Requirements Engineering

CinemaSYS

Submitted By: Kieran Stack (T-00184968)

Computing with Software Development

Date Submitted: 12/04/2019

**Table of Contents**

[1. Introduction/overview 4](#_Toc531329522)

[2. Functional Components 5](#_Toc531329523)

[3. User Requirements 6](#_Toc531329524)

[3.1 CinemaSYS will perform Screen management. 6](#_Toc531329525)

[3.1.1 CinemaSYS will add a screen 6](#_Toc531329526)

[3.1.2 CinemaSYS will change the details of a screen 6](#_Toc531329527)

[3.1.3 CinemaSYS will remove a screen 6](#_Toc531329528)

[3.2 CinemaSYS will manage movies. 6](#_Toc531329529)

[3.2.1 CinemaSYS will add a new movie 6](#_Toc531329530)

[3.2.2 CinemaSYS will change the details of a movie 6](#_Toc531329531)

[3.2.3 CinemaSYS will remove an old movie 6](#_Toc531329532)

[3.3 CinemaSYS will manage screenings. 6](#_Toc531329533)

[3.3.1 CinemaSYS will schedule a screening 6](#_Toc531329534)

[3.3.2 CinemaSYS will change the details of a screening 6](#_Toc531329535)

[3.3.3 CinemaSYS will delete an old screening 6](#_Toc531329536)

[3.4 CinemaSYS will manage sales. 6](#_Toc531329537)

[3.4.1 CinemaSYS will record seat tickets sold 6](#_Toc531329538)

[4. System Requirements 7](#_Toc531329539)

[4.1. System Level Use Case Diagram 7](#_Toc531329540)

[4.2. Manage Screen 8](#_Toc531329541)

[**4.2.1.** **Add Screen** 8](#_Toc531329542)

[**4.2.2.** **Change Screen** 11](#_Toc531329543)

[**4.2.3.** **Change Screen Status** 14](#_Toc531329544)

[4.3. Manage Movie 17](#_Toc531329545)

[**4.3.1.** **Add Movie** 17](#_Toc531329546)

[**4.3.2.** **Change Movie** 20](#_Toc531329547)

[**4.3.3.** **Remove Movie** 23](#_Toc531329548)

[4.4. Manage Screening 26](#_Toc531329549)

[**4.4.1.** **Schedule Screening** 26](#_Toc531329550)

[**4.4.2.** **Change Screening** 29](#_Toc531329551)

[**4.4.3.** **Remove Screening** 32](#_Toc531329552)

[4.5. Sales 35](#_Toc531329553)

[**4.5.1.** **Seat Tickets** 35](#_Toc531329554)

[**4.5.2.** **Yearly Revenue by Month** 38](#_Toc531329555)

[5. System Model 42](#_Toc531329556)

[5.1. Level-0 DFD 42](#_Toc531329557)

[5.2. Level-1 DFD 42](#_Toc531329558)

[5.3. Level-2 DFD (Process P1: Title) 42](#_Toc531329559)

[5.4. Level-2 DFD (Process P2: Title) 42](#_Toc531329560)

[5.5. Level-2 DFD (Process P3: Title) 42](#_Toc531329561)

[6. Data Model (Class Diagram) 43](#_Toc531329562)

[6.1. Class Diagram 43](#_Toc531329563)

[6.2. Relational Schema 43](#_Toc531329564)

[6.3. Database Schema 43](#_Toc531329565)

[7. Conclusion 44](#_Toc531329566)

[8. Appendices 45](#_Toc531329567)

[8.1. Appendix A – Title 45](#_Toc531329568)

[8.2. Appendix B – Title 45](#_Toc531329569)

# Introduction

This document thoroughly details as well as explains, the design process and requirements for the proposed cinema system. In this document the overall system is detailed using a System Model, Data Schema and User Requirements. The document also demonstrates all functionality of the system including Manage Movies, Manage Screens, Schedule Screenings and Manage Sales which will be designed for ease of use for the staff interacting with the system.

A prototype of the system described in the document will also be included for clarity on what the completed system will look like.

# Functional Components

This section of the document shows the functional components of the software system.

There are 4 main components in the proposed system.

# User Requirements

This section presents the user requirements as a set of abstract statements.

## CinemaSYS will perform Screen management.

### CinemaSYS will add a screen

### CinemaSYS will change the details of a screen

### CinemaSYS will remove a screen

## CinemaSYS will manage movies.

### CinemaSYS will add a new movie

### CinemaSYS will change the details of a movie

### CinemaSYS will remove an old movie

## CinemaSYS will manage screenings.

### CinemaSYS will schedule a screening

### CinemaSYS will change the details of a screening

### CinemaSYS will delete an old screening

## CinemaSYS will manage sales.

### CinemaSYS will record seat tickets sold

# System Requirements

The system is broken to four main components.

The screen management component will be able to add a screen and can change a screen.

## System Level Use Case Diagram

A brief introduction of the high-level modules/components of the system.

**CINEMA SYSYEM**

**Manage Screens**

Manager

**Manage Movies**

**Manage Screenings**

Customer

Staff

**Manage Sales**

## Manage Screen

This component provides, functions to add, change and remove a screen.

### **Add Screen**

This function will allow the manager to add a screen to the Screen file.

Manager



|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add Screen** | |
| **Use Case Id** | S1 | |
| **Priority** | 1 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | A new screen is added, which requires the manager to add the details of the screen. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**The manager invokes the Add Screen function    **Step 5:** The manager enters the screen details:   * Number of Seats * Screen Type   **Step 6:** The manager confirms that the screen is to be set | **Step 2:**The system determines the next ScreenNo  **Step 3:** The system retrieves details of all screen types from the Screen Type file and loads on the UI  **Step 4:** The system displays the UI      **Step 7:** The system validates the data entered:   * All fields must be entered * Number of seats must be numeric and in range 100-200   **Step 8:** System sets screen status to ‘Available’  **Step 9:** The system saves the Screen details in the Screen File:   * ScreenNo * Number of Seats * Screen Type * Status   **Step 10:** The system displays a confirmation message  **Step 11:** The system clears the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| Invalid Data Found |  | **Step 6:** Invalid data is entered  **Step 7:** The System displays an appropriate error message |
| **Conclusions** | A new screen is created and added to the Screen File. | |
| **Post conditions** | A film may now be screened in this screen. | |
| **Business Rules** | Seats per screen must be set between 100 and 200 | |
| **Implementation Constraints** |  | |

### **Change Screen**

This function will allow the manager to change a screen in the Screen file.

Manager



|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Change Screen** | |
| **Use Case Id** | S2 | |
| **Priority** | 2 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | A screen is changed, which requires the manager to change details of the screen. | |
| **Preconditions** | A Screen must already be registered on the System.  The Screen Status must be set to ‘Removed’. | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**The manager invokes the Change Screen function      **Step 4:** The manager selects the screen to change    **Step 7:** The manager makes the required changes:   * Number of Seats * Screen Type   **Step 8:** The manager confirms that the screen details are to be edited | **Step2:** The system retrieves a summary of details of all available screens from the Screen File in order of ScreenNo on UI  **Step 3:**The system displays the UI    **Step 5:** The system checks to make sure Screens status is set to ‘Removed’  **Step 6:** Details of the selected screen are retrieved from the screen file and displayed on the UI for changing    **Step 9:** The system validates the data entered:   * All fields must be entered * Number of seats must be numeric and in range 100-200   **Step 10:** The system updates the Screen details in the Screen File:   * Number of Seats * Screen Type   **Step 11:** The system displays a confirmation message  **Step 12:** The system clears the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| Invalid Data Found |  | **Step 4:** System returns error, only screens with status set to ‘Removed’ may be changed  **Step 8:** Invalid data is entered  **Step 9:** The System displays an appropriate error message |
| **Conclusions** | A screen is updated to the Screen File. | |
| **Post conditions** |  | |
| **Business Rules** | Only an available screen can be updated. | |
| **Implementation Constraints** |  | |

### **Change Screen Status**

This function will allow the manager to change a screen status from ‘Available’ to ‘Removed’ or from ‘Removed’ to ‘Available’ in the Screen Status file.

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Change Screen** **Status** | |
| **Use Case Id** | S3 | |
| **Priority** | 3 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | A Screen Status is changed, which requires the manager to confirm the change of the Screen Status. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**The manager invokes the Change Screen Status function      **Step 4:** The manager selects the Screen Status to be changed  **Step 5:** The manager confirms that the Screen Status is to be changed | **Step 2:**The system retrieves a summary of details of all screens from the Screen Status File in order of ScreenNo on the UI  **Step 3:** The system displays the UI  **Step 6:** Details of the selected Screen Status are retrieved from the Screen Status File:   * ScreenNo * Status   **Step 7:** System checks to see if any screening is scheduled for screen selected  **Step 8:** System saves screen status from ‘Available’ to ‘Removed’ or ‘Removed’ to ‘Available’  **Step 9:** The system displays a confirmation message  **Step 10:** The system clears the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| Invalid Data |  | **Step 5:**  The system outputs an appropriate error message |
| **Conclusions** | A screen status is changed in the Screen Status file. | |
| **Post conditions** | The screen may now be available or may be removed. | |
| **Business Rules** |  | |
| **Implementation Constraints** | Screen Status may only be changed to ‘Removed’ if the screen selected has no screenings currently on going. | |

## Manage Movie

This component provides the functions to add, change and remove a movie.

### **Add Movie**

This function will allow the manager to add a movie to the Movie file.

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Add Movie** | |
| **Use Case Id** | M1 | |
| **Priority** | 1 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | A new movie is added, which requires the manager to add the details of the movie. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**The manager invokes the Add Movie function    **Step 5:** The manager enters the movie details:   * Name * Running Time * Description * Age Rating   **Step 6:** The manager confirms the movie details | **Step 2:**The system determines the next MovieID  **Step 3:** The system retrieves details of all age ratings from the Age Ratings File and loads on UI  **Step 4:** The system displays the UI        **Step 7:** The system validates the data entered:   * All fields must be entered * Running Time must be a numeric value   **Step 8:** The system updates the movie details in the Movie File:   * MovieID * Name * Running Time * Description * AgeRatingID   **Step 9:** The system updates the Age Ratings File  **Step 9:** The system displays a confirmation message  **Step 10:** The system clears the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| Invalid Data Found |  | **Step 6:** Invalid data is entered  **Step 7:** The System displays an appropriate error message |
| **Conclusions** | A new movie is created and added to the Movie File. | |
| **Post conditions** | The movie may now be scheduled. | |
| **Business Rules** | Each movie must have an assigned age rating. | |
| **Implementation Constraints** |  | |

### **Change Movie**

This function will allow the manager to add a movie to the Movie file.

Manager



|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Change Movie** | |
| **Use Case Id** | M2 | |
| **Priority** | 2 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | A movie is changed, which requires the manager to edit the details of the movie. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**The manager invokes the Change Movie function.    **Step 3:** The manager searches the movie they wish edit by name  **Step 5:** The manager selects the movie to edit  **Step 6:** The manager makes the required changes   * Name * Running Time * Description * Age Rating   **Step 8:** The manager confirms that the movie details are to be edited | **Step 2:** The system displays the UI    **Step 4:** The system displays the movie matching the search onto the UI  **Step 7:** Details of the selected movie are retrieved from the movie file and displayed on the UI      **Step 9:** The system validates the data entered:   * All fields must be entered * Running Time must be a numeric value   **Step 10:** The system updates the new Movie details in the Movie File   * Name * Running Time * Description * Age Rating   **Step 11:** The system displays a confirmation message  **Step 11:** The system clears the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| Invalid Data Found |  | **Step 3:** Invalid data is entered  **Step 4:** The System displays an appropriate error message  **Step 8:** Invalid data is entered  **Step 9:** The System displays an appropriate error message |
| **Conclusions** | A movie is updated to the Movie File. | |
| **Post conditions** | The movie may now be scheduled with the correct details. | |
| **Business Rules** | Each movie must have an age rating listed. | |
| **Implementation Constraints** |  | |

### **Remove Movie**

This function will allow the manager to remove a movie in the Movie file.

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Remove Movie** | |
| **Use Case Id** | S3 | |
| **Priority** | 3 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | A movie is removed, which requires the manager to confirm the removal of the movie. | |
| **Preconditions** | A Movie must already be registered on the System.  A Screening must not be scheduled with the movie that will be removed. | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**The manager invokes the Remove Movie function.    **Step 3:** The manager searches the movie to remove  **Step 5:** The manager confirms that the movie is to be removed | **Step 2:** The system displays the UI  **Step 4:** The system displays the movie matching the search onto the UI  **Step 6:** Details of the selected movie are retrieved from the Movie file   * MovieID * Name * Running Time * Description * Age Rating   **Step 7:** System checks to see if any screening is scheduled for movie selected  **Step 8:** The system deletes the selected movie from the Movie File  **Step 9:** The system displays a confirmation message  **Step 10:** The system clears the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| Invalid Data Found |  | **Step 2:** The System fails to retrieve any movie details from the Movie File  **Step 3:** The system alerts the manager that there are no movie records to be removed  **Step 5:**  The system outputs an appropriate error message |
| **Conclusions** | A movie is deleted from the Movie File. | |
| **Post conditions** | The movie cannot be used anymore as it has been deleted. | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

## Manage Screening

### **Schedule Screening**

This function will allow the manager to schedule a screening to the Screening File.

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Schedule Screening** | |
| **Use Case Id** | Sc1 | |
| **Priority** | 1 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | A new screening is added, which requires the manager to add the details of the screening. | |
| **Preconditions** | The movie that will be shown must already be registered on the system  The screen that will be used must already be registered on the system | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**The manager invokes the Schedule Screening function.      **Step 5:** The manager selects the movie he wants to set up a screening for  **Step 6:** The manager selects the screen he wants to set up a screening for  **Step 7:** The manager selects the screen he wants to manage  **Step 8:** The manager enters the screening details:   * Date From * Date To * Time * ScreenNo * Movie Id     **Step 10:** The manager confirms the screening details | **Step 2:** The system determines the next screening number  **Step 3:** The system retrieves the summary of details of all available screens from the Screen File  **Step 4:** The system displays the UI            **Step 9:** The system validates the data entered:   * All fields must be entered * Maximum seats will be set equal to Seat amount referenced from number of seats connected to ScreenNo   **Step 11:** The system saves the screening details in the Screening File.   * ScreeningNumber * Date From * Date To * Time * MovieID * ScreenNo * MaxSeats   **Step 12:** The system displays a confirmation message  **Step 13:** The system clears the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| Invalid Data Found |  | **Step 8:** Invalid data is entered.  **Step 9:** The System displays an appropriate error message. |
| **Conclusions** | A new screening is created and added to the Screening File. | |
| **Post conditions** | The Screening has now been scheduled. | |
| **Business Rules** | Each screening can only book up to the screen’s maximum capacity.  Screens have a pre-set times of 12:00, 14:00, 18:00, 21:00, 23:00. | |
| **Implementation Constraints** |  | |

### **Change Screening**

This function will allow the manager to change a screening in the Screening file.

Manager



|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Change Screening** | |
| **Use Case Id** | Sc2 | |
| **Priority** | 2 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | A screening is changed, which requires the manager to edit details of the screening. | |
| **Preconditions** | A Screening must already be registered on the System. | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**The manager invokes the Change Screening function      **Step 4:** The manager selects the screening he wants to edit  **Step 6:** The manager makes the required change:   * Date From * Date To * Time * Movie * Screen   **Step 7:** The manager confirms that the screening details are to be edited. | **Step 2:**The system retrieves a summary of details of all Screenings from the Screening File  **Step 3:** The system displays the records in the Screening File by Screening number  **Step 5:** Details of the selected screening are retrieved from the Screening File and displayed on the UI:   * ScreeningNumber * Date From * Date To * Time * Movie * Screen   **Step 8:** The system validates the data entered:   * All fields must be entered * Screenings must not conflict   **Step 9:** The system saves the new Screening details in the Screening File  **Step 10:** The system displays a confirmation message  **Step 11:** The system clears the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| Invalid Data Found |  | **Step 7:** Invalid data is entered.  **Step 8:** The System displays an appropriate error message |
| **Conclusions** | A screening is updated to the Screening File. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### **Remove Screening**

This function will allow the manager to remove a screening in the Screening file.

Manager

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Remove Screening** | |
| **Use Case Id** | Sc3 | |
| **Priority** | 3 | |
| **Source** | Manager | |
| **Primary Business Actor** | Manager | |
| **Other Participating Actors** |  | |
| **Description** | A screening is removed, which requires the manager to confirm the removal of the screening | |
| **Preconditions** | A Screening must already be registered on the System. | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:**The manager invokes the Remove Screening function.    **Step 4:** The manager selects the screening to remove.  **Step 5:** The manager confirms that the screening is to be removed. | **Step 2:**The system retrieves a summary of details of all screenings from the Screening File  **Step 3:** The system displays the records in the Screening File in an on the UI.      **Step 6:** Details of the selected screening are retrieved from the Screening File  **Step 7:** The system deletes the selected screening from the Screening File.   * ScreeningNumber * Date From * Date To * Time * MovieID * ScreenNo   **Step 8:** The system displays a confirmation message |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  | **Step 2:** The System fails to retrieve any Screen details from the Screen file  **Step 3:** The system alerts the manager that there are no screening records to be removed.  **Step 5:**  The system outputs an appropriate error message |
| **Conclusions** | A screening is deleted from the Screening File. | |
| **Post conditions** | The screening cannot be used anymore as it has been deleted. | |
| **Business Rules** | Cannot be removed if tickets are book | |
| **Implementation Constraints** |  | |

## Sales

### **Make Sale**

This function will allow the staff to record booked tickets for the customer.

Staff

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Make Sale** | |
| **Use Case Id** | Sa1 | |
| **Priority** | 1 | |
| **Source** | Staff | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** | Customer | |
| **Description** | A seat will be booked. | |
| **Preconditions** | The screening must already be scheduled. | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The staff invoke the Sales function      **Step 6:** Staff select details   * Movie * Rate   **Step 7:** The staff confirms their choices | **Step 2:** The system determines the next SaleID  **Step 3:** The system retrieves details of all screenings from the Screenings File and loads on UI  **Step 4:** The system retrieves details of all Rates from the Rates File and loads on UI  **Step 5:** The system displays the UI      **Step 8:** The system saves the sales in the Sales File.   * Movie * Rate   **Step 9:** The system reduces the amount of tickets available for the screening by subtracting the amount of tickets booked from MaxSeats in the Screening file  **Step 10:** The system displays a confirmation message  **Step 11:** The system clears the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
| Invalid Data Found |  | **Step 7:** Invalid data is entered.  **Step 8:** The System displays an appropriate error message |
| **Conclusions** | A new sale is completed and added to the Sales File. | |
| **Post conditions** | The Sale is now recorded. | |
| **Business Rules** | Each screen cannot exceed its maximum capacity.  Each rate is as specified:  Standard Rate - 10.00  Children’s Rate - 5.00  Student Rate - 8.00  Senior Rate - 6.00 | |
| **Implementation Constraints** |  | |

### **Yearly Revenue by Month**

This function will allow the business to record the yearly revenue by month.

Staff

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Yearly Revenue by Month** | |
| **Use Case Id** | Sa2 | |
| **Priority** | 2 | |
| **Source** | Staff | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | A yearly revenue by month will be generated. | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The staff invoke the Generate Yearly Revenue function  **Step 4:** The staff print the analysis | **Step 2:** The System loads the yearly revenue analysis on the UI  **Step 3:** The system displays the UI  **Step 5:** The system prints the analysis  **Step 6:** The system clears the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The staff views/prints the yearly revenue analysis. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

### **Sold Out Screenings**

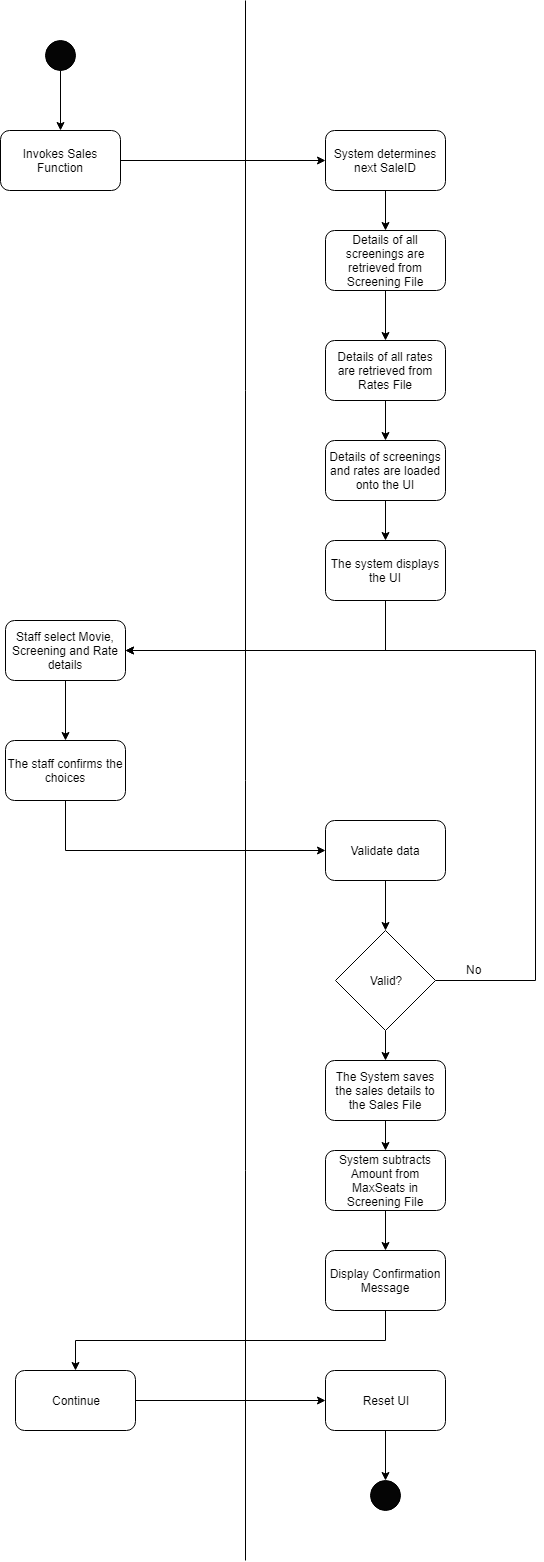
This function will allow the business to track the amount of sold out screenings.

Staff

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | **Sold-Out Screens** | |
| **Use Case Id** | Sa3 | |
| **Priority** | 3 | |
| **Source** | Staff | |
| **Primary Business Actor** | Staff | |
| **Other Participating Actors** |  | |
| **Description** | A weekly list of the sold-out screenings | |
| **Preconditions** |  | |
| **Trigger** |  | |
| **Expected Scenario** | **Actor Action** | **System Response** |
|  | **Step 1:** The staff invoke the Sold-Out Screenings function    **Step 4:** The staff print the analysis | **Step 2:** The System retrieves all weekly screenings from the Screening file  **Step 3:** The system retrieves the screens that are sold out  **Step 4:** The system displays the UI  **Step 6:** The system clears the UI |
| **Alternate Scenarios** | **Actor Action** | **System Response** |
|  |  |  |
| **Conclusions** | The staff views/prints a list of the Sold-Out Screenings. | |
| **Post conditions** |  | |
| **Business Rules** |  | |
| **Implementation Constraints** |  | |

**Activity Diagram**

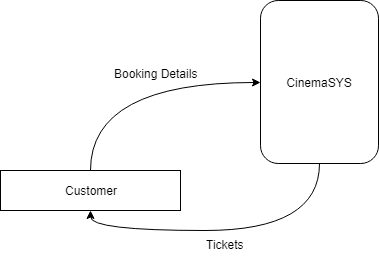
|  |  |
| --- | --- |
| **Staff** | **Cinema**  **System** |



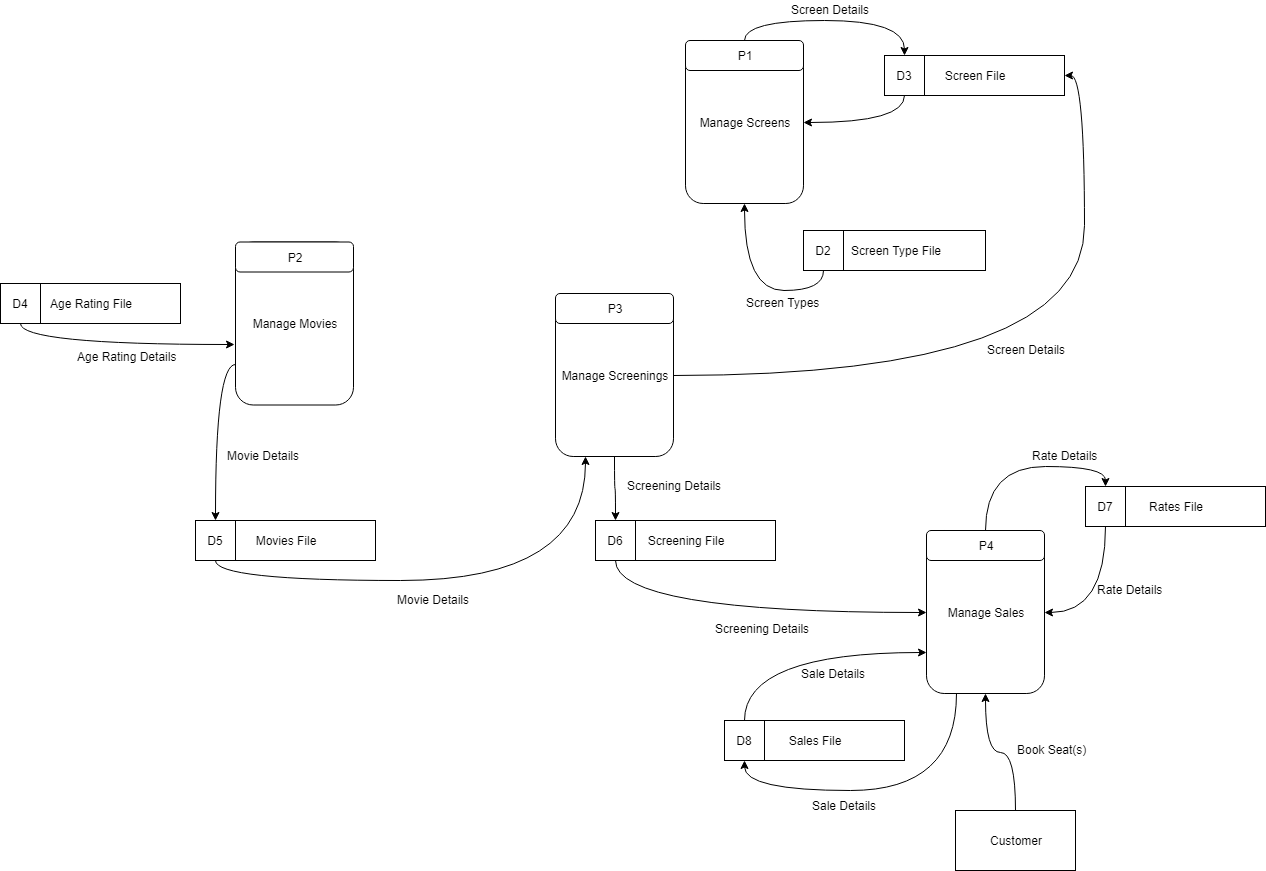
# System Model

The following dataflow diagrams have been produced for the system:

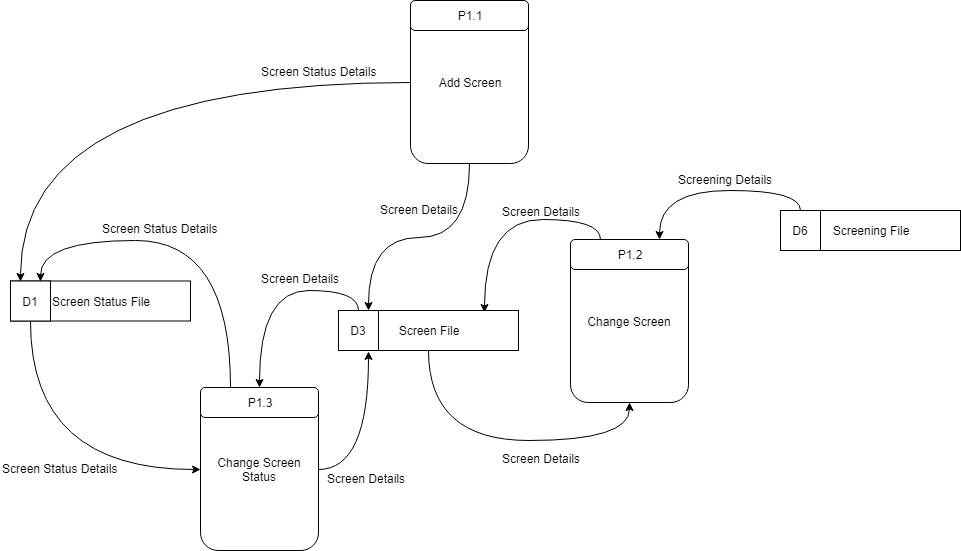
## Level-0 DFD



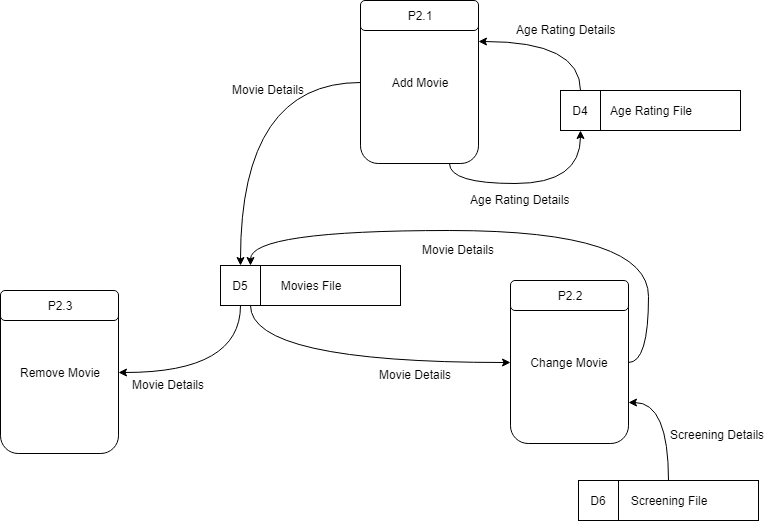
## Level-1 DFD



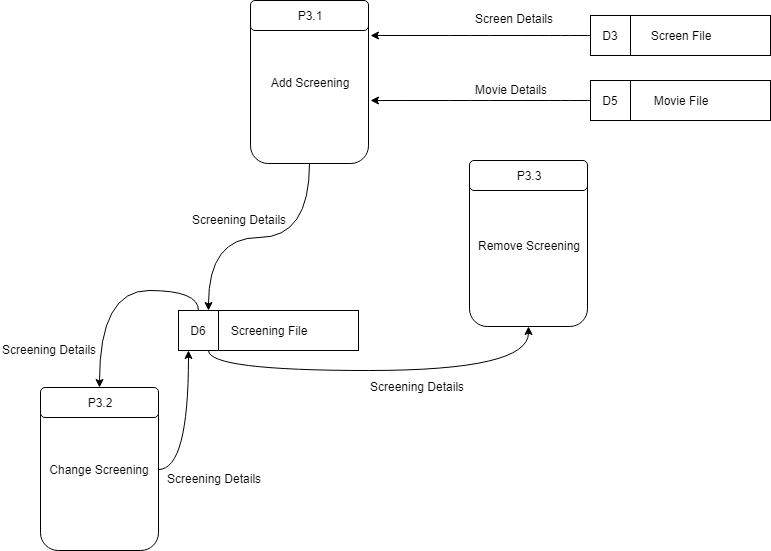
## Level-2 DFD (Process P1: Screens)



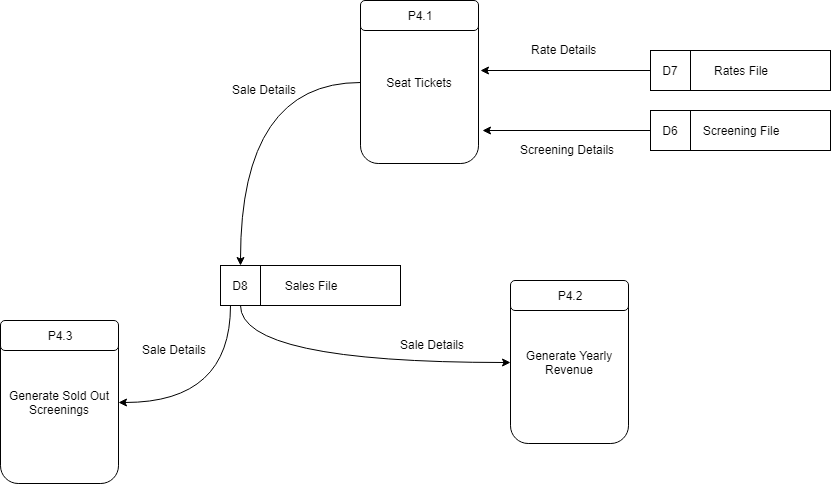
## Level-2 DFD (Process P2: Movies)



## Level-2 DFD (Process P3: Screenings)



## Level-2 DFD (Process P3: Sales)

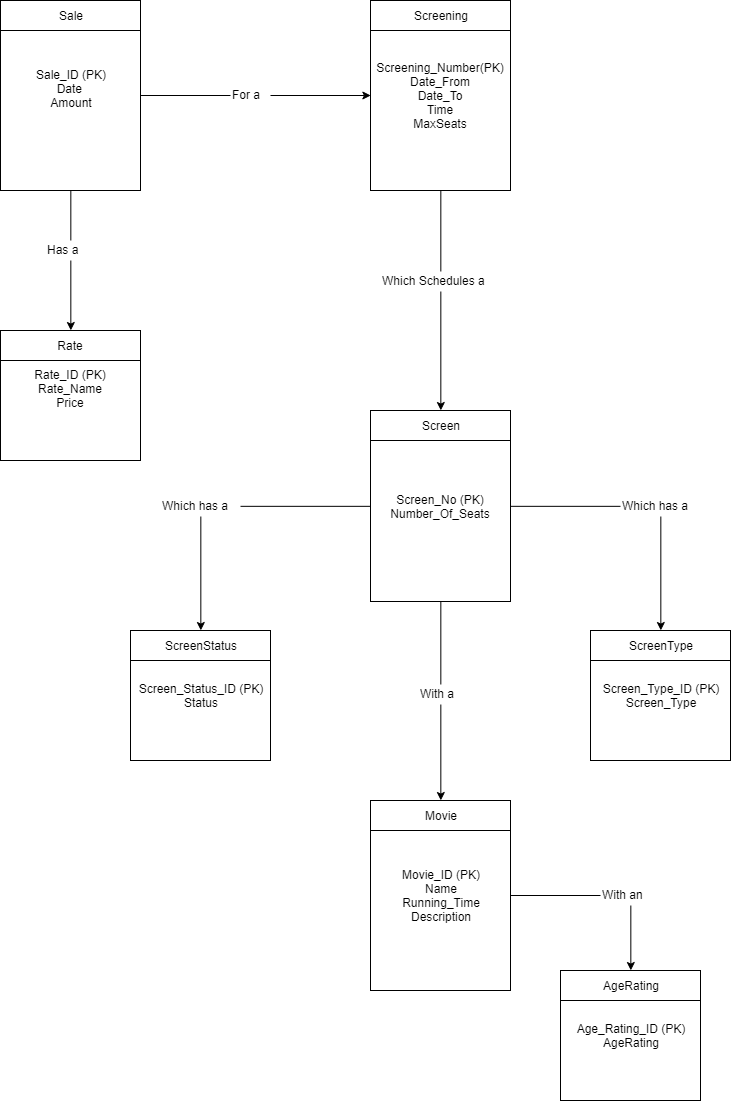


# Data Model (Class Diagram)

The following data models have been produced for the system.

## Class Diagram

Class diagram shows objects & attributes



## Relational Schema

AgeRating {Age\_Rating\_ID, Age\_Rating}

Movie {Movie ID, Name, Running\_Time, Description, Age\_Rating\_ID}

Rate {Rate\_ID, Rate\_Name, Rate, Description}

Sale {Sale\_ID, Date, Amount, Rate\_ID, Screening\_Number}

Screen {Screen\_No, Number\_Of\_Seats, Screen\_Status\_ID, Screen\_Type\_ID}

Screening {Screening\_Number, Date\_From, Date\_To, Time, Max\_Seats, Movie\_ID, Screen\_ID}

ScreeningTImes {Screening\_Time\_ID, Screening\_Time}

ScreenStatus {Screen\_Status\_ID, Status, Description}

ScreenType {Screen\_Type\_ID, Screen\_Type}

## Database Schema

A definition of the database to be implemented.

This includes primary key, foreign key and other constraints to be implemented.

**Schema: CinemaSYS**

**Relation:** AgeRating

Attributes:

Age\_Rating\_ID numeric (2) UNIQUE NOT NULL

Age\_Rating char (3) NOT NULL

**Primary Key:** Age\_Rating\_ID

**Relation:** Movie

Attributes:

Movie\_ID numeric UNIQUE NOT NULL

Name char (20) NOT NULL

Running\_Time time NOT NULL

Description char (280) NOT NULL

Age\_Rating\_ID numeric (2) NOT NULL

**Primary Key:** Movie\_ID

**Foreign Key:** Age\_Rating\_ID REFRENCING AgeRating [Age\_Rating\_ID]

**Relation:** Rate

Attributes:

Rate\_ID numeric (2) UNIQUE NOT NULL

Rate\_Name char (20) NOT NULL

Rate numeric (2) NOT NULL

**Primary Key:** Rate\_ID

**Relation:** Sale

Attributes:

Sale\_ID numeric (2) UNIQUE NOT NULL

Date date NOT NULL

Amount numeric (3) NOT NULL

Rate\_ID numeric NOT NULL

Screening\_Number numeric (2) NOT NULL

**Primary Key:** Sale\_ID

**Foreign Key:** Rate\_ID REFRENCING Rate [Rate\_ID]

Screening\_Number REFRENCING Screening [Screening\_Number]

**Relation:** Screen

Attributes:

Screen\_No numeric (2) UNIQUE NOT NULL

Number\_Of\_Seats numeric (3) NOT NULL

Screen\_Status\_ID numeric (2) NOT NULL

Screen\_Type\_ID numeric (2) NOT NULL

**Primary Key:** Screen\_No

**Foreign Key:** Screen\_Status\_ID REFRENCING ScreenStatus [Screen\_Status\_ID]

Screen\_Type\_ID REFRENCING ScreenType [Screen\_Type\_ID]

**Relation:** Screening

Attributes:

Screening\_Number numeric (2) UNIQUE NOT NULL

Date\_From date NOT NULL

Date\_To date NOT NULL

Time time NOT NULL

Max\_Seats numeric (3) NOT NULL

Movie\_ID numeric (3) NOT NULL

Screen\_No numeric (2) Not NULL

**Primary Key:** Screening\_Number

**Foreign Key:** Movie\_ID REFRENCING Movie [Movie\_ID]

Screen\_No REFRENCING Screen [Screen\_No]

**Relation:** Screening\_TImes

Attributes:

Screen\_Time\_ID numeric (1) UNIQUE NOT NULL

Status Time (1) VARCHER(8 BYTE)

**Primary Key:** Screen\_Time\_ID

**Relation:** ScreenStatus

Attributes:

Screen\_Status\_ID numeric (2) UNIQUE NOT NULL

Status char (1) NOT NULL

**Primary Key:** Screen\_Status\_ID

**Relation:** ScreenType

Attributes:

Screen\_Type\_ID numeric (2) UNIQUE NOT NULL

Screen\_Type char (10) NOT NULL

**Primary Key:** Screen\_Status\_ID

# Conclusion

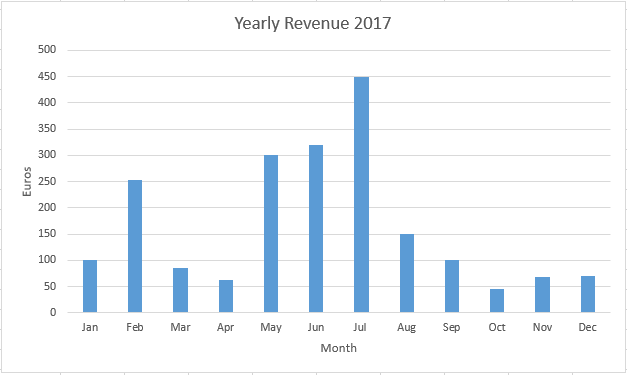
Every element of the System has been clearly outlined in this document through the use of User Case Diagrams, User Case Narratives, System Model and the Database Schema. Overall, I think this system should fulfil all requirement’s which are needed by the cinema.

The only other recommendation I would make to improve the system would be to add a Customer function so that repeat customers could sign up for a premium discount service which rewards their patronage and incentivises return visits.

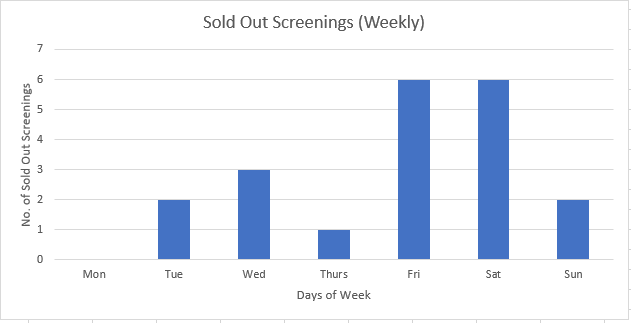
# Appendices

## Appendix A – Sample Reports

Yearly Revenue by Month



Weekly Sold Out Screenings



## Appendix B – Title

Might include:

* **Lookup / Reference tables**