**Use Case 1: SW\_Staff\_Detail**

**Overview:**

This use case allow staffs or administrators to check their information during staff or administrator login.

**Precondition:**

1. The database is accessible
2. Staff\_Mode\_View is displayed.

**Scenario:**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1.Staff clicks on the Staff Detail tab on the Staff\_Mode\_View | 1. Staff\_Detail\_View tab appears. |
| 2.Staff clicks on Logout button on the Staff\_Mode\_View | 2. The Staff\_Mode\_View is destroyed,and Operator is returned to Login\_View. |

**Scenario Notes:**

Item 2 is not mandatory unless the staff want to return to the Login\_View.

Post Conditions:

1. The staff is returned to the Login\_View (if Logout button is selected).

Exceptions: Use cases Utilized:

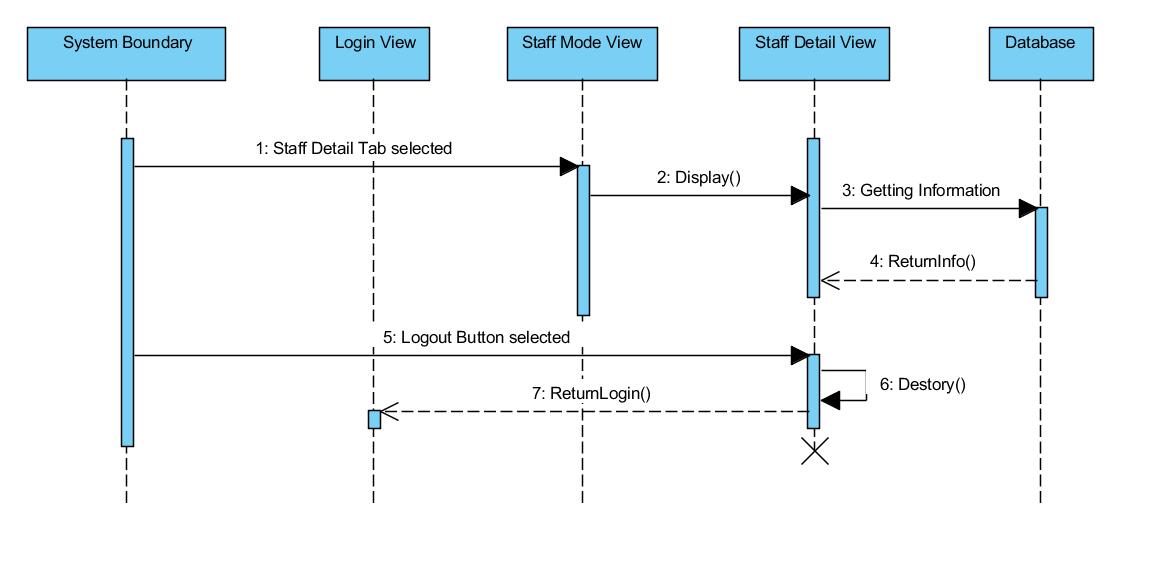
1. The database cannot be accessed. None

Required GUI: Timing Constraints:

1. Login\_View None

2. Staff\_Mode\_View

UC1 Staff Detail



1. Staff selects Staff Detail Tab

2. Display Staff\_Detail\_View

3. Getting Information from database

4. Return information from to Staff\_Detail\_View

5. If Logout Button selected then,

6. Destroy the Staff\_Detail\_View

7. Return to Login\_View

**Use Case 2: SW\_Staff\_Flight\_Info**

**Overview:**

This use case allows staffs or administrators to check flight information base on customer’s request.

**Precondition:**

1. The database is accessible
2. Staff\_Mode\_View is displayed.

**Scenario:**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1.staff clicks on the Flight info tab on the Staff\_Mode\_View | 1. Flight\_View tab appears. |
| 2.Enter Departure Date | 2.Departure date field is updated |
| 3.select Airline | 3. Airline combo box is updated. |
| 4.select type of Trip | 4. Trip field is updated. |
| 5.select departure location | 5. Departure location combo box is updated. |
| 6.select destination | 6. Destination combo box is updated. |
| 7.select region | 7. Region field is updated. |
| 8.Staff clicks on Search button on the Flight\_Info\_View | 8. The table is refresh base on the updated information. |
| 9.Staff clicks on Logout button on the Staff\_Mode\_View | 9. The Staff\_Mode\_View is destroyed. The database is updated, and Operator is returned to Login\_View. |

**Scenario Notes:**

Item 2 to 7 can be performed in any order but mandatory. Item 8 has to be selected for the purpose of refreshing the table. Item 9 is not mandatory unless staffs want to return to Login\_View.

Post Conditions:

1. Previous search from the staff is saved on the page if the view is not destroyed.

2. The staff is returned to the Login\_View (if Logout\_Button is selected).

Exceptions: Use cases Utilized:

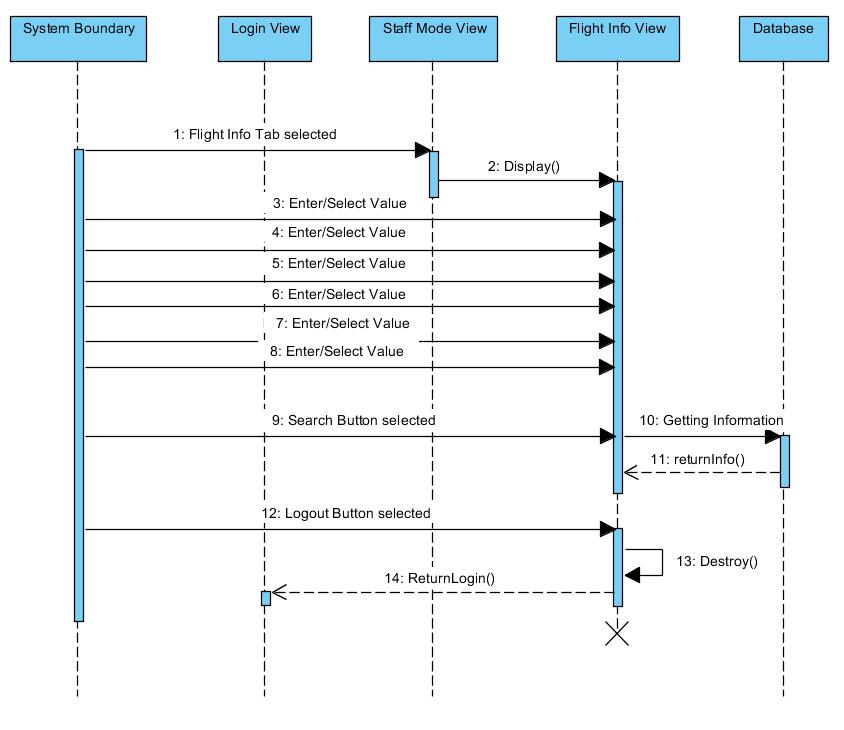
1. The database cannot be accessed. None

Required GUI: Timing Constraints:

1. Login\_View None

2. Staff\_Mode\_View

Uc2:



1. Staff selects Flight Info tab

2. Display Flight\_Info\_View

3-8. Enter or select value

9. Staff selects Search Button

10. Getting information from database

11. Return information to Flight\_Info\_View

12. If Logout Button is selected

13. Destroy Flight\_Info\_View

14. Return to Login\_View

**Use Case 3: SW\_Staff\_Customer\_Info**

**Overview:**

This use case allows staffs or administrators to check clients or customers information base on his/her full name and date of birth.

**Precondition:**

1. The database is accessible
2. Staff\_Mode\_View is displayed.

**Scenario:**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1.staff clicks on the Customer info tab on the Staff\_Mode\_View | 1. Customer\_Info\_View tab appears. |
| 2. Enter Customer's Last name | 2. Customer Last Name field is updated. |
| 3. Enter Customer’s date of birth. | 3. Customer Date of Birth field is updated. |
| 4. Staff clicks on Search Button on the Customer\_Info\_View. | 4. The Cusmtomer’s information is refreshed base on the Customer Name and Date of Birth fields. |
| 5.Staff clicks on Booking History button on the customer\_Info\_View | 5. Booking\_History\_View pop-up appears. |
| 6. Staff clicks on Print button. | 6. The software is connected to computer’s default printer and ask for permission. |
| 7. Staff clicks on Cancel button. | 7. the Booking\_History\_View pop-up is destroyed, and return to Customer\_Info\_View tab. |
| 8.Staff clicks on Logout button on the Staff\_Mode\_View | 8. The Staff\_Mode\_View is destroyed. The database is updated, and Operator is returned to Login\_View. |

**Scenario Notes:**

Item 2 and 3 may perform in any order, but they are mandatory for Staffs in order to perform item 5. Item 7 is mandatory for staffs to exit the Booking\_History\_View. Item 8 is not mandatory unless staffs want to return to Login\_View.

Post Conditions:

1. Booking history is printed (if Print Button is selected).

2. The staff is returned to the Login\_View(if Logout button is selected).

3. Customer information is displayed (if Search Button is selected).

Exceptions: Use cases Utilized:

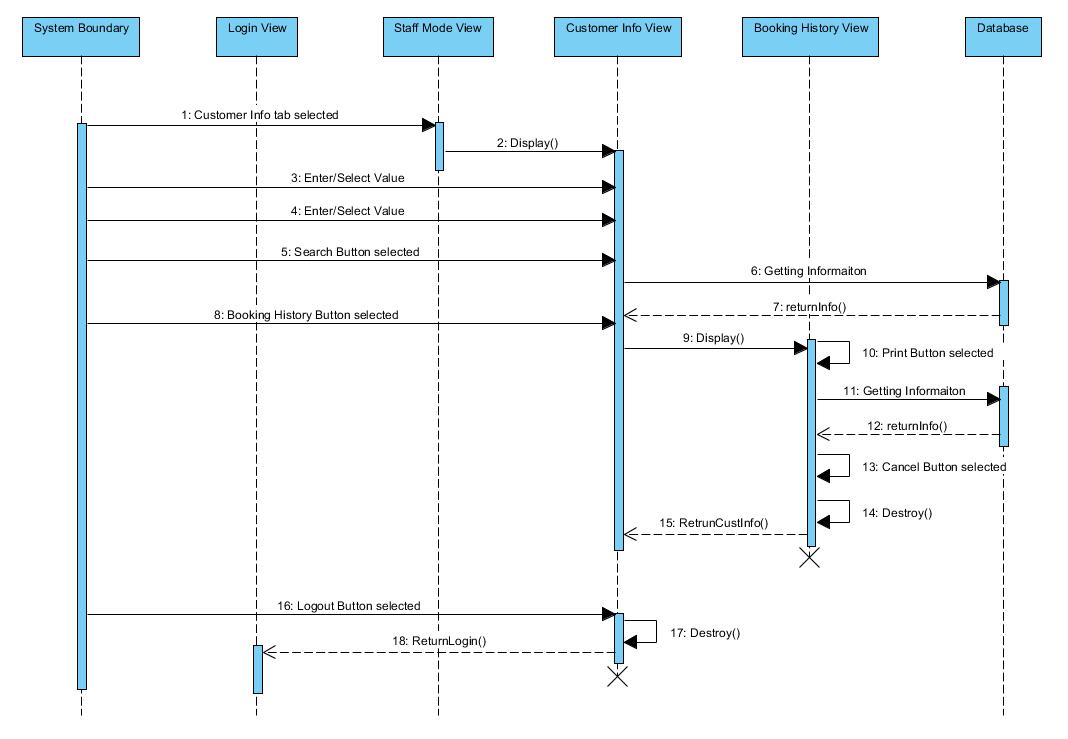
1. The database cannot be accessed. None

Required GUI: Timing Constraints:

1. Login\_View None

2. Staff\_Mode\_View

5. Customer\_Info\_view



UC3 Staff checks Customer’s information

1. Staff selects Customer Info tab

2. Display Customer\_Info\_View

3-4. Enter or select Value

5. Staff selects Search Button

6. Getting information from database

7. Return information to Customer\_Info\_View

8. If Booking History Button is selected

9. Display Booking\_History\_View

10. If Print Button is selected,then

11. Getting information from database

12. Return information to Booking\_History\_View

13. If Cancel Button is selected,then

14. Destroy Booking\_History\_View

15. Return to Customer\_Info\_View

16. If Logout Button is selected

17. Destroy Customer\_Info\_View

18. Return to Login\_View

\*NOTE: step 3 is referring to Customers’ name and 4 is Date of birth.

**Use Case 4: Staff\_Changes**

**Overview:**

This use case allows staffs or administrators to change based on customer's request of incorrect personal information.

**Precondition:**

1. The database is accessible
2. Staff\_Mode\_View is displayed.

**Scenario:**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1.staff clicks on the Changes tab on the Staff\_Mode\_View | 1. Changes\_View tab appears. |
| 2.enter Customer's name | 2. Customer Name field is updated. |
| 3. Enter Customer’s date of birth. | 3. Customer Date of Birth field is updated. |
| 4.Staff clicks on Search Button on the Changes\_View | 4. The Cusmtomer’s information is refreshed base on the Customer Name and Date of Birth fields. |
| 5. Edit Name | 5. Name field is updated. |
| 6, Edit Gender | 6, Gender field is updated |
| 7. Edit phone number. | 7. Phone number field is updated. |
| 8. Edit E-mail. | 8. E-mail field is updated. |
| 9. Edit address. | 9. Address field is updated. |
| 10.staff clicks on submit Button | 10. The Confirmation pop-up is created. |
| 11, staff clicks on Search History Button | 11, the Searching History of a customer pops up |
| 12. Staff clicks on Yes button. | 12. The Confirmaton pop-up is destroyed. The database is updated. And staff is returned to Changes\_View tab. |
| 13. Staff clicks on Cancel Button. | 13. The Confirmation pop-up is destroyed.The database is not updated. |
| 14.Staff clicks on Logout button on the Staff\_Mode\_View | 14. The Staff\_Mode\_View is destroyed. The database is updated, and Operator is returned to Login\_View. |

**Scenario Notes:**

Item 2 and 3 may be done in any order, but they are mandatory for item 4. Item 5 to 8 may be done in any order and staff may choose to update any of them. Item 9 may be performed before or after Item 7 to 10 changed. Item 10 and 11 are mutually exclusive. Item 10 is not mandatory unless staff want to return to Login\_View.

Post Conditions:

1. Name is updated in the database (if YES button was selected).
2. Phone number is updated in the database (if YES button was selected).
3. E-mail is updated in the database (if YES button was selected).
4. Address is updated in the database (if YES button was selected).

5. The staff is returned to the Login\_View (if Logout Button is selected).

Exceptions: Use cases Utilized:

1. The database cannot be accessed. None

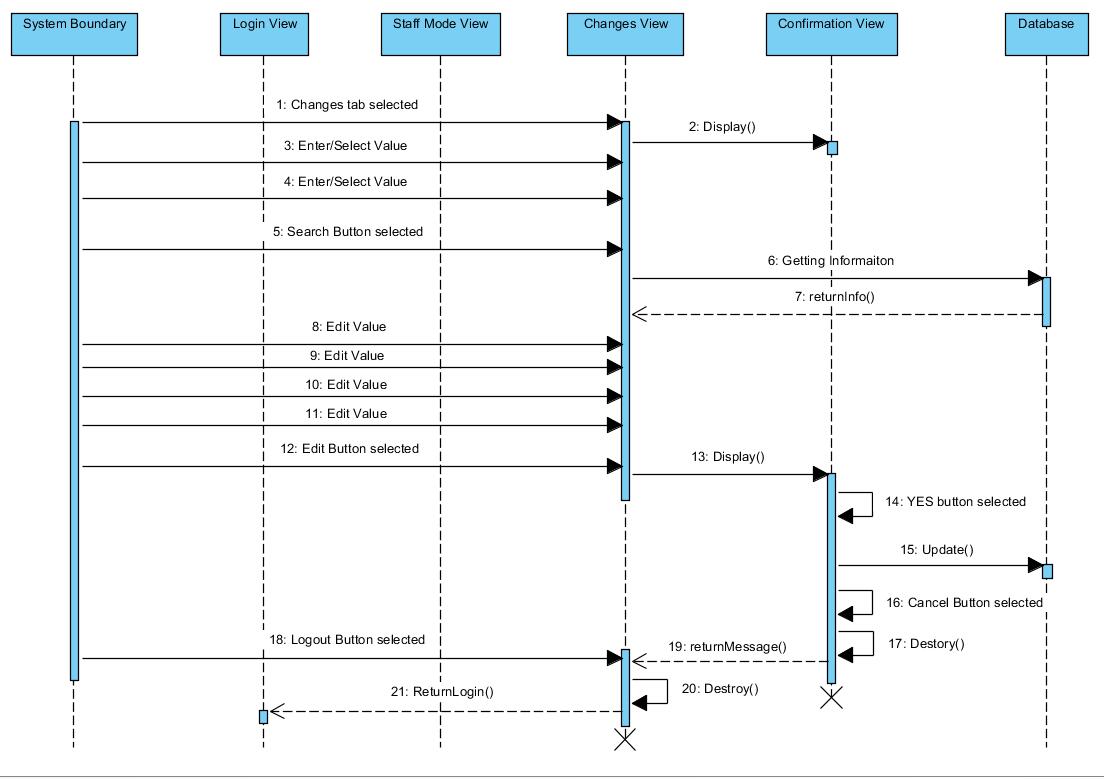
Required GUI: Timing Constraints:

1. Login\_View None

2. Staff\_Mode\_View

6. Changes\_View

UC 4:

UC4 Staff changes customers’ information

1. Staff selects Changes tab

2. Display Changes\_View

3-4. Enter or select Value

5. Search Button is selected

6. Getting information from database

7. Return information to Changes\_View

8-11. Edit Values

12. If Edit Button is selected, then

13. Display Confirmation\_View

14. If YES Button is selected,then

15. Update the database

16. If Cancel Button is selected, then

17. Destroy Confirmaton\_View

18. Return to Changes\_View

19. If Logout Button is selected,then

20. Destroy Changes\_View

21. Return to Login\_View

\*NOTE: step 3 is referring to Customers’ name, and 4 is Date of birth. Step 8 is referring to Name, 9 is phone number, 10 is E-mail, and 11 is address.

Used case 6: SW Customer Login

Overview:

This Use Case allow customers to login, and the customer will define them as guest or customer with an account. The customer will see the main frame which include searching flight, redeem reward point, my trips, recommend friends, and account information.

Precondition:

1. The Login\_View is displayed.
2. The database is accessible.

Scenario Notes:

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1.The customer click on login as customer button | 1. The customer is required to enter user name and password. |
| 2.The customer enter user name | 2. The user name field is updated |
| 3.The customer enter password | 3. The password filed is updated with encryption. |
| 4.The customer click on login button | 4. System start proceed login information. |
| 5. System validate username and password | 5. System check username and password is correct or not. |
| 6. System found out username and password is wrong | 7. System send a warning message indicate what is wrong. |
| 7. System found out username and password is correct | 7.The login\_View is destroyed, and the Customer\_View will be displayed. |
| 8.The customer click on login as guest button | 8. The login\_View is destroyed, and the Customer\_View will be displayed. |

Scenario Note:

Item 2 and 3 is mandatory after item 1. Item 1 and 5 are mutually exclusive. Steps 5, 6, and 7 are system actions that interact with database to proceed the user information

Post Conditions:

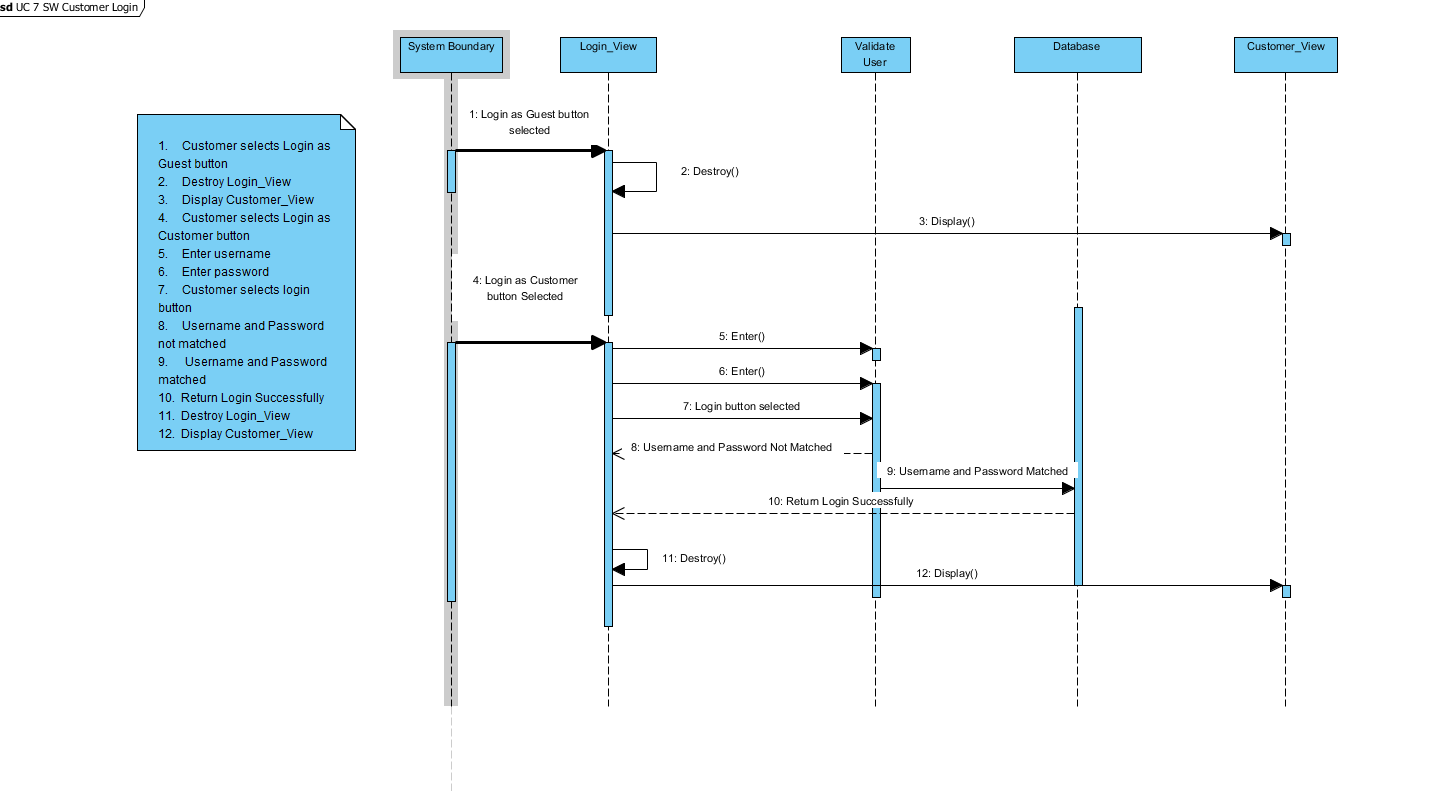
1. The customer is login as guest(if login as guest button is selecter)
2. The customer is login as customer with an account (if login as customerbutton is selecter)
3. The Customer\_View is displayed.

Exception: Use Cases Utilized:

1. The database cannot be accessed. None

Required GUI Timing Constraints:

1. Customer\_View None
2. login\_View



Used Case 8: SW Staff Login

Overview:

This Use Case allow staff to login. The staff will see the main frame which include staff detail, flight info, customer info, changes, and message.

Precondition:

1. The Login\_View is displayed.
2. The database is accessible.

Scenario Notes:

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1.The staff click on login as staff button | 1. The staff is required to enter user name and password. |
| 2.The customer enter user name | 2. The user name field is updated |
| 3.The customer enter password | 3. The password filed is updated with encryption. |
| 4.The staff click on login button | 4. System start proceed login information.. |
| 5. System validate username and password | 5. System check username and password is correct or not. |
| 6. System found out username and password is wrong | 7. System send a warning message indicate what is wrong. |
| 7. System found out username and password is correct | 7.The login\_View is destroyed, and the Staff\_Mode\_View will be displayed. |

Scenario Note:

Item 2 and 3 is mandatory after item 1. Steps 5, 6, and 7 are system actions that interact with database to proceed the user information

Post Conditions:

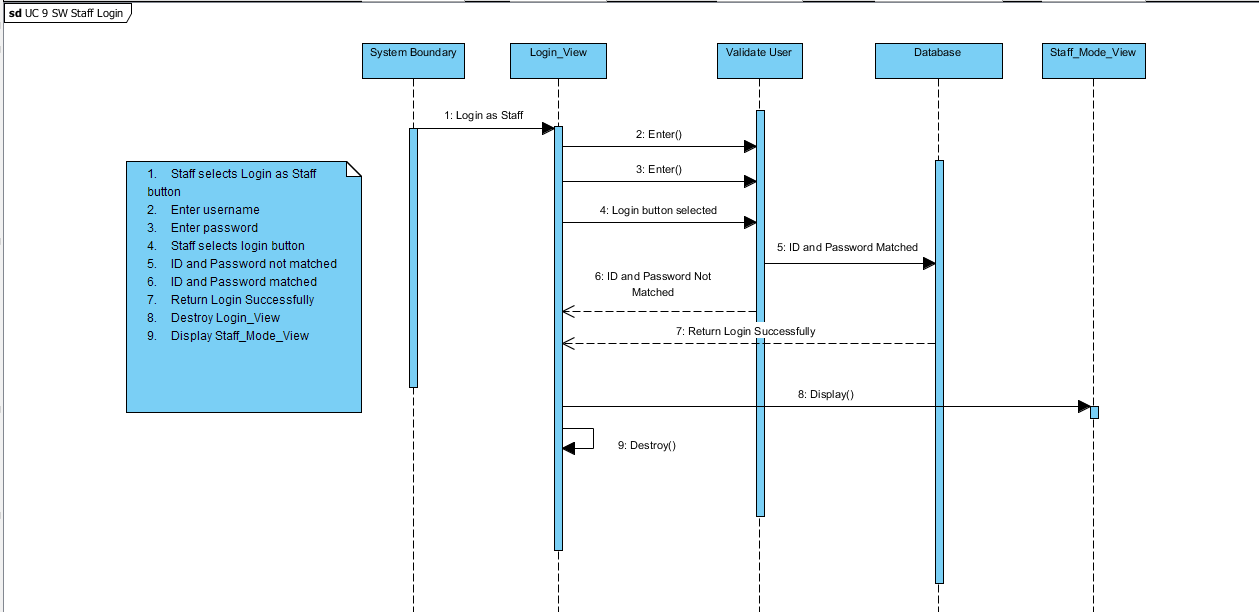
1. The staff is login to staff mode (if login as staff button is selected)
2. The Staff\_Mode\_View is displayed.

Exception: Use Cases Utilized:

1. The database cannot be accessed. None

Required GUI Timing Constraints:

1. Staff\_Mode\_View None
2. login\_View



Used Case 9: SW retrieving passwords

Overview:

This Use Case allow customers to retrieving password when they forget their password.

Precondition:

1. The Login\_View is displayed.
2. The database is accessible.

Scenario Notes:

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. The customer click on forget password button | 1. The Login\_View is destroyed, and the Forget\_Password\_View will be displayed. |
| 2. The customer enter personal email | 2. The personal email field is updated |
| 3 The customer click on send button | 3. System start proceed E-mail. |
| 4. System validate E-mail | 4. System check E-mail is matching to email bound with the account or not. |
| 5. System found out E-mail is not matching | 5. System display a message “the email is not matching”, and ask customer to enter email again. |
| 6. System found out E-mail is matching | 6. System display a text filed ask customer to enter confirmation code with a countdown clock of 90 seconds. |
| 7. The customer enter confirmation code | 7. The confirmation code field is update |
| 8. The customer click on resend button | 8. A text filed will displayed ask customer to enter confirmation code with a new countdown clock of 90 seconds. |
| 9. The customer click on submit button | 9. System start proceed confirmation code. |
| 10. System validate confirmation code | 10. System check confirmation code is correct or not. |
| 11. System found out confirmation code is wrong | 11. customers allow to enter code again in the remaining of time(any time left in 90seconds). |
| 12. System found out confirmation code is correct | 12.The Change\_Password\_View will display and the Forget\_Password\_View is destroyed |
| 13. The customer click on back button | 13. The Forget\_Password\_View is destroyed and the Login\_View is displayed |

Scenario Note:

The customers need to enter correct confirmation code in 90seconds, otherwise the code is expired. Steps 9 and 13 are mutually exclusive. Steps 4,5,6 are system action that interact with database to proceed user E-mail. Steps 10,11,12 are system action that interact with database to proceed confirmation code.

Post Conditions:

1. The customer enter Change\_Password\_View(if submit button is selected)
2. The customer enter Login\_View(if back button is selected)

Exception: Use Cases Utilized:

1. The database cannot be accessed. None

Required GUI Timing Constraints:

1. Login\_View 1. 90 seconds
2. Forget\_Password\_View
3. Change\_Password\_View

**Use Case 10: Operator\_Searches\_For\_Flights\_In\_DB**

**Overview:**

This Use Case enables customers or guests (once they are logged in) to search for flights based on their choice of destinations and days and times.

**Precondition:**

1. The database is accessible.
2. Customer\_View is displayed.

**Scenario:**

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. Operators clicks on login button on login\_view | 1. Login\_view is destroyed |
| 2. Operator clicks on the Round trip tab. | 2. Round trip text box is selected. |
| 3. Operator enters the destination. | 3. Flying to field is updated. |
| 4. Operator enters where he/she is flying from. | 4. Flying from field is updated. |
| 5. Operator enters the departure date. | 5. Departure date field is updated. |
| 6. Operator enters returning date. | 6. Return date field is updated. |
| 7. Operator clicks on the Search button. | 7. The Customer\_View is destroyed, the DB is updated, and the customer is taken to the Booking\_View. |
| 8. Operator clicks on the One-way trip tab. | 8. One-way trip text box is selected. |
| 9. Operator clicks on the sign out button | 9 . Customer\_view is destroyed |

**Scenario Notes:**

After item 1 is performed, items 2, 3, 4, 5, 6, and 7 must be done in exact order. Once operator performs item 8, he/she must perform items 2, 3, 4, 6, and 7 in that exact order. This Use Case allows the modification of all these values. All fields have to be fulfilled by the operator.

**Post Conditions:**

1. All inputs from the Operator is updated in the database (if Search button was selected).

2. The Operator is taken to the Booking page.

3. The Operator is taken to Login\_View

**Exceptions:**  **Use cases Utilized:**

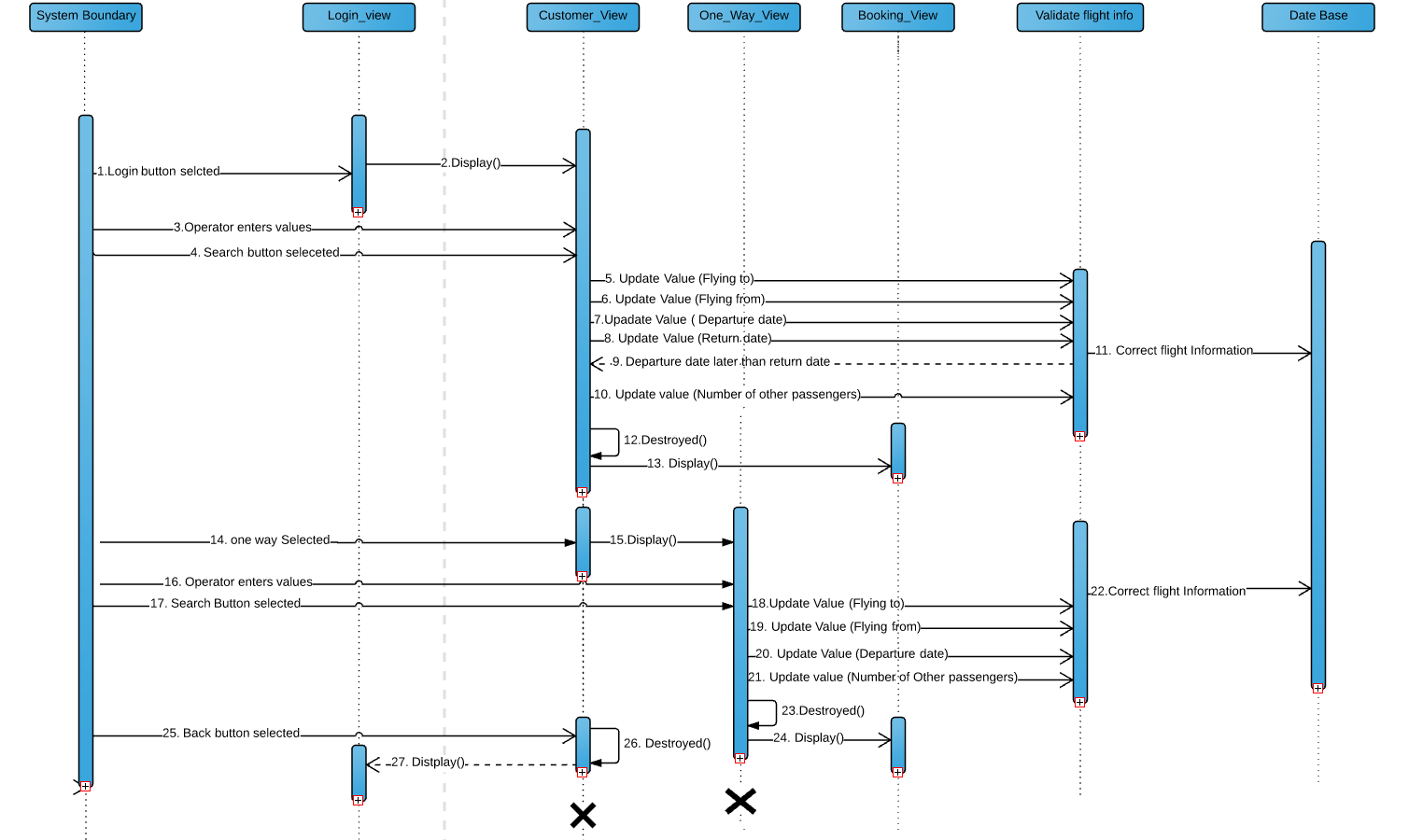
1. The database cannot be accessed. None

**Required GUI:** **Timing Constraints:**

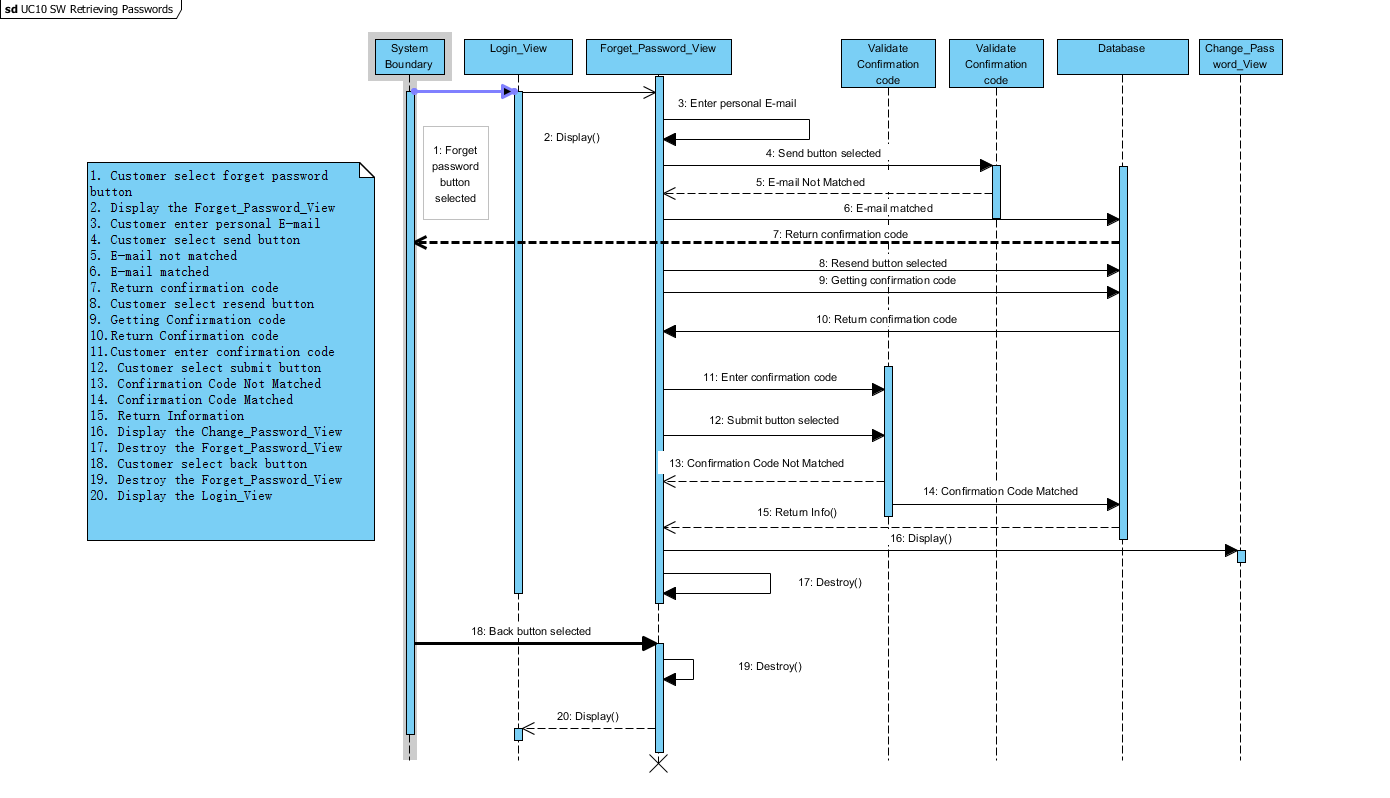
1. Customer\_View 4. Booking\_View None

3.Login\_View 2. One\_way\_View

UC11: Interaction Diagram



|  |
| --- |
| 1.Operator selects Login button  2. Display Customer\_View  3.Operators enters roundtrip flight information  4.If search button is selected then:  5-11. Enter correct roundtrip way information into database  12.Destroy the Customer\_View  13. Display the Booking\_View  14. Else if one-way tab is selected  15. Display One-way\_View  16.Operator enters one-way flight information  17.Search button selected  18-22. Enter correct one-way information into database  23.Destroy One-way\_View  24. Display Booking\_View  25.Else the sign out button is selected  26.Destroy the Customer\_view  27. Display Login\_View |



Used Case 12: SW Customer Checkout

Overview:

This Use Case allow customers to pay their payment for flight tickets. After customers pay their payment they will get a confirmation number for their flights.

Precondition:

1. The Flight\_Info\_View is displayed.
2. Customers have chosen their ticket.
3. The database is accessible.

Scenario:

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1. Customer click on the checkout button | 1. The Payment\_View will pop up. |
| 2. Customer enter card number | 2. The card number field will be updated |
| 3. Customer enter expiration date | 3. The expiration date field will be updated |
| 4. Customer enter CVV | 4. The CVV field will be updated |
| 5. Customer enter Name on card | 5. The Name on card field will be updated |
| 6. Customer enter address | 6. The address field will be updated |
| 7. Customer enter zip code | 7. The zip code field will be updated |
| 8. Customer enter contact phone number | 8. The contact phone number field will be updated |
| 9. Customer enter E-mail address | 9. The email address field will be updated |
| 10. Customer click on submit button | 10. The payment will be proceed. |
| 11. System validate payment information | 11.System check payment information is correct or not. |
| 12. System found out payment information is wrong | 12.System send a warning message will show depend on error that customer having. |
| 13. System found out payment information is correct | 13.System send a confirmation number, and receipt will be shown to customer. |
| 14. Customer click on cancel button | 14. The payment will be cancel. The Payment\_View is destroyed, and the Flight\_Info\_View will pop up. |

Scenario Note:

Item 2-13 may be done in any order. Item 12 is not mandatory, and other items are mandatory. Step 14 and 18 are mutually exclusive. Steps 15, 16, and 17 are system actions that interact with database to proceed the payment information

Post Conditions:

1. The customer pay their payment, a confirmation number, and receipt is given.(If Submit button is selected)
2. The customer is return to Flight\_Info\_View. (If the Cancel button is selected)

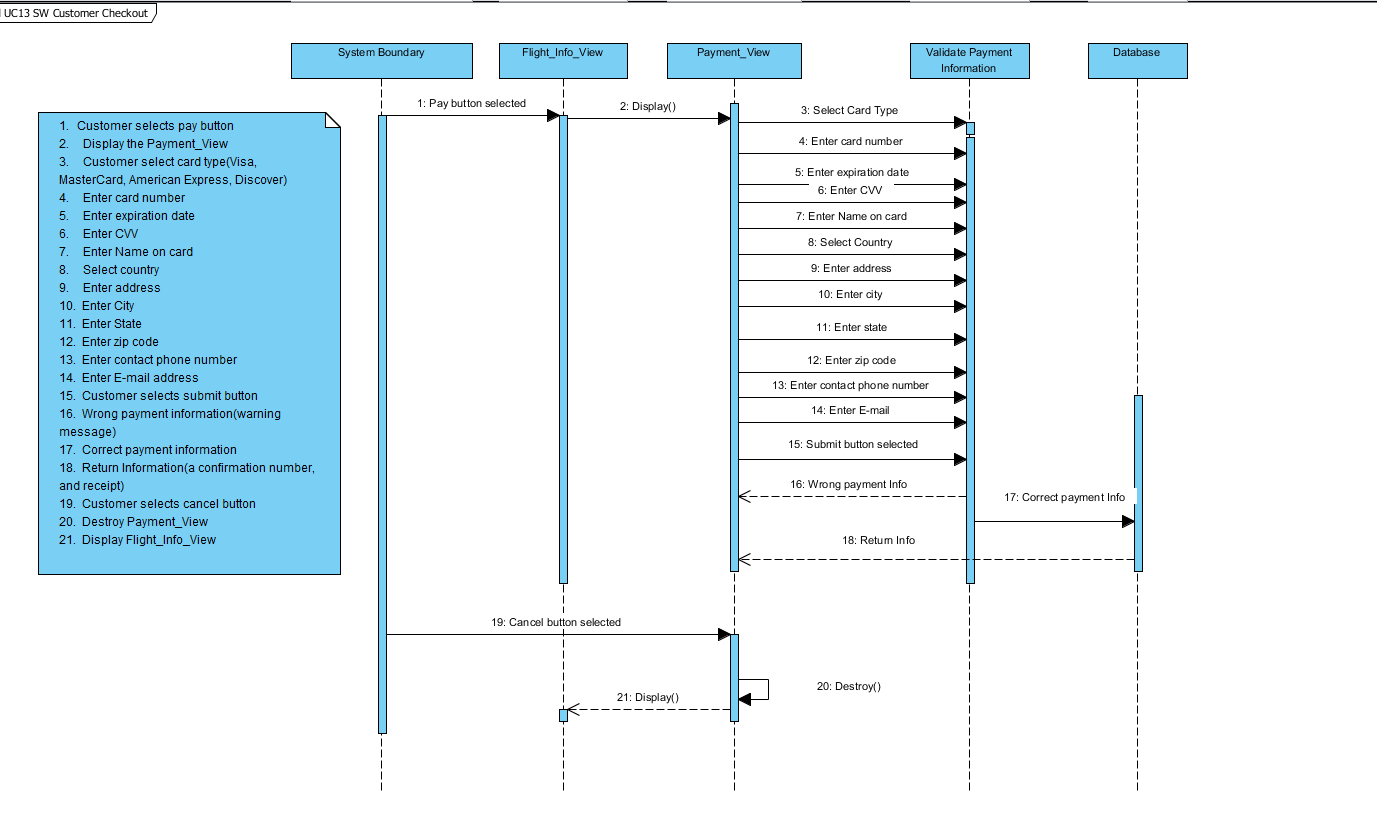
Exception: Use Cases Utilized:

1. The database cannot be accessed. None

Required GUI Timing Constraints:

1. Flight\_Info\_View None

2. Payment\_View



**Use Case 16: Guset Booking**

Overview:

This use case enables guests enter and confirm personal information.

Precondition:

1. Database is accessible
2. Customers have selected flights already
3. Select\_trip\_view is displayed

Scenario:

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1.Customers click **continue** booking button on select\_trip\_view | 1.Jump to guest\_view, save this selected flight trip information in DB. |
| 2.Enter guest **first name** | 2.The first name field will be update |
| 3.Enter guest **last name** | 3.The last name field will be update |
| 4.Enter guest **date of birth** | 4.The date of birth field will be update |
| 5.Click **back** button on guest\_view. | 5. Jump to select\_trip\_view |
| 6.Click **continue** button on guest\_view. | 6.Pop up sign\_up\_prompt\_view |
| 7.Click **register** button on sign\_up\_prompt\_view | 7.Pop up sign\_in\_view |
| 8.Click **close** button on sign\_up\_prompt\_view | 8. Sign\_up\_prompt\_view will disappear, and delete this selected flight trip information in DB. |
| 9.Click **sign in** button on sign\_in\_view | 9. Sign\_in\_view and Sign\_up\_prompt\_view will disappear, and update this flight trip information to DB |
| 10.Click **confirm** button on confirm\_info\_view | 10.Jump to payment\_view |
| 11.Click **back** button on confirm\_info\_view | 11.Confirm\_info\_view popup will disappear |

Scenario Note:

Action 2,3, and 4 may be done in any order. Action 5 & 6, 7 & 8, and 10 & 11 are mutually exclusive.

Post condition:

1. The personal information of guest is updated in DB (if confirm button was selected).
2. Booking history of guest who sign up as a customer is updated in DB (if sign in and confirm button was selected).
3. Flight trip information is deleted for whom did not sign up (if close button was selected).
4. Page is jumped to payment\_view (if confirm button was selected).

Exception: Use Case Utilized:

1.Database cannot be access None

Required GUI: Timing Constraints:

1.Select\_trip\_view 30 minutes   
2.Confirm\_info\_view  
3.Payment\_view   
4.Additional\_passenger\_view

**Use Case:19: change password**

Overview:

This use case enables customers change their account password.

Precondition:

1. Database is accessible
2. Customers click change password button
3. Profile\_view is displayed

Scenario:

|  |  |
| --- | --- |
| Action | Software Reaction |
| 1.Customers click **change password** button on profile\_view | 1.Jump to change\_password\_view. |
| 2.Enter old password | 2.The old password field will be updated |
| 3.Click **continue** button on change\_password\_view | 3.If the password is match with DB, jump to new\_password\_view. Otherwise, error message pops up. |
| 4.Click **cancel** button on change\_password\_view | 4.Return to profile\_view |
| 4.Enter new password | 4.The new password field will be updated |
| 5.Enter confirm new password | 5.The confirm new password field will be updated, error message appears if confirm new password did not match new password |
| 6.Click **back** on new\_password\_view | 6.Return to change\_password\_view |
| 7.Click **change** on new\_password\_view | 7.Success change password message pops up, and return to profile\_view |

Scenario Note:

Action 3 & 4 and 6 & 7 are mutually exclusive.

Post condition:

1. The new password is updated in DB (if change button was selected).
2. The old password is delete in DB (if change button was selected).

Exception: Use Case Utilized:

1.Database cannot be access None

Required GUI: Timing Constraints:

1.Profile\_view 30 minutes   
2.Change\_password\_info\_view  
3.New\_password\_view

Use Case 20: Redeem gifts and coupon.

Overview:

This use case enables customers to redeem gifts and coupon by using their gift points.

Precondition:

1. The database is accessible
2. Customer redeeming\_view is displayed.

Scenario:

|  |  |
| --- | --- |
| **Action** | Software Reaction |
| 1.Operators click on Reward button on the customer\_view. | 1. Redeeming\_view pop-up. |
| 2.Operators check the box behind either gifts or coupons that they want. | 2. The coupons or gifts are selected, check boxes are updated. |
| 3.Operators click on the redeem button. | 3. The Redeeming\_view is destroyed, the DB is updated, and customers are taken to customer view |
| 4.Operators click on back button | 4. The Redeeming\_view is destroyed, and customers are taken to customer\_view |
| 5. Operators click on the sign out button | 5. The Redeeming\_view is destroyed, and customers are taken to login\_view. |

Scenario Note:

Items 1,2,3 must be done in exact order. This use case permits operators to check for any check boxes. However, operators must check at least one check box before they click the redeem button. Item 4 can be done at any time. Item 3, 4 and 5 are mutually exclusive.

Post Condition:

All gifts and coupons that are checked and customers’ reward points are updated to database (if redeem button is selected).

The operators are taken to MyReward\_view/login\_view.

Exceptions: Use case Utilized:

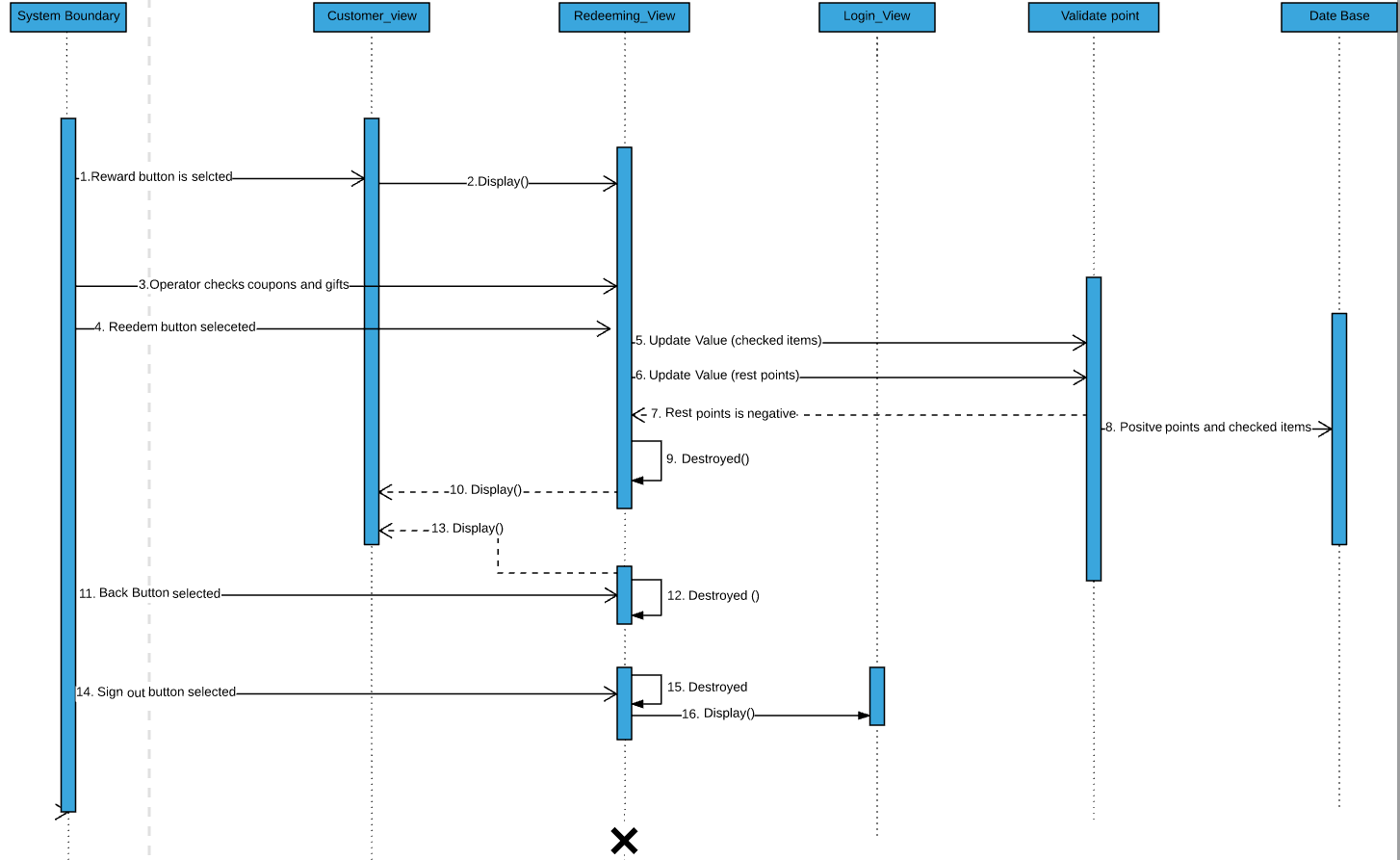
1, The DB cannot be accessed NONE

Required GUI:

1, Redeeming\_view 2, login\_view Timing Constraints:

3.Customer\_View NONE

UC21: Interaction Diagram



|  |
| --- |
| 1.Operator selects reward button  2. Display Redeeming\_View  3.Operators checks coupons and gifts boxes  4.If redeem button is selected then:  6-8. Enter correct coupons, gifts and rest points information into database  9.Destroy the Redeeming\_View  10. Display Customer\_ View  11. Else if back button selected:  12. Destroyed Redeeming\_View  13. Display the Customer\_View  14.Else the sign out button is selected:  15.Destroy the Redeeming\_View  16. Display Login\_View |

Use Case 28: Friend recommending

Overview:

This use case enables customers to invite their friends to sign up T2P and get benefits from it.

Precondition:

1. The database is accessible
2. Recommending\_view is displayed.

Scenario:

|  |  |
| --- | --- |
| **Action** | Software Reaction |
| 1. Operators click on Recommend button on the Customer\_view. | 1. Recommending\_View pop-up |
| 2.Operators enter their friends’ email addresses. | 2. Email address fields are updated |
| 3. Operators click on the send button. | 3. database is updated, emails with a link to T2P and “recommend friends” codes are sent to email addresses in the background. Operators are taken to the Customer\_view. |
| 5. Operators click the back button. | 5. Recommending friend view is killed. Operators are taken to Customer\_view. |
| 6.Operators click the sign\_out button | 6. Recommending view is killed. Operators are taken to Login\_view |

Scenario Note:

Item 1,2,3 have to be done in order. Item 3 can be clicked only if item 2 is full field by the operators. Date base is accessed only if item 3 has been done. This use case allows the modification of all these values. Items 3,5,6 are mutually exclusive.

Post Condition:

Emails are sent.

The operators are taken to Customer\_view.

The operators are taken to login\_view

Exceptions: Use case Utilized:

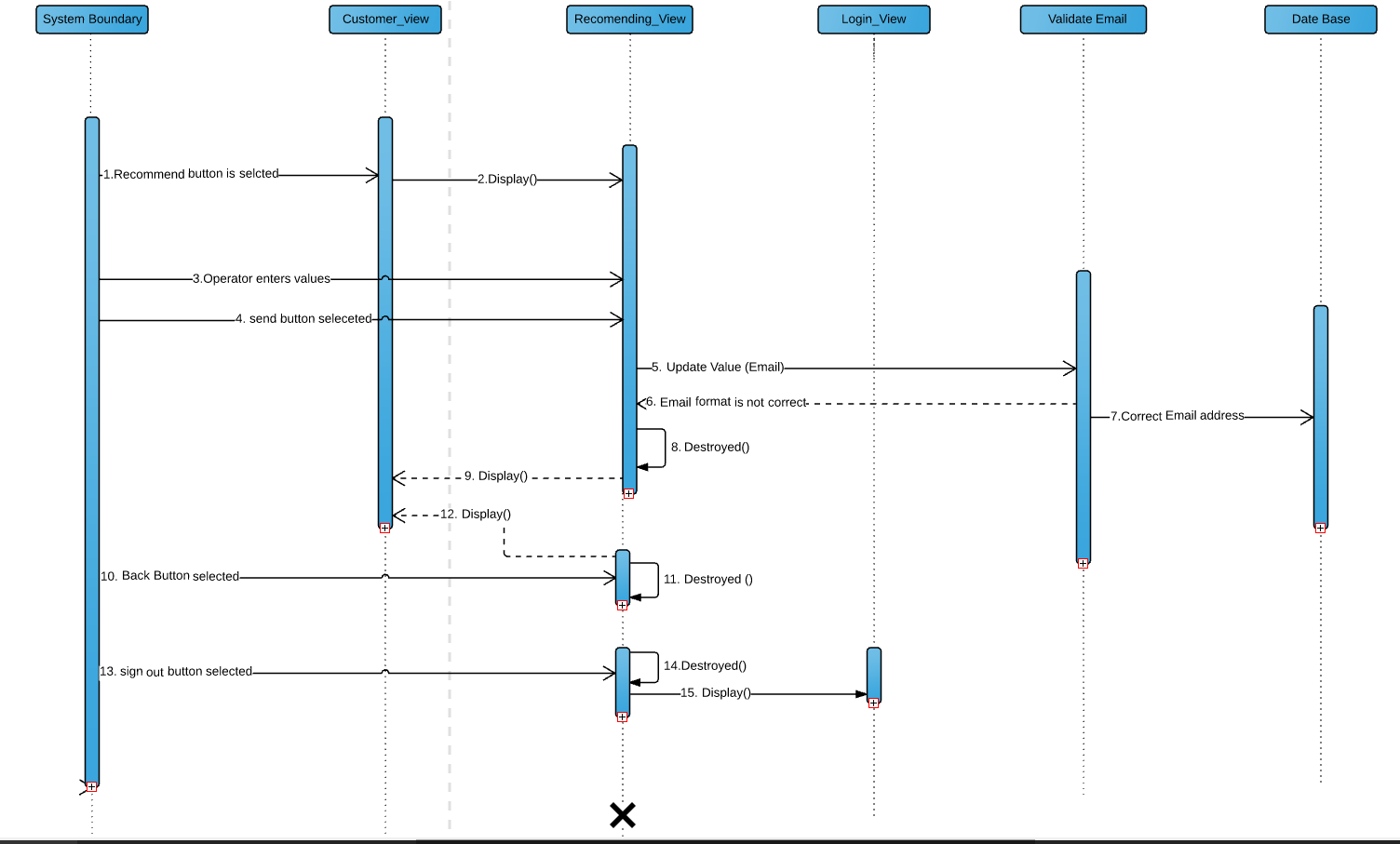
1, The DB cannot be accessed NONE

Required GUI:

1, Customer\_view 3. Recommending\_View Timing Constraints:

2, Login\_view NONE

UC 28: Interaction Diagram



|  |
| --- |
| 1.Operator selects recommend button  2. Display Recommending\_View  3.Operators enters Email Address  4.If send button is selected then:  5-7. Enter correct Email address into database  8.Destroy the Recommending\_View  9. Display Customer\_View  10. Else if back button is selected:  11. Destroyed the Recommending\_View  12. Display Customer\_View  13.Else the sign out button is selected  14.Destroy the Recommending\_View  15. Display Login\_View |

Use Case 29: New users get rewards from "recommend friends" code

Overview:

This use case enables customers who do not enter recommend friends code while sign up.

Precondition:

1. The database is accessible
2. RecommendingReward\_view is displayed.
3. Operators do not enter “recommend friends” code while sign up

Scenario:

|  |  |
| --- | --- |
| **Action** | Software Reaction |
| 1.Operators click Reward button on Reward button on customer view. | 1. RecomendingReward\_view is pop-up |
| 2.Operators enter their friends’ “Recommend Friends” code. | 2. code fields are updated |
| 3. Operators click on the enter button. | 3. The Reedeming\_view is updated, the DB is updated, and customers are taken to the Redeeming\_View |
| 5. Operators click the cancel button. | 5. RecommendingReward\_view is killed. Operators are taken to Customer\_view. |
| 6.Operators click the sign\_out button | 6. RecommendingReward\_view is killed. Operators are taken to login\_view |

Scenario Note:

Item 1,2,3 have to be done in order. Item 3 can be clicked only if item 2 is full field by the operators. The date base is updated only if item 3 is selected. This use case allows the modification of all these values. Item 3,5,6 are mutually exclusive.

Post Condition:

The inputs from operators are updated to database (if the enter button was clicked)

The operators are taken to customer\_view.

The operators are taken to login\_view

Exceptions: Use case Utilized:

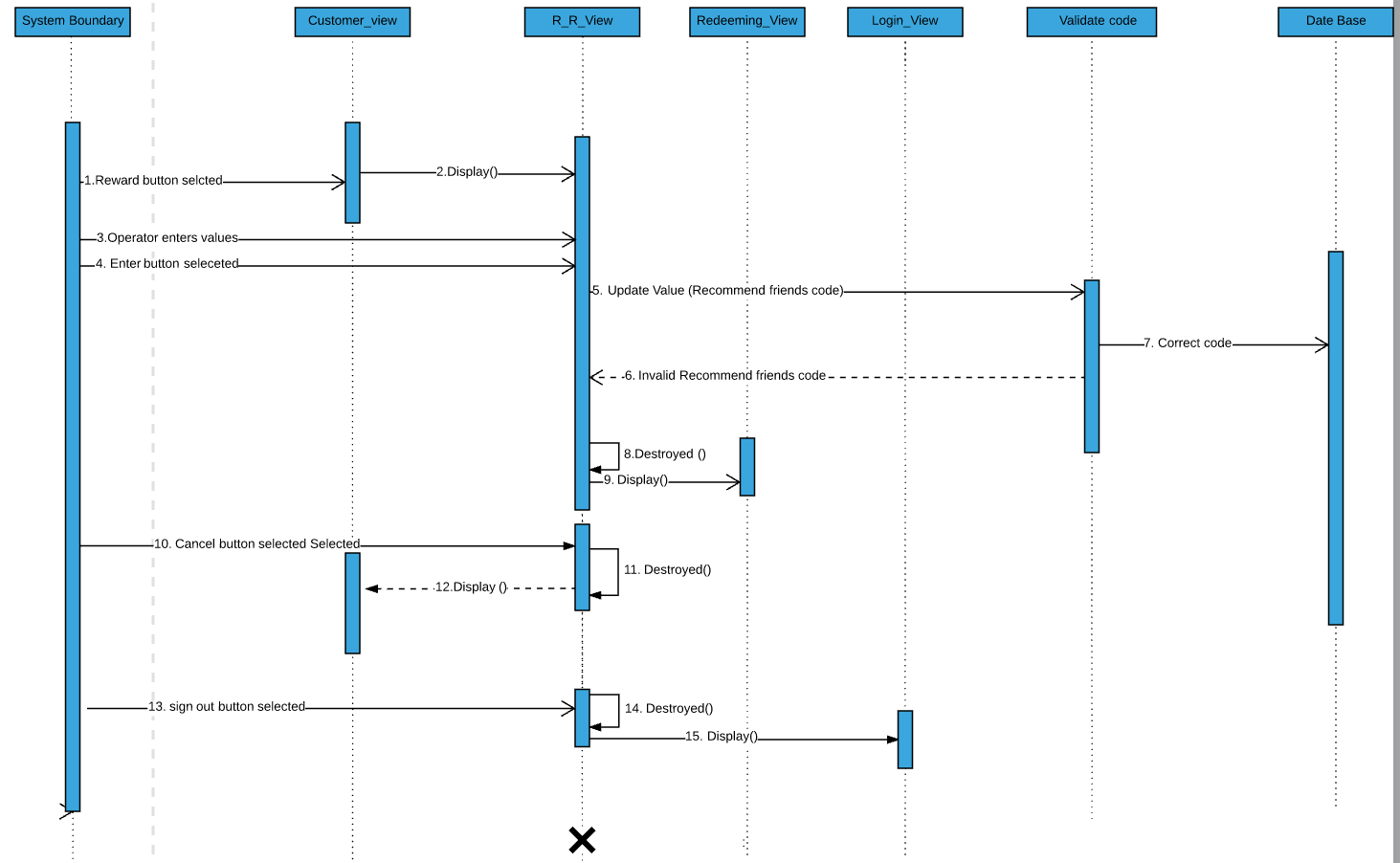
1, The DB cannot be accessed NONE

Required GUI:

1, customer\_view 3.RecomendingReward\_view Timing Constraints:

2, login\_view 4.Redeeming\_view NONE

UC 29: Interaction Diagram



|  |
| --- |
| 1.Operator selects reward button  2. Display R\_R\_View  3.Operators enters Recommend friends code  4.If Enter button is selected then:  5-7. Update valid Recommend friends code into database  8.Destroy the R\_R\_View  10. Display the Redeeming\_View  10. Else if cancel button is selected  11. Destroyed the R\_R\_View  12. Display the Costomer\_View  13.Else the sign out button is selected  14.Destroy the R\_R\_View  15. Display Login\_View |

\*R\_R view = RecomendingReward view

Use Case 35: Guests sign up.

Overview:

This use case enables guests to sign up to T2P by entering their names, addresses, email addresses, passwords, phone numbers and recommending friends.

Precondition:

1. The database is accessible.
2. Login view is displayed.

Scenario:

|  |  |
| --- | --- |
| **Action** | Software Reaction |
| 1.Operators click on sign up button on Login button | 1. login\_view will pop-up |
| 2. Operators enter their Email addresses. | 2. Email address fields are updated. |
| 3.Operators enter their passwords. | 3. Password address fields are updated. |
| 4. Operators reenter their passwords. | 4. Confirm password fields are updated. |
| 5.Operators enter their date of birth. | 5. date of birth fields are updated. |
| 6. Operators enter their First names. | 6. First name field is updated. |
| 7.Operators enter their Last names. | 7. Last name field is updated. |
| 8.Operators enter their phone numbers. | 8. phone number field is updated. |
| 9. Operators enter their user name | 9. username field is updated |
| 10. Operators reenter their username | 10. confirm user name field is updated |
| 9. Operators enters their friend’ “recommend friends” codes. | 9. Recommend friends code fields are updated. |
| 10. Operators click on the ok button. | 10. Sign\_up\_view is destroyed and operators are taken to customer\_view. |
| 11. Operators click on the back button. | 11. sign\_up\_view is destroyed and operators are taken to login\_view. |

Scenario Note:

Item 1,2,3,4,5,6,7,8, are not necessarily be done in order. Item 2 and 3 must be exactly same. Item 7 and 8 are not necessarily be fulfilled by operators. Item 1 to 6 have to be fulfilled by the operators. Items 9 and 10 are mutually exclusive. Database is updated only if item 10 is done.

Post Condition:

1. All inputs from the Operator is updated to database (if sign up button is selected)

2.The operators are taken to customer\_view.

3.The operators are taken to login\_view

Exceptions: Use case Utilized:

1, The DB cannot be accessed NONE

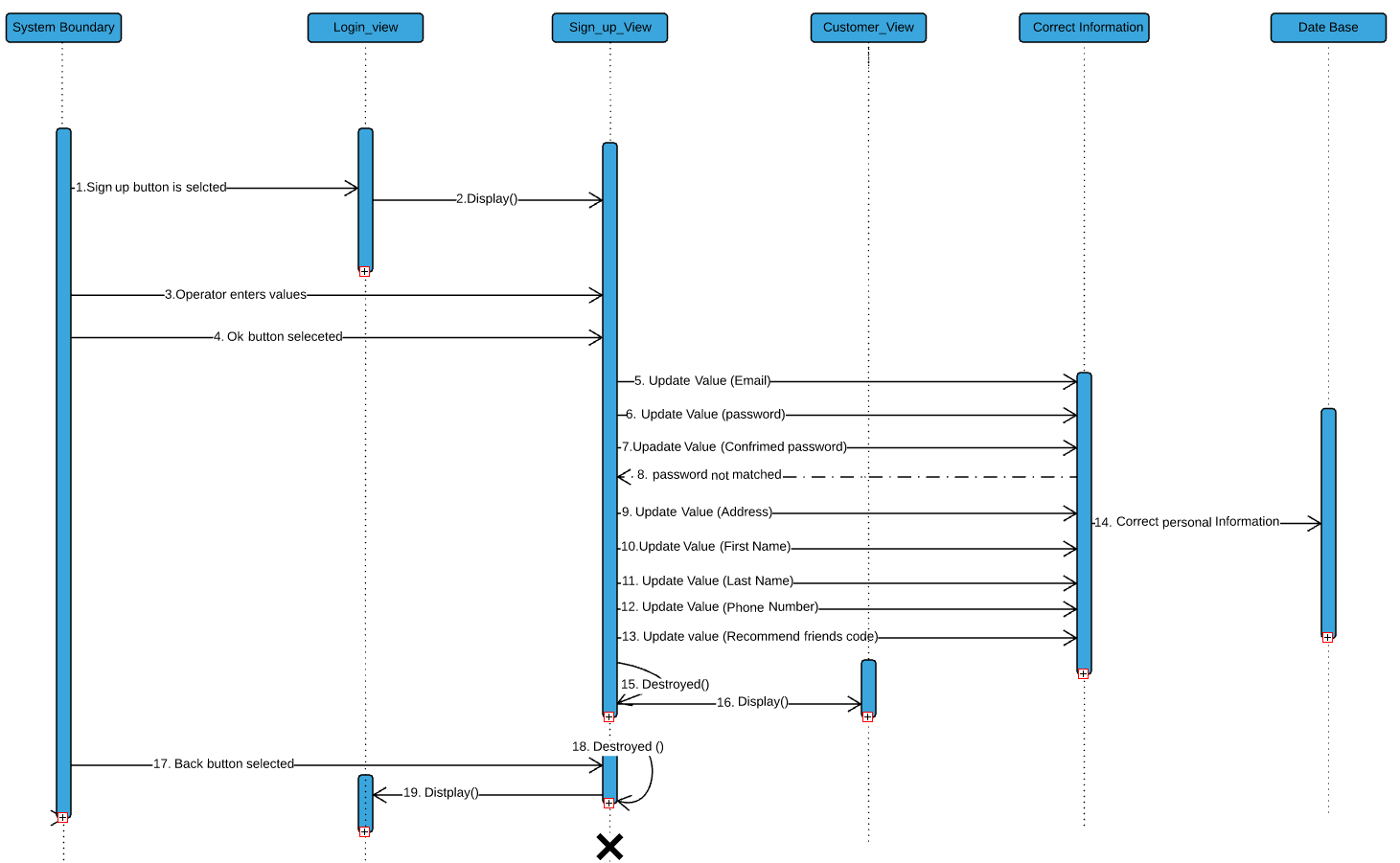
Required GUI:

1, customer\_view 3.Login\_view Timing Constraints:

2 login\_view NONE

-

UC 34: Interaction Diagram



|  |
| --- |
| 1.Operator selects Sign up button  2. Display Sign\_up\_View  3.Operators Enters Values  4.If Ok button is selected then:  5-14. Enter correct personal information into the database  15. Destroy the Sign\_up\_View  16. Display Customer\_View1  17.Else the back button is selected  18.Destroy the Sign\_up\_View  19.Display Login\_View |