Inventory Management System (IMS)

Design Document

Overview of Software Design and Architecture

Kyle Parker

Christopher Andrews

Chaitanya Chakka

Erik Wojcik

Sarim Janjua

Mohib Ahmed

Nick Turner

Quentin Terry

Contents

1. Introduction

This software is an application that will track stock and inventory maintenance. Ideally implemented by a logistics company, distribution, or any retail company with a large stock of distributed item in order to ensure that normal business operations run smoothly.

The primary goal of this application is to reduce the manual effort involved in the maintenance of inventory. The manual process is tedious and it requires the staff to count the items sold and update the inventory with available stock. The products that are low in stock need to be restocked by filling out forms and sending requests to vendors. The goal of this application is to offer efficiency to the clients and reduce costs associated with managing company inventory. The client will be able to integrate their entire business and increase productivity. The end result will be happier customers and higher sales.

1.1 Document Conventions

Product - Goods maintained by the company will henceforth be called “product” throughout the document

Supplier – Dealer/Vendor supplying the manufactured goods which are sold directly or repackaged and sold to clients

Request for Proposal (RFP) – Proposal sent out to supplier to provide an updated price for the product

Purchase – Product purchased from Supplier will be referred to as a purchase

Customer – Entity buying from the company

Order – Product sold to a customer

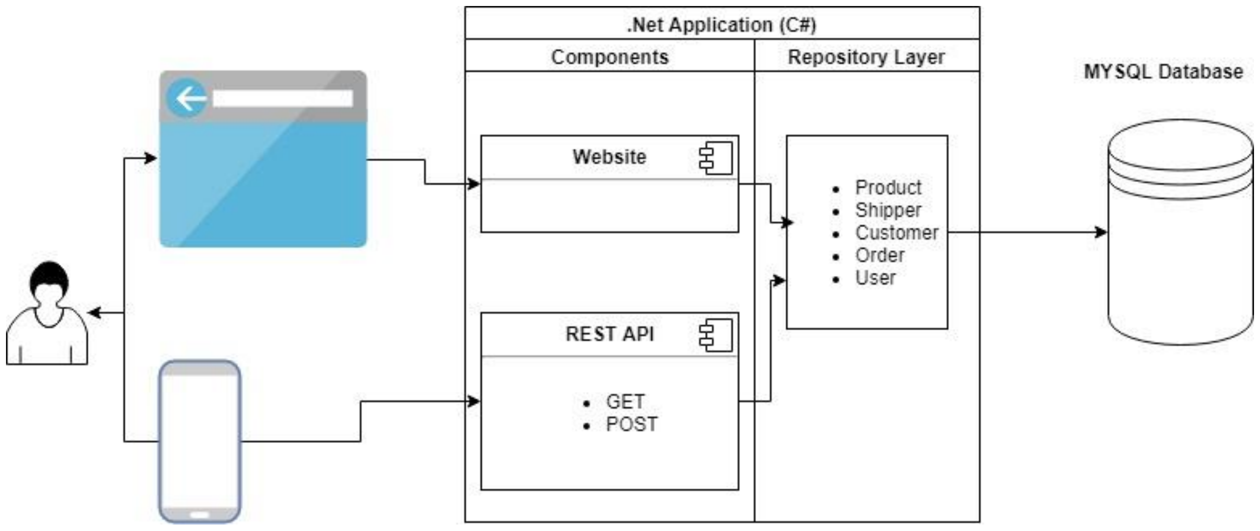
Shipper – Freighters who move the goods from supplier, ship an order to a customer

Project Scope: The proposed project is an implementation of a general inventory control system. The intent is to market the product and customize it specific to clients. The design would be flexible to allow customizations.

Manual systems involve the staff to keep track of the inventory. This involves an end of the day process of counting all the items sold and count of inventory left. This process involves too much of effort and can be better performed by using a software system.

2. Software Architecture Overview

2.1 System Architecture



2.2 General Software Components

- MY SQL Database
- Git Hub Repository

2.3 Website Development Components

The website has been built on Microsoft Asp .Net MVC framework and requires the following components for Development and Publishing

- Visual Studio 2017 Community Edition
- .Net 4.7
- MYSQL Data Connector

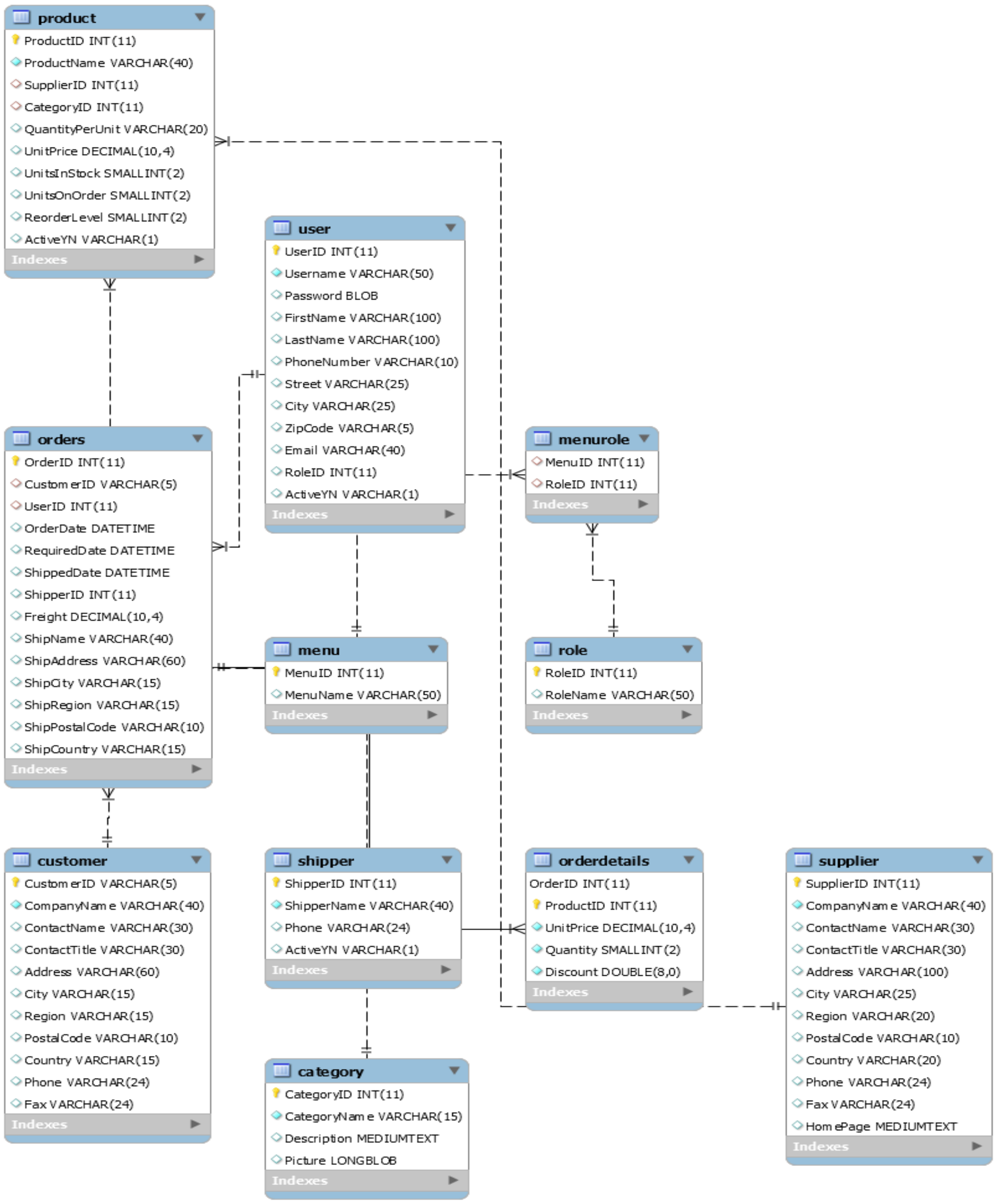
2.4 Mobile Development Components

Flutter (Google's mobile app SDK) is being used for developing native interfaces for iOS and Android.

3. Data Model

The first diagram is our ER diagram, that explains the relationships inside our MySQL database.

It shows primary keys, foreign keys, data types.



4. API

Restful API has been developed using Microsoft Web API framework. The internal architecture is to reuse the Repository Layer between Restful API and Website. This will ensure the changes to Data Model will be at a higher level of abstraction and the API is servicing the changes.

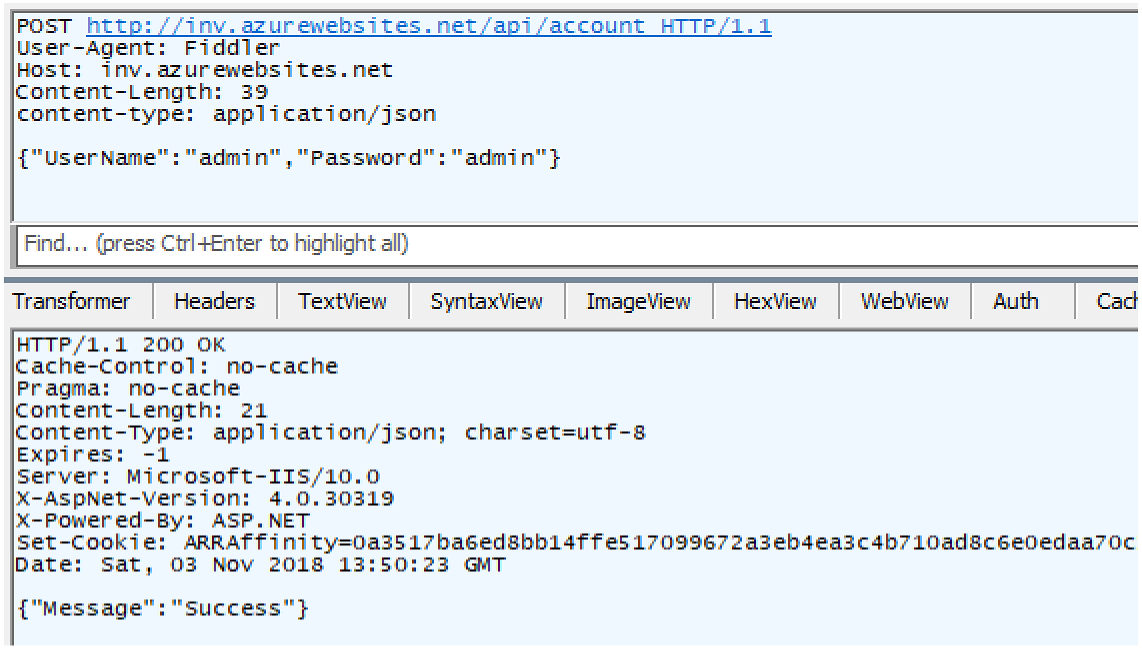
4.1 Endpoint

The endpoints are grouped together based on the functional classification

1. Account – servicing security needs, user information
2. Data – providing list of shippers, products, customers
3. Order – providing list of orders, ability to create new orders

1. Account - POST endpoint for authentication

http://inv.azurewebsites.net/api/account/validatelogin



```
POST http://inv.azurewebsites.net/api/account HTTP/1.1
User-Agent: Fiddler
Host: inv.azurewebsites.net
Content-Length: 39
content-type: application/json

{"UserName":"admin","Password":"admin"}
```

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Length: 21
Content-Type: application/json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Set-Cookie: ARRAffinity=0a3517ba6ed8bb14ffe517099672a3eb4ea3c4b710ad8c6e0edaa70c
Date: Sat, 03 Nov 2018 13:50:23 GMT

{"Message":"Success"}
```

2. Data - GET endpoints

http://inv.azurewebsites.net/api/data/shippers

http://inv.azurewebsites.net/api/data/products

http://inv.azurewebsites.net/api/data/customers

3. Order
   a. GET endpoints

      http://inv.azurewebsites.net/api/order

      returns a list of 20 orders sorted by order date descending

      http://inv.azurewebsites.net/api/order/11058

      order and order details when queried by id

   b. POST endpoint

      http://inv.azurewebsites.net/api/order

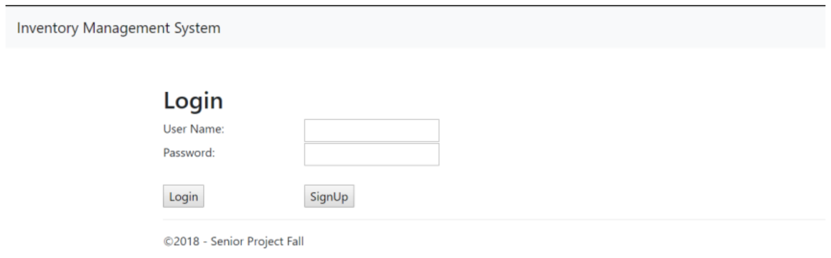
      This endpoint is not yet operational, but this is hook which accepts a post but does nothing.

5. User Interface

- Login screen
- Display current inventory
- Provide interface to order more inventory
- Voice / Language recognition for more convenient search
- Allow notifications when inventory is running low
- Basic settings page

5.1 Website

1. Login Screen



Inventory Management System

Login

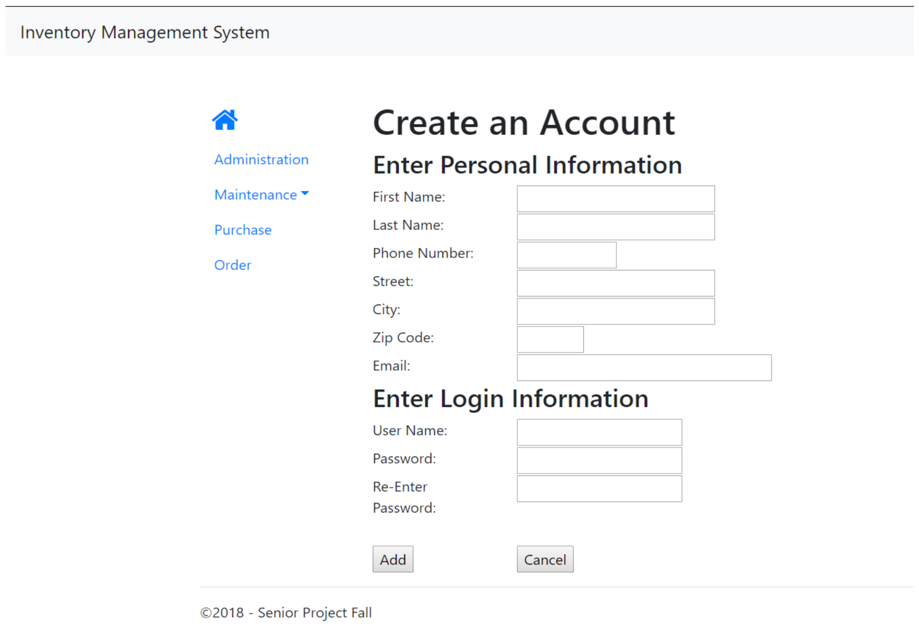
User Name:

Password:

Login SignUp

©2018 - Senior Project Fall

2. Sign Up Screen



Inventory Management System

Administration

Maintenance

Purchase

Order

Create an Account

Enter Personal Information

First Name:

Last Name:

Phone Number:

Street:

City:

Zip Code:

Email:

Enter Login Information

User Name:

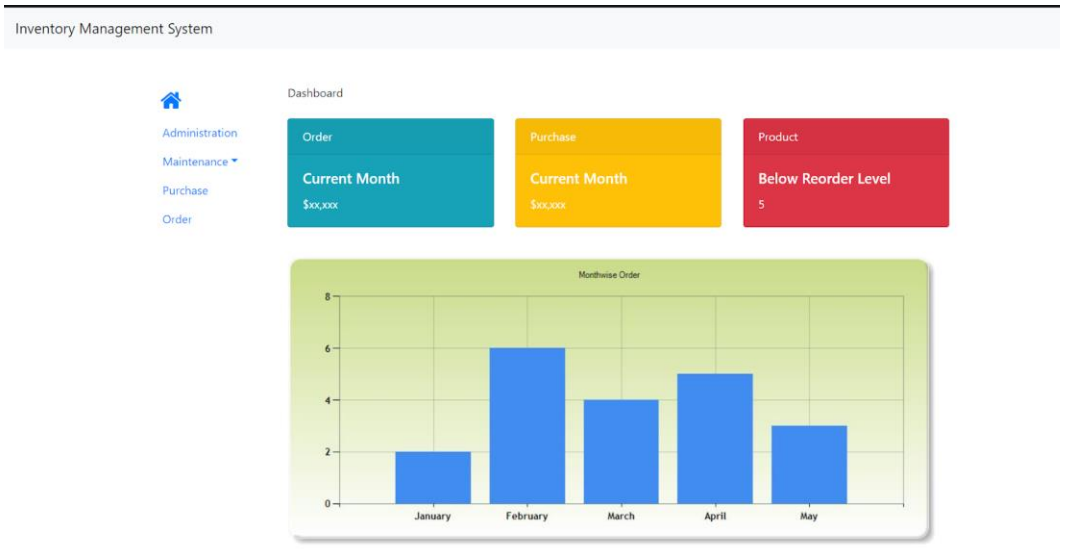
Password:

Re-Enter Password:

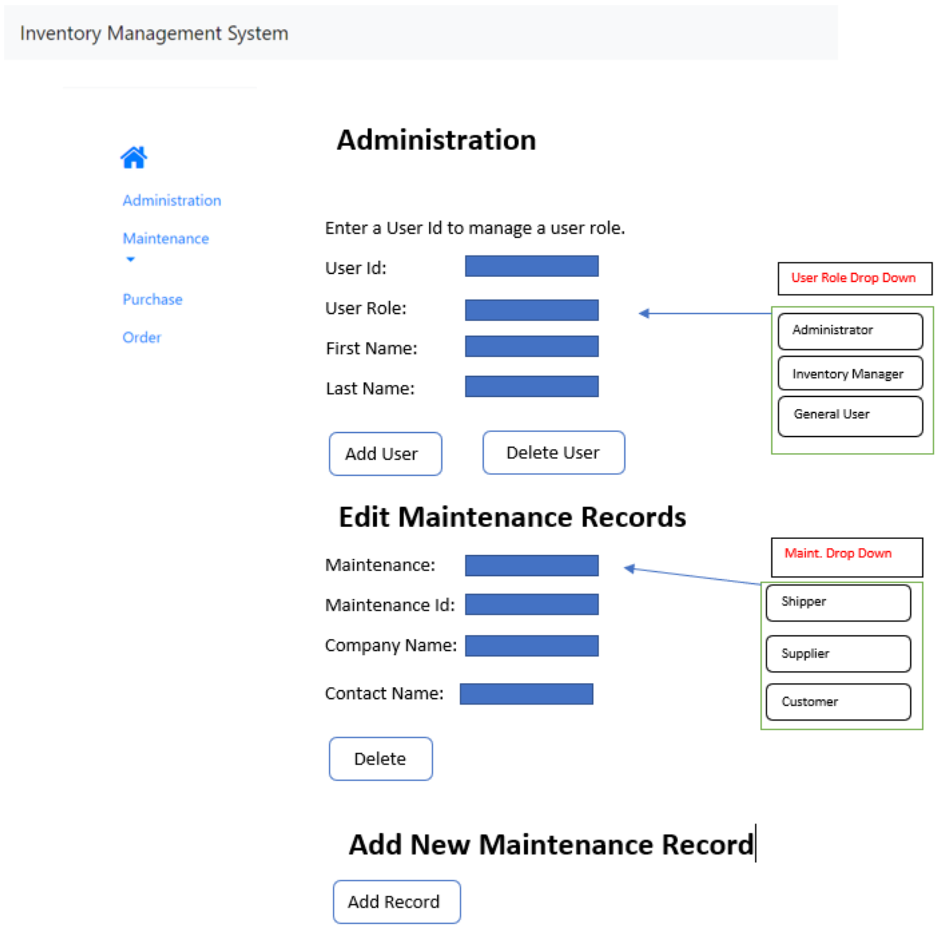
Add Cancel

©2018 - Senior Project Fall

3. Home Screen



4. Administration


Inventory Management System

Administration

Enter a User Id to manage a user role.

User Id:

User Role:

First Name:

Last Name:

Edit Maintenance Records

Maintenance:

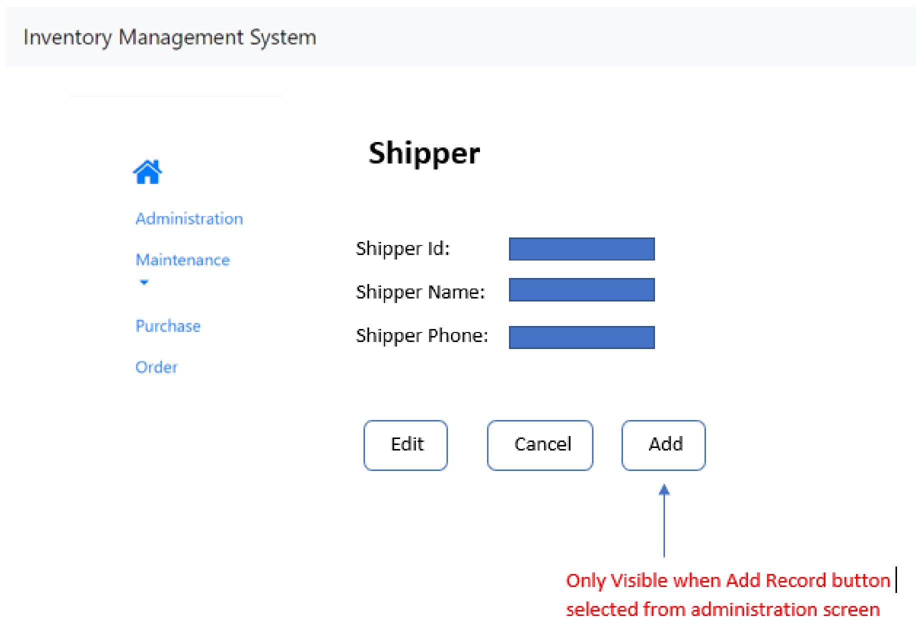
Maintenance Id:

Company Name:

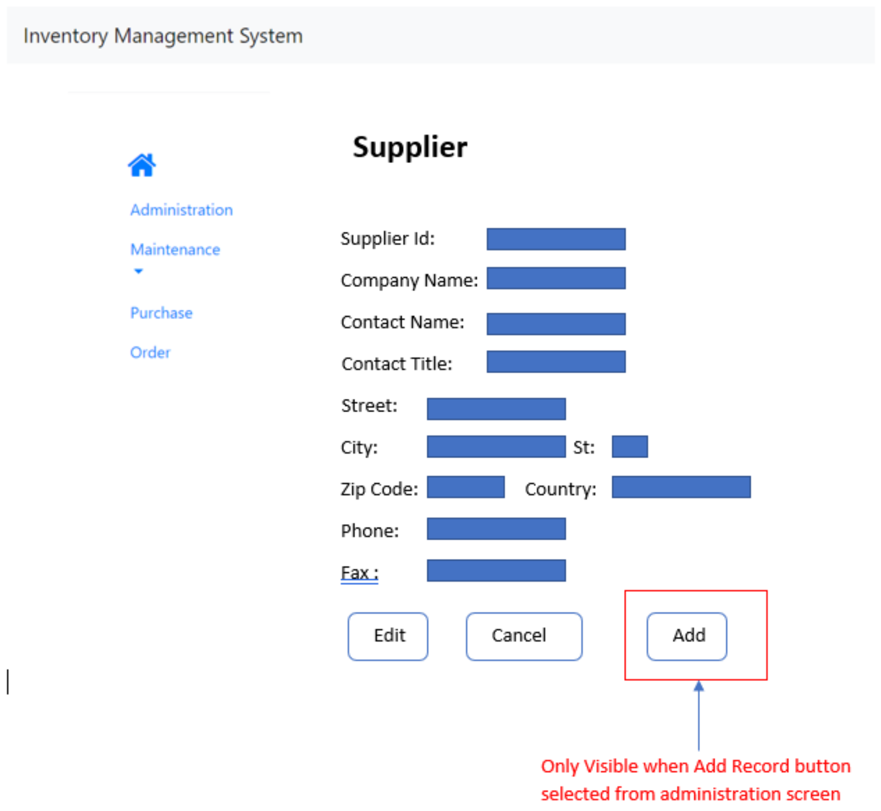
Contact Name:

Add New Maintenance Record

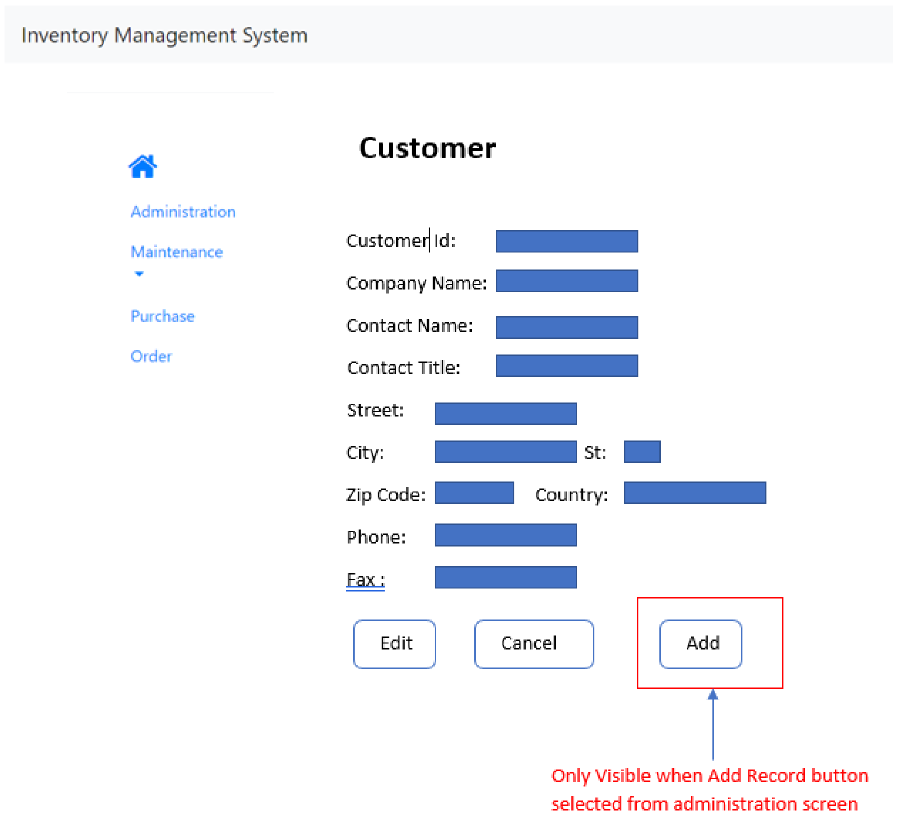
5. Shipper


Inventory Management System
Shipper
Administration
Maintenance
Purchase
Order
Shipper Id:
Shipper Name:
Shipper Phone:
Edit
Cancel
Add
Only Visible when Add Record button selected from administration screen

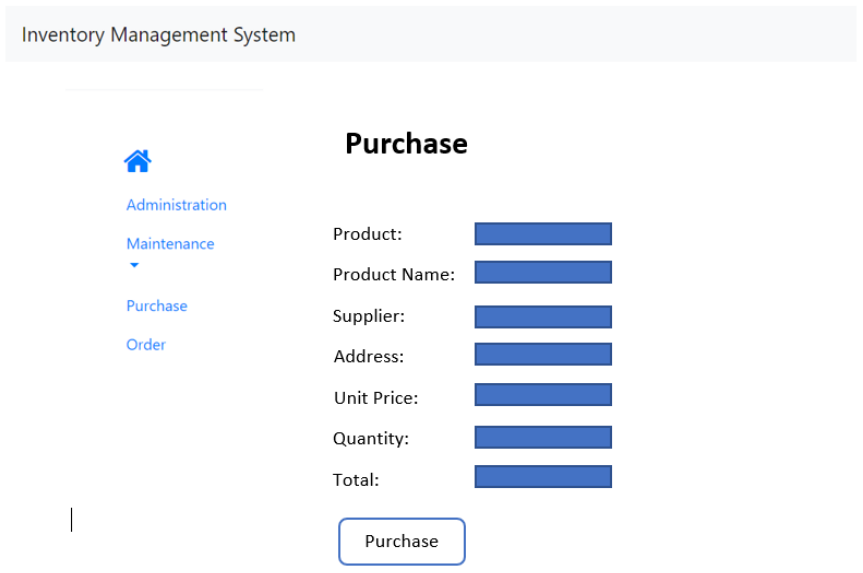
6. Supplier



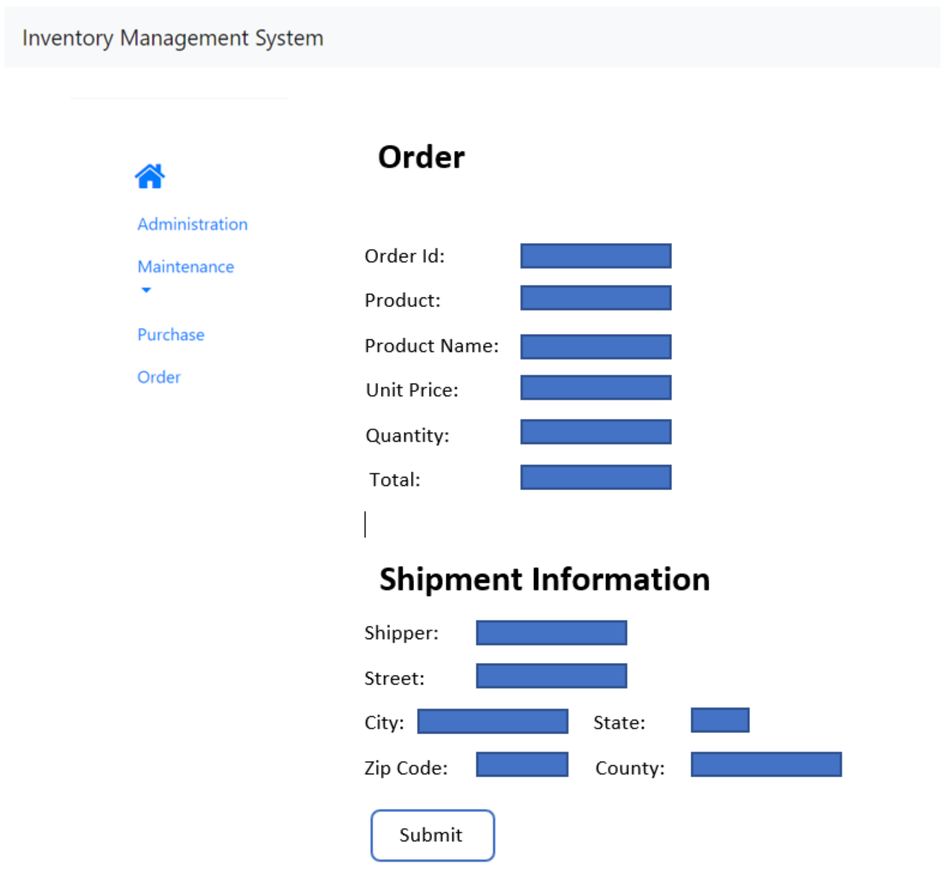
7. Customer



8. Purchase Screen



9. Order Screen



Inventory Management System

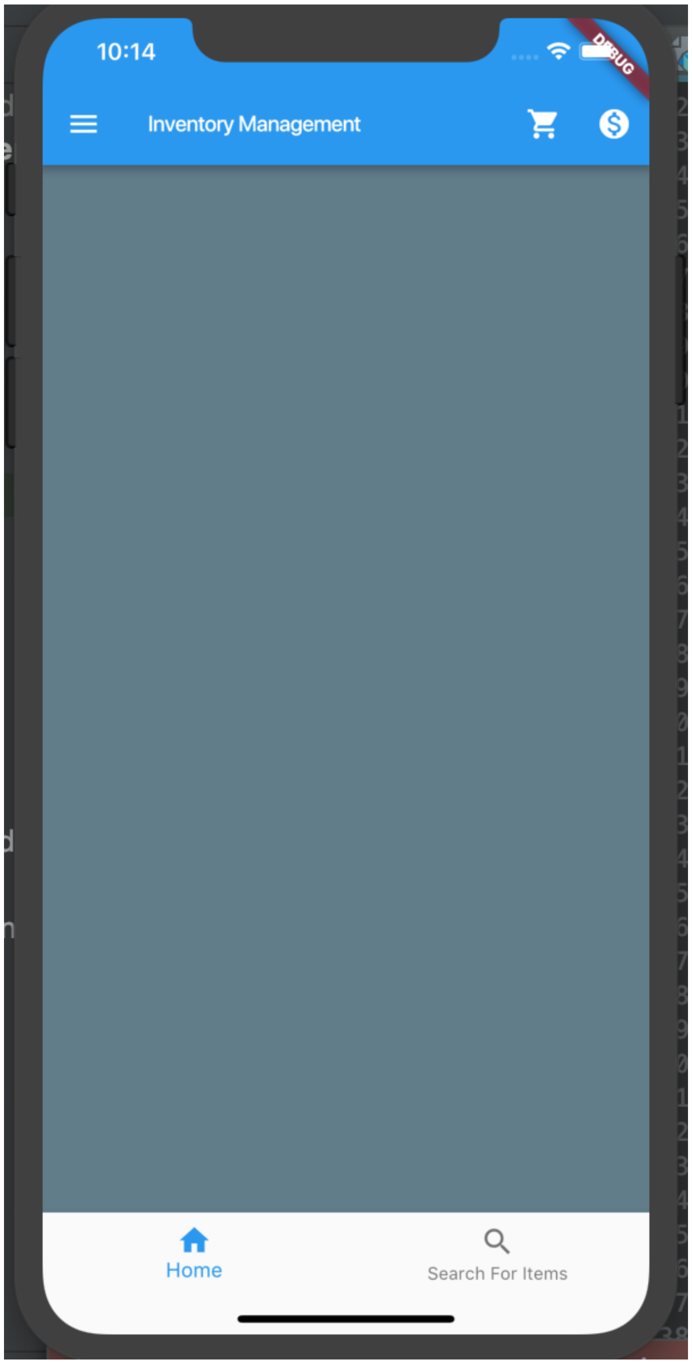
Order

Order Id:	
Product:	
Product Name:	
Unit Price:	
Quantity:	
Total:	

Shipment Information

Shipper:			
Street:			
City:		State:	
Zip Code:		County:	

5.2 Mobile Application Mockup Views



This shows the main screen of the mobile portion of this project. From here, you can search for items, check a shopping cart (orders manually added for purchase) and checkout.

6. System Features

6.1 Website

Inventory Control

Description

Show flow of items going in and out of inventory, total counts, and orders that are pending, as well as trends in purchases. Keep detailed list of items in inventory and allow for frequent changes and monitoring as well as frequent updates.

Functional Requirements

1. Snapshot of current stock in hand

2. Allow for permissions of different users to ensure inventory security using log-in.

Ordering System

Description

The ordering system is a very important part of this software system, as well as a company's main operations. The easier it is to manage orders, the better the relationship is between buyer and seller. This system will allow users functionality with inventory. This includes placing orders to restock inventory items.

Functional Requirements

1. Provide an interface to place an order. Allow placing of multiple products in one order (shopping cart) and display available quantity

2. Notifications regarding an order can be sent directly to the customer

3. Based on available inventory/processing of an order. Appropriate actions like RFP, notification to the internal user should be sent (via email, or dashboard on Inventory Management Console).

4. Multi-notification systems will be used to ensure that the user is well informed of the inventory and the traffic of transactions. This would be more beneficial then an auto ordering or auto-restocking option, because that could lead to money being wasted. For example, if an item isn't selling anymore, then next time they may want to order less of it or not that at all, if they are losing money on it.

Administration System

Description

The administration system will provide all the maintenance interfaces for keeping the system updated with latest information. Shipper, Customer, Supplier and Internal User creation / updates/ deletion. The user can utilize this part of the application to manage their inventory as needed and perform the management tasks needed to maintain and retain information.

Functional Requirements

1. Shipper Maintenance interface allows the internal user to update information, add a

new shipper and delete a shipper

2. Customer maintenance interface will allow internal user to add a new customer to update information and delete and existing customer

3. Supplier maintenance interface allows to add, update or delete a vendor and update important information related to the vendor

4. Internal User maintenance allows permission to the internal users depending on the modules authorized

6.2 Mobile Application

Description

The mobile application is designed to be an alternative to the web application, and can be installed on both Android and iOS devices. The user is able to log in and access some of the same information displayed on the web app, just in a more compact format. Think, if a stock clerk was walking around a stockroom or a store, doing manual maintenance, they would not want to carry a laptop around with them, this is where a mobile application would come in handy. That is Some functionality might be limited, as the mobile application is primarily meant to display current inventory and order more if needed. Inside are some features such as Shopping cart, where items that need to be newly supplied or resupplied can be added, and checkout can proceed, if administrator privileges are granted to the current user. There is also a search bar for searching for items to see the current status of them in stock. Keeping the app lightweight allows for a faster and simpler user experience for customer as major changes and transactions would most likely be accessed from a desktop or laptop. The app is more of a real world application to make the key information a customer would need faster than if they were viewing the website from their phone. The mobile application's UI will be user friendly and easy to navigate. The UI will be simple but very effective in completing user/customer tasks. The mobile application will be highly responsive and complete tasks requested at a high performance.