Software Engineering 2021-2 Manuel Medina

Group 3 - Market Management System

Imer Lopez

Miguel Tzub

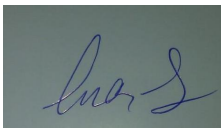
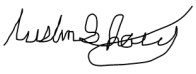
Jadah Castillo

Joel Ical

Rafael Ramos

Austin Shaw

May 16, 2022

SIGNATURE BLOCK				
Statement	I did my share of the work, and I have a general understanding of the contents of the assignment.			
Team Member	**Contribution**	**% of Total**	**Signature**	**Date**
Imer Lopez	*Customer System Requirements, Uses Cases Casual Descriptions, User Interface - Preliminary Design, Domain Analysis, Interaction Diagram, System Architecture*			*16/04/2022*
Jadah Castillo	*Actors and Goals, Traceability Matrix, Fully Dressed Description, Domain Analysis, Interaction Diagram.*			*16/04/2022*
Rafael Ramos	*Customer statement of requirements, Use Cases, Traceability Matrix, Identifying Subsystems, Architecture Styles, System Requirements,, Algorithms/DataStructures,*			*16/04/2022*
Austin Shaw	*Use Case Diagrams, Sequence Diagrams,, Domain Analysis, Class Diagram, Project Management*			*16/04/2022*
Miguel Tzub	*Identifying subsystems, Architecture style, Hardware Requirements, Project Management ,Class Diagram*			*16/04/2022*

Joel Ical	*User Effort Estimation, Domain Analysis*		*Joel Ical*	*16/04/2022*

Note: See point allocations in the table below.		Team Member Name					
		Imer	Miguel	Jadah	Joel	Rafael	Austin
Responsibil	Summary of Changes (*5 points*)	16.16%	16.16%	16.16%	16.16%	16.16	16.16
	Sec.1: Customer Statement of Requirements (*6 points*)	0%	0%	0%	50%	50%	0%
	Sec.2: Glossary of Terms (*4 points*)	17%	16.16%	16.16%	16.16%	16.16%	16.16%
	Sec.3: System Requirements (*6 points*)	70%	0%	0%	0%	30%	0%

i t y Lev el	Sec.4: Functional Requirements Specification (*30 points*)	10%	0%	40%	0%	10%	40%
	Sec.5: Effort Estimation (*4 points*)	50%	0%	0%	50%	0%	0%
	Sec.6: Domain Analysis (25 *points*)	12%	12%	0%	28%	36%	12%
	Sec.7: Interaction Diagram (*40 points*)	50%	0%	50%	0%	0%	0%
	Sec.8: Class Diagram and Interface Specification (*20 points*)	0%	67%	0%	0%	0%	33%
	Sec.9: System Architecture and System Design (*15 points*)	0%	50%	0%	0%	50%	0%
	Sec.10: Algorithm and Data Structure (*4 points*)	0%	0%	0%	0%	100%	0%

	Sec.11: User Interface Design and Implementation (*11 points*)	100%	0%	0%	0%	0%	0%
	Sec.12: Design of Tests (*12 points*)	30%	20%	10%	10%	20%	10%
	History Of works (*5 points*)	16.16%	16.16%	16.16%	16.16%	16.16%	16.16%
	Project Management (*13 points*)	16.16%	16.16%	16.16%	16.16%	16.16%	16.16%

Table of Content

Summary of Changes **5**

Customer Statement Requirement **6**
- Problem Statement 6
- Glossary 8

System Requirements **9**
- Functional Requirements: 9
- Non-Functional Requirements: 10
- On-Screen Appearance Requirements: 11
- FURPS Table 12

Functional Requirement Specification **12**
- Stakeholders 12
- Actor and Goal 13
- Use Cases 15
- Use Case Diagram 17
- Vendor Use Case 18
- Management Use Case 19
- Traceability Matrix 20

User Interface Specification **21**
- Effort Estimation using Use Case Points 21

Analysis and Domain Model **23**
- Conceptual Model 23
- Concept Definitions 28
- Association Definitions 34
- Attributes Definitions 39
- Traceability Matrix 40
- System Operation Contracts 42
- Data Model and Persistent Data Storage 47
- Mathematical Model 58

Interaction Diagrams **60**

Class Diagram and Interface Specification **71**
- Class Diagram: 71
- Data Types and Operation Signatures: 72
- Traceability Matrix: 73

System Architecture **75**

Subsystems: 75
Architecture Styles: 76
Mapping Subsystems to Hardware: 77
Connectors and Network Protocols: 77
Global Control Flow: 77
Hardware Requirements: 78

Algorithms and Data Structures 78

User Interface Design and Implementation 80

Design of Tests 109

History of Works 118

References 119

Summary of Changes

1. Updated Customer Statement requirement
2. Updated Functional Requirement and priority weight based on changes
3. Updated Use Case and Description and requirements covered
4. Updated Use Cases Diagrams based on the newly updated Use Cases
5. Updated Use Case Diagram Traceability Matrix
6. Updated Use Case Fully Dressed description based on newly updated Use Cases
7. Updated System Sequence Diagrams based on newly updated Use Cases
8. Updated Database Schema
9. Updated Sequence Diagram

Customer Statement Requirement

Problem Statement

Local Markets in Belize, without a doubt, are an imminent microeconomic section that contributes greatly to our economy. Local markets also contribute to our culture and gastronomy. The market is an open space where culture diversity can be experienced at first hand. It is known to have an atmosphere of community and camaraderie where farmers and business owners' main goal is to meet local consumers who are interested in local food for their everyday consumption and products for their use. The food and products can vary and at times can expand to livestock and non-produce. Furthermore, the local market is where potential suppliers for your business among your market vendor cohort usually emerge.

Presently, Belize has special days set aside as "Market Day" scheduled specifically for market vendoring. The dates may vary depending on the district; and at times can include all weekdays. Management and administration of most marketplaces has become the responsibility of the respective City and Town councils. These entities assign business stalls, and ensure revenue collections either on a daily, weekly or monthly basis. Currently, the councils do not have an automated management system in place. The importance of an automated system is necessary as the management of a market plaza can become cumbersome at times when assigning stalls as a stall can be assigned to the same vendors. Another challenge is this Covid19 pandemic where necessary strict protocols are enforced such as social distance, proper hygiene booths installation and limited number of consumers at the market at a given time.

Furthermore, the present system lacks many features and mainly entails the use of spreadsheets to keep track of all business stalls with respective fees. The current management challenges are all vulnerable to human error and proper record management may be lost.

With Covid19 cases continuously on the rise, the demand for a modern management system is critical. Business owners and consumers are often focused on minimal interaction, especially securing a business stall and publicly advertising products and or services throughout the world wide web. It is therefore imperative that a user-friendly market management system be designed. The aim for this system is 1. To allow vendors to register into the system to obtain a stall, 2. Proper stall management within the market plaza, 3. Proper record management of payment fees via issued receipts, and 4. reports of annual incomes. This can be achieved via a system application that manages the market plaza by stalls and showcases the wide variety of services and products offered. Another feature will be to allow customers to purchase or place a market item on hold. This application will be an essential tool to improve the productivity and management of all market stakeholders.

Glossary

Market Manager – Manager is defined as a person in position of authority over a group of people or establishment. This entity is responsible for generating reports, updating data, resetting passwords, etc within the webapp.

Customer - A customer is defined as a client requesting a service or product (eg. Vegetables).

Vendor - A vendor is an entity that is paid for goods that are provided.

Database - A database is defined as a collection of data.

Demographic - Demographic is defined as the statistical information, such as age, of a population.

Information System - Information System is defined as a system designed to process and collect data in order to generate information regarding a certain entity.

Reports - A report is a document containing information regarding a particular matter.

Market - a place for regular gathering of people for the purchase and sale of provisions and other commodities.

Market stall/booth - is a structure used by merchants to display and house their merchandise in a market, fairs and conventions.

Grid Map - is a structure map of the market premise.

User - is identified as market manager, vendor or customer.

System Requirements

Priority Weight (PW)	**Description**
3	Very Important
2	Important
1	Least Important

Functional Requirements:

Identifier	PW	Requirement
Req1	3	The system shall provide a grid map of the whole market to view the different sections of the market. Users would be able to perform a zoom view and 360 views of the stalls.
Req2	2	The system shall provide a search input field for all stalls on specific markets across the country.
Req3	3	The system shall provide a route directly from the current location to the nearest market.
Req4	3	The system shall allow the users to view a virtual tour of each stall in the market. Listing of their product or service they offer.
Req5	3	The system shall allow users to reserve/buy items for pickup or delivery. Users will perform this function with all available items of each stall.
Req6	2	The system shall allow vendor users to create a portfolio regarding products or services. Add Items
Req7	2	The system shall allow vendors to view rentals, security, and other dashboard views.
Req8	3	The system shall allow vendors to generate reports based on fees paid.
Req9	2	The system shall allow vendors to request or inquire via a form to management.
Req10	2	The system shall allow management to enable/disable available stalls for the vendor in the market space.
Req11	3	The system shall allow management to generate reports on vendor payments.
Req12	3	The system shall allow management to view and approve requests or address inquiries from vendors
Req13	2	The system shall allow vendors to select an available market stall
Req14	2	The system shall allow vendors to cancel a reserved market stall

Non-Functional Requirements:

Identifier	PW	Requirement
NonReq1	3	The system aims to enhance the customer experience of individuals that visit the local markets in Belize by previewing what to expect.
NonReq2	3	The system will bring a new perspective on shopping in local markets across Belize.
NonReq3	2	The system shall provide a mobile friendly interface which can be viewed on a mobile device.
NonReq4	3	The system will showcase local vendors' products to local and foreign customers.
NonReq5	3	The system aims to facilitate vendors in locating available stalls in local markets.
NonReq6	2	Only Vendor and Management users will require credential to access their respective section on the system
NonReq7	2	The system must produce accurate report payment on each vendors
NonReq8	3	The system should only show on the grid map the stall in which vendor is selling their product or service
NonReq9	2	The system should only display items added by the vendor on their portfolio on their item listing.
NonReq10	1	The system should be fast and fluid
NonReq11	3	The system aims to facilitate market management by providing function features to enhance managerial effectiveness overall.
NonReq12	2	The system should be up and running 24/7
NonReq13	2	The system should log out users after fifteen minutes of inactivity.
NonReq14	3	The system shall provide open access to all the customer's features without having an account on the system
NonReq15	3	The system shall provide access only to authorize vendor that

		has an account in the system
NonReq16	2	The system shall provide access only to authorize management that has an account in the system

On-Screen Appearance Requirements:

Identifier	PW	Requirement
AR1	3	The system should have a user-friendly interface to navigate the platform easily. Usability is a priority to guarantee an excellent user experience.
AR2	2	The system shall use google maps for several maps or direction features
AR3	2	The system shall provide a color scheme for the different sections in the market on the map grid for a more straightforward distinction of each area.
AR4	2	The system must have a consistent look across different browsers and screen resolutions
AR5	2	The system shall have a data dashboard for the purpose of data extraction.
AR6	2	The system shall have a consistent navbar and footer across the platform
AR7	3	The system is designed such that it can work on mobile, laptop, tablet, and computer screen resolutions

FURPS Table

Functionality	● All features implemented in the interface should be mobile-friendly for better platform usability. ● For some features, the user will be required to have stable internet connection
Usability	● The web application will have a user-friendly interface.
Reliability	● The virtual tour of each stall should run smoothly with any

	disturbance ● The system should validate all data entered by the vendor and management.
Performance	● Support multiple users' usage ● Effective Navigate of each stall on the grid map
Supportability	● Web Application mobile friendly and supported by all modern browsers ● Web App build using classes for scalability

Functional Requirement Specification

Stakeholders

1. Customers
2. Vendors
3. Market Management

Actor and Goal

Actor	Roles	Types	Goals
System	Responsible for: 1. Allowing for profile creation for Vendors and Management Staff. 2. Providing an	Initiating	1. Allowing users to gain access to the application via mobile. 2. Provide fast

	interface to explore the market via the grid map view. 3. Generating reports on vendors' products, payment etc. 4. Allowing management to assign stalls to vendors. 5. Sending and Receiving orders. 6. Sending and Receiving requests/inquiries. 7. Providing an interface for making product/ services portfolios for vendors. 8. Providing an interface for viewing rentals, security, and other dashboard views. 9. Generating queries on stalls/markets.		communication between customers and vendors. 3. Provide communication between vendors and management. 4. Provide accurate payment reports. 5. Allow management to keep track of stalls, vendors. 6. Providing customers an easy way to navigate to and through the market.
Customer	Responsible for: 1. Exploring the market's stalls/vendors. 2. Reserving services/products.	Initiating	1. Viewing the grid map and searching for stalls. 2. Select and reserve services/ products when wanted. 3. Collecting their receipt and services/ products.
Vendor	Responsible for: 1. Creating their vendor profile. 2. Creating a portfolio of products. 3. Receiving orders from customers.	Initiating Initiating (Participating)	1. Updating and making the products/ services for their portfolio. 2. Providing data

	4. Sending inquiries to management about stalls. Portfolios etc. 5. Checking rentals, security, and other dashboard views. 6. Checking their fees and payments through reports.	Initiating Initiating (Participating)	from their payments to management.
Management	Responsible for: 1. Creating their management profile. 2. Managing and enabling available stalls and vendors throughout the market. 3. Generating vendor payment reports 4. Viewing vendor requests/inquiries.	Initiating	1. Ensure proper functioning of the system. 2. Assign available stalls to vendors. 3. Responding to vendor requests/inquiries .

Use Cases

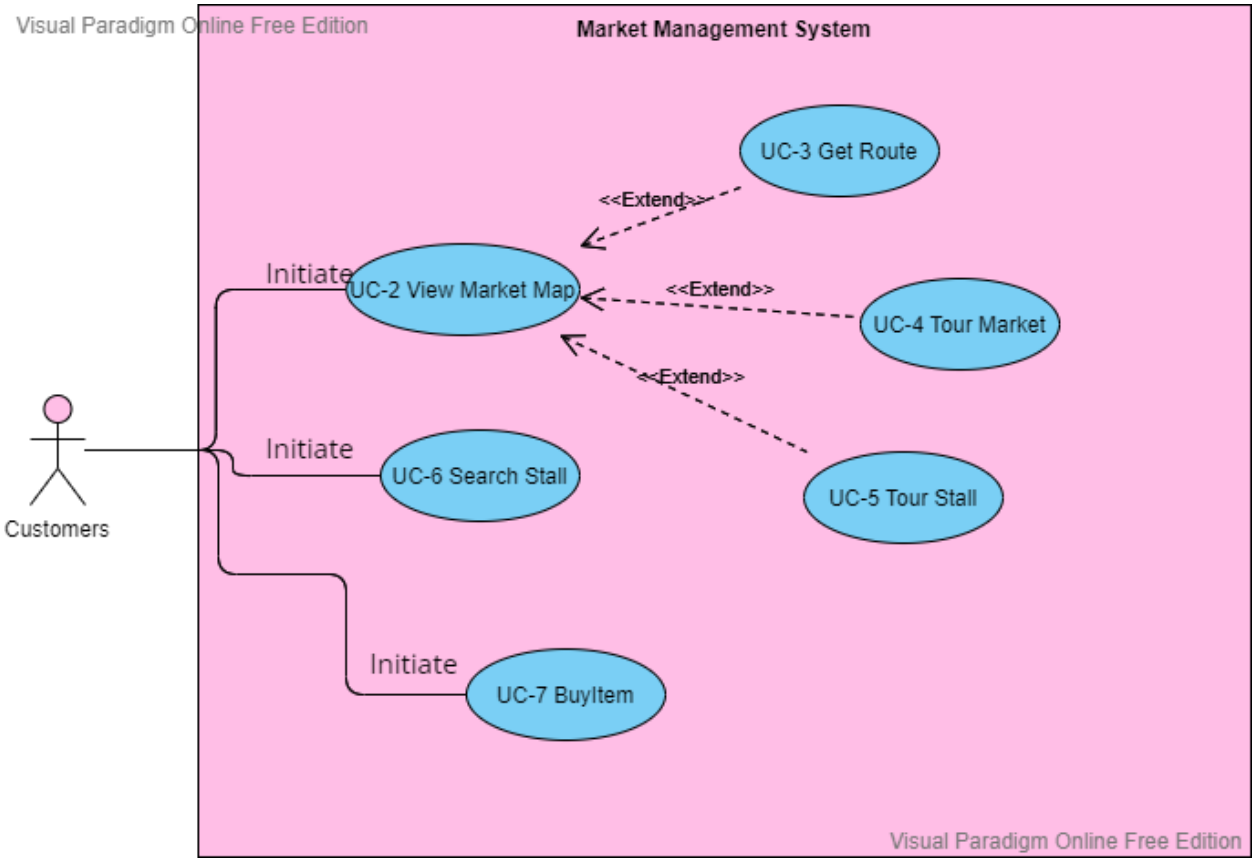
Name	Description	Requirement Covered
UC-1: AuthVendorandManagement	a.) Vendors are required to have an account to access the application b.) Management are required to have an account to access the application	NonReq15, 16
UC-2: ViewMarketMap	Customers will get access to the grid map of the market where they can explore the different sections in the premise.	Req-1,
UC-3: GetMarketRoute	Customers will be able to get the route to the nearest market from their current location	Req-3

UC-4: TourMarket	Customers will be able to explore the market different sections	Req-1
UC-5: TourStall	Customers will be able to do a virtual tour on each stall in the market. Listing products	Req-4
UC-6: SearchStall	Customers will be able to search for a stall on the desire market across the country	Req-2
UC-7: BuyItem	Customers will be able to reserve a product from the listing products of each stall	Req-5
UC-8: CreateProfile	Vendor’s create a profile	Req-6
UC-9: AddProducts	Vendors will be able add products to their product	Req-6
UC-10: ViewInvoice	Vendors will be able to view the expense invoices of the stall.	Req-7
UC-11: GenerateReport	Vendors will be able to generate report based on their invoice expenses	Req-8
UC-12: SubmitRequest	System allows market vendors to submit request forms for the purpose of inquiring available market stalls or reporting any market issue.	Req-9
UC-13: ReserveStall	Vendors will be able to reserve available stall	Req-13
UC-14: CancelReserveStall	Vendors will be able to cancel reserve stall	Req-14
UC-15: ApproveReserveStall	Management will be able to approve the request stall by a vendor.	Req-12
UC-16: EnableStall	Management will be able to enable all available stalls in the market.Manage stalls status	Req-10

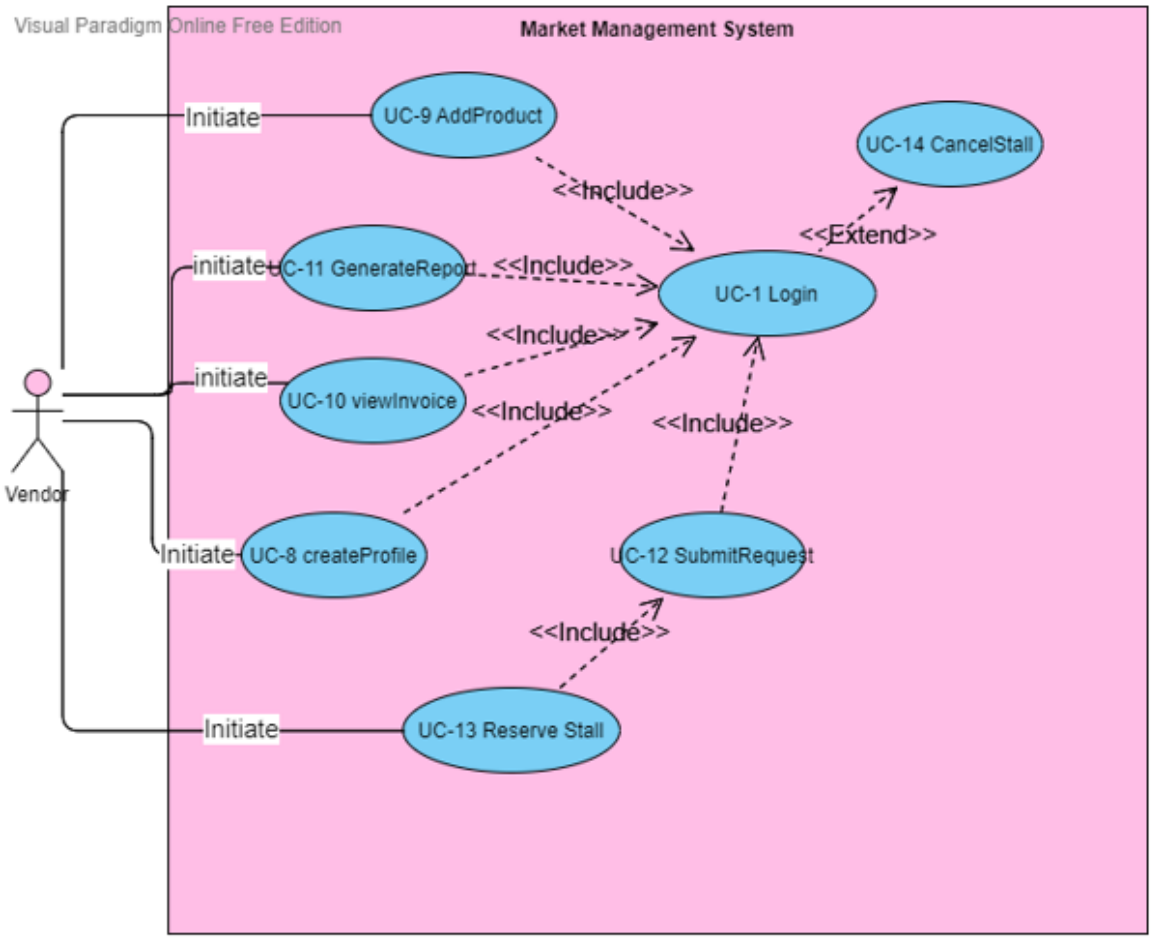
UC-17: DisableStall	Management will be able to disable specific stalls in the market	Req-10
UC-18: GenerateInvoices	Management will generate an invoice on vendor stall usage and other fees.	Req-11
UC-19: ViewInquires	Management will be able to view requests or inquiries from vendors.	Req-12

Use Case Diagram

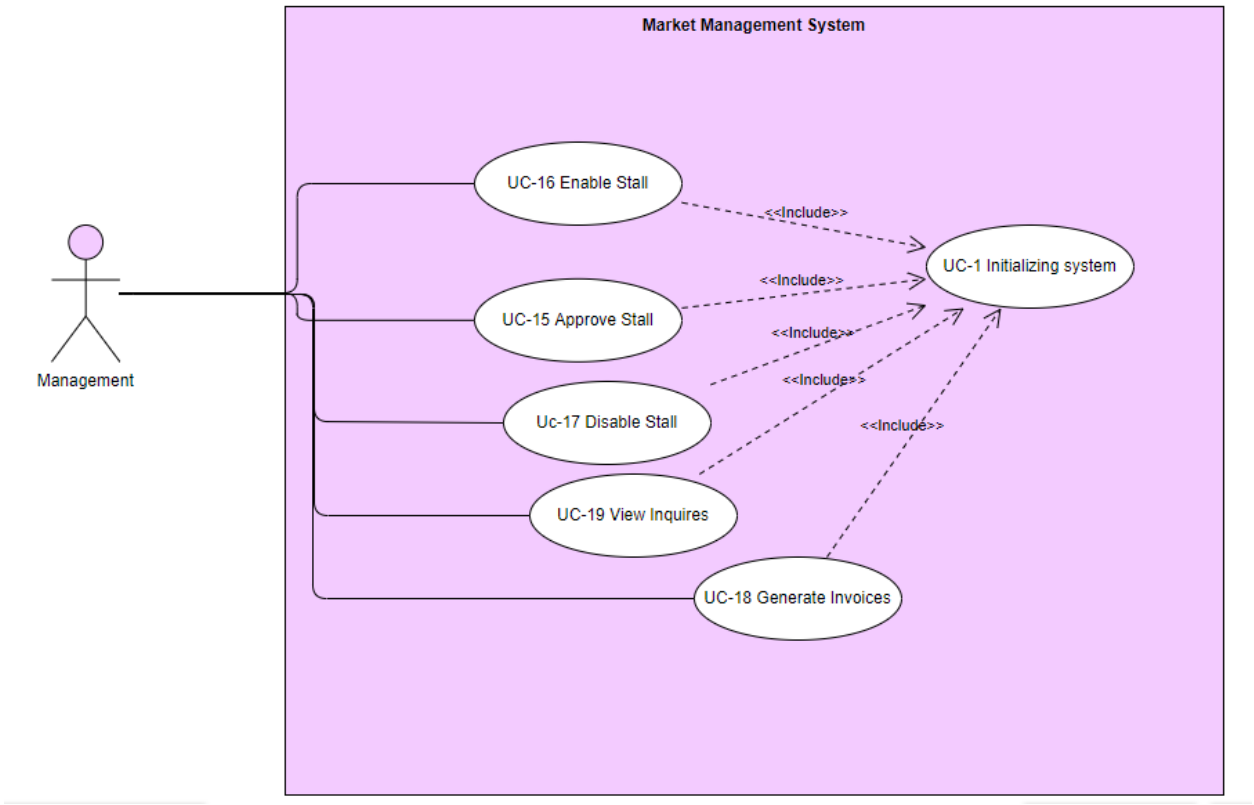
1.) Customer Use Case Diagram


Visual Paradigm Online Free Edition
Market Management System
UC-3 Get Route
<<Extend>>
Initiate
UC-2 View Market Map
<<Extend>>
UC-4 Tour Market
<<Extend>>
Initiate
UC-6 Search Stall
UC-5 Tour Stall
Customers
Initiate
UC-7 BuyItem
Visual Paradigm Online Free Edition

2.) Vendor Use Case



3.) Management Use Case



Traceability Matrix

The traceability matrix below maps out the system requirements to use cases. The purpose of the matrix is to ensure the system requirements are met and all use cases have a purpose. In the following matrix, let “UC” stand for “Use Case”.

REQ	PW UC	UC 1	UC 2	UC 3	UC 4	UC 5	UC 6	UC 7	UC 8	UC 9	UC 10	UC 11	UC 12	UC 13	UC 14	UC 15	UC 16	UC 17	UC 18	UC 19
REQ1	3		**X**		**X**															
REQ2	2						**X**													
REQ3	3			**X**																
REQ4	3					**X**														
REQ5	3							**X**												
REQ6	2								**X**	**X**										
REQ7	2										**X**									
REQ8	3											**X**								
REQ9	2												**X**							
REQ 10	2																**X**	**X**		
REQ 11	3																		**X**	
REQ 12	3															**X**				**X**
REQ 13	2													**X**						
REQ 14	2														**X**					
NONREQ 15	3	**X**																		
NONREQ 16	3	**X**																		

Max PW *1-Most Important 3-Least Important*	**3**	**3**	**3**	**3**	**3**	**2**	**3**	**2**	**2**	**2**	**3**	**2**	**2**	**2**	**3**	**2**	**2**	**3**	**3**
Total Weight	**6**	**3**	**3**	**3**	**3**	**2**	**3**	**2**	**2**	**2**	**3**	**2**	**2**	**2**	**3**	**2**	**2**	**3**	**3**

User Interface Specification

Effort Estimation using Use Case Points

Actor Type	Simple Average Complex	Average	Complex
Number of Actors	1	2	2
Weight	1	2	3
UAW	11		

Use Cases Types	Simple Average Complex	Average	Complex
Number of Use Cases	5	9	5
Weight	1	2	3
UUCW	38		

Unadjusted Use Case Points = UAW + UUCW = 11 + 38 = 49

Technical Factor	Weight	Perceived Complexity	Calculated Factor
Distributed system	2	3	6
Performance Objectives	2	3	6

End-user Efficiency	1	3	3
Complex Processing	1	2	2
Reusable Code	1	2	2
East to Install	0.5	0	0
Easy to use	0.5	2	1
Portable	2	1	2
Easy to change	1	4	4
Concurrent	1	3	3
Security Features	1	4	4
Access to Third Parties	1	1	1
Training needs	1	0	0
Technical Factors Total			34
Technical Complexity Factor			0.94

Environmental Factors	Weight	Calculated Factors	Impact
Familiar with the development process	1.5	3	4.5
Application Experience	0.5	4	2
Object Oriented Experience	1	4	4

Lead Analyst capability	0.5	3	1.5
Motivation	1	4	4
Stable requirements	2	1	2
Part-time staff	-1	0	0
Difficult Programming Language	-1	3	-3

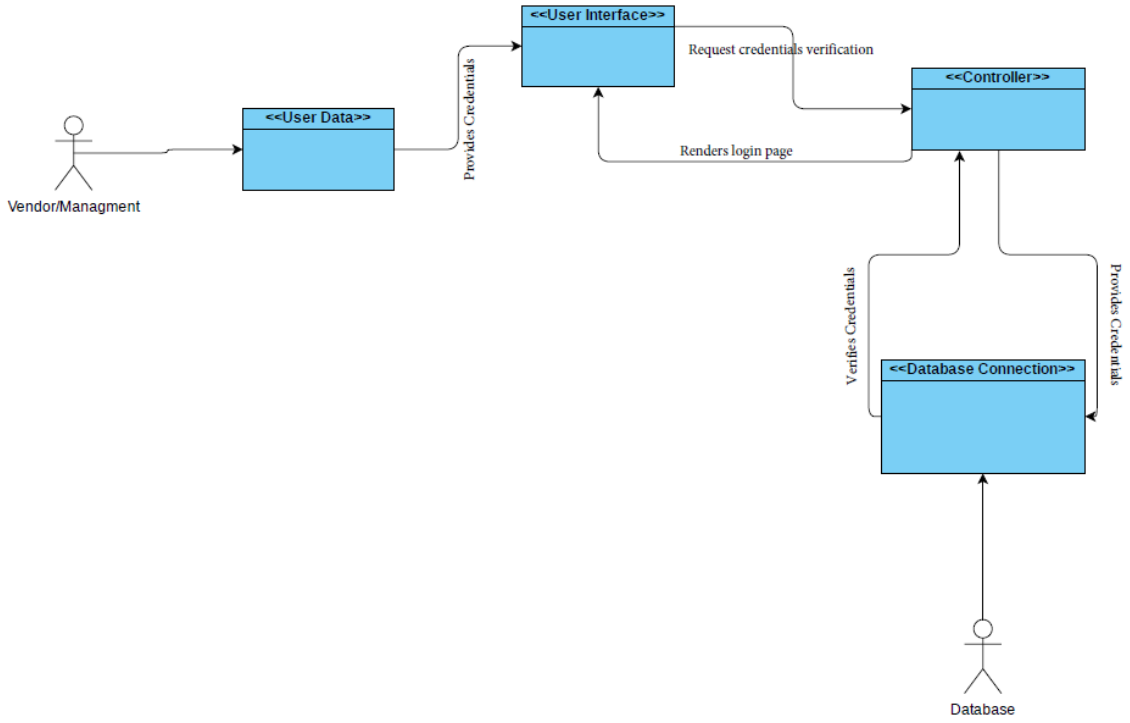
Efactor 15 Environment Complexity Factor 0.95

Adjusted Use Case Points = 49 * 0.94 * 0.95 = 43.75
Staff hour per UCPoint = 28
Effort Estimate = 43.75 * 28 = 1225 hrs

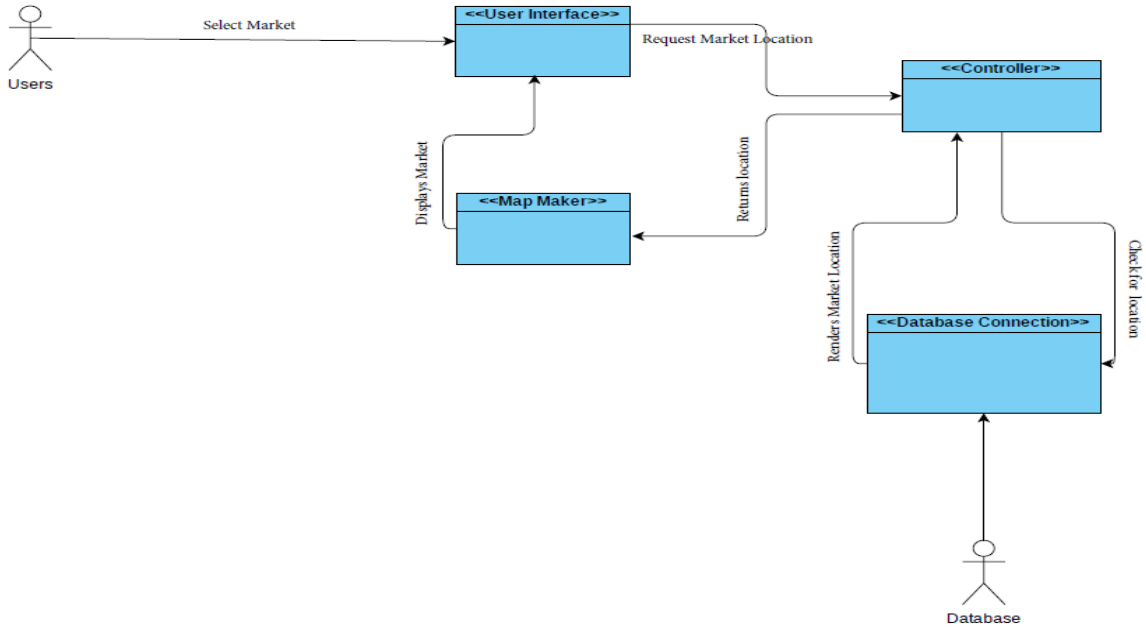
Analysis and Domain Model

Conceptual Model

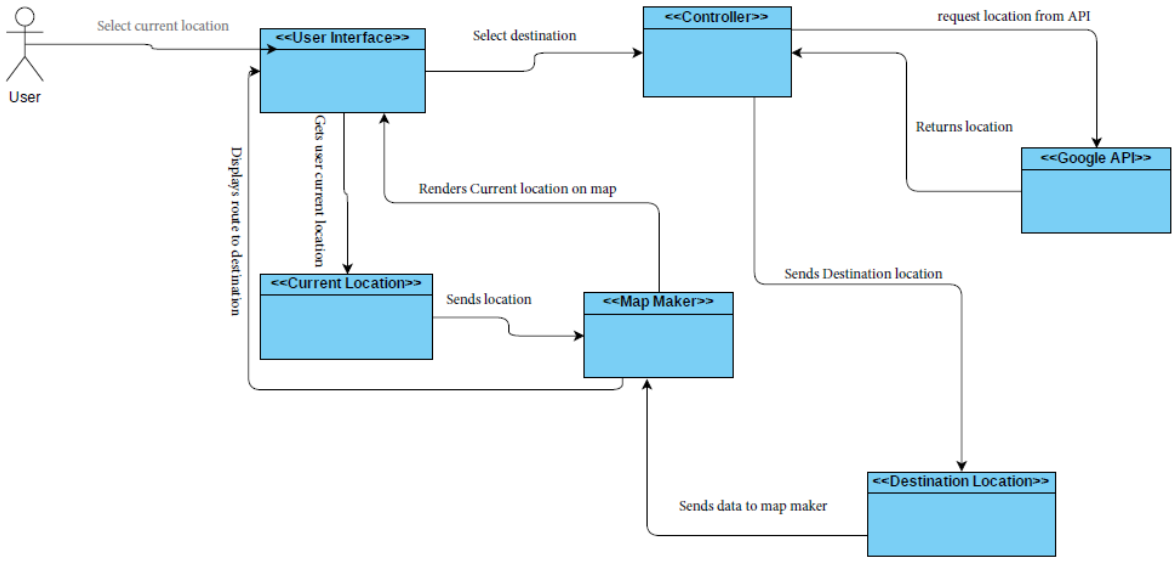
UC-1:AuthVendorandManagment



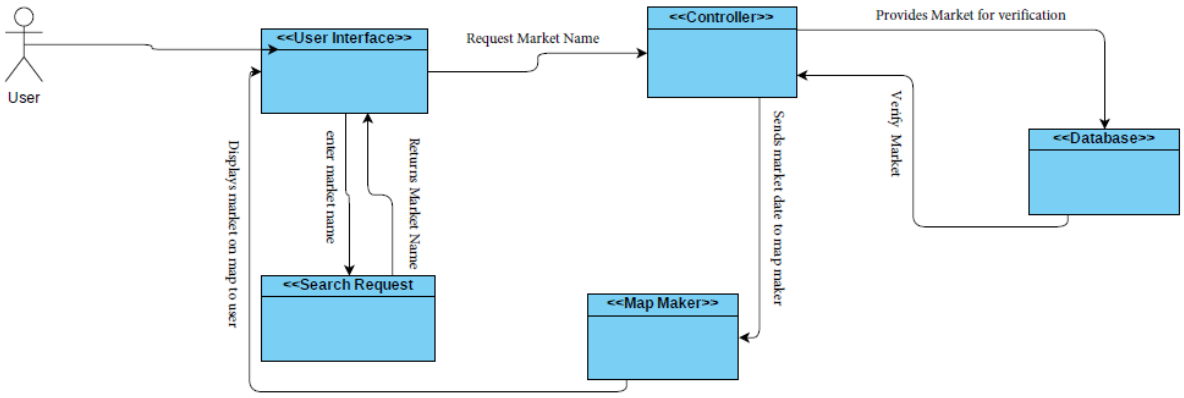
UC-2: ViewMarketMap



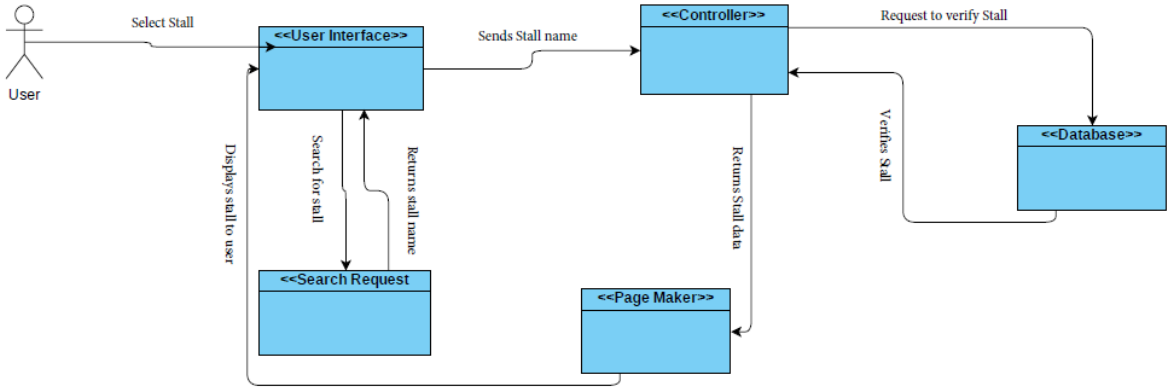
UC3-GetMarketRoute



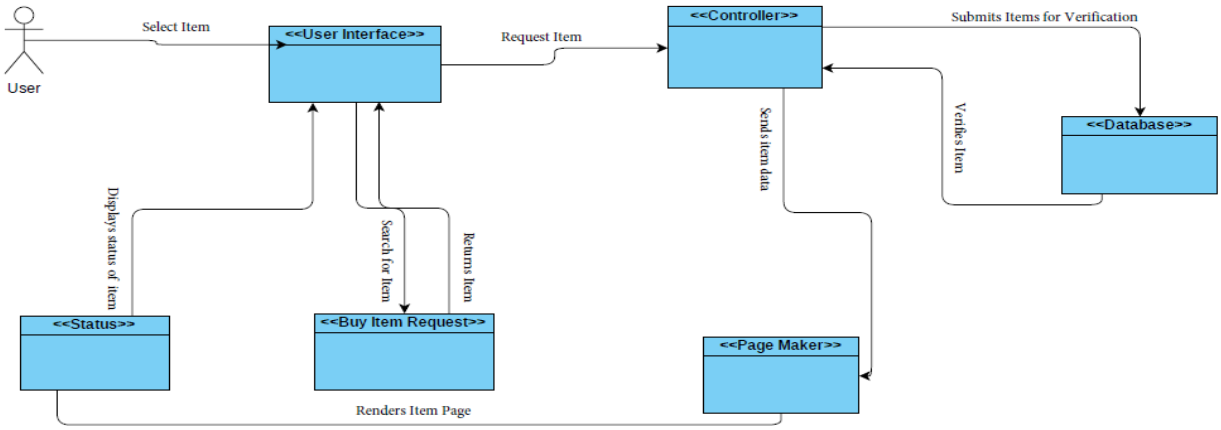
UC-4: TourMarket



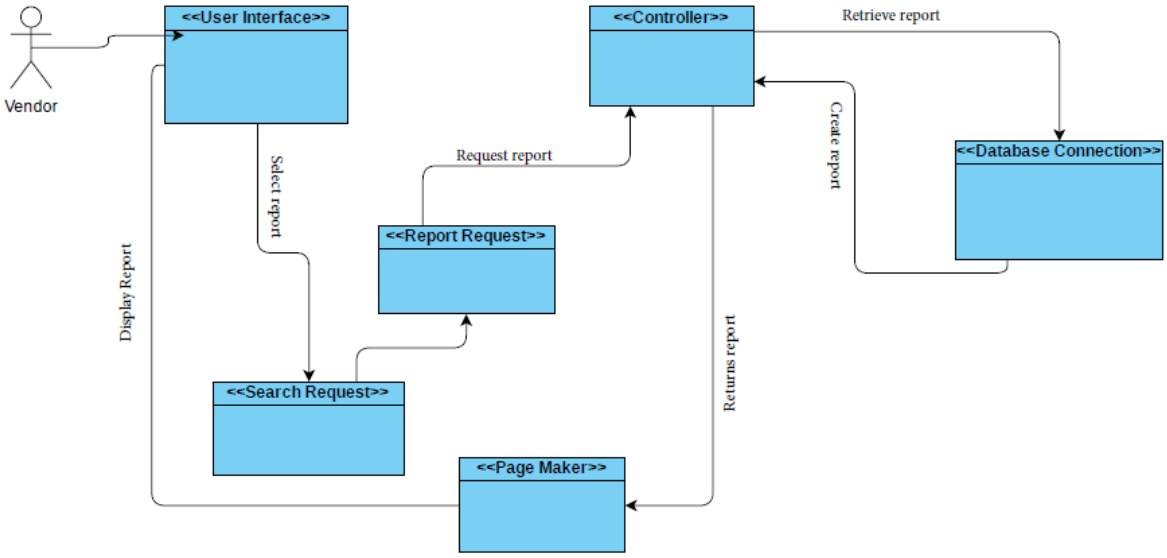
UC-5: TourStall



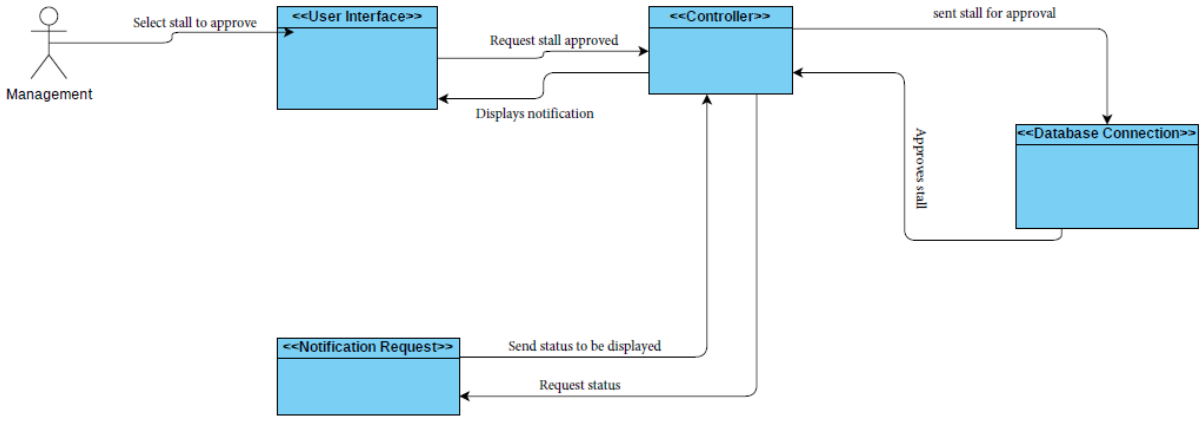
UC-7:BuyItem



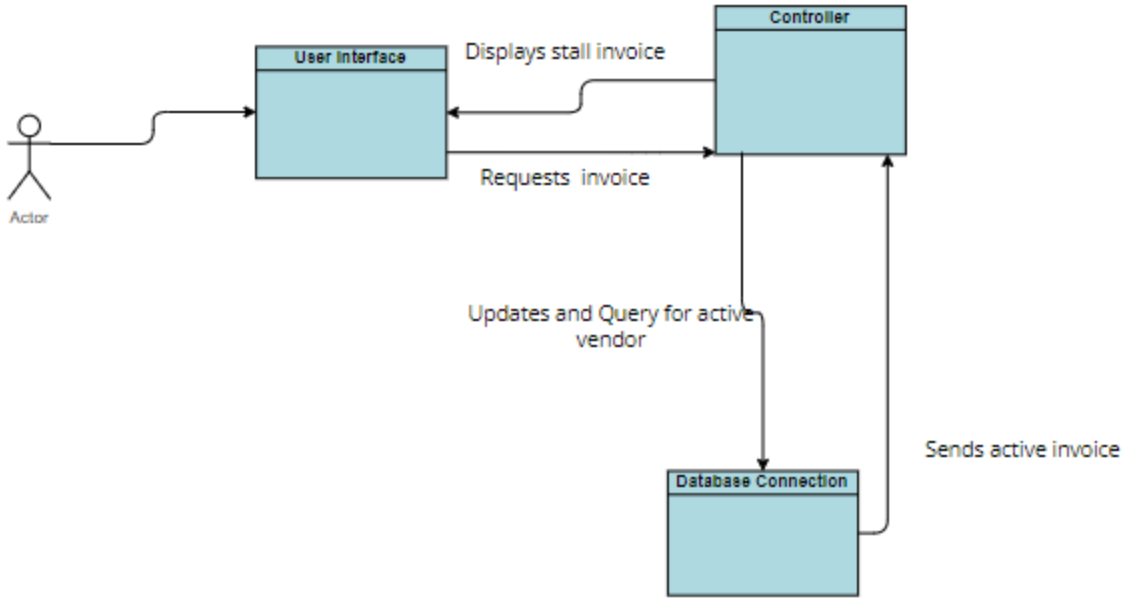
UC-11:Generate Report



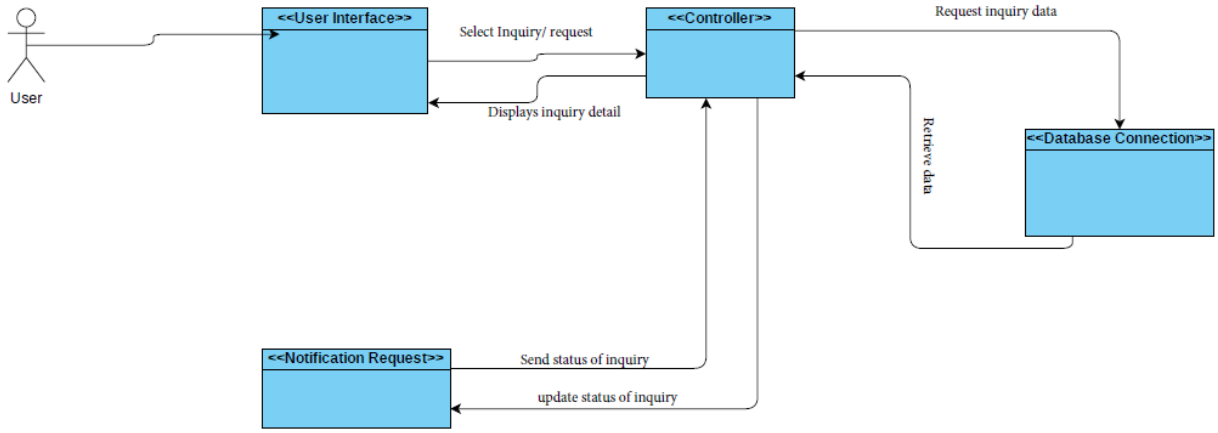
UC-15:ApproveStall



UC-18:GenerateInvoices



UC-19:ViewInquiries



Concept Definitions

Figure 1: Concept Definition for UC-1: AuthVendorandManagement

Responsibility Description	Type	Concept Name

Rs1. Coordinate actions of concepts associated with this use case and delegate the work to other concepts.	D	Controller
Rs2. Render a display to the actor's Web browser for login credentials input	K	User Interface Page
Rs3. Container for user's authentication data, such as username, password.	K	User Data
Rs4. Prepares a database query that verifies the user's credentials entered	K	Database Connection
Rs5..HTML document that shows the actors its login portal.	K	User Interface Page

Figure 2: Concept Definition for UC-2:ViewMarketMap

Responsibility Description	Type	Concept Name
Rs1. Coordinate actions of concepts associated with this use case and delegate the work to other concepts.	D	Controller
Rs2. Render a map to the actor's Web browser for the purpose of an overview of market stalls.	D	Map Maker
Rs3 HTML document that shows the actor the current map context, what actions can be done, and outcomes of the previous actions	K	User Interface Page

Figure 3: Concept Definition for UC-3:GetMarketRoute

Responsibility Description	Type	Concept Name
Rs1. Coordinate actions of concepts associated with this use case and delegate the work to other concepts.	D	Controller

Rs2. Render a map to the actor's Web browser for display	D	Map Maker
Rs3. Get actor’s current location	K	Current Location
Rs4. Render a map of the nearest market locations	K	Destination Location
Rs5. Renders a map route to the actors Web browsers for display	D	Map Maker

Figure 4: Concept Definition for U4-: TourMarket

Responsibility Description	Type	Concept Name
Rs1. Coordinate actions of concepts associated with this use case and delegate the work to other concepts.	D	Controller
Rs2. Render a map to the actor's Web browser for display	D	Map Maker
Rs3 Search option on map for a specific section accessed by tapping.	K	Search Request
Rs4. HTML document that shows the actor the selected map sections.	K	User Interface Page

Figure 5: Concept Definition for UC-5:Tour Stall

Responsibility Description	Type	Concept Name
Rs1. Coordinate actions of concepts associated with this use case and delegate the work to other concepts.	D	Controller
Rs2. Search option for specific market stall by tapping.	K	Search Request
Rs3. Render a market stall virtual reality to the actor's Web browser for display	D	Page Maker
Rs4. HTML document that shows the actor a virtual reality of products on selected market stalls.	K	User Interface Page

Figure 6: Concept Definition for UC-7:BuyItem

Responsibility Description	Type	Concept Name
Rs1. Coordinate actions of concepts associated with this use case and delegate the work to other concepts.	D	Controller
Rs2. Render market stall virtual reality to the actor's Web browser for display	D	User Interface Page
Rs3. Displays list of products for the purpose of details, status and prices.	K	Status

Rs4. Form specifying the required fields needed to buy item	D	Buy Item Request
Rs5. Prepares a database query to update database with purchased item	D	Database Connection

Figure 7: Concept Definition for UC-11:GenerateReport

Responsibility Description	Type	Concept Name
Rs1. Coordinate actions of concepts associated with this use case and delegate the work to other concepts.	D	Controller
Rs2. Search bar specifying the search parameters for database user lookup.	K	Search Request
Rs3. Prepare a database query that matches the user's search criteria and retrieves the resulting record set from the database.	D	Database Connection
Rs4. HTML document that shows the actor, the retrieved records, and what actions can be done.	K	User Interface Page
Rs5. User for which a report will be generated.	K	Report Request
Rs6. Render pdf report for sending to the actor's Web browser for display.	D	PageMaker

Figure 8: Concept Definition for UC-15:ApproveStall

Responsibility Description	Ty	Concept Name
Rs1. Coordinate actions of concepts associated with this use case	D	Controller

and delegate the work to other concepts.		
Rs2. Prepare a database query to retrieve all market stall request	D	Database Connection
Rs3.HTML document that displays all market stall pending approval request	K	User Interface Page
Rs4. Notify the user that the market stall has been approved or denied.	D	Notification Request
Rs5. Prepare a database query to update the user data to to assigned stall	D	Database Connection

Figure 9: Concept Definition for UC-18:GenerateInvoices

Responsibility Description	Type	Concept Name
Rs1. Coordinate actions of concepts associated with this use case and delegate the work to other concepts.	D	Controller
Rs2.Prepare a database query to retrieve all active vendors invoices.	D	Database Connection
Rs3. HTML document that displays all market stall invoices	K	User Interface Page
Rs4 Prepare a database query to retrieve any special charges and adds it to report	D	Database Connection
Rs5. HTML document that displays updated fees on market stall invoices	K	User Interface Page

Figure 10: Concept Definition for UC-19:ViewInquiries

Responsibility Description	Type	Concept Name
Rs1. Coordinate actions of concepts associated with this use case and delegate the work to other concepts.	D	Controller
Rs2.Prepare a database query to retrieve all inquiries	D	Database Connection
Rs3. HTML document that displays all details of inquiries	K	User Interface Page
Rs4. Notifies vendor that that inquiry has been addressed	D	Notification Request
Rs5. Prepare a database query to update inquiries as solved and removed from the list.	D	Database Connection

Association Definitions

Figure 11: Association Definition for UC-1: AuthVendorandManagement

Concept Pair	Association Definition	Association Name
Controller ↔ User Interface Page	Controller receives credentials entered in the User Interface page	Receives
User Interface Page ↔ User Data	Data entered in the interface page is saved for processing	Stores data
User Data ↔ Database Connection	The user data entered is then queried for the purpose of credential verification	Verifies
Database Connection ↔	Renders interface page for display to the	Displays

Concept Pair	Association Definition	Association Name
Controller ↔ User Interface Page	Controller receives credentials entered in the User Interface page	Receives
User Interface Page	actor’s browser	

Figure 12: Association Definition for UC-2: ViewMarketMap

Concept Pair	Association Definition	Association Name
Controller ↔ Map Maker	Controller creates a map for the actor to select his/her desired market location	Render map
Map Maker ↔ User Interface Page	Renders market stalls interface page for display to the actor's browser	Displays

Figure 13: Association Definition for UC-3:GetMarketRoute

Concept Pair	Association Definition	Association Name
Controller ↔ Map Maker	Controller creates a map for the actor to select his/her desired market location	Render map
Controller ↔ Current Location	Gets the current location of the actor	Provide location
Controller ↔ Destination Location	Gets the destination location of the customer	Provide location
Controller ↔ Page Maker	Render a map preview of the route	Render map

Figure 14: Association Definition for U4-: TourMarket

Concept Pair	Association Definition	Association Name
Controller ↔ Map Maker	Controller creates a map for the actor to select his/her desired market location	Render map
Controller ↔ Search Request	Controller passes a map search query	Queries
Controller ↔ User Interface Page	Controller creates a map of the selected market for the actor to select his/her market stall.	Render map

Figure 15: Association Definition for U5-: TourStall

Concept Pair	Association Definition	Association Name
Controller ↔ Search Request	Controller passes a search query for the selected market grid for the purpose stall view.	Queries
Search Request ↔ Page Maker	Page Maker prepares to display selected market stalls.	Prepares
Controller - User interface page	Controller sends a list of items available at the market stall.	Display

Figure 16: Association Definition for UC-7:BuyItem

Concept Pair	Association Definition	Association Name
Controller ↔ User Interface Page	Controller creates an interface for the purpose of market item to be selected	Displays
User Interface Page ↔ Status	The User Interface page displays the availability of items on a per item basis.	Provide Results

Controller ↔ Buy Item request	Controller accepts data for the item about to be purchased.	Receives
Controller ↔ Database Connection	Controller updates the database with items purchased.	Saves to

Figure 17: Association Definition for UC-11:GenerateReport

Concept Pair	Association Definition	Association Name
Controller - user interface page	Interface created by the controller for report to be chosen	Display
Controller - Database connection	Controller updates and retrieves info from the database to create a report	Queries
Controller - Search Request	Controller accepts input data to lookup results through the database	Queries
Controller-Page Maker	Controller generates the pdf report to be displayed	Display

Figure 18: Association Definition for UC-15:ApproveStall

Concept Pair	Association Definition	Association Name
Controller ↔ User Interface	Interface displays all stalls pending approval received from the database	Display
Controller ↔ Database Connection	Controller Retrieves and updates market stall and user information in the database	Saves to

Controller ↔ Notification Request	Controller displays a notification of stall status	Display

Figure 19: Concept Definition for UC-18:GenerateInvoices

Concept Pair	Association Definition	Association Name
Controller ↔ Database Connection	Controller queries, retrieves active vendor invoices and update report	Saves to
Controller ↔ User Interface	Controller prepares and displays stall invoices and updated fees	Prepare

Figure 20: Association Definition for UC-19:ViewInquiries

Concept Pair	Association Definition	Association Name
Controller ↔ Database Connection	Controller retrieves and updates inquiry information	Queries
Controller ↔ User Interface	Controller creates a document and displays inquiry details	Display

Controller ↔ notification request	Controller sends a notification when inquiry is addressed	Notifies

Attributes Definitions

Concept	Attributes Definition	Attributes Description
Controller	NA	
User Interface Page	NA	
User Data	User Information	Personal information such as firstname, last name, email, and address of user.
Database Connection	NA	
Map Maker	Current Location Destination Location	GPS location such as the current and destination location.
Current Location	NA	
Destination Location	NA	
Page Maker	NA	
Search Request	Stall ID	Market Stall ID number
Status	NA	

Buy Item Request	Product Information	Product information such as item name
Report Request	User list	list of records from which report can be generated.
Notification Request	User ID	Vendor user ID

Traceability Matrix

The traceability matrix below maps out the use cases to the domain concepts from which they were derived from. This mapping is important - it ensures there is a logical progression. It also shows the concepts that are widely used. For example, as can be seen in the table below, the database connection, page maker, and interface pages are concepts that map out to each use case. These three concepts are the core components of our web-app. Often they are referred to as the database, server, and browser.

	Domain Concepts									
Use Case	**UC 1**	**UC 2**	**UC 3**	**UC 4**	**UC 5**	**UC 7**	**UC1 11**	**UC1 15**	**UC1 8**	**UC19**
PW	6	3	3	3	3	3	3	3	3	3

Controller	x	x	x	x	x	x	x	x	x	x
User Interface Page	x	x		x	x	x	x	x	x	x
User Data	x									
Database Connection	x					x	x	x	x	x
Map Maker		x	x	x						
Current Location			x							
Destination Location			x							
Page Maker			x		x		x			
Search Request				x	x		x			
Status						x				
Buy Item Request						x				
Report Request							x			
Notification Request								x		x

System Operation Contracts

Use Case	UC-1: AuthVendorandManagement
Contract Name	authUser(username, password, userType)
Responsibilities	Authenticate Vendor and Management user to access their respective section on the system
Type:	System
Exceptions	No access to the system when wrong credentials is input or select wrong user type
Preconditions	User must have an account on the system
Postconditions	User account information will be verify

Use Case	UC-2: ViewMarketMap
Contract Name	viewMarketMap(marketName)
Responsibilities	To display a gridMap of the different market across country when one is select
Type:	System

Exceptions	none
Preconditions	Select a market of choice
Postconditions	Display grid map

Use Case	UC-3: GetMarketRoute
Contract Name	getMarketRoute()
Responsibilities	Provide direction route from current location to the nearest market or select desire market
Type:	System
Exceptions	none
Preconditions	Allow to access current location
Postconditions	Provide route direction from current location to desire market location

Use Case	UC-4: TourMarket
Contract Name	marketTour()
Responsibilities	Provide VR Navigation in the different market premise

Type:	System
Exceptions	none
Preconditions	Select Market to start market tour
Postconditions	Display VR Navigation

Use Case	UC-5: TourStall
Contract Name	stallTour()
Responsibilities	Provide VR Navigation in vendor’s stall
Type:	System
Exceptions	none
Preconditions	Select desire stall to perform tour stall
Postconditions	Display Stall VR Navigation

Use Case	UC-7: BuyItem
Contract Name	buyItem()
Responsibilities	Allow user to buy item available in listing on each stall

Use Case	UC-7: BuyItem
Type:	System
Exceptions	none
Preconditions	Enter necessary info to make the purchase
Postconditions	Confirmation of purchase item

Use Case	UC-11: GenerateReport
Contract Name	produceReport()
Responsibilities	Produce Reports
Type:	System
Exceptions	none
Preconditions	Select filter to generate desire reports
Postconditions	Display Graphics and provide pdf downloads

Use Case	UC-15: ApproveStall
Contract Name	approveStall()

Responsibilities	Assign a stall to a vendor
Type:	System
Exceptions	none
Preconditions	Stall should be available for a vendor occupation
Postconditions	Save assign stall to vendor account

Use Case	UC-18: GenerateInvoices
Contract Name	produceInvoices()
Responsibilities	Produce Invoices for all vendors according stall fees and other fees charge
Type:	system
Exceptions	none
Preconditions	Vendor should have account and stall assign to their account
Postconditions	Invoices saved in the system

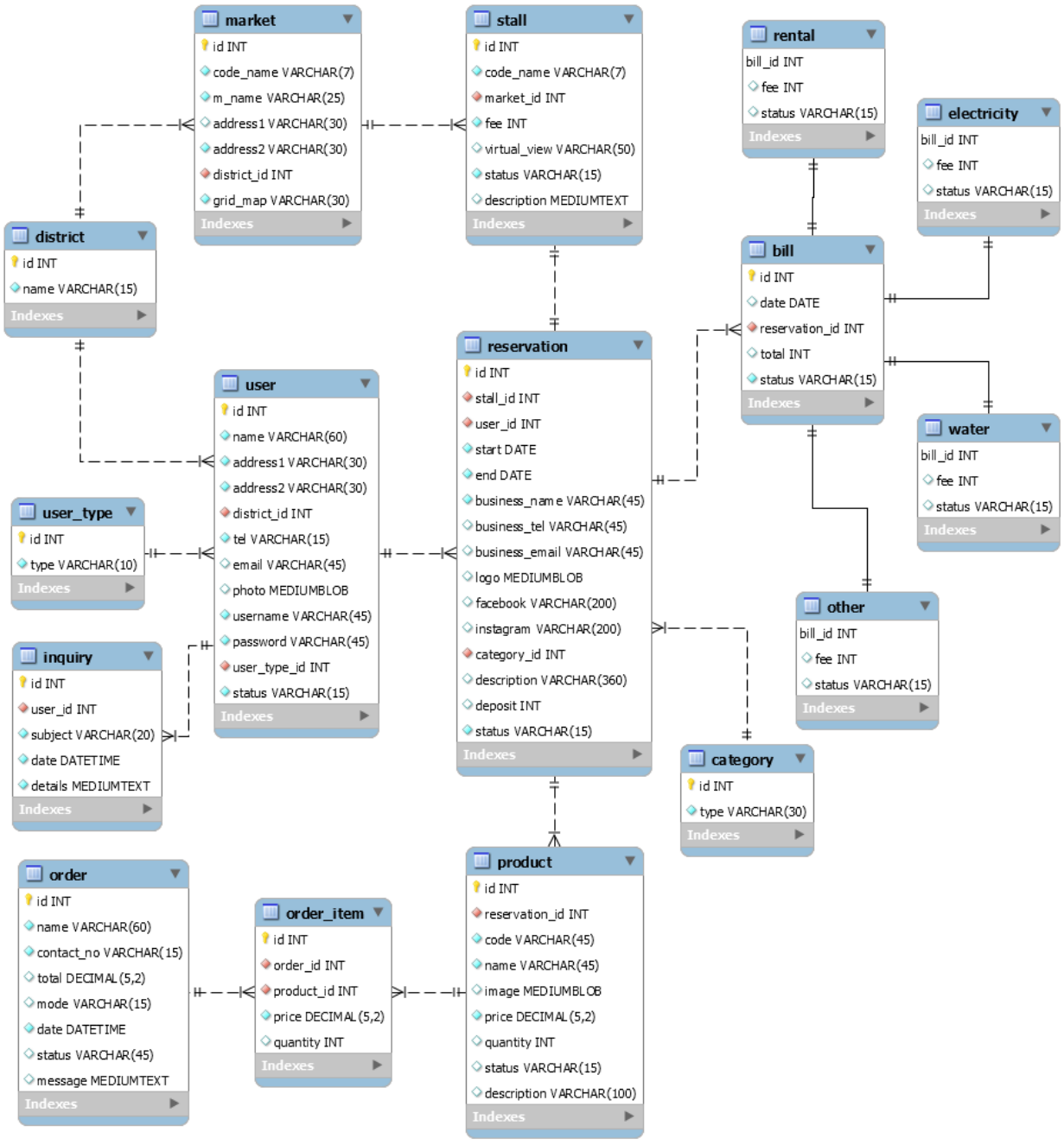
Use Case	UC-19: ViewInquires

Contract Name	getInquires()
Responsibilities	Display and List all Inquiries from vendors
Type:	System
Exceptions	none
Preconditions	Inquiries should be submit by vendors
Preconditions	Access to Inquiries to address issues.

Data Model and Persistent Data Storage

The Market Management System (MMS) consists of various data that must outlive a single execution. Therefore, the data will be saved in a database. A database server will be used to store and manage the data. Moreover, the main data the system wants to store is the user's credentials and vendors personal information as well as vendors product details. Below is a more detailed description of the data that will be stored.

Database Schema


market
id INT
code_name VARCHAR(7)
m_name VARCHAR(25)
address1 VARCHAR(30)
address2 VARCHAR(30)
district_id INT
grid_map VARCHAR(30)
Indexes
stall
id INT
code_name VARCHAR(7)
market_id INT
fee INT
virtual_view VARCHAR(50)
status VARCHAR(15)
description MEDIUMTEXT
Indexes
rental
bill_id INT
fee INT
status VARCHAR(15)
Indexes
electricity
bill_id INT
fee INT
status VARCHAR(15)
Indexes
district
id INT
name VARCHAR(15)
Indexes
bill
id INT
date DATE
reservation_id INT
total INT
status VARCHAR(15)
Indexes
reservation
id INT
stall_id INT
user_id INT
start DATE
end DATE
business_name VARCHAR(45)
business_tel VARCHAR(45)
business_email VARCHAR(45)
logo MEDIUMBLOB
facebook VARCHAR(200)
instagram VARCHAR(200)
category_id INT
description VARCHAR(360)
deposit INT
status VARCHAR(15)
Indexes
user
id INT
name VARCHAR(60)
address1 VARCHAR(30)
address2 VARCHAR(30)
district_id INT
tel VARCHAR(15)
email VARCHAR(45)
photo MEDIUMBLOB
username VARCHAR(45)
password VARCHAR(45)
user_type_id INT
status VARCHAR(15)
Indexes
water
bill_id INT
fee INT
status VARCHAR(15)
Indexes
user_type
id INT
type VARCHAR(10)
Indexes
other
bill_id INT
fee INT
status VARCHAR(15)
Indexes
inquiry
id INT
user_id INT
subject VARCHAR(20)
date DATETIME
details MEDIUMTEXT
Indexes
category
id INT
type VARCHAR(30)
Indexes
order
id INT
name VARCHAR(60)
contact_no VARCHAR(15)
total DECIMAL(5,2)
mode VARCHAR(15)
date DATETIME
status VARCHAR(45)
message MEDIUMTEXT
Indexes
order_item
id INT
order_id INT
product_id INT
price DECIMAL(5,2)
quantity INT
Indexes
product
id INT
reservation_id INT
code VARCHAR(45)
name VARCHAR(45)
image MEDIUMBLOB
price DECIMAL(5,2)
quantity INT
status VARCHAR(15)
description VARCHAR(100)
Indexes

Data Dictionary

District: use for storing district names		
Attribute	**Description**	**Sample**
id	Auto increment district number in the system	2
name	The name of the district (Belize = 1)	Cayo

User_type: use for storing user types		
Attribute	**Description**	**Sample**
id	Auto increment user_type number in the system	2
type	Specifies type of user (admin = 1)	Vendor

User: use for keeping information of users (admin and vendor)		
Attribute	**Description**	**Sample**
id	Auto increment user number in the system	2
name	User's name	John Doe
address1	The name of the street where user resides	Iguana Street
address2	The name of village, town or city	Belmopan

district_id	Identification number of district	2
tel	User's phone number	660-1234
email	User's email address	johndoe@gmail.com
username	User's username to login to the system	johndoe34
password	User's password to access the system	jonetwo12
user_type_id	User type identification number (vendor = 2)	2
status	Suspended = 0, Active = 1, Pending = 2	1

Market: use for recording the details of the market		
Attribute	**Description**	**Sample**
id	Auto increment market number in the system	1
code_name	Market's identifiable code name	SI-22
m_name	The name of the market	San Ignacio Market
address1	The street name where market is located	Main Road
address2	Village/Town/City where market is located	San Ignacio
district_id	Identification number of district	2
grid_map	The image name stored in the system that shows layout of the market	gridMap-SIM.png

Stall: use for recording the detail of the market stall		
Attribute	**Description**	**Sample**
id	Auto increment stall number in the system	1
code_name	Stall's identifiable code name	SI-AA
market_id	Market's identification number	1
fee	The amount charged for monthly rental	100
virtual_view	The name of the file stored in the system that provides users a virtual tour of the stall	v-tour-SI-AA.
status	Available = 0, Occupied = 1	1
Description	Brief description of the stall	Stall size is 10 by 10

Category: use for storing general category of product or service offered		
Attribute	**Description**	**Sample**
id	Auto increment category number in the system	1
type	Specifies category of product or service	Fruits and Vegetables

Reservation: use for recording the detail of market stall reservation		
Attribute	**Description**	**Sample**
id	Auto increment number in the system	1
stall_id	Identification number of market stall	1
user_id	Vendor’s Identification number	2
start	The date the reservation starts	2022-03-01
end	The date the reservation expires	2022-08-31
business_name	The business name of market stall	John’s Organic Fruits
business_tel	The business contact number	660-1234
business_email	The business email of vendor	jorganicfruit@gmail.com
facebook	The facebook link of the business	
instagram	The instagram link of the business	
category_id	Category identification number	1
about_us	A brief description of the business	We sell a variety of fresh, organic fruits.
deposit	The amount that the vendor paid when the reservation becomes active	100
status	Expired (0), Active (1) and Pending (2)	1

Bill: use for recording the amount of vendors monthly bill		
Attribute	**Description**	**Sample**
id	Auto increment bill number in the system	1
month	Month of the bill	April
year	Year of the bill	2017
reservation_id	Reservation number of the bill	1
total	Total amount of the bill	125
status	0 = Unpaid, 1 = Paid, 2 = Partial Paid	1

Rental: use for recording the amount of rental fee		
Attribute	**Description**	**Sample**
bill_id	Auto increment bill number in the system	1
fee	The amount of room fee	100
status	0 = Unpaid, 1 = Paid	1

Electricity: use for recording the amount of electricity fee (optional)		
Attribute	**Description**	**Sample**
bill_id	Auto increment bill number in the system	1
fee	The amount of electricity fee	0
status	0 = Unpaid, 1 = Paid	

Water: use for recording the amount of water fee (optional)		
Attribute	**Description**	**Sample**
bill_id	Auto increment bill number in the system	1
fee	The amount of water fee	25
status	0 = Unpaid, 1 = Paid	1

Other: use for recording the amount of other fee (optional)		
Attribute	**Description**	**Sample**
bill_id	Auto increment bill number in the system	1
fee	The amount of other fee (Late Payment fee)	0
status	0 = Unpaid, 1 = Paid	

Product: use for recording products or services sold in market stall		
Attribute	**Description**	**Sample**
id	Auto increment product number in the system	1
reservation_id	Identification number of reservation	1
code	Product code assigned by vendors	GV11
name	The name of the product	Guava
image	Uploaded image details	
price	The assigned price of product or service	2.00
quantity	The amount of product in stock	15
status	The availability of product or service	In Stock
description	A brief description of the product or service	Sold by pound

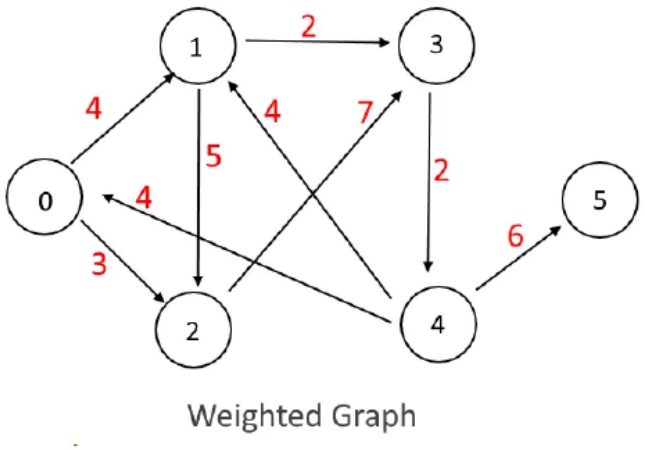
Order: use for recording items ordered/reserved by customers		
Attribute	**Description**	**Sample**
id	Auto increment order number in the system	1
name	The name of customer making the order	Marshall Matters
contact_no	Phone number of customer	623-4567
product_id	Identification number of product	1
quantity	Quantity of product ordered	2
total	The amount to be paid	4.00
mode	Pick up (P) or deliver (D) item	P
date	The date the order is created	2022-04-01 02:30:25.000000
status	The vendor confirms or cancels order	confirm
message	Note to vendor (e.g. delivery address)	Will pick up at 10 a.m.

Inquiry: use for recording details of inquiries sent from vendor to admin		
Attribute	**Description**	**Sample**
id	Auto increment inquiry number in the system	1
user_id	Identification number of user	2
subject	Short title of message	Jane’s number
date	The date the inquiry is created	2022-04-01 02:30:25.000000
details	The message of the user	Hello, I would like to request Jane's number.

Mathematical Model

Map

The primary mathematical model used in this map is the concept of 2D Graphs. More specifically, a 2D weighted graph. An example of such a graph can be seen below.

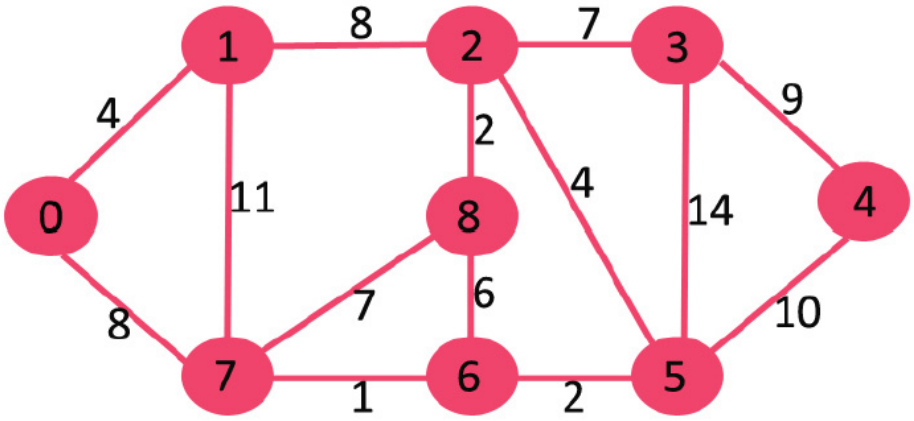


Weighted Graph

The graph is then displayed on the Google Map Interface at specified coordinates.

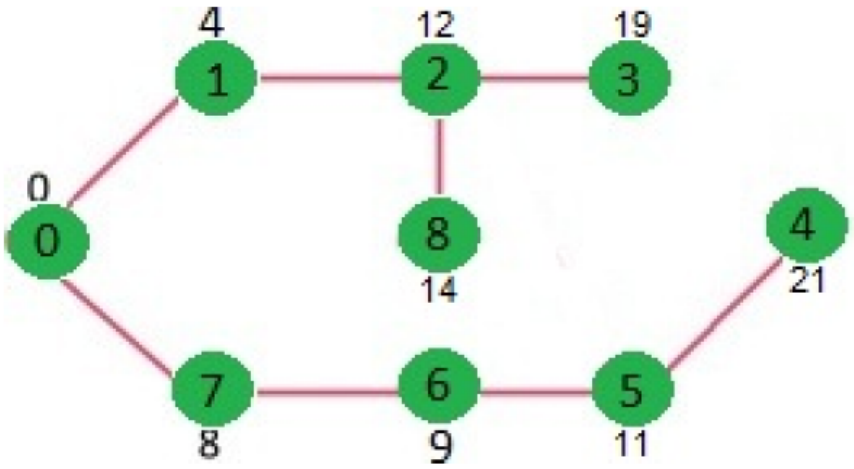
Route

The routes for the graph are made by the Google Directions API using Dijkstra's Algorithm. Dijkstra's Algorithm finds the shortest path to all vertices from a source vertex. An example of a completed shortest path tree can be seen below.



This shortest tree path is then used by the Google Directions API to find the shortest path between two locations.

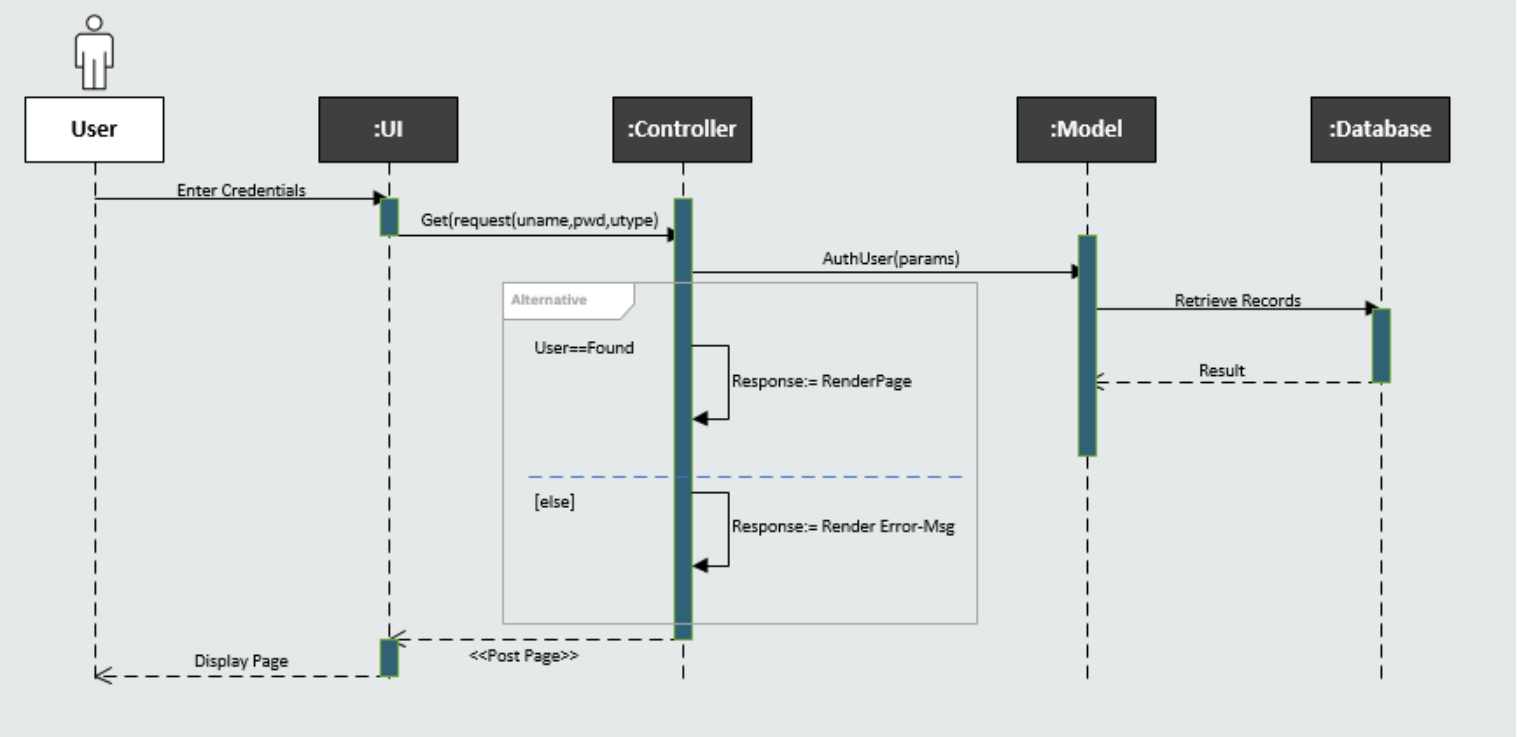
Base Weighted Graph



Shortest Tree Path Based On Previous Weighted Graph

Interaction Diagrams

Sequence Diagram for UC-1: AuthVendorandManagement

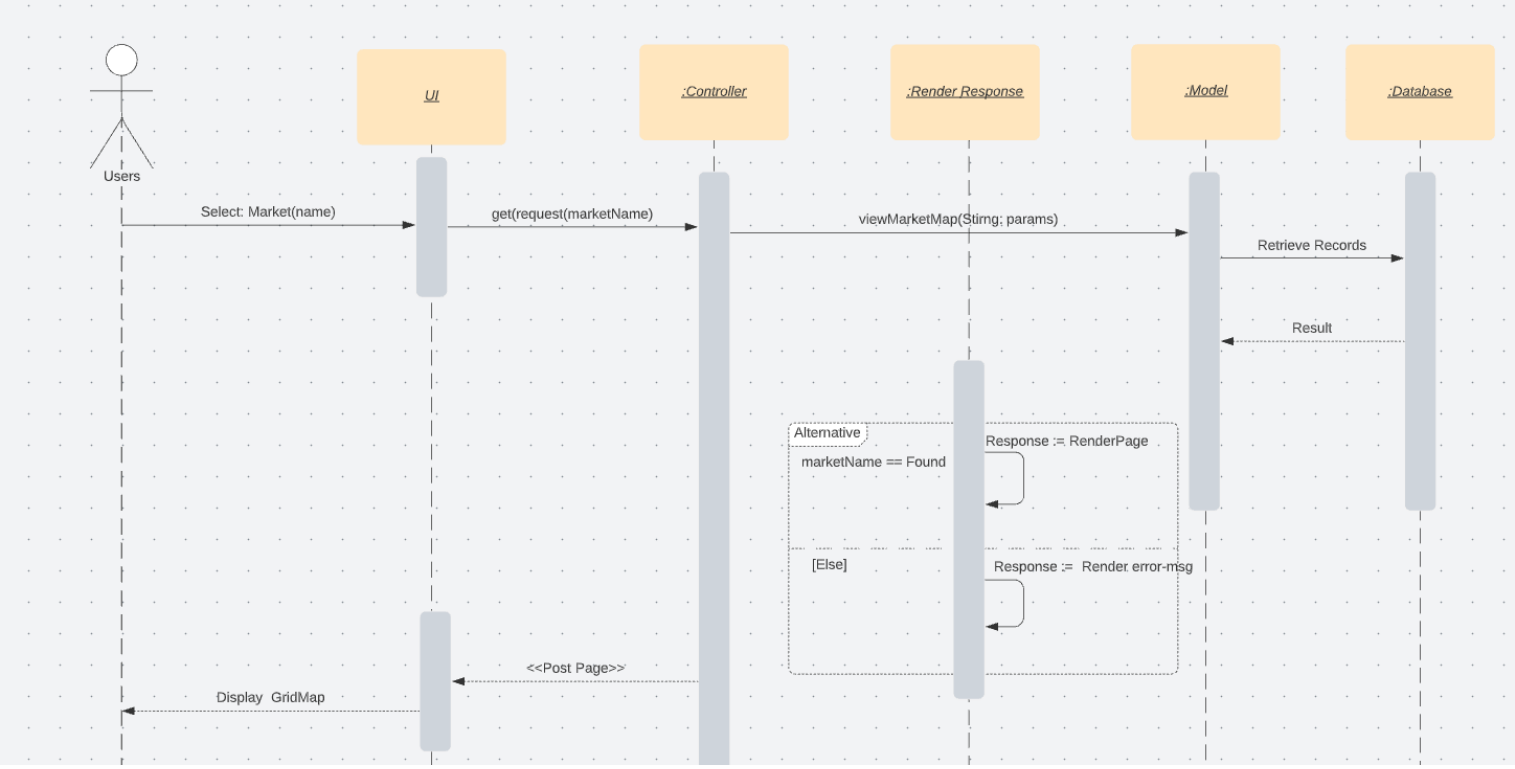


Description for UC-1:

AuthVendorandManagement is the task that checks an account's credentials to make sure that they are a valid user. A user will go to the login page and enter their credentials and press login. This sends a get request to the CONTROLLER for the username, password and user type.

It passes this through AuthUser(params) through the MODEL. The CONTROLLER has two cases: (1) where it renders the page and (2) or gives an error message based on if the user was found. DATABASE is where the records are stored and retrieved, it also produces the results through the MODEL. The CONTROLLER then posts the necessary page to the UI which displays the page. This interaction diagram evolves from the system sequence diagram: AuthVendorandManagement.

Sequence Diagram for UC-2: ViewMarketMap

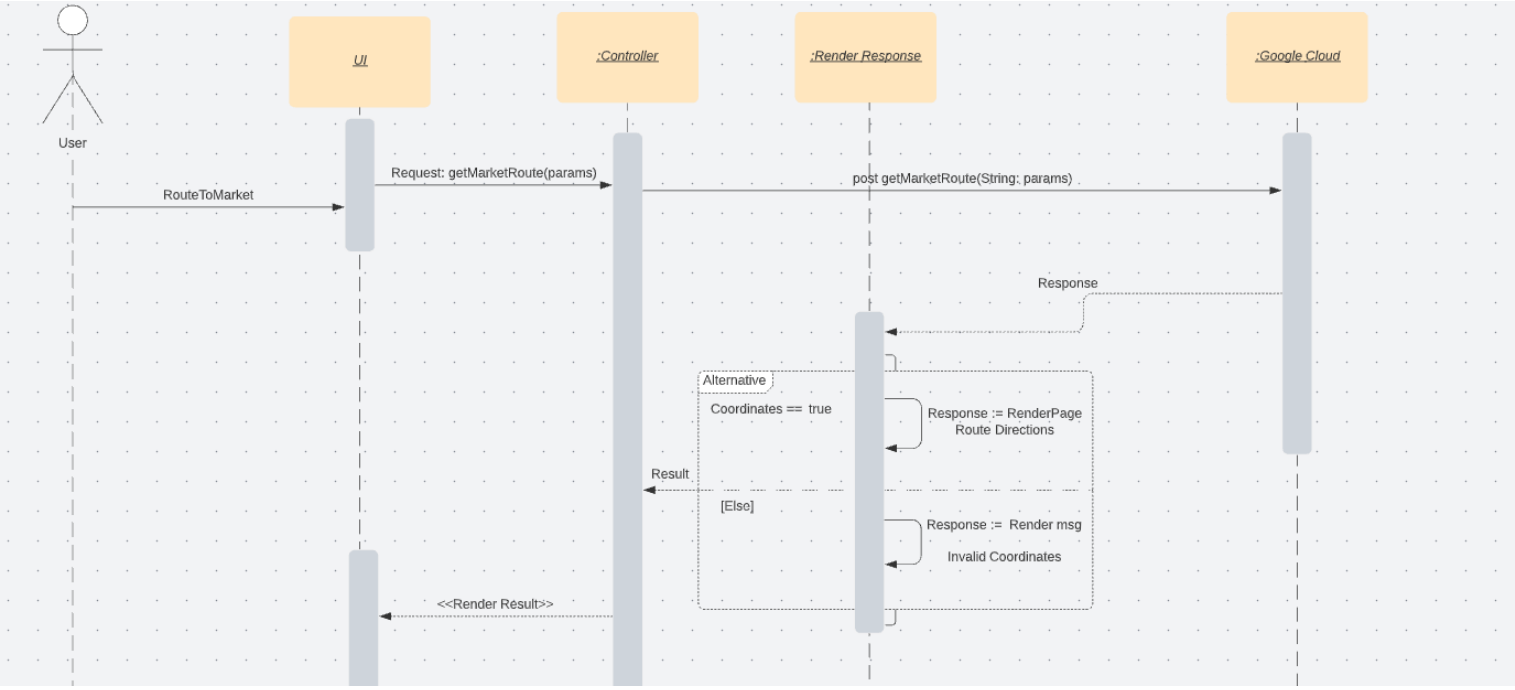


Description for UC-2:

ViewMarketMap provides a grid map of the whole market to view the different sections of the market. From the main screen the user will select a market by its name through the UI. This sends a get request to the CONTROLLER for the market name.Continuing with this name as the parameters, RENDER RESPONSE then calls view market map to MODEL.

If the market's name was found then it will generate the page for the whole market map however, if the market name does not exist then it will generate an error message.DATABASE is where the records are stored and retrieved, it also produces the results through the MODEL. The CONTROLLER then posts the necessary page to the UI which displays the page. This interaction diagram evolves from the system sequence diagram:ViewMarketMap.

Sequence Diagram for UC-3: GetMarketRoute

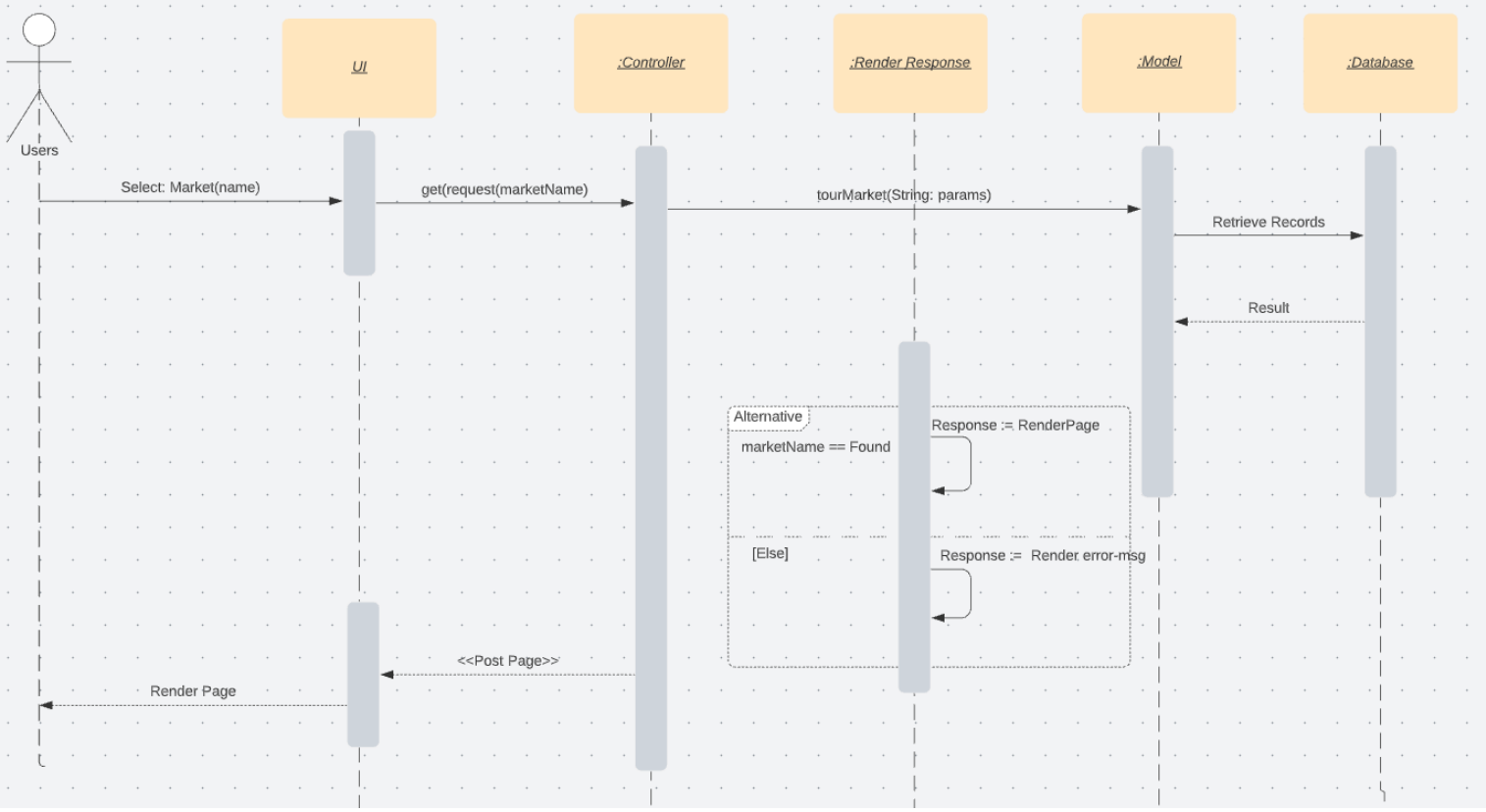


Description for UC-3:

GetMarketRoute will provide directions from the user's current location to the nearest local market. The user will click on a button on the UI first to grant access to obtain the user’s current location then on the map it will populate several nearby locations. The user will select the desired location. The data entered on the UI will be gathered and passed to the controller to handle the data.

CONTROLLER will then process to call google api and receive a response from GOOGLE CLOUD. The response will be handled to verify any error or missing data. Once the response return has no error message the controller will render the direction route to the UI screen. This interaction diagram evolves from the system sequence diagram: GetMarketRoute.

Sequence Diagram for UC-4: TourMarket

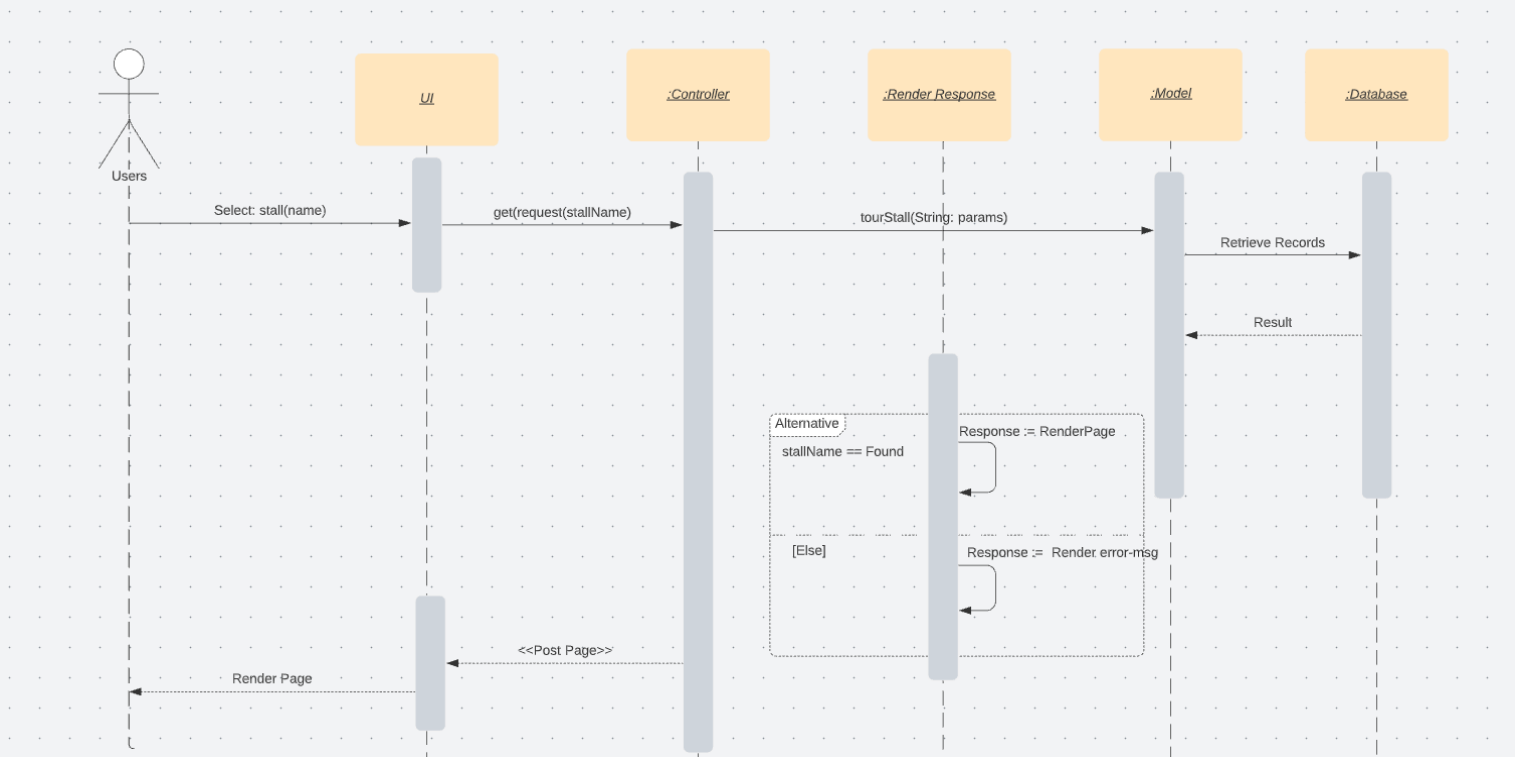


Description for UC-4:

TourMarket provides a view of the different sections of the market. From the main map the user will select a market by its name through the UI. This sends a get request to the CONTROLLER for the market name.Continuing with this name as the parameters, RENDER RESPONSE then calls tourMarket to MODEL.

If the market's name was found then it will generate the page for the sectioned map however, if the market name does not exist then it will generate an error message.DATABASE is where the records are stored and retrieved, it also produces the results through the MODEL. The CONTROLLER then posts the necessary page to the UI which displays the page. This interaction diagram evolves from the system sequence diagram: TourMarket.

Sequence Diagram for UC-5: TourStall

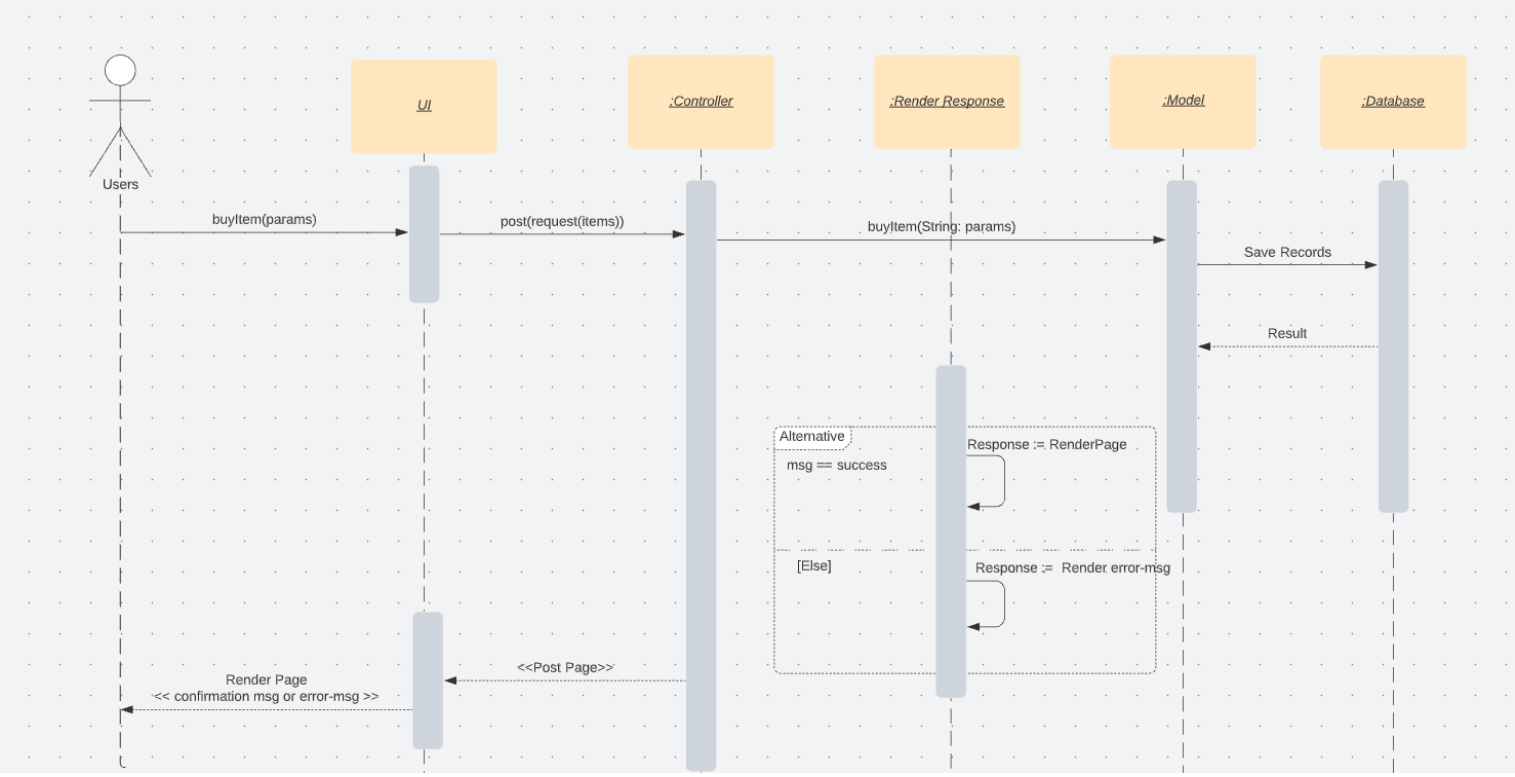


Description for UC-5:

TourStall provides a view of the different stalls of the sections in the market. From the sectionscreen the user will select a stall by its name through the UI. This sends a get request to the CONTROLLER for the stall name.Continuing with this name as the parameters, RENDER RESPONSE then calls tourStall to MODEL.

If the stall's name was found then it will generate the page for the specific stall however, if the stall name does not exist then it will generate an error message.DATABASE is where the records are stored and retrieved, it also produces the results through the MODEL. The CONTROLLER then posts the necessary page to the UI which displays the page. This interaction diagram evolves from the system sequence diagram: TourStall.

Sequence Diagram for UC-7: BuyItem

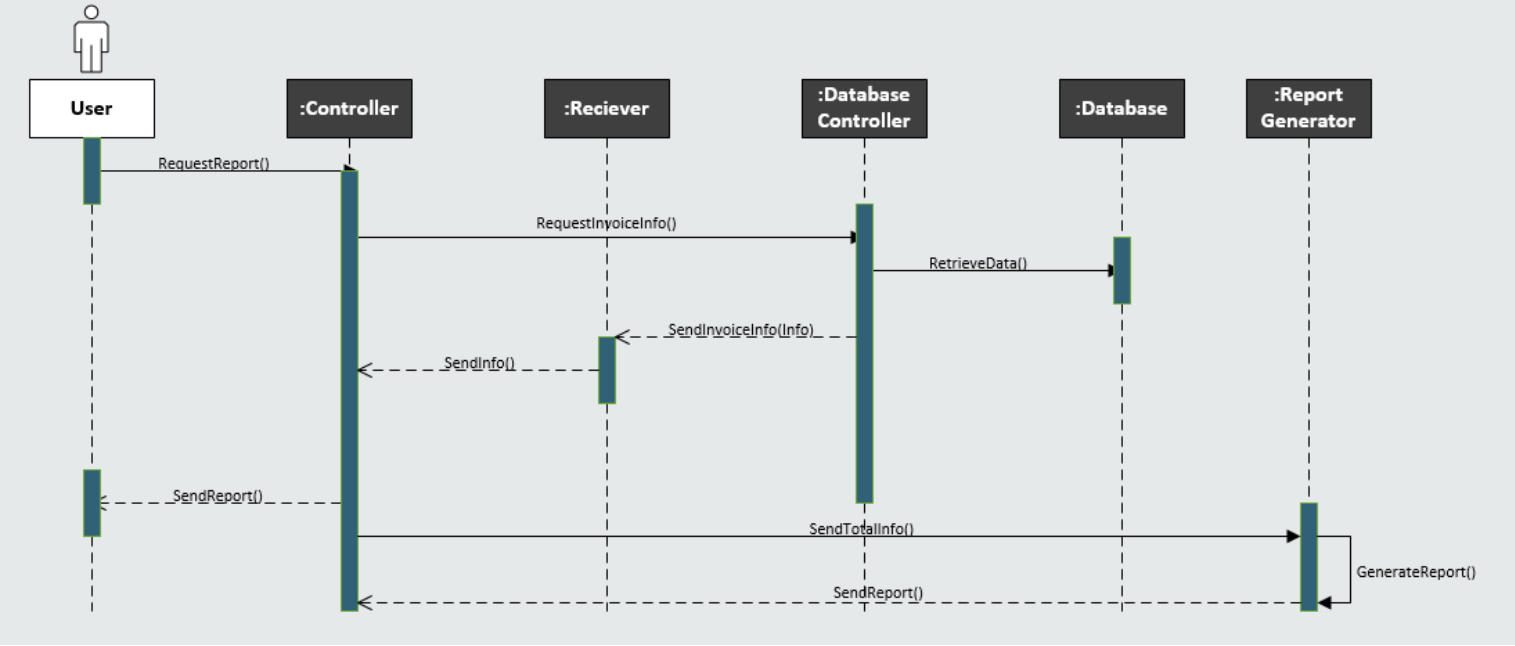


Description for UC-7:

BuyItem reserves a product from the listing products of each stall. From the stall the user will select a stall's item by its parameters and send them to the UI. This sends a post request to the CONTROLLER for the item.Continuing with these parameters, RENDER RESPONSE then calls buy item to MODEL. If the msg was a success then it will generate the page for the specific stall however, if the message failed then it will generate an error message.

DATABASE is where the records are stored and retrieved, it also produces the results through the MODEL. The CONTROLLER then posts the necessary page to the UI which renders the page which is either a confirmation or an error message. This interaction diagram evolves from the system sequence diagram: BuyItem.

Sequence Diagram for UC-11: GenerateReport

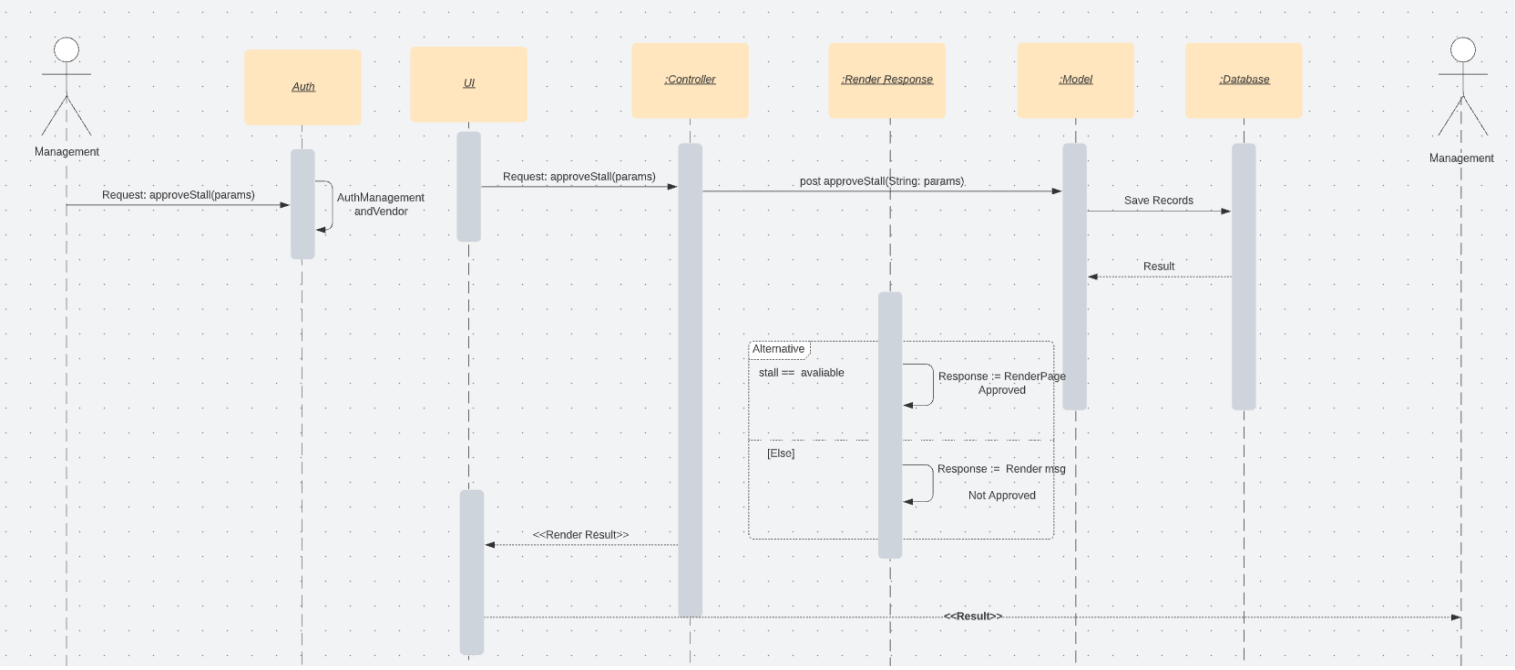


Description for UC-11:

GenerateReport is responsible for retrieving the costs a certain vendor incurs over a session. The vendor requests to view all the costs they owe to management. The CONTROLLER receives the request and tells the database that the following data are needed. To get the data, the CONTROLLER sends a request to Database via DATABASE CONTROLLER and the data requested is then sent to the RECEIVER.

The receivers then send the data to the CONTROLLER, which is then compiled, and sent to the REPORT GENERATOR. The REPORT GENERATOR generates a report from the information received from the CONTROLLER and the information is then sent to the vendor for viewing. This interaction diagram evolves from the system sequence diagram: GenerateReport.

Sequence Diagram for UC-15: ApproveReservedStall

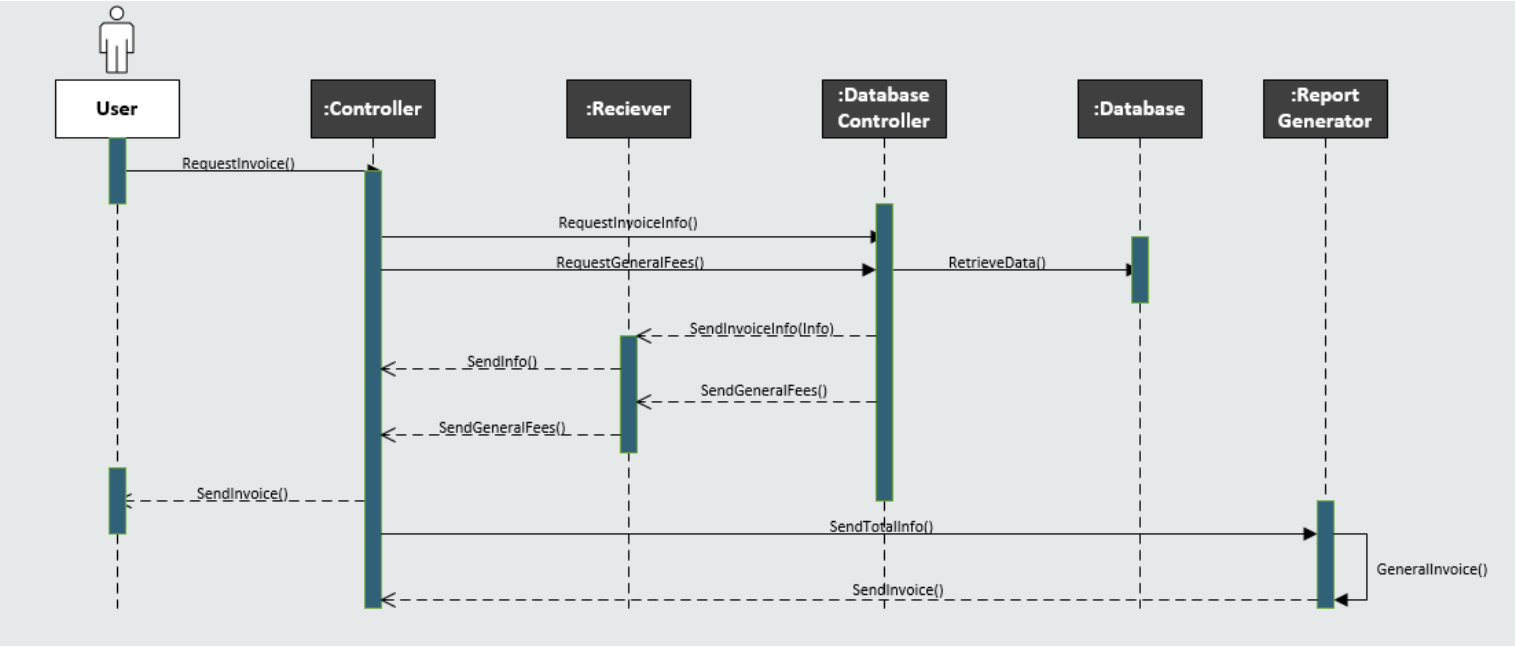


Description for UC-15:

ApproveReservedStall allows management to be able to approve the request stall by a vendor. The Management user will first authenticate to access the system so the AUTH will be the first action to approve the stall. Management will select the stalls they will approve of on the UI. The request data will be passed to the CONTROLLER to execute the request.

The CONTROLLER will pass the request to the MODEL to communicate with the database and to run the query request. The result is sent to the MODEL then passed to RESPONSE Render to verify if any error exists in the response result. Result is passed to the CONTROLLER to handle the result and lastly display the response on the UI. This interaction diagram evolves from the system sequence diagram:ApproveReservedStall.

Sequence Diagram for UC-18: GenerateInvoices

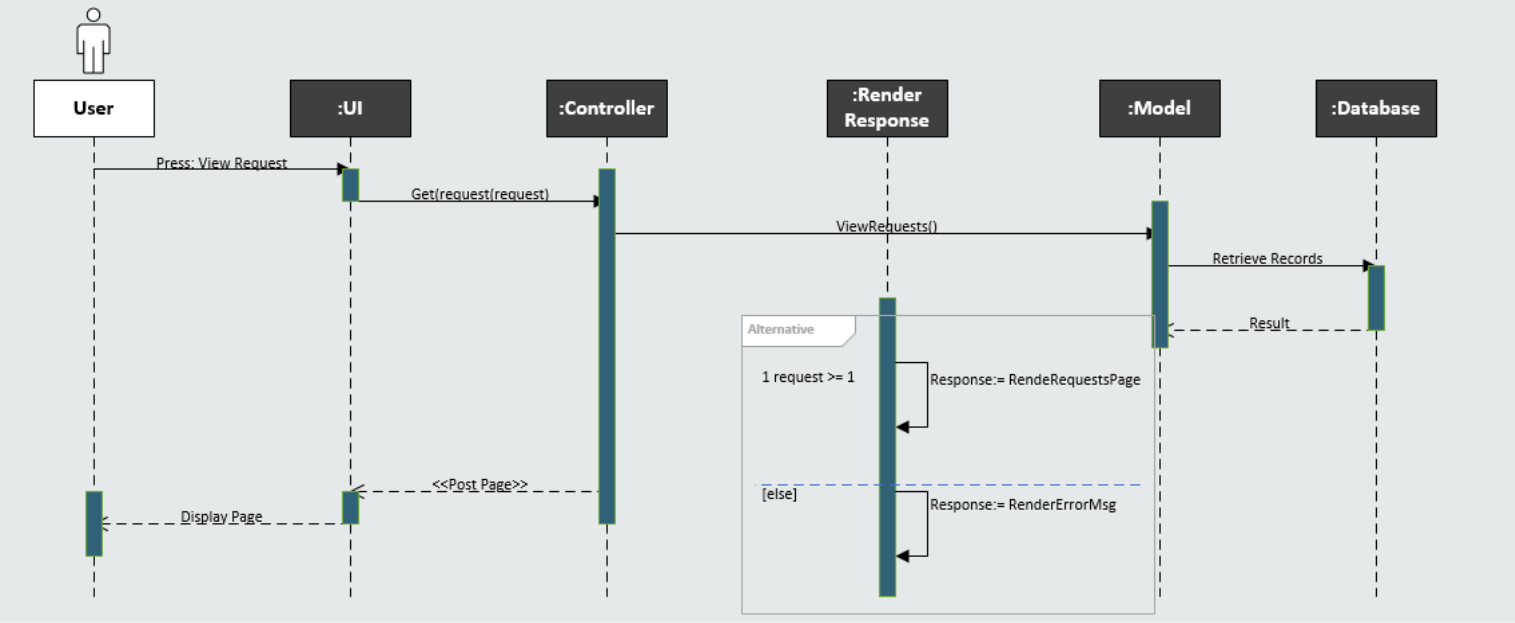


Description for UC-18:

GenerateInvoices Management will generate an invoice on vendor stall usage and other fees. The management makes a request to view all the costs of a specific vendor. The CONTROLLER receives the request and tells the database that the following data are needed. To get the data, the CONTROLLER sends 2 requests to Database via DATABASE CONTROLLER and the data requested is then sent to the RECEIVER.

The receivers then send the data to the CONTROLLER, which is then compiled, and sent to the REPORT GENERATOR. The REPORT GENERATOR generates a report from the information received from the CONTROLLER and the information is then sent to management for viewing. This interaction diagram evolves from the system sequence diagram: GenerateInvoices.

Sequence Diagram for UC-19: ViewInquires



Description for UC-19:

ViewInquires allows management to be able to view requests or inquiries from vendors. The Management user will first authenticate to access the system so the AUTH will be the first action to view the requests/inquiries. Management will select the view requests button on the UI. The request will be passed to the CONTROLLER to execute the request.

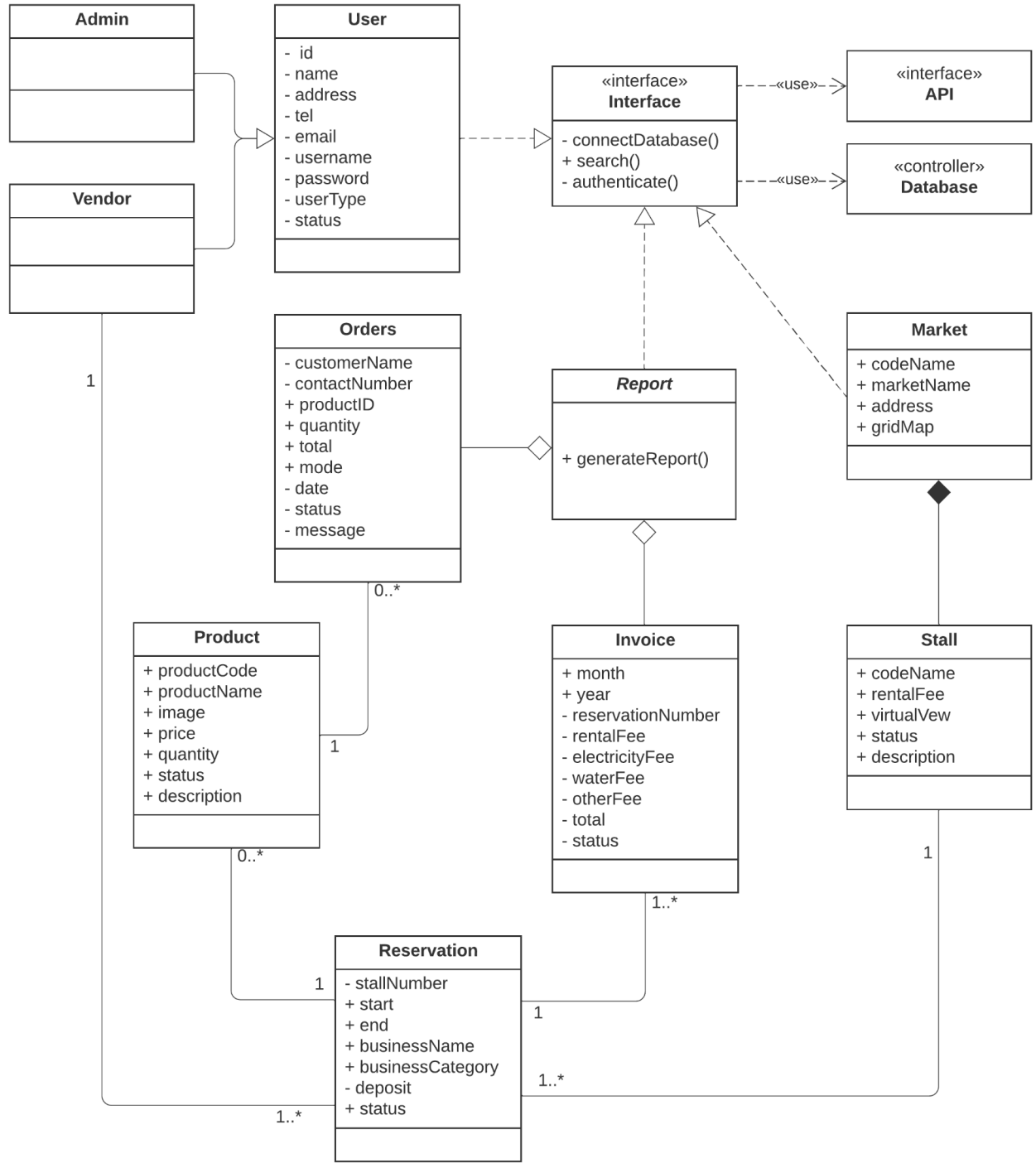
The CONTROLLER will pass the request to the MODEL to communicate with the database and to run the query request. The result is sent to the MODEL then passed to RESPONSE RENDER to verify if any error exists in the response result. Result is passed to the CONTROLLER to handle the result and lastly display the response on the UI. This interaction diagram evolves from the system sequence diagram: ViewInquires.

With the singleton design pattern in mind, we ensured that an object is shared globally with the resource. The one object thats shared amongst the other resources without needing to recreate that object or lose information.

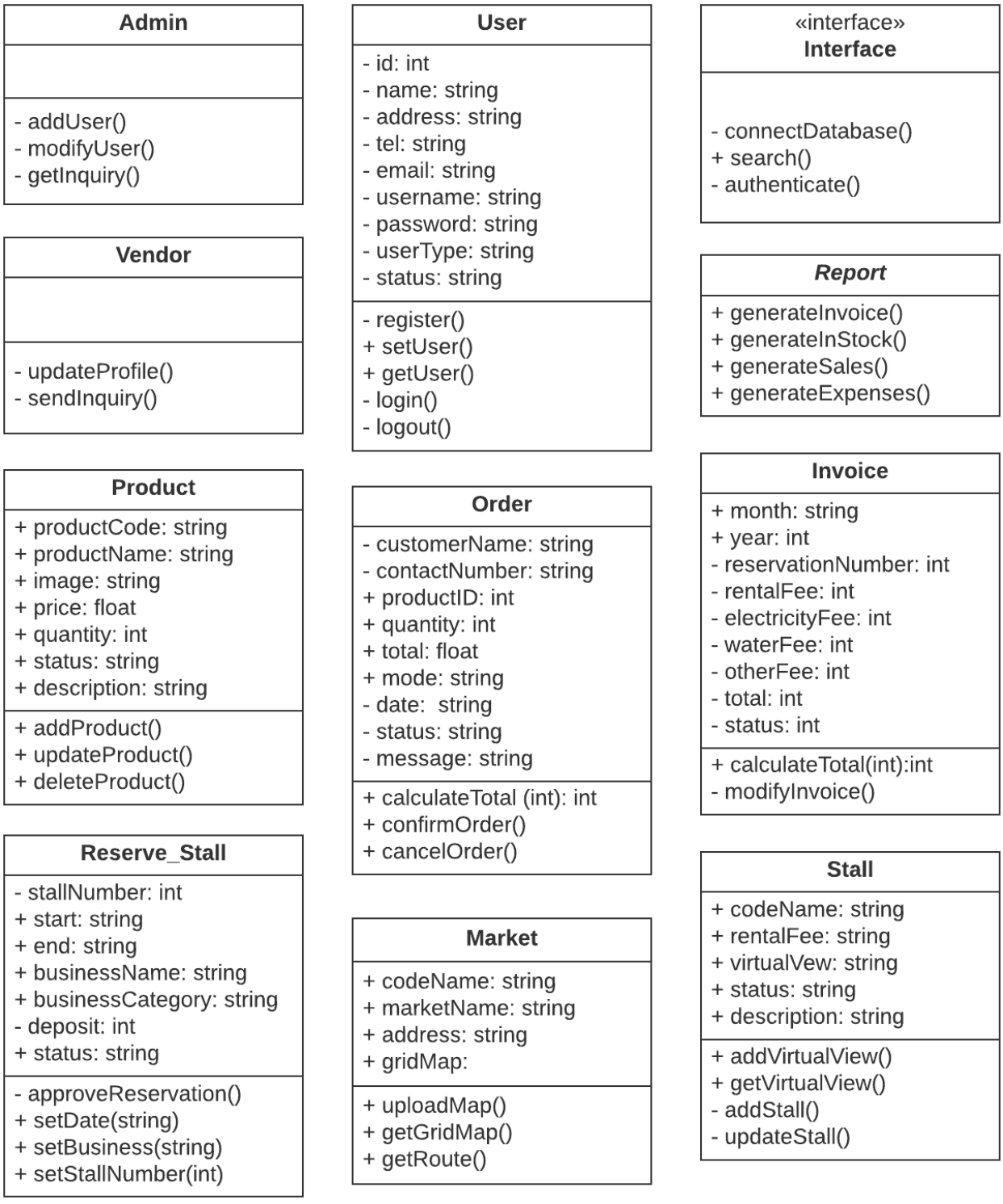
Function

Class Diagram and Interface Specification

Class Diagram:



Data Types and Operation Signatures:



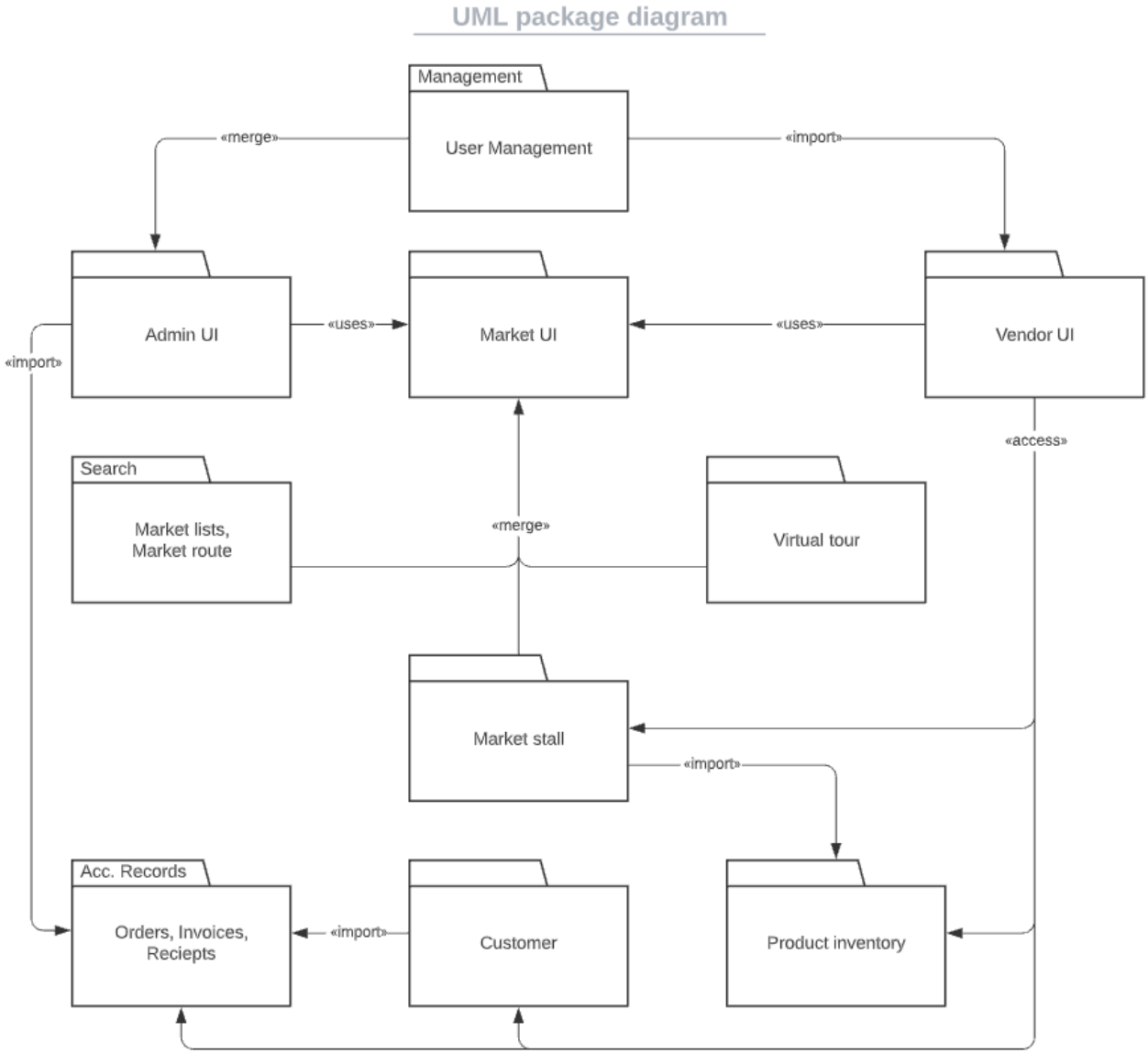
Traceability Matrix:

	Admin	Vendor	User	orders	products	Reserve stall	market	interface	report	Invoice	stall
Controller								x	x	x	
User Interface Page							x	x	x	x	x
User Data	x					X			x	x	x
Database Connection					x		x		x	x	x
Map Maker								x			
Current Location			x					x			
Destination Location			x					x			
Page Maker							x	x			

Search Request	x	x	x					x			
Status	x	x	x			x	x				
Buy Item Request		x	x		x		x				
Report Request	x	x	x						x	x	
Notification Request		x	x					x			

System Architecture

Subsystems:

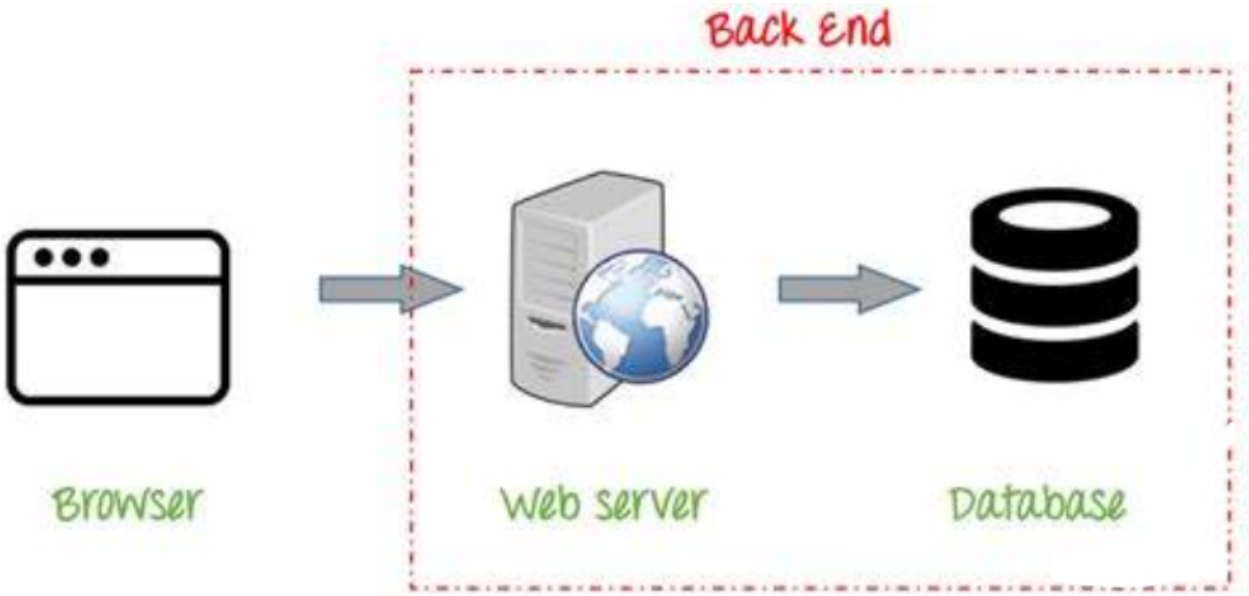


- ***Market UI:*** The primary functionality would be to display details of market stalls and the products. The UI will include search and registration functionality.
- ***Admin UI:*** The Admin UI manages the availability of market stalls. It provides the ability to make updates, generate reports and monitor overall user interaction or requests.
- ***Vendor UI:*** This UI allows vendors to display their products in their assigned stall. Vendors can monitor customer orders or purchases. They have direct interaction with customers.
- ***User Management:*** Manages vendor account details and market stalls.

- ***Search:*** Users can be able to search for a particular market, which includes location and route.
- ***Virtual Tour:*** Customers can tour market stalls in a 360 view format.
- ***Market Stall:*** Provides detail of each market stall including current products or services offered
- ***Product Inventory:*** A database will be help vendors keep track of products
- ***Customer:*** Stores customer information
- ***Accounting Records:*** Stores proof of payment or purchase order documents; inquiries and requests forms

Architecture Styles:

The software architectural style used in the design is a three-tier architecture (Client-Server): Presentation, application and data. The presentation tier consists of Admin, Market and Vendor UI where it displays information relating to services such as market details, virtual tours and products offered, respectively. The application tier includes the functionality of generating reports or searching market details and processing vendor, customer requests or orders. Lastly, the data tier would include the storage of vendor details, market details on a local server. The market stall, orders, fees and payments in the design would fall under the data tier.



Mapping Subsystems to Hardware:

The Market Management System will have subsystems running on multiple machines. The system will run on three(3) separate interfaces - the Customer, Vendor and Management.

- Any device capable of supporting a web browser can be used by the Customer or Vendor. For example, all modern smartphones and tablets can be used because they are equipped with more than sufficient hardware capabilities.

- Any computer meeting the hardware requirements as stated in the section titled, ‘System Architecture -Hardware Requirements’ can be used by the Manager.

- The web-server and database server will be run on a central machine that will be managed by the Administrator.

Connectors and Network Protocols:

The Market Management will be hosted on one server running MySQL database and Apache web server. With the use of any modern web browser, the clients (users) will be able to access the system from anywhere using the HTTP web protocol. HTTP was chosen because it is cost effective unlike HTTPS which requires the purchase of a security certificate. Furthermore, it uses less resources unlike HTTPS which consumes more CPU cycles and memory per request. This may not be an important factor on the client side but on the server side every little bit helps to maintain performance particularly with the number of country-wide requests.

Moreover, the HTTP protocol will allow transactions and data to be uploaded on the website, system, and database. Additionally, the application will use NTP services from an external server to maintain accurate time for all records and logs.

Global Control Flow:

Time-dependency

The system is a real-time system. Its whole purpose revolves around real-time responses which then determine the course of action to take and how quickly to take it.

E*xecution-orderliness*

The system is designed to be event-driven where every user, vendors and customers in particular, can interact with the system and perform actions in different order. We intend to implement a timer in the login function of the marketing management system. After a specific amount of inactive time period has passed, the system will automatically log out the vendor or the administrator.

Hardware Requirements:

The marketing management system will require a local server and PC with a minimum of 500 GB of space, 4gb of memory, 2.0 GHZ or above of processor speed. Windows 7 or above is the preferred operating system. For administrator(s) to monitor and maintain the system, at least 10mbps of stable internet connection is required.

Algorithms and Data Structures

The system primarily consists of a pathing algorithm in order to provide the shortest path between two locations on the map. Namely, location of the customer and location of the market. The algorithm will be responsible for identifying this shortest path in order to ensure efficiency of travel. This algorithm will take place using an external API, namely, Google Directions.

The system will be in charge of receiving both locations from the two independent parties, then sending it to the API. When the system retrieves the API response, the system will then show the customer the shortest route between the two locations.

Algorithm:

Market customer requests shortest route to market:

While “requesting_shortest_route” is true

Set “Market Location” to Current Location

If cannot get Current Location

Inform user that Locations services is disabled or Internet is not available

continue

End if

Send Current Location and Customer Location to Directions API

Fetch Shortest Route from API

Display Shortest Route on map view

End While

According to Crovari in an article from 2019, Google Directions uses Dijkstra's Algorithm to find the shortest route.

Algorithm Dijkstra(W[1..n, 1..n])

// Shows Dijkstra's Algorithm to find the shortest route in a weighted graph. According to GeeksforGeeks

// Input: A weighted graph W

// Output: Shortest Path Tree

Create empty set S that will hold shortest path tree

Assign all vertices distance values of INF

Assign source vertex distance value of 0

While S does not have all vertices

a.) Pick unrouted vertex U

b.) Include U in S

c.) Update shortest path tree with shortest path to vertex U

End While

Return S

Using this shortest path tree, the Directions API returns the shortest path between two locations.

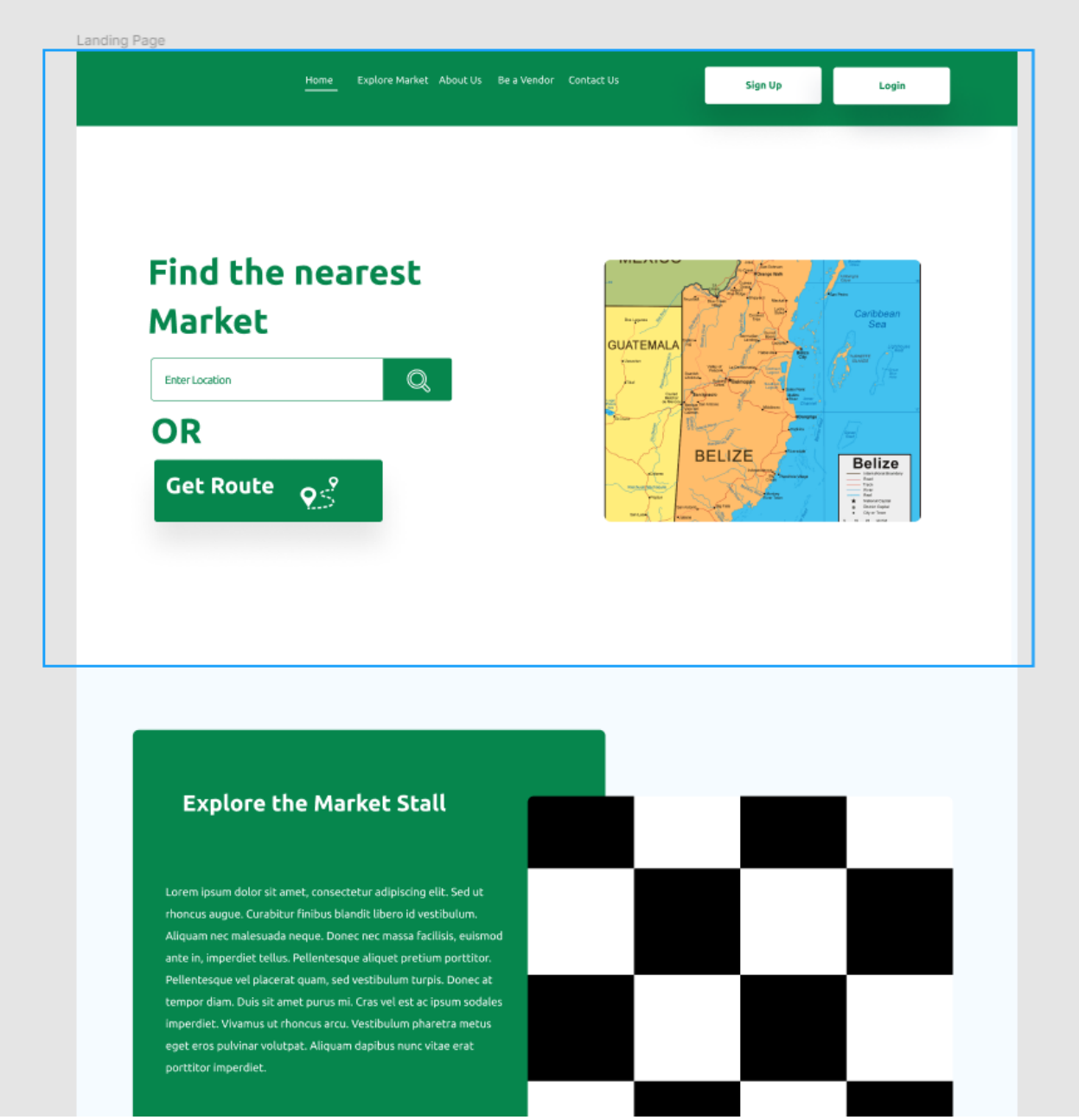
Concurrency

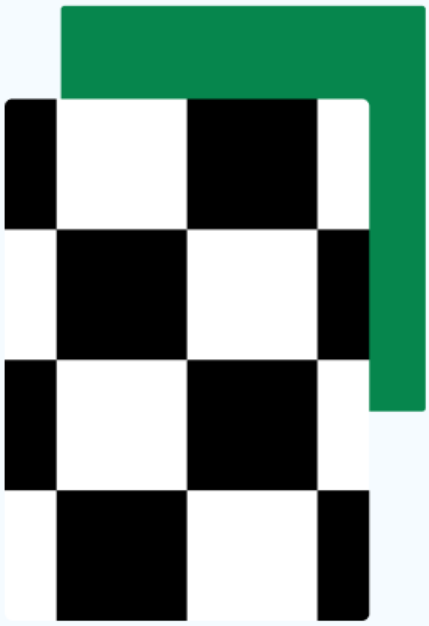
This system will allow for multiple users to be using the system at the same time. Since Users will be able to affect data on the system, an exception needs to be put in place that will allow for data to remain consistent throughout the system lifetime. This case specifically applies to different customers attempting to view and purchase items from the same market stall in a short amount of time.

Especially with poor internet connectivity, like what is normally experienced by users of 3G in Belize. In order to synchronize the data, and allow for the information to remain consistent, as well as to not double purchase items, a check within the System will ensure that no other customer has already purchased items twice.

User Interface Design and Implementation

Customer Main Page:

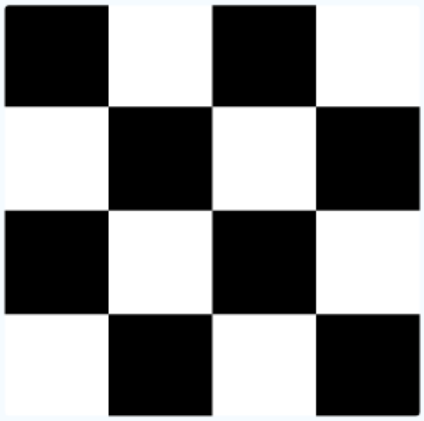




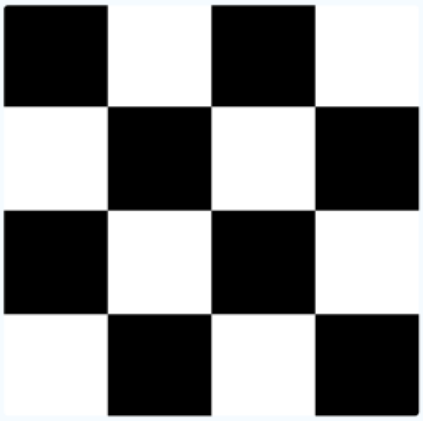
About US

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed ut rhoncus augue. Curabitur finibus blandit libero id vestibulum. Aliquam nec malesuada neque. Donec nec massa facilisis, euismod ante in, imperdiet tellus. Pellentesque aliquet pretium porttitor. Pellentesque vel placerat quam, sed vestibulum turpis. Donec at tempor diam. Duis sit amet purus mi. Cras vel est ac ipsum sodales imperdiet. Vivamus ut rhoncus arcu. Vestibulum pharetra metus eget eros pulvinar volutpat. Aliquam dapibus nunc vitae erat porttitor imperdiet.

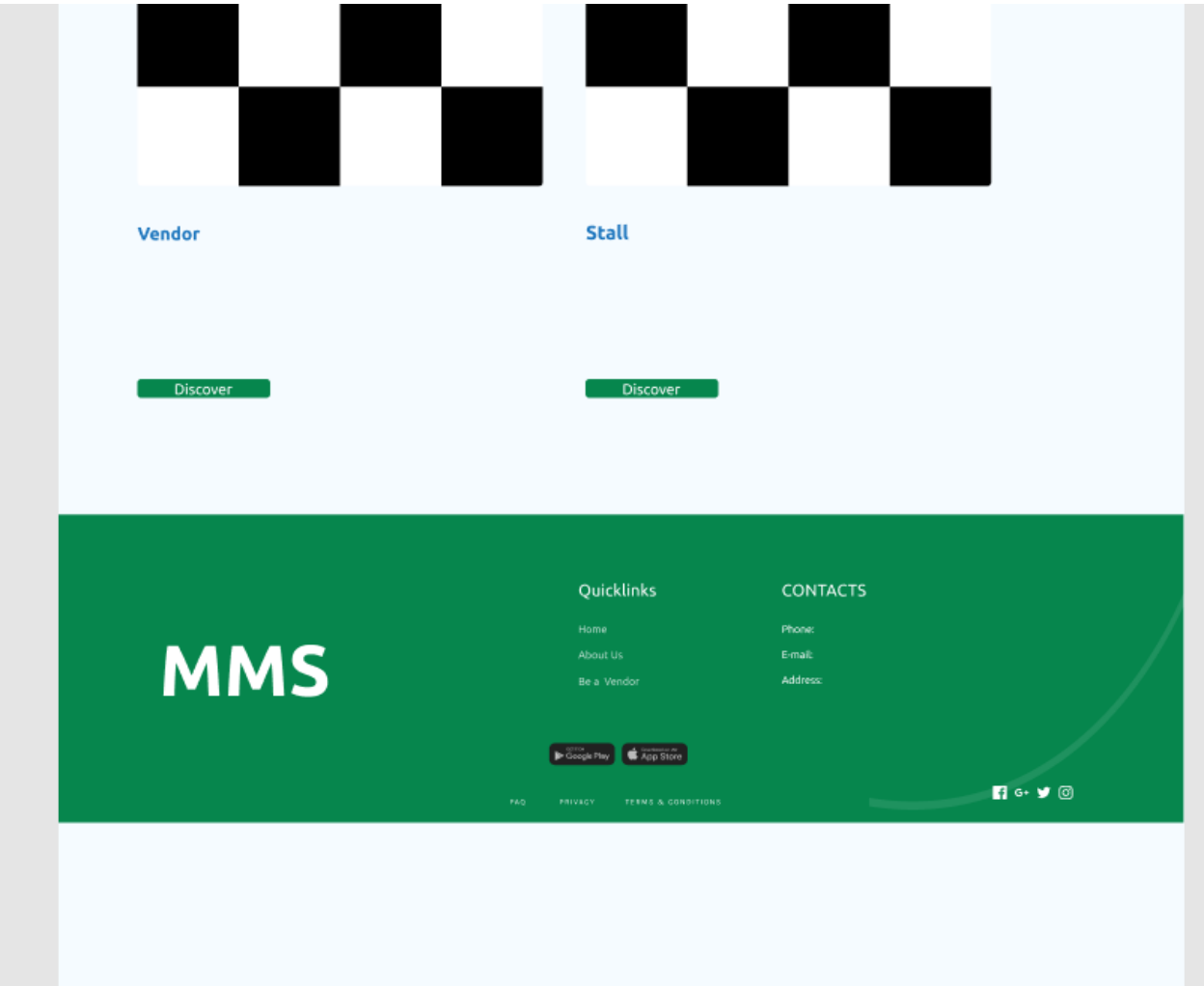
Be a Vendor



Vendor



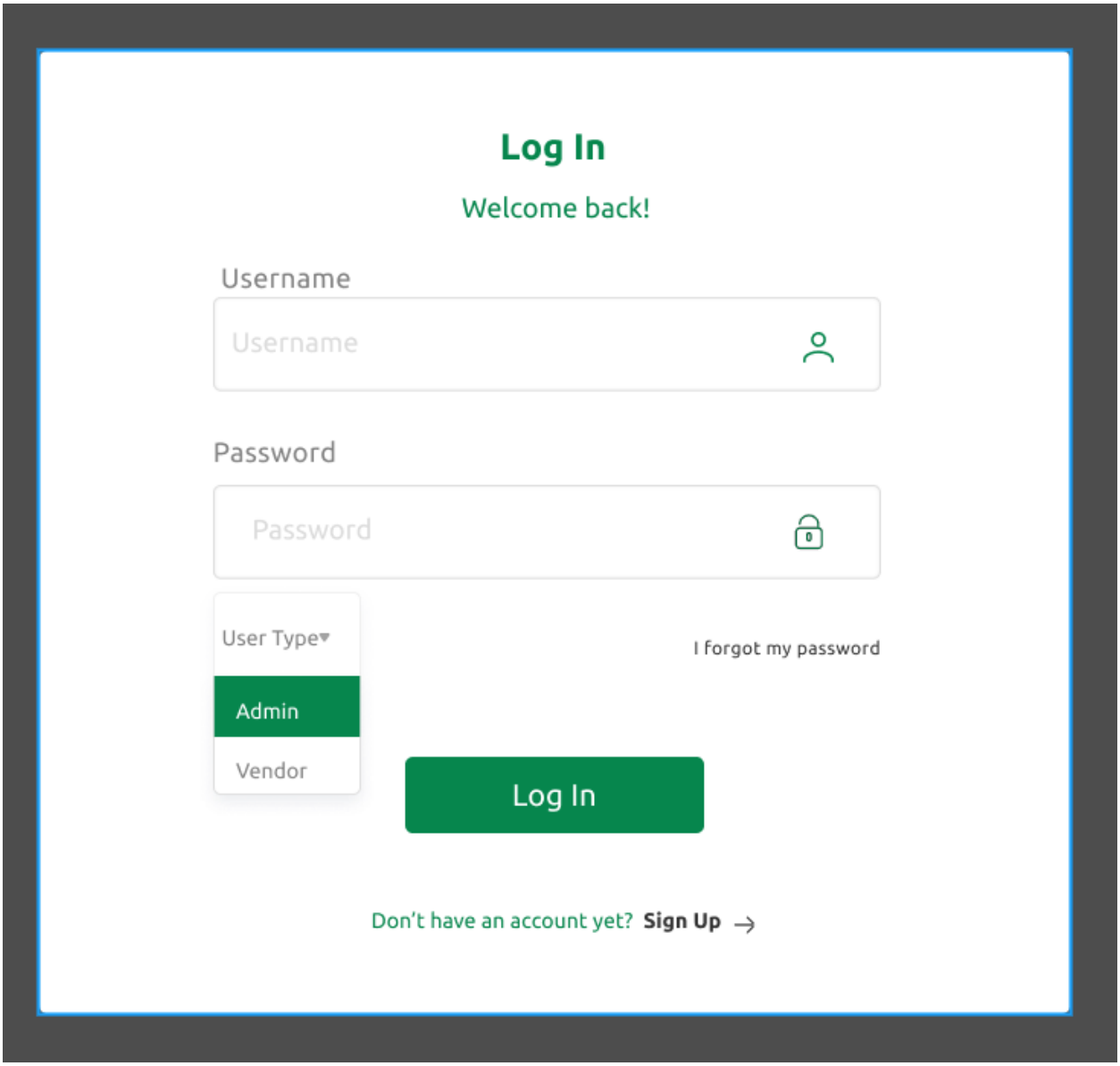
Stall

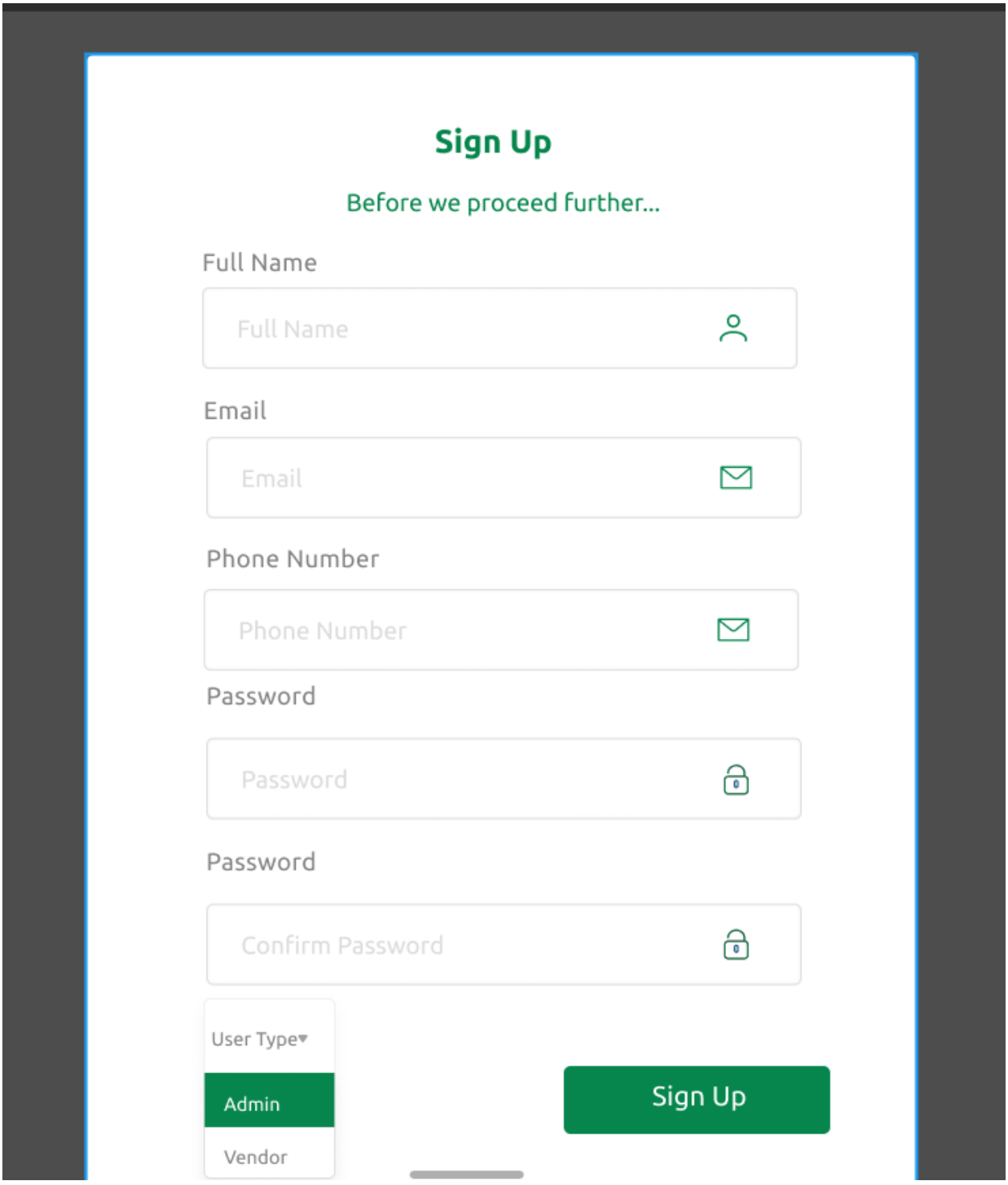


- The diagrams above display the customer's main page with additional info about the system and other services offered to vendors.

AuthVendorAndManagement:

- The design below displays the login and sign-up screen for vendors and management to access the system.



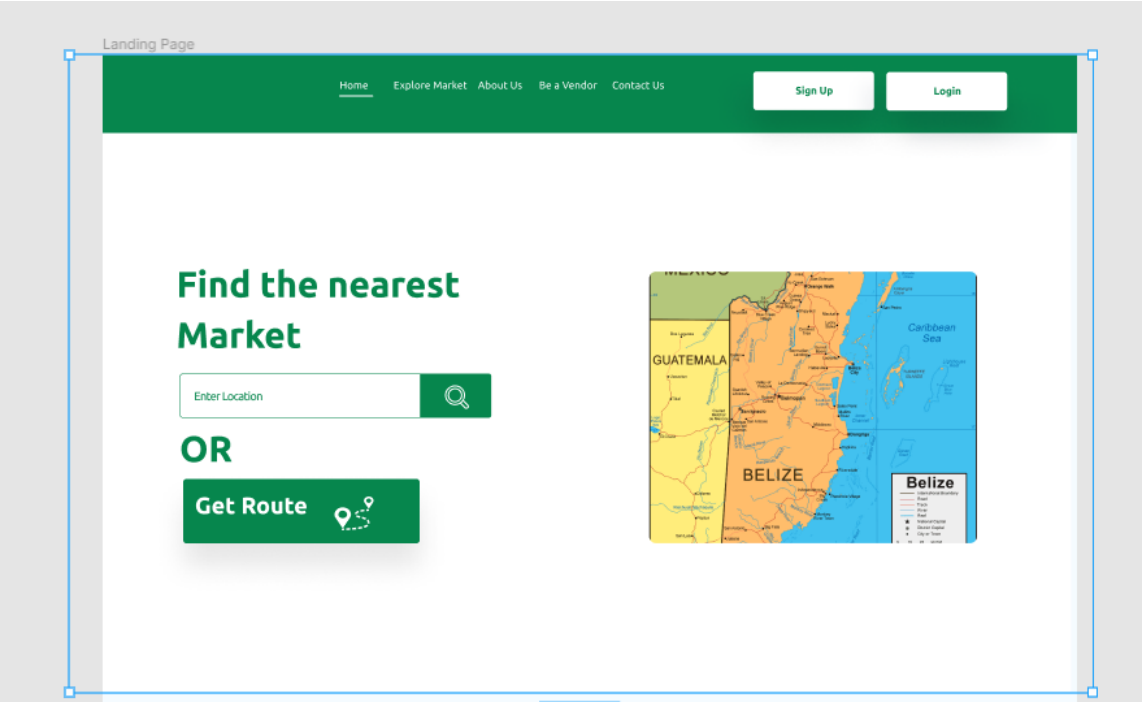

Sign Up
Before we proceed further...
Full Name
Full Name
Email
Email
Phone Number
Phone Number
Password
Password
Password
Confirm Password
User Type
Admin
Vendor
Sign Up

View Market Map

- The design below displays a market map screen where the customer can view the market in 360 modes.

Get Market Route

- The design below displays the route directions screen.

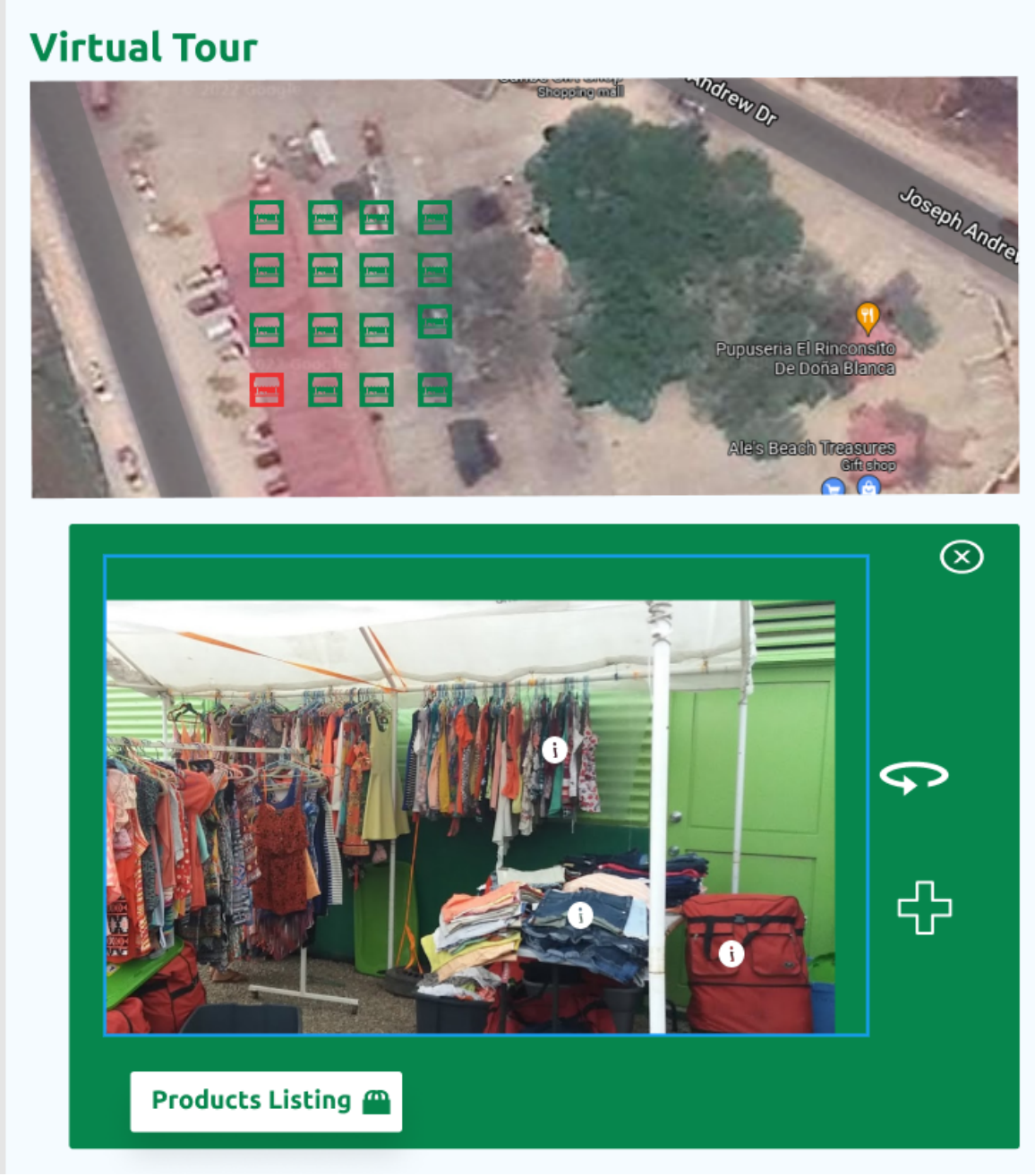


Tour Market

- The design below displays the tour market screen where customers will perform VR tours.

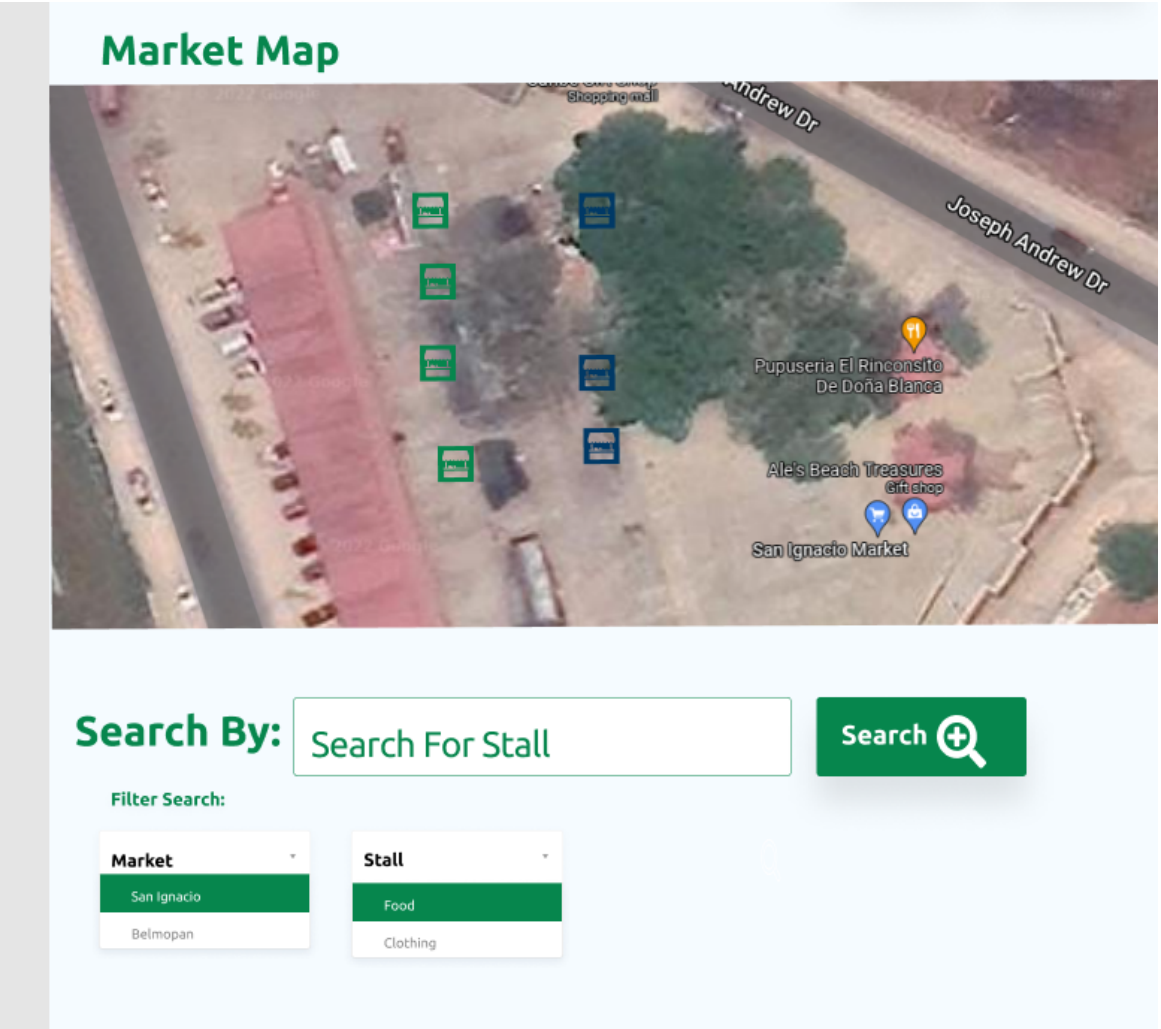
Tour Stall

- The design below displays the tour market screen where customers will perform VR tours.



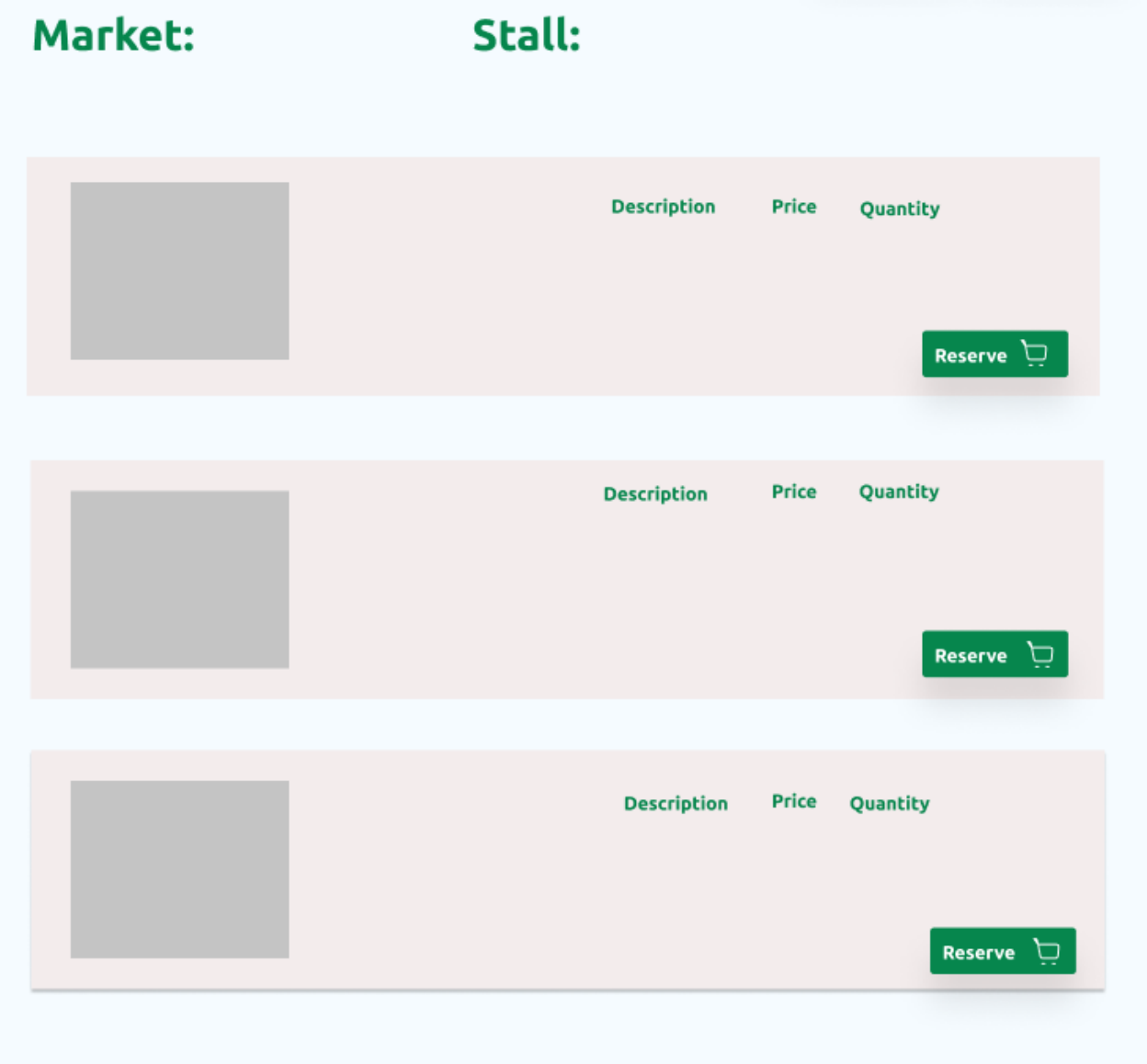
Search Stall

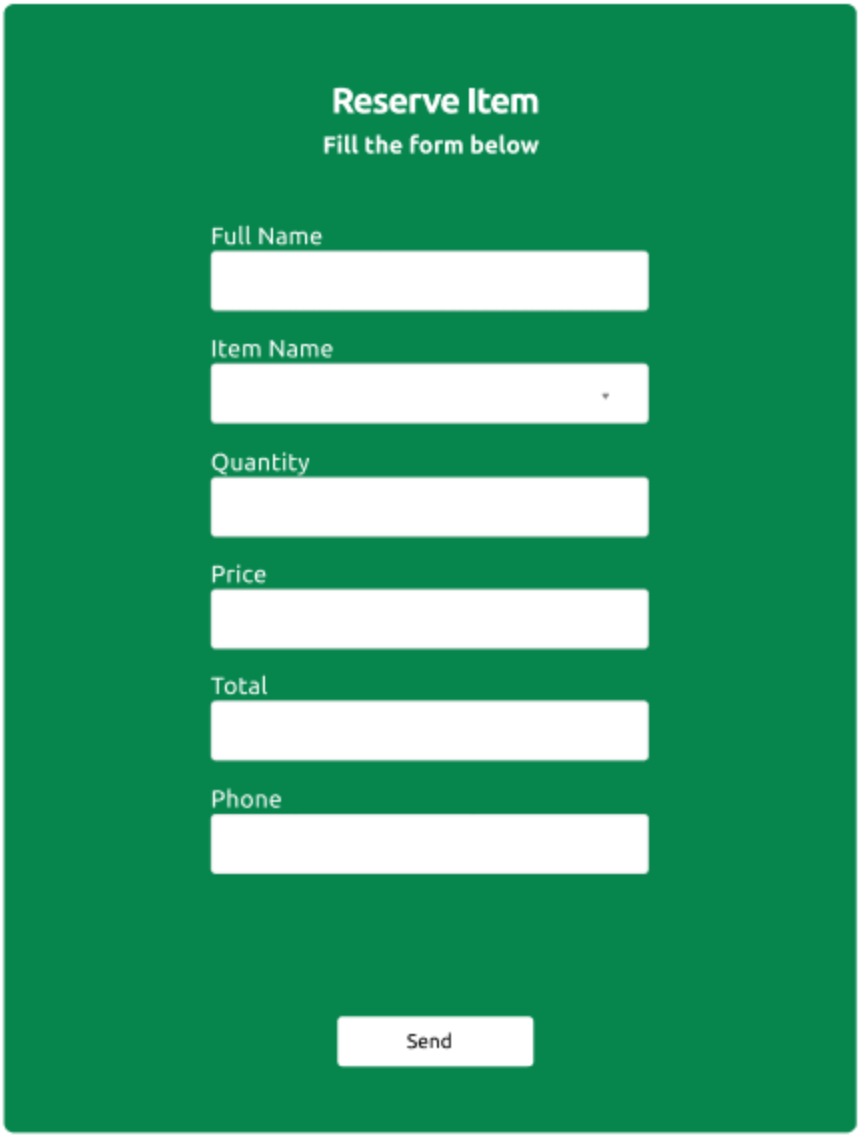
- The design below displays the search stall screen where a customer will be able to search for a stall in a market.



Buy Item

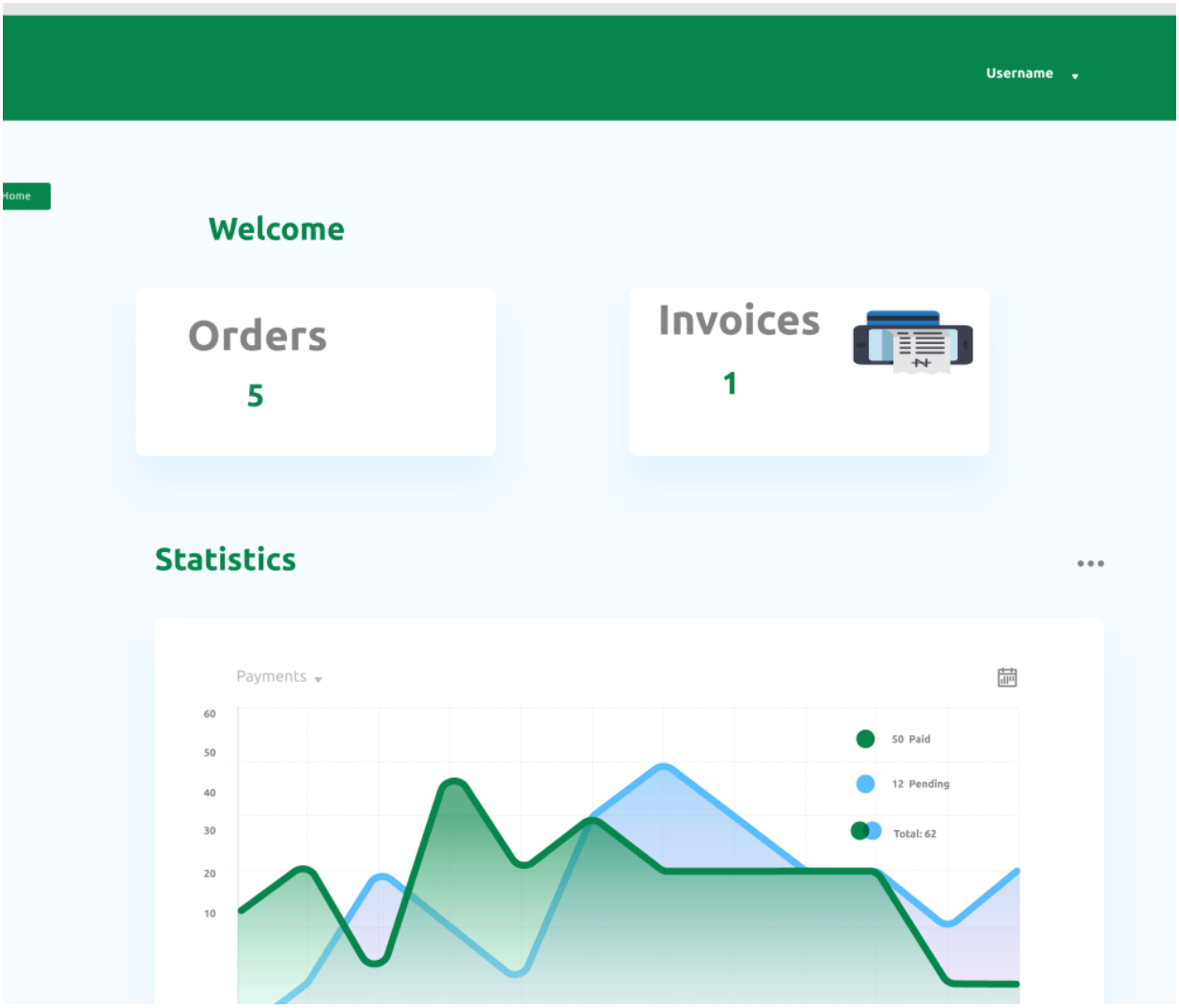
- The design below displays the buy item screens where the customer will be able to purchase products from the stalls in the market.

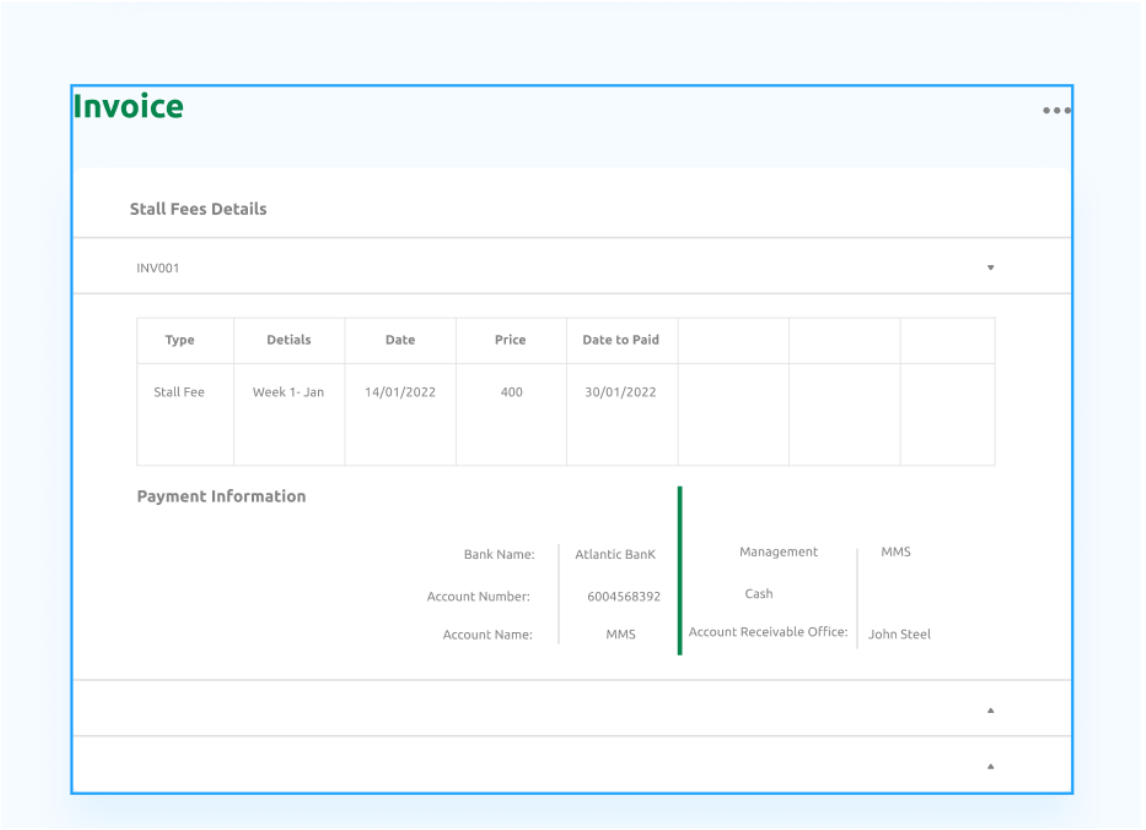


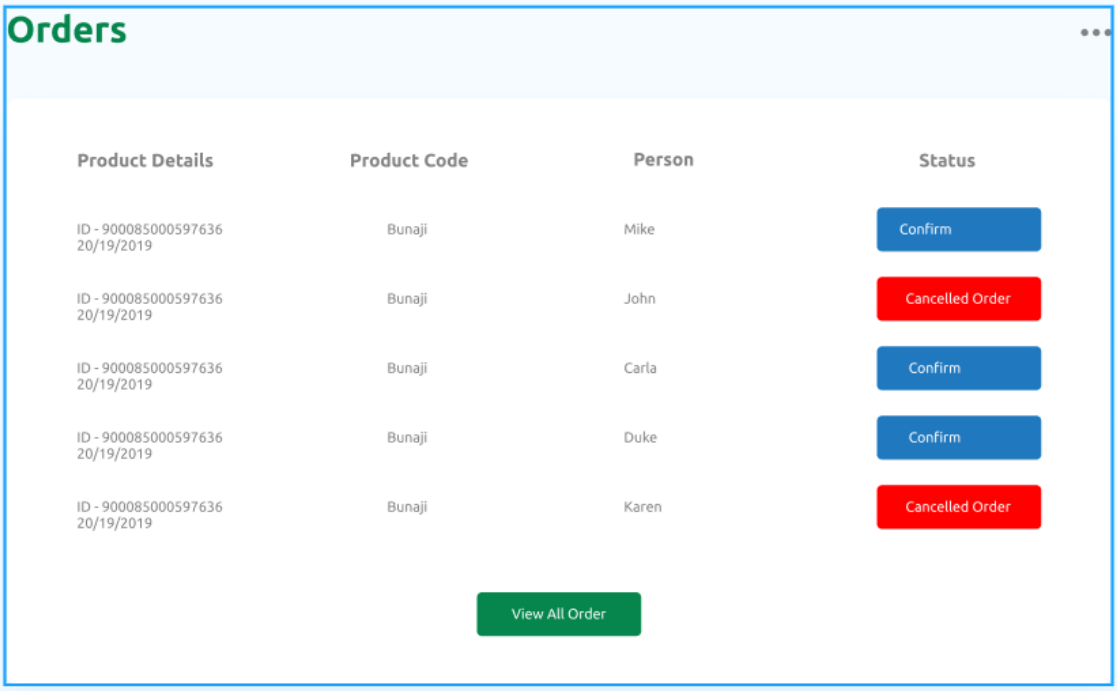

Reserve Item
Fill the form below
Full Name
Item Name
Quantity
Price
Total
Phone
Send

Vendor DashBoard

- The design below displays the primary vendor's dashboard screen.

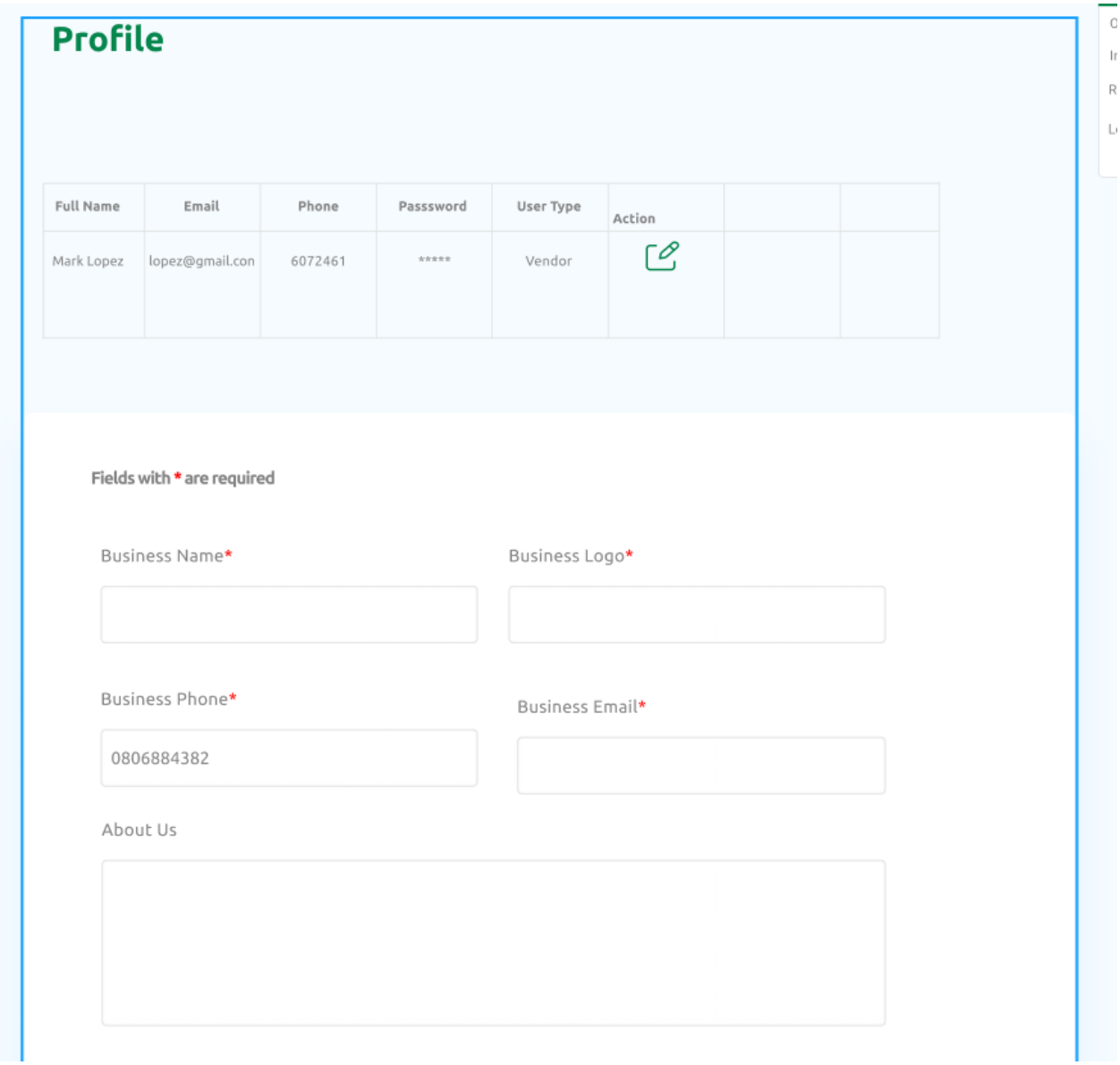



Invoice
Stall Fees Details
INV001
Type
Detials
Date
Price
Date to Paid
Stall Fee
Week 1- Jan
14/01/2022
400
30/01/2022
Payment Information
Bank Name:
Atlantic BanK
Account Number:
6004568392
Account Name:
MMS
Management
MMS
Cash
Account Receivable Office:
John Steel


Orders
Product Details
Product Code
Person
Status
ID - 900085000597636
20/19/2019
Bunaji
Mike
Confirm
ID - 900085000597636
20/19/2019
Bunaji
John
Cancelled Order
ID - 900085000597636
20/19/2019
Bunaji
Carla
Confirm
ID - 900085000597636
20/19/2019
Bunaji
Duke
Confirm
ID - 900085000597636
20/19/2019
Bunaji
Karen
Cancelled Order
View All Order

Create Profile:

- The design below displays the profile screen where the vendor will add information regarding their stall and business info.



Fields with * are required

Business Name*

Business Logo*

Business Phone*

0806884382

Business Email*

About Us

Business Category *

Food

Clothing

Vegetable and Friuts

Social Media

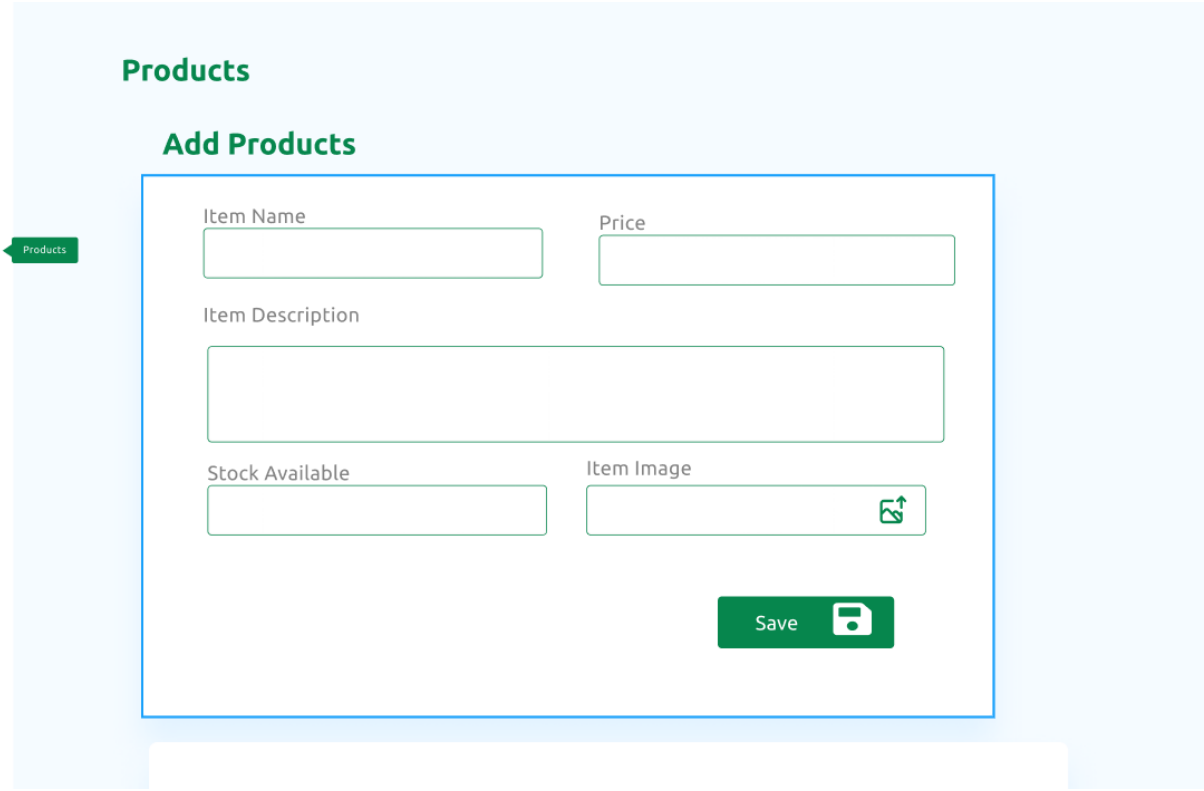
Facebook

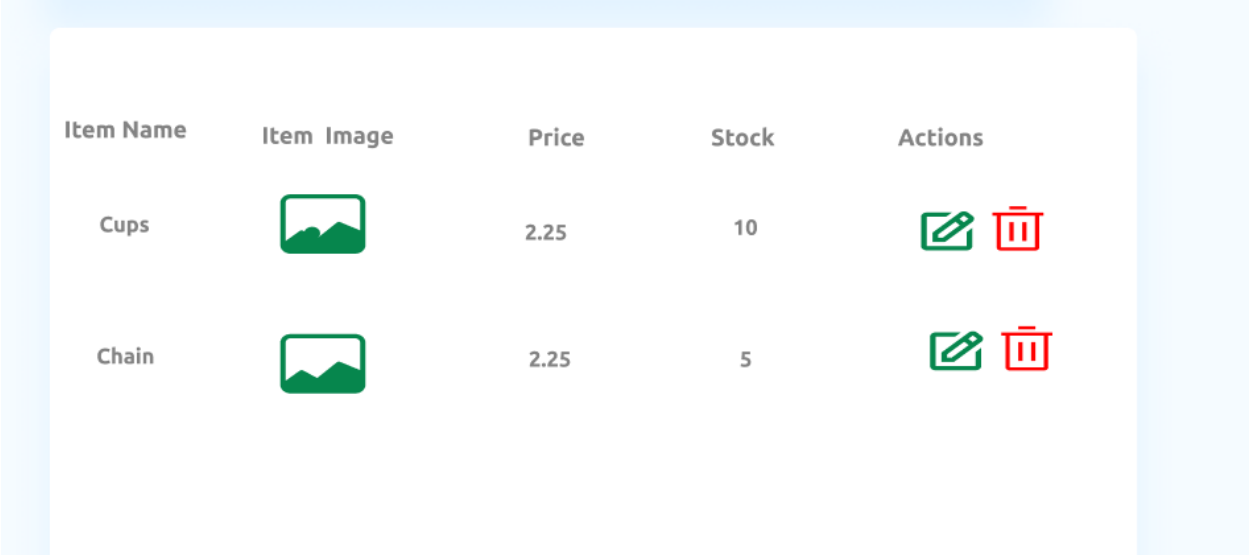
Instagram

Save

Add Products:

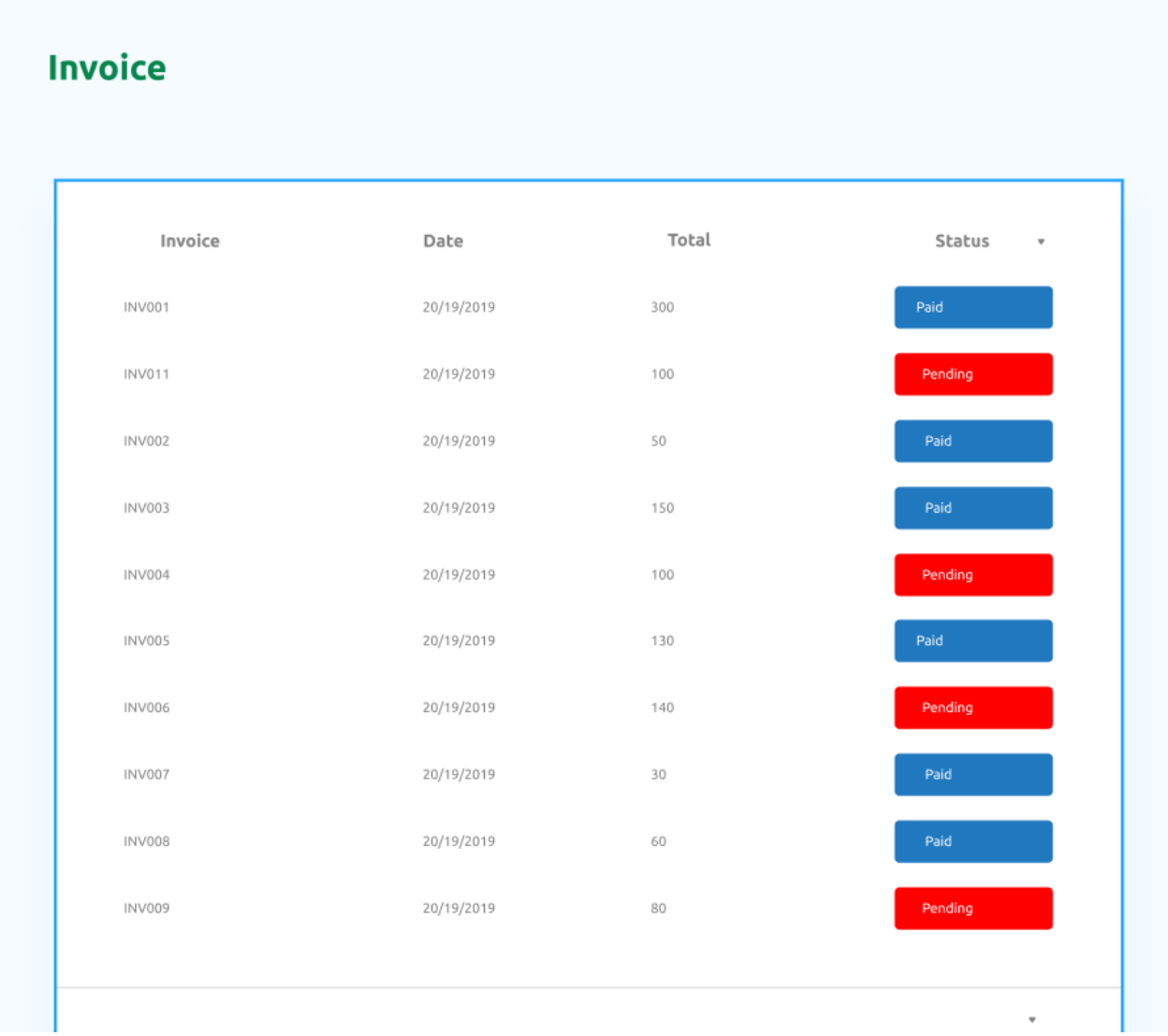
- The design below displays the add products screen where the vendor will add products available in their stall.





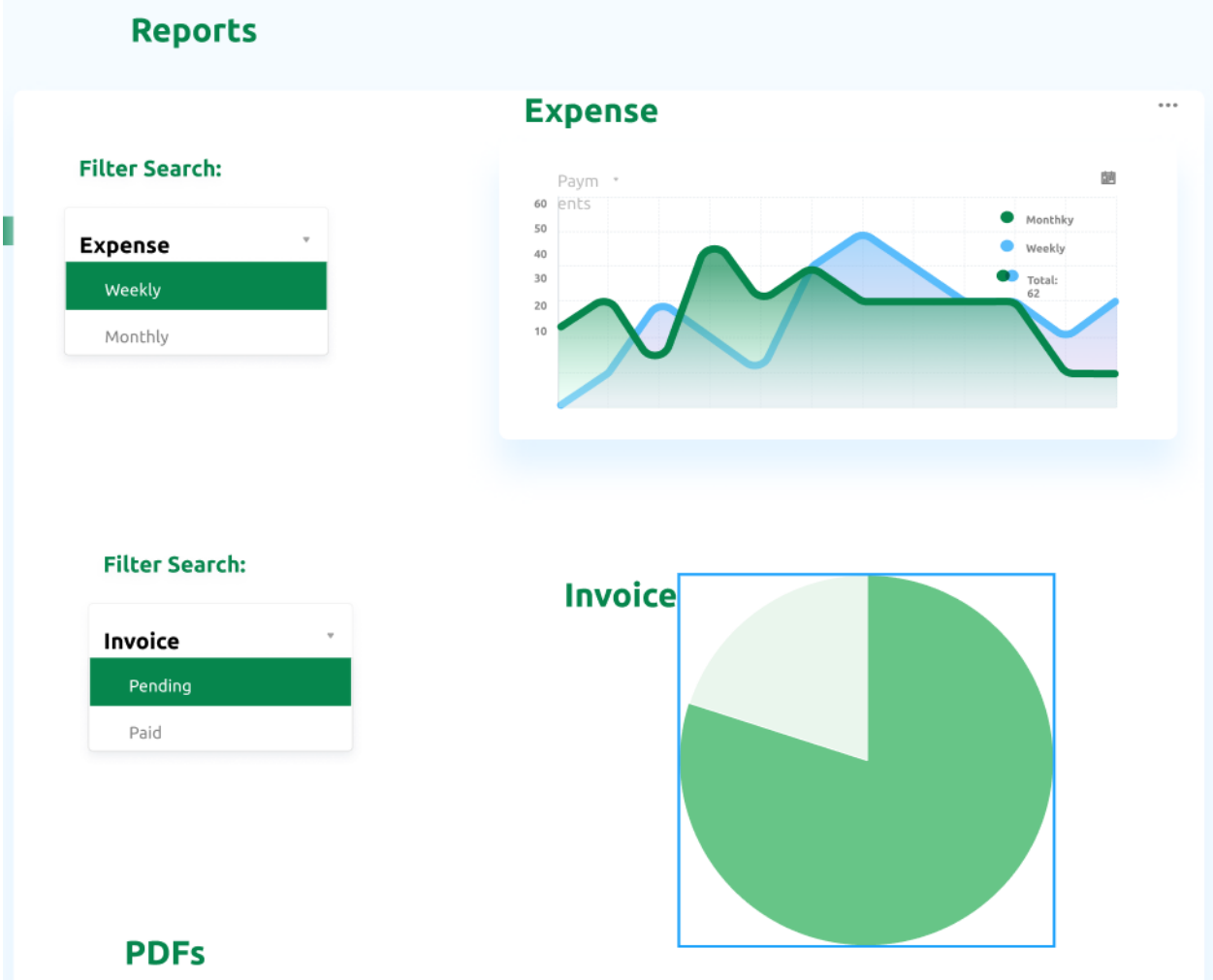
View Invoice:

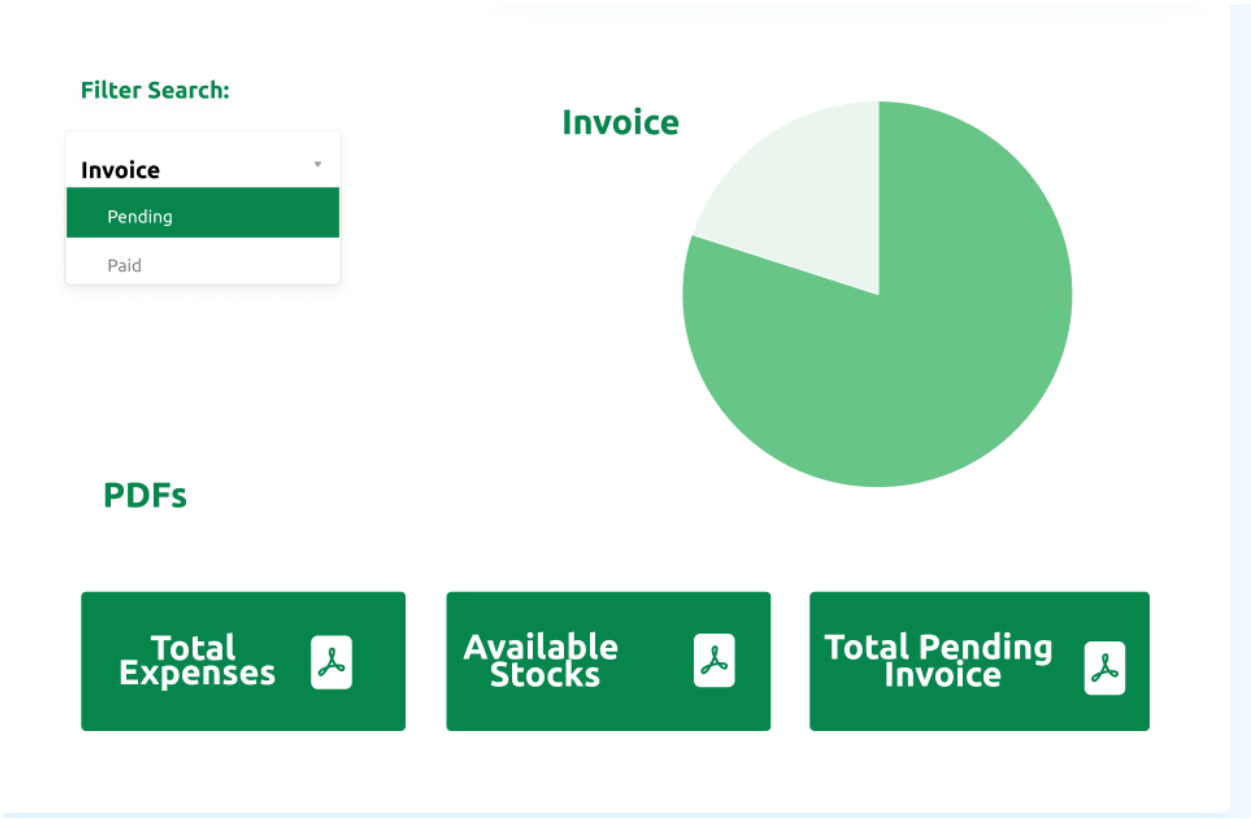
- The design below displays a view invoice screen where the vendor can view their invoice regarding their stall.



Generate Report

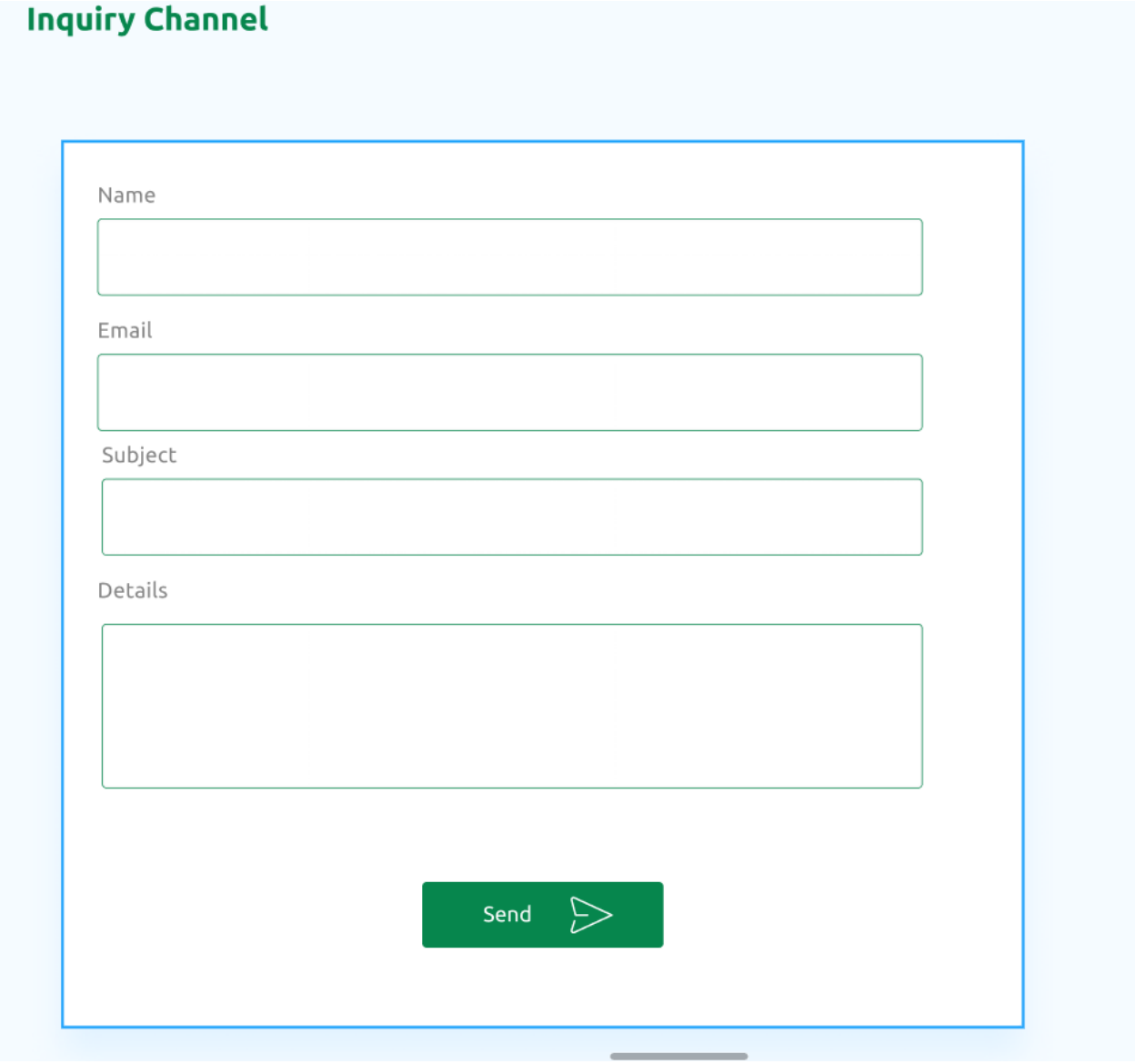
- The design below displays the generated report screen where the vendors will be able to produce expense reports.




Filter Search:
Invoice
Pending
Paid
Invoice
PDFs
Total Expenses
Available Stocks
Total Pending Invoice

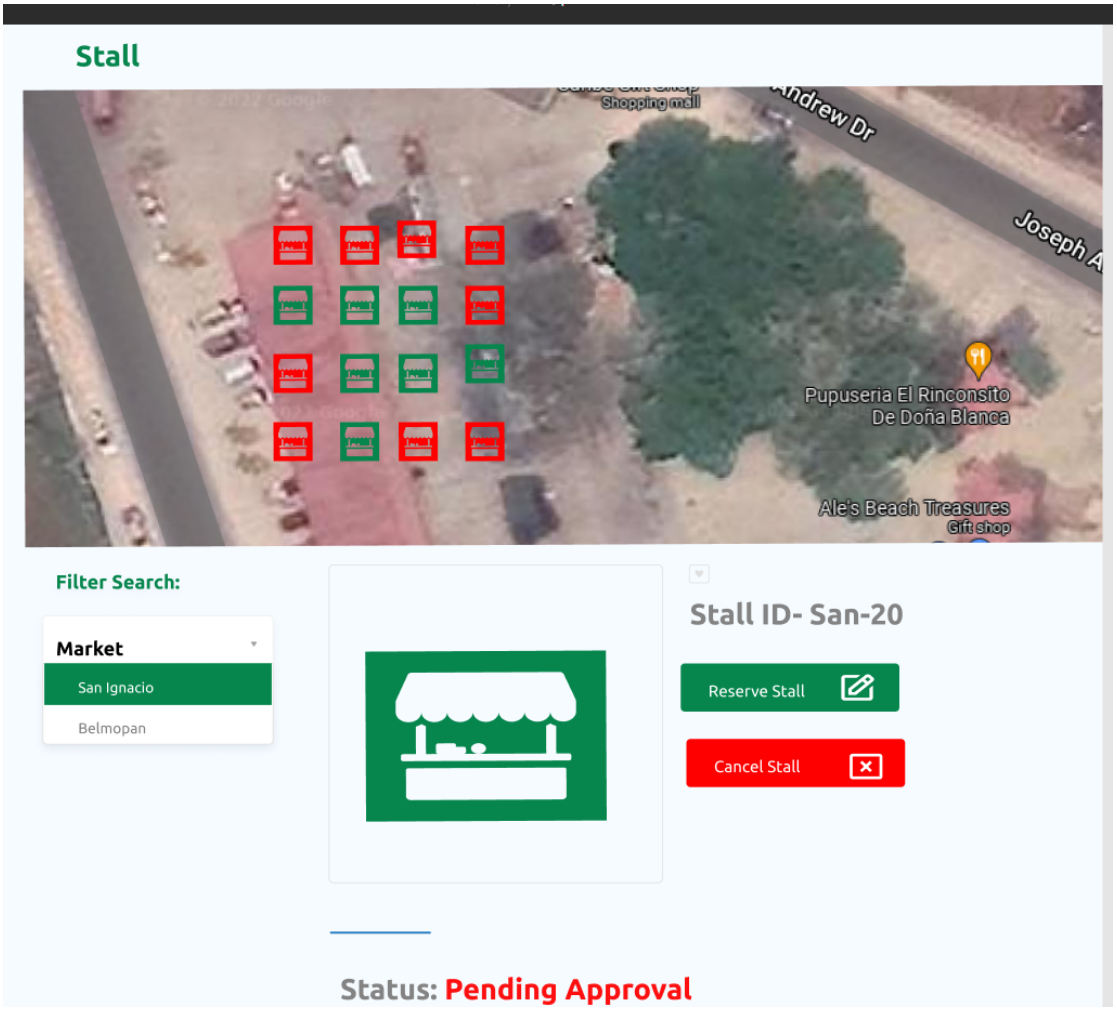
Submit Request

- The design below displays the submit request screen where the vendor will be able to submit inquiries.



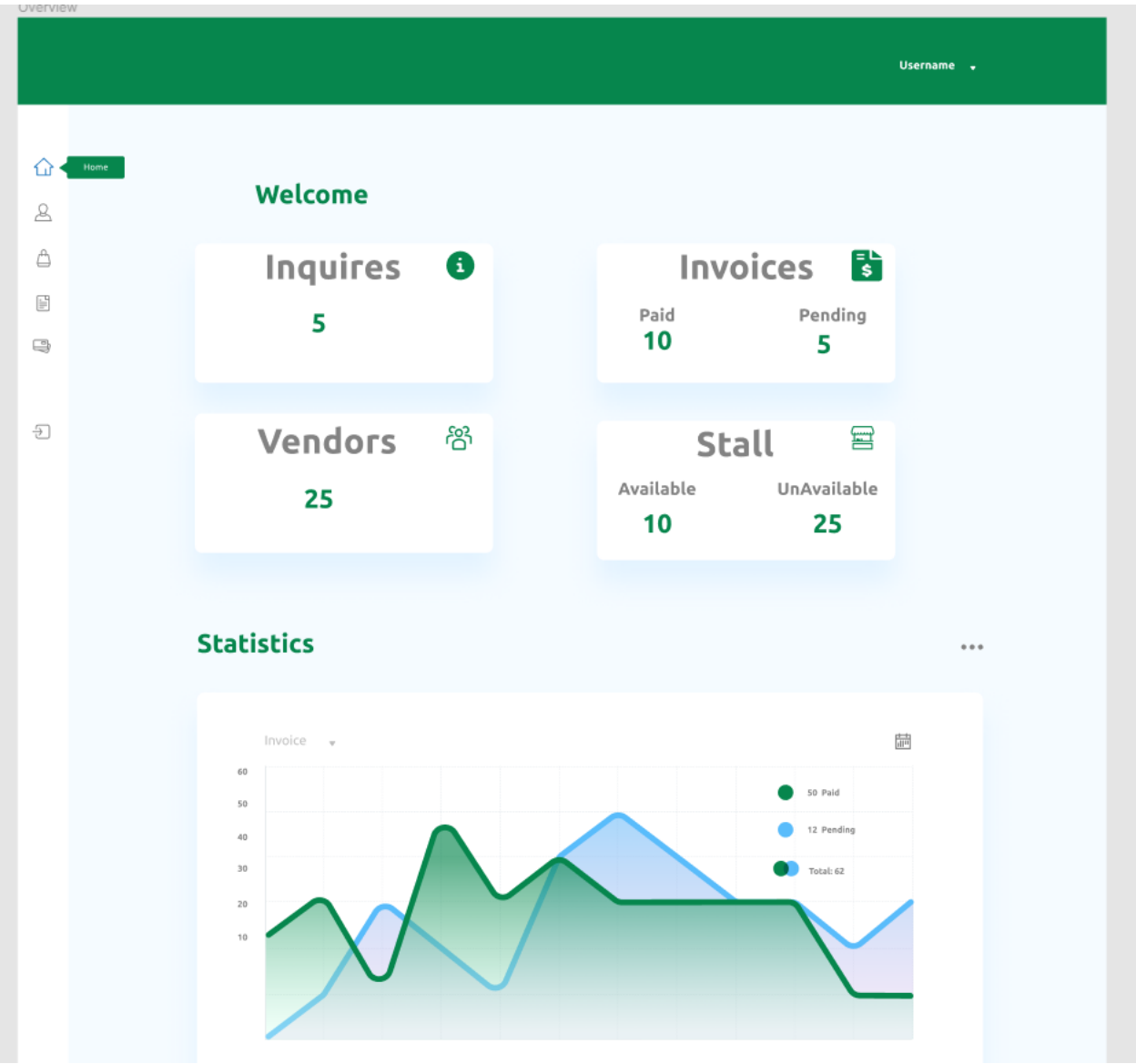
Reserve Stall and Cancel Stall

- The design below displays the reserve stall and cancel screens where the vendor will be able to reserve a stall and also cancel the stall reservation.

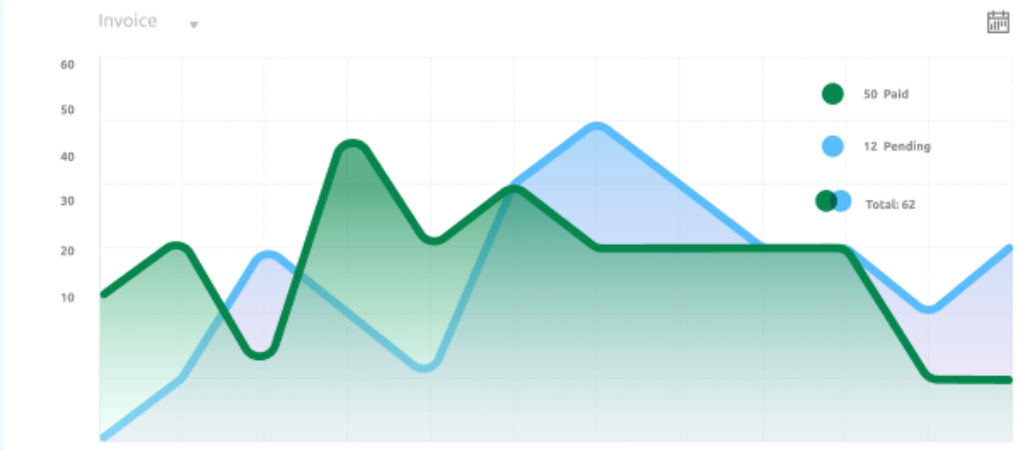


Management Dashboard

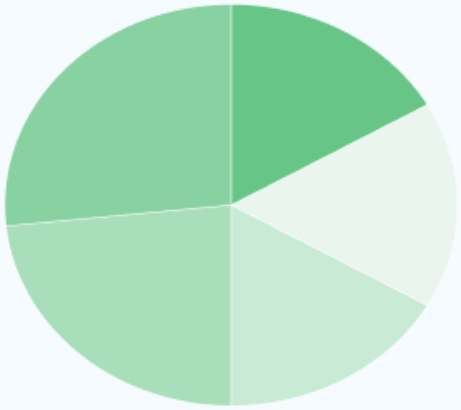
- The design below displays the management dashboard screen.



Statistics

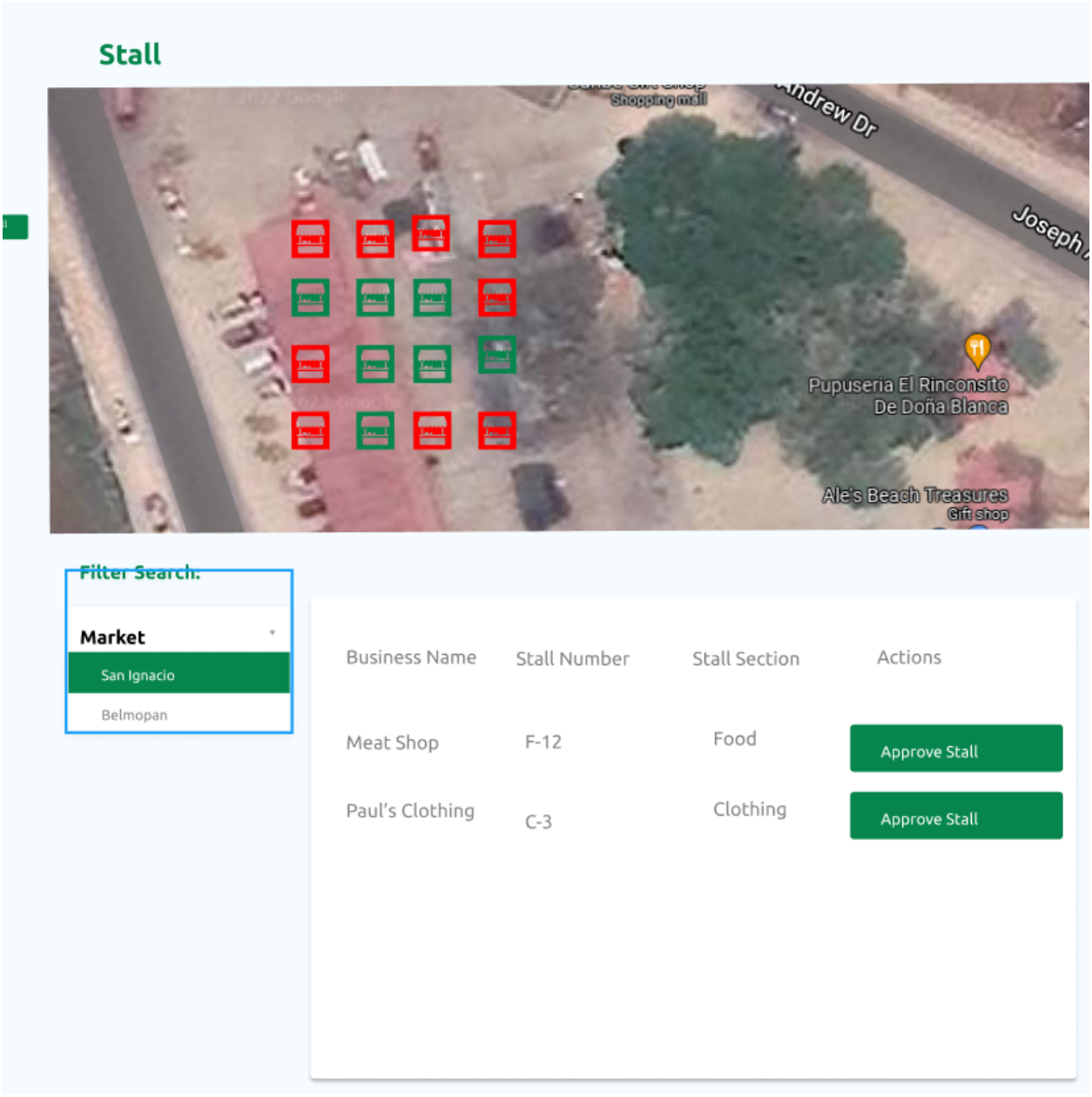


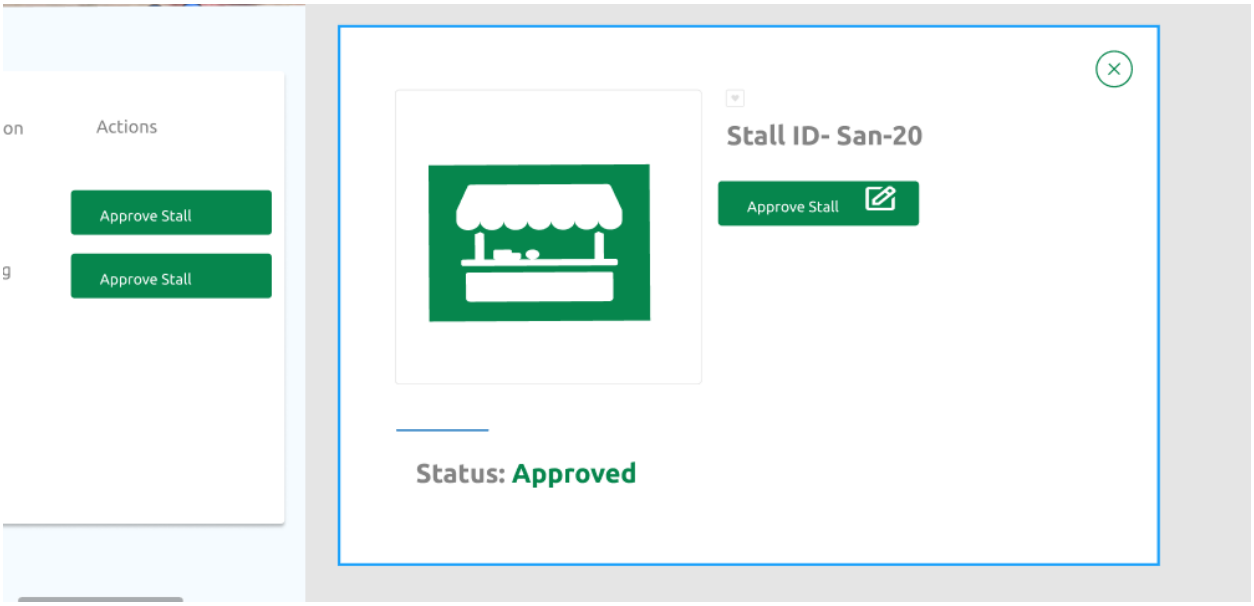
Types of Vendors



Approve Reserved Stall

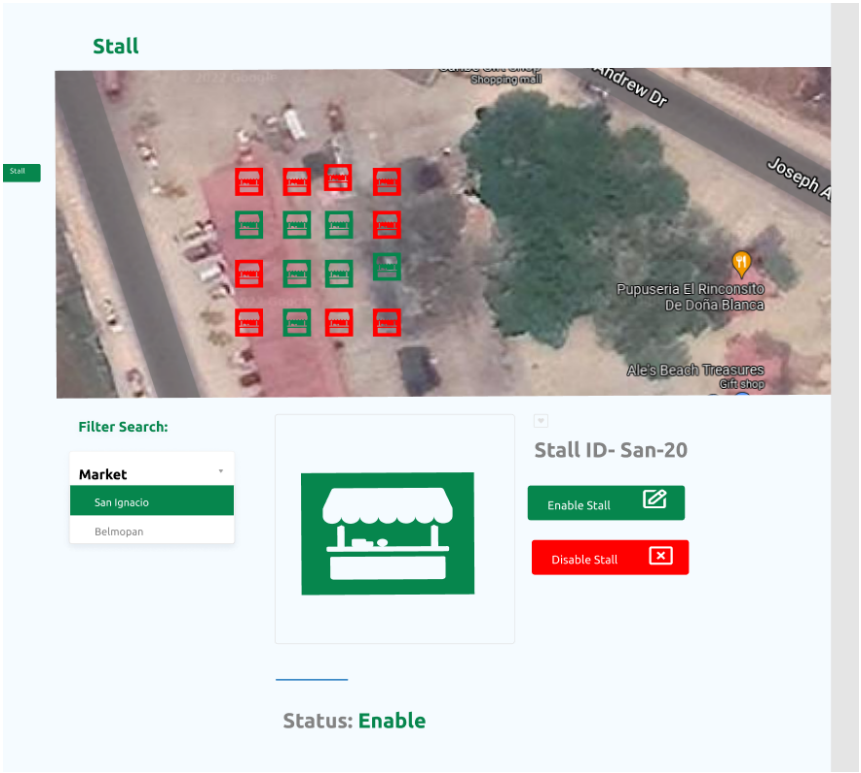
- The design below displays the approved stall screen where management can approve the stall that the vendor requests to reserve.





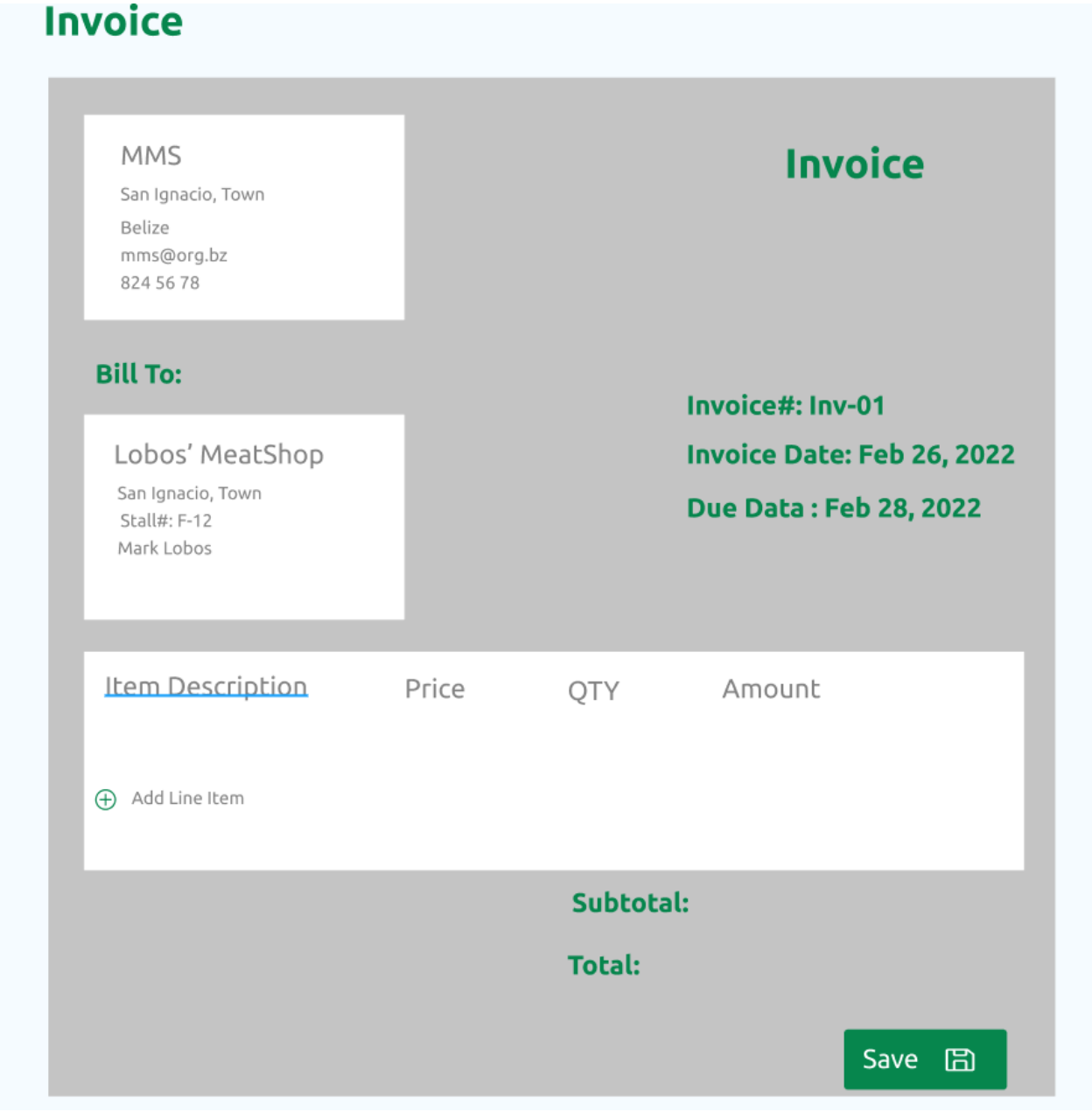
Enable and Disable Stall

- The design below displays the enable and disable stall screen where management can enable or disable stalls in the different markets.



Generate Invoices

- The design below displays the generated invoice screen where the management users will produce invoices for vendors.

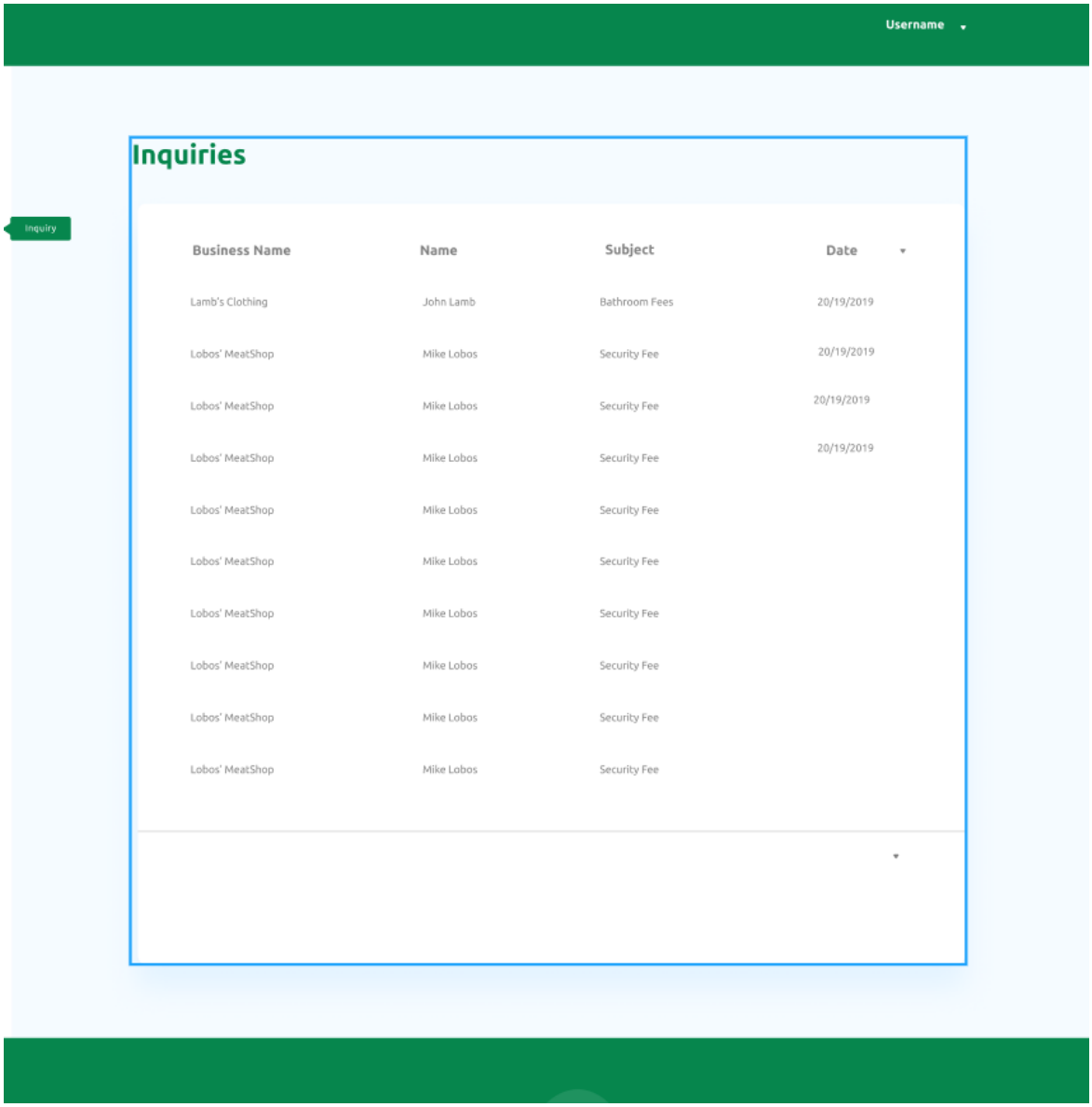


Invoices

Invoice	Due Date	Total	Status
INV001	20/19/2019	300	Paid
INV011	20/19/2019	100	Pending
INV002	20/19/2019	50	Paid
INV003	20/19/2019	150	Paid
INV004	20/19/2019	100	Pending
INV005	20/19/2019	130	Paid
INV006	20/19/2019	140	Pending
INV007	20/19/2019	30	Paid
INV008	20/19/2019	60	Paid
INV009	20/19/2019	80	Pending

View Inquires

- The design below displays the view inquiries screen where management will view the vendor's inquiries.



In terms of ease of use, the MMS system will provide a user-friendly user interface for all the users interacting with the platform. The platform will be flexible with different screen devices, so the system usability is excellent for all users. MMS platform UI design will be implemented with utmost simplicity for user learnability.

Design of Tests

Ten use cases will be utilized in this design test stage. The functions that will be tested will provide crucial feedback on how each function will react when the user interacts with the system. The test case will reference a use case or several use cases. Some test cases will use other cases as part of their test procedures. Each test case will pass on a fail or success scenario. Thus, the design test intends to provide the scope of the implementation of the system.

Test Case 1

Test Case:	TC-1
Use Case in Test	UC-1: AuthVendorandManagement
Criteria for success/fail:	Test is successful once the vendor and management is authenticated by the database as a valid vendor/ management.
Input Data:	Alphanumeric
Test Procedure:	Expected Output
Step 1: Enters correct username, password and type Step 2: Enters correct username, invalid password and type Step 3: Enters correct username, password and invalid type Step 2: Enters Invaild username, password and type	Successful login Failed Login Failed Login Failed Login

This Test case tests the authentication(UC-1) of vendors/ management trying to login into the system.

Test Case 2

Test Case:	TC-2
Use Case in Test:	UC-2: ViewMarketMap
Criteria for success/fail:	Successful if market grid map is retrieved
Input Data:	Location = Cayo
Test Procedure:	Expected Output
Step 1: Enter/select location in search bar Step 2: Click on market name	Successful display of market names Successful display of market grid map
This test case test the function getMarketMap to obtain a grid map of a specific market	

Test Case 3

Test Case:	**TC-3**
Use Case in Test	UC-3: GetMarketRoute
Criteria for success/fail:	The test is a success if the user enters the right coordinates to the desired marketplace location.
Input Data:	Enter current location coordinates and market coordinates:String
Test Procedure:	Expected Output

Step 1: Click on the button to obtain current location and select pin mark location on the map of the nearby markets in the area . ● call function “ getMarketRoute(c-latitude,c-longitude, d-latitude, d-longitude)” with valid data	Success: If a current location is obtained and an available pin market location is selected from the map
Step 2: Click on the button to obtain the current location and pin mark location on the map of the nearby markets in the area is not picked. ● call function “ getMarketRoute(c-latitude,c-longitude, d-latitude, d-longitude)” with invalid data	Fail: If a current location is obtained and an available pin market location is not selected from the map
This Test Case will mainly test functions of UC-3: GetMarketRoute	

Test Case 4

Test Case:	TC-4
Use Case in Test	UC-4: TourMarket
Criteria for success/fail:	The test is a success if the user selects a valid market name from the list provided.
Input Data:	Market Name: String
Test Procedure:	Expected Output
Step 1: Select the desire market from the dropbox ● Call function “tourMarket(marketName)” with valid data	Success : if a valid market name is selected.
Step 2: No Selection is done from the market lists dropbox ● Call function “tourMarket(marketName)” with invalid data	Fail: if a invalid market name is selected.
This Test Case will mainly test functions of UC-4: TourMarket	

Test Case 5

Test Case:	TC-5
Use Case in Test	UC-5: TourStall
Criteria for success/fail:	The test is a success if the user select a valid stall in the desired market
Input Data:	Stall id : String
Test Procedure:	Expected Output
Step 1: Select a stall from market grid map ● Call function “tourStall(name) with valid data	Success : if a stall is select in the grid map
Step 2: No stall is selected from market grid map ● Call function “tourStall(name) with invalid data	Fail : if no stall is select in the grid map

Test Case 6

Test Case:	TC-6
Use Case in Test	UC-7: BuyItem
Criteria for success/fail:	Test is successful when the item intended to be purchased by the customer is available on the respective market stall.
Input Data:	Integer
Test Procedure:	Expected Output
Step1.Call function buyItem (data) with valid data Step2.Call function buyItem (data) with invalid data	Success: Market product is available at respective market stalls. Fail: Market product is not available at respective market stalls.

This Test case tests the BuyItem (UC7) of the customer who reserves/purchases an item on the system.

Test Case 7

Test Case:	TC-7
Use Case in Test	UC-11: GenerateReport
Criteria for success/fail:	Successful when the total price of a vendor stall is output in report form
Input Data:	none
Test Procedure:	Expected Output
Step 1: Click on button ● Call function generatereport returns true if the vendor had 1 stall in account history	A report showing their expenses and special charges
Step 2: click on button ● Call function generateInvoices returns false as the vendor has no prior stall purchases in history.	No report is generated

Test Case 8

Test Case:	TC-8
Use Case in Test	UC-15: ApproveReservedStall
Criteria for success/fail:	Successful when the manager is successfully able to approve a market stall reservation request. First it verifies if market stall is available, reserved and checks status before any approval is processed.
Input Data:	Alphanumeric
Test Procedure:	Expected Output
Step 1 Calls function getStallByNumber(stallNum) with valid data Step 2 Calls getStallByNumber(stallNum) with invalid data	Success: A market stall request approval is successful when the market stall is available for assignment Fail: A market stall approval fails when no market stall is available or does not exist.

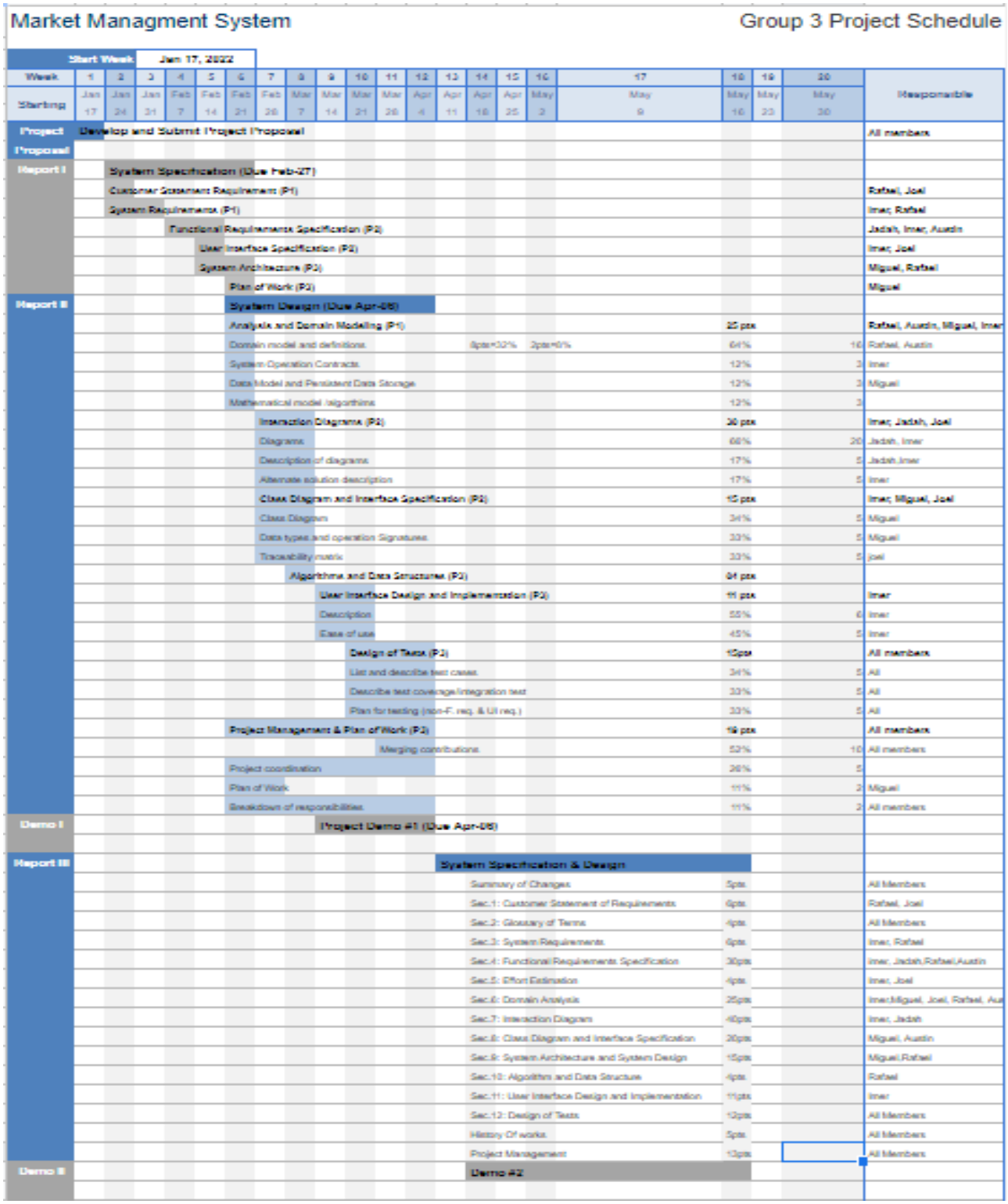
Test Case 9

Test Case:	TC-9
Use Case in Test	UC-18: GenerateInvoices
Criteria for success/fail:	Successful if all fees for a certain vendor are output in report form.
Input Data:	Vendor Name: String
Test Procedure:	Expected Output
Step 1: A valid vendor's name is chosen from a list of active vendors. ● Call function generateInvoices returns true because at least 1 fee was found.	A report of a certain vendor's costs of services incurred to management.
Step 2: An invalid vendor's name is chosen from a list of active vendors. ● Call function generateInvoices returns true because no fee was found.	No report is generated.

Test Case 10

Test Case:	TC-10
Use Case in Test	UC-19: ViewInquires
Criteria for success/fail:	Successful if inquiry list or message is shown
Input Data:	None
Test Procedure:	Expected Output
Step 1: Click on Inquiry Button	Successful display of inquiry list or a message that says: No Inquiries
The test case will test the function getInquiry to view inquiries.	

History of Works



Market Managment System — Group 3 Project Schedule

Start Week: Jan 17, 2022

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Responsible
Starting	Jan 17	Jan 24	Jan 31	Feb 7	Feb 14	Feb 21	Feb 28	Mar 7	Mar 14	Mar 21	Mar 28	Apr 4	Apr 11	Apr 18	Apr 25	May 2	May 9	May 16	May 23	May 30	
Project Proposal	Develop and Submit Project Proposal																				All members
Report I	System Specification (Due Feb-27)																				
	Customer Statement Requirement (P1)																				Rafael, Joel
	System Requirements (P1)																				Iner, Rafael
			Functional Requirements Specification (P2)																		Jadah, Iner, Austin
					User Interface Specification (P2)																Iner, Joel
					System Architecture (P2)																Miguel, Rafael
						Plan of Work (P2)															Miguel
Report II						System Design (Due Apr-06)															
						Analysis and Domain Modeling (P1)												25 pts			Rafael, Austin, Miguel, Iner
						Domain model and definitions								8pts=32%	2pts=8%			64%		16	Rafael, Austin
						System Operation Contracts												12%		3	Iner
						Data Model and Persistent Data Storage												12%		3	Miguel
						Mathematical model /algorithms												12%		3	
							Interaction Diagrams (P2)											30 pts			Iner, Jadah, Joel
							Diagrams											66%		20	Jadah, Iner
							Description of diagrams											17%		5	Jadah,Iner
							Alternate solution description											17%		5	Iner
							Class Diagram and Interface Specification (P2)											15 pts			Iner, Miguel, Joel
							Class Diagram											34%		5	Miguel
							Data types and operation Signatures											33%		5	Miguel
							Traceability matrix											33%		5	joel
								Algorithms and Data Structures (P2)										04 pts			
								User Interface Design and Implementation (P2)										11 pts			Iner
								Description										55%		6	Iner
								Ease of use										45%		5	Iner
									Design of Tests (P2)									12pts			All members
									List and describe test cases									34%		5	All
									Describe test coverage/integration test									33%		5	All
									Plan for testing (non-F. req. & UI req.)									33%		5	All
						Project Management & Plan of Work (P2)												19 pts			All members
										Merging contributions								53%		10	All members
						Project coordination												26%		5	
						Plan of Work												11%		2	Miguel
						Breakdown of responsibilities												11%		2	All members
Demo I									Project Demo #1 (Due Apr-06)												
Report III													System Specification & Design								
														Summary of Changes				5pts			All Members
														Sec.1: Customer Statement of Requirements				6pts			Rafael, Joel
														Sec.2: Glossary of Terms				4pts			All Members
														Sec.3: System Requirements				6pts			Iner, Rafael
														Sec.4: Functional Requirements Specification				30pts			Iner, Jadah,Rafael,Austin
														Sec.5: Effort Estimation				4pts			Iner, Joel
														Sec.6: Domain Analysis				25pts			Iner,Miguel, Joel, Rafael, Au
														Sec.7: Interaction Diagram				40pts			Iner, Jadah
														Sec.8: Class Diagram and Interface Specification				20pts			Miguel, Austin
														Sec.9: System Architecture and System Design				15pts			Miguel,Rafael
														Sec.10: Algorithm and Data Structure				4pts			Rafael
														Sec.11: User Interface Design and Implementation				11pts			Iner
														Sec.12: Design of Tests				12pts			All Members
														History Of works				5pts			All Members
														Project Management				12pts			All Members
Demo II														Demo #2							

Project Coordination

The following is what has been implemented:
- The database has been created and populated with data.
- The ability to generate billing report
- A fully functional logging in system.
- Vendor UI
- Manager UI
- Customer UI (Partially)

Future Works

The MMS - Market Management System lacks key features. Some of these features are the below and are likely to be integrated easily.

1. Integration of online payment system for products/Invoices/rental.
2. Integration of automatic email notification for alerts on overdue invoices
3. Integration of Google API - Google has updated its policy which made the API no longer free to use and now requires payment.
4. Integration of a Alpha version of Virtual Reality library - currently the beta is available for Public use

References

IBM Cloud Education. (2020, October 28). Three-tier Architecture. IBM. Retrieved February 26, 2022, from https://www.ibm.com/cloud/learn/three-tier-architecture#:~:text=Three%2Dtier%20architecture%20is%20a,associated%20with%20the%20application%20is

Lynch, A. (2021, July 7). UML Package Diagram. Edrawsoft. Retrieved February 26, 2022, from https://www.edrawsoft.com/uml-package.html

Sommerville, I, (2015). Software Engineering 10th, Ed., Addison-Wesley