**Catch the Movies!**

**Project Mission Statement:**

Aims to provide a platform for users to view and book movie tickets from various cinemas companies such as Cathay, Golden Village and Shaw.

This will eliminate the trouble of having to view across the different platforms.

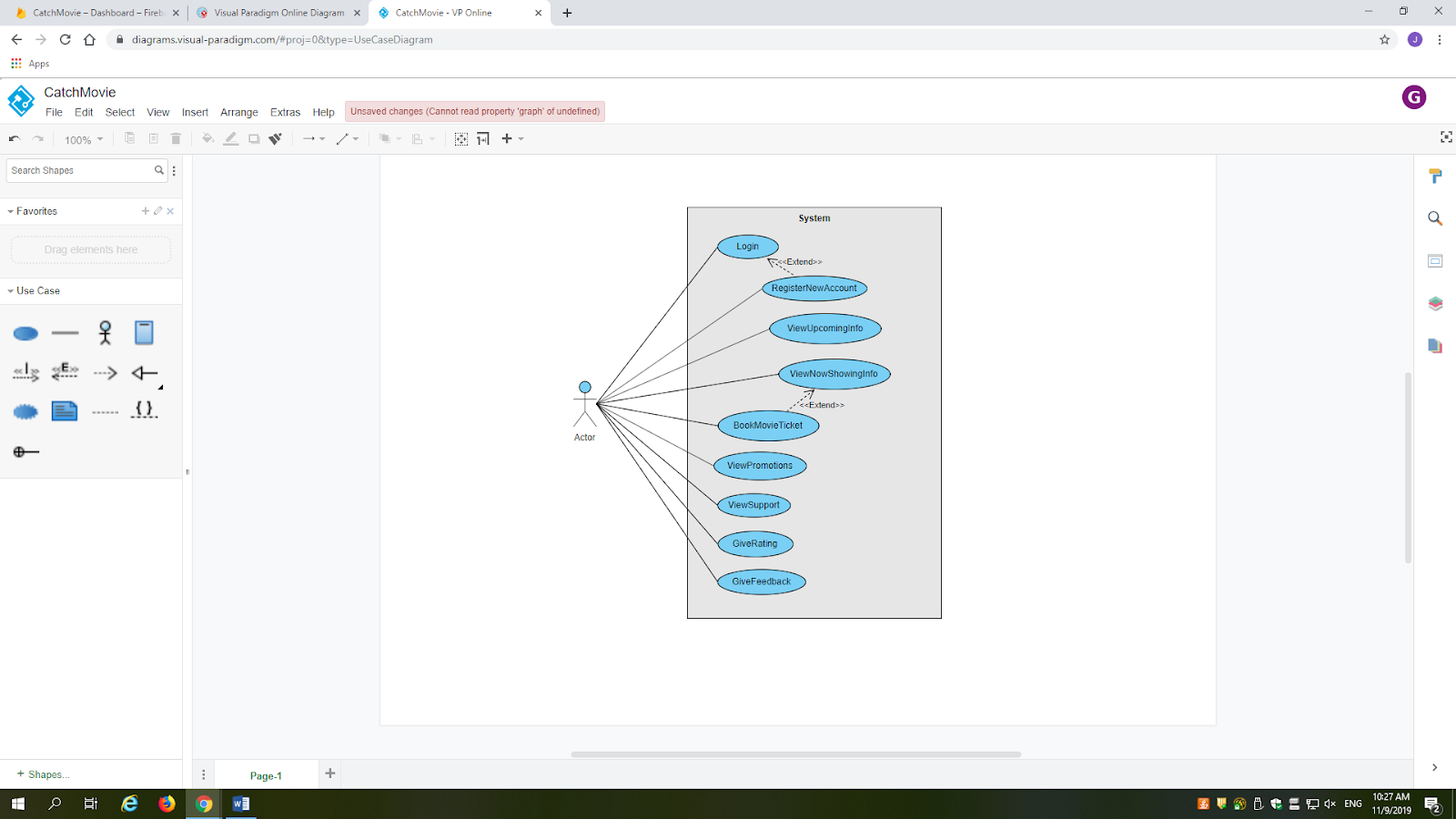
**Functional Requirements:**

1. The system must allow users to enter the system by displaying a login page.
   1. Existing users must be able to login with their member account by entering their username and password.
      1. The system must be able to validate that both fields are filled up.
      2. The system must be able to validate that username and password are correct
      3. If login details are incorrect, the system must display an error message.
2. The system must be able to allow new users to register a member account.
   1. The system must ask the user to enter his personal information.
   2. The personal information must include
      1. Email address
      2. Username
      3. Password
      4. Confirm Password
   3. The system must validate all required fields are filled up and correct
      1. Validity of email address
      2. Username availability
      3. Validity of password
      4. Confirm password matches password
   4. The system must be able to store credentials of new user.
3. The system must allow users to query the system for getting movie information.
   1. Users must be able to query the system for ‘Now showing’ movies.
      1. Users must be able to filter ‘Now Showing’ movies by supplying a movie title or a date or both movie title and date.
         1. A query containing a movie title must display movie information for the supplied movie title which includes:
            1. Movie description (genre, running time, short synopsis)
            2. Movie rating
            3. Movie date
            4. Movie timing
            5. Cinema name
            6. Ticket price
            7. Number of seats left
         2. A query containing a date must display movie information for the supplied date which includes:
            1. Movie titles available for that date
            2. Movie descriptions (genre, running time, short synopsis)
            3. Movie ratings
            4. Movies timings
            5. Cinema names
            6. Ticket prices
            7. Number of seats left
         3. A query containing both movie title and date must display movie information for the supplied movie title and supplied date which includes:
            1. Movie description (genre, running time, short synopsis)
            2. Movies rating
            3. Movies timing
            4. Cinema name
            5. Ticket price
            6. Number of seats left
   2. Users must be able to query the system for upcoming movies.
      1. A query for upcoming movies must display movie information of all upcoming movies which includes:
         1. Movie title.
         2. Movie description
         3. Release date.
4. The system must allow users to book movie tickets through official cinema website
   1. The system must direct users to the booking page of the official cinema website
5. The system must allow users to view ongoing promotions.
6. The system must allow users to view support/help.
7. The system must allow users to give ratings for the application.
   1. The system must retrieve all now showing movie titles
   2. The system must allow user to give rating from a scale of 1-5
8. The system must allow users to give feedback about the app via a comment box
   1. The user must be able to write their feedback of at most 200 words
9. The system must allow users to exit/log out of the system by bringing the user back to login page.

**Non-functional Requirements:**

1. The system must strive for consistency
   1. The layout of the application must be consistent throughout
2. The system must permit easy reversal actions
3. The system must offer informative feedback
   1. Should be able to indicate to the user where they are at in the process
4. The system must reduce short term memory load
   1. The user should not have to remember more than 5 information
5. The system must be able to support internal locus of control
   1. To give the user the sense of control of events occurring and the system will behave as what they expect
6. The system must return the queries within 5 seconds
7. The system must be able to prevent SQL injection
8. The system must offer features that are intuitive to use
   1. System should be easy to navigate
9. The system must be able to redirect to respective cinema’s payment page within 10 seconds

**Use Case Diagram**



**Use Case Description**

**Use Case Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1 | | |
| Use Case Name: | RegisterNewAccount | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | New user register for a member account |
| Preconditions: | This use case extends the Login use case. It is invoked when the user chooses to create a new account in the login UI. |
| Postconditions: | The system must report a message indicating successful registration and store the new user credential in database |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user chooses to register for a new account in login page 2. The system redirects user to new member registration page 3. The user fills his email address, username, password and confirms the password 4. The system checks the validity of each of the required field 5. If all information is verified, the system stores the credentials of the new user in database and displays message “Successful registration” |
| Alternative Flows: | AF-S5:   1. If any of the required field is invalid or empty, the system display “Invalid registration details” 2. The system returns to step 2   AF-S5:   1. If the password and confirm password fields do not match, the system display “Password do not match” 2. The system returns to step 2 |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**Use Case Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2 | | |
| Use Case Name: | Login | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | Existing user login to the system |
| Preconditions: | The user must have a registered account stored with the system in order to login |
| Postconditions: | The system must report a message indicating successful login |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user chooses to login to his account in login page 2. The user fills his username and password 3. The system checks the validity of the username and password 4. If the credential is verified, the system displays message “Successful login” |
| Alternative Flows: | AF-S5:   1. If the credential is invalid (username does not exist, or username and password do not match), the system displays message “Incorrect login details” 2. The system returns to step 2 |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**Use Case Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3 | | |
| Use Case Name: | ViewNowShowingInfo | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | User to view detailed information of the now showing movies |
| Preconditions: | The user must have login to the system using the Login use case |
| Postconditions: | The system must display detailed information of the now showing movies depending on user’s query |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user chooses to view now showing movies 2. The system retrieves all now showing movies and display all available movie titles to user 3. The user query movie information for now showing movies by supplying a movie title or supplying a date or supplying both movie title and date. 4. The system displays the results according to user’s query |
| Alternative Flows: | AF-S4:   1. If the user supplies a movie title, the system retrieves movie information for the supplied movie title consisting of movie description, movie rating, upcoming showtimes, cinema name, ticket price and number of seats left. 2. The system proceeds to step 2   AF-S4:   1. If the user supplies a date and time, the system retrieves movie information for the supplied date. The movie information consist of all available movie titles for that along with movie description, movie ratings, movie exact timings, cinema names, ticket prices and number of seats left. 2. The system proceeds to step 2   AF-S4:   1. If the user supplies both a movie title and a date, the system retrieve movie information for the supplied movie title and supplied date. The movie information consist of the movie description, movie rating, exact timings, cinema names, ticket prices and number of seats left. 2. The system proceeds to step 2 |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**Use Case Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 4 | | |
| Use Case Name: | ViewUpcomingInfo | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | User to view detailed information of the upcoming movies |
| Preconditions: | The user must have login to the system using the Login use case |
| Postconditions: | The system must display detailed information of the upcoming movies depending on user’s query |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The user chooses to view upcoming movies in UI 2. The system retrieves all upcoming movies and displays all upcoming movie titles to user 3. The user query movie information for upcoming movies by supplying a movie title 4. The system displays the results according to user’s query, consisting of movie description and release date. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**Use Case Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 5 | | |
| Use Case Name: | BookMovieTicket | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | This functionality allows user to book movie tickets from official cinema webpage |
| Preconditions: | This use case extends ViewNowShowingInfo use case. It is invoked whenever the user decides to book movie tickets after viewing the showtimes. |
| Postconditions: | The system must direct users to the booking page of the official cinema website |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. The system displays detailed information of the now showing movies as in ViewNowShowingInfo use case. 2. The user press book ticket for his corresponding choice of movie showtime and cinema. 3. System directs user to the booking page of the official cinema website |
| Alternative Flows: | AF-S2 :   1. If there are no seats left, the system must now allow user to click the book ticket button and displays message “No seats left, please choose other timing” 2. The system returns to step 1 |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**Use Case Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 6 | | |
| Use Case Name: | ViewPromotions | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | This functionality allows user to view the current promotions of movie(s) |
| Preconditions: | The user must have login to the system using the Login use case |
| Postconditions: | System displays all available promotions to user |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. User click on the view promotion tab in UI 2. System should display all the promotions that is going on for this month |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**Use Case Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 7 | | |
| Use Case Name: | ViewSupport | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | This functionality allows user to view help/support regarding how to use the app |
| Preconditions: | The user must have login to the system using the Login use case |
| Postconditions: | System displays help/support information to user |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. User click on the view support tab in UI 2. System should display help/support messages to user |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**Use Case Template**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 8 | | |
| Use Case Name: | GiveRating | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | This functionality allows user to give ratings for the application |
| Preconditions: | The user must have login to the system using the Login use case |
| Postconditions: | System must store ratings given by the user |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. User click on the give ratings button in UI 2. The system displays rating options from scale 1-5 3. The user inputs his rating 4. System stores ratings given by user |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**Use Case Template**

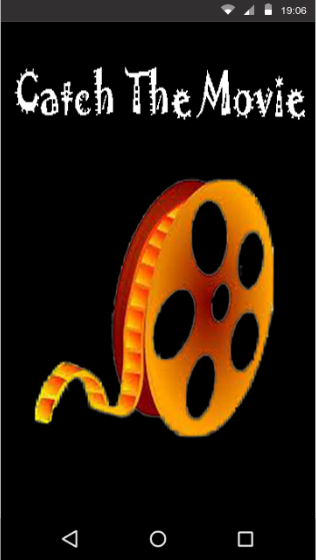
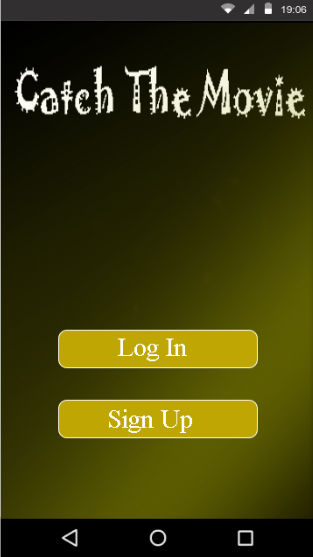
|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 9 | | |
| Use Case Name: | GiveFeedback | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

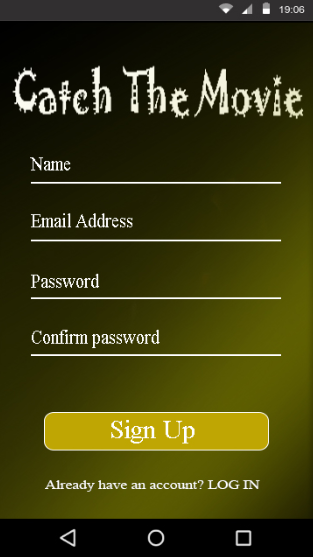
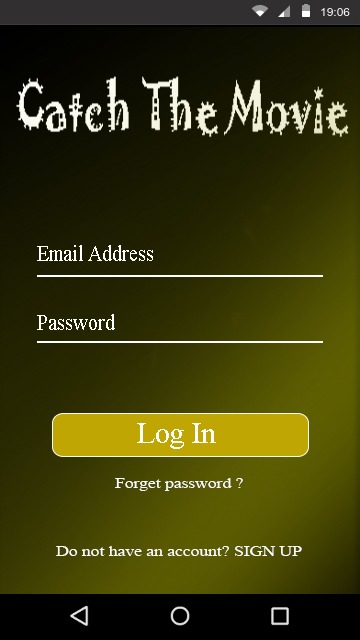
|  |  |
| --- | --- |
| Actor: | User |
| Description: | This functionality allows user to provide feedback about the app to developer team |
| Preconditions: | The user must have login to the system using the Login use case |
| Postconditions: | System stores user’s feedback |
| Priority: |  |
| Frequency of Use: |  |
| Flow of Events: | 1. User click on the give feedback tab in UI 2. System should display a text box for user to input their feedback of maximum 200 words. 3. System stores user feedback. |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

**UI MOCKUPS**

Login or Sign up?

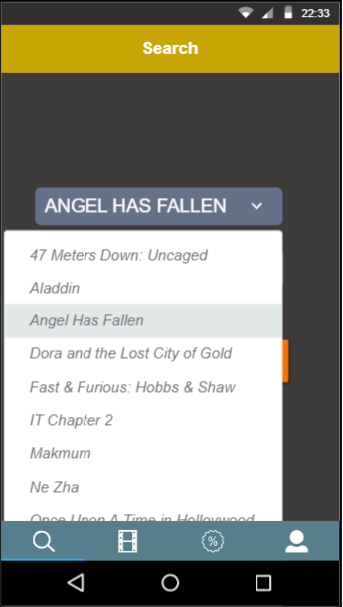
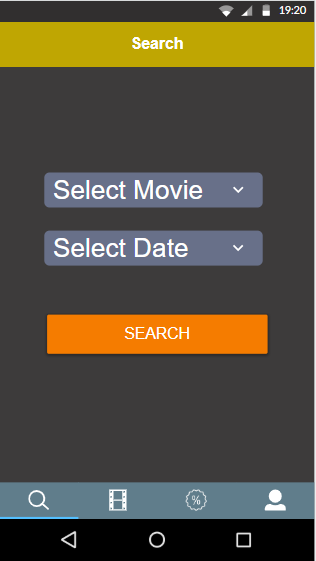
Loading Page:





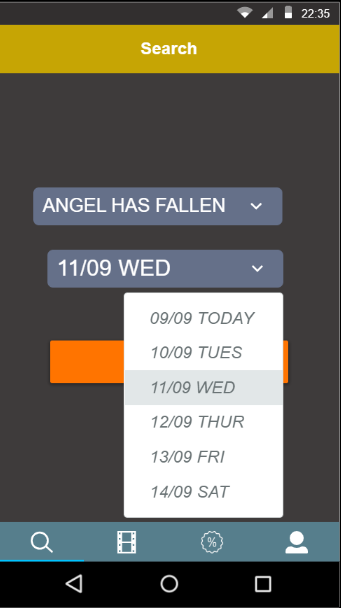
Sign up Page

Login Page

****

For example, user selects “Angel Has Fallen” as the movie

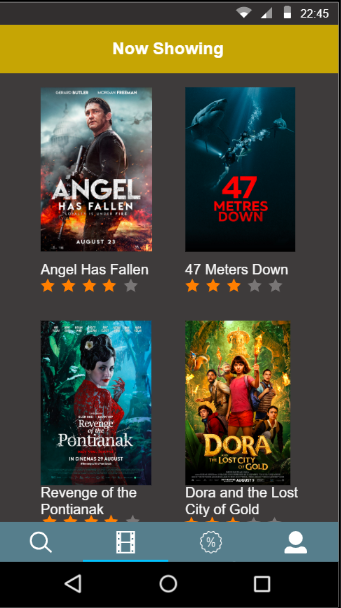
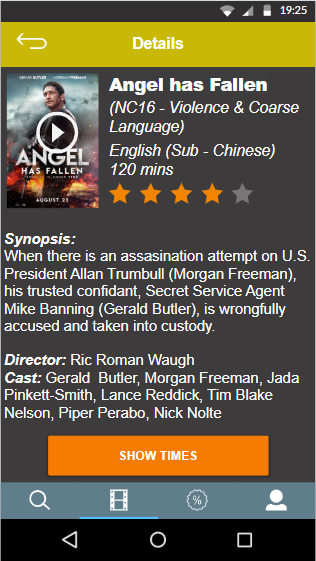
**Search Tab** – User is allowed to select Movie and Date



User selects date as “11/09 WED”

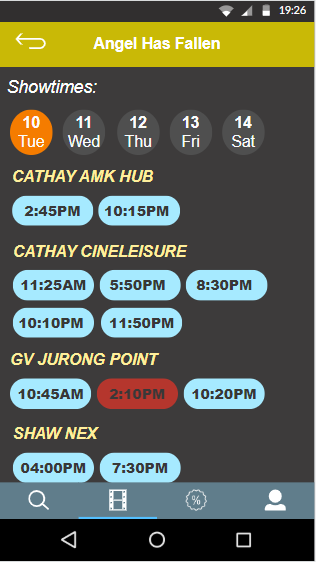
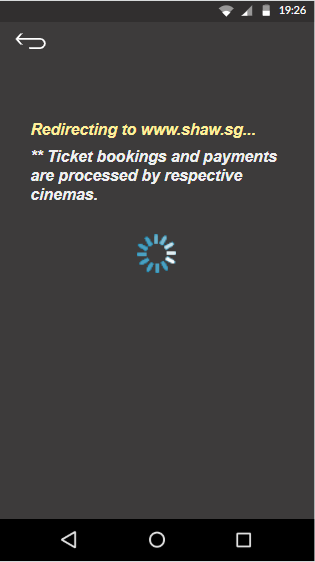
and press the “Search” button

The system brings the user to the selected movie, “Angel Has Fallen” and date on the 11th, Wed



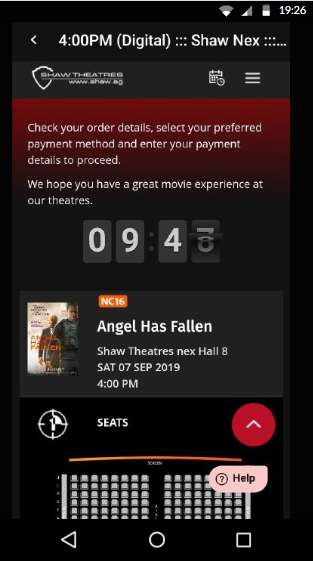
If user is to press on a particular movie, more details of the movies will be showed

**Movie Tab** will show “Now Showing” movies in a list form with the ratings of the movie

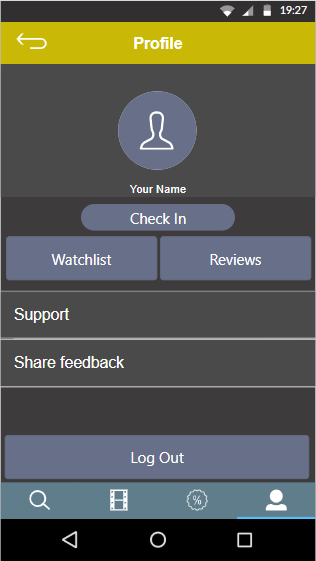
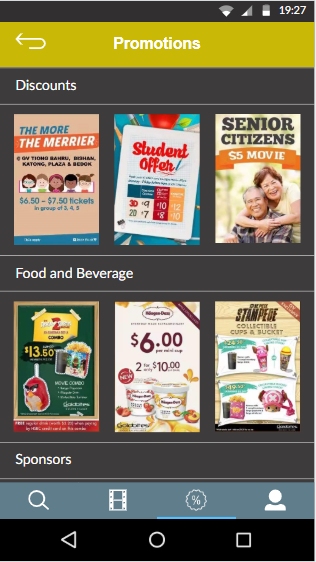


If user decides to purchase by clicking a timeslot at the previous page, they will be redirected to the respective cinema

If user is to click, “Show Times” button, different time slows will be shown according to respective cinemas

****

This is an example of a user choosing to purchase their ticket from Shaw



**Promotion tab**

**Profile tab**