Software Design   
Specification

for

Purchase Requisition System

**Version <1.0>**

**Group No.: 1**

|  |  |  |
| --- | --- | --- |
| **Abdulrahman Yousef** | **1171300042** |  |
| **Gabriel Lim Kee Chai** | **1141127848** |  |
| **Yousef Ashraf Hafez** | **1161302668** |  |
| **Hossain, Mohammad Mubdiul** | **1161303847** |  |
| **Gani Mohammed Osman** | **1171300039** |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
| **Date:** | **5/5/2019** |

**Contents**

**CONTENTS 2**

**REVISIONS 3**

**1** **SYSTEM OVERVIEW 4**

1.1 Description 4

1.2 Actors 4

1.3 Assumptions and Dependencies 4

1.4 Use Case Diagram 4

**2** **DATA DESIGN 5**

2.1 Data Dictionary 5

2.2 Data Structures 5

**3** **ARCHITECTURE DESIGN 6**

3.1 Software Architecture 6

**4** **INTERFACE DESIGN 7**

4.1 Main Screens 7

4.2 Subsystem 1 Screens 7

4.3 Subsystem 2 Screens 7

**5** **COMPONENT DESIGN 8**

5.1 Main Components 8

**6** **DEPLOYMENT DESIGN 8**

6.1 Deployment Diagram 8

**Revisions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| 1.0 | Gabriel Lim  Abdulrahman Yousef  Yousef Hafez  Mohammad Mubdiul Hossain  Gani Mohammed Osman | Identification and description of System Overview, involving Actors and use cases and assumption and dependencies of the system. Data design, Architecture design, Interface design, Component design and deployment design is also identified and added into this documentation. | 5/5/2019 |

# System Overview

## Description

*This system is a Purchase Requisition System. There are 4 actors involved in the system namely the Staff, the Finance Staff, the Manager and the Vendor. The Staff will use the system to submit Purchase Requisition (PR) and will be able to track the status of each PR submission and view the PR submitted. The Manager is responsible for reviewing the PR submissions, either approving or denying after viewing them. The Finance Staff is able to Create Request for Quotation (RFQ) based on the approved PR, Create Purchase Order (PO) based on Quotation received, Create Delivery Order (DO) based on the hard-copy DO that will be sent with products and goods ordered, Create Invoice based on the hard-copy invoice as well and lastly Make Payment to the vendors based on the saved Invoice. The Vendor will be able to View RFQ, Create Quotation based on RFQ received and View PO. This system will be a web-based system and a database will be integrated in order to store and retrieve the data.*

## Actors

Staff:

1. create PR
2. View PR
3. create item

Manager:

1. View Pending PR
2. Approve/Deny PR submission

Finance Staff:

1. View Approved PR
2. Submit Request for Quotation (RFQ)
3. View Quotation
4. Create Purchase Order (PO)
5. Create Delivery Order (DO)
6. View DO
7. Create Invoice
8. View Invoice
9. Make Payment

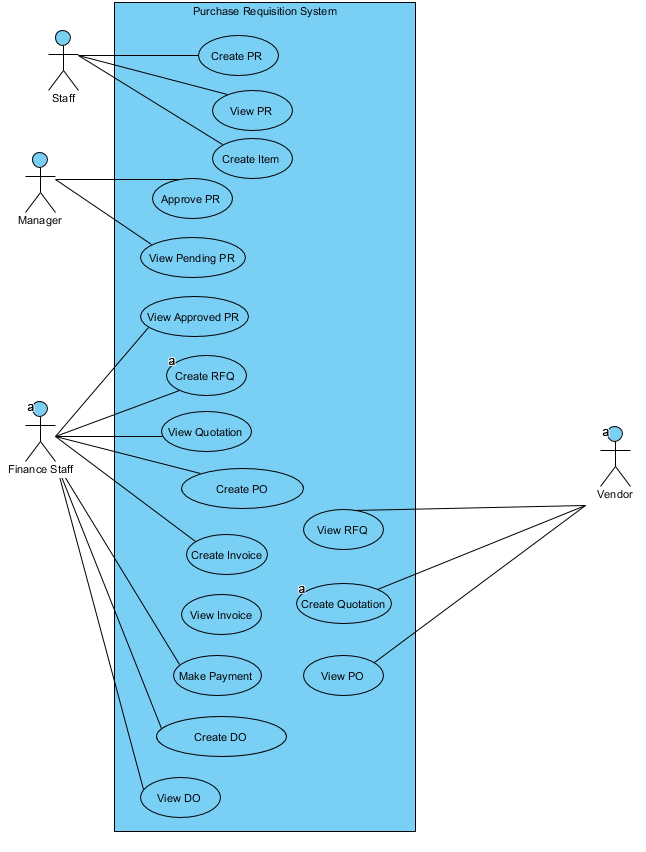
Vendor:

1. Submit Quotation
2. View RFQ
3. View PO

## Assumptions and Dependencies

* The system is assumed to be used inside Malaysia only, and that all actors are inside Malaysia.
* The system is assumed to work when accessed parallelly form all actors.
* It is assumed that there are available stocks for products and goods with the vendor.
* It is assumed that no tax will be charged for the products and goods ordered.
* It is assumed that the system will be dependent on a single database.
* It is assumed that the database has infinite storage.
* It is assumed that the vendors available in the system are registered with the buying company
* It is assumed that all deliveries will be successful.
* It is assumed that all RFQ will be responded by at least one vendor.

## Use Case Diagram

**

# Data Design

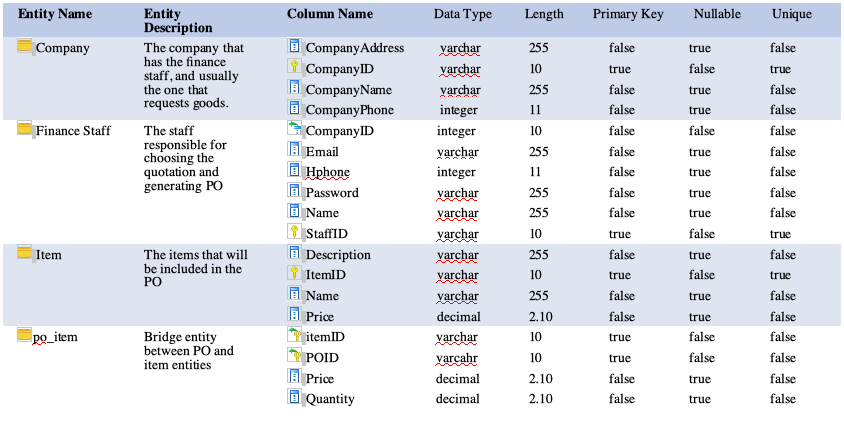
## Data Dictionary

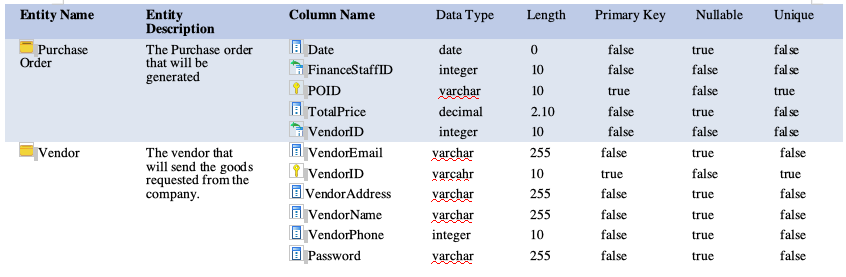
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Table* | *attribute* | *data type* | *format* | *required* | *pk/fk* | *fk reference table* | *Description* |
| *PR* | *prID* | *varchar(10)* | *XXXX* | *yes* | *pk* |  | *PR ID number* |
| *date* | *Date* | *1234/12/31* | *yes* |  |  | *Submission Date* |
| *status* | *varchar(10)* | *XXXX* | *yes* |  |  | *Current status* |
| *staffID* | *varchar(10)* | *XXXX* | *yes* | *fk* | *staff* | *ID for the staff who submitted it* |
| *managerID* | *varchar(10)* | *XXXX* |  | *fk* | *manager* | *Approver ID* |
| *reason* | *varchar(60)* | *XXXX* |  |  |  | *Reason for rejection* |
| *staff* | *staffID* | *varchar(10)* | *XXXX* | *yes* | *pk* |  | *Staff ID* |
| *name* | *varchar(10)* | *XXXX* | *yes* |  |  | *Staff name* |
| *manager* | *managerID* | *varchar(10)* | *XXXX* | *yes* | *pk* |  | *Manager id* |
| *name* | *varchar(10)* | *XXXX* | *yes* |  |  | *Manager name* |
| *item* | *itemID* | *varchar(10)* | *XXXX* | *yes* | *pk* |  | *Item id* |
| *name* | *varchar(10)* | *XXXX* | *yes* |  |  | *Item name* |
| *price* | *decimal* | *xx.xx* |  |  |  | *price* |
| *description* | *varchar(60)* | *xxxx* |  |  |  | *Brief description* |
| *pr\_item* | *prID* | *varchar(10)* | *XXXX* | *yes* | *pk,fk1* | *PR* | *PR id* |
| *item ID* | *varchar(10)* | *XXXX* | *yes* | *pk,fk2* | *item* | *Item id* |
| *quantity* | *integer* | *123456* | *yes* |  |  | *Number of quantity* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Entity Name** | **Column Name** | **Data Type** | **Length** | **PK** | **Nullable** | **FK** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Delivery\_Order** | DO\_ID | char | 10 | true | false |  |
|  | Finance\_StaffID | integer | 10 | false | false | Finance\_Staff |
|  | Date | date | 0 | false | false |  |
|  | subTotal | decimal | 2,10 | false | true |  |
|  | POID | char | 10 | false | false | Purcahse\_Order |
|  | VendorID | char | 10 | false | false | Vendor |
|  | ShippingAgency | varchar | 255 | false | true |  |
|  | DeliveryFee | decimal | 2,10 | false | true |  |
|  | Description | varchar | 255 | false | true |  |
| **DO\_Item** | DO\_ID | char | 10 | true | false | Delivery\_Order |
|  | ItemID | char | 10 | true | false | Item |
|  | Quantity | integer | 10 | false | true |  |
|  | subTotal | decimal | 2,10 | false | true |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity | Entity Description | Attribute | Datatype(Length) | Primary key | Foreign Key | Unique |
| Company | The company will act as a buyer in the system | Company\_Id  Company\_Name  Company\_Address  Company\_Phone | Varchar(250)  Varchar(250)  Varchar(250)  Varchar(250) | Yes  No  No  No | No  No  No  No | Yes  No  No  No |
| Item | This table will store Item details | Item\_Id  Item\_Name  Item\_Price  Item\_description | Varchar(250)  Varchar(250)  Decimal(11)  Varchar(250) | Yes  No  No  No | No  No  No  No | Yes  No  No  No |
| Invoice | This table  Will store all the data related to invoice | Invoice\_Id  Date  Company\_Id  Vendor\_Id  Staff\_id  Total\_Price | Varchar(250)  Date  Varchar(250)  Varchar(250)  Varchar(250)  Decimal(11) | Yes  No  No  No  No  No | No  No  Yes  Yes  Yes  No | Yes  No  No  No  No  No |
| Invoice\_Item | This table will store all the item details that would be in invoice. | Invoice\_id  Item\_id  Item\_quantity | Varchar(250)  Varchar(250)  Integer(11) | Yes  Yes  No | Yes  Yes  No | No  No  No |
| Payment | This table will store the Payment details for every Invoice | Payment\_Id  Account\_Number  Account\_Name  Date  Invoice\_Id  Amount | Varchar(250)  Varchar(250)  Varchar(250)  Date  Varchar(250)  Decimal(11) | Yes  No  No  No  No  No | No  No  No  No  Yes  No | Yes  No  No  No  Yes  No |
| Finance\_Staff | This table will store the information of Fianance\_Staff | Staff\_Id  Staff\_Name  Staff\_Phone | Varchar(250)  Varchar(250)  Varchar(250) | Yes  No  No | No  No  No | Yes  No  No |

**

**

## Data Structures

### *2.2.1 Map data structure in any class with items and quantity*

*This data structure is ideal for our situation to facilitate having items with its quantity at all times under the attribute itemsAndQuantity.*

|  |  |
| --- | --- |
| *Item (object)* | *Quantity (int)* |
| *Keyboard* | *3* |
| *Mouse* | *6* |
| *PC* | *1* |

***2.2.2* Finance Staff**

Finance staff is stored in an Arraylist array to hold the financeStaff object,this object will be fetched from the finance staff table in the database.

|  |  |
| --- | --- |
| Index | Finance staff Object |
| 1 | Natalie Holt |
| 2 | Kameko Reid |
| 3 | Kendall Craft |
| 4 | Taylor Christian |

ArrayList<Finance\_Staff> fs =new Array\_List<Finance\_Staff>();

### 2.2.3 Delivery\_Order

Delivery Order list is stored in an Arraylist of Delivery\_Order instances, that will get objects from the DO\_item table and DO table.

|  |  |
| --- | --- |
| Index | DO Object |
| 1 | DO0001 |
| 2 | DO0002 |
| 3 | DO0003 |
| 4 | DO0004 |

ArrayList<Delivery\_Order> D = new ArrayList<Delivery\_Order>();

### 2.2.4 Vendor

Vendors are stored in an Arraylist of Vendor instances, it will gets its instance from the Vendor table.

|  |  |
| --- | --- |
| Index | Vendor Object |
| 1 | Eliana Mcgowan |
| 2 | Rama Bright |
| 3 | Alana Kline |

ArrayList<Vendor> v = new ArrayList<Vendor>();

### 2.2.5 Items for DO

Items and their quantities are specified as *Map* data structure, specifically *Map<Item,int>.* The purpose of this is to associate each Item object with it’s own quantity.

example:

Map<Item,int> items = new Map<Item, int>();

|  |  |  |
| --- | --- | --- |
| Index | Item Object (Key) | Quantity (Value) |
| 1 | Keybaord | 2 |
| 2 | PC | 2 |
| 3 | Mouse Pad | 9 |
| 4 | Remote | 1 |

### 2.2.6 Purchase Order

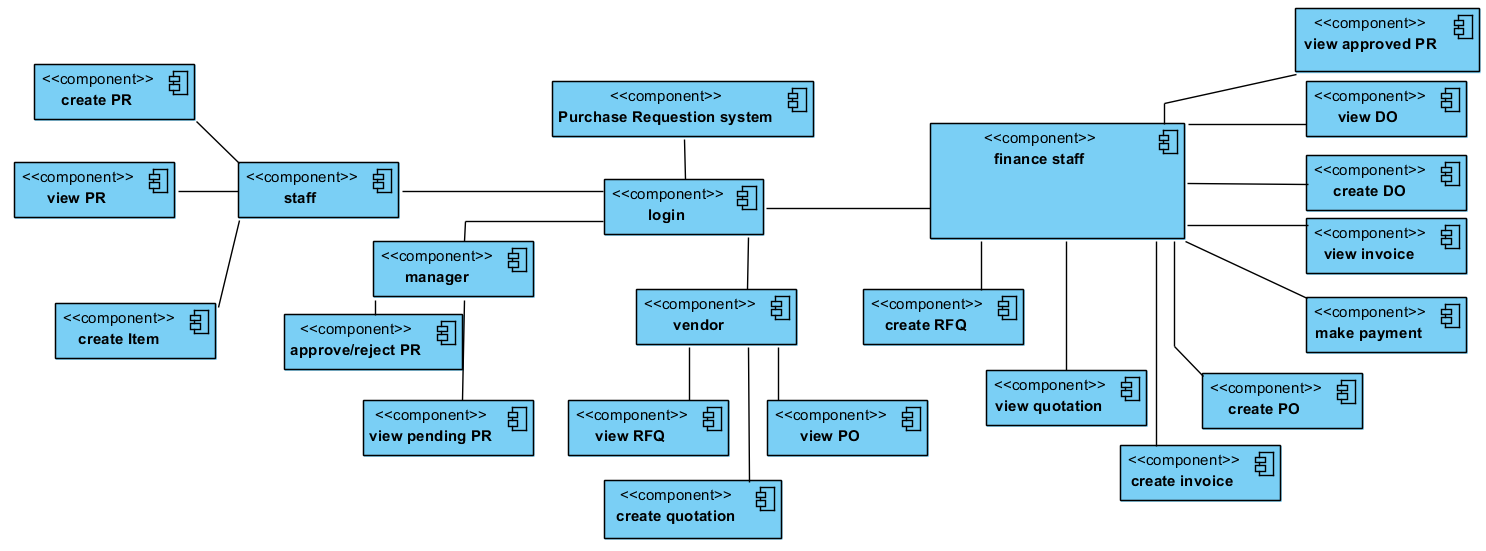
The Purchase Orders will be stored in an Arraylist of Purchase order instances, that will get it’s information from the PO\_item table and PO table. For demonstration the id Property of the Purchase Object will be used to represent the object.

|  |  |
| --- | --- |
| Index | PO Object |
| 1 | PO0001 |
| 2 | PO0002 |
| 3 | PO0003 |
| 4 | PO0004 |

*ArrayList<PO> f = {new PO(), new PO()};*

# Architecture Design

## Software Architecture



*The Purchase Requisition System starts with the first subsystem. The login. After anyone of the four actors logs inside the system, his/her respective subsystem will begin to run.*

*We decided to separate the system into 5 subsystems.*

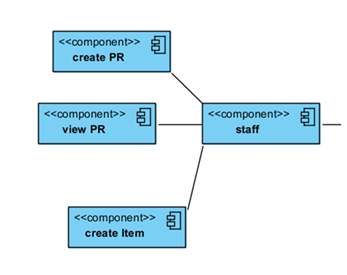
|  |  |
| --- | --- |
| *Subsystem name* | *Responsibility* |
| *Login* | *group* |
| *staff* | *Abdulrahman* |
| *manager* | *Abdulrahman* |
| *Vendor* | *Yousef / Gabriel / Mubdiul / Gani* |
| *Finance staff* | *Yousef / Gabriel / Mubdiul / Gani* |

### login subsystem

*its a basic login module. The actor provides id and password and it will verify and direct the actor to the appropriate subsystem according.*

### staff subsystem

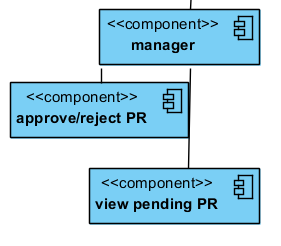
*This part represents the first actor “Staff” as everything under it is considered all of the staff’s option in our system.*

**

## 

## manager subsystem:

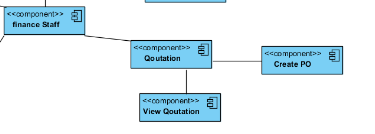
This part represents the first actor “manager” as everything under it is considered all of the staff’s option in our system.



## 

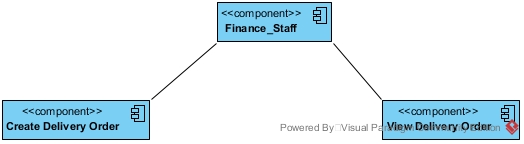
## Finance Staff subsystem - Purchase Order:

*This part represents the first actor “Finance Staff” as everything under it is considered one of the Finance staff’s option in our system.*

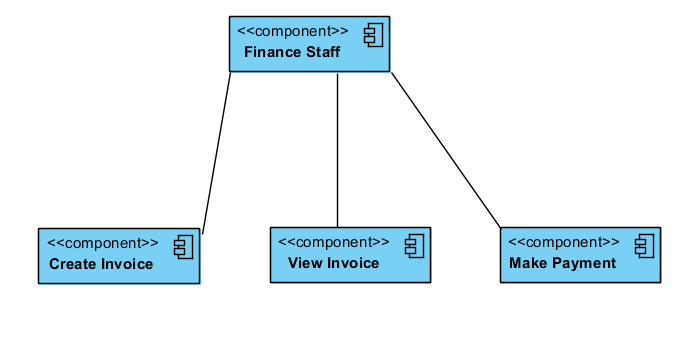


**Finance Staff Subsystem - Delivery Order**

This component diagram represents after logging as Finance staff, he/she can will have two options to select. One is creating the delivery order and the other one is showing the delivery orders.

****

**Finance Staff Subsystem - Invoice and payment**

****

After login in the system by the Finance staff. He /she can see Invoice option.

If he/she can click on the invoice, new window will open with this three option(Create invoice, View invoice, Make Payment) shows in above diagram.

## 

# Interface Design

## Main Screens

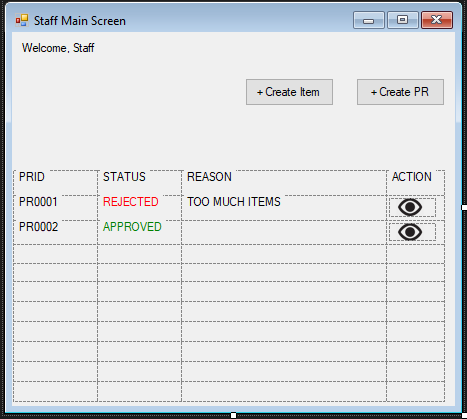
### Login Screen

*The login screen for the Purchase Requisition System. User fill in the Username and Password and press “Login” to access the system.*



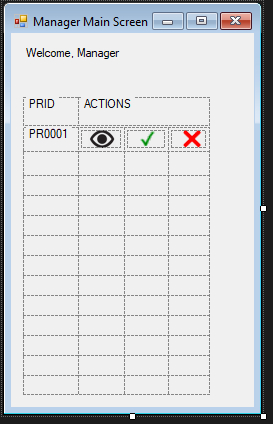
### Staff Main Screen

This is the main screen for Staff. On login, they will be able to view the status of the PRs they submitted. They could choose to view the PR, Create PR or Create Item if the item they need is an item that was never requested in the system.



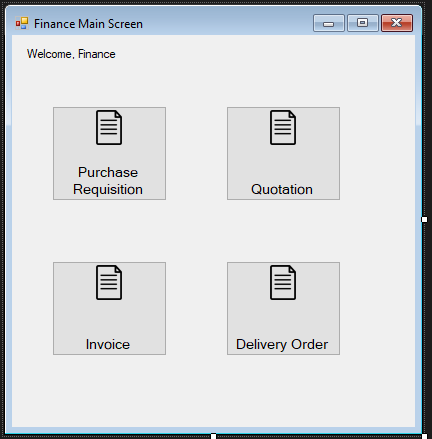
### Manager Main Screen

This is the main screen for Manager. On login, they will be able to view the PRs submitted by Staffs. They could choose to view the PR, approve or reject the PR.



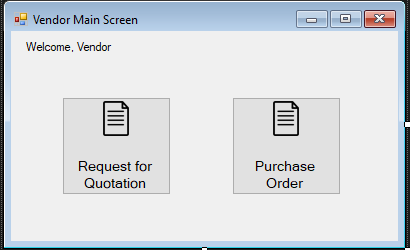
### Finance Main Screen

This is the main screen for Finance Staff. On login, there are 4 options for them, Purchase Requisition, Quotation, Invoice or Delivery Order. Purchase Requisition is for viewing approved PR and create RFQ. Quotation is for viewing quotations sent by Vendor and create PO. Invoice is for viewing invoices, creating invoice or make payment. Delivery Order is for viewing delivery order or create DO.



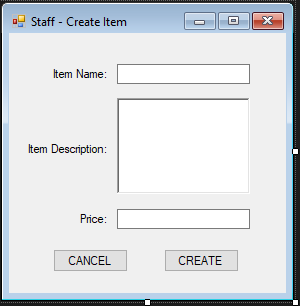
### Vendor Main Screen

This is the main screen for Vendor. On login, there are 2 options for them, Request for Quotation or Purchase Order. Request for Quotation is for viewing RFQ by buying company and Creating Quotation while Purchase Order is for viewing PO sent by buying company.



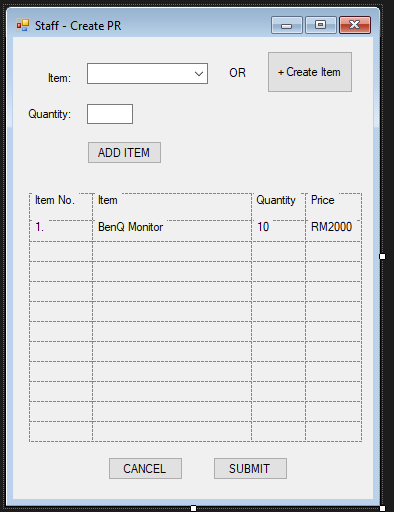
## Subsystem 1 Screen (Staff - Create Item)

This is the screen for Staff to create new item. Staff fill in item name, item description and price for the item and press “Create” to create new item. Alternatively, they can press “Cancel” to return to main screen.

**

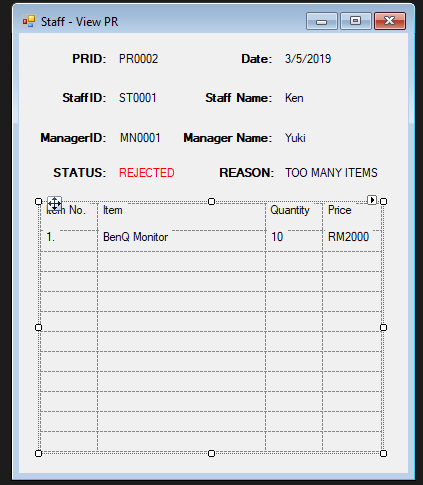
## Subsystem 2 Screen (Staff - Create PR)

This is the screen for staff to create PR. Staff add item to the PR by choosing from a dropdown list that shows available item in the system. If the item is new, the staff can choose to press “Create Item” which brings him/her to the item creation page. After adding all desired item, the staff press “Submit” to submit the PR form to Manager for review. Alternatively, the staff can press “Cancel” to be brought back to the main screen.



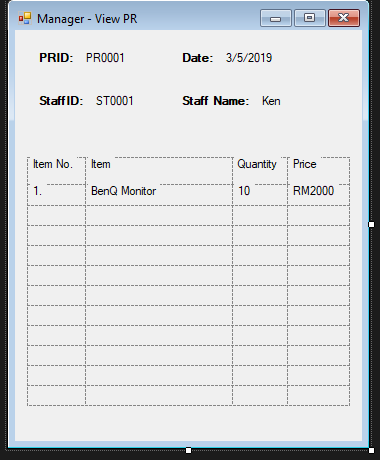
## Subsystem 3 Screen (Staff - View PR)

This is the view PR screen for Staff. After the Staff click on the “Eye” a.k.a View button of a PR on the main screen, he/she will be brought here. The PR form consist of PRID, Date, StaffID, Staff Name, ManagerID, Manager Name, Status, Reason and Items details.



## Subsystem 4 Screen (Manager - View PR)

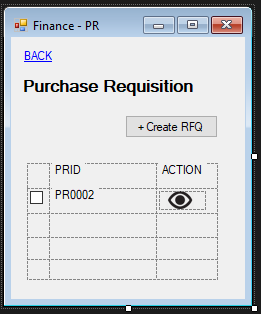
This is the view PR screen for Manager. After the Manager click on the “Eye” a.k.a View button of a PR on the main screen, he/she will be brought here. The PR form consist of PRID, Date, StaffID, Staff Name and Items details.



## Subsystem 5 Screens (Finance - Create RFQ)

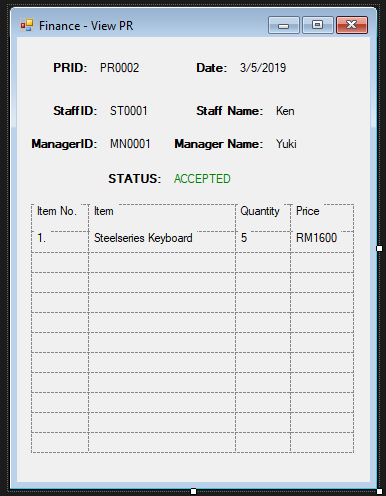
### PR List Screen

This is the PR List screen for Finance Staff. After Finance Staff click on the “Purchase Requisition” button from the main screen, he/she will be brought here. A list of approved PRs will be shown to the Finance Staff. He/She will be able to choose either to View the approved PRs or Create RFQ.



### View PR Screen

This is the view PR screen for Finance Staff. After the Finance Staff click on the “Eye” a.k.a View button of a PR on the PR List screen, he/she will be brought here. The PR form consist of PRID, Date, StaffID, Staff Name, ManagerID, Manager Name, Status and Items details.



### Create RFQ Screen

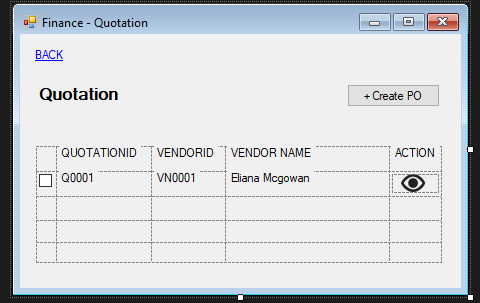
*This is the create RFQ screen for Finance Staff. After the Finance Staff ticks a PR and press Create RFQ, he/she will be brought here. The RFQ form consists of company name, company address, date, finance staff name, finance staff email, finance staff phone and item details. The Finance Staff just have to fill in the due date for the RFQ and press “Submit” to send the RFQ to the vendors. Alternatively, the Finance Staff can press “Cancel” to be brought back to the PR List screen.*



## Subsystem 6 Screens (Finance – Create PO)

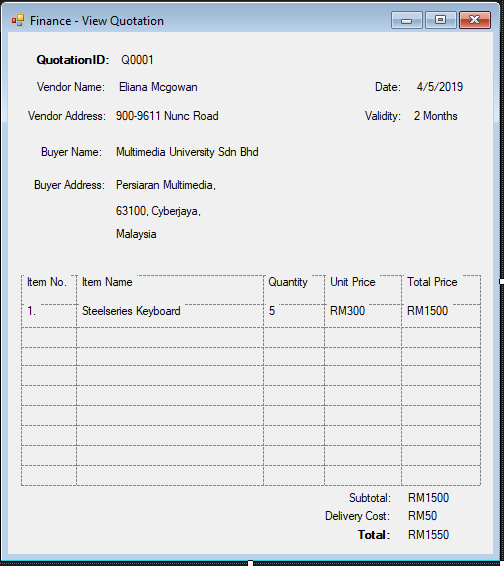
### Quotation List Screen

This is the Quotation List screen for Finance Staff. After Finance Staff click on the “Quotation” button from the main screen, he/she will be brought here. A list of quotations will be shown to the Finance Staff. He/She will be able to choose either to View the quotations or Create PO.



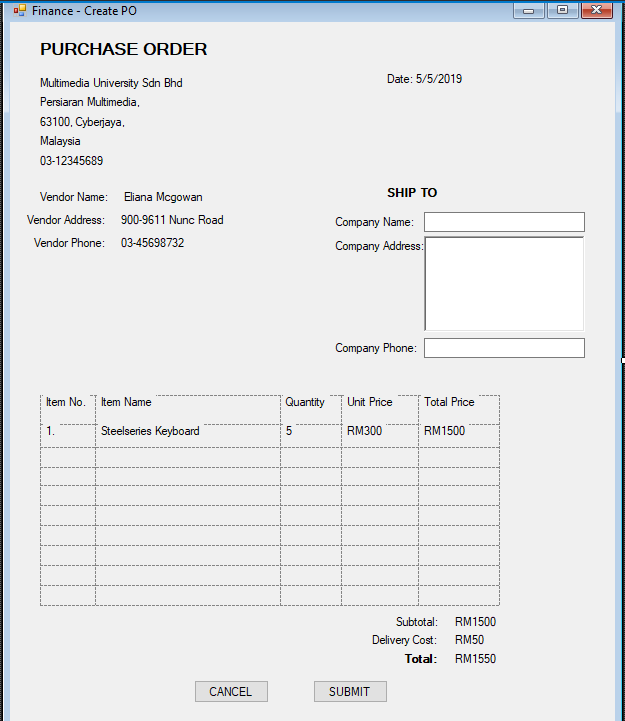
### View Quotation Screen

This is the View Quotation screen for Finance Staff. After the Finance Staff click on the “Eye” a.k.a View button of a quotation on the Quotation List screen, he/she will be brought here. The Quotation form consist of QuotationID, Date, Validity, Vendor Name, Vendor Address, Buyer Name, Buyer Address, Items and Price details.



### Create PO Screen

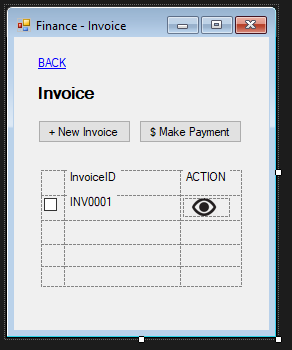
*This is the create PO screen for Finance Staff. After the Finance Staff ticks a Quotation and press Create PO, he/she will be brought here. The PO form consists of company name, company address, company phone, date, vendor name, vendor address, vendor phone, item and price details. The Finance Staff just have to fill in the SHIP TO details and press “Submit” to send the PO to the vendors. Alternatively, the Finance Staff can press “Cancel” to be brought back to the Quotation List screen.*



## Subsystem 7 Screens (Finance – Create Invoice)

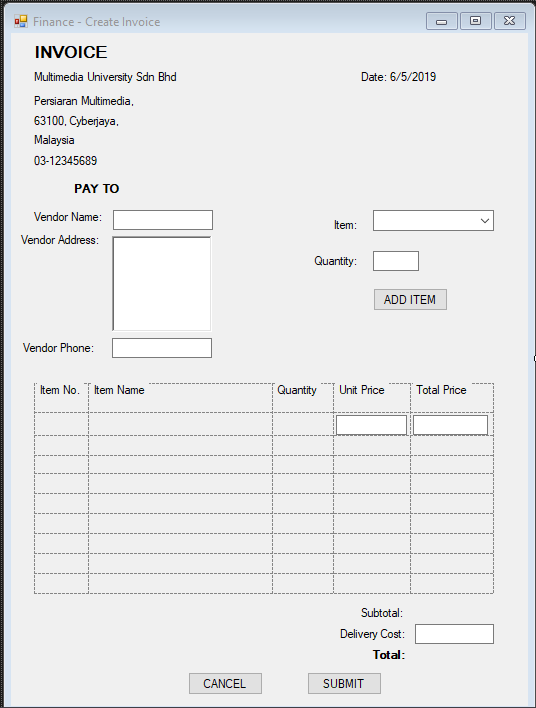
### Invoice List Screen

This is the Invoice List screen for Finance Staff. After Finance Staff click on the “Invoice” button from the main screen, he/she will be brought here. A list of invoices will be shown to the Finance Staff. He/She will be able to choose either to View the invoices , Create Invoice or Make Payment.



### Create Invoice Screen

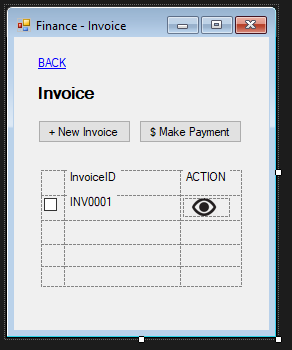
*This is the create Invoice screen for Finance Staff. After the Finance Staff press “New Invoice”, he/she will be brought here. The Invoice form consists of company name, company address, company phone and date details. The Finance Staff have to fill in the PAY TO details and add items to the form and press “Submit” to save the invoice in the system. Alternatively, the Finance Staff can press “Cancel” to be brought back to the Invoice List screen.*

**

## Subsystem 8 Screens (Finance – Make Payment)

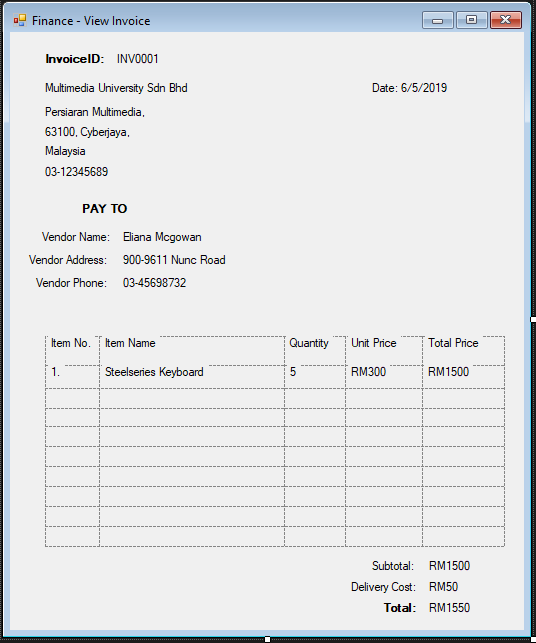
### Invoice List Screen

This is the Invoice List screen for Finance Staff. After Finance Staff click on the “Invoice” button from the main screen, he/she will be brought here. A list of invoices will be shown to the Finance Staff. He/She will be able to choose either to View the invoices , Create Invoice or Make Payment.



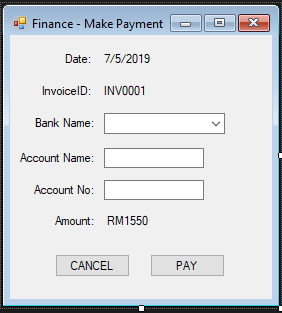
### View Invoice Screen

This is the View Invoice screen for Finance Staff. After the Finance Staff click on the “Eye” a.k.a View button of an invoice on the Invoice List screen, he/she will be brought here. The Invoice form consist of InvoiceID, Date, Company Name, Company Address, Company Phone, Vendor Name, Vendor Address, Vendor Phone, Items and Price details.



### Payment Screen

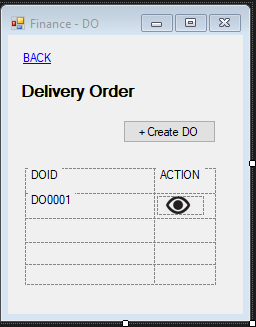
*This is the Make Payment screen for Finance Staff. After the Finance Staff ticks an Invoice and press “Make Payment”, he/she will be brought here. The Payment form consists of date, invoiceID and amount details. The Finance Staff have to fill in the Bank Name, Account Name and Account No and press “PAY” to make payment to the Vendor. Alternatively, the Finance Staff can press “Cancel” to be brought back to the Invoice List screen.*



## Subsystem 9 Screens (Finance – Create DO)

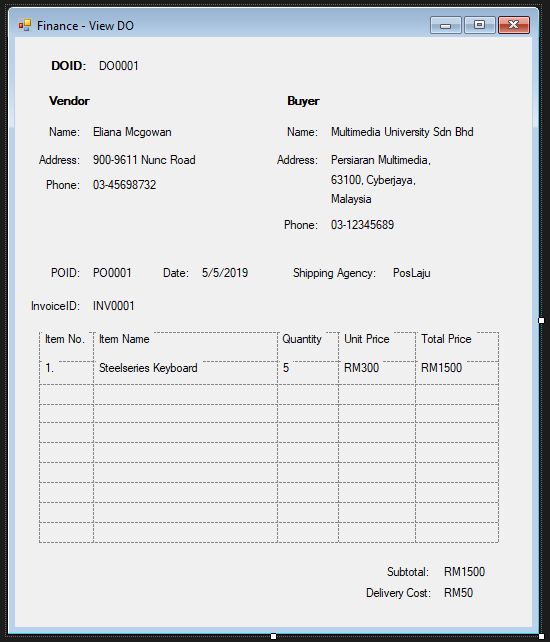
### DO List Screen

This is the DO List screen for Finance Staff. After Finance Staff click on the “Delivery Order” button from the main screen, he/she will be brought here. A list of DOs will be shown to the Finance Staff. He/She will be able to choose either to View the DOs or Create DO.



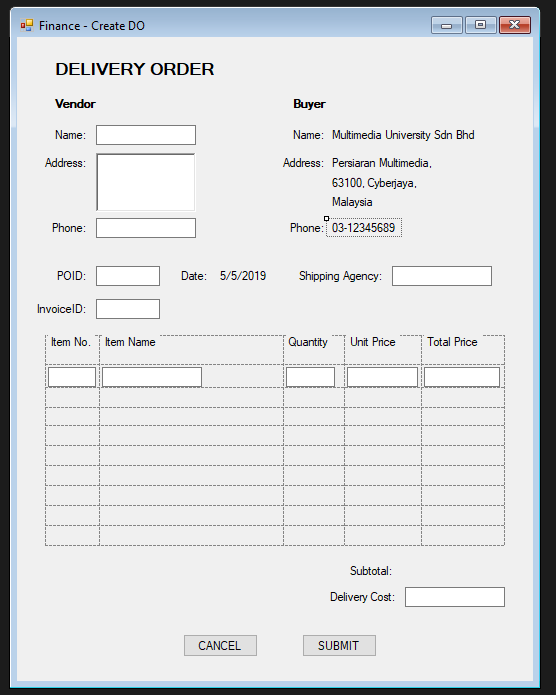
### View DO Screen

This is the View DO screen for Finance Staff. After the Finance Staff click on the “Eye” a.k.a View button of a DO on the DO List screen, he/she will be brought here. The DO form consist of DOID, Vendor Name, Vendor Address, Vendor Phone, Company Name, Company Address, Company Phone, POID, Date, Shipping Agency, InvoiceID, Items and Price details.



### Create DO Screen

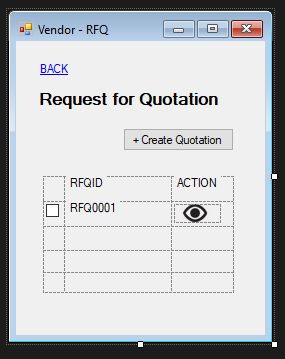
*This is the create DO screen for Finance Staff. After the Finance Staff press “Create DO”, he/she will be brought here. The DO form consists of company name, company address, company phone and date details. The Finance Staff have to fill in the Vendor, POID, Shipping Agency, InvoiceID, item, price and delivery cost details to the form and press “Submit” to save the DO in the system. Alternatively, the Finance Staff can press “Cancel” to be brought back to the DO List screen.*

**

## Subsystem 10 Screens (Vendor – Create Quotation)

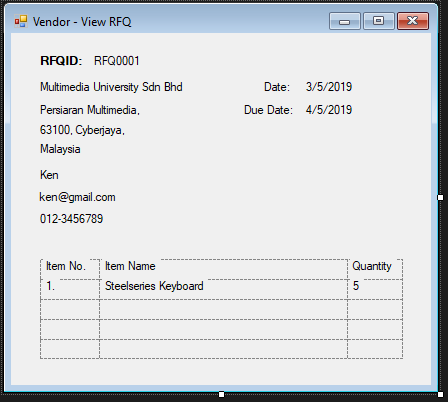
### RFQ List Screen

This is the RFQ List screen for Vendor. After Finance Staff click on the “Request for Quotation” button from the main screen, he/she will be brought here. A list of RFQs will be shown to the Vendor. He/She will be able to choose either to View the RFQs or Create Quotation.



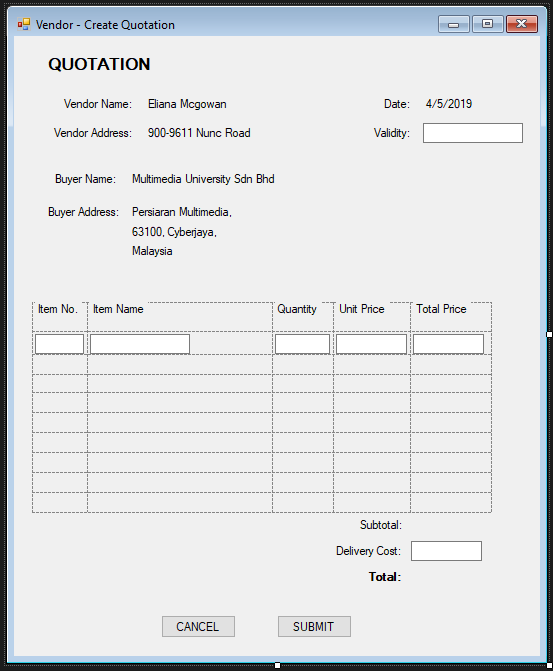
### View RFQ Screen

This is the View RFQ screen for Vendor. After the Vendor click on the “Eye” a.k.a View button of a RFQ on the RFQ List screen, he/she will be brought here. The RFQ form consist of RFQID, Date, Due Date, Company Name, Company Address, Finance Staff Name, Finance Staff Email, Finance Staff Phone and Items details.



### Create Quotation Screen

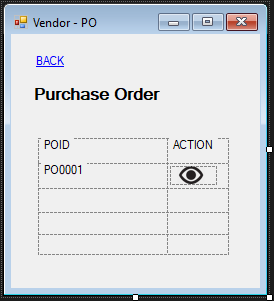
*This is the create Quotation screen for Vendor. After the Vendor tick a RFQ and press “Create Quotation”, he/she will be brought here. The Quotation form consists of vendor name, vendor address, company name, company address and date details. The Vendor have to fill in the Validity, item, price and delivery cost details to the form and press “Submit” to send the quotation to the buying company. Alternatively, the Vendor can press “Cancel” to be brought back to the RFQ List screen.*

**

## Subsystem 11 Screens (Vendor – View PO)

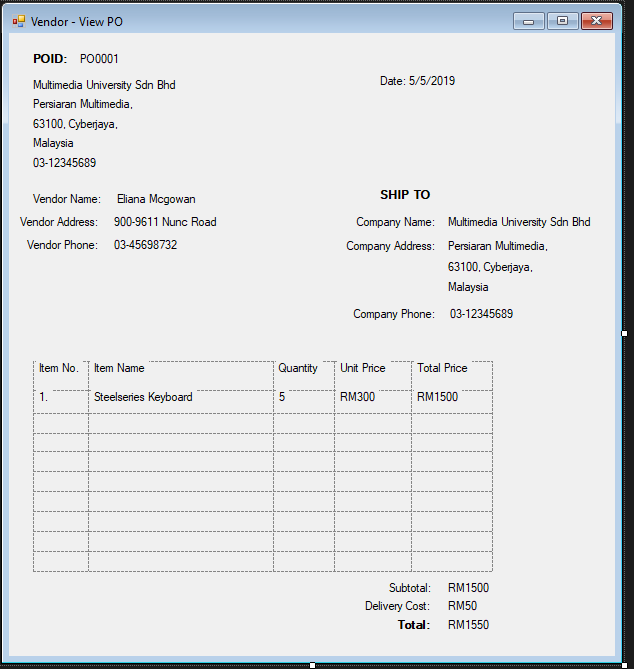
### PO List Screen

This is the PO List screen for Vendor. After Vendor click on the “Purchase Order” button from the main screen, he/she will be brought here. A list of POs will be shown to the Vendor.



### View PO Screen

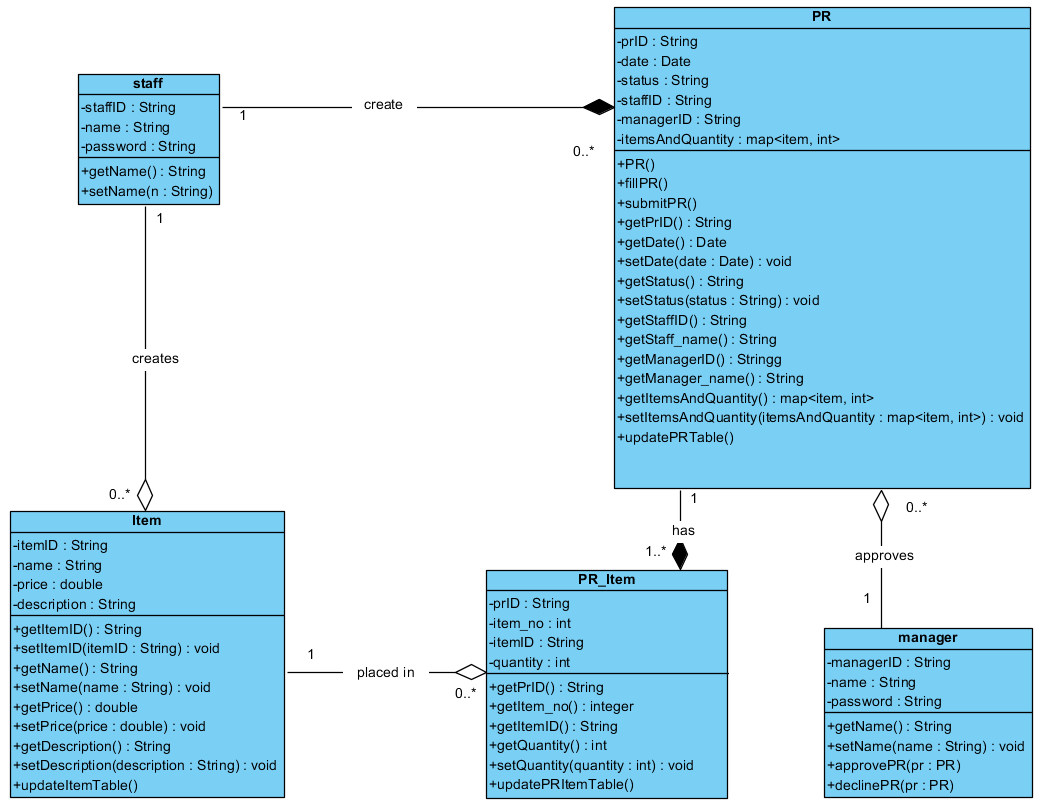
This is the View PO screen for Vendor. After the Vendor click on the “Eye” a.k.a View button of a PO on the PO List screen, he/she will be brought here. The PO form consist of POID, Date, Company Name, Company Address, Company Phone, Vendor Name, Vendor Address, Vendor Phone, Items and Price details.



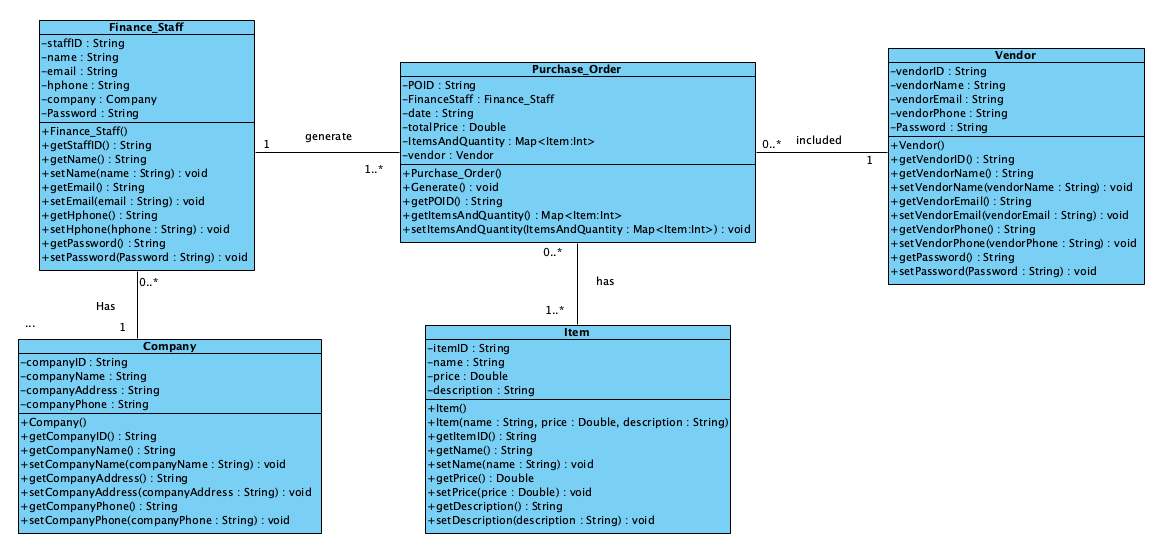
# Component Design

## Main Components

#### Staff and Manager subsystem Class diagram:

**

#### Finance Staff And purchase order class diagram:

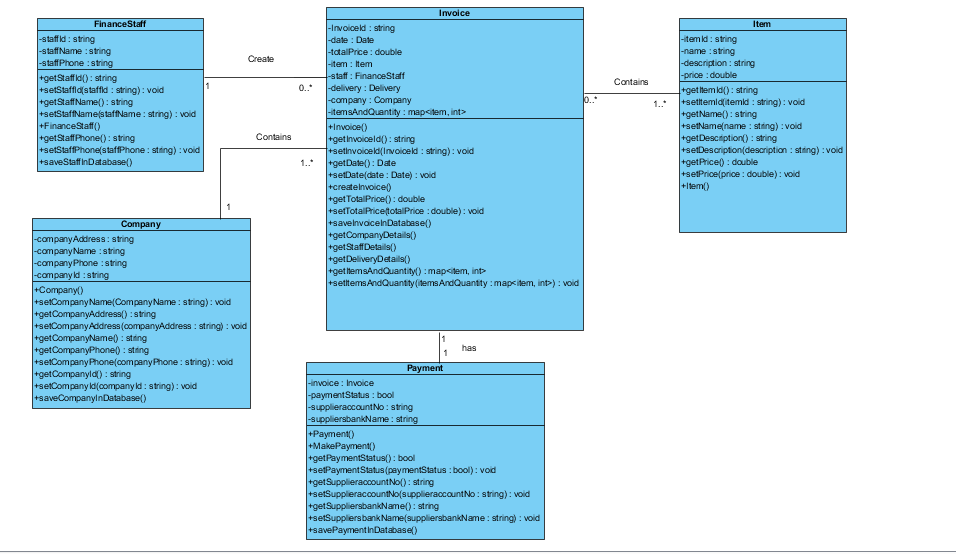


In this class diagram, all the possible classes that will be included along with their attributes, constructors, necessary getters and setters, will be included. The most essential class in this class diagram is the *Purchase Order Class.* The Purchase Order Class, seen in here is only to show potential information that will be found in this class diagram if this class diagram was an independent system, built to work on it’s own. When integrated with the larger system, some

attributes may be added or deleted, such as: *Request for quotation object, Vendor class, Total price attribute, etc.* Depending on the programming language that will be used.

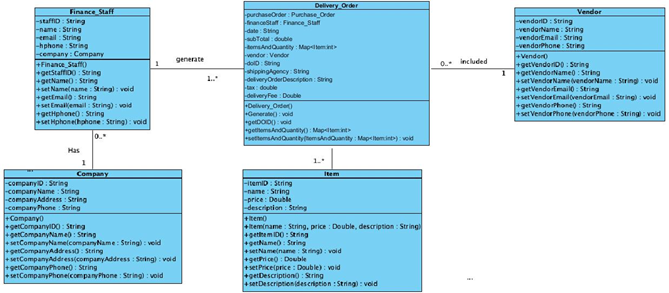
Purchase Order Class (POC), is a class that will work as a medium to deliver the information taken from the quotation to the PO table inside our database. POC will have almost all information needed to form a PO form, that includes but not limited to: PO ID, Current Date, Vendor information, Finance staff information, items and their quantities, class constructor, and setters and getters for all attributes, except for PO ID.

**Invoice and payment class diagram:**



The class diagram includes all the possible attributes along with the constructor and their getter and setter methods. Company may have many finance staff.Finance staff can create 0 to many invoice.Invoice contains company details,vendor details and item details.One invoice can have only one payment.

#### Finance Staff And Delivery Order class diagram:



The class diagram includes all the possible attributes along with the constructor and their getter and setter methods. Company may have many finance staff. One staff can generate many Delivery order. But a delivery order will be created by only one finance staff. Delivery\_Order is the main class responsible for creating a Delivery Order form. It contains key attributes such as purchase order. From this user will know which purchase it belongs to. It includes the finance staff, shipping agency, DO description, DOID, vendor, fee, subtotal and most importantly the item list. The item list(Map) contains the list of items and quantities. Delivery order can get the list from Purchase order class. From there staff can add the item to generate DO from depending on the shipment. Delivery Order will contain the vendor information from where the item is sent.

### Staff Main Screen

*The staff main screen represents the “Staff” component as it as the main control and the highest level in the component. Basically, after the system level login screen directs the staff to this screen, the staff subsystem starts here.*

***Initiate PR list:***

*This function creates the PR list that contains each PR which was created by the staff. This PR list resides inside the staff main menu to be displayed upon initialization. This function is a part of the initialization of the staff main menu.*

*Input: staffID*

*Output: PRlist*

*Function Initiate\_PR\_list(staffID)*

*indexes ß Retrieve number of indexes from PR table with staffID == pr.staffID*

*For i ß 0 to indexes do*

*Retrieve PR data from PR table*

*Initiate PR object*

*Insert PR object to PRlist*

*End for*

*Return PR list*

*End function*

## view PR:

*This function fills the text fields of the PR view screen; so, the PR can be viewed.*

*Input: index number*

*Output: display of PR*

*Function viewPR(indexNumber)*

*Initialize staff name*

*Initialize manager name*

*staff name ß retrieve staff name from staff table with*

*staffID == PRlist[indexNumber].staffID*

*manager name ß retrieve manager name from manager table*

*with managerID == PRlist[indexNumber].managerID*

*insert the PRID of PRlist[indexNumber] into PRID text field*

*insert the date of PRlist[indexNumber] into date text field*

*insert the staffID of PRlist[indexNumber] into staffID text field*

*insert the managerID of PRlist[indexNumber] into managerID text field*

*insert the status of PRlist[indexNumber] into status text field*

*insert the reason of PRlist[indexNumber] into reason text field*

*insert the staff name into staff name text field*

*insert the manager name of into manager name text field*

*for each item in PRlist[indexNumber]*

*insert the item number into item number text field*

*insert the item name into item name text field*

*insert the item description into item description text field*

*insert the item price into item price text field*

*insert the item quantity into item quantity text field*

*end for each*

*end function*

## create Item:

*This function creates an item to be stored inside the item table, in preparation for it to be added to PR.*

*Input: item name, item desciprion, item price*

*Output: item id*

*Function create\_item(itemName, itemDescription, itemPrice)*

*Initialize item*

*item.name ß itemName*

*item.descripotion ß itemDesciprion*

*item.price ß itemPrice*

*insert item as a new record to item table to generate item id*

*Return item.id*

*End function*

## 

## create PR:

*This function creates a PR to be stored inside the PR table and submitted, in preparation for it to be processed by the manager.*

*Input: selection of items and quantity*

*Output: PRID*

*Function create\_PR()*

*Initialize PR*

*PR.date ß system current date*

*PR.staffID ß staffID*

*PR.status ß pending*

*Do*

*Select item from drop list*

*Fill quantity field*

*Add item*

*Insert item and quantity to PR.itemsAndQuantity*

*Update item\_PR table*

*While submit button is not clicked*

*// only reach here if submit is clicked*

*insert PR as a new record to PR table to generate PR id*

*Return PR.id*

*End function*

### 5.1.2 Manager Main Screen

The manager main screen represents the “manager” component as it as the main control and the highest level in the component. Basically, after the system level login screen directs the manager to this screen, the manager subsystem starts here.

**initiate PR list:**

This function creates the PR list that contains each PR which has its status as “Pending”. This PR list resides inside the manager main menu to be displayed upon initialization. This function is a part of the initialization of the manager main menu.

Input: null

Output: PRlist

Function Initiate\_PR\_list()

indexes ß Retrieve number of indexes from PR table with status == pending

For i ß 0 to indexes do

Retrieve PR data from PR table

Initiate PR object

Insert PR object to PRlist

End for

Return PR list

End function

## Approve/reject PR:

This function updates the PR status from “Pending” to either approved or rejected.

Input: PRobject

Output: null

Function approvePR(PRobject) / rejectPR(PRobject)

newStatus ß approved/rejected

PRobject.status ß newStatus

UpdatePRTable(PRobject.id , newStatus)

End function

## view PR:

This function fills the text fields of the PR view screen; so, the PR can be viewed.

Input: index number

Output: display of PR

Function viewPR(indexNumber)

Initialize staff name

Initialize manager name

staff name ß retrieve staff name from staff table with

staffID == PRlist[indexNumber].staffID

insert the PRID of PRlist[indexNumber] into PRID text field

insert the date of PRlist[indexNumber] into date text field

insert the staffID of PRlist[indexNumber] into staffID text field

insert the status of PRlist[indexNumber] into status text field

insert the reason of PRlist[indexNumber] into reason text field

insert the staff name into staff name text field

for each item in PRlist[indexNumber]

insert the item number into item number text field

insert the item name into item name text field

insert the item description into item description text field

insert the item price into item price text field

insert the item quantity into item quantity text field

end for each

end function

### 5.1.3 Finance Staff Main Screen - Purchase Order

The finance staff main screen represents the “Purchase Order”. Basically, after the system level login screen directs the finance staff to finance staff main menu, then this screen is chosen by the finance staff.

### Initiate PO list

This function creates the PO list that contains each PO which was created by the Finance staff. This PO list resides inside the Finance staff – purchase order main menu to be displayed upon initialization. This function is a part of the initialization of the Finance staff – purchase order main menu.

Input: staffID

Output: POlist

Function Initiate\_PO\_list(staffID)

*indexes Retrieve number of indexes from PO table with staffID == po.staffID*

For i 0 to indexes do

*Retrieve PO data from PO table*

*Initiate PO object*

*Insert PO object to POlist*

*End for*

*Return PO list*

*End function*

### 

### View PO

This function fills the text fields of the PO view screen; so, the PO can be viewed.

Input: index number

Output: display of PO

Function viewPO(indexNumber)

*Initialize staff name*

*staff name <- retrieve staff name from staff table with*

*staffID == POlist[indexNumber].staffID*

*insert the POID of POlist[indexNumber] into POID text field*

*insert the date of POlist[indexNumber] into date text field*

*insert the staffID of POlist[indexNumber] into staffID text field*

*insert the status of POlist[indexNumber] into status text field*

*insert the reason of POlist[indexNumber] into reason text field*

*insert the staff name into staff name text field*

for each item in POlist[indexNumber]

*insert the item number into item number text field*

*insert the item name into item name text field*

*insert the item description into item description text field*

*insert the item POice into item POice text field*

*insert the item quantity into item quantity text field*

end for each

end function

### create PO

This function creates a PO to be stored inside the PO table and submitted. It takes the items and their quantities from the PO\_item table in the database then fills the items list in the PO form

Input: selection of quotation

Output: POID

Function create\_PO()

*Initialize PO*

*PO.date system current date*

*PO.staffID staffID*

*PO.status pending*

*Do*

*Add item to PO.list*

*Add Quantity to PO.Quantity*

*Update item\_PO table*

*While submit button is not clicked*

*// only reach here if submit is clicked*

*insert PO as a new record to PO table to generate PO id*

*Return PO.id*

*End function*

## 5.1.4 Finance Staff Main Menu for DO:

After login as finance staff, the finance staff menu will show the finance staff four options. Where he/she can select the menu to go to Delivery Order Menu. In delivery order menu there will be a list for existing DO ids. These DO ids can be view by clicking the Eye Icon. They can create a DO list form simply by clicking Create DO button. Pseudocode is used to show how each of these task can be done in detail.

## Initiating Delivery Order Menu:

In the Delivery Order Menu constructor, the DO list will be loaded. Finance Staff will click the button to initiate

Input:  
Output: DOID list

Function DeliveryOrderMenu()

*indexes <- Retrieve number of indexes from DOtable*

*For i <- 0 to indexes do*

*Retrieve DO data from DO table*

*Initiate DO object*

*Insert DO object to DOlist*

*End for*

*Return DO list*

*End function*

## View DO

Input: index number of DOlist

Output: display of DO

Function viewDO(index)

Initialize finance\_staff name

Initialize company\_name

Finance\_staff <- retrives from DOlist.finance\_staff

Company <- retrives from DOlist.company

insert the DOID of DOlist[indexNumber] into DOID text field

insert the date of DOlist[indexNumber] into date text field

insert the staffID of DOlist[indexNumber] into staffID text field

insert the Vendor.VendorName of DOlist[indexNumber] into VendorName text field

insert the Vendor.VendorAddress of DOlist[indexNumber] into VendorAddress text field

insert the Vendor.VendorPhone of DOlist[indexNumber] into VendorPhone text field

insert the Purchase\_Order.PO\_ID of DOlist[indexNumber] into POID text field

insert the shipping\_agency of DOlist[indexNumber] into Shipping Agency text field

insert the Company.Company\_name of into Buyer\_name text field

insert the Company.Company\_Address of into Buyer\_Address text field

insert the Company.Company\_Phone of into Buyer\_Phone text field

insert the Delivery\_Cost of into Delivery Cost text field

for each item in DOlist[indexNumber]

insert the Item\_ID into item number text field

insert the item name into item name text field

insert the item price into item price text field

insert the item quantity into item quantity text field

end for each

insert the Subtotal of DOlist into Subtotal text field

insert the Delivery\_Cost of DOlist into Delivery Cost text field

end function

## Create Delivery Order

Finance staff will enter the information in the input text field from the Create Delivery Order menu. After entering everything the finance staff will click on submit. Submit will initiate Delivery Order class to generate the DO form.

Function Submit()

initialize DO

initialize DO\_Map\_List

DO.date<- system date

DO.finance\_staff<- staffID

DO.Purchase\_Order<-Purchase\_Order

DO.Vendor<-DO.Purchase\_Order

for each

DO\_Map\_List(index)<- retrieve list of item and quantity from PO\_List table

end for

remove or change quantity -> DO\_Map\_List

DO.setItemsAndQuantities(DO\_Map\_List)

insert DO.delivery\_cost from Delivery Cost input textfield

DO.Generate()

end

**5.1.5 Finance Staff Main Screen - Invoice**

The finance staff main screen represents the “Invoice”. Basically, after the system level login screen directs the finance staff to finance staff main menu, then this screen is chosen by the finance staff.

### Initiate InvoiceList

This function creates the Invoice list that contains each Invoice which was created by the staff. This invoice list resides inside the finance staff main menu to be displayed upon initialization.This invoice list will use in payment and view invoice function.

Function Initiate\_InvoiceList()

indexes <- Retrieve number of indexes from Invoice table

For i <- 0 to indexes do

Retrieve Invoice data from Invoice table

Initiate Invoice object

Insert invoice object to invoiceList

End for

Return invoiceList

End function

### View Invoice

This function fills the text fields of the invoice view screen retrieving data from database. so, the invoice can be viewed.

Function viewInvoice(indexNumber)

Initialize companyName

Initialize companyAddress

Initialize companyPhone

Initialize vendorName

Initialize vendorAddress

Initialize vendorPhone

companyName , retrieve companyName from Company table with

companyId == InvoiceList[indexNumber].companyId

companyAddress <- retrieve companyAddress from Company table with

companyId == InvoiceList[indexNumber].companyId

companyPhone <- retrieve companyPhone from Company table with

companyId == InvoiceList[indexNumber].companyId

vendorName <- retrieve VendorName from Vendor table with

vendorId == InvoiceList[indexNumber].vendorId

vendorAddress <- retrieve vendorAddress from Vendor table with

vendorId == InvoiceList[indexNumber].vendorId

vendorPhone <- retrieve vendorPhone from Vendor table with

vendorId == InvoiceList[indexNumber].vendorId

insert the invoiceID of InvoiceList[indexNumber] into InvoiceId text field

insert the date of InvoiceList[indexNumber] into date text field

insert the companyName into companyName text field

insert the companyPhone into companyPhone text field

insert the companyAddress into companyAdress text field

insert the vendorName into vendorName text field

insert the vendorPhone into vendorPhone text field

insert the vendorAddress into vendorAddress text field

for each item in InvoiceList[indexNumber]

insert the item number into item number text field

insert the item name into item name text field

insert the item description into item description text field

insert the item price into item price text field

insert the item quantity into item quantity text field

end for each

end function

**Create invoice**

This function Shows the how the invoice will be generated in the system.  
at first finance staff will select the item and after that the he/she need to put the quantity and price in the box field.

Function create\_Invoice()

Initialize Invoice

Invoice.date <- system current date

Do

Select item from drop list

Insert Quantity

Insert Price

Insert item and quantity to Invoice.itemsAndQuantity

Update Invoice\_Item table

While (submit button is not clicked)

Insert subtotal

Insert delivery cost

Insert total price

insert invoice as a new record to invoice table to generate invoice id

Return invoice.id

End function

**Payment**

This function is shows how the payment process will be work in the system to make payment for every invoice.

Function make payment(indexNumber)

*Insert the totalPrice of InvoiceList[indexNumber] into amount text field*

*Insert the invoiceId of invoiceList[indexNumber] into invoice Id text field*

*selects the Bank Name from droplist*

*Fill the Account Name text field*

*Fill the Account No text field*

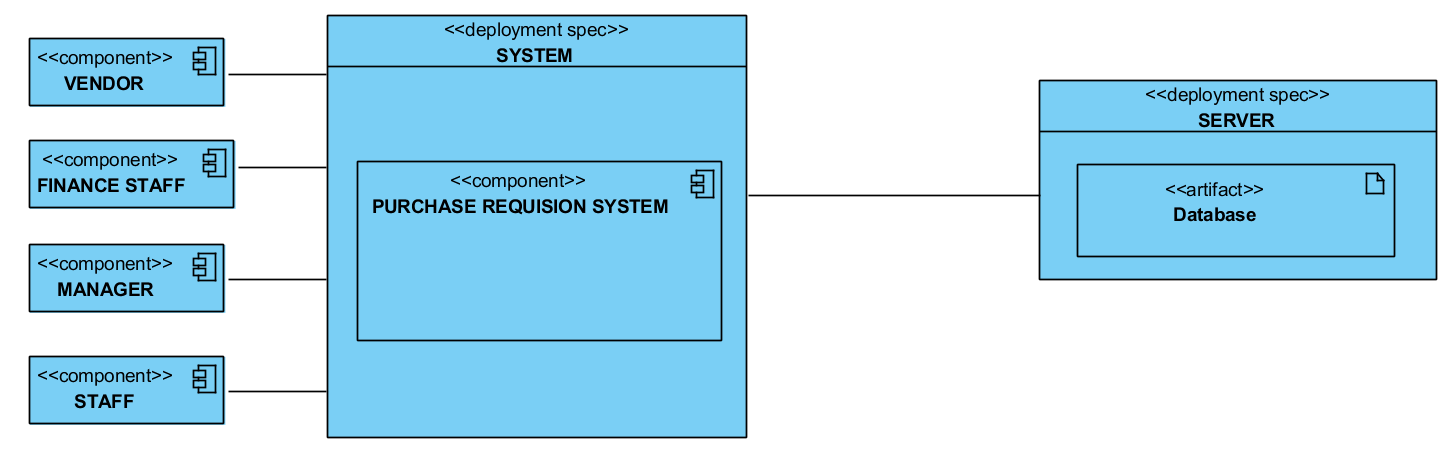
*Update Payment table*

*Return paymentId*

*End function*

# Deployment Design

## Deployment Diagram

**

*The main system is the bridge between our four potential actors and the Database.*

# Updated Requirements

*<This section is* ***Optional.*** *Place the requirements that have been updated from the SRS, particularly those that would have impact on the software design.>*