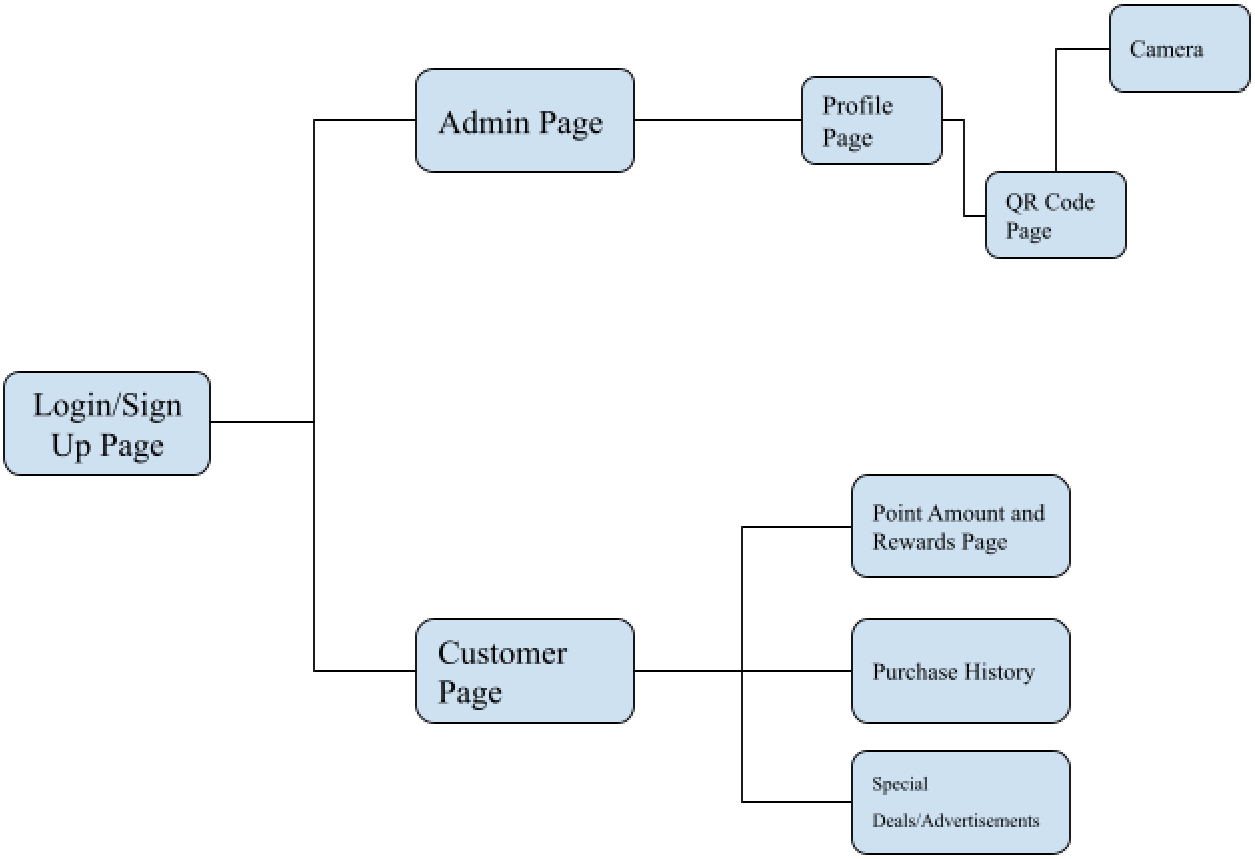
Criterion B - Design Overview

[B.X] Appendix index.

Modular Design

The provided diagram illustrates the incorporation of additional modules into the program, detailing their arrangement within the system. Certain modules will exclusively be accessible through the custodian/admin login, while others will be confined to the customer/client login. Following login, a comprehensive menu will enable access to all modules. Module design, descriptions, and definitions have been shown and approved by client in [B.2]



Proposed Modules

Module Name	Explanation
Login/Sign Up Page	The first module that the user will see when the program is opened. After entering their login credentials or signing up, the user can see the next sections of the application for the client side. Admin will have its own pre-built key which will require special login information.
Admin Page	The primary page of the admin/owner will allow

	admins to access all profile pages. Profiles will be found through a search function to search for a specific profile, returning to the profile page if found or returning an error if unable to be found.
Profile Page	This module will allow admins to look at the primary information of a specific profile like the last transaction performed on this account and total loyalty points. This page will allow admins to access the QR code system which will enable for admins to use their camera to add or decrease loyalty points based on prebuilt QR codes. In order to verify an account's loyalty points there admins can refer to purchase history, which will store the more recent input from the QR code.
QR Code Page	This module will run an algorithm to calculate the total amount spent in a given set and calculate the total amount of loyalty points accumulated from a given purchase. There will also be manual increase and decrease options to loyalty along with a button to go into the QR reader to read the data of purchased goods.
Camera	The Camera module will contain a camera which can read number values off of QR codes. The camera page will exit when either the exit button is clicked or a value is read.
Customer Page	The primary page of a customer of the Sari-Sari store will allow for customers to access all information related to their account including, purchase history, loyalty points, rewards able to be purchased for loyalty points, and advertisements or special current deals.
Point Amount and Rewards Page	This page will display customer's current total of points including the different possible rewards they can currently get with the rewards points they have.
Purchase History	This module will have the most recent purchase imputed onto this account in Filipino currency.
Special Deals/Advertisements	This module will be a section of the application in which customers are able to check for new deals at the store that they may have not known about.

Database Entity Relationship Diagram

Data Dictionary for the customer entity:

Field Name	Data Type	Description	Notes
Username	String	Identification of account	Can't be the same as another user
Password	String	A string of characters, symbols, and number that verifies access to given account	Must have one capital, one lower case, one number, and one special symbol
currentLoyaltyPoints	Int	The current amount of loyalty points in account	

Methods for customer entity:

Method Name	Description	Notes
LogIn()	Activates the verification of account, checks for proper username and password	
Register()	Sends username, password, and date into database, validating the new information	This will check if it fills out all password requirements
signOut()	Signs the user out of the current account	

Data Dictionary for the admin entity:

Field Name	Data Type	Description	Notes
Username	String	Identification of account	Can't be the same as another user
Password	String	A string of characters, symbols, and number that verifies access to given account	Must have one capital, one lower case, one number, and one special symbol
adminAccess	Boolean	Gaines access to all admin related modules.	Will check whether this boolean is set to true in each restricted module to ensure only admins have power to look at other accounts

Methods for admin entity:

Method Name	Description	Notes
LogIn()	Activates the verification of account, checks if valid for admin access	
signOut()	Signs the user out of the current account	
qrLoad()	Activates QR code system, increasing or decreasing the loyalty points in a given purchase into the chosen account's system	

Data Dictionary for the QR Code entity:

Field Name	Data Type	Description	Notes
priceGood	Float	Value which holds the current cost of the good the QR code reads	
currentCart	Float	Total accumulated costs of the goods	
accumulatedLoyalty	Int	Calculates the amount of loyalty points gained from the current cart after all purchases have been processed. This value will be added or subtracted from currentLoyaltyPoints and if added will also increase lifetimeLoyaltyPoints	Calculations will be rounded down to the integer using the equation: $\frac{currentCart}{50} =$ accumulatedLoyalty, as 50 pesos is equivalent to one loyalty point

Methods for QR code entity:

Method Name	Description	Notes
calculateLoyalty()	Will calculate the total amount of loyalty accumulated from the single purchase for processing	Uses equation referenced above: $\frac{currentCart}{50} =$ accumulatedLoyalty, as 50 pesos is equivalent to one loyalty point
increaseLoyalty()	From the subclass increase QR code, this subclass comes from when the increasePoints bolean is true, increasing the	

	currentLoyaltyPoints of the user by the amount gained from calculateLoyalty()	
decreaseLoyalty()	From the subclass decrease QR code, this subclass comes from when the increasePoints bolean is false, decreasing the currentLoyaltyPoints of the user by the amount gained from calculateLoyalty()	

Data Dictionary for the rewards entity:

Field Name	Data Type	Description	Notes
qualifyForReward	Boolean	Identification of whether for a given reward is currentLoyaltyPoints equal or greater to the cost of the reward	Rewards which show true to this statement shine yellow to show user they qualify for the reward
rewardName	String	The variable which holds the name value of a specific reward	
rewardCost	Int	The variable which holds the costs of a given reward type.	This is what qualifyForReward boolean checks for.

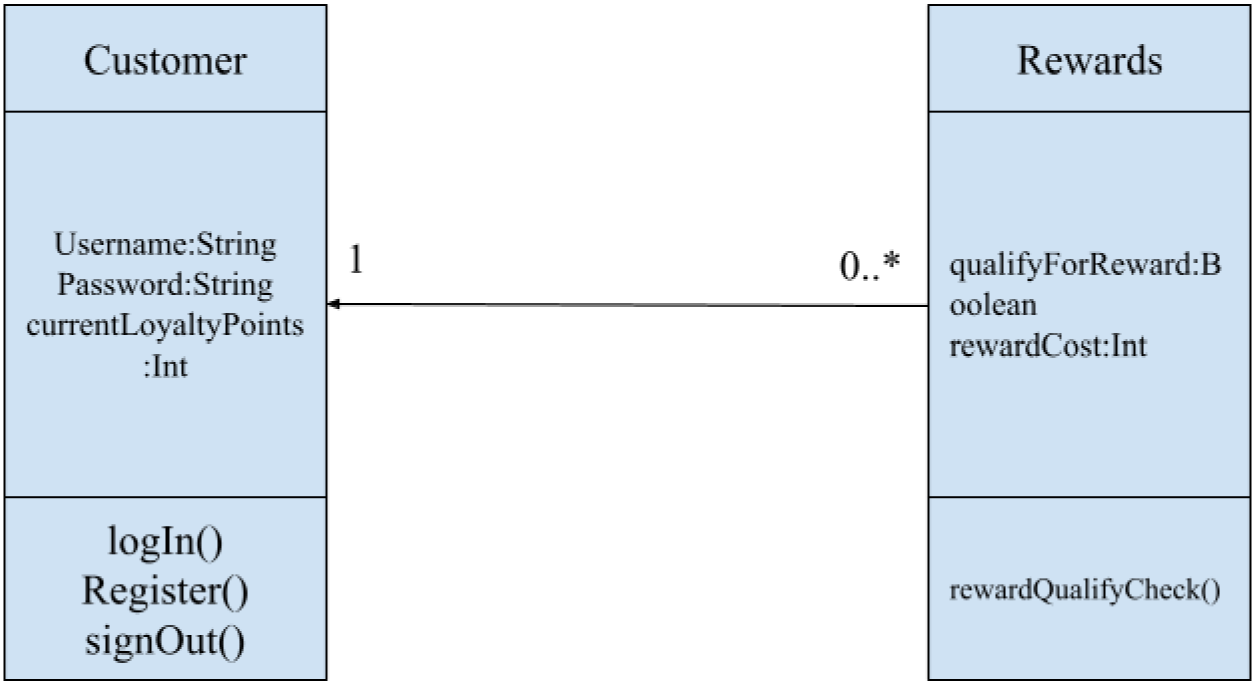
Methods for rewards entity:

Method Name	Description	Notes
rewardQualifyCheck()	Will verify whether or not a given reward entity is able to be purchased with the current loyalty points. This will highlight the reward as yellow	Checks the rewardCost variable to the currentLoyaltypoints of the current account

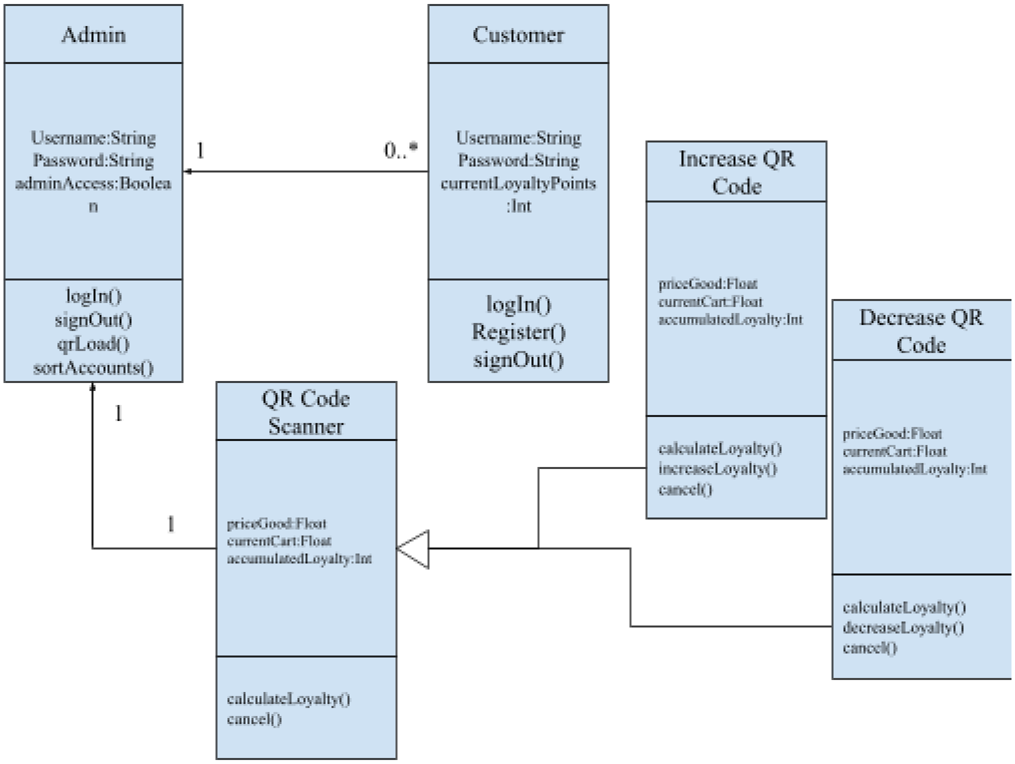
UML Diagram

I explained the relationships between every object and method, for the customer and admin respectively, of the program to the client, and have been approved for each by the client [B.3].

Account UML Diagram

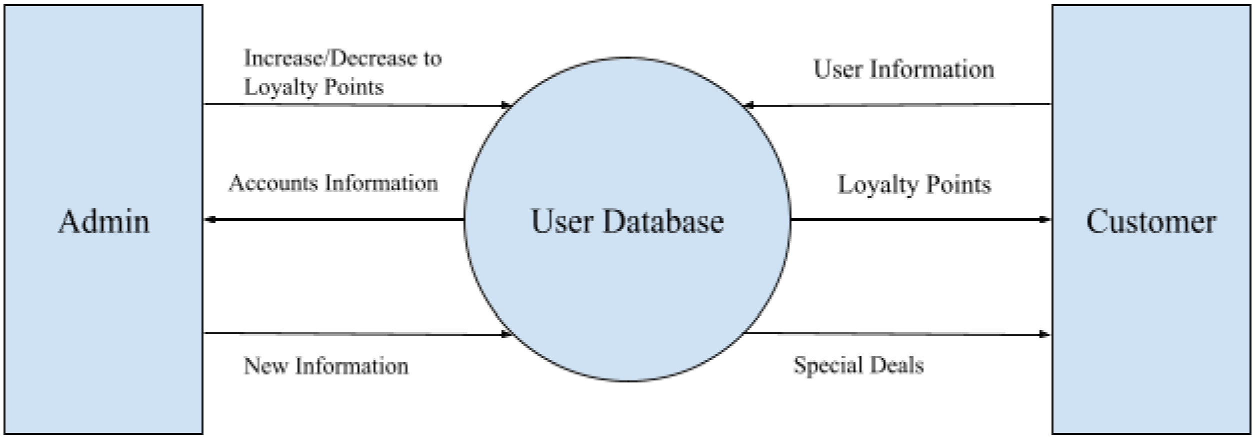


Admin UML Diagram



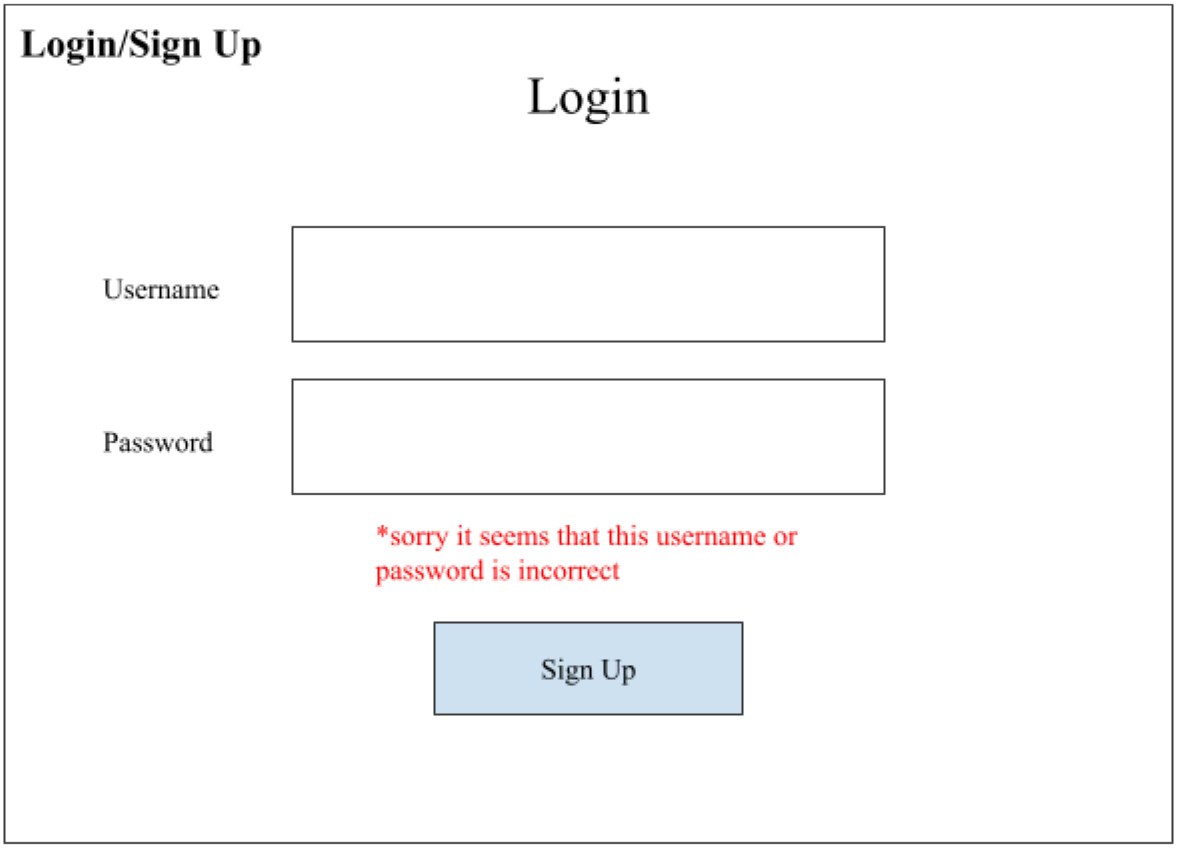
General Data Flow Diagram

I showed the data flow that will occur in the database to the admin and customer respectively, and have been approved by the client [B.4].



Prototype Designs

I explained in more detail each modular design, utilizing prototypes to show the client how the program will most likely look like. To explain necessary logic to client, I utilized both logical flowchart and simple pseudocode. Each design and logic system of the modules have been approved by the client [B.5].



Login/Sign Up

Login

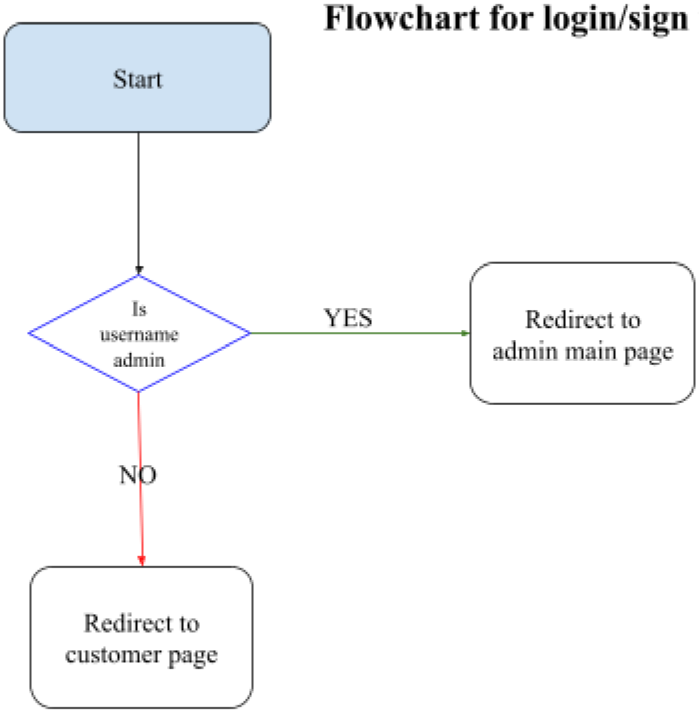
Username

Password

*sorry it seems that this username or password is incorrect

Sign Up

Flowchart for login/sign up

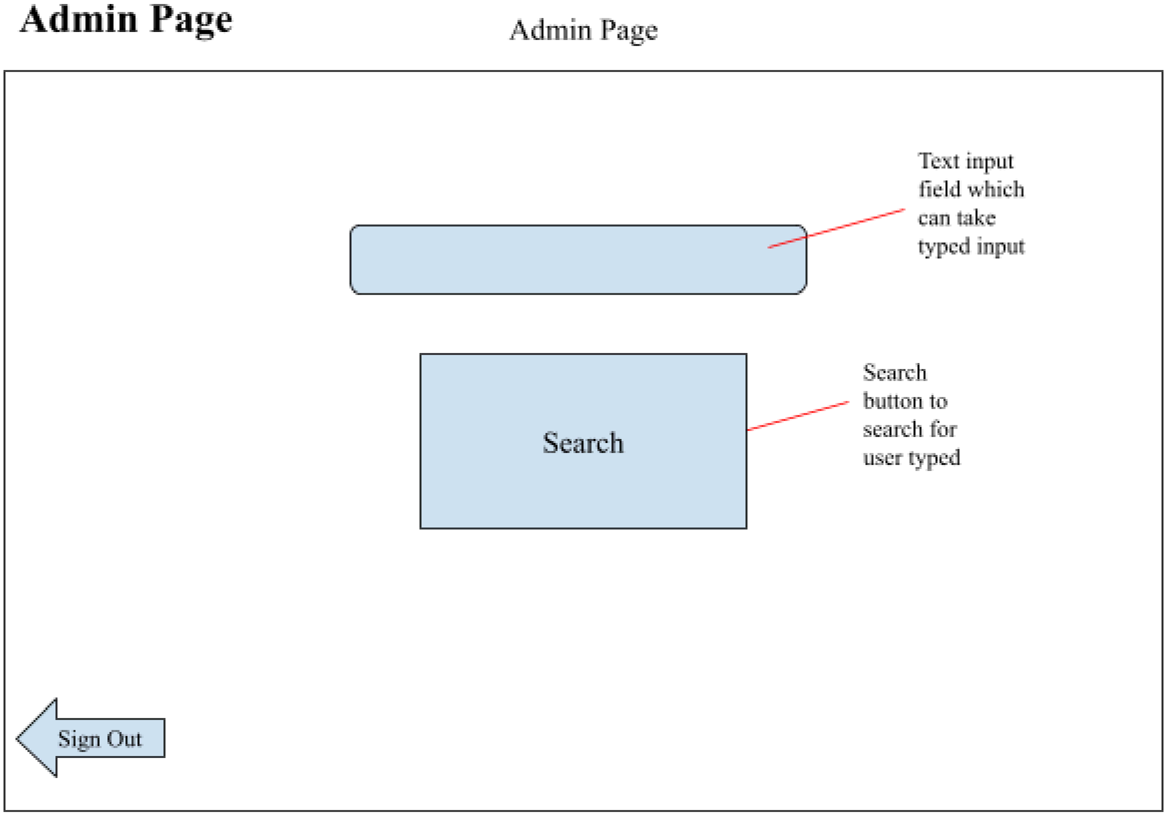


Pseudocode for login system:

```
If (Username = Admin username AND Password = Admin password)
        adminAccess = TRUE
```

```
*Sends user to admin module section
else if (Username = Customer username AND Password = Customer password)
adminAccess = FALSE
*Sends user to customer module section
else
MESSAGE: “*sorry it seems that this username or password is incorrect”
*Keeps user in sign in module
```

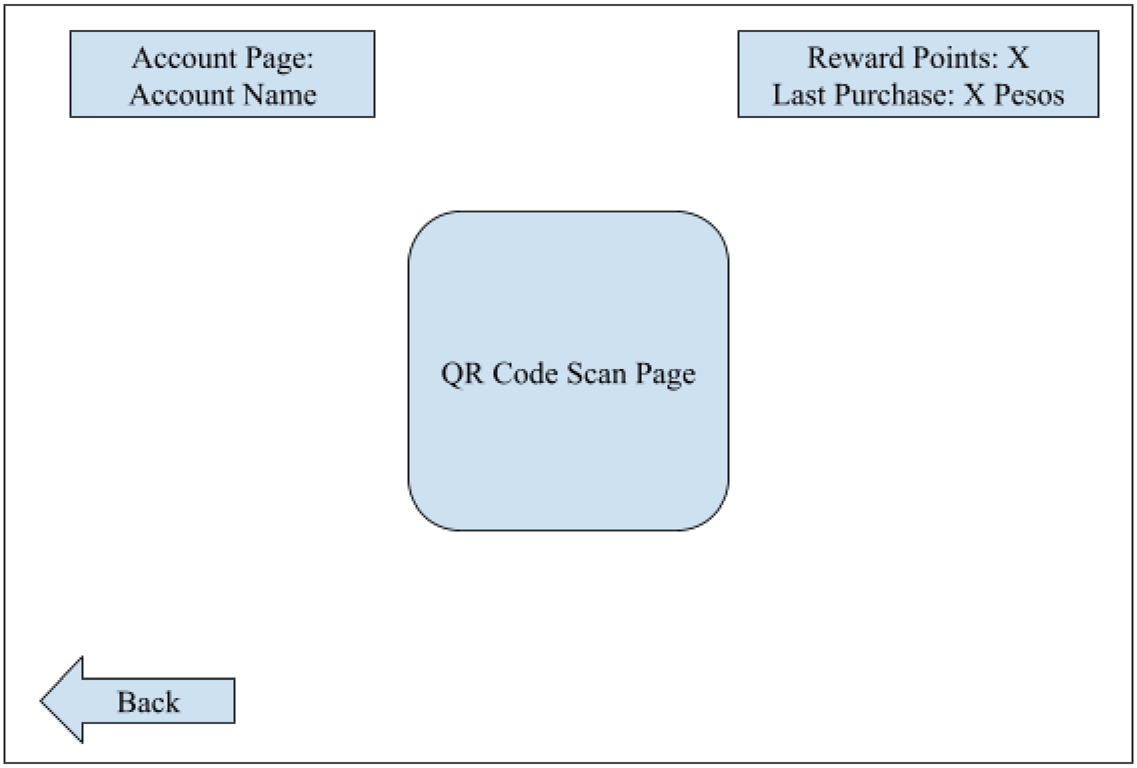
Admin Page



Text input field which can take typed input

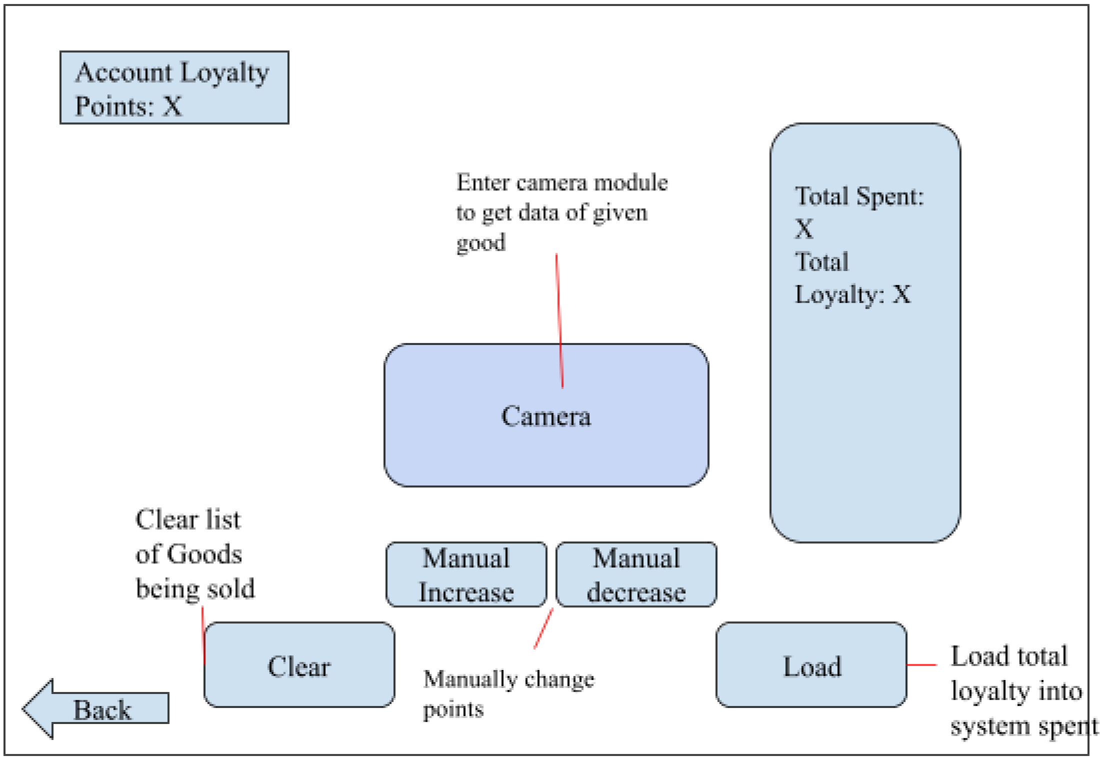
Search button to search for user typed

Profile Page



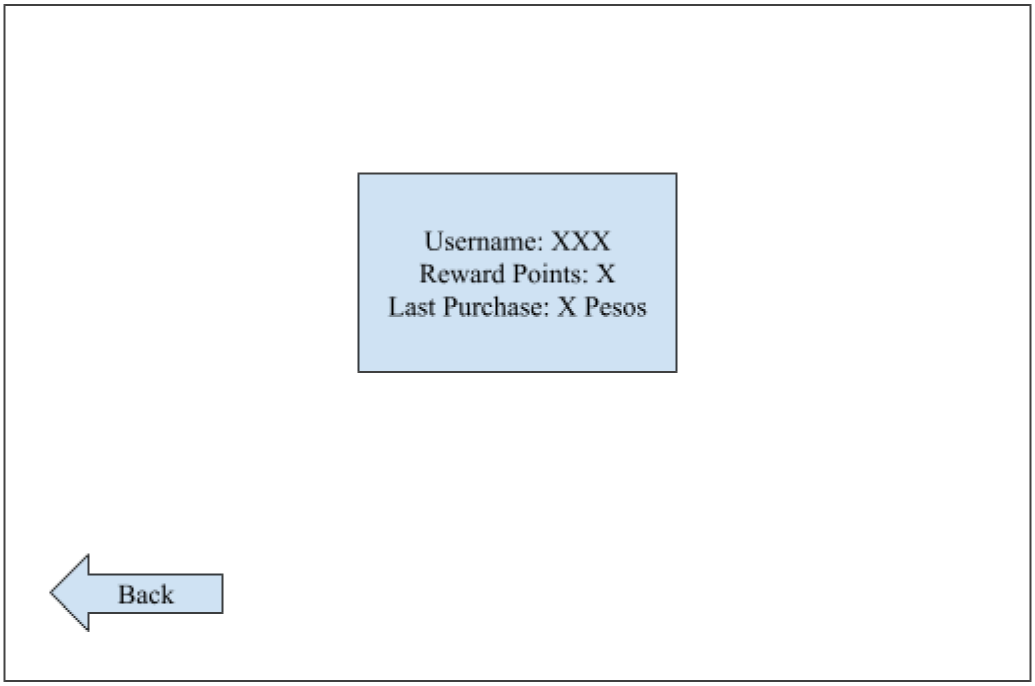
QR Code Page

QR Code Scanner

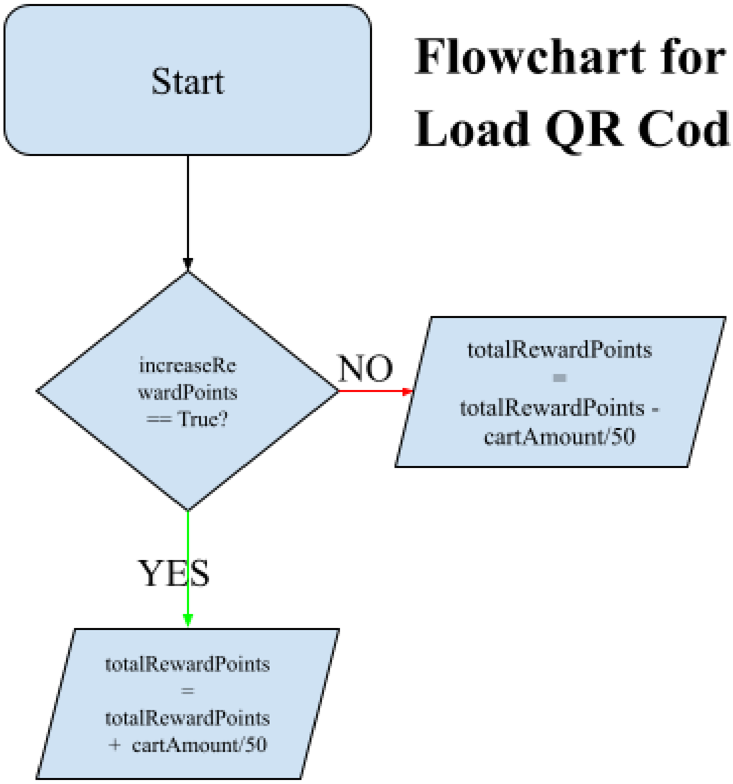

Account Loyalty Points: X
Enter camera module to get data of given good
Total Spent: X
Total Loyalty: X
Camera
Clear list of Goods being sold
Manual Increase
Manual decrease
Clear
Load
Load total loyalty into system spent
Back
Manually change points

Purchase Data/Purchase History Page

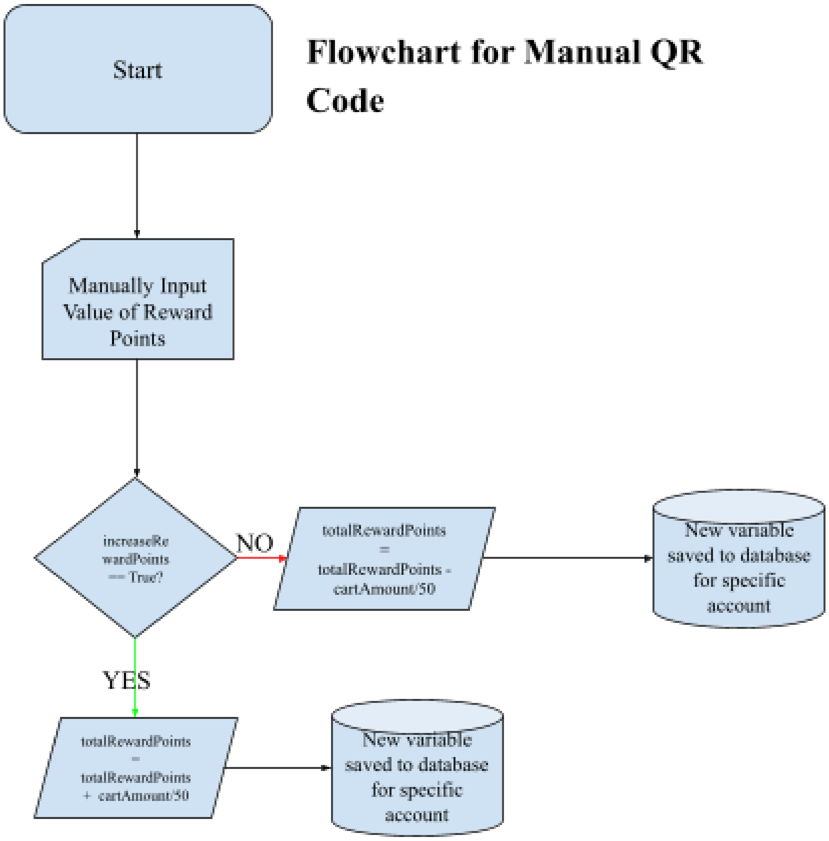
Purchase History



Flowchart for Load QR Code


Start
increaseRewardPoints == True?
NO
totalRewardPoints = totalRewardPoints - cartAmount/50
YES
totalRewardPoints = totalRewardPoints + cartAmount/50

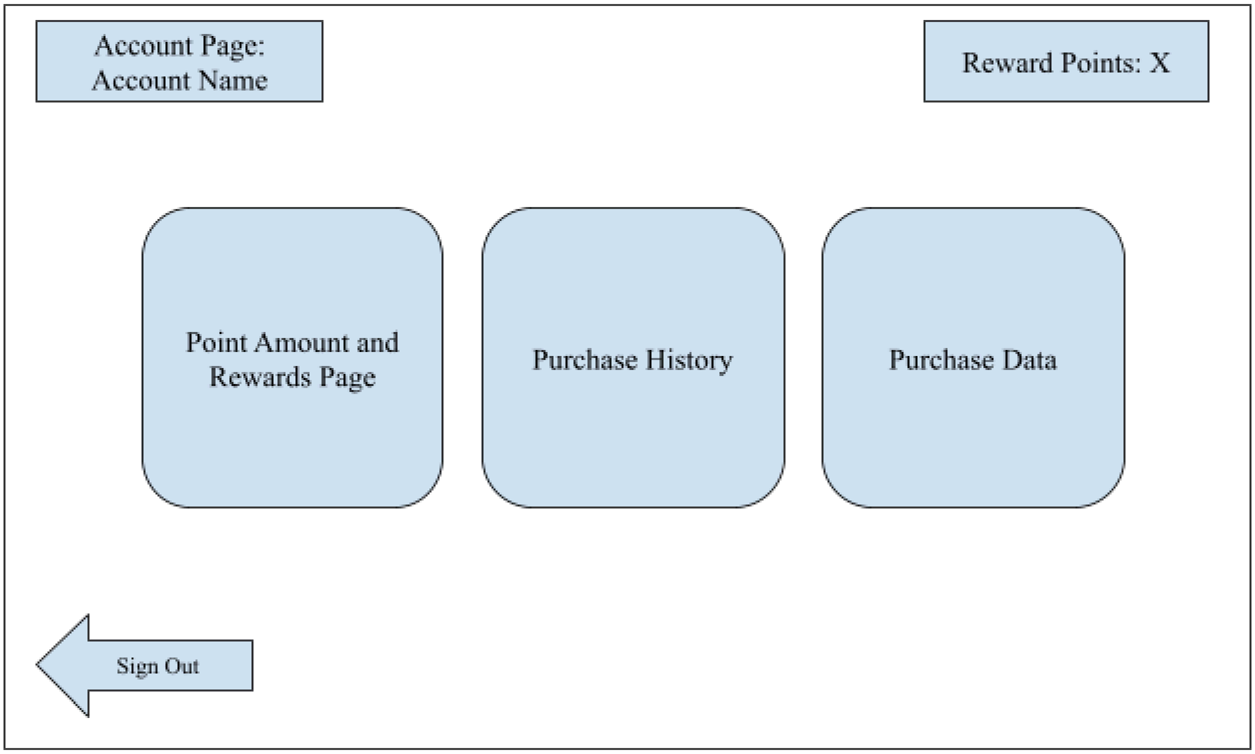
Flowchart for Manual QR Code


Start
Manually Input Value of Reward Points
increaseRewardPoints == True?
NO
totalRewardPoints = totalRewardPoints - cartAmount/50
New variable saved to database for specific account
YES
totalRewardPoints = totalRewardPoints + cartAmount/50
New variable saved to database for specific account

Pseudocode for QR Code system: (*note openCart and increasePoints boolean variables are used as temporary variables to simplify the increase and decrease process for pseudocode and flowchart)

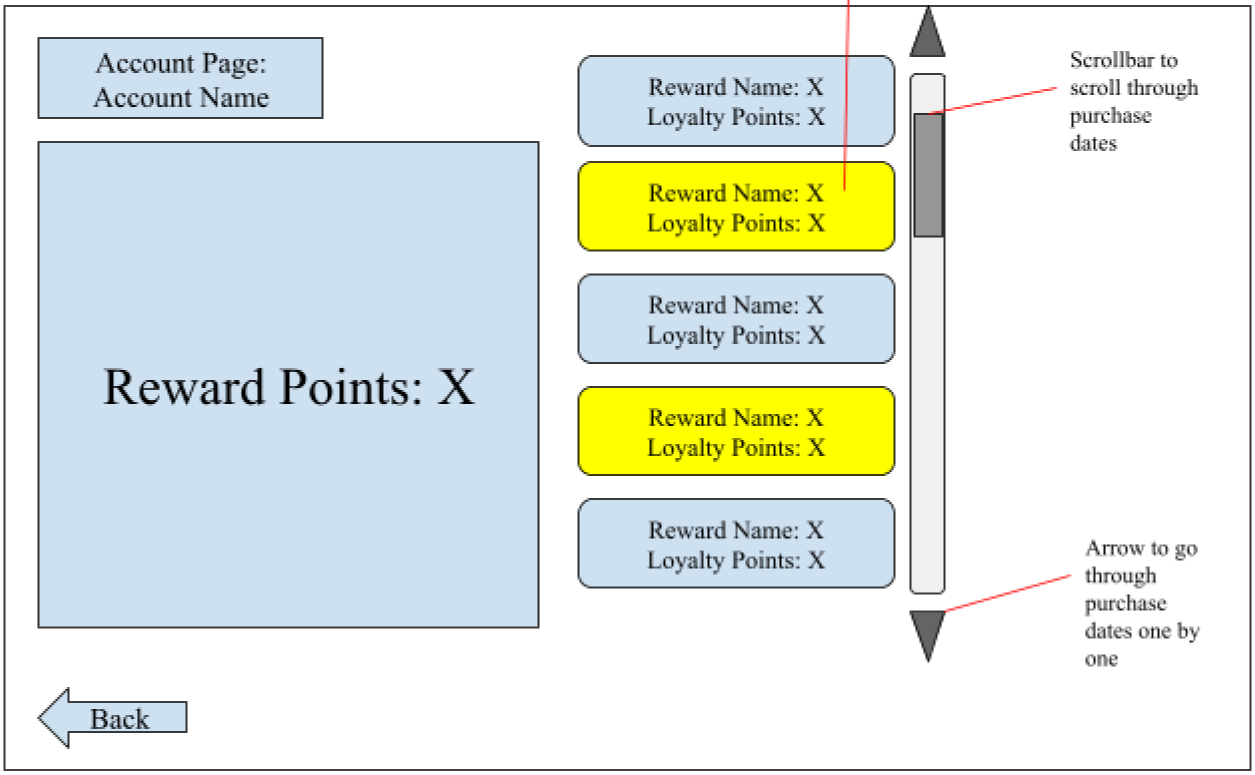
```
Loop while (openCart = True):
    currentCart += priceGood
    accumulatedLoyalty = int(currentCart / 50)
    end loop
if (increasePoints = True):
    lifetimeLoyaltyPoints = lifetimeLoyaltyPoints +  accumulatedLoyalty
    currentLoyaltyPoints = currentLoyaltyPoints + accumulatedLoyalty
    else if (increasePoints = False):
        currentLoyaltyPoints = currentLoyaltyPoints - accumulatedLoyalty
    end if
```

Customer Page



Points and Reward Page

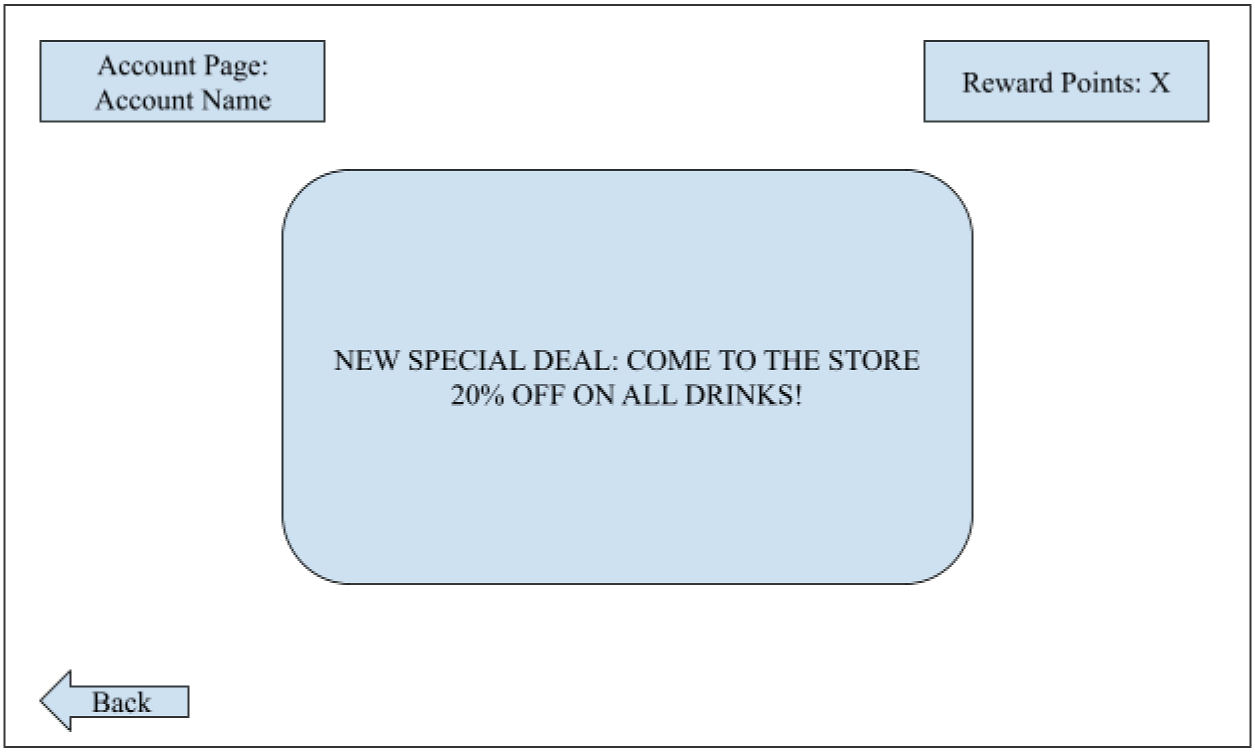
Highlights yellow if qualifies for reward



Scrollbar to scroll through purchase dates

Arrow to go through purchase dates one by one

Special Deals Page



Test Plan

Test plan has been shown to client, to show how verifying the functionality of the program will be performed, and has been approved [B.6].

Action to Test	Method of Testing	Expected Result
Log in - test if users with correct credentials are able to access their appropriate modules and if an error message shows for incorrect credentials. Also will rest whether admin login key works	Input correct credentials (customer-side) Input incorrect credentials (customer-side) Input correct credentials (admin-side) Input incorrect credentials (admin-side) Input null values	Customers are directed to the customer page and to their own profile upon inputting correct credentials. Admin is sent to the admin page and to the list of different profiles. Error message shows upon inputting incorrect credentials.
Admin Page	Test the buttons for all the modules	Verify functionality for admin page access Confirm access to admin-only features such as the QR system.
Profile Page	Test buttons for all the modules Check information in each section.	Validate the display of user information (name, email, total points, etc.)
Purchase Data	Confirm data integrity by comparing stored data with input data (points and date in time). Check for accurate retrieval of purchase data based on user queries.	Gain the ability to access information about past purchases relayed to the account
QR Code Page	Manual testing, scan and verify QR code information is being loaded into account Test calculations of reward points from total purchases	Allows for QR system to access camera and load information from made QR codes. Verify software to make calculations based on total amount spent
Customer Page	Check if data that is displayed is	Validate proper display of

	of validity Test buttons for access to rest of the modules	customer information. Ensure that buttons are functional
Point Amount and Rewards Page	View page and verify point amount and awards are properly calculated and imputed from the QR system. Verify that all rewards available are valid.	Verify accurate calculation and display of points. Confirm proper display of available rewards.
Purchase History	Check that previously imputed data from QR code is shown.	Ensure accurate recording of purchase transactions in the database Verify correct display of purchase history on the user interface.
Special Deals/Advertisements	Check if special deals and advertisement banners are functional.	Confirm proper display of special deals and advertisements on the UI