HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION

**FACULTY FOR HIGH QUALITY TRAINING**



**REPORT**

**OBJECT – ORIENTED SOFTWARE ENGINEERING**

**PROJECT: BUILDING ANDROID APPLICATION**

**BOOKING CINEMA TICKET**

**Instructor**: **Mai Anh Thơ**

|  |  |
| --- | --- |
| **STUDENT CODE** | **FULL NAME** |
| **20110388** | **Nguyễn Trung Nguyên** |
| **20110356** | **Phan Tấn Cường** |
| **20110421** | **Nguyễn Thanh Minh Triết** |

***Ho Chi Minh City***

***13/4/2023***

**COMMENTS**

**…………………………………………………………………………………….…………………………………………………………………………………….**

**…………………………………………………………………………………….**

**…………………………………………………………………………………….**

**…………………………………………………………………………………….**

**…………………………………………………………………………………….**

**…………………………………………………………………………………….**

**…………………………………………………………………………………….**

**…………………………………………………………………………………….**

**…………………………………………………………………………………….**

**…………………………………………………………………………………….**

**…………………………………………………………………………………….**

**Score: …………………………**

*Sign*

**Mai Anh Thơ**

# ACKNOWLEDGEMENT

For the completion of this thesis, I would like to express my sincere thanks to:

The Board of Directors of Ho Chi Minh City University of Technology and Education for creating favorable conditions in terms of facilities with a modern library system, a variety of books and documents that are convenient for searching and researching information.

I would like to thank the subject lecturer – Ms. Mai Anh Thơ for his dedicated and detailed teaching so that I have enough knowledge and apply them to this report.

Due to the lack of experience in making talent as well as the limitations of knowledge, in the report will certainly inevitably have shortcomings. We look forward to receiving comments, suggestions and criticism from the teacher to improve the report.

Finally, I would like to wish you good health, success and happiness.

**TABLE OF CONTENT**

[ACKNOWLEDGEMENT 2](#_Toc132268332)

[INTRODUCTION 4](#_Toc132268333)

[REQUIREMENT 5](#_Toc132268334)

[USE CASE DIAGRAM 7](#_Toc132268335)

[CONCEPTUAL DIAGRAM 9](#_Toc132268336)

[1. User 10](#_Toc132268337)

[1.2. Register 14](#_Toc132268338)

[1.3. Use Case Log Out: 17](#_Toc132268339)

[1.4. Use Case Forget Password 20](#_Toc132268340)

[1.5. Use Case View Information Movie 24](#_Toc132268341)

[1.6. Use Case Manage Account: 28](#_Toc132268342)

[1.7. Use Case Change Personal’s Information 32](#_Toc132268343)

[1.8. Use Case Change Password 34](#_Toc132268344)

[1.9. Use Case Booking Tickets 38](#_Toc132268345)

[1.10. Use Case Booking Food & Beverages: 42](#_Toc132268346)

[1.11. Use Case Payment 45](#_Toc132268347)

[1.12. Use Case Film History 50](#_Toc132268348)

[2. Admin 53](#_Toc132268349)

[2.1. Use Case Manage Film 53](#_Toc132268350)

[2.2. Use Case Manage Schedule 56](#_Toc132268351)

[2.3. Use Case Manage Seat 60](#_Toc132268352)

[2.4. Use Case Manage Food & Beverages 63](#_Toc132268353)

[2.5. Use Case Statictis 67](#_Toc132268354)

[WORK ASSIGNMENT TABLE 71](#_Toc132268355)

[REFERENCES 72](#_Toc132268356)

# INTRODUCTION

The group's topic is to design an application on the Android platform, capable of helping users can book movie tickets online without going to competitors,

The group chose this topic because of the fact that nowadays, when people want to see a movie, people have to go to the theater to buy tickets and have to wait until the movie starts showing. If we get close to the newly purchased showtime, there will be no more tickets or no more places we want. On the other hand, when buying tickets, they often have to wait in long lines on weekends to buy tickets. Therefore, when this online ticket booking application is completed, it will help a lot of people save time when they want to see a certain movie, and can watch it conveniently.

# REQUIREMENT

**Functional requirement :**

The system consists of the following main components:

1. Guest section (User is not logged in / does not have an application account):

* Guest can see posters of new cinema movies displayed on the main screen.
* Guests can click on the movie poster to view information related to the movie (actors, movie intro, film genre, etc.).
* On the movie information display, Guest can click on the movie trailer to view the trailer.
* Guests can see the critics' and audience's ratings below shown on the info screen (under the movie trailer video).
* The notice board showing two methods of login and registration must appear, when the Guest clicks on the booking method in the movie information screen; as well as in the menu section on the main screen.
* Guest can choose the registration method to create their app account.

2. User section (User logged in with application account):

* User is entitled to Guest features and can book tickets.
* User can recover password if forgot password.
* User can change personal information as well as password (for password, a confirmation code will be sent to User's contact method).
* In the ticket booking section, users can choose movie information (cinema, showtime, seats, type of seats).
* User can order more fast food of the theater.
* Users can use vouchers/points cards before paying (if any)
* Users can pay using multiple payment methods.
* After payment, User will have payment invoice shown on screen.
* User can view transaction history in the last 3 months.

3. Admin section:

* Admin interacts on a separate interface that is different from that of Guest and User.
* Admin can add delete edit movie / movie information / fast food information.
* Admin can see the theater's sales through the theater's monthly revenue statistics table

**Non-functional requirements :**

1. The system ensures ease of use for users

# USE CASE DIAGRAM

1. **User Use Case:**

****

1. **Admin Use Case:**

****

# CONCEPTUAL DIAGRAM



## User

* 1. **Use Case Login:**

**Use case description**

|  |  |
| --- | --- |
| **Use Case ID** | UC-1.1 |
| **Use Case Name** | Login |
| **Description** | This use case allows the actor to login application by account. |
| **Actor(s)** | User/Admin |
| **Priority** | Must Have |
| **Precondition(s):** | Users need to have a BoxCine app account |
| **Post-Condition(s):** | User successfully login to the application. The system records successful login to BoxCine application. |
| **Main Flow** | 1. User enters account information with user gmail and password. 2. User selects “User” role 3. The user presses the “Login” button. 4. System check login information 5. The system authenticates the login information successfully. 6. The system returns the user main form |
| **Alternative Flow** | [1.1]Admin enters account information with admin gmail and password  [2.1]Admin selects "Administrrator” role  [6.1]The system return the admin main form |
| **Exception Flow** | [3.1] User enters incorrect login information, user repeats step (1).  [3.2] User enter information “User” role. However,user selects “Administrator” radio button. The system notifies error message and user repeats step(4).  [3.3] User enter information “Administrator” role. However,user selects “User” radio button. The system notifies error message and user repeats step(4).  [3.4] User have not sign up an account yet, user do UC-1.2 and UC-1.1 end. |

**Sequence diagram**

****

**Collaboration diagram**

****

## Register

**Use case description**

|  |  |
| --- | --- |
| **Use Case ID** | UC-1.2 |
| **Use Case Name** | Register |
| **Description** | This use case allows actor to create an application account and allows the actor to sign in application after having an account. |
| **Actor(s)** | User/Admin. |
| **Priority** | Must Have |
| **Precondition(s):** |  |
| **Post-Condition(s):** | The system returns actor main form after registering actor account. |
| **Basic Flow** | 1. User presses the “Register” button. 2. The system returns the BoxCine application account registration form. 3. User fills in gmail and password field and selects “User” role. 4. User presses the “Register” button. 5. The system returns user main form |
| **Alternative Flow** | [1.1]Admin presses the “Register” button  [2.1] Admin fills in gmail and password field and selects “Administrator” role.  [3.1] Admin presses the “Register” button.  [4.1]The system returns admin main form |
| **Exception Flow** | [4.1] The user enters the invalid information, the system reports an error, the user performs step (3) |

**Sequence diagram**

**Collaboration diagram**

## Use Case Log Out:

**Use case description**

|  |  |
| --- | --- |
| **Use Case ID** | UC-1.3 |
| **Use Case Name** | Log out |
| **Description** | This Use Case allows users to log out of their application account. |
| **Actor(s)** | User |
| **Priority** | Must Have |
| **Precondition(s):** | The user account has been successfully logged into the application.  The user's device is already connected to the internet. |
| **Post-Condition(s):** | User successfully logs out of the application.The system records the successful logout from BoxCine application. |
| **Basic Flow** | (1) User accesses BoxCine application.  (2)The user has logged into the BoxCine application.  (3) The user chooses the method of logging out of the system.  (4) The system validates the login information successfully and allows the user to access the application with the guest account. If the user does not confirm the logout from the system, the system will keep the status quo.  (5) The system records successful logout from BoxCine application. |
| **Alternative Flow** |  |
| **Exception Flow** | 4c. The system validates the logout information fails and displays the message.Use Case stops. |

**Sequence diagram**

****

**Collaboration diagram**

## Use Case Forget Password

**Use case description**

|  |  |
| --- | --- |
| **Use Case ID** | UC-1.4 |
| **Use Case Name** | Forget Password |
| **Description** | This Use Case allows users to recover passwords when they forget or lose their current passwords. |
| **Actor(s)** | Guest |
| **Priority** | Must Have |
| **Precondition(s):** | The user account has been created and exists on the system. |
| **Post-Condition(s):** | The user receives the message "BoxCine application account password reset successful" from the Gmail communication method. |
| **Basic Flow** | (1) User accesses BoxCine application.  (2) User chooses to forget BoxCine account password  (3) User fills in the account name via BoxCine application  (4) The system will send sms containing OTP code to the user's phone number.  (5) After the user authenticates the OTP code on the system, the system authenticates and sends a password reset application to the user via the email registered by the user.  (6) The user enters gmail and clicks on the website link to reset the password on Gmail.  (7) After entering the new password and re-authenticating the new password, the user selects the “Agree” button and the system will re-enter the user password.  (8) After validating the new password, the system will take the user back to the login page |
| **Alternative Flow** |  |
| **Exception Flow** | 4c. The system validates that the account information is incorrect or does not exist.  The system displays the wrong account name or the account name does not exist.  4c(1) User did not receive OTP  User can ask the system to resend the code  4c(2)User entered wrong password OTP  The system asks the user to re-enter the OTP code. |

**Sequence diagram**

****

**Collaboration diagram**

## Use Case View Information Movie

**Use case description**

|  |  |
| --- | --- |
| **Use Case ID** | UC-1.5 |
| **Use Case Name** | View Information Movie |
| **Description** | Allows user to view movie descriptions (genre, actors, directors, etc.) |
| **Actor(s)** | User |
| **Priority** | Must have |
| **Precondition(s):** | User successfully login application |
| **Post-Condition(s):** | The user will enter a describing information form which related to the movie. |
| **Basic Flow** | 1. The system asks to display a list of movie posters. 2. The list of posters will be displayed on the main screen of the application. 3. User clicks on the movie poster image. 4. The application displays a form describing the movie information. |
| **Alternative Flow** | [3.1] User clicks on thegenre poster image.   1. The list of posters will be displayed on the genre form of the application. 2. User clicks on the movie poster image. 3. The application displays a form describing the movie information |
| **Exception Flow** | (2.1) The main screen does not display a list of movie posters. The system reports an error and the use case stops.  (4.1) The movie information description form screen does not display movie related information. The system reports an error and the use case stops.  (4.2) The movie information description screen shows the wrong movie information.  (4.3) The movie trailer video in the movie information description form displays a network error message. User checks the network and does step (2)  (4.4) After the user clicks the movie trailer video, the video shows the wrong movie trailer video with the movie the user selected in the main screen. |

**Sequence diagram**

****

**Collaboration diagram**

****

## Use Case Change Password

**Use case description**

|  |  |
| --- | --- |
| **Use Case ID** | UC-1.8 |
| **Use Case Name** | Change Password |
| **Description** | This use case allows the user to change the user's account password, the system must update for the next logins. |
| **Actor(s)** | User |
| **Priority** | Optional |
| **Precondition(s):** | User must be awake UC-1.6 |
| **Post-Condition(s):** | The user receives the message “Password changed successfully from the communication method. |
| **Basic Flow** | (1) After entering the "personal information" form successfully, the user clicks on the password item.  (2) in the password section, the user password will be displayed.  (3) The user selects the change button in the password section  (4) The system will switch to the password change form.  (5) User enters a new password.  (6) User presses the button and confirms the new password.  (7) The system sends a confirmation code to the user gmail.  (8) User waits and receives confirmation code  (9) The user enters the confirmation code in the password change form and clicks the “Confirm” button.  (10) The system notifies the successful password update and the user confirms it. |
| **Alternative Flow** |  |
| **Exception Flow** | (6.1) User enters wrong information system reports error in step 7, user repeats step (4)  (8.1) The user did not receive a message containing the confirmation code to the user gmail. User can wait a few minutes and then check gmail or user can go back to step 7 and press the "resend code" button.  (9.1) User enters incorrect confirmation code, user repeats step 10.  (9.2) User timed out to enter confirmation code, user repeats from step 7 and clicks "Resend code". |

**Sequence diagram**

****

**Collbroartion diagram**

****

## Use Case Booking Tickets

**Use case description**

|  |  |
| --- | --- |
| **Use Case ID** | UC-1.9 |
| **Use Case Name** | Booking Tickets |
| **Description** | This Use Case allows users to book tickets directly on this movie ticket booking application and can issue that ticket to the cashier for confirmation. |
| **Actor(s)** | User |
| **Priority** | Must Have |
| **Precondition(s):** | User must login and user has already been in view detail information film form. |
| **Post-Condition(s):** | Display the ticket information screen that you booked. |
| **Basic Flow** | 1. User presses “book ticket” button 2. The system returns to booking ticket. 3. User selects cinema 4. User selects screening time 5. User selects the seat position and the number of viewing seats. 6. User selects payment method. 7. User presses “Accept” button 8. The system returns detail information ticket screen |
| **Alternative Flow** |  |
| **Exception Flow** | [7.1] User have not selected fields (cinema/screening time/number of seat) , the system notifies error message and UC-1.9 end. |

**Sequence diagram**

****

**Collaboration diagram**

## Use Case Booking Food & Beverages:

**Use case description**

|  |  |
| --- | --- |
| **Use Case ID** | UC-1.10 |
| **Use Case Name** | Booking Food & Beverages |
| **Description** | This use case allows the user to order more food or drink while watching the movie, and the information about the food or drink will display the ticket information form. |
| **Actor(s)** | User |
| **Priority** | Optional |
| **Precondition(s):** | User have to be in booking ticket form. |
| **Post-Condition(s):** | The food/drink information selected by the user is displayed on the booking ticket form. |
| **Basic Flow** | The system returns a form containing a list of food and food items.  User chooses the food or drink which user want to select  User presses “Accept” button. |
| **Alternative Flow** |  |
| **Exception Flow** | (1.1) The system does not display the food & drink list, the system reports an error and the UC-1.10 end. |

**Sequence diagram**

****

**Collaboration diagram**

## Use Case Payment

**Use case description**

|  |  |
| --- | --- |
| **Use Case ID** | Usecase-1.11 |
| **Use Case Name** | Payment |
| **Description** | This use case allows users to pay their own costs. |
| **Actor(s)** | User |
| **Priority** | Must Have |
| **Precondition(s):** | -User has logged in account  -User has made the booking function  -Users must have an e-wallet, a bank account |
| **Post-Condition(s):** | - The user successfully pays the cost of using the service.  - The system records the successful payment operation. |
| **Basic Flow** | (1) In the ticket information form, the user presses the "Payment" button.  (2)The user selects the payment method and confirms the payment of his expenses.  (3) The system will record it after the user has successfully paid and send the invoice to the user.  (4) The system records successful payment operations on the system. |
| **Alternative Flow** | 2a.User chooses payment method by bank account.  2a(1) The system switches the screen to the bank selection screen and confirms the bank information.  3a.After the user selects the bank and confirms the bank information and selects the payment button.The user needs to enter the security code sent by the bank.  4a.User selects the cost payment confirmation button.  5a.After successful transfer, the system confirms and takes the user back to the app  2b.User chooses payment method by e-wallet.  2b(1) The system switches the screen to the screen to select the e-wallet the user uses and confirm the wallet information.  3b. After the user selects the e-wallet, confirm the e-wallet information in use and select the payment button. The user needs to enter the security code of the e-wallet sent back.  4b.User selects the payment confirmation button  5b.After successful transfer, the system confirms and takes the user back to the application. |
| **Exception Flow** | 4c. The system validates incorrect bank credentials, e-wallets  4c(1) User chooses to cancel login.  Use Case stops.  4c(2) The system of the bank, e-wallet confirming customer account information is not enough to pay the cost  Use Case stops.  4c(3)The user cancels the payment or closes the application.  Use Case stops. |

**Sequence diagram**

******

**Collaboration diagram**

## Use Case Film History

**Use case description**

|  |  |
| --- | --- |
| **Use Case ID** | Usecase-1.12 |
| **Use Case Name** | Film History |
| **Description** | This usecase allows users to view transaction history with the store for tickets, food and drinks |
| **Actor(s)** | User |
| **Priority** | Must Have |
| **Precondition(s):** | -User has logged in account  -The user has successfully paid for the in-app transaction. |
| **Post-Condition(s):** | -Users can review their transaction history.  -Users can use transaction history such as invoices to get movie tickets, food and drinks. |
| **Basic Flow** | (1)User accesses BoxCine application.  (2)Users go to the list and click on “my ticket” to review the transaction history.  (3) The system will display the transactions that the user's account has transacted online within the last 3 months. |
| **Alternative Flow** |  |
| **Exception Flow** | 4c. Database system error Use Case stopped |

**Sequence diagram**

****

**Collaboration diagram**



## Admin

## Use Case Manage Film

**Use case description**

|  |  |
| --- | --- |
| **Use Case ID** | UC-2.1 |
| **Use Case Name** | Manage Film |
| **Description** | This use case allows admin to add, edit, delete movies or movie information. For users to interact and use the service |
| **Actor(s)** | Admin |
| **Priority** | Must Have |
| **Precondition(s):** | -Admin needs to have the information of the movie to be managed (movie name, description, duration, rating, director's name, ...) |
| **Post-Condition(s):** | -Admin receives a success message from the notification window |
| **Basic Flow** | (1) Admin accesses BoxCineAdmin application.  (2) Admin clicks the 'Manage movies' button.  (3) The system returns the movie management form  (4) Admin clicks 'add' button to add a new movie [4.1][4.2].  (5) Admin enters information in the movie management form including information (movie name, description, duration, rating, director's name, ...) and press the 'Save' button to complete the job [5.1] .  (6) The system returns the admin to the home page. |
| **Alternative Flow** | [4.1] Admin clicks the 'Edit' button to edit a movie's information.  [4.2] Admin clicks the 'Delete' button to delete a movie. Skip step 5, continue to step 6  [5.1] Admin edits information on the movie management form including information (movie name, description, duration, rating, director's name, ...) and press the 'save' button to complete the work. |
| **Exception Flow** | 5.a Admin entered incorrect input information. The system displays the message 'Incorrect input information' and lets the admin re-enter the data.  4.a. Admin enters information already on the database system. The system displays the message 'The movie you entered is already on the system, please try again.' and the system returns the admin to the homepage.  4.2.a Admin deletes an upcoming movie that has already been booked. The system reported an "cannot delete" error and returned the admin to the homepage |

**Sequence diagram**

****

**Collaboration diagram**

## Use Case Manage Schedule

**Use case description**

|  |  |
| --- | --- |
| **Use Case ID** | UC-2.2 |
| **Use Case Name** | Manage Schedule |
| **Description** | This use case allows the admin to add, delete, and edit the projections of each movie. For users to interact and use the service. |
| **Actor(s)** | Admin |
| **Priority** | Must Have |
| **Precondition(s):** | Admin needs to have information about the screenings to be managed (Room number, Showtime, end time, ) |
| **Post-Condition(s):** | Admin receives success message from notification window |
| **Basic Flow** | (1) Admin accesses BoxCineAdmin application.  (2) Admin clicks the 'Manage projection' button.  (3) The system returns the list of movies that need to be managed.  (4) Admin selects the movie that needs to be managed.  (5) The system returns the selected movie presentation management form.  (6) Admin clicks 'Add' button to add a new show for that movie [6.1][6.2].  (7) Admin enters information into the projection management form including information (showing date, show time) and presses the 'save' button to complete the work [7.1].  (8) The system returns the admin to the homepage. |
| **Alternative Flow** | [6.1] Admin clicks the 'Edit' button to edit the show of a movie.  [6.2] Admin clicks the 'Delete' button to delete a show of the movie. Skip step 7, continue to step 8  [7.1] Admin edit the information on the projection management form including the information (showing date, time) and press the 'Save' button to complete the work. |
| **Exception Flow** | 7.a. Admin entered incorrect input information. The system displays the message 'Incorrect input information' and lets the admin re-enter the data.  6.a. Admin enters information already on the database system. The system displays the message 'The movie you entered is already on the system, please try again.' and the system returns the admin to the homepage.  6.2.a Admin deletes a show that has already been booked. The system reported an "cannot delete" error and returned the admin to the homepage |

**Sequence diagram**

****

**Collaboration diagram**

## Use Case Manage Seat

**Use case description**

|  |  |
| --- | --- |
| **Use Case ID** | UC-2.3 |
| **Use Case Name** | Manage Seat |
| **Description** | This use case allows admin to add, edit, delete seats to watch movies. For users to interact and use the service. |
| **Actor(s)** | Admin |
| **Priority** | Must Have |
| **Precondition(s):** | Admin needs to have information about the seats to be managed (Room Number, Row, Seat Number, Status, Seat Type) |
| **Post-Condition(s):** | Admin receives success message from notification window |
| **Basic Flow** | (1) Admin accesses BoxCineAdmin application.  (2) Admin clicks the button “Manage seats”.  (3) The system returns a list of movies that need to be managed.  (4) Admin selects a movie to manage seats.  (5) The system returns the seat management form of the selected movie.  (6) Admin clicks on the available seat to switch to disable mode [6.1].  (7) Admin click save to save the changes [7.1].  (8) The system returns the admin to the homepage. |
| **Alternative Flow** | [6.1] Admin click on disabled seat to switch to usable mode. Then continue to step 7 |
| **Exception Flow** | 6.a. Admin disable seats already booked. The system says "This seat cannot be disabled, please try again". Go back to step 5 |

**Sequence diagram**

****

**Collaboration diagram**



## Use Case Manage Food & Beverages

**Use case description**

|  |  |
| --- | --- |
| **Use Case ID** | UC-2.4 |
| **Use Case Name** | Manage Food & Beverages |
| **Description** | This use case allows admin to add, remove, edit food & drinks. For users to interact and use the service. |
| **Actor(s)** | Admin |
| **Priority** | Must Have |
| **Precondition(s):** | Admin needs to have information about food & drinks to be managed (Name of dish or drink, unit price, quantity) |
| **Post-Condition(s):** | Admin receives success message from notification window |
| **Basic Flow** | (1) Admin accesses BoxCineAdmin application.  (2) Admin clicks the button 'Manage food & drink'.  (3) The system returns the food & beverage management form.  (4) Admin clicks 'Add' button to add a new food & drink [6.1][6.2].  (5) Admin enters information into the food & drink form including information (Name of dish or drink, unit price, quantity) and presses the 'Save' button to complete the job [7.1].  (6) The system returns the admin to the home page. |
| **Alternative Flow** | [4.1] Admin clicks the 'edit' button to edit a food & drink.  [4.2] Admin clicks 'delete' button to delete a food & drink. Skip step 5, continue to step 6  [5.1] Admin edit the information on the food & beverage management form including the information (Name of dish or drink, unit price, quantity) and press the 'save' button to complete the job. |
| **Exception Flow** | 5.a Admin entered incorrect input information. The system displays the message 'Incorrect input information' and lets the admin re-enter the data.  4.a. Admin enters information already on the database system. The system displays the message 'The food & drink you entered is already on the system, please try again.' and the system returns the admin to the homepage. |

**Sequence diagram**

****

**Collaboration diagram**

## Use Case Statictis

**Use case description**

|  |  |
| --- | --- |
| **Use Case ID** | UC-2.5 |
| **Use Case Name** | Statictis |
| **Description** | This use case allows the admin to calculate the theater's revenue in a certain period of time |
| **Actor(s)** | Admin |
| **Priority** | Must Have |
| **Precondition(s):** | Admin needs to have the previous information (Revenue in days) |
| **Post-Condition(s):** | The system will return a page of results that the admin searched for |
| **Basic Flow** | (1) Admin accesses BoxCineAdmin application.  (2) Admin selects 'Revenue Statistics' function.  (3) Admin selects the departure time and selects the next time for the system to calculate. The admin then clicks the 'Stats' button  (4) After having input data, the system will display a screen of output statistics.  (5) Admin clicks “OK” button to exit.  (6) The system will return the admin to the homepage. |
| **Alternative Flow** | [5.1] The user selects the 'Print' button. The system will print the statistics to a pdf file and save it to the admin's computer. Then do step 6 |
| **Exception Flow** | [3.1] Admin enters information that the next time is smaller than the input time. The system says 'invalid input, please try again' and go back to step 3 |

**Sequence diagram**

****

**Collaboration diagram**



# WORK ASSIGNMENT TABLE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Task | Name | Result | Sign |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

# REFERENCES