Stock and Sales Management

1. Introduction

A sales and inventory system is a software-based business solution used to simultaneously track sales activity and inventory. Manufacturers and trade resellers can both benefit from a single thorough solution, where single transaction entry records necessary details on the customer, products purchased, price and date while also updating inventory levels.

Using computerized sales and inventory system allows for much greater accuracy in stocking and product management. They encourage ease of interaction between employees and shoppers as transactions are processed and items move from business to the consumer. Computerized sales help provide better insight into which products are most popular. It allows for enhanced marketing, stocking and oversight of critical sales objective.

1. Objectives

The objectives of this system are:

1. To help the management in tracking the sales and stocks of the restaurant
2. To implement the stock and sales management system to the best satisfaction of the customer
3. Use IDE Python and database MySQLite to facilitate this process

The normal process of this system is to fill the data and then process the data which used to cause a lot of inconvenience to both the administrator and customers. In using databases, the inputs of the stocks and sales will be facilitated. The stock and sales management system can be presented in views of the different involved with it. These are the administrators.

1. Functions
2. Login Page

* The user will login with its own username and password

1. Admin Window

* The user will be able to choose which functions he will use namely: Stock Entry and Sales Entry
* The user can see the overall summary of the stock and sales by clicking the summary button.

1. Kitchen Window

* Allows the user or admin to input kitchen item
* Allows the user to input the designated price of the item
* Allows the user to input the number of items currently in the kitchen
* Allows the user to delete the item entered if items are sold out.

1. Cold Drink Window

* Allows the user or admin to input cold drinks item
* Allows the user to input the designated price of the item
* Allows the user to input the number of items currently in the cold drinks
* Allows the user to delete the item entered if items are sold out.

1. Hard Drink Window

* Allows the user or admin to input hard drinks item
* Allows the user to input the designated price of the item
* Allows the user to input the number of items currently in the hard drinks
* Allows the user to delete the item entered if items are sold out.

1. Database (MySQLite Browser)

* The information or data is stored in SQLite Browser. Every details is seen in the database.

1. Architectural Design

The architecture of Stock and Sales Management is based on the three-tier architecture. This mainly consists of three layers namely:

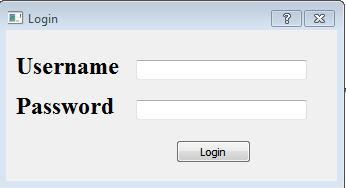
* Presentation Tier
* Business Tier
* Data Access Tier

A picture containing object

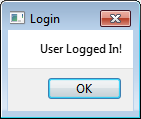
Description generated with very high confidence

The presentation layer is the topmost layer of the application and it is responsible for the user interface of the application which deals with the presentation of data to the user. The business logic layer is the middle layer of the application and the business logic for the Stock and Sales management application would be present here. It is responsible for the information exchange between the user interface and the database. The database layer is the final layer of the application. All the data related to the system is stored and retrieved from her

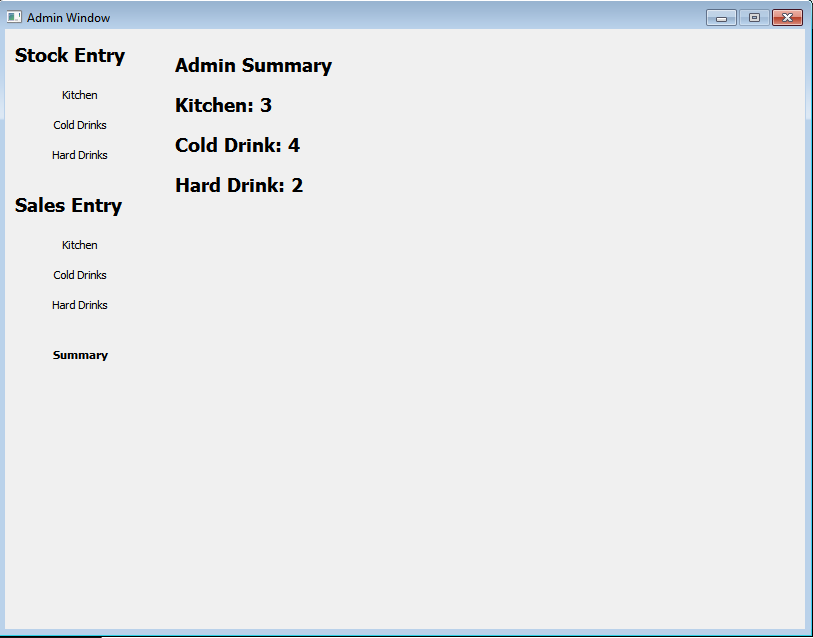
1. Graphical User Interface
2. Login Window



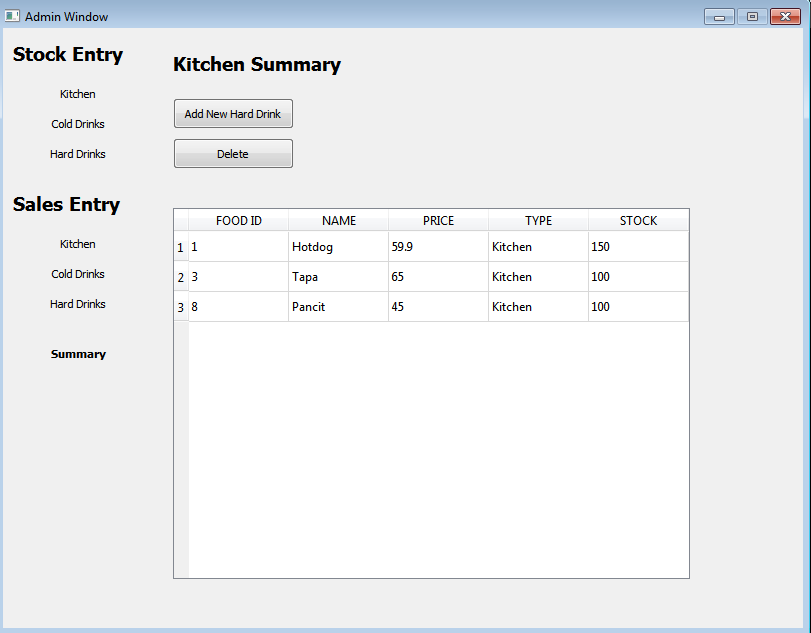
1. Login Confirm

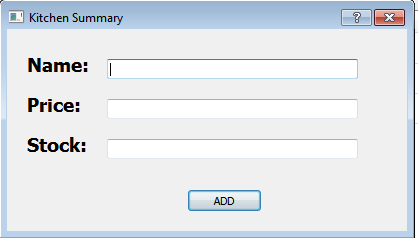


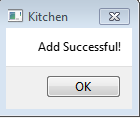
1. Main Window



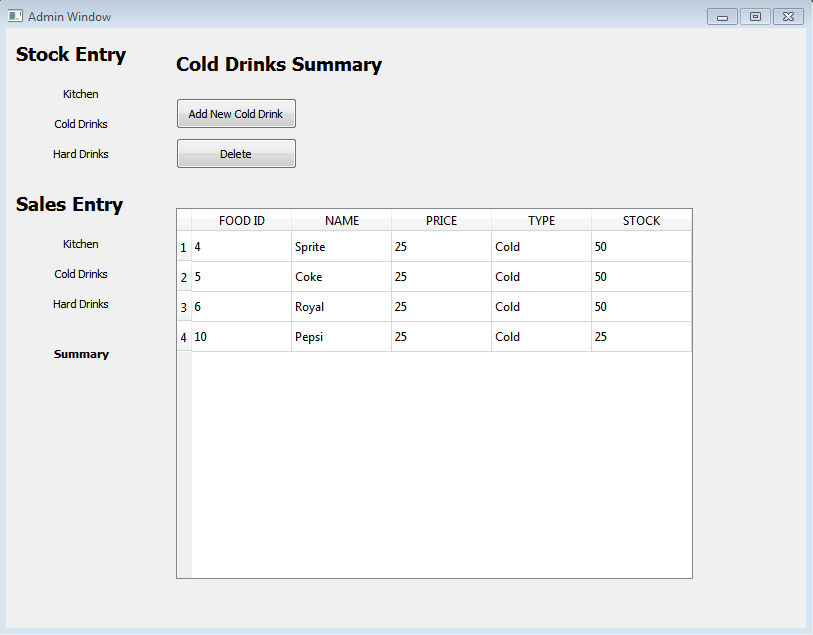
1. Kitchen Summary Window



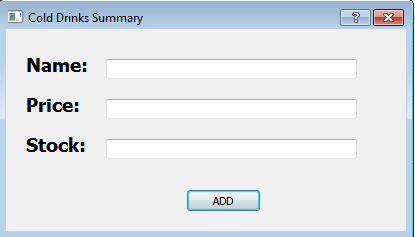
1. Kitchen Add New Kitchen Item Window
2. Kitchen Add confirm



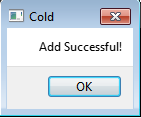
1. Cold Drink Window



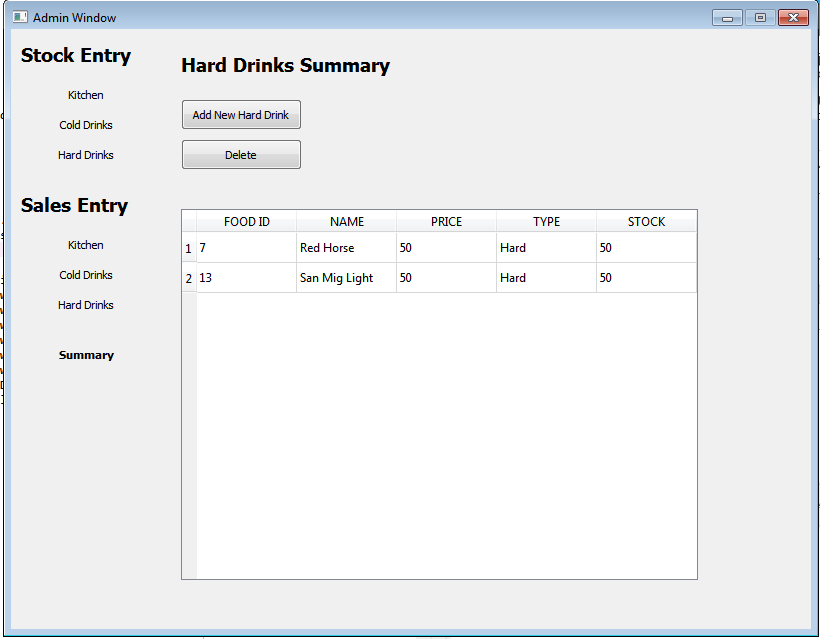
1. Cold Drink Add Window

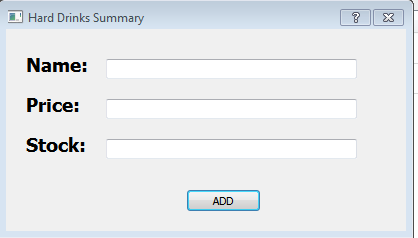


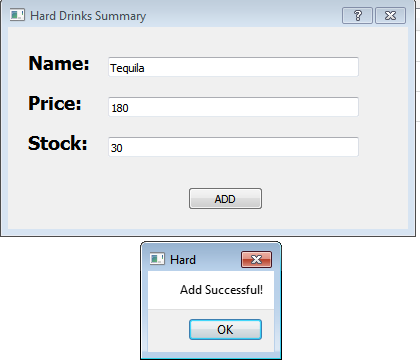
1. Cold Drink Add Confirm



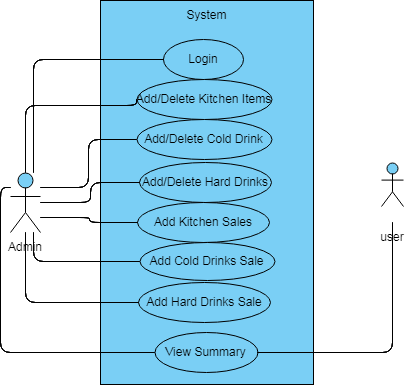
1. Hard Drink Window



1. Hard Drink Add Window
2. Hard Drink Confirm



1. Use Case Diagram



For the Use Case Diagram of the system, the whole system is seen. It contains use cases and actors outside the box. The system name is Stock and Sales Management. As you can see, the ovals represent the system’s function. These functions are login, adding or deleting kitchen items, adding or deleting cold drinks, adding or deleting hard drinks, adding kitchen sales, adding cold drink sales, adding hard drink sales. The actor of the system is the admin. The admin is the one who uses all the functions except for finding the matching flight. Search engine is the one responsible to find a matching flight. The user or customer will only be able to view the summary as it contains sensitive information

1. Use Case Specification

|  |  |
| --- | --- |
| **Use Case Name** | Customer Login Page |
| **Description** | Allows the user to login and access their accounts |
| **Actor** | Customer/Passenger |
| **Pre-Condition** | None |
| **Main Sequence** | 1. User enters “username” and “password” 2. Click “Login” 3. Checks missing or wrong username or password 4. Login |
| **Alternative Sequence** | System prompts the user that the information is missing or incorrect |
| **Post Condition** | The user enters the book a flight page. |

|  |  |
| --- | --- |
| **Use Case Name** | Admin Login Page |
| **Description** | Allows the admin to login and access the account |
| **Actor** | Administrator |
| **Pre-Condition** | Only the admins are required to login |
| **Main Sequence** | 1. Admin enters “username” and “password” 2. Click “Login” 3. Checks missing or wrong username or password 4. Login |
| **Alternative Sequence** | System prompts the admin that the information is missing or incorrect |
| **Post Condition** | The admin enters the managing flights, managing passengers, and managing users. |

|  |  |
| --- | --- |
| **Use Case Name** | MainWindow |
| **Description** | Allows the user to select a task |
| **Actor** | Admin |
| **Pre-Condition** | None |
| **Main Sequence** | 1. Will click to the tasks to use |
| **Alternative Sequence** | None |
| **Post Condition** | The user will be directed to corresponding task page |

|  |  |
| --- | --- |
| **Use Case Name** | Kitchen Summary |
| **Description** | Allows the admin to add or delete kitchen items |
| **Actor** | Administrator |
| **Pre-Condition** | The admin must login the admin account |
| **Main Sequence** | 1. Click the task to be performed 2. Redirects to the assigned task. |
| **Alternative Sequence** | None |
| **Post Condition** | The user will be directed to corresponding task page |

|  |  |
| --- | --- |
| **Use Case Name** | Add New Kitchen Item |
| **Description** | Allows the admin to add new kitchen items |
| **Actor** | Administrator |
| **Pre-Condition** | The admin must login the admin account |
| **Main Sequence** | 1. Enter the name of the item 2. Enter the price of the item 3. Enter the number /stock of the item |
| **Alternative Sequence** | None |
| **Post Condition** | The item is added in the database. |

|  |  |
| --- | --- |
| **Use Case Name** | Delete Kitchen Item |
| **Description** | Allows the admin to delete kitchen items from the database |
| **Actor** | Administrator |
| **Pre-Condition** | The admin must login the admin account |
| **Main Sequence** | 1. Select the item to be deleted 2. Click the delete button |
| **Alternative Sequence** | None |
| **Post Condition** | The item is deleted in the database. |

|  |  |
| --- | --- |
| **Use Case Name** | Cold Drinks Summary |
| **Description** | Allows the admin to add or delete cold drinks |
| **Actor** | Administrator |
| **Pre-Condition** | The user must login the user account |
| **Main Sequence** | 1.Click the task to be performed  2.Redirects to the assigned task. |
| **Alternative Sequence** | None |
| **Post Condition** | The user will be directed to corresponding task page |

|  |  |
| --- | --- |
| **Use Case Name** | Add New Cold Drink Item |
| **Description** | Allows the admin to add new Cold Drink items |
| **Actor** | Administrator |
| **Pre-Condition** | The admin must login the admin account |
| **Main Sequence** | 1. Enter the name of the item 2. Enter the price of the item 3. Enter the number /stock of the item |
| **Alternative Sequence** | None |
| **Post Condition** | The item is added in the database. |

|  |  |
| --- | --- |
| **Use Case Name** | Delete Cold Drink Item |
| **Description** | Allows the admin to delete cold drinks items from the database |
| **Actor** | Administrator |
| **Pre-Condition** | The admin must login the admin account |
| **Main Sequence** | 1. Select the item to be deleted 2. Click the delete button |
| **Alternative Sequence** | None |
| **Post Condition** | The item is deleted in the database. |

|  |  |
| --- | --- |
| **Use Case Name** | Hard Drink Summary |
| **Description** | Allows the admin to add or delete hard drink items |
| **Actor** | Administrator |
| **Pre-Condition** | The admin must login the admin account |
| **Main Sequence** | 1. Click the task to be performed 2. Redirects to the assigned task. |
| **Alternative Sequence** | None |
| **Post Condition** | The user will be directed to corresponding task page |

|  |  |
| --- | --- |
| **Use Case Name** | Add New Hard Drink Item |
| **Description** | Allows the admin to add new hard drink items |
| **Actor** | Administrator |
| **Pre-Condition** | The admin must login the admin account |
| **Main Sequence** | 1. Enter the name of the item 2. Enter the price of the item 3. Enter the number /stock of the item |
| **Alternative Sequence** | None |
| **Post Condition** | The item is added in the database. |

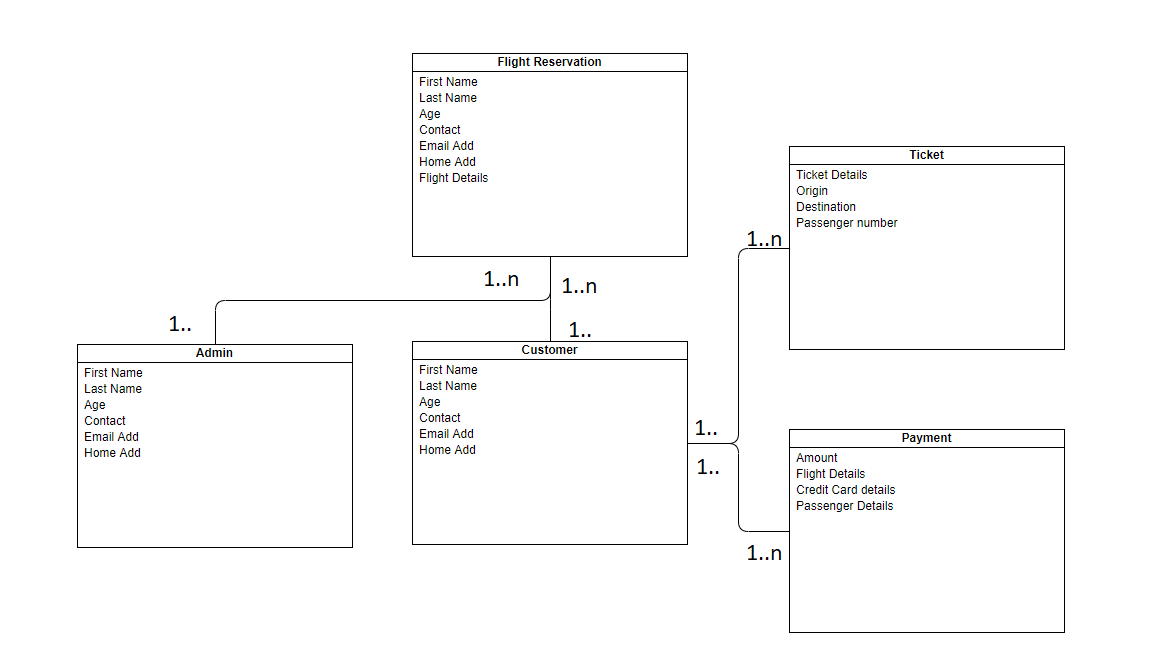
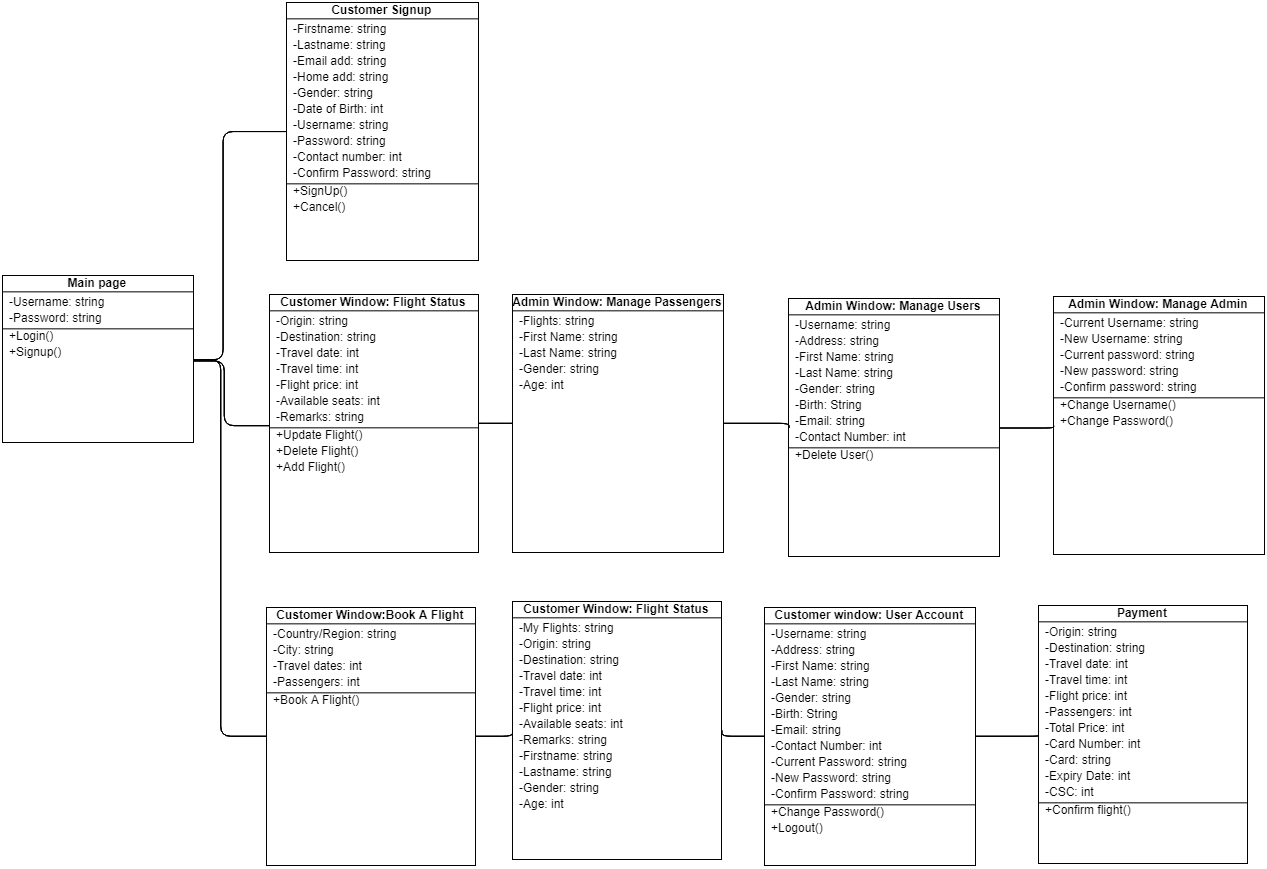
|  |  |
| --- | --- |
| **Use Case Name** | Delete Hard Drink Item |
| **Description** | Allows the admin to delete hard drinks items from the database |
| **Actor** | Administrator |
| **Pre-Condition** | The admin must login the admin account |
| **Main Sequence** | 1. Select the item to be deleted 2. Click the delete button |
| **Alternative Sequence** | None |
| **Post Condition** | The item is deleted in the database. |

|  |  |
| --- | --- |
| **Use Case Name** | Kitchen Sales Entry |
| **Description** | Allows the admin to add kitchen sales |
| **Actor** | Administrator |
| **Pre-Condition** | The admin must login the admin account |
| **Main Sequence** | 1. Click the add button 2. Enter the item 3. Enter the quantity of sales |
| **Alternative Sequence** | None |
| **Post Condition** | The user will be directed to corresponding task page |

|  |  |
| --- | --- |
| **Use Case Name** | Cold Drinks Sales Entry |
| **Description** | Allows the admin to add kitchen sales |
| **Actor** | Administrator |
| **Pre-Condition** | The admin must login the admin account |
| **Main Sequence** | 1. Click the add button 2. Enter the item 3. Enter the quantity of sales |
| **Alternative Sequence** | None |
| **Post Condition** | The user will be directed to corresponding task page |

|  |  |
| --- | --- |
| **Use Case Name** | Hard Drinks Sales Entry |
| **Description** | Allows the admin to add kitchen sales |
| **Actor** | Administrator |
| **Pre-Condition** | The admin must login the admin account |
| **Main Sequence** | 1. Click the add button 2. Enter the item 3. Enter the quantity of sales |
| **Alternative Sequence** | None |
| **Post Condition** | The user will be directed to corresponding task page |

|  |  |
| --- | --- |
| **Use Case Name** | Summary Page |
| **Description** | Allows the user to view the overall summary |
| **Actor** | Customer/Administrator |
| **Pre-Condition** | The user must login the user account |
| **Main Sequence** | 1. Click “Summery” |
| **Alternative Sequence** | None |
| **Post Condition** | The user will be directed to the Summary page |

1. Class Diagram  
     
   Conceptual Class Diagram  
     
   The Figure above is a business level type of representation. The model represents the relationship of each class. [1..] being atleast one and [1..n] being one or more than one. The following attributes and class names are of pseudo types that are near identical to the implementation of the code.   
     
     
     
     
     
     
     
     
     
     
     
     
   Detailed Class Diagram  
   

The figure above is the detailed class diagrams. It shows all the class in a sequence that would most likely happen when the program has began to start. The diagram shows all the attributes and functions that were needed to complete the project.

1. Test Case Diagram

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE #** | **FUNCTION** | **EVENT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** |
| TC # 1 | User Login | Enter the correct username and password then click Login and press Enter. | Redirected to booking a flight page. | OK |
|  |  | Enter a mismatching username and password then click Login then press Enter. | Reload login page, clear username and password fields. | OK |
|  |  | Enter admin as username and password then click Login then press Enter. | Redirect to managing a flight, managing a passenger, and managing a user’s page. | OK |
| TC # 2 | User Registration | Fill up the necessary information needed then click Sign Up. | Redirect to Login Page. | OK |
|  |  | Incorrect input or lack of input to the required fields. | Shows the message box that it is required to fill up all the required fields. | OK |
| TC # 3 | Managing Flights | Select the desired date and choose the desired passenger in the drop-down list. Click Delete Flight. | The flight of the selected passenger is now deleted in the database. | OK |
|  |  | Select the desired date and choose the desired passenger in the drop-down list. Put Remarks then click Update. | The flight of the selected passenger is now updated in the database. | OK |
| TC # 4 | Managing Passengers | Select the desired date and choose the desired passenger in the drop-down list. Click delete passenger. | The chosen passenger is now deleted in the database. | OKS |
| TC # 5 | Managing Users | Select a username. | The information like the name, email, and address are shown in the screen. | OK |
|  |  | Select a username then click Delete User | The user is now deleted in the database. | OK |
| TC # 6 | Booking a Flight | Select the source or destination, travel dates, and passengers. | The details are displayed on the program. | OK |
| TC # 7 | Flight Status |  |  |  |
| TC # 8 | Payment Method | Select the kind of method when paying. | The system then tells the confirmation. | OK |
| TC # 9 | Logout Page | Click Logout | The user or admin is directed to the login page. | OK |

1. Errors Encountered

|  |  |
| --- | --- |
| Date | Error |
| 9/20/18 | * Failed to upload the document (Gantt Chart) under the Documentation folder in the Github. The file was already reuploaded inside the folder. |
| 9/21/18 | * The title of our project did not match with the system name in the use case diagram. |
| 9/22/18 | * At first, we failed to download the pyqt5 on the path environment because the command prompt is not in the administrator. |
| 9/23/18 | * No errors |
| 9/24/18 | * No errors |
| 9/25/18 | * At first, we were unable to convert the ui to python because of the wrong command. |
| 9/26/18 | * No errors |
| 9/27/18 | * No errors |
| 9/28/18 | * No errors |
| 9/29/18 | * No errors |
| 9/30/18 | * No errors |
| 10/1/18 | * No errors |
| 10/2/18 | * No errors |
| 10/3/18 | * No errors |
| 10/4/18 | * This is the start of Spring3. We were unable to assign correctly the task to each of us. |
| 10/5/18 | * No errors |
| 10/6/18 | * No errors |
| 10/7/18 | * No errors |
| 10/8/18 | * No errors |
| 10/9/18 | * The group unable to finish the tasks at that moment. |
| 10/10/18 | * No errors |
| 10/11/18 | * The Cancel button is not yet working. |
| 10/12/18 | * No errors |

1. Recommendation
2. Conclusion  
   For the conclusion of this project, We can say that it was a success. Having a finished program up and running, the group achieved the goal to create a Flight reservation system using python environment. The group managed to implement the airline reservation system to the best satisfaction of the customer according to what the group can offer. And also the group managed to use database MySQlite to facilitate this process.