**Design Document**

The C++ code for the final project is divided into 5 major parts:

1. Int main() (main.cpp)
2. User base class (user.h and user.cpp)
3. SelfCheck derived class (selfcheck.h and selfcheck.cpp)
4. CashierCheck derived class (cashiercheck.h and cashiercheck.cpp)
5. UserInfo structure (userinfo.h)

*Main.cpp()*:

The user will mainly interact with this part of the program, which will act as a control flow as users pick different options. There are few different variables and objects that are declared throughout the course of the main function

Variables:

* Int userIsRegistered: This variable is either 0 for not registered users, 1 for regular loyalty users, and 2 for executive loyalty users. The value of this variable is determined by comparing the entered ID and password and the ID and password array that was set up prior to beginning the program (regRecords and execRecords).
* UserInfo regRecords[10]: This array houses the information of all regular loyalty users (to a maximum of 10 individuals) by using the UserInfo structure. This is used in int main() to verify if the entered ID and password has already been registered
* UserInfo execRecords[10]: This array houses the information of all executive loyalty users (to a maximum of 10 individuals) by using the UserInfo structure. This is used in int main() to verify if the entered ID and password has already been registered
* userInfo cashierRecords[10]: This array houses the information of all cashiers (to a maximum of 10 individuals by using the userInfo structure. This is used in the CashierCheck object.

Objects:

* User initialUser: this object acts as the initial class which determines if the user would like self-checkout or cashierCheckout. There is a parametrized constructor, which will be discussed later.
* SelfCheck selfUser: This object is used for individuals that would like to have self-checkout. It has a parametrized constructor, which will be discussed later.
* CashierCheck cashierUser: This object is used for individuals that would like to have a cashier check out. It has a parameterized constructor, which will be displayed later.

The main function of int main() is to gather all initial ID and passwords and determine which type of check out they would like by invoking many objects.

*User base class:*

The User class acts as a base class and stores all major functions that require the private variables of the User class. The following is the list of private, protected, and public variables and methods that are present in this class:

Private:

* Int option: Variable for user to select cashier or self-checkout options
* Float totalPrice: Variable for the price the user will have to pay
* Int itemsBought[10]: array of number of items user bought for 10 different foods
* String inventoryName[10]: array of names of the 10 different foods in the cafeteria. Note: the array index of itemsBought corresponds to the name lying in the same index of inventoryName
* Float inventoryPrice[10]: array of prices for the 10 different foods in the cafeteria. Note: the array index of itemsBought and inventoryName corresponds to the price of the same index.

Protected:

* Int userID: Stores user’s ID
* Int userPassword: Stores user’s password
* String userEmail: Stores user’s email
* Int loyaltyType: Stores loyalty type of user based on the same code of userIsRegistered in main.cpp

Public:

* Int prevRegularID[10] and int prevExecID[10]: stores all IDs for all members
* Int prevRegularPasswords[10] and int prevExecPasswords[10]: stores passwords for all members. The information in a particular index in one of these arrays is the password to the ID present in the same index of the corresponding ID array.
* Int cashierID[10] and int cashierPasswords[10]: stores ID and passwords in the same way as the member ID and passwords
* User(userInfo regularID[10], userInfo execID[10], userInfo cashierRec[10], int enteredID, int enteredPassword, int enteredLoyalty): Constructor to initialize members of the class and store the user’s entered ID and password
* void **buyItems**(): Function to state the number of items that the user wants to buy
* void **calculatePrice**(): Function to calculate the total price of all goods, including loyalty discounts
* bool **checkUniqueID** (int proposedID): Function to check if an entered ID has already been used in prevRegularID or prevExecID.
* void **getInfo**(): Asks user if they want to have self-check or cashier checkout
* void **greeting**(): Function for initial greeting and calls function that gets ID and password and how user wants to checkout items
* void **getUserCredentials**(): Gets user’s ID and password
* void **purchaseSummary**(): Returns summary of all purchases done by user
* int **returnID**(): Returns ID of user
* int **returnLoyalty**(): Returns loyalty level of user based on same coding as loyaltyType
* int **returnOption**(): Returns whether user wants self-check (1) or cashier checkout (2).
* int **returnPassword**(): Returns password of user
* void **setID**(int id): Sets ID of class from user
* void **setLoyalty**(int level): Sets loyalty of class from user
* void **setPassword**(int uPassword): Sets password of class from user
* void **updateRecords**(int currentID, int newID, int currentPassword, int newPassword) Updates previous records with the ID and password.
* bool **validateUser**(int proposedID, int proposedPassword): Validates if user is registered with entered ID and password

*SelfCheck* derived class:

The SelfCheck class acts as derived class that is publicly inherited from the User class. This class stores all major functions for self-checking out. The following are the public methods that are present in this class:

* SelfCheck(userInfo regularID[10], userInfo execID[10], userInfo cashierRec[10], int enteredID, int enteredPassword, int enteredLoyalty): This constructor is linked to the User base class to initialize variables
* void **greeting**(): This function overrides the greeting function declared in the User class. This function will call the buyItems() function and the purchaseSummary() function from the base class.

*CashierCheck* derived class

The CashierCheck class acts as derived class that is publicly inherited from the User class. This class stores all major functions for self-checking out. The following is the public variables and methods that are present in this class:

* CashierCheck(userInfo regularID[10], userInfo execID[10], userInfo cashierRec[10], int enteredID, int enteredPassword, int enteredLoyalty): This constructor is linked to the User base class to initialize variables
* void **checkInfo**(): Displays the ID and the password of the user
* void **dataEntry**(): Asks cashier If they want to update, view or add information. Once done, records will be updated
* void **getLoyalty**(): Adds a new user if they want to sign up for loyalty
* void **greeting**(): Gets cashier login and then directs to dataEntry function
* void **summaryOfMembers**(): Summarizes ID of all members in records and sorts via bubble sort
* void **updateInfo**(): Function to update records with new users and existing users
* bool **validateCashier**(int cashID, int cashPass): Function to make sure that the cashier is registered by comparing their entered ID and password with existing records

*userInfo* structure

This structure is used for initializing users and cashier information. The following are the variables in the structure:

* int **userID**: Variable to store user’s ID
* int **password**: Variable to store user’s password