GoGoGrocery

System Specification

Prepared for:

Mr. Cameron, GoGoGrocery

Prepared by:

Nicholas Sutanto, Software Engineer

OpenXcell

**Table of Contents**

**Executive Summary**…………………………………………………………….2

1. **Introduction……………………………………………………………….3**
   1. Problem Statement…………………………………………………………………3
   2. System Services…………………………………………………………………….3
   3. Non-Functional Requirements and Design Constraints……………………......4
   4. System Evolution………………………………………………………..................4
   5. Document Outline…………………………………………………………………..5
2. **Structural Model………………………………………………………….6**
   1. Introduction …………………………………………………………………………6
   2. Class Diagram………………………………………………………………………6
   3. Metadata…………………………………………………………………………….7
3. **Architecture Design……………………………………………………17**
   1. Introduction………………………………………………………………………...17
   2. Infrastructure Model……………………………………………………………….17
   3. Hardware and Software Requirements…………………………………………18
   4. Security Plans…………………………………………………………………......19
4. **User-Interface……………………………………………………………20**
   1. User-Interface Requirements and Constraints………………………………...20
   2. Forms: Screen/User-Interaction Design………………………………………..20
   3. Reports: “Printed Output” Design……………………………………………….25
5. **Appendices………………………………………………………………26**
   1. Bibliography………………………………………………………………………26
   2. Supporting Documentation……………………………………………………..26

**Executive Summary**

GoGoGrocery is hoping to work with a software development company to aid in the production of their new application. GoGoGrocery has collaborated with OpenXcell to help design, develop, and maintain the GoGoGrocery mobile application. The GoGoGrocery application will help support customers order groceries online and assist drivers in receiving the most optimal routes when delivering. Together, OpenXcell and GoGoGrocery have conducted feasibility assessments and analysis and concluded that the GoGoGrocery application is quite feasible and low risk. Both companies realize that there will be many benefits if an application such as GoGoGrocery is implemented to the public.

**1.0 Introduction**

**1.1 Problem Statement and Project Vision**

GoGoGrocery is a company founded by a group of students who plan to develop an application that aids people in getting groceries. In collaboration with OpenXcell, GoGoGrocery has designed and developed a mobile application to enable its customers to get groceries delivered in the comfort of their own home. GoGoGrocery realizes the power of technology can be used to benefit the community. By integrating the use of this application into their company, it would not only be convenient for the customers but also a milestone for the company.

GoGoGrocery recognizes that not everyone has a mode of transportation to purchase groceries. The elderly, international students, and low-income households are some of the demographics that are affected by the problem. GoGoGrocery hopes to eliminate or at least minimize the problem. They believe the GoGoGrocery application can be utilized to cut the cost, time, and effort of customers having to travel to a grocery store. OpenXcell and GoGoGrocery plan to build an application that meets the needs of their customers, delivery drivers, and company. The application is intended to run in the King and Snohomish County at the time of launch. There is a possibility of expanding to other States but, it all depends on the demand of the application. The GoGoGrocery application will be used for grocery orders by the customers, locating the most convenient store for the customer and drivers, and delivery routes for the drivers.

**1.2 System Services**

This section describes the functional requirements for the GoGoGrocery Application. A more detailed explanation of requirements can be found in the system proposal.

Customer Functional requirements:

* Create an account to access the application and search for groceries.
* Select their favorite store and view groceries.
* Pay for their groceries.

Application Functional requirements:

* Request customer location to search for the nearest grocery stores.
* Display available items in categories and allow customers to search for items.
* Assign a driver to a customer when an order is placed.
* Provides a delivery route for the driver.

Driver Functional requirements:

* Drivers must make an account. They also must sign up to be a driver.
* The drivers must be able to view the items the customers purchased
* The driver must be given clear instructions regarding the safe delivery of grocery items.

**1.3 Non-Functional Requirements and Design Constraints**

This section lists the major non-functional requirements and design constraints to be considered when developing the GoGoGrocery application. For a more detailed explanation of the non-functional requirements, please refer to the system proposal.

* The GoGoGrocery application must run on multiple platforms. For example, IOS and Android. There is no plan for web browsers as of now.
* Drivers will use implemented GPS tools for directions.
* The application must have a simple GUI for ease of use.
* GoGoGrocery must process actions in a reasonable time on all devices. Processes such as searching for items, connecting to a driver, and checkout.
* GPS must be quick enough to make changes to routes when there are roadblocks present or when the driver uses an alternate route.
* GoGoGrocery will be developed by early 2021 or latest by late 2021
* The goal is to develop GoGoGrocery in a cost-efficient manner.
* The schedule for developing the application is must be somewhat flexible due to the circumstances of COVID-19.

**1.4 System Evolution**

The goal is to develop the GoGoGrocery application as cost-efficient and error-free as possible. As the GoGoGrocery application continues to be used, maintenance and upgrades will have to be done to be up to date and run as planned. After the final development of the GoGoGrocery application, GoGoGrocery will still be collaborating with OpenXcell to further maintain and upgrade the application. OpenXcell is responsible for conducting routine maintenance and repair. Once there is a definite show of success by the application, GoGoGrocery can implement features that were planned but were not able to be implemented. The features may be beneficial for the future of the GoGoGrocery application.

**1.5 Document Outline**

* Introduction– Information about the application including problem statement, functional and non-functional requirements, and system evolution.
* Structural Model– A class diagram that shows how the system of the application works.
* Architecture Design–Two deployment diagrams that will provide a clear picture of the architecture of the system.
* User-Interface–Basic requirements and constraints for the user-interface design of the GoGoGrocery application.
* Appendices – Bibliography and supporting documents.

|  |  |
| --- | --- |
|  |  |

































































































































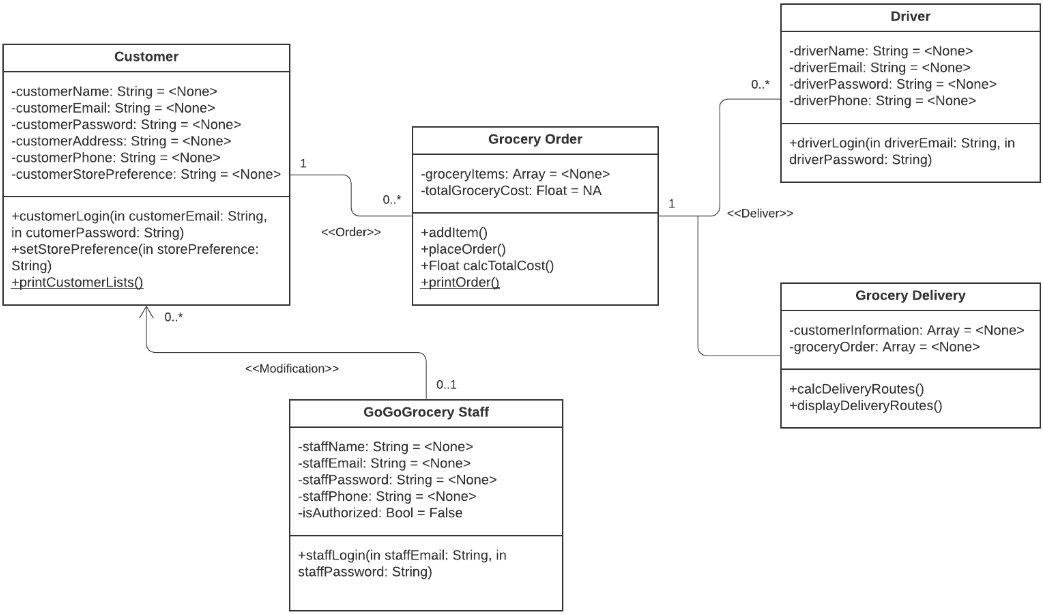


**2.0 Structural Model**

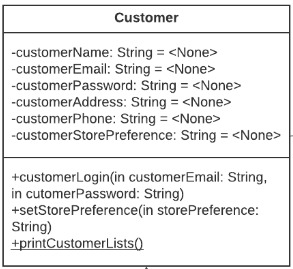
**2.1 Introduction**

This section will elaborate on the process of the GoGoGrocery system. The section will include a class diagram and an explanation of each class in detail.

**2.2 Class Diagram**



**2.3 Metadata**



**Description**: This class represents a customer of the GoGoGrocery application.

**Visibility**: Public

**Is Abstract**: No

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Data Type | Is Derived | Is Read Only | Visibility | Multiplicity | Default Value |
| customerName | First, Last name. | String | No | No | Private | 1 | None |
| customerEmail | Email to login to the app. | String | No | No | Private | 1 | None |
| customerPassword | The password to log in. | String | No | No | Private | 1 | None |
| customerAddress | Delivery address. | String | No | No | Private | 1 | None |
| customerPhone | Phone number. | String | No | No | Private | 1..\* | None |
| customerStorePreference | Preferred store of the customer. | String | No | No | Private | 0..\* | None |

**Operations:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Description** | **Return Type** | **Visibility** | **Is Query** |
| customerLogin | Customer logs in to the application to add information and view items. | None | Public | No |
| setStorePreference | Customers can set a grocery store as their favorite**.** | None | Public | No |
| printCustomerLists | Prints a list of GoGoGrocery customers. | None | Public | No |

**Processing Outlines:**

* customerLogin(in customerEmail: String, in customerPassword: String)

Customer inputs email and password

If valid

Allow customer to login into the application

Else

Prompt user to enter information again

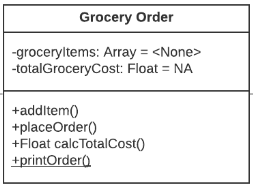
* setStorePreference(in storePreference: String)

A function that allows the customer to save their preferred store.

Update the customerStorePreference attribute with a new parameter that is passed on.

* printCustomerLists()

Returns a string of the customer’s information.



**Description**: This class represents a customer’s order of the GoGoGrocery application.

**Visibility**: Public

**Is Abstract**: No

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Data Type | Is Derived | Is Read Only | Visibility | Multiplicity | Default Value |
| groceryItems | An array that holds the quantity, name, and price of an item. | Array | No | No | Private | 1 | None |
| totalGroceryCost | The total cost of groceries bought by the customer. | String | Yes | No | Private | 1 | N/A |

**Operations:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Description** | **Return Type** | **Visibility** | **Is Query** |
| addItem | Customers can add items to their cart. | None | Public | No |
| placeOrder | Allows the customer to place an order. | None | Public | No |
| calcTotalCost | Calculates the total cost of groceries. | Float | Public | No |
| printOrder | Print a list of items the customer has ordered. | None | Public | Yes |

**Processing Outlines:**

* addItem()

While (The customer selects an item and the quantity)

Add item to the cart

* placeOrder()

Check customer address and delivery times

Prompt to enter payment information on the next page

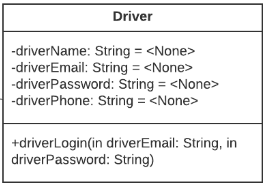
* calcTotalCost()

Total cost = (item1 \* quantity) + (item2 \* quantity) + …

return totalCost

* printOrder()

Returns a string of the customer’s order.



**Description**: This class represents a driver of the GoGoGrocery application.

**Visibility**: Public

**Is Abstract**: No

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Data Type | Is Derived | Is Read Only | Visibility | Multiplicity | Default Value |
| driverName | First, Last name. | String | No | No | Private | 1 | None |
| driverEmail | Email to login to the app. | String | No | No | Private | 1 | None |
| driverPassword | The password to log in. | String | No | No | Private | 1 | None |
| driverPhone | Phone number. | String | No | No | Private | 1..\* | None |

**Operations:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Description** | **Return Type** | **Visibility** | **Is Query** |
| driverLogin | The driver logs in to the GoGoGrocery application with a driver account. | None | Public | No |

**Processing Outlines:**

* driverLogin(in driverEmail: String, in driverPassword: String)

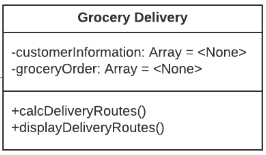
Driver inputs email and password

If valid

Allow the driver to login into the application

Else

Prompt user to enter information again



**Description**: This class represents a grocery delivery order of the GoGoGrocery application.

**Visibility**: Public

**Is Abstract**: No

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Data Type | Is Derived | Is Read Only | Visibility | Multiplicity | Default Value |
| customerInformation | An array that holds the customer’s information | Array | No | No | Private | 1 | None |
| groceryOrder | An array that holds the customer’s name and their order | Array | No | No | Private | 1 | None |

**Operations:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Description** | **Return Type** | **Visibility** | **Is Query** |
| calcDeliveryRoutes | Calculates a delivery route for the driver based on the customer’s address. | None | Public | Yes |
| displayDeliveryRoutes | Displays the delivery address to the driver | None | Public | Yes |

**Processing Outlines:**

* calcDeliveryRoutes()

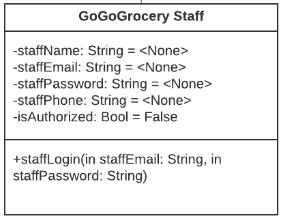
Use the integrated MapKit styled tool to calculate routes to the customer’s address.

Generate the directions

Output directions

* displayDeliveryRoutes()

Returns customer name and delivery address



**Description**: This class represents a staff of the GoGoGrocery application.

**Visibility**: Public

**Is Abstract**: No

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Data Type | Is Derived | Is Read Only | Visibility | Multiplicity | Default Value |
| staffName | First, Last name. | String | No | No | Private | 1 | None |
| staffEmail | Email to login to the app. | String | No | No | Private | 1 | None |
| staffPassword | The password to log in. | String | No | No | Private | 1 | None |
| staffPhone | Phone number. | String | No | No | Private | 1..\* | None |
| isAuthorized | To check if the user is an authorized staff member | Bool | No | No | Private | 1 | False |

**Operations:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Description** | **Return Type** | **Visibility** | **Is Query** |
| staffLogin | The staff member logs in to the GoGoGrocery application to view or modify the system. | None | Public | No |

**Processing Outlines**:

* staffLogin(in staffEmail: String, in staffPassword: String)

Staff inputs email and password

If valid

Allow the driver to login into the application

Else

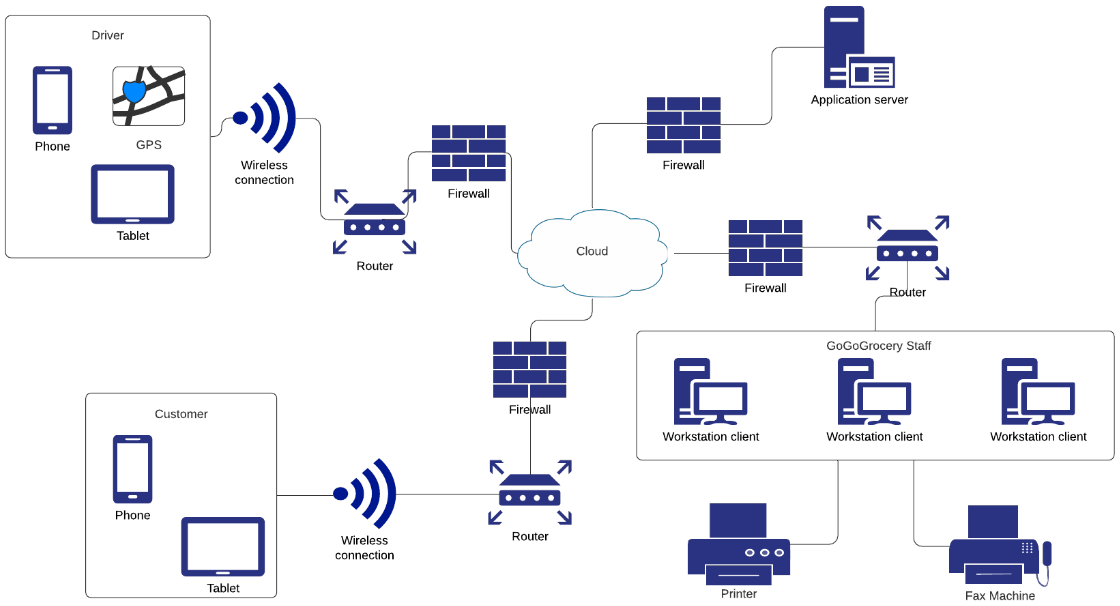
Prompt user to enter information again

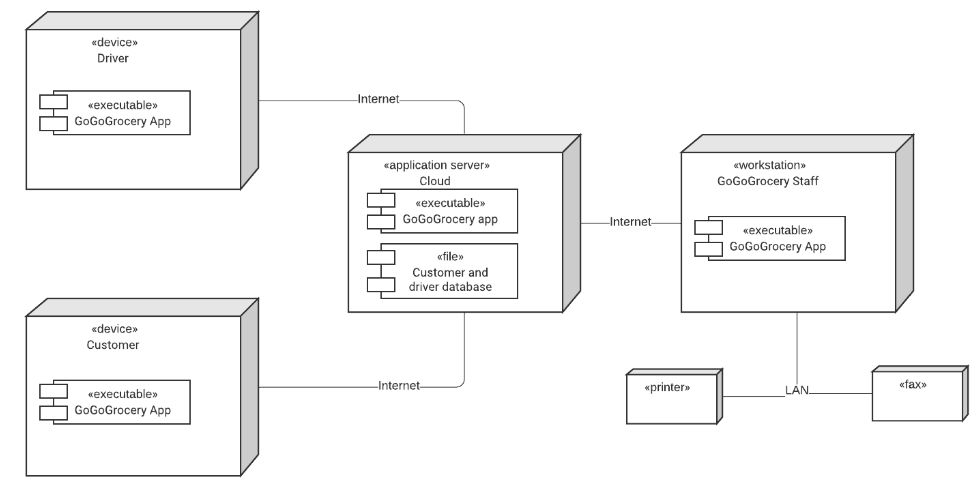
**3.0 Architecture Design**

**3.1 Introduction**

This section of the system specification will include two infrastructure models that showcase the architecture and nodes and artifacts of the GoGoGrocery application. This section will also cover the hardware and software requirements, and security plans GoGoGrocery is going to implement. OpenXcell recommends that the GoGoGrocery system should run as a two-tiered system where the customer’s device will host the presentation logic while the processes such as application and storage logic will be done by a cloud server. OpenXcell believes that this approach is best for the GoGoGrocery application as it is reliable, flexible, and provides varying options for storage and customer devices.

**3.2 Infrastructure Model**

**Deployment Diagram 1: Architecture Overview**

**Deployment Diagram 2: Nodes and Artifacts**

**3.3 Hardware and Software Requirements**

Required hardware components include:

* Computers in workstations for the GoGoGrocery staff. Standard issue performance would be sufficient.
* Printers and fax machines to print and receive reports.
* Mobile devices or tablets owned by customers.
* Mobile devices or tablets owned by the drivers. (GoGoGrocery will not provide these devices for the drivers).
* Renting or owning a cloud server.

Required software components:

* GoGoGrocery must support different mobile operating systems such as iOS and Android.
* A GPS tool in the GoGoGrocery application.
* The cloud service provider must use database software to store information.
  1. **Security Plans**

There will be some major threats that may hinder the operation of the GoGoGrocery Application. These major threats include unauthorized user access into the system or the loss of sensitive information. There are only so many things GoGoGrocery and OpenXcell can do as the application is heavily reliant on the transference approach. This means GoGoGrocery and OpenXcell are relying on a third party to handle the security aspect of the system. The list of threats to the application and how to control them are the following:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Threats  Components | Power loss | Theft | Unauthorized access | Virus | System failure | Natural disaster |
| Server | 2, 10 | 1, 10 | 3, 7, 8 | 1, 4, 7, 8 | 1, 10 | 1, 10 |
| Database | 10 | 10 | 10 | 1, 4, 7, 8 | 10 | 1, 10 |
| Customer Devices | X | X | 3, 7, 8 | 3, 4 | X | X |
| Driver Devices | X | 9 | 3, 7, 8 | 3, 4 | X | X |
| GoGoGrocery computers | 1, 2 | 1, 6, 8 | 3, 7, 8 | 3, 4, 5 | 10 | 1, 10 |

Controls:

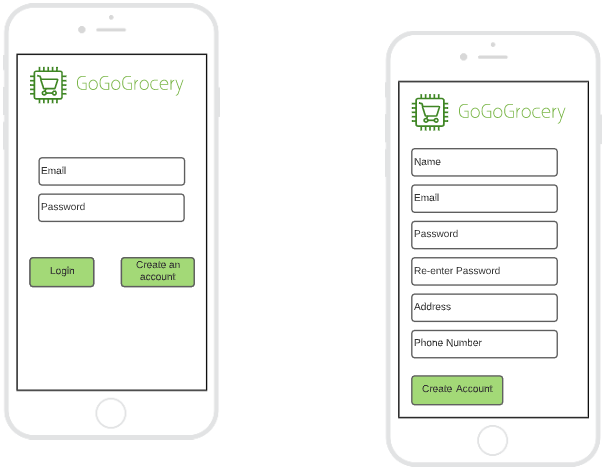
1. Disaster recovery plan
2. Back-up power generators
3. Application layer firewall
4. Anti-virus software
5. User training on viruses
6. Hardware insurance
7. Data encryption
8. Strong password software
9. Driver safety procedures
10. Reputable cloud service provider

**4.0 User-Interface**

**4.1 User-Interface Requirements and Constraints**

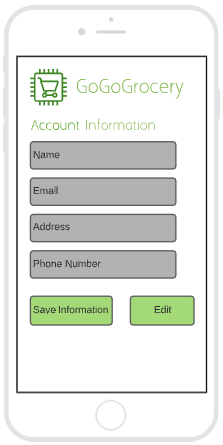
This section of the system specification will show a basic outline of the user-interface design of the application will look like. The designs shown are from the customer, driver, and GoGoGrocery staff screens. The aim is to create a user-interface that is easy to use and efficient as we understand that not everyone is used to using technology. Thus, the user interface must be understandable to everyone of all levels of technological backgrounds. Other than ease of use and efficiency, we want the design of the user-interface to look aesthetically pleasing. This allows the customers to be more attracted to using the application which will help with the longevity of the GoGoGrocery application. The “screens” shown below are a draft of what we believe is the best design choice for the application.

**4.2 Forms: Screen/User-Interaction Design**



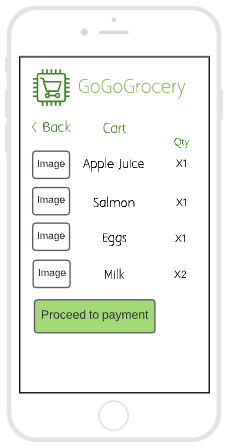
Login screen for users

Account creation screen for users



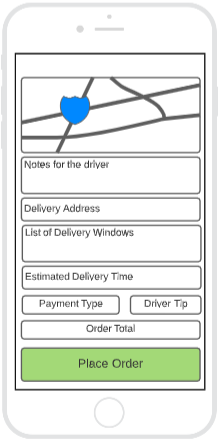
Once the user has logged in or created an account, they can view their account information. The text boxes are greyed out to indicate that the information cannot be changed unless the user selects the edit function.

This is the main screen for the customers. The screen will display the preferred stores or stores nearest to the customer with a logo of the store and its name.



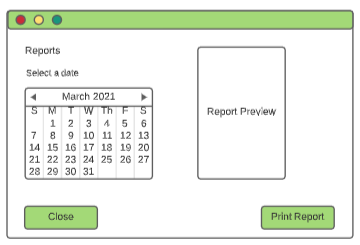
After the customer has chosen a store, the customer can choose an item and add it to the cart. The item selected will show a screen shown in this image.

This screen shows the cart with all the items the customer has added. The cart will show an image of the item, the name, and the quantity added. The customer will then press on the proceed to payment function to enter their payment details.



This screen shows the payment page for the customer.

This screen shows the driver’s view when delivering an order. It shows a GPS with the directions to the delivery address. The driver can also check the customer’s order again.



This is the main screen of the GoGoGrocery staff when a report is to be printed. The screen shows a calendar for the staff to click on to search for a report from that day. A preview of the report is also shown.

**4.3 Reports: “Printed Output” Design**

**GoGoGrocery** <Date>

**Customer Report**

<Customer Name>

<Date of Order>

**Items Ordered:**

Apple Juice x1 <Price>

Salmon x1 <Price>

Eggs x1 <Price>

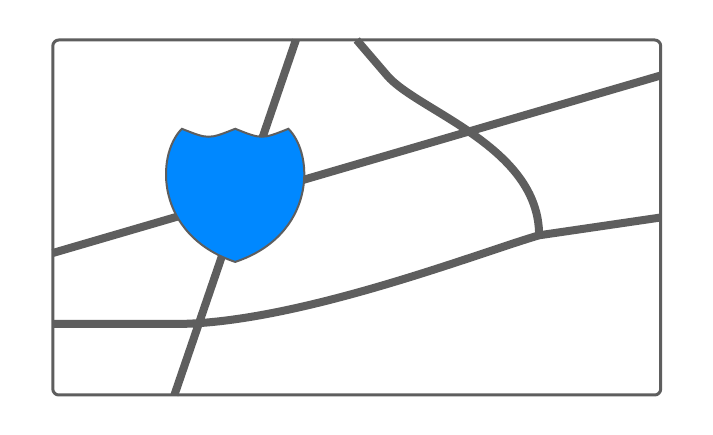
Milk x2 <Price>

**Total Price:** <Total Price>

**Delivery Routes:**

Delivery Address

<Customer Address>

Delivery Map

Estimated Delivery Time

<Delivery Time>

**5.0 Appendices**

**5.1 Bibliography**

Cameron, Andy. *Use Case Diagrams and Descriptions*. Oct. 2020, https://canvas.spu.edu/courses/45434/files/2191819?module\_item\_id=495711. PowerPoint Presentation.

Cameron, Andy. *Security*. Nov. 2020, https://canvas.spu.edu/courses/45434/files/2253424?module\_item\_id=502922. PowerPoint Presentation

Cameron, Andy. *Structural and Class UML Diagrams*. Nov. 2020, https://canvas.spu.edu/courses/45434/files/2220986?module\_item\_id=499138. PowerPoint Presentation

Larman, Craig. *Applying UML and Patterns: an Introduction to Object-Oriented Analysis and Design and Iterative Development 3rd Edition*. Addison Wesley Professional, 2004.

**5.2 Supporting Documentation**

“Online Diagram Software & Visual Solution.” Lucidchart, Lucid Software Inc., www.lucidchart.com/.

Q & A Session (This Q & A session is also available in the System Proposal)

1. **How will GoGoGrocery drivers be paid?**

* GoGoGrocery drivers will be paid by the hour depending on how many orders they have received. Their earnings will also be increased through customer tips.

1. **Will your application be available for all platforms?**

* Our application will run on popular mobile operating systems such as iOS, and android. There is no plan to develop a website version of the application yet.

1. **Should an account be created to access the application?**

* Yes, an account must be created to use the application. We require an account to access the application because it allows the application to track your favorite items and stores.

1. **Is there a certain budget/requirement to hire drivers?**

* All drivers for GoGoGrocery will be people from the community who sign up to be a driver. As long as the individual has the minimum requirements, required documents, and passes the screening.

1. **When will the application be available for public use?**

* There is no set date for the application to be public yet. A timeframe is set (March 2021) but it may be delayed due to the ongoing COVID-19 pandemic.

1. **What payment methods are available?**

* Available payment methods include Apple Pay, PayPal, and debit/credit card. There is no option for cash payment as the driver will need the customer’s funds to pay for the groceries while in the store.