Microsoft account

[Email address]

Abstract

[Draw your reader in with an engaging abstract. It is typically a short summary of the document.   
When you’re ready to add your content, just click here and start typing.]

[Document title]

[Document subtitle]

**Table of content**

Contents

[**1.** **Problem Analysis:** 2](#_Toc68557800)

[**1.1.** **Functionality:** 2](#_Toc68557801)

[**1.2.** **Python Modules:** 2](#_Toc68557802)

[**2.** **Design:** 4](#_Toc68557803)

[**2.1.** **Rules:** 4](#_Toc68557804)

[**2.2.** **Description:** 4](#_Toc68557805)

[**3.** **Plan** 6](#_Toc68557806)

[**3.1.** **Algorithms:** 6](#_Toc68557807)

[**4.** **USER TEST DOCUMENT:** 18](#_Toc68557808)

[**Login** 18](#_Toc68557809)

[**Topup:** 19](#_Toc68557810)

[**View Information:** 19](#_Toc68557811)

[**View Information:** 20](#_Toc68557812)

[**Update Information:** 21](#_Toc68557813)

[**Buy Product:** 21](#_Toc68557814)

**Online Car Wash Administration System**

# **Problem Analysis:**

## **Functionality:**

This system is created to act as a car wash system.   
It allows users to sign up and login to use different functionalities of the project.

It also allows 2 different kinds of employees , i.e: Employee and Manager to sign up as well.   
Both these employees have almost identical available actions in the application as well as the Manager having a few extra functionalities than the Employee.

The users, employees and managers can add money into their washcard to buy services from the store.   
Each kind of user, gets a different kind of discount on the prices.   
  
Users can level up their membership by spending more money on the system, which will give them more discounts.   
  
The system uses local files to store all the sign up details of all kinds of users, they current amount in their washcards, the number of items on sale and their details as well as the total profit gained for each item.

## **Python Modules:**

1. **class Person:**

This class holds all the basic info that is common between all kinds of users whether it be a simple user, manager or employee

1. **class Employee(Person):**

This class extends from the Person class and is only used to create objects for manager and employee type users.

It stores their necessary info such as status and employee id.

1. **class PrintProfits(QDialog):**

This class opens up a dialog box, that prints the list of product names present in the products & prices.txt file as well as the individual profit for each item.

1. **class App(QMainWindow):**

This is the Main window that appears when the program starts and takes the information for any kind of user to register. It also gives the button to move onto the sign in screen

1. **class Signinwindow(QDialog):**

This is the dialog box that takes the necessary info needed for signin and signs the user into the system if correct credentials are provided. It takes username and password as input.

1. **class UserMain(QMainWindow):**

This screen shows users the options available to a normal user.

This includes, viewing info, purchasing products, and adding amount to their washcards.

1. **class ProductView(QMainWindow):**

This screens shows all kinds of users the list of currently available products.

It applies specific discount on all the prices depending on the user type, such as a regular user, employee, manager etc.

1. **class EmployeeMain(QMainWindow):**

This is the main window that an employee sees after logging into the system.

It provides all the functionality that the employee can perform, such as :

Viewing user details , showing user card’s details , printing washcard records and buying available products.

1. **class UserPrintinfowindow(QDialog):**

This screen displays the information of user onto a dialog box.

1. **class usercarddetails(QDialog):**

This dialog box, displays the details of the washcard of a specified user.

1. **class ManagerMain(QMainWindow):**

This window provides all the functionalities that a manager can do after logging into the system.

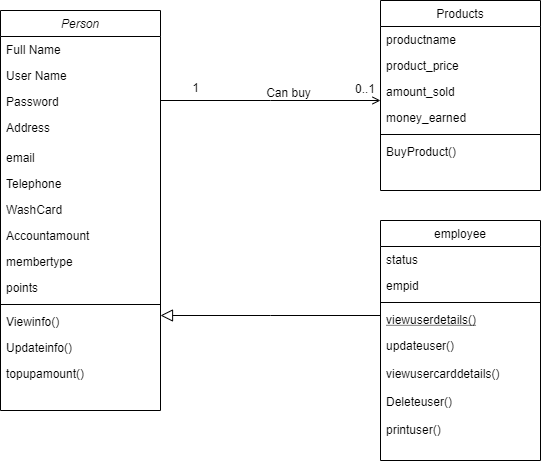
The manager can access this screen after signing in.

1. **class updateinfowindow(QDialog):**

this dialog box, shows the current information of any user and gives us the ability to update any field according to our will.

# **Design:**

**UML Class Diagram:**



## **Rules:**

* Person is a Base Class.
* Employee inherits all the attributes and functions of the person.
* Person can buy one or many products.

## **Description:**

There are 3 major classes of the object:

1. **Person:**

A person has multiple attributes like Full name, username , password , address , email, telephone, washcard , accountamount , membertype and points. It is base class and can perform following functionalities:

**View info():**

User/ Person can view his/her complete information.

**Update info():**

User can update all the attributes of the person mentioned above.

**Top up Amount():**

Person can add balance to his personal account.

1. **Employee:**

An employee has attributes like status and empid. A status of an employee can be employee or a manager. An employee performs different functionalities based on his/her status. Few of them are mentioned below:

**View user Details():**

An Employee can View Details of all the users present in the system.

**Update User():**

Employee can update information of a specific user.

**View User Card Details():**

Employee can also View information regarding the balance of users registered.

**Delete User():**

Employee can delete an existing user.

**Print User():**

Employee can also print details of an existing user.

1. **Product:**

An information saved on the file regarding the product is product name, product price, amount sold and money earned.

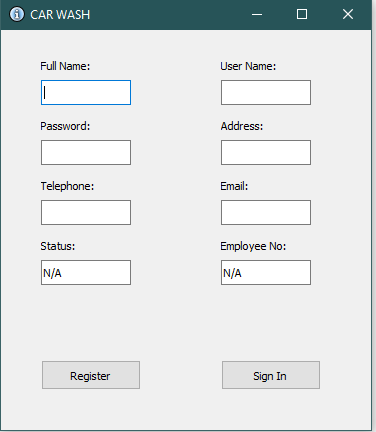
**Buy Product():**

A person or an employee can buy a product.

# **Plan**

## **Algorithms:**

1. **Main Screen :**



Enter all necessary info.

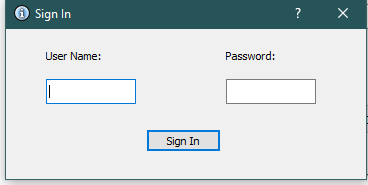
If simple user, leave status and employee no as empty.

Else if employee or manager, write Employee or Manager in status as well as 01 or 02 in Employee no.

Click register to register the specified user type.

Or, press Sign in to move onto sign in screen.

1. Sign in Screen :

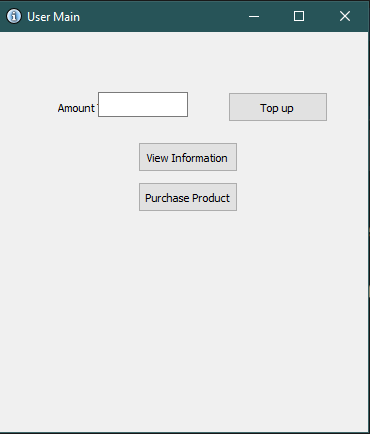


Enter username and password and press signin button.

If correct credentials, takes to the Main Page, depending on the type of user.

If incorrect info, the program closes.

1. **User Main :**



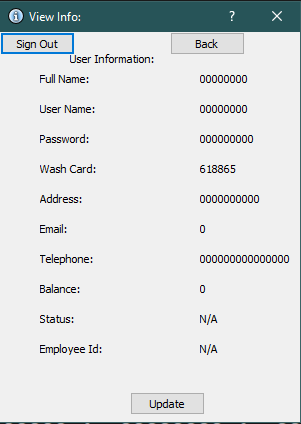
This screen appears if the user logs into the system as a regular user.

User can write amount to be added to his washcard.

User can view his profile info.

User can click purchase product button to see list of products that he can buy.

1. **View info:**



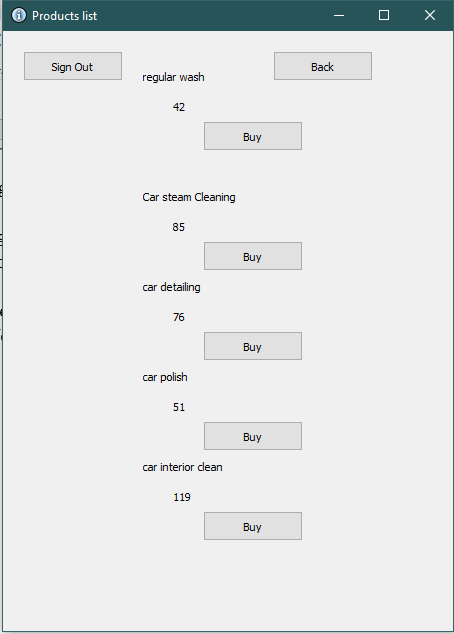
Views currently logged in user’s info.

Click sign out to sign out of the system and move to first register screen.

Click back button to go back to User Main screen.

Click update to open user information update screen, where user can update current information.

1. **Purchase products:**



Displays all the available products.

Applies discount on the prices of each product, based on user type: e-g:

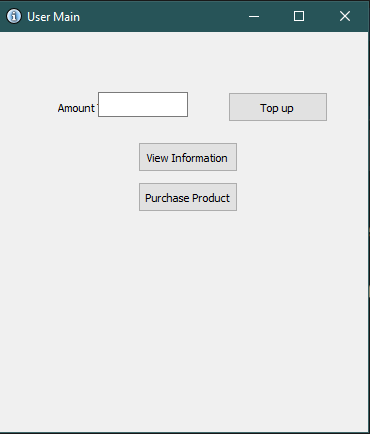
Regular user get 15 % discount, employees get 30 %, while managers get 50 % discount.

Click buy button to buy any product.

After clicking buy, it checks if user has enough amount in their washcard to buy product or not.

If enough amount is present, then this amount is deducted from their washcard and product is bought, else the message that not enough amount in washcard is printed.

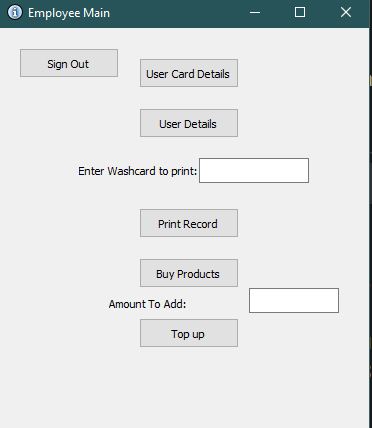
1. **Top up:**



User enters the amount he wants to add to their washcard in the amount textbox and presses the Top up button.

Program adds the amount to their already existing amount present in the record.txt file.

1. **Employee Main Screen:**

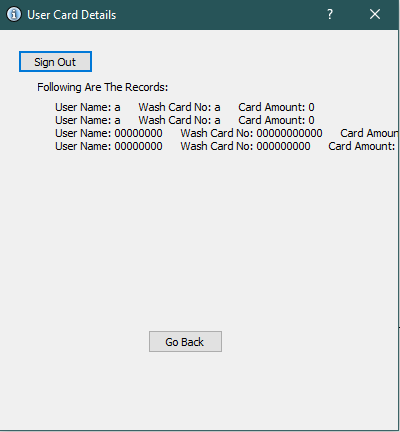


Screen displayed after an Employee logs into the system.

Gives option to view details of user’s cards, view user’s details, display information for a specific washcard number,

Print records, buying products and adding amount to his washcard.

1. **User Card Details:**

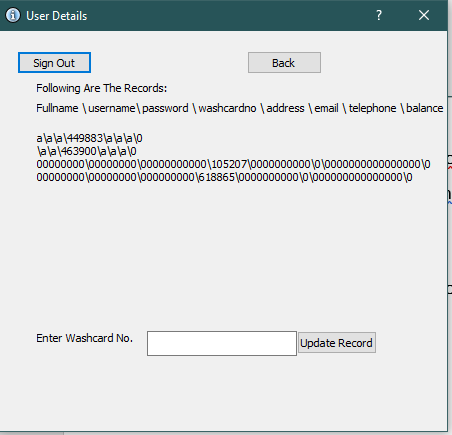


Employee can view the list of all washcards in the system.

It display information such as username , washcard number, as well as the amount present on the washcard.

Employee can click on signout to sign out of his account or click go back to go back to Employee Main screen.

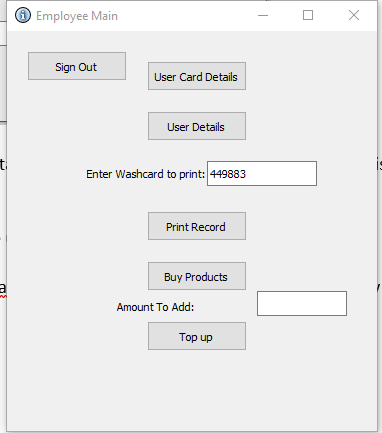
1. **User Details:**



Employee clicks user details button on employee main screen to get to this screen.   
  
He can then view all the users saved in the record.txt file.

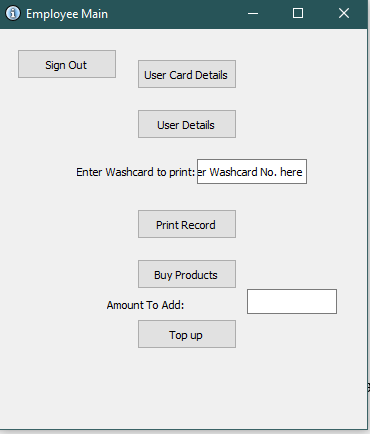
He can also enter the washcard number of any specific user to update any detail of that user.

1. **Print info:**

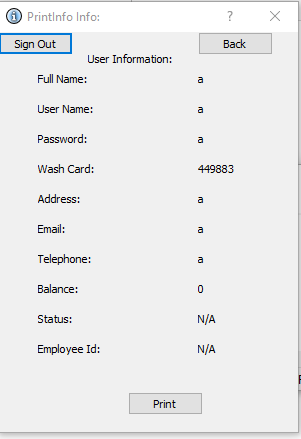


Employee has to enter the washcard number of any user whose information he wishes to see, and then click print record.

If no washcard number is provided a message appears in the text label that a washcard number should be provided :



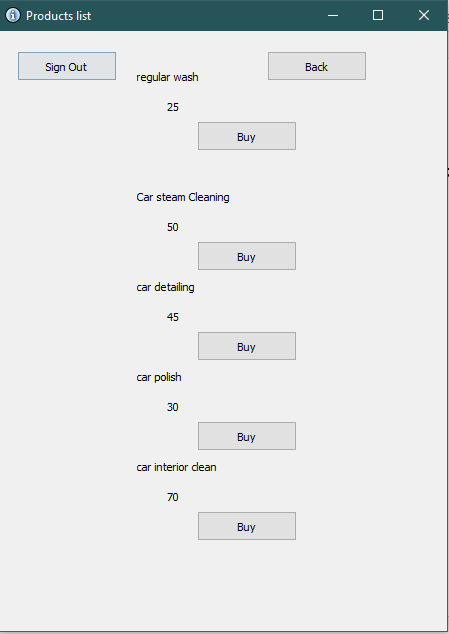
After providing washcard number, employee clicks print record to view the details of the specific user.



This now displays the information for the user whose washcard number was provided on the previous screen.

Click back to go to previous screen.

1. **Employee buy products:**



Same as user buy products, but, the price of each product is discounted by 30 %.

Same as user, if the employee clicks buy button, it first checks if employee has enough amount in washcard, if yes then product is bought and if no then message printed that not enough amount.

# **USER TEST DOCUMENT:**

## **Login**

|  |  |  |
| --- | --- | --- |
| Test Case ID | TC-1.1 | |
| Test Case Name | User Login | |
| Test Data | Username: 00000000  Password: 00000000000 | |
| Pre-Condition | User must be registered to the system. | |
| Actions | | System Response |
| 1. User clicks on the sign in button. 2. User enters his/her username and password. 3. User clicks signin. | | 1. The System opens the isgnin form. 2. System validates the credentials of the user. 3. The system directs the user to the main screen based on user type. |
| Result | | * Directs the user to the Main screen. * **PASS** |

**Test verification:** This testcase shows that the user is successfully logged in and the user can perform all the available actions.

**Testing Environment:** Python GUI

## **Topup:**

|  |  |  |
| --- | --- | --- |
| Test Case ID | TC-1.2 | |
| Test Case Name | Topup | |
| Test Data | Amount : 200 | |
| Pre-Condition | User must be logged into the system. | |
| Actions | | System Response |
| 1. User enters amount in the textbox. 2. User clicks the Top up button. | | 1. The System shows the entered amount in the textbox. 2. System adds the amount in the records.txt file and displays the results in the command line. |
| Result | | * Amount added in the records.txt file. * **PASS** |

**Test verification:** This testcase shows that the user is successfully able to add amount to their washcard.

**Testing Environment:** Python GUI

## **View Information:**

|  |  |  |
| --- | --- | --- |
| Test Case ID | TC-1.3 | |
| Test Case Name | View information | |
| Test Data |  | |
| Pre-Condition | User must be logged into the system. | |
| Actions | | System Response |
| 1. User clicks the view information button on the main menu. 2. User views the information provided on the window. | | 1. The System opens up a new window. 2. System retrieves the information from the records.txt file and displays it in the window. |
| Result | | * User Information displayed in the new window. * **PASS** |

**Test verification:** This testcase shows that the user is successfully able to view his personal info.

**Testing Environment:** Python GUI

## **View Information:**

|  |  |  |
| --- | --- | --- |
| Test Case ID | TC-1.4 | |
| Test Case Name | View information | |
| Test Data |  | |
| Pre-Condition | User must be logged into the system. | |
| Actions | | System Response |
| 1. User clicks the view information button on the main menu. 2. User views the information provided on the window. | | 1. The System opens up a new window. 2. System retrieves the information from the records.txt file and displays it in the window. |
| Result | | * User Information displayed in the new window. * **PASS** |

**Test verification:** This testcase shows that the user is successfully able to view his personal info.

**Testing Environment:** Python GUI

## **Update Information:**

|  |  |  |
| --- | --- | --- |
| Test Case ID | TC-1.5 | |
| Test Case Name | Update information | |
| Test Data | Fullname , user name, password , wash card, Address , Email , telephone | |
| Pre-Condition | User must be logged into the system. | |
| Actions | | System Response |
| 1. User clicks the view information button on the main menu. 2. User then clicks the update button at the bottom of the screen. 3. User enters the values in the specified text boxes. 4. User clicks update button. | | 1. The System opens up a new window. 2. System retrieves the information from the records.txt file and displays it in the text boxes in the update information window. 3. Program display the user’s entered values in the text boxes. 4. Program updates the values in the records.txt file and goes to view information screen. |
| Result | | * User Information displayed in the new window. * **PASS** |

**Test verification:** This testcase shows that the user is successfully able to Update his personal info.

**Testing Environment:** Python GUI

## **Buy Product:**

|  |  |  |
| --- | --- | --- |
| Test Case ID | TC-1.6 | |
| Test Case Name | View information | |
| Test Data |  | |
| Pre-Condition | User must be logged into the system. | |
| Actions | | System Response |
| 1. User clicks on Buy Product Button 2. User clicks on buy | | 1. System opens the product view 2. System checks the validity of the request |
| Result | | * Amount deducted from account * **PASS** |

**Test verification:** This testcase shows that the user is successfully able to buy the product

**Testing Environment:** Python GUI