouse into zones based on an algorithm
12. The items will be slotted into their prescribed zones
13. Be able to add items to the warehouse
14. View a detailed list of all the items stored in the warehouse.
15. View a list of all the items in the warehouse shelf-wise
16. Receive notifications when a warehouse zone is running out of space
17. Receive notifications when an item is about to run out of the warehouse
18. Receive notifications when an item is about to expire

### **2.3.1.4 Dispatcher Requirements**

19. Get item list for dispatch
20. Select what items need to be dispatched
21. Dispatch items from the warehouse according to order
22. Receive notification when a new order is received

## **2.4 Non- Functional Requirements**

### **2.4.1 Interface requirements**

The solution will offer an easy to use and highly intuitive. Graphical user interface, following standard HCI (Human computer interaction) principles. As the target user base people very large and will contain people from various different demographics the GUI will need to effective and attractive to many different types of users. Google's Material design language will be used as it will provide users with experience with Google products additional confidence navigation the web application.

#### **2.4.2 Availability Requirements**

The solution is being designed as a web application and must be available to users 24/7 throughout the year, to achieve this, the system will be designed in such a way as to minimize downtime during maintenance and or system crashes.

#### **2.4.3 Security Requirements**

The system is designed as such that no user can access it without proper authorization, for the user to access the system they must sign up and they will be given the proper view of the website according to their role and degree of authorization. The database management system will handle the data security and integrity issues automatically.

#### **2.4.4 Performance Requirements**

The performance requirements of the system are not high since you only need a browser to run it but the internet connection must be stable and of good speed at least 1Mbps or the system will feel slow. The system will keep the database up to date as all the changes in data will be reflected on the database after each transaction or update so concurrency is not an issue. We need to make sure that one data transaction is completed in 10 seconds, so 1Mbps connection is sufficient, any better speed might result in an improved response time.

#### **2.4.5 User Documentation**

The documents that will be delivered along with the software are the software requirements specification document, software design document, test case document and the code listing.

## 2.5 Use case Design

The Use Case Diagram show the actors involved in the system and the task that are performed by these action in the system

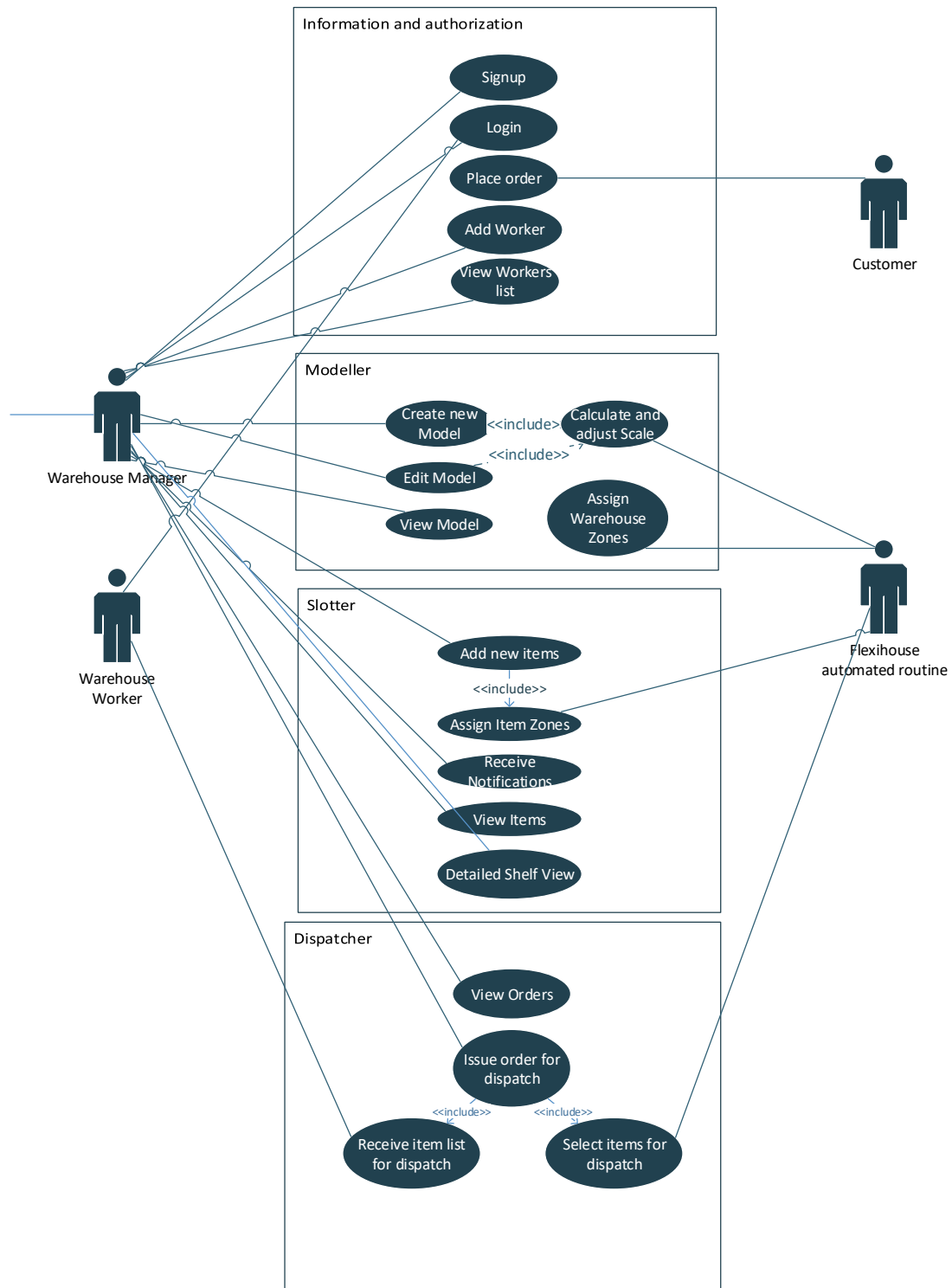


Figure 3: Use Case Diagram

## 2.6 Use case Descriptions

The following tables define the use cases in complete detail.

*Table 1: User Sign up use case*

Use-Case Name:	User Sign up	Use-Case Type
		Business Requirements: Yes
Use-Case ID:	FI-U01	
Priority:	Medium	
Source:	FS-1	
Primary Actor(s):	Warehouse Manager	
Secondary Actor(s):	N/A	
Description:	This use case allows the actors to sign up to use the services provided by the application.	
Pre-Condition:	To successful registration the user needs to fill the form and select a specific role.	
Trigger:	This use case is initiated when a user registers on to the system	
Typical Course of Events:	ACTOR ACTION	SYSTEM RESPONSE

	<p><b>Step 1 – The actor clicks on the register button on the lading page.</b></p> <p><b>Step 2 – The website opens the registration page which asks actor to input his general details such as name, address, phone etc. It also asks the user to select their role</b></p> <p><b>Step 3 – The actor enters his complete details and submits the form</b></p> <p><b>Step 4 – The form is validated and verified where necessary.</b></p> <p><b>Step 5 – The actor gets registered to the website</b></p> <p><b>Step 6 – System takes actor to their dashboard.</b></p>
<b>Alternate Courses:</b>	<p>Step 4:</p> <p><b>Alt-1:</b> The actor will be asked to complete/re-enter any invalid field such as duplicate email.</p>
<b>Conclusion:</b>	This use case concludes when the actor register the account.
<b>Post condition:</b>	The account information is stored into the database and the dashboard is displayed to the actor.
<b>Business Rules:</b>	User must have a valid email information.
<b>Implementation Constraints and Specifications:</b>	Use case must be available to every new user.
<b>Assumptions:</b>	The user can cancel the registration at any time by closing the page, hitting the browser back button or the device stopped working abruptly.



Table 2: User Sign-in use case

Use-Case Name:	User Sign in	Use-Case Type
Use-Case ID:	FI-U02	Business Requirements: Yes
Priority:	Medium	
Source:	FS-2	
Primary Actor(s):	Warehouse Worker, Warehouse Manager	
Secondary Actor(s):	N/A	
Description:	This use case allows the user to sign in to the application to use the services	
Pre-Condition:	To successful sign in the user needs to fill the form with his authentic credentials.	
Trigger:	This use case is initiated when a user sign in to the system	
Typical Course of Events:	ACTOR ACTION	SYSTEM RESPONSE
	Step 1 – The user clicks on the login button on the landing page.	Step 2 – The website opens the sign in page which asks user to input his credentials
	Step 3 – The user enters his credentials and clicks sign in button	Step 4 – The credentials are authenticated from the database

	<b>Step 5</b> – System takes user to their appropriate dashboard.
<b>Alternate Courses:</b>	<p>Step 4:</p> <p><b>Alt-1:</b> If the user is not authenticated the system will highlight error, and request the information to be re-entered.</p>
<b>Conclusion:</b>	This use case concludes when the user is authenticated from the database.
<b>Post condition:</b>	The user is redirected to the dashboard.
<b>Business Rules:</b>	User must have a valid credentials.
<b>Implementation Constraints and Specifications:</b>	Use case must be available to every registered user.
<b>Assumptions:</b>	The user can cancel the login at any time by closing the page, hitting the browser back button or the device stopped working abruptly.

Table 3: Warehouse Modeller Use Case

Use-Case Name:	New Warehouse Model	Use-Case Type
----------------	---------------------	---------------

Use-Case ID:	FI-U03	Business Requirements: Yes
Priority:	High	
Source:	FS-7,8,10	
Primary Actor(s):	Warehouse Manager	
Secondary Actor(s):	N/A	
Description:	This use case allows the user to create a model of their warehouse	
Pre-Condition:	The user should be logged into the application as a warehouse manager	
Trigger:	This use case is when a user first signs in as a warehouse manager	
Typical Course of Events:	ACTOR ACTION	SYSTEM RESPONSE
	Step 1 – The user clicks Create Warehouse Layout option	Step 2 – The system redirects the user to the warehouse modeller page and asks them to enter the dimensions of their warehouse
	Step 3 – The user enters the dimensions and clicks ok	
		Step 4 – The system then calculates a scale for the model and presents the toolbox to the user for all the necessary modelling.
	Step 5 – The User designs the layout of their warehouse and clicks save	

	<p><b>Step 6</b> – The system saves all the data regarding the warehouse along with the model.</p> <p><b>Step 7</b> – The system runs an algorithm and creates zones in the warehouse and saves the data in the database</p> <p><b>Step 8</b> – The system presents a save successful option to the user</p>
<b>Alternate Courses:</b>	None
<b>Conclusion:</b>	This use case concludes when the user successfully saves the model of the warehouse.
<b>Post condition:</b>	All the data from the modeller and the zones is saved in the database
<b>Business Rules:</b>	User can only use the tools available in the tool box for creating the model
<b>Implementation Constraints and Specifications:</b>	Use case must be available to every registered Warehouse Manager.
<b>Assumptions:</b>	The user can cancel the creation at any time by closing the page, hitting the browser back button or the device stopped working abruptly.

Table 4: Warehouse view model Use Case

Use-Case Name:	View Warehouse Model	Use-Case Type
		Business Requirements: Yes
Use-Case ID:	FI-U04	

Priority:	Medium	
Source:	FS – 9, 14	
Primary Actor(s):	Warehouse Manager	
Description:	This use case allows the user to view the model of their warehouse and view the items shelf wise	
Pre-Condition:	The user must be logged into the system and must have created a warehouse model	
Trigger:	This use case is initiated when a user selects the view model option	
Typical Course of Events:	ACTOR ACTION	SYSTEM RESPONSE
	Step 1 – The user clicks on the view model option	Step 2 – System outputs the previously created model of the warehouse
	Step 3 – The user clicks on the shelf	Step 4 – System outputs the list of items for that shelf
Alternate Courses:		
Conclusion:	This use case concludes when the user selects some other option	
Post condition:		
Business Rules:	The view model should be available at all times once the model is created	

<b>Implementation</b>	Use case must be available to every registered user.
<b>Constraints and Specifications:</b>	
<b>Assumptions:</b>	

Table 5: Edit Warehouse Model Use Case

Use-Case Name:	Edit Warehouse Model	Use-Case Type
		Business Requirements: Yes
Use-Case ID:	FI-U05	
Priority:	Medium	
Source:	FS- 8	
Primary Actor(s):	Warehouse Manager	
Secondary Actor(s):	N/A	
Description:	This use case allows the user to edit a previously created model of the warehouse.	
Pre-Condition:	The ware house model must have been created	
Trigger:	This use case is initiated when a user presses the edit warehouse button	
Typical Course of Events:	ACTOR ACTION	
	SYSTEM RESPONSE	
	Step 1 – The user clicks on the edit warehouse button	
	Step 2 – The system loads the previously created model from the warehouse and displays it and calls the modeller functionality.	
Alternate Courses:	Step 2 – If warehouse model is not created an error message will be displayed	
Conclusion:	This use case concludes when the user presses the save button	

<b>Post condition:</b>	The system stores the new layout in the database
<b>Business Rules:</b>	The use case should be available at all times once the model is created
<b>Implementation Constraints and Specifications:</b>	Use case must be available to every registered Warehouse Manager.
<b>Assumptions:</b>	There is an existing layout of the warehouse
<b>Open Issues:</b>	None



Table 6: Add items use case

Use-Case Name:	Add items	Use-Case Type
		Business Requirements: Yes
Use-Case ID:	FI-U06	
Priority:	High	
Source:	FS – 11, FS – 12	
Primary Actor(s):	Warehouse Manager	
Description:	This use case allows the user to add new items to the warehouse	
Pre-Condition:	- The user should be logged into the application  -The user must have created a warehouse model	
Trigger:	When the user selects the add new items option	
Typical Course of Events:	ACTOR ACTION	SYSTEM RESPONSE
	Step 1 – The user clicks on add new items button   <	

	<p><b>Step 5 – The user can scroll through the list of all the locations where the items need to be stored</b></p> <p><b>Step 6 – The new items and their locations are saved in the database</b></p>
<b>Alternate Courses:</b>	<p>Step 4:</p> <p>The algorithm fails and returns an error that there is no space in the warehouse</p>
<b>Conclusion:</b>	This use case concludes when the items are added in the database
<b>Post condition:</b>	The add items form is reset
<b>Business Rules:</b>	
<b>Implementation Constraints and Specifications:</b>	- Use case must be available to every registered Warehouse Manager.
<b>Assumptions:</b>	Model for warehouse is created
<b>Open Issues:</b>	None

Table 7: View All Items Use Case

Use-Case Name:		View All Items	Use-Case Type
			Business Requirements: Yes
Use-Case ID:	FI-U07		
Priority:	Low		
Source:	FS-13		
Primary Actor(s):	Warehouse Manager		
Secondary Actor(s):	N/A		
Description:	This use case allows the user to get a detailed list for all the items in their warehouse.		
Pre-Condition:	- There must be items stored in the warehouse		
Trigger:	This use case is initiated when a user selects view items option		
Typical Course of Events:	ACTOR ACTION		SYSTEM RESPONSE
	Step 1 – The user selects view items option		Step 2 – The system loads all the relevant details regarding the items in that particular warehouse.
Alternate Courses:	Step 2:		

	If no items are found display a message that there are no items in the database
<b>Conclusion:</b>	This use case concludes when the user selects and calls another feature
<b>Post condition:</b>	The dataset is removed from main memory
<b>Business Rules:</b>	None
<b>Implementation Constraints and Specifications:</b>	<ul style="list-style-type: none"><li>- Use case must be available to every registered Warehouse Manager.</li><li>- There should be at least one item in the warehouse</li></ul>
<b>Assumptions:</b>	The warehouse model has been created

Table 8: Order items Use Case

Use-Case Name:	Order Items	Use-Case Type
		Business Requirements: Yes
Use-Case ID:	FI-U08	
Priority:	medium	
Source:	FS-3	
Primary Actor(s):	Customer	
Description:	The use case will be allow the user to order items from the warehouse	
Pre-Condition:		
Trigger:	The use case will be triggered when the user selects order items option	
Typical Course of Events:	ACTOR ACTION	SYSTEM RESPONSE
	Step 1 – The user selects the order items option	Step 2 – The system displays the order items form
	Step 3 – The user specifies the warehouse, the items and their quantity and submits the form	Step 4 – The system validates the form and makes sure all the required details are filled correctly
		Step 5 – The order is stored in the database

<b>Alternate Courses:</b>	Step 5:  The system presents an error if the form contains mistakes or incorrect data and the user is asked to fill the form correctly
<b>Conclusion:</b>	This use case concludes when the system successfully saves the order
<b>Post condition:</b>	A notification is generated for the warehouse manager that a new order is received
<b>Business Rules:</b>	The order can only be to the warehouse specified by the user
<b>Implementation Constraints and Specifications:</b>	<ul style="list-style-type: none"><li>- Use case must be available to every Customer.</li><li>- The items being ordered must be present in the warehouse</li></ul>
<b>Assumptions:</b>	None

Table 9: Add Worker use case

Use-Case Name:	Add Worker	Use-Case Type
		Business Requirements: Yes
Use-Case ID:	FI-U09	
Priority:	Medium	
Source:	FS-6	
Primary Actor(s):	Warehouse Manager	
Secondary Actor(s):	N/A	
Description:	This use case allows the warehouse manager to add new workers to their warehouse	
Pre-Condition:	The user must be logged in to the system	
Trigger:	This use case is initiated when the user selects the add worker option from the dashboard	
Typical Course of Events:	ACTOR ACTION	SYSTEM RESPONSE
	Step 1 – The user selects the add worker option	Step 2 – The system loads a form which asks for worker’s information, their designation and their responsible work areas.
	Step 3 – The user fills the form presses submit	

	<b>Step 4</b> – The data of new worker is verified and stored in database
<b>Alternate Courses:</b>	
<b>Conclusion:</b>	This use case concludes when the worker's information is saved
<b>Post condition:</b>	None
<b>Business Rules:</b>	- none
<b>Implementation Constraints and Specifications:</b>	- Use case must be available to every registered Warehouse Manager.
<b>Assumptions:</b>	None



Table 10: View Workers List use case

Use-Case Name:	View Workers List	Use-Case Type
		Business Requirements: Yes
Use-Case ID:	FI-U10	
Priority:	Low	
Source:	FS-5	
Primary Actor(s):	Warehouse Manager	
Description:	The use case when called upon will display the workers working in the warehouse	
Pre-Condition:	User must be logged in to the system	
Trigger:	This use case is initiated when a user selects the view workers option	
Typical Course of Events:	ACTOR ACTION	SYSTEM RESPONSE
	Step 1 – The user presses the view workers button	Step 2 – The System displays a list of workers
Alternate Courses:	Step 2:  If the list is empty display a message	

<b>Conclusion:</b>	This use case concludes when another feature is called by the user
<b>Post condition:</b>	none
<b>Business Rules:</b>	none
<b>Implementation Constraints and Specifications:</b>	- Use case must be available to every registered Warehouse Manager.
<b>Assumptions:</b>	None

Table 11: Detailed shelf view use case

Use-Case Name:	Receive Notifications	Use-Case Type
		Business Requirements: Yes
Use-Case ID:	FI-U11	
Priority:	Medium	
Source:	FS- 14	
Primary Actor(s):	System	
Secondary Actor(s):	Warehouse Manager	
Description:	The use case will allow the user to check each shelf and see the details of the items stored in each slot of the shelf	
Pre-Condition:	-The user must be logged in	
Trigger:	This use case is initiated when the user clicks on the shelf on view warehouse page	
Typical Course of Events:	ACTOR ACTION	
	SYSTEM RESPONSE	
	Step 1 – User Clicks on shelf	
	Step 2 – a model is opened and displayed where each shelf slot is visible, empty slots are coloured green whereas filled slots are red	
	Step 3 – The user hovers on a particular slot	

	<b>Step 4</b> – The details of the item stored in the slot are displayed
<b>Alternate Courses:</b>	
<b>Conclusion:</b> This use case once the user closes the view	
<b>Post condition:</b>	
<b>Business Rules:</b> - none	
<b>Implementation Constraints and Specifications:</b> - Use case must be available to every registered Warehouse Manager.	
<b>Assumptions:</b> None	

Table 12: Receive Notifications use case

Use-Case Name:	Receive Notifications	Use-Case Type
Use-Case ID:	FI-U12	Business Requirements: Yes
Priority:	Medium	
Source:	FS- 16,17,18,22	
Primary Actor(s):	System	
Secondary Actor(s):	Warehouse Manager	
Description:	The use case will give the user information regarding concerns.	
Pre-Condition:	-The user must be logged in	
Trigger:	This use case is initiated when the condition for the notification is met	
Typical Course of Events:	ACTOR ACTION	SYSTEM RESPONSE
		<p><b>Step 1</b> – System queries and checks the rules</p> <p><b>Step 2</b> – if the condition is met, the notification is issued</p> <p><b>Step 3</b> – The user clicks the show notifications button</p> <p><b>Step 4</b> – The notifications are displayed to the user</p>

<b>Alternate Courses:</b>	
<b>Conclusion:</b>	This use case concludes once the notifications are seen
<b>Post condition:</b>	
<b>Business Rules:</b>	- none
<b>Implementation Constraints and Specifications:</b>	
<b>Assumptions:</b>	None

Table 13: Issue Order for Dispatch Use Case

Use-Case Name:	Issue Order for Dispatch	Use-Case Type
		Business Requirements: Yes
Use-Case ID:	FI-U13	
Priority:	High	
Source:	FS- 19,20,21	
Primary Actor(s):	Warehouse Manager	
Secondary Actor(s):	Warehouse Worker	
Description:	This use case makes a list of the items that need to be dispatched	
Pre-Condition:	-User must be logged in  -There must be an order placed by the Customer	
Trigger:	This use case is initiated when the user activates the item dispatch option	
Typical Course of Events:	ACTOR ACTION	SYSTEM RESPONSE
	Step 1 – The user selects the item dispatch option  Step 3 – The user selects the order that needs to be dispatched	Step 2 – The system presents a list of all the orders placed by customers  Step 4 – The system decides where the items are picked from

	<p><b>Step 5 – The user confirms the issue list option</b> and makes a segmented list according to zones</p> <p><b>Step 6 – The warehouse worker receives the list and confirms when the items are ready for dispatch</b></p> <p><b>Step 7 – Once all the items are ready the items are dispatched</b></p>
<p><b>Alternate Courses:</b></p>	
<b>Conclusion:</b>	This use case concludes once the user confirms the dispatch
<b>Post condition:</b>	None
<b>Business Rules:</b>	- none
<b>Implementation Constraints and Specifications:</b>	<p>- Use case must be available to every registered Warehouse Manager.</p> <p>- - The model for the warehouse must be created</p>
<b>Assumptions:</b>	-There are items in the warehouse



## 2.7 Software Development Life Cycle Model

For the development of this project we will be using agile methodologies for software development. **Agile methodologies** will allow us the freedom to adjust development and adapt to problems during the development, especially with the tight time schedule for this project. Team members will use methods like pair programming to achieve project goals.

The team decided on this model enabled us to have the flexibility of working on the project with the time constraints. The team would often sit together or communicate on voice chat (Skype) to program parts of the project where input from both members were required. Weekly meetings with project advisor allowed us to demonstrate our project discussing possible changes or work around to problems that we encountered.

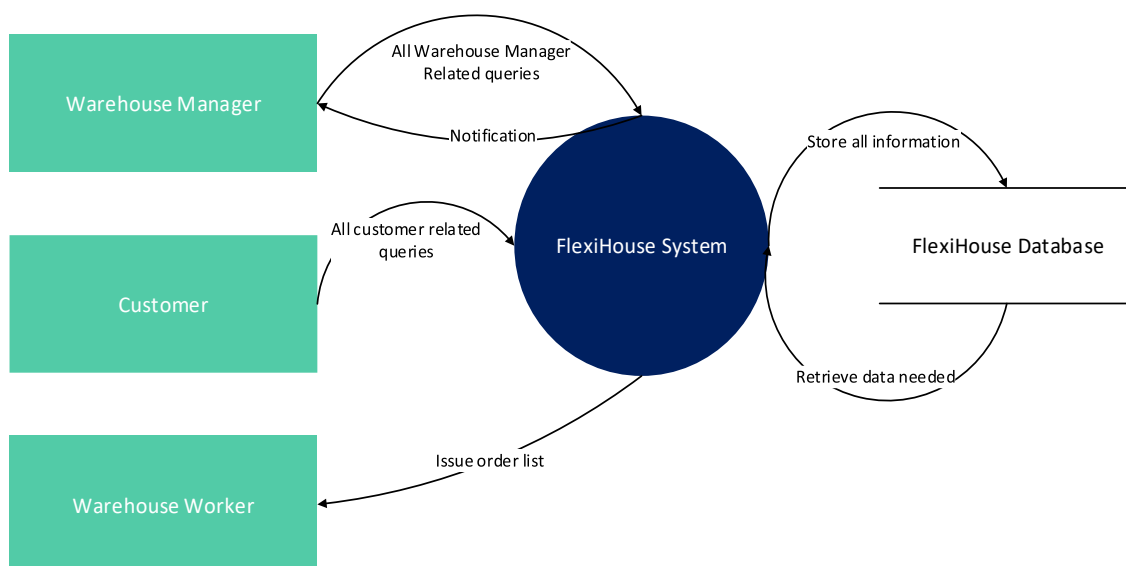
## 3 System Design and Implementation

### 3.1 System Architecture

This section discusses the system architecture of the Flexihouse Web Application along with description of its major components.

The solution being designed is a web application and we will be using ASP.NET MVC 5 to develop the project. We have chosen this technology because it is stable and has been in use for a considerable amount of time. One of the biggest advantages of using MVC is separation of concerns. The loose coupling between the components allows us to do parallel development.

As you can see in the context diagram below, there are three external interfaces to our system so each has to be separated from the other to minimize dependencies.



*Figure 4 Flexihouse Context Diagram*

The asp.net MVC architecture is a simple architecture which handles requests from user in specific manner. All the requests are first resolved by the router and then is passed to the controller, the controller decides which view to call and which model to bind to it. The following diagram shows the order in which ASP.NET processes requests.

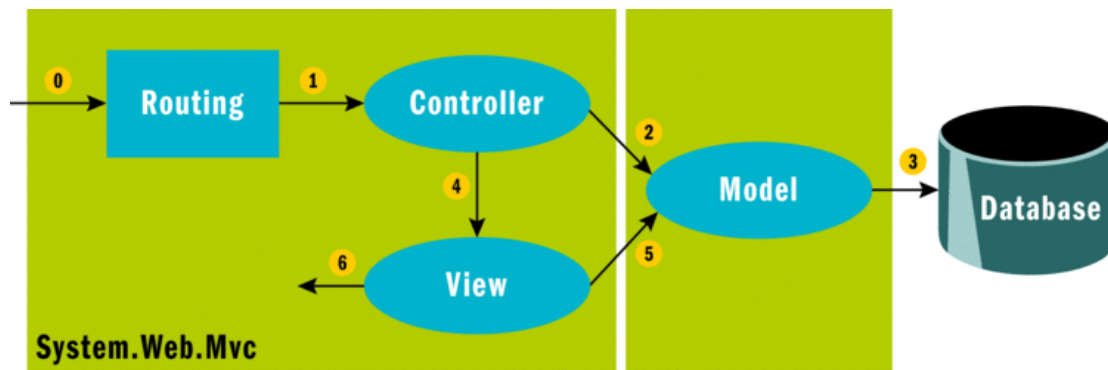


Figure 5: ASP.NET MVC Request Pipeline

As you can see there is a bit of separation between the model and the rest of the components, this is to explain that model is not accessed for all operations, sometimes no data is needed and thus the controller just returns the view directly.

The complete breakdown of Flexihouse is given below:

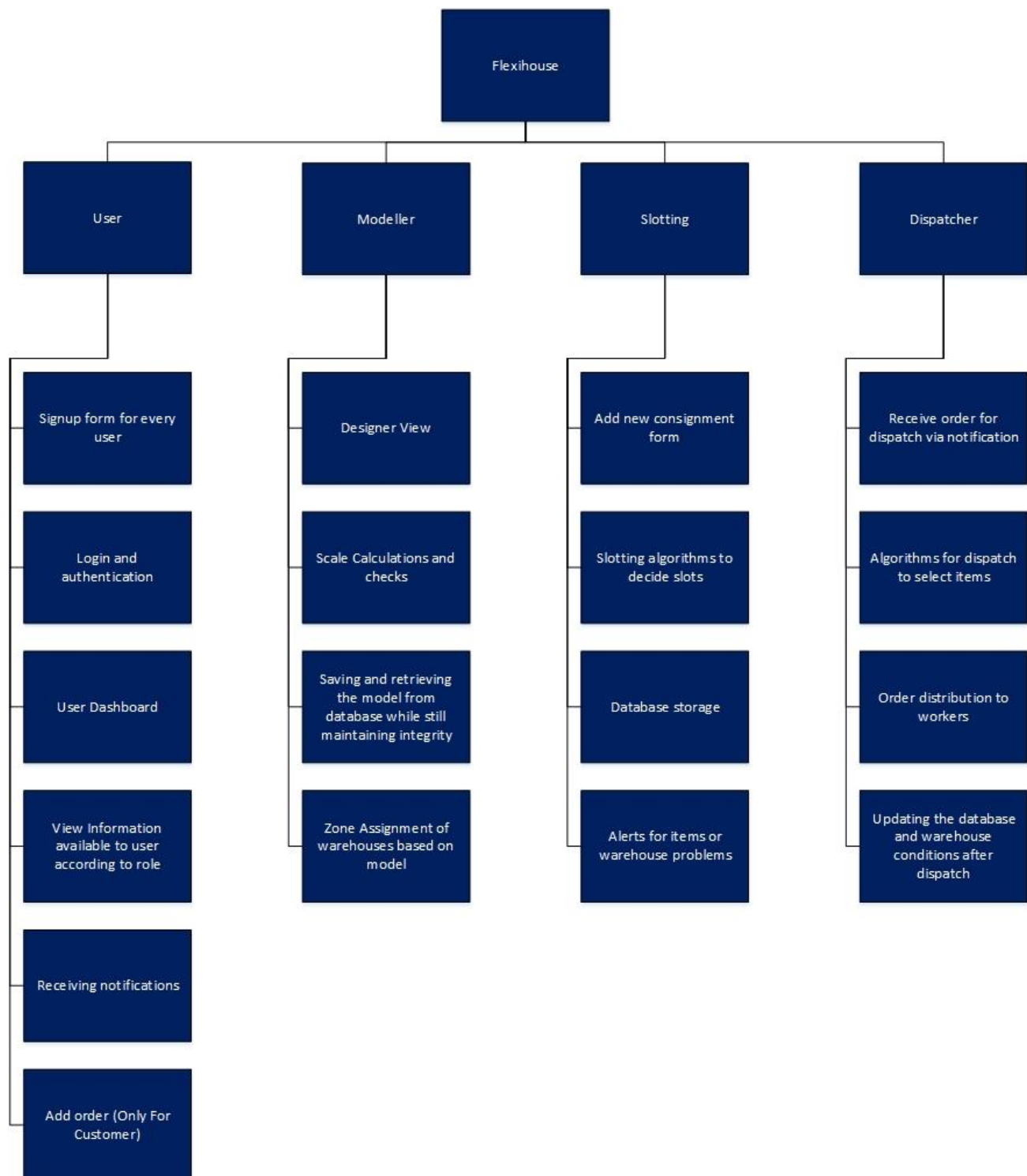


Figure 6: Product Breakdown Structure

This solution is divided into four major modules. An explanation of the modules is given below:

### 3.1.1 User Module

The user module deals with the niche features that the users expect from a particular application of this type. These include **User authentication**, **viewing information** and **receiving alerts or notifications**. The user module contains a dashboard in which the user options are listed according to their role.

### 3.1.2 Modeller Module

This module is related to the work done in the project for creating the model of the warehouse. We have a designer view in which the user designs the layout of the warehouse. The model is then processed to assign zones to it for item slotting optimization later on. The model of the warehouse must be editable and so database operations are required to save the model in its current state and retrieve it again. The model information will be stored in the **warehouse** table in the database. The designer view used multiple **Jquery** plugins.

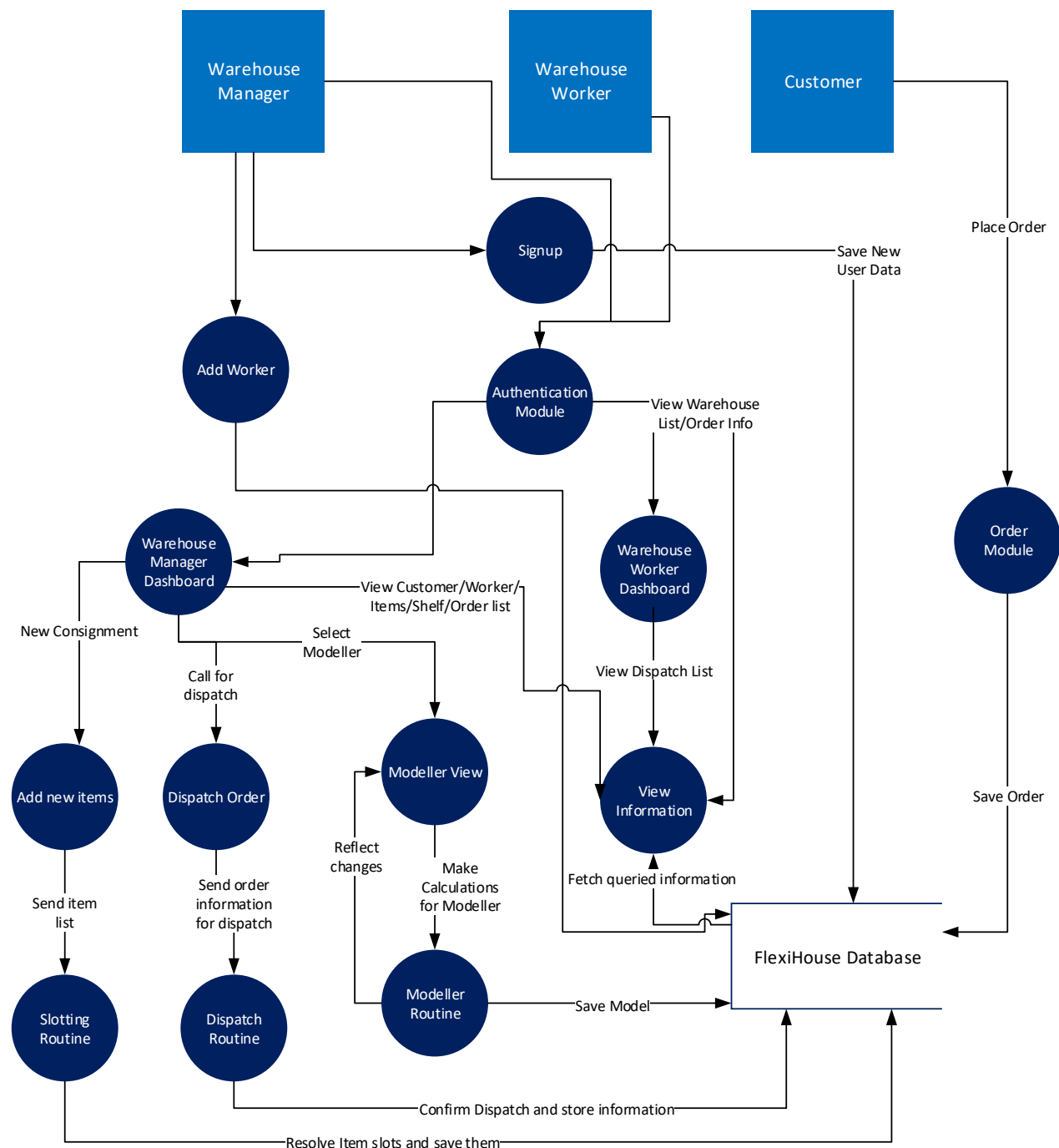
### 3.1.3 Slotting Module

This module handles all the related workings of adding new items in a warehouse, when a consignment arrives the user enters all the information regarding the items in the database. Then the slotting algorithm runs (**a combination of class based slotting and Fractional knapsack solution**) and decides where the items will be stored in the warehouse in terms of **shelf space optimization** as well as **picking time optimization**. If the warehouse is running out of space or an item is about to expire a notification is generated from this module.

### 3.1.4 Dispatcher Module

This module decides how to dispatch items from the warehouse according to the order placed by the customer. For **order picking** the module is running **an order batching algorithm based on seeds**. The orders are batched and instructions are generated for the workers to follow. To decide which item should be picked and then notifies the relevant worker in the warehouse to pick that order. Once all the items are picked, the transaction is marked as completed and the items are dispatched.

This section focuses on the data flow within the system. Every major module has a routine which makes all the calculations and handles the saving of data in the database.



### 3.3 Database Design

#### 3.3.1 Database Diagram

This section discusses the database structure of FlexiHouse. The project is using SQL Server database to store all the information. Following is the database ERD for the system.

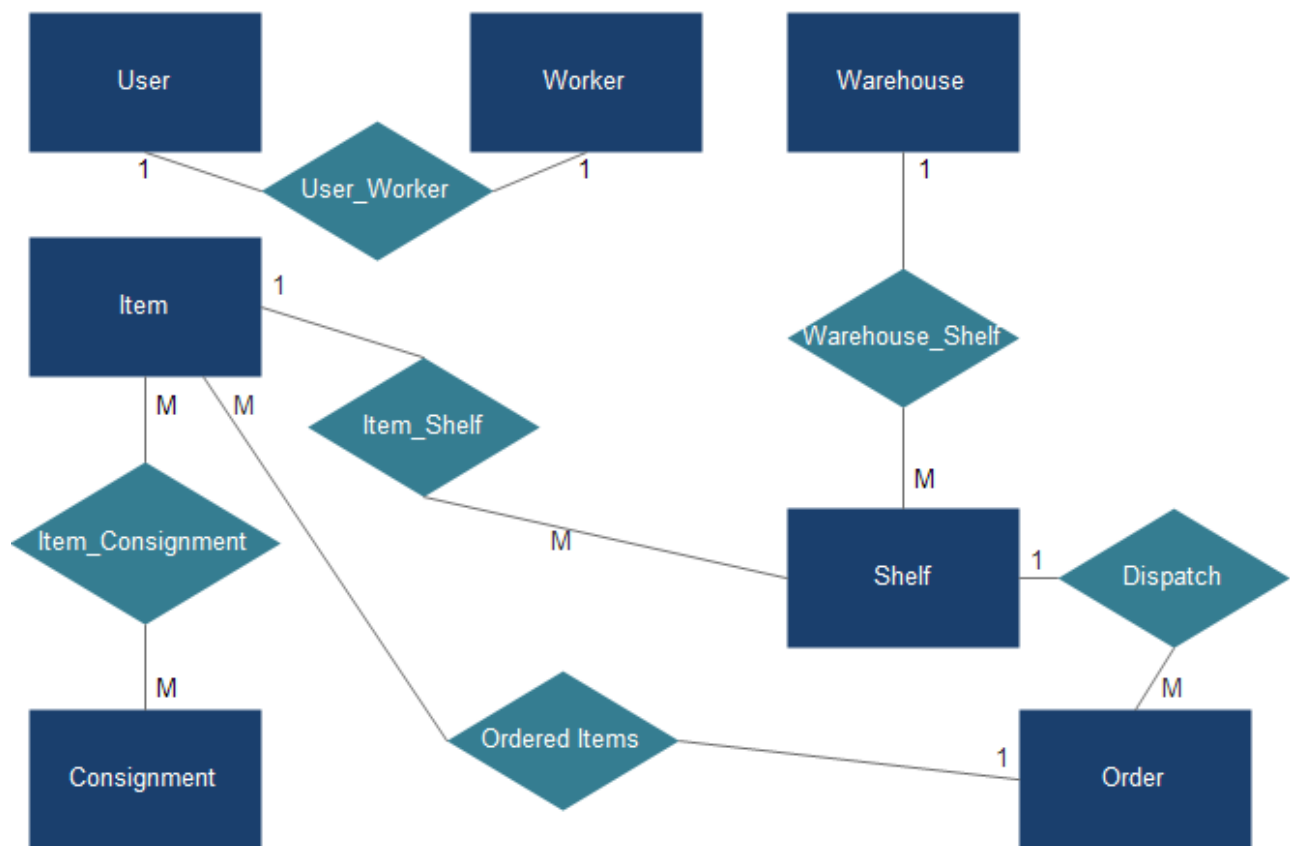


Figure 8: FlexiHouse ERD

There are some relationships in this database but only one extra table results from the relationships because of many to many cardinality. The database is not completely normalized for faster retrieval of data.

### 3.3.2 Data Dictionary

This section displays tables with the data definition of the various model/schema used in the database design.

Table 14: User Table

Field Name	Type	Constraints	Definition
<b>Id</b>	Int	Auto Generated, Primary Key	
<b>Username</b>	String	Unique, Required	required for authentication
<b>Password</b>	String	Required	Hashed Password
<b>Name</b>	String	Required	Contains the full name of user
<b>Role</b>	String	Required	Role of user
<b>Email</b>	String	Required	Email id of user
<b>Contact</b>	String		Contact information of user (Phone/Mobile)
<b>Registered</b>	Date		Account Creation Date



Table 15: Warehouse Table

Field Name	Type	Constraints	Definition
<b>Warehouse_Id</b>	Int	Auto Generated, Primary Key	Unique Id for warehouse
<b>Warehouse_Name</b>	String	Required	Name of warehouse
<b>Warehouse_Owner</b>	Int	Required, foreign key	User id of warehouse owner
<b>Dimensions</b>	String	Required	Dimensions of warehouse
<b>Scale</b>	String		Scaled dimensions of warehouse
<b>Shelves</b>	Int	Required	No. of shelves in the warehouse
<b>Shelf_dim</b>	String	Required	Shelf dimensions
<b>Model</b>	String		The layout of the Model

Table 16: Shelf Table

Field Name	Type	Constraints	Definition
<b>Shelf_id</b>	Int	Auto Generated, Primary Key	Unique Id for shelf in any warehouse
<b>Shelf_name</b>	String	Required	Name of shelf assigned in a particular warehouse
<b>Warehouse</b>	Int	Required, foreign key	Identifies which warehouse the shelf belongs to
<b>Class</b>	String	Required	Class/Zone the shelf belongs to
<b>Shelf_items</b>	String	Required	Information regarding the storage of items on the shelf
<b>Slots</b>	Int	Required	No. of slots for items on the shelf

Table 17: Item Table

Field Name	Type	Constraints	Definition
Item_id	Int	Auto Generated, Primary Key	Unique Id for item in any warehouse
Item_name	String	Required	Product name
Manufacturer	String		Product Manufacturer name
Country	String		Country the product is manufactured in
Item_code	String		Universal Product Code of item

Table 18: Item Details Table

Field Name	Type	Constraints	Definition
Item_id	Int	Primary Key, Foreign Key	Unique Id for item in any warehouse
Dimensions	String		Length x width x height in cm
Weight	Float		Weight of the product in grams
Picture	Image		A picture of the product

Table 19: Consignment Table

Field Name	Type	Constraints	Definition
<b>Consignment_id</b>	Int	Auto Generated, Primary Key	Unique Id for consignment
<b>Warehouse</b>	Int	Required, Foreign Key	Id of warehouse that the consignment belongs to
<b>Supplier</b>	String	Required	Name of consignment supplier
<b>Weight</b>	Int		Weight of consignment
<b>Arrival_date</b>	Date	Required	Arrival Date of Consignment
<b>Status</b>	String		Tells what state the consignment is in

Table 20: *item\_consignment Table*

Field Name	Type	Constraints	Definition
<b>Item_id</b>	Int	Foreign Key, Primary Key	Id of item from item table
<b>Consignment_id</b>	int	Foreign key, Primary key	Id of consignment
<b>Quantity</b>	int	Required	Quantity of item in the consignment
<b>Expiry</b>	Date		Item expiry date

Table 21: Order Table

Field Name	Type	Constraints	Definition
<b>Order_id</b>	Int	Auto Generated, Primary Key	Unique Id for order
<b>Customer_id</b>	Int	Required, foreign key	Identity of customer
<b>Order_date</b>	Date	Required	Date at which order was placed
<b>Order_status</b>	String	Required	It signifies order status e.g. Unseen Order, Seen Order, Dispatched Order
<b>Warehouse_id</b>	Int	Required	Warehouse to which the order is placed
<b>Instructions</b>	String		Instructions for order picking
<b>Dispatch Number</b>	Int		Identifies what orders were dispatched together

Table 22: Order Items Table

Field Name	Type	Constraints	Definition
Item_id	Int	Primary Key, Foreign Key	Unique Item id
Order_id	Int	Foreign Key, Primary Key	Identifies the order
Quantity	Int	Required	Quantity of item which is ordered



Table 23: Workers Table

Field Name	Type	Constraints	Definition
<b>Worker_id</b>	Int	Primary Key, Foreign Key	Identifies the worker
<b>Warehouse_id</b>	Int	Foreign Key, Primary Key	Identifies the warehouse
<b>Assigned Shelves</b>	Int	Required	The shelves this worker is responsible for

### 3.3.3 Database Normalization

We normalized databases as much as possible but not all tables could be in the third normal form as it was not feasible, these tables were de-normalized mainly for ease of programming.

## 3.4 Sequence Diagrams

This section explains the flow or sequence of the features of FlexiHouse in the form of sequence diagrams.

### 3.4.1 Signup Sequence

Use Case Reference: FI-U01

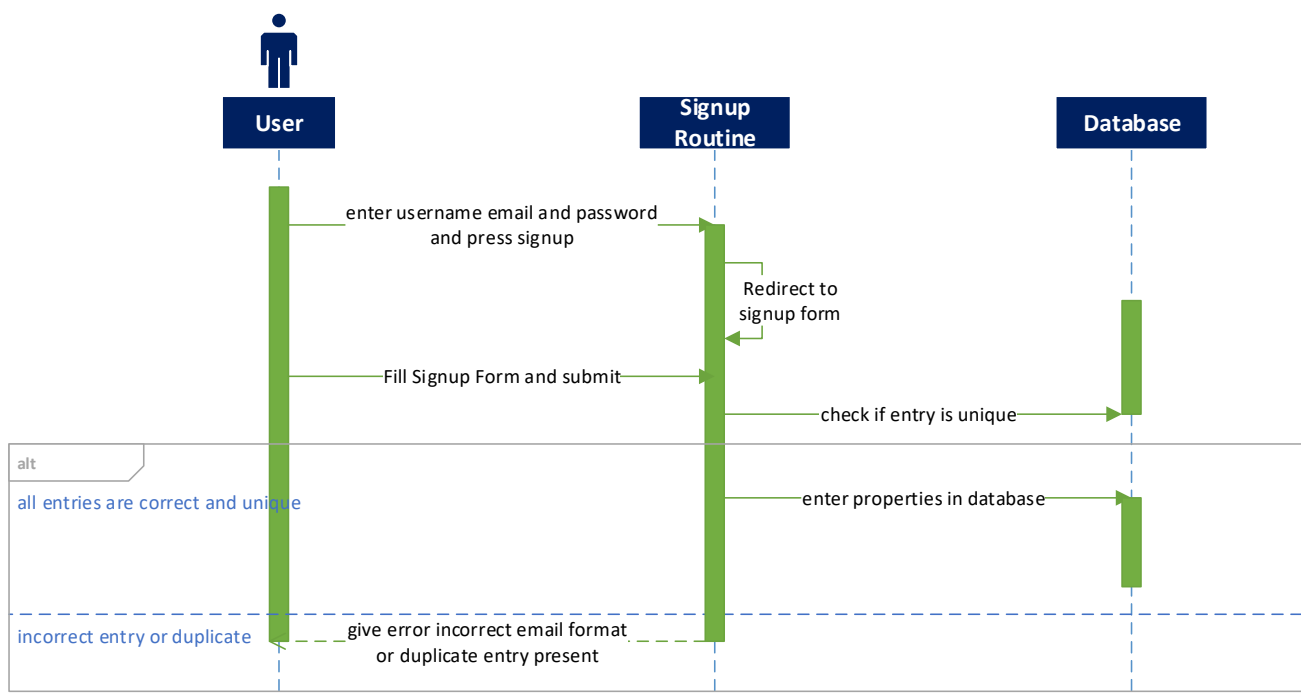


Figure 9: Flexihouse Signup Sequence Diagram

### 3.4.2 Login Sequence

Use Case Reference: FI-U02

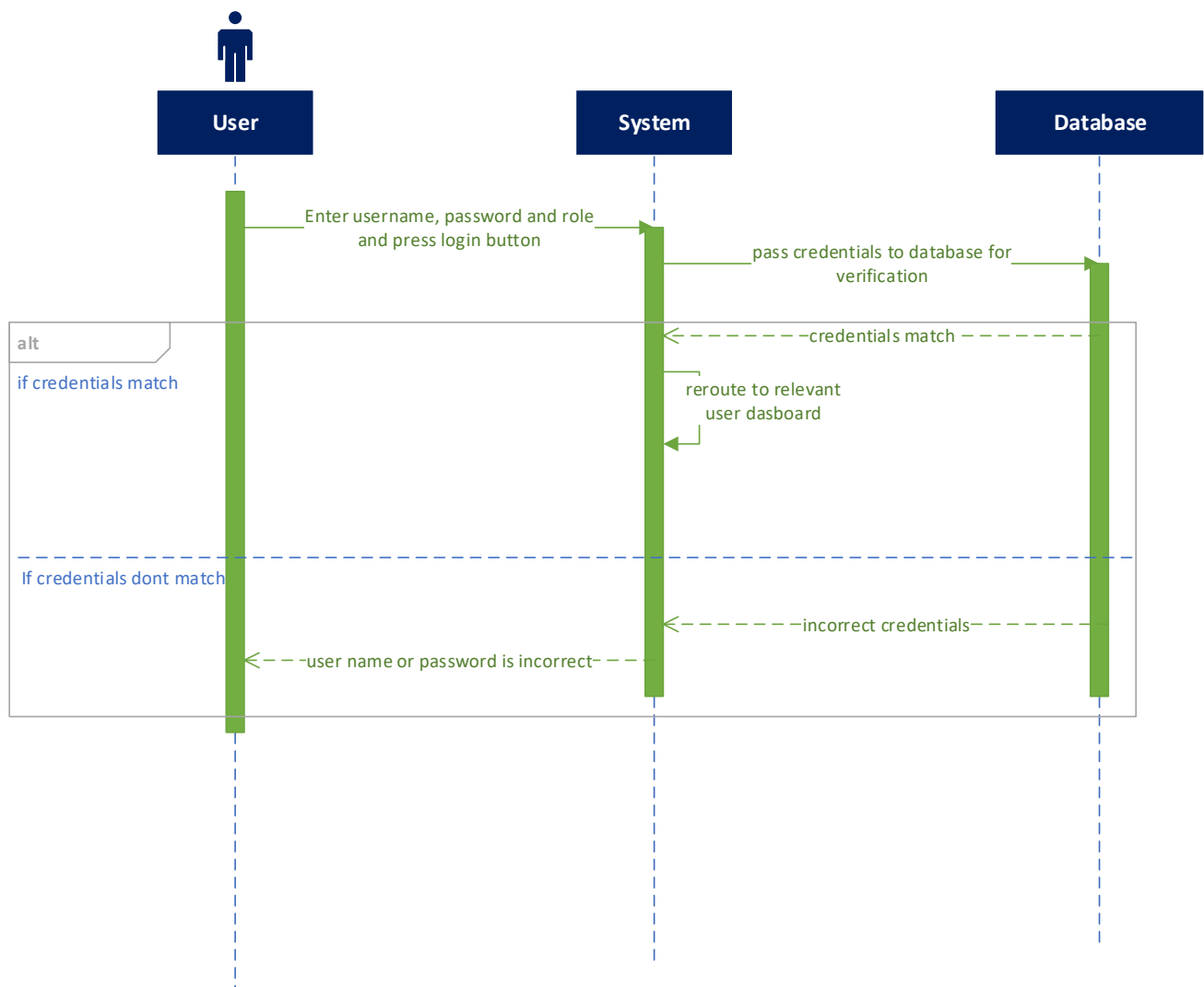


Figure 10: Flexihouse Login Sequence Diagram

### 3.4.3 Modeller Sequence

Use Case Reference: FI-U03, FI-U05

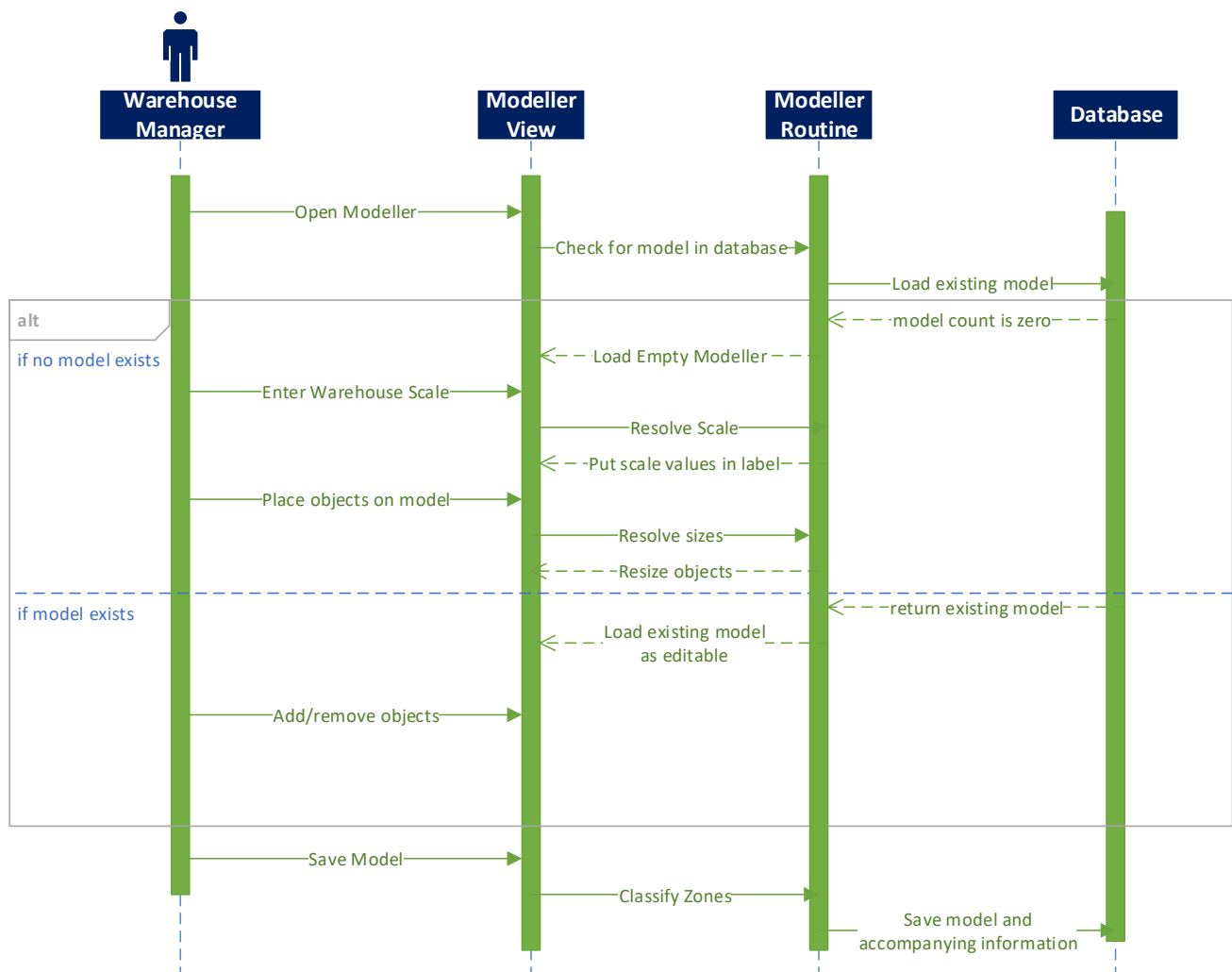


Figure 11: Flexihouse Modeller Sequence Diagram

### 3.4.4 Slotting Sequence

Use Case Reference: FI-U06

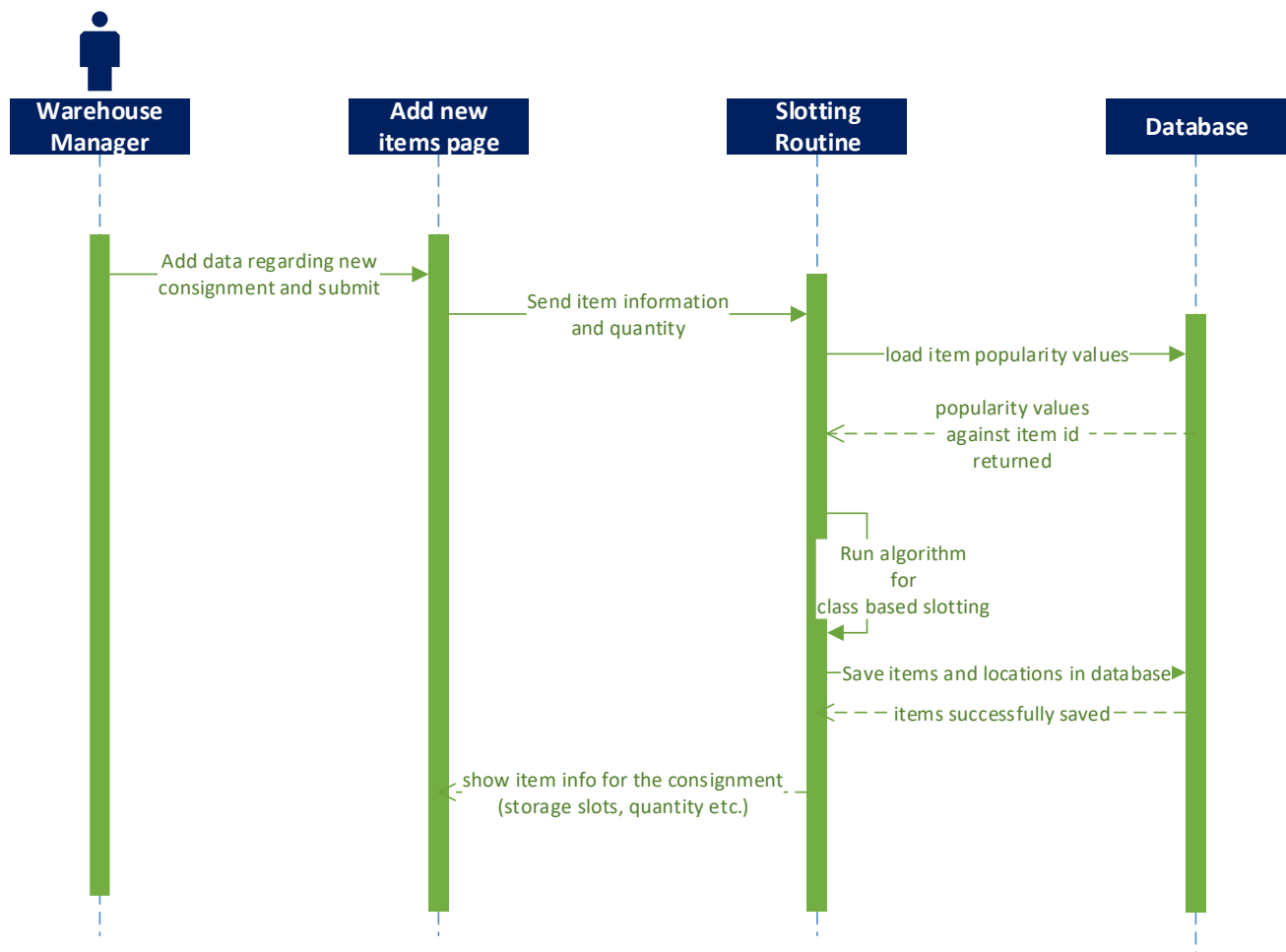


Figure 12: Flexihouse Slotting Sequence Diagram

### 3.4.5 View Information Sequence

Use Case Reference: FI-U07, FI-U09, FI-U10, FI-U11

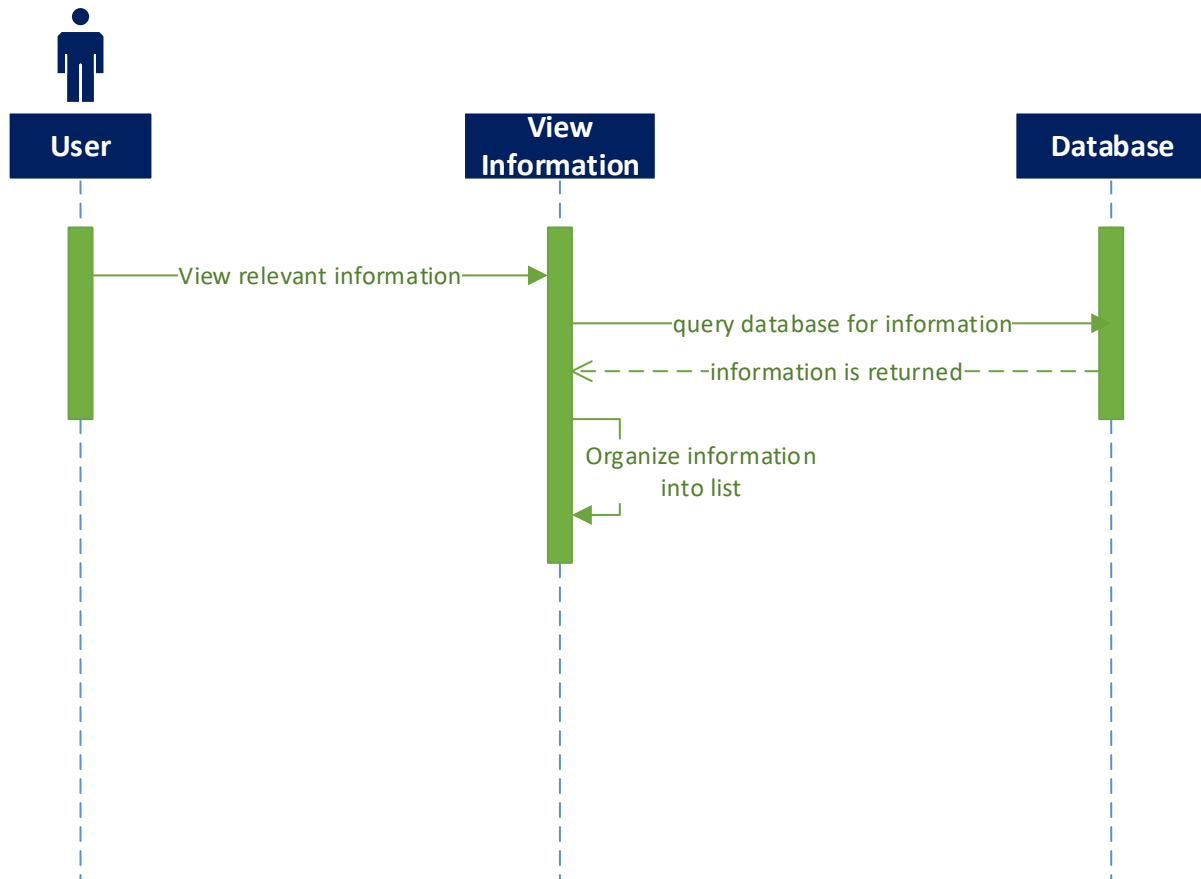


Figure 13: Flexihouse View Information Sequence Diagram

### 3.4.6 Place Order Sequence

Use Case Reference: FI-U08

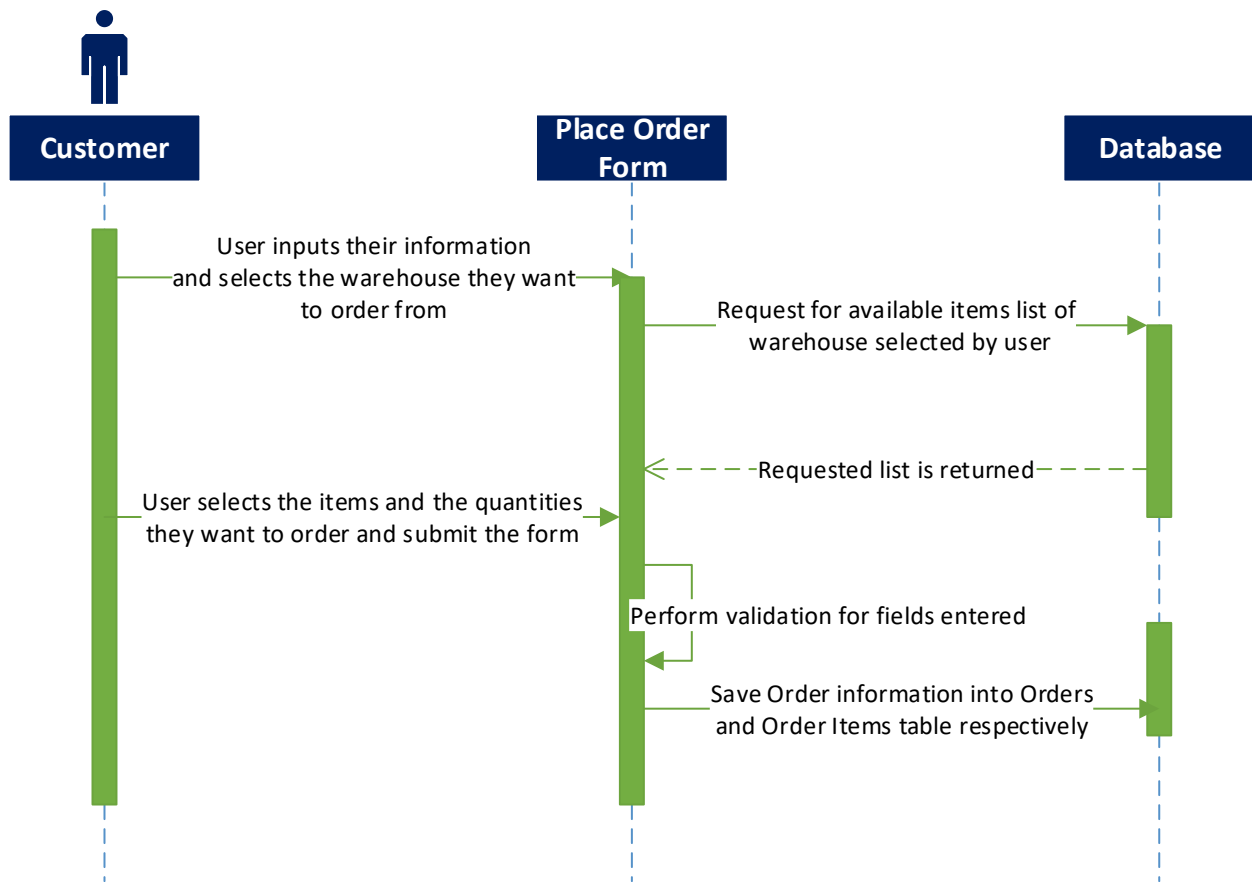


Figure 14: Place Order Sequence Diagram

### 3.4.7 Dispatch Items Sequence

Use Case Reference: FI-U13

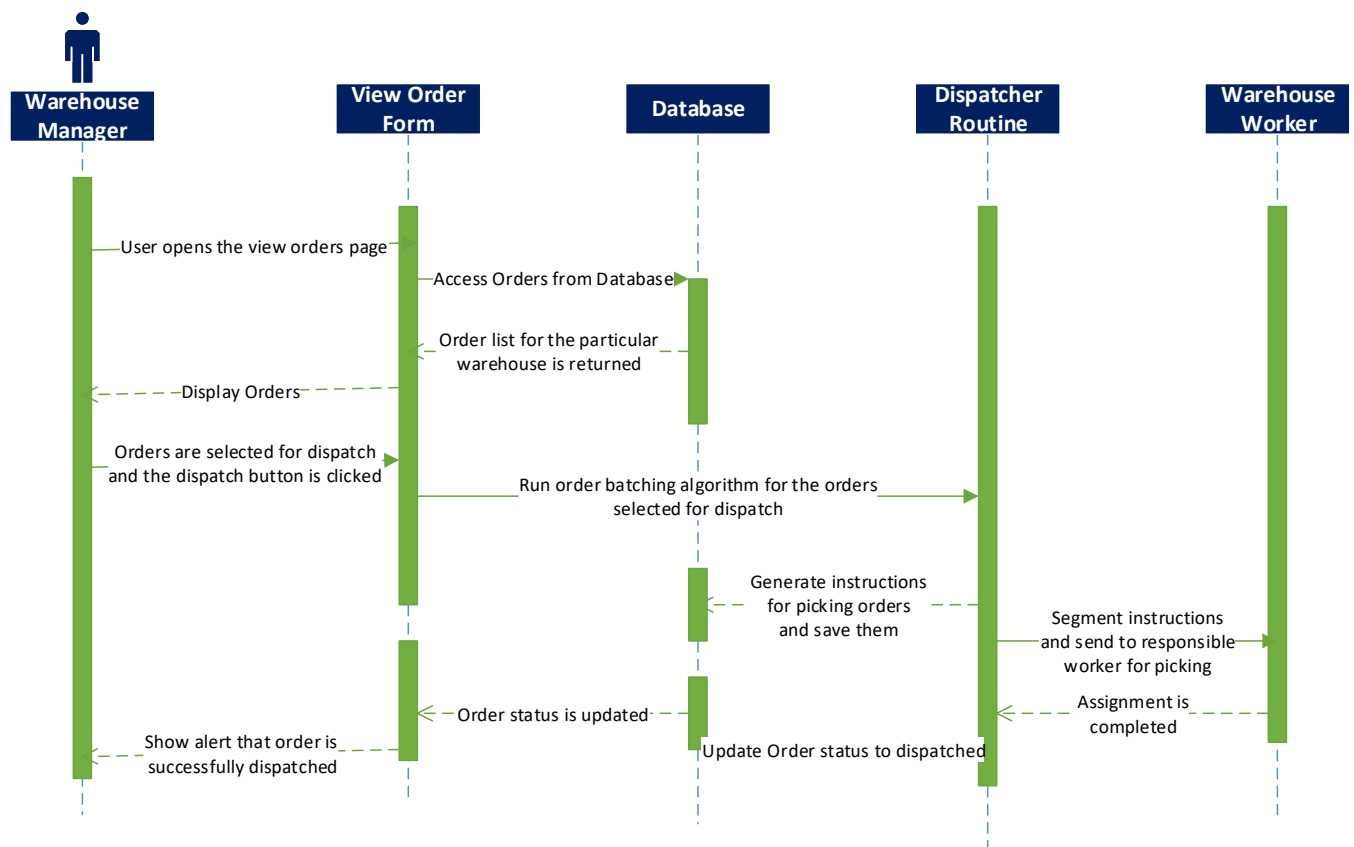


Figure 15: Dispatch Items Sequence Diagram



### 3.5 Requirements Traceability Matrix

The following matrix maps the functional recruitments with the use and test cases that satisfy them.

*Table 24: Requirement Traceability Matrix*

Use Case No.	Functional Specification No.	Test Case No.	Screen Shot	Progress
FI-U01	FS-1	FI - T01	SS – 01	Completed
FI -U02	FS-2	FI - T02	SS – 01	Completed
FI -U03,FI – U05	FS-7,8,10	FI - T03	SS – 02, SS – 03	Completed
FI -U04	FS-9,14	FI -T04	SS – 04	Completed
FI -U06	FS-13,15	FI - T05	SS – 05	Completed
FI -U07	FS-14	FI - T06	SS – 06	Completed
FI -U08	FS-3	FI - T07	SS – 12, SS – 13, SS - 14	Completed
FI -U09	FS-6	FI - T08	SS – 08	Completed
FI -U10	FS-5	FI – T10	SS – 09	Completed
FI -U11	FS-15	FI – T11	SS – 04	Completed
FI –U12	FS- 17,18,19	FI – T12	SS – 16	Completed
FI –U13	FS- 20,21,22	FI – T13	SS – 18	Completed

## 4 Testing

### 4.1 Scope of Testing

In testing we will test all the components of the system if possible. We will categorize the components of the system into high, moderate and low priority tiers. We will start with the testing of high priority components and then make our way down the priority chain. We will not be performing load and stress testing as our application might not be uploaded on a web server.

### 4.2 Test Plan Strategy

While testing our system, we will mostly be performing white box testing during the development stage and then move on to black box testing. The testing and development stage will move together, as every component which is being developed is tested individually at first and then integrated into the system. We will also design the test cases when we start the development of that module of the system because the test case may change depending on the implementation of the previous module.

As mentioned before the white box testing will take place during the development of the module's functions and at the end of the development module, we will perform black box testing to ensure everything is working as desired. After unit testing we will move onto integration testing, first we will integrate one module into the system completely and then worry about the integration between the modules themselves.

#### 4.2.1 Unit Testing

##### **Definition:**

Unit Test consists of testing individual programs or subroutines as they are written instead of testing the entire system after it has been written. The testing of the smaller building blocks is done first and then these blocks are combined and tested. Unit testing means testing each function independently to verify correct processing in a stand-alone environment.

In the unit testing phase we will test each form for its functionality individually to see if they produce the desired results, most of the tests will be written using a positive test case.

##### **Participants:**

Talha Amin, Amirul Sunesara

### **Methodology:**

- During the development phase every developer will test the component they were designing on a small scale (5-6 scenarios of data).
- After the development phase, in the testing phase all the unit test cases written will be tested by the participants mentioned above. Each participant will be responsible for the unit testing of the component they created.

### **4.2.2 Integration Testing**

#### **Definition:**

In integration testing, we will be using top down approach starting with the web forms that are encountered earlier in the functionality and then we will test each of our components with the database to make sure the correct type of data is being inserted and retrieved.

#### **Participants**

Talha Amin, Amirul Sunesara

#### **Methodology:**

- All the components that were coupled with each other were tested four times, once with correct values for both components, once with incorrect values for both and once with incorrect for the first only and correct for the second and vice versa.

### **4.2.3 System Testing**

#### **Definition:**

System testing is the part in which the testing team checks the overall functionality of a system as a whole. The non-functional requirements of the system are also checked in this phase.

#### **Participants**

Talha Amin, Amirul Sunesara

#### **Methodology:**

- Black box testing will be done in the system with boundary values.

- If there is a problem then it will be spotted and fixed and the whole testing phase starting from unit tests will be done on it.
- NFRs will only be checked for the system if they are applicable

### 4.3 Test Environment

Since the testing being done is mostly white box, we will be using the development environment itself for the testing so that we can see the code at all times. The black box testing will be done using a web browser because the system is a web application. The tools for testing which will be used are jsfiddle for UI and client side logic testing. Postman for testing JSON data representation. Visual studio and SQL server for backend code and database.

### 4.4 Schedule

Testing Activities	Begin	End	Person Responsible
Designing Test Cases	16-09-2016	28-03-2017	Talha Amin, Amirul Sunesara
Executing Test Cases	1-10-2016	05-05-2017	Talha Amin, Amirul Sunesara
Unit Testing	1-10-2016	30-04-2017	Talha Amin, Amirul Sunesara
Integration Testing	10-11-2016	01-05-2017	Talha Amin, Amirul Sunesara
System Testing	02-03-2017	05-05-2017	Talha Amin, Amirul Sunesara

## 4.5 Test Case Design and Description

This sections contains tables displaying the test cases for each of the different use cases documented earlier in the report.

*Table 25 Signup Test Case*

<b>Test-Case-ID</b>	FI-T01				
<b>Use-Case-ID</b>	FI-U01				
<b>Purpose:</b>	Testing the Signup Form.				
<b>Pre-requisite:</b>	User to be on the home page.				
<b>Test Data:</b>	<u>S. No</u>	<u>Variable</u>	<u>Valid Input</u>	<u>Invalid Input</u>	<u>Empty</u>
	1	Full Name	Alphabets allowed (MAX 50)	More than 50 alphabets.	Not allowed
	2	Email	Valid Email	Invalid Regular Expression for email	Not allowed
	3	Role	Select any of the three roles	No role Selected	Not allowed
	4	Password	More than 7 characters	Less than 7 characters	Not Allowed
	5	Repeat Password	Same value as password	Different value than password	Not Allowed
	6	Contact	Numerical of 11 digits	Non numerical or length more or less than 11	Not Allowed
	7	Username	Any string		Not Allowed
<b>Steps:</b>	Steps to carry out the test: 1. The user fills the form and clicks on the <b>Register</b> button on the landing page. 2. Information is verified. In case of empty on required field or improperly entered information the user is prompted to correct the information. 3. Once verified information is stored in the database. User is redirected to dashboard for designing warehouse.				
<b>Notes and Questions:</b>					

*Table 26: Login Test Case*

Test-Case-ID	FI-T02																			
Use-Case-ID	FI-U02																			
Purpose:	Testing the Login Form.																			
Pre-requisite:	User to be on the home page.																			
Test Data:	<table><tr><th><u>S. No</u></th><th><u>Variable</u></th><th><u>Valid Input</u></th><th><u>Invalid Input</u></th><th><u>Empty</u></th></tr><tr><td>1</td><td>Username</td><td>Any string</td><td></td><td>Not allowed</td></tr><tr><td>2</td><td>Password</td><td>Any String of more than 7 characters</td><td>String less than 7 characters</td><td>Not allowed</td></tr></table>					<u>S. No</u>	<u>Variable</u>	<u>Valid Input</u>	<u>Invalid Input</u>	<u>Empty</u>	1	Username	Any string		Not allowed	2	Password	Any String of more than 7 characters	String less than 7 characters	Not allowed
<u>S. No</u>	<u>Variable</u>	<u>Valid Input</u>	<u>Invalid Input</u>	<u>Empty</u>																
1	Username	Any string		Not allowed																
2	Password	Any String of more than 7 characters	String less than 7 characters	Not allowed																
Steps:	<p>Steps to carry out the test:</p> <ol style="list-style-type: none"><li>1. The user fills the form and clicks on the <b>Login</b> button on the landing page.</li><li>2. Information is verified. In case of empty on required field or improperly entered information the user is prompted to correct the information.</li><li>3. The information is verified from database, if information is correct user is redirected to dashboard.</li></ol>																			
Notes and Questions:																				

Table 27: Design Warehouse Test Case

<b>Test-Case-ID</b>	<b>FI-T03</b>				
<b>Use-Case-ID</b>	<b>FI-U03, FI-U04</b>				
<b>Purpose:</b>	Testing the Design Warehouse Form.				
<b>Pre-requisite:</b>	User to be on the Warehouse Manager Dashboard.				
<b>Test Data:</b>	<b>S. No</b>	<b>Variable</b>	<b>Valid Input</b>	<b>Invalid Input</b>	<b>Empty</b>
	1	Warehouse Actual Length	Any positive number	Negative value	Not allowed
	2	Warehouse Actual Width	Any positive number	Negative value	Not allowed
	3	Shelf Actual Length	Any positive number	Negative value	Not allowed
	4	Shelf Actual Width	Any positive number	Negative value	Not allowed
	5	Number of shelf rows	Any positive integer value.	Negative value or float value.	Not allowed
<b>Steps:</b>	<p>Steps to carry out the test:</p> <ol style="list-style-type: none"> <li>1. The user enters the information and presses ok.</li> <li>2. The warehouse scale is adjusted according to the dimensions provided.</li> <li>3. The design components are available to the user to design.</li> <li>4. The user designs the warehouse and presses save.</li> </ol> <p>The warehouse design is saved into warehouse.</p>				

Table 28: View Warehouse Test Case

<b>Test-Case-ID</b>	<b>FI-T04</b>
<b>Use-Case-ID</b>	<b>FI-U04</b>
<b>Purpose:</b>	Testing the View Warehouse Form.
<b>Pre-requisite:</b>	User to be on the Warehouse Manager Dashboard.
<b>Test Data:</b>	No information will be entered
<b>Steps:</b>	<p>Steps to carry out the test:</p> <ol style="list-style-type: none"> <li>1. User selects the view warehouse option from their dashboard.</li> <li>2. The warehouse model stored in the database is fetched and displayed</li> <li>3. The model displayed must be of the correct warehouse</li> </ol>

Table 29: Add Items Test Case

Test-Case-ID	FI-T05				
Use-Case-ID	FI-U05				
Purpose:	Testing the Add Items Form.				
Pre-requisite:	User to be on the Add items page.				
Test Data:	<u>S. No</u>	<u>Variable</u>	<u>Valid Input</u>	<u>Invalid Input</u>	<u>Empty</u>
	1	Supplier Name	Alphabets	Numeric Value	Not allowed
	2	Arrival Date	Selected date from date picker	Non date value	Not allowed
	3	Item Name	Any string		Not allowed
	4	Item Product code	Valid string of product code	Value doesn't satisfy regex	Not allowed
	5	Country	Valid Country Selected	No country selected	Not allowed
	6	Manufacturer	Any string		Not allowed
	7	Expiry Date	Selected date from date picker	Non date value	Allowed
	8	Quantity	Positive Integer Value	Negative or float value	Not allowed
Steps:	Steps to carry out the test:				
	<div><div>1.</div><div>The user enters all the information.</div></div> <div><div>2.</div><div>When item name is entered check from database, if it exists fill all the information regarding the item other than quantity and expiry date.</div></div> <div><div>3.</div><div>The user submits the form.</div></div> <div><div>4.</div><div>Make entry in database in items table if new item, make entry in consignment table, make entries in item-consignment table and also make entries in items warehouse table.</div></div> <div><div>5.</div><div>Run the slotting algorithm.</div></div> <div><div>6.</div><div>Save the instructions generated in consignment table.</div></div>				



Table 30: View Items Test Case

<b>Test-Case-ID</b>	<b>FI-T06</b>
<b>Use-Case-ID</b>	<b>FI-U07</b>
<b>Purpose:</b>	Testing the View Items Form.
<b>Pre-requisite:</b>	User to be on the Warehouse Manager Dashboard.
<b>Test Data:</b>	No information will be entered
<b>Steps:</b>	Steps to carry out the test: <ol style="list-style-type: none"> <li>1. User selects the view items option from their dashboard.</li> <li>2. The items stored in the database is fetched and displayed</li> <li>3. The items are displayed properly in a table</li> </ol>
<b>Notes and Questions:</b>	

Table 31: Place Order Test Case

<b>Test-Case-ID</b>	<b>FI-T07</b>				
<b>Use-Case-ID</b>	<b>FI-U08</b>				
<b>Purpose:</b>	Testing the Place Order Form.				
<b>Pre-requisite:</b>					
<b>Test Data:</b>	<b>S. No</b>	<b>Variable</b>	<b>Valid Input</b>	<b>Invalid Input</b>	<b>Empty</b>
	1	Customer Name	Alphabets	Numeric Value	Not allowed
	2	Customer Email	Selected date from date picker	Non date value	Not allowed
	3	Customer Address	Any string		Not allowed
	4	Warehouse Selected	Any Registered warehouse		Not allowed
	5	Items	Selected from drop down	Item not present in warehouse	Not allowed
	6	Quantity	Positive Integer Value	Negative or float value	Not allowed
<b>Steps:</b>	Steps to carry out the test: <ol style="list-style-type: none"> <li>1. User selects the place order option on the home screen.</li> <li>2. The place order form is displayed</li> <li>3. The user information must be in the correct format</li> <li>4. It should store properly in the database</li> </ol>				

Table 32: Add Worker Test Case

<b>Test-Case-ID</b>	<b>FI-T08</b>				
<b>Use-Case-ID</b>	<b>FI-U09</b>				
<b>Purpose:</b>	Testing the Worker Signup Form.				
<b>Pre-requisite:</b>	Manager must be logged in and present on Worker signup page.				
<b>Test Data:</b>	<u><b>S. No</b></u>	<u><b>Variable</b></u>	<u><b>Valid Input</b></u>	<u><b>Invalid Input</b></u>	<u><b>Empty</b></u>
	1	Full Name	Alphabets allowed (MAX 50)	More than 50 alphabets.	Not allowed
	2	Email	Valid Email	Invalid Regular Expression for email	Not allowed
	3	Selected shelves	Select shelves to assign	-	allowed
	4	Password	More than 7 characters	Less than 7 characters	Not Allowed
	5	Repeat Password	Same value as password	Different value than password	Not Allowed
	6	Contact	Numerical of 11 digits	Non numerical or length more or less than 11	Not Allowed
	7	Username	Any string		Not Allowed
<b>Steps:</b>	Steps to carry out the test: <ol style="list-style-type: none"> <li>1. The user fills the form and clicks on the <b>Register</b> button on the landing page.</li> <li>2. Information is verified. In case of empty on required field or improperly entered information the user is prompted to correct the information.</li> <li>3. Once verified information is stored in the database.</li> </ol>				
<b>Notes and Questions:</b>					

Table 33: Order Items Test Case

<b>Test-Case-ID</b>	<b>FI-T09</b>				
<b>Use-Case-ID</b>	<b>FI-U08</b>				
<b>Purpose:</b>	Testing the Place order form				
<b>Pre-requisite:</b>	Customer must have accessed the place order option				
<b>Test Data:</b>	<b><u>S. No</u></b>	<b><u>Variable</u></b>	<b><u>Valid Input</u></b>	<b><u>Invalid Input</u></b>	<b><u>Empty</u></b>
	1	Full Name	Alphabets allowed (MAX 50)	More than 50 alphabets.	Not allowed
	2	Email	Valid Email	Invalid Regular Expression for email	Not allowed
	3	Contact	Contact number	Invalid phone/cell number	Not allowed
	4	Address			Not Allowed
	5	Country	Select Country from drop down list	No country Selected	Not Allowed
	6	Warehouse	Select warehouse from which to order	No warehouse selected	Not Allowed
	7	Items	Select items which the user wants to order		Not Allowed
	8	Quantity	Positive integer value		
<b>Steps:</b>	Steps to carry out the test: <ol style="list-style-type: none"> <li>1. The user enters their personal and organizational information in the first step</li> <li>2. Information is verified. In case of empty on required fields or improperly entered information the user is prompted to correct the information.</li> <li>3. If the information is valid the user is taken to step 2 where they choose the items to order.</li> <li>4. The quantity of the items should be positive and within the limit of the available quantity</li> <li>5. The information is verified and stored in the database in the format specified</li> </ol>				
<b>Notes and Questions:</b>					

Table 34: View workers test case

<b>Test-Case-ID</b>	<b>FI-T010</b>
<b>Use-Case-ID</b>	<b>FI-U010</b>
<b>Purpose:</b>	Testing the view workers form
<b>Pre-requisite:</b>	Manager must be logged in
<b>Test Data:</b>	
<b>Steps:</b>	Steps to carry out the test: <ol style="list-style-type: none"> <li>1. The user must click the view workers option</li> <li>2. The information regarding the worker should be fetched from the database.</li> <li>3. The information to be displayed must be neatly arranged in a table</li> </ol>
<b>Notes and Questions:</b>	

Table 35: Detailed Shelf View test case

<b>Test-Case-ID</b>	<b>FI-T011</b>
<b>Use-Case-ID</b>	<b>FI-U011</b>
<b>Purpose:</b>	Testing the detailed shelf view feature
<b>Pre-requisite:</b>	Manager must be logged in
<b>Test Data:</b>	
<b>Steps:</b>	Steps to carry out the test: <ol style="list-style-type: none"> <li>1. The user must click the view warehouse option and select a particular shelf</li> <li>2. The information regarding the shelf should be fetched from the database.</li> <li>3. The information to be displayed must be neatly arranged in the view.</li> <li>4. Empty slots should be green in colour whereas filled slots should be red in colour</li> <li>5. On hover the item stored should be displayed if the slot is filled.</li> </ol>
<b>Notes and Questions:</b>	

Table 36: Receive notifications test case

<b>Test-Case-ID</b>	<b>FI-T012</b>
<b>Use-Case-ID</b>	<b>FI-U012</b>
<b>Purpose:</b>	Testing the notifications feature
<b>Pre-requisite:</b>	Manager, Worker must be logged in
<b>Test Data:</b>	
<b>Steps:</b>	Steps to carry out the test: <ol style="list-style-type: none"> <li>1. The user refreshes the page</li> <li>2. If there are any new orders or consignments added, the notification tab should display the number of notifications on it</li> <li>3. On selecting the notification, the user must be redirected to the specific page</li> </ol>
<b>Notes and Questions:</b>	

Table 37: Issue order for dispatch test case

<b>Test-Case-ID</b>	<b>FI-T013</b>
<b>Use-Case-ID</b>	<b>FI-U013</b>
<b>Purpose:</b>	Testing the order dispatch feature
<b>Pre-requisite:</b>	Manager must be logged in
<b>Test Data:</b>	
<b>Steps:</b>	Steps to carry out the test: <ol style="list-style-type: none"> <li>1. The user must be present on the view orders page.</li> <li>2. The user selects the order and clicks dispatch.</li> <li>3. The instructions list is passed on to the specific workers.</li> <li>4. They select the option “mark as complete” after performing the required task</li> <li>5. The order status is updated to “dispatched”</li> <li>6. The order is moved to the dispatched orders page.</li> </ol>
<b>Notes and Questions:</b>	

## **4.6 Status Reporting**

All the project members are responsible for weekly status reporting of the test cases and changes to be made to the test if there are features changes in the use case.

## **4.7 Risks and Assumptions**

- Computers with internet connections will be available at all times.
- Testers have the knowledge required to understand the bugs in the system.
- All the test cases pertaining to the module are written in advance.

## **4.8 Major Deliverables**

- Test plan document
- Test Cases

## **4.9 Exit Criteria**

- No more time remaining before delivery
- All high priority and mid bugs are removed
- There are no more bugs in the system

## 5 Conclusion

### 5.1 Lessons Learnt

This project helped us in getting substantiable experience in asp.net MVC and javascript technologies. We also learnt how to apply generic algorithms like knapsack algorithms into practical situations and how to modify algorithms according to the case.

We also learnt how to use version control as we used it as a medium to transfer the files of the project between the group members throughout the development of the project. This was a major learning point in our project because version control is used extensively in the professional environment and our experience of using it in this project will help us in the long run.

Overall the project was a culmination of our four years of learning and a few more tools and techniques that we learnt during the development of this project and helped us in developing a lot of skills for the future.

### 5.2 Future Enhancements

Following are some possible enhancements for the future in our project:

#### 5.2.1 Allow for more pallet sizes

The slotting algorithm can be modified to allow for different pallet sizes if the user desires. This problem stems from the fact that many warehouses don't use the standard ISO pallet size as we have used in our system.

#### 5.2.2 Improve Warehouse Modelling

The warehouse modelling feature can be improved upon and the model can be 3d, although there are many api's available for 3d modelling, none of them are designed for warehouses.

#### 5.2.3 Special items storage Conditions/ Designated area slotting

This is once again an improvement on the slotting algorithm, the system can be modified to allow for specific zones to be created in the warehouse for items that require special conditions like refrigeration.

## 6 Appendix


### 6.1 Glossary

Table 38: Key Terms

Key Words	Definitions
<b>Pallet</b>	A portable platform on which goods can be moved, stacked, and stored, especially with the aid of a forklift.
<b>Classes/Zones</b>	The shelf's priority in the warehouse
<b>0-1 Knapsack solution</b>	Dynamic Programming algorithm designed to solve knapsack problem
<b>Warehouse Model</b>	The floor plan of the warehouse
<b>Transaction</b>	Every item picked for dispatch is used to make up one transaction
<b>ASP.NET MVC</b>	The ASP.NET MVC framework provides an alternative to the ASP.NET Web Forms pattern for creating Web applications. The ASP.NET MVC framework is a lightweight, highly testable presentation framework that (as with Web Forms-based applications) is integrated with existing ASP.NET features, such as master pages and membership-based authentication.
<b>JS FIDDLE</b>	Java script, HTML and CSS Testing tool



## 6.2 Screenshots



The logo for FlexiHouse features a blue outline of a house with a bench inside. Below the house, the word "FLEXIHOUSE" is written in a blue, sans-serif font.

**Login to our site**

Enter username and password to log on:

**Sign in!**

**Sign up now**

Fill in the form below to get instant access:

**Sign me up!**

Figure 16: SS1 - Landing Page

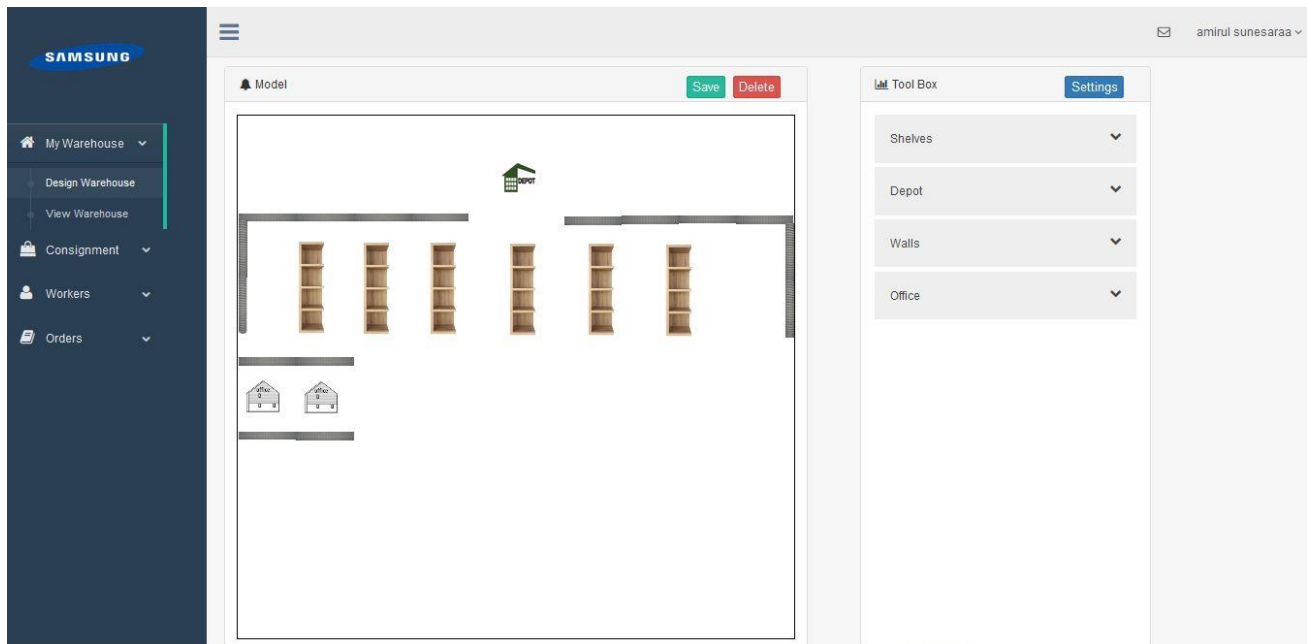


Figure 17: SS2 - Warehouse Design

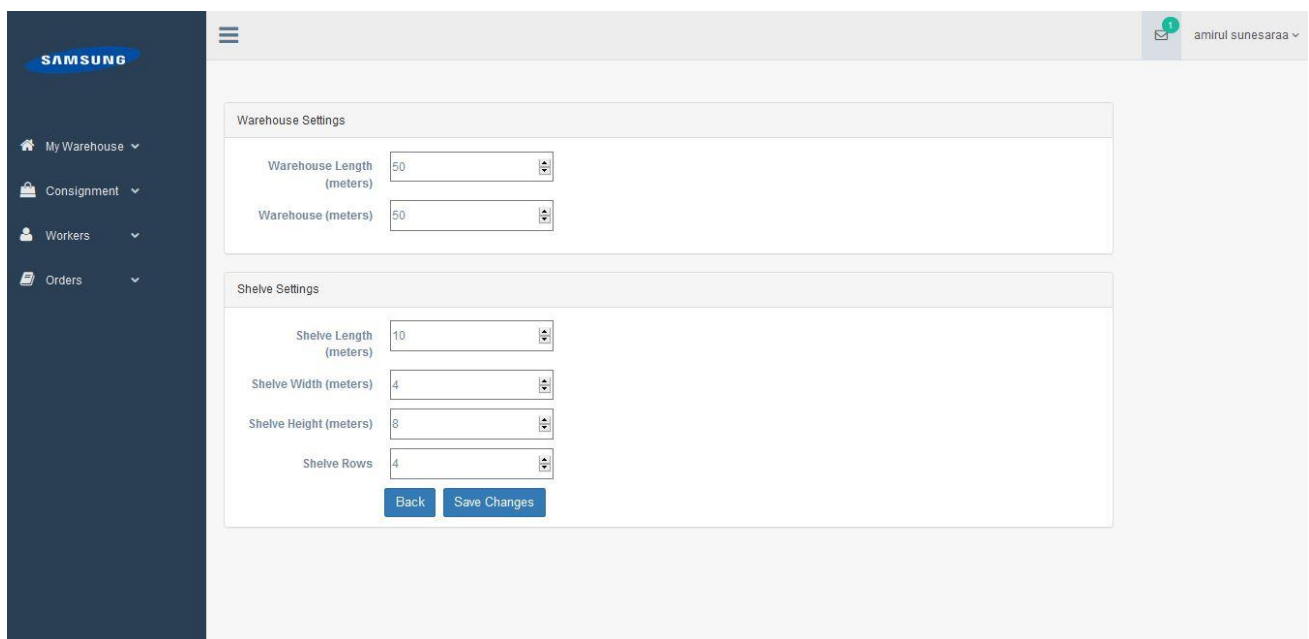


Figure 18: SS3 – Warehouse Dimensions

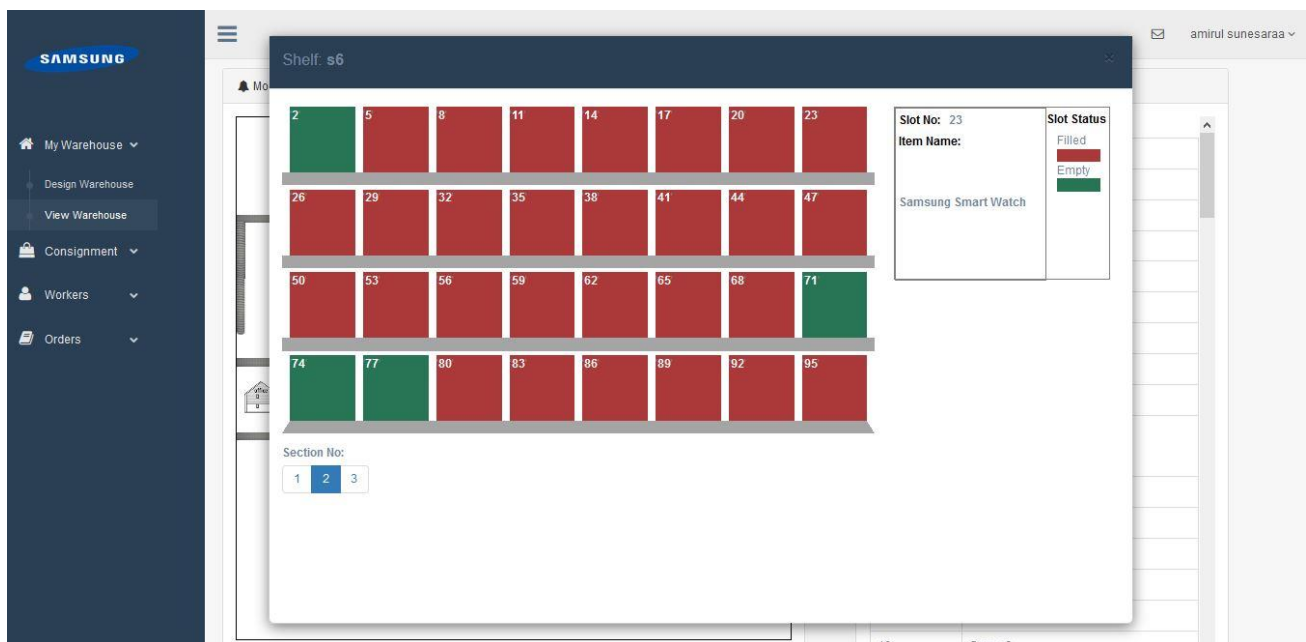


Figure 19: SS4 – Detailed Shelf View

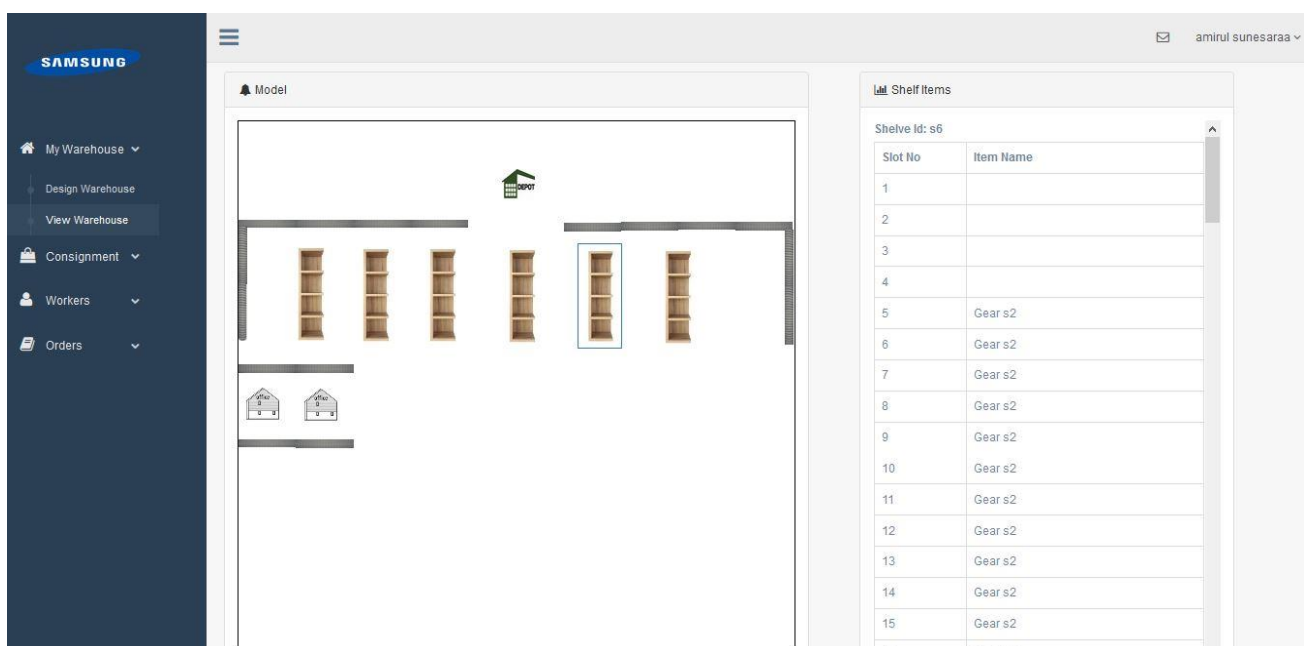


Figure 20: SS5 - View Warehouse

**SAMSUNG**

My Warehouse  
Consignment  
Add a Consignment  
View Consignments  
View Items  
Workers  
Orders

amirul sunesaraa

### Consignments

**Add a Consignment**

Consignment No: 10

Supplier Name: Soft Inc.

Arrival Date: 20/May/2017

Item Name	Quantity	Expiry Date	Product Code	Manufacturer	Country
samsung gear s2	2		S7100	Samsung	China

+ Add More Items

Save Consignment

Figure 21: SS5 – Add Consignment

**SAMSUNG**

My Warehouse  
Consignment  
Add a Consignment  
View Consignments  
View Items  
Workers  
Orders

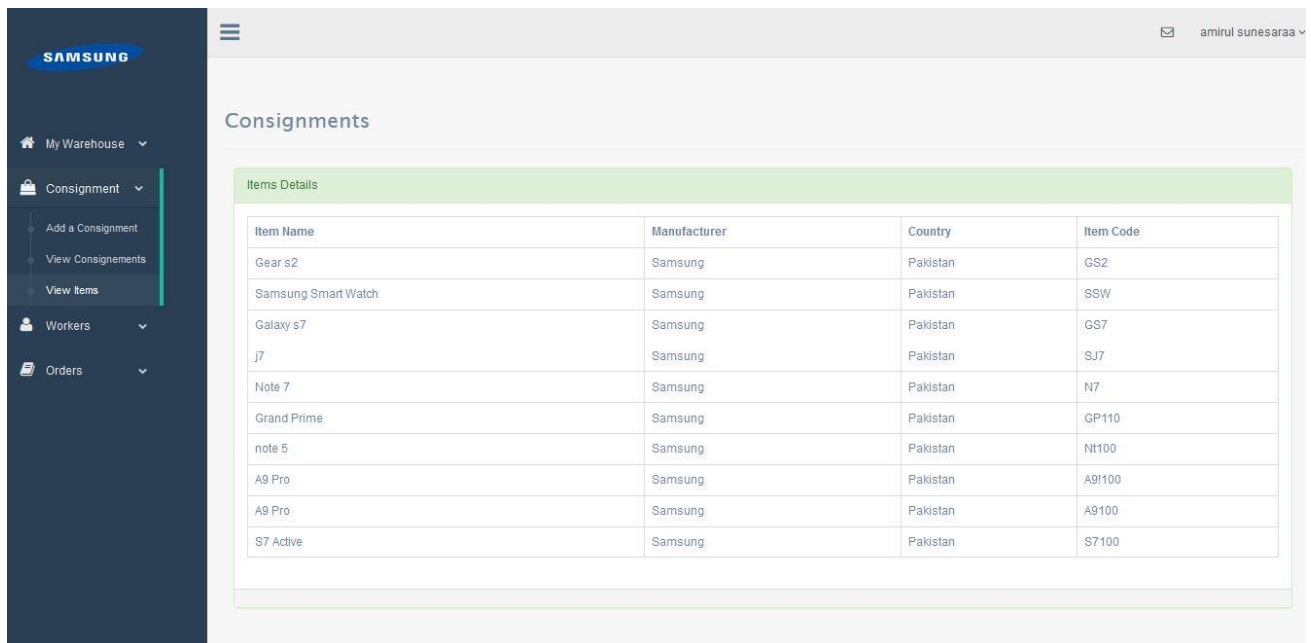
amirul sunesaraa

### Consignments

**Consignments**

Consignment ID	Supplier Name	Total Items	Arrival Date	
1	Samsung International	2	4/12/2017 12:00:00 AM	Edit Details
2	Samsung local	2	4/11/2017 12:00:00 AM	Edit Details
3	Samsung International	1	4/29/2017 12:00:00 AM	Edit Details
4	Samsung local	1	5/11/2017 12:00:00 AM	Edit Details
5	Abu Baker	1	5/16/2017 12:00:00 AM	Edit Details
6	Samsung Asia	1	5/25/2017 12:00:00 AM	Edit Details
7	Samsung Asia	1	5/25/2017 12:00:00 AM	Edit Details
8	Samsung Asia	1	5/30/2017 12:00:00 AM	Edit Details

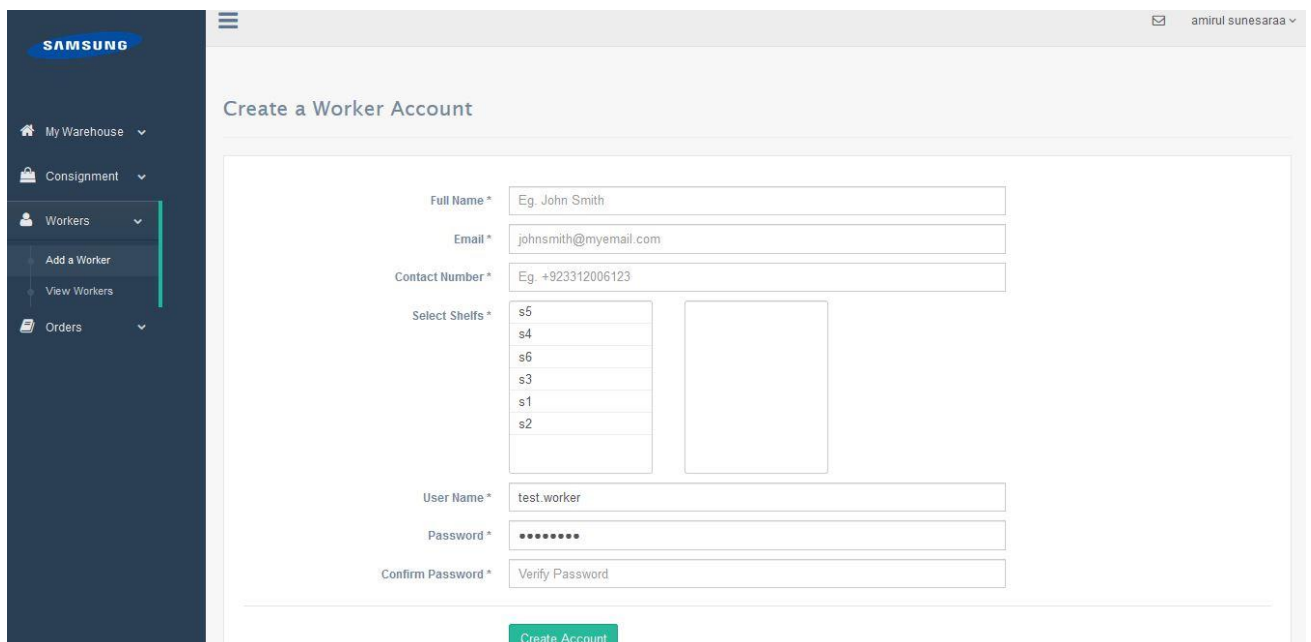
Figure 22: SS6 – View Consignments



The screenshot shows the 'Consignments' page in a Samsung FlexiHouse application. The left sidebar contains navigation options: My Warehouse, Consignment (selected), Add a Consignment, View Consignments, View Items, Workers, and Orders. The main content area is titled 'Consignments' and features a table of items under the heading 'Items Details'.

Item Name	Manufacturer	Country	Item Code
Gear s2	Samsung	Pakistan	GS2
Samsung Smart Watch	Samsung	Pakistan	SSW
Galaxy s7	Samsung	Pakistan	GS7
J7	Samsung	Pakistan	SJ7
Note 7	Samsung	Pakistan	N7
Grand Prime	Samsung	Pakistan	GP110
note 5	Samsung	Pakistan	Nt100
A9 Pro	Samsung	Pakistan	A9100
A9 Pro	Samsung	Pakistan	A9100
S7 Active	Samsung	Pakistan	S7100

Figure 23: SS7 – View Items



The screenshot shows the 'Create a Worker Account' page in the Samsung FlexiHouse application. The left sidebar is the same as in Figure 23, with 'Workers' selected. The main content area is titled 'Create a Worker Account' and contains a form with the following fields:

- Full Name \*: Eg. John Smith
- Email \*: johnsmith@myemail.com
- Contact Number \*: Eg. +923312006123
- Select Shelves \*: A list of shelves (s5, s4, s6, s3, s1, s2) and a large empty box for additional shelves.
- User Name \*: test.worker
- Password \*: A field with masked characters (dots).
- Confirm Password \*: Verify Password

A green 'Create Account' button is located at the bottom of the form.

Figure 24: SS8 – Create Worker Account

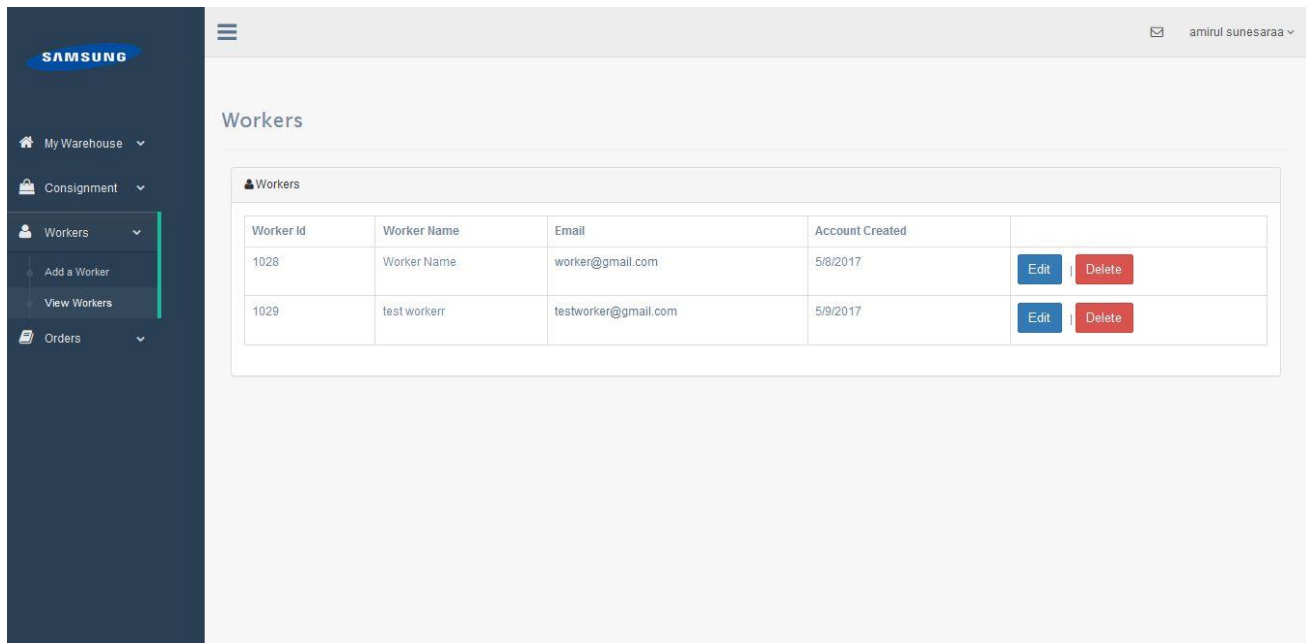


Figure 25: SS9 – View Registered Workers

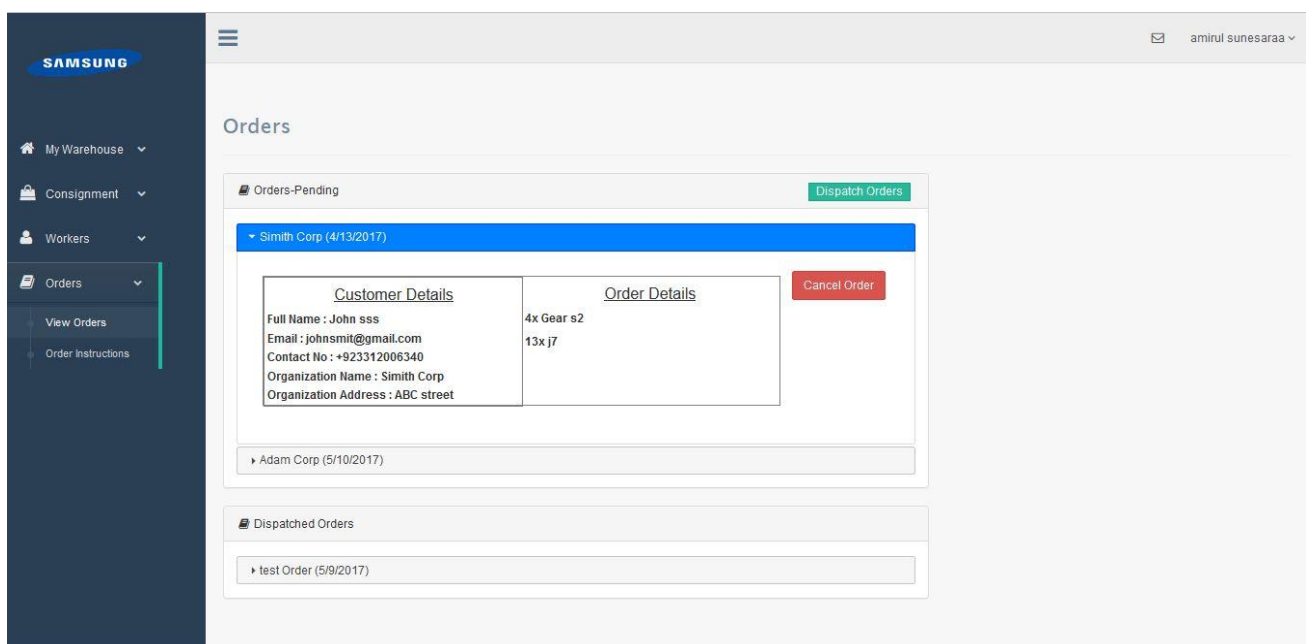
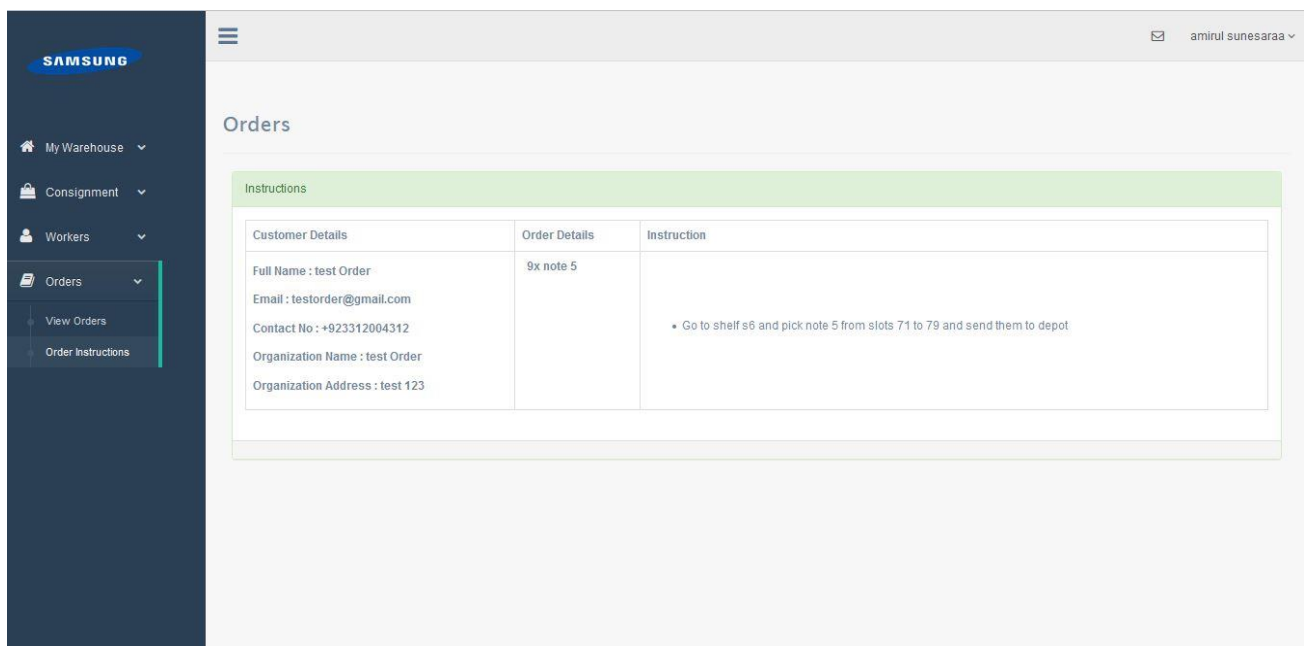


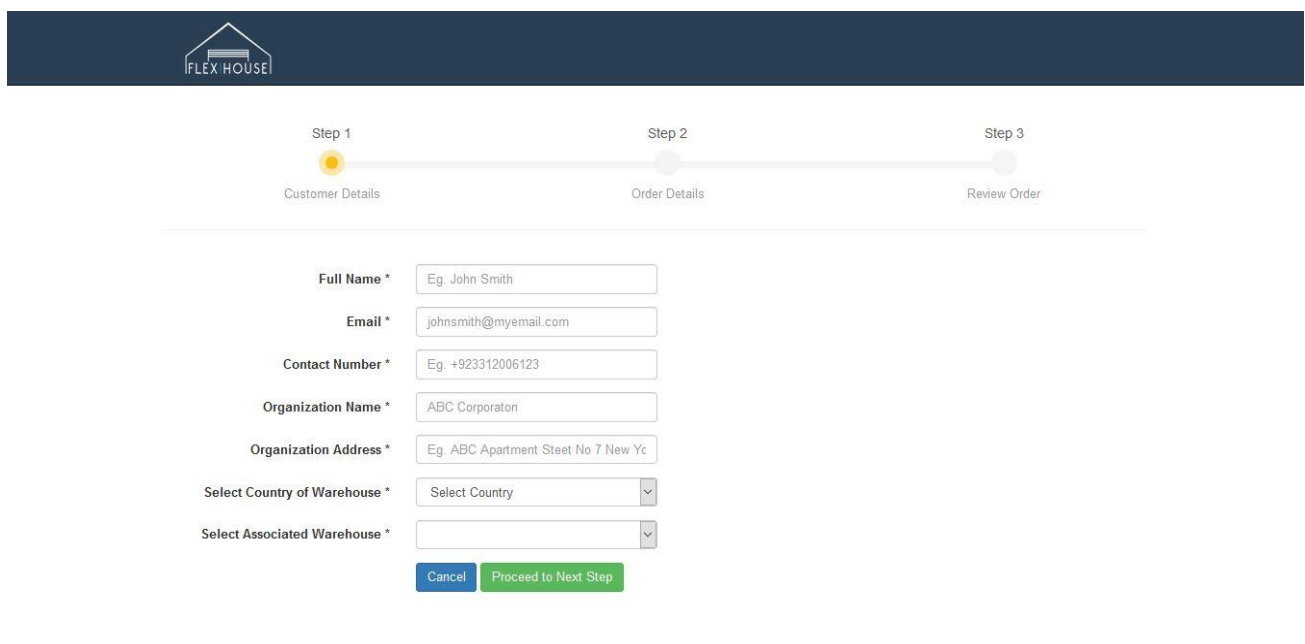
Figure 26: SS10 – View Orders



The screenshot shows a web interface for Samsung orders. On the left is a dark blue sidebar with the Samsung logo and a menu with options: My Warehouse, Consignment, Workers, Orders (selected), View Orders, and Order Instructions. The main content area is titled 'Orders' and features a green header for 'Instructions'. Below this is a table with three columns: Customer Details, Order Details, and Instruction.

Customer Details	Order Details	Instruction
Full Name : test Order Email : testorder@gmail.com Contact No : +923312004312 Organization Name : test Order Organization Address : test 123	9x note 5	<ul style="list-style-type: none"> <li>Go to shelf s6 and pick note 5 from slots 71 to 79 and send them to depot</li> </ul>

Figure 27: SS11 – Order Instructions




The screenshot shows the FlexiHouse logo at the top. Below it is a progress bar with three steps: Step 1 (Customer Details, active), Step 2 (Order Details), and Step 3 (Review Order). The main form is for Step 1, 'Customer Details', and includes the following fields:

- Full Name \***: Text input with placeholder 'Eg. John Smith'
- Email \***: Text input with placeholder 'johnsmith@myemail.com'
- Contact Number \***: Text input with placeholder 'Eg. +923312006123'
- Organization Name \***: Text input with placeholder 'ABC Corporation'
- Organization Address \***: Text input with placeholder 'Eg. ABC Apartment Steet No 7 New Yc'
- Select Country of Warehouse \***: Dropdown menu with 'Select Country' as the selected option
- Select Associated Warehouse \***: Dropdown menu

At the bottom of the form are two buttons: 'Cancel' (blue) and 'Proceed to Next Step' (green).

Figure 28: SS12 – Order step 1



Step 1

Step 2

Step 3

Customer Details

Order Details


Review Order

Item Name	Manufacturer	Item Code	Available Quantity	Choose Quantity	
Samsung Smart Watch	Samsung	SSW	30	15	
Gear s2	Samsung	GS2	26	10	<a href="#">Remove</a>

[+ Add More Items](#)

[Go Back](#) [Proceed to Next Step](#)

Figure 29: SS13 – Order step 2



Step 1

Step 2

Step 3

Customer Details

Order Details

Review Order

Customer Details

Full Name : Usman khalid  
Email : usmankhalid123@gmail.com  
Contact No : +923312964234  
Organization Name : Genpack electronics  
Organization Address : Saddar Mobile Market  
Selected Warehouse : Samsung Regional Warehouse

Order Details

15x Samsung Smart Watch  
10x Gear s2

[Go Back](#) [Complete Order](#)

Figure 30: SS14 – Order step 3



**SAMSUNG**

My Warehouse  
Consignment  
Workers  
Orders

**Edit Profile**

Full Name \* amirul sunesaraa

Email \* amirul.sunesara@gmail.com

Contact Number \* +923312006340

User Name \* amirul.sunesara

Password \* .....

Confirm Password \* .....

Edit Account

Figure 31: SS15 – Edit Profile

**FLEXIHOUSE**

Dashboard  
Consignments  
New Consignments 5  
Slotted Consignments  
Orders

**Consignments**

	Supplier Name	Total Items	Arrival Date	
New	Samsung International	2	4/12/2017 12:00:00 AM	Consignment Instructions
New	Samsung local	2	4/11/2017 12:00:00 AM	Consignment Instructions
New	Samsung International	1	4/29/2017 12:00:00 AM	Consignment Instructions
New	Samsung local	1	5/11/2017 12:00:00 AM	Consignment Instructions
New	Abu Baker	1	5/16/2017 12:00:00 AM	Consignment Instructions

Figure 32: SS16 – Worker - New Consignments

**Consignment Details**

Supplier Name : Samsung International  
Total Items : 2  
Arrival Date : 4/12/2017

**Consignment Items**

Item Name	Quantity	Expiry Date	Product Code
Gear s2	30		GS2
Samsung Smart Watch	30		SSW

**Slotting Instructions**

Place Gear s2 on shelf 3s6, from slot 1 to slot 30  
Place Samsung Smart Watch on shelf 3s6, from slot 31 to slot 60

[Back to Consignments](#) [Mark as Complete](#)

**Received Orders**

- Smith Corp  
Received Order From John sss 4/13/2017
- test Order  
Received Order From test Order 5/9/2017
- Received Consignment from Samsung local 4/11/2017
- Received Consignment from Samsung International 4/29/2017
- Received Consignment from Samsung local 5/11/2017
- Received Consignment from Abu Baker 5/16/2017

Figure 33: SS17 – Consignment Instruction Page

**Orders**

	Customer Details	Order Details	Dispatched on	Instruction	
New	<p>Full Name : John sss</p> <p>Email : johnsmit@gmail.com</p> <p>Contact No : +923312006340</p> <p>Organization Name : Simith Corp</p> <p>Organization Address : ABC street</p>	<p>4x Gear s2</p> <p>13x j7</p>	4/13/2017	<ul style="list-style-type: none"> <li>Go to shelf s6 and pick Gear s2 from slots 1 to 4 and send them to depot</li> <li>Go to shelf s3 and pick j7 from slots 21 to 33 and send them to depot</li> </ul>	<a href="#">Mark as Completed Order</a>
New	<p>Full Name : test Order</p> <p>Email : testorder@gmail.com</p> <p>Contact No : +923312004312</p> <p>Organization Name : test Order</p> <p>Organization Address : test 123</p>	<p>9x note 5</p>	5/9/2017	<ul style="list-style-type: none"> <li>Go to shelf s6 and pick note 5 from slots 71 to 79 and send them to depot</li> </ul>	<a href="#">Mark as Completed Order</a>

Figure 34: SS18 – New Orders