***Definition:***

Nowadays, people are very busy that they don’t have spare time for entertainment. It’s really difficult to find out the various entertainments such as movies, dramas, music, sports… and get the tickets directly at the tickets office.

“***Easytickets.com***” can be thought as the best solution for this. It helps customers view information about movies, shows, sports. And then, they can book the tickets online. It also helps company to manage information about user, movies, events. Now, company can sell tickets, advertise movies/shows/sports easily. Customers can get the tickets at home just a click.

***Solution Team***

**Client/Project Undertaken:** Showbiz Company.

1. **List of inputs to the system**
   * **User management**
     + Username.
     + Password.
     + Role.
   * **Customers**
     + Username.
     + Password.
     + Name.
     + Birthday.
     + Address.
     + Phone.
     + Email.

* **Cinema** 
  + Cinema name.
  + Cinema address.
  + Cinema phone.
  + Area name.
* **Place**
  + Place name.
  + Place address.
  + Place phone.
  + Seat row.
  + Area name.
* **Movie**
  + Movie name.
  + Genre.
  + Language.
  + Rated.
  + Director.
  + Cast.
  + Running time.
  + Ticket price.
  + Begin date.
  + End date.
  + Point.
  + Description.
  + Image.
  + Media.
* **Event**
  + Event name.
  + Event type.
  + Begin date.
  + Running time.
  + Ticket price.
  + Point.
  + Description.
  + Image.
* **Promotion**
  + Promotion name.
  + Discount.
  + Begin date.
  + End date.
  + Description.
* **Room**
  + Room name.
* **Time\_slot**
  + Time slot.

1. **List of outputs expected from the system**

* **Customers**
  + Display customer’s information.
  + Display movies | shows information.
  + Display searching information.
  + Display booking ticket information.
  + Display bill after booking ticket.
* **User management**
  + Display user’s information.
  + Display movies | shows information.
  + Display statistics.

1. **Overview of processes involved in the system**

* Admin login.
* Admin inserts new user.
* Admin changes user’s information.
* Admin inserts | updates | deletes | views movies | shows information.
* Customers login.
* Customers change | view their profile.
* Customers book the tickets.

1. **Hardware and software required for implementing the project**

* **Hardware**
  + CPU: Intel core i3 or higher.
  + Memory: 4 GB RAM or higher.
  + HDD: 20 GB.
* **Software**
  + Operation system: Windows 7 or higher.
  + Java EE 7 or higher.
  + JDK 1.7 or higher.
  + NetBeans 8.2 or higher.
  + Glassfish Server 4.1.
  + MS SQL Server 2012 or higher.

**Customer’s Acceptance Criteria.**

|  |  |
| --- | --- |
| **No.** | **Customer’s Acceptance Criteria** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |

**Project Plan**

1. **Project Details :**

* **Name of website:** easytickets.com
* **Date of Project Plan:** September 10, 2018
* **Project Vision/Objectives:**
* Easytickets.com helps the company to advertise, distribute tickets efficiently. It also helps customers to view or update their profile.
* The customers can view movies | shows | sports information conveniently. Hence, they can the tickets easily.
* **Scope :**
  + The customers can’t pay online with this website.
* **Our understanding of the client:**

Showbiz is the company is principally into ticket sales, marketing and distribution of event tickets. The services include providing information on multiple event categories, providing ticketing facilities for several leading multiplexes, arenas and stadiums.

* **Project Organization with Responsibilities Authorities:**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Members** | **Role** | **Responsibilities** |
| 1 | Nguyễn Hoàng Nhạc | Team leader | Managing, taking the request, analyzing, designing, coding and testing. |
| 2 | Nguyễn Phạm Bích Phượng | member | Taking the request, analyzing, designing, coding and testing. |
| 3 | Phạm Cát Tường | member | Taking the request, analyzing, designing, coding and testing. |
| 4 | Nguyễn Tấn Thông | member | Taking the request, analyzing, designing, coding and testing. |

1. **Project Initiation/Requirement Document :**

* CRS document.
* Design standards document that the customers require.

1. **Deliverables:**

* **Application:**
  + - Website source code.
    - Website database.
* **Document:**
  + Installation guide document.
* User guide document.
* Design document.
* Program document.
* Testing document.

1. **Project Dependencies:**
   * Membership’s skills is different.

* Limit research resources.
* Time rushed.

1. **Major Project Milestones**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Responsibilities** | **Schedule** | |
| **From** | **To** |
| 1 | Definition | Sep. 10, 2018 | Sep. 10, 2018 |
| 2 | Requirement analysis | Sep. 11, 2018 | Sep. 13, 2018 |
| 3 | Design | Sep. 16, 2018 | Sep. 22, 2018 |
| 4 | Coding | Sep 23, 2018 | Oct. 29, 2018 |
| 5 | Testing | Sep. 30, 2018 | Oct. 06, 2018 |
| 6 | Maintenance | Oct. 07, 2018 | Oct. 13, 2018 |

1. **Quality Plan:**

* **Review Activities:**Review project’s document and website application every 2 days.
* **Testing Activities:**All modules must be tested by cross-examination.
* **Backup and Recovery Strategies:**
  + Synchronize data on the member’s computer.
  + Update anti-virus application and scan virus regularly.
  + Make a data backup in Github, Google drive and hard drive**.**

**Document Design**

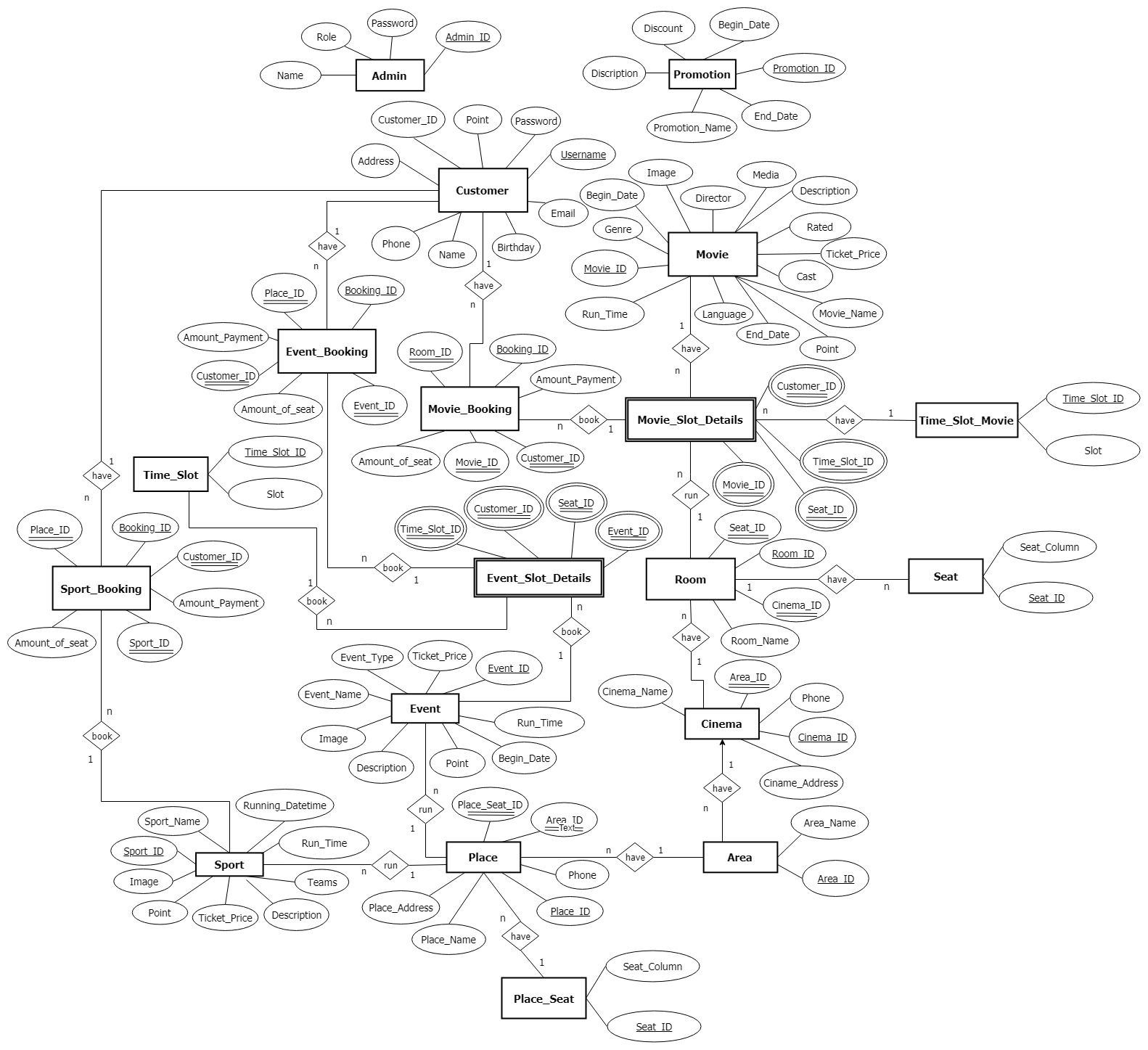
|  |  |
| --- | --- |
| **Property** | **Value** |
| Document theme and color scheme | Orange |
| Form-Background Color | White |
| Title-Font Size | 13px |
| Title-Font Color | Black |
| Title-Font Style | Bold |
| Title-Alignment | Center |
| Background Color of Controls on the form | Orange |
| Foreground Color of Controls the form | Orange |
| Control Caption-Font-Size | 13px |
| Control Caption-Font-Color | Black |
| Control Caption-Font-Style | Bold |
| Control caption and controls alignment | Left |
| Command button-Alignment | Center |

1. **List of forms to be created**

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Description** | **Controls on the Document** |
| **Add, Update,**  **Delete,**  **Event**  **tickets** | Admin add, update, delete event tickets |  |
| **Add, Update, Delete, Sport tickets** | Admin add, update, delete sport tickets |  |

|  |  |  |
| --- | --- | --- |
| **Add, Update,**  **Delete,**  **Promotion** | Admin add, update, delete Promotion |  |
| **Add Movie tickets** | Admin add movie ticket information |  |
| **Update movie tickets** | Admin update movie ticket information |  |

[**Entity Relationship Diagram (ERD)**](https://www.smartdraw.com/entity-relationship-diagram/):



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Admin** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Admin\_ID | INT | PK | No | Identity, id of admin |
| Name | NVARCHAR (50) |  | No | Administrator’s name |
| Password | NVARCHAR (50) |  | No | Account admin password |
| Role | NVARCHAR (50) |  | No | Administrator’s role |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Area** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Area\_ID | INT | PK | No | Identity, id of Area |
| Area\_Name | NVARCHAR (50) |  | No | Area’s name |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Cinema** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Cinema\_ID | INT | PK | No | Identity, id of Cinema |
| Cinema\_Name | NVARCHAR (50) |  | No | Cinema’s name |
| Cinema\_Address | NVARCHAR (500) |  | No | Address of Cinema |
| Phone | NVARCHAR (15) |  | No | Phone of Cinema |
| Area\_ID | INT | FK | No | Area of cinema |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Customer** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Customer\_ID | INT | PK | No | Identity, id of Customer |
| Username | NVARCHAR(50) |  | No | Account customer login |
| Password | NVARCHAR(100) |  | No | Customer’s password |
| Name | NVARCHAR(500) |  | No | Customer’s name |
| Birthday | DATE |  | No | Customer’s birthday |
| Email | NVARCHAR(50) |  | No | Customer’s email |
| Address | NVARCHAR(MAX) |  | Yes | Customer’s address |
| Phone | NVARCHAR(15) |  | Yes | Customer’s phone |
| Point | INT |  | Yes | Customer’s point |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Event** | | | | |
| **Fields name** | **Data Type** | **Key** | **NULL** | **Description** |
| Event\_ID | INT | PK | No | Identity, id of Event |
| Event\_Name | NVARCHAR(Max) |  | No | Name of Event |
| Event\_Type | NVARCHAR(50) |  | No | Type of Event |
| Begin\_Date | DATE |  | No | Date begin Event |
| Run\_Time | INT |  | No | Performance time of Event |
| Ticket\_Price | INT |  | No | Ticket price of Event |
| Point | INT |  | No | Cumulative point of Event |
| Description | NVARCHAR(MAX) |  | No | Description of Event |
| Image | NVARCHAR(500) |  | No | Images of Event |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Event\_Booking** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Booking\_ID | INT | PK | No | Identity, id of Booking |
| Customer\_ID | INT | FK | No | Identity, id of Customer booking |
| Place\_ID | INT | FK | No | Id of performance Place |
| Event\_ID | INT | FK | No | Id of performance Event |
| Amount\_Of\_Payment | INT |  | No | Amount of money booking |
| Amout\_Of\_Seat | INT |  | No | Amount of seat booking |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Event\_Slot\_Details** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Event\_ID | INT | PK | No | Id of Event |
| Customer\_ID | INT | PK | No | Id of Customer |
| Place\_Seat\_ID | INT | PK | No | Id of seat in Place |
| Time\_Slot\_ID | INT | PK | No | Id of performance slot |

*Primary key(Event\_ID, Customer\_ID, Place\_Seat\_ID, Time\_Slot\_ID)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Movie\_Booking** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Booking\_ID | INT | PK | No | Identity, id of Booking |
| Customer\_ID | INT | FK | No | Identity, id of Customer booking |
| Room\_ID | INT | FK | No | Id of performance room |
| Movie\_ID | INT | FK | No | Id of performance movie |
| Amount\_Of\_Payment | INT |  | No | Amount of money booking |
| Amout\_Of\_Seat | INT |  | No | Amount of seat booking |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Movie** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Movie\_ID | INT | PK | No | Identity, id of movie |
| Movie\_Name | NVARCHAR(Max) |  | No | Name of movie |
| Genre | NVARCHAR(100) |  | No | Genre of movie |
| Language | NVARCHAR(500) |  | No | Language in movie |
| Rated | INT |  | No | Rated age of customer |
| Cast | NVARCHAR(Max) |  | No | Casts information of movie |
| Director | NVARCHAR(100) |  | No | Director information of movie |
| Run\_Time | INT |  | No | Performance time of movie |
| Ticket\_Price | INT |  | No | Ticket price of movie |
| Begin\_Date | DATE |  | No | Start date of movie |
| End\_Date | DATE |  | No | End date of movie |
| Point | INT |  | No | Cumulative point of movie |
| Description | NVARCHAR(Max) |  | No | Description of movie |
| Image | NVARCHAR(500) |  | No | Images of movie |
| Media | NVARCHAR(500) |  | No | Trailer of movie |

**Movie\_Slot\_Details:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Movie\_ID | INT | PK | No | Id of Movie |
| Customer\_ID | INT | PK | No | Id of Customer |
| Seat\_ID | INT | PK | No | Id of seat in Room |
| Time\_Slot\_ID | INT | PK | No | Id of performance movie slot |

*Primary key(Movie\_ID, Customer\_ID, Seat\_ID, Time\_Slot\_ID)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Place** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Place\_ID | INT | PK | No | Identity, id of Place |
| Place\_Name | NVARCHAR (50) |  | No | Place’s name |
| Place\_Address | NVARCHAR (500) |  | No | Address of Place |
| Phone | NVARCHAR (15) |  | No | Phone of Place |
| Area\_ID | INT | FK | No | Id of area of Place |
| Seat\_ID | INT | FK | No | Id of seat in Place |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Place\_Seat** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Seat\_ID | INT | PK | No | Identity, id of Seat in Place |
| Seat\_Column | NVARCHAR (5) |  | No | Column of seat |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Promotion** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Promotion\_ID | INT | PK | No | Identity, id of Promotion |
| Promotion\_Name | NVARCHAR (50) |  | No | Promotion’s name |
| Discount | INT |  | No | Discount info of Promotion |
| Begin\_Date | DATE |  | Yes | Begin date Promotion |
| End\_Date | DATE |  | Yes | End date Promotion |
| Description | NVARCHAR (1000) |  | No | Description info of Promotion |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Room** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Room\_ID | INT | PK | No | Identity, id of Room in Cinema |
| Room\_Name | NVARCHAR (50) |  | No | Room name in Cinema |
| Cinema\_ID | INT | FK | No | Id of Cinema have this Room |
| Seat\_ID | INT | FK | No | Id of Seat in Room |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Seat** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Seat\_ID | INT | PK | No | Identity, id of Seat in Room |
| Seat\_Column | NVARCHAR (5) |  | No | Column of seat |

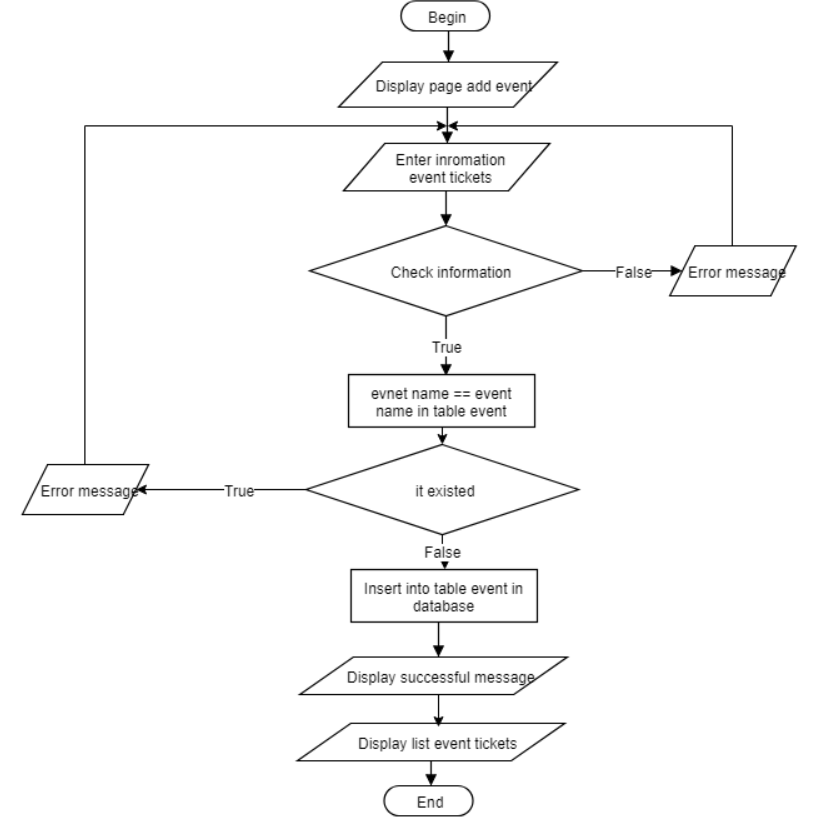
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Sport** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Sport\_ID | INT | PK | No | Identity, id of Sport |
| Sport\_Name | NVARCHAR(Max) |  | No | Name of Sport |
| Teams | NVARCHAR(500) |  | No | Participating teams of Sport |
| Running\_Datetime | DATETIME |  | No | Date time begin Sport |
| Run\_Time | INT |  | No | Performance time of Sport |
| Ticket\_Price | INT |  | No | Ticket price of Sport |
| Point | INT |  | No | Cumulative point of Sport |
| Description | NVARCHAR(MAX) |  | No | Description of Sport |
| Image | NVARCHAR(500) |  | No | Images of Sport |
| Media | NVARCHAR(500) |  | No | Trailer of Sport |

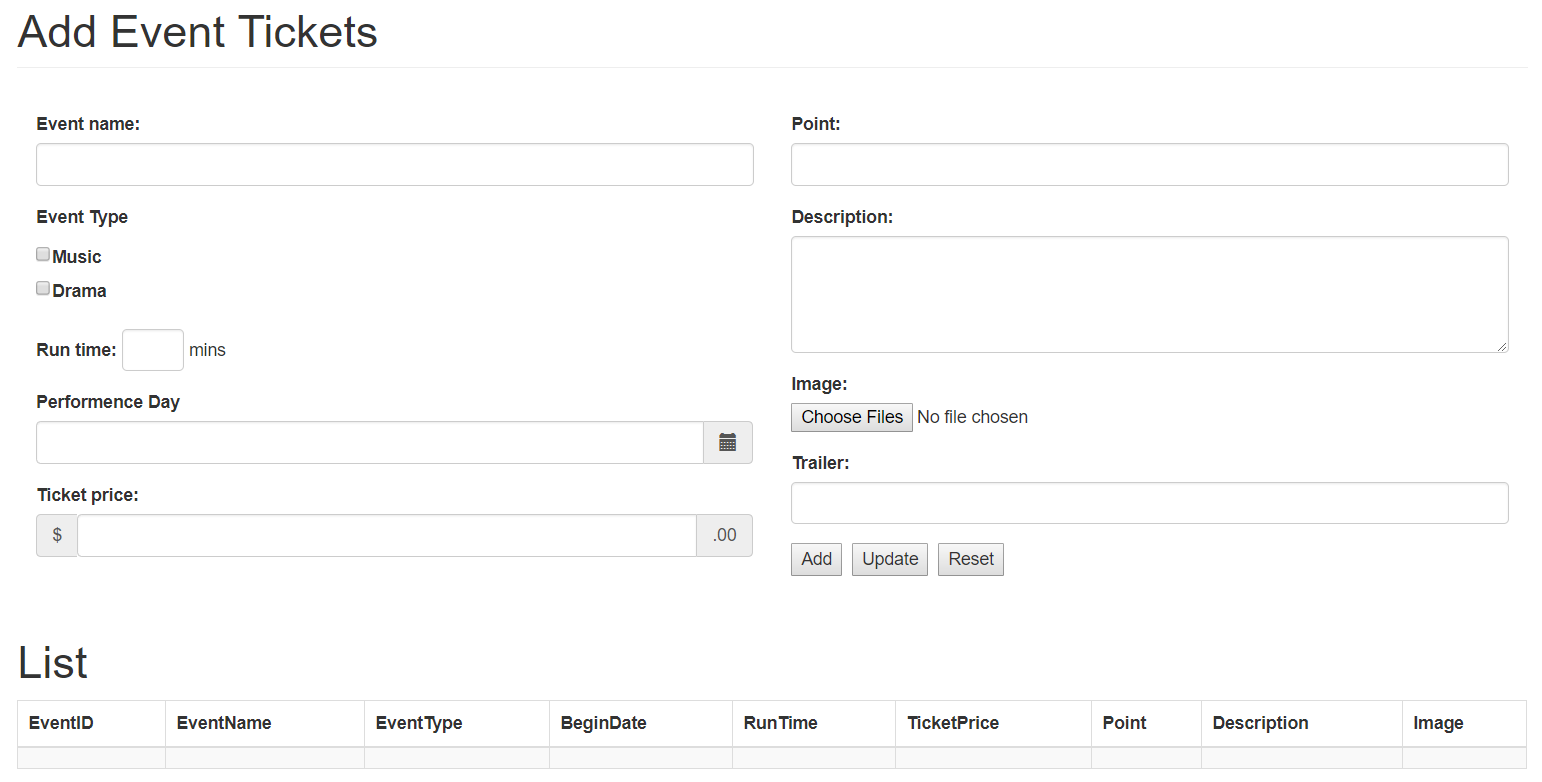
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Sport\_Booking** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Booking\_ID | INT | PK | No | Identity, id of Booking |
| Customer\_ID | INT | FK | No | Identity, id of Customer booking |
| Place\_ID | INT | FK | No | Id of performance Place |
| Sport\_ID | INT | FK | No | Id of performance Sport |
| Amount\_Of\_Payment | INT |  | No | Amount of money booking |
| Amout\_Of\_Seat | INT |  | No | Amount of seat booking |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Time\_Slot** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Time\_Slot\_ID | INT | PK | No | Identity, id of Slot |
| Slot | DATETIME |  | No | Slot info |

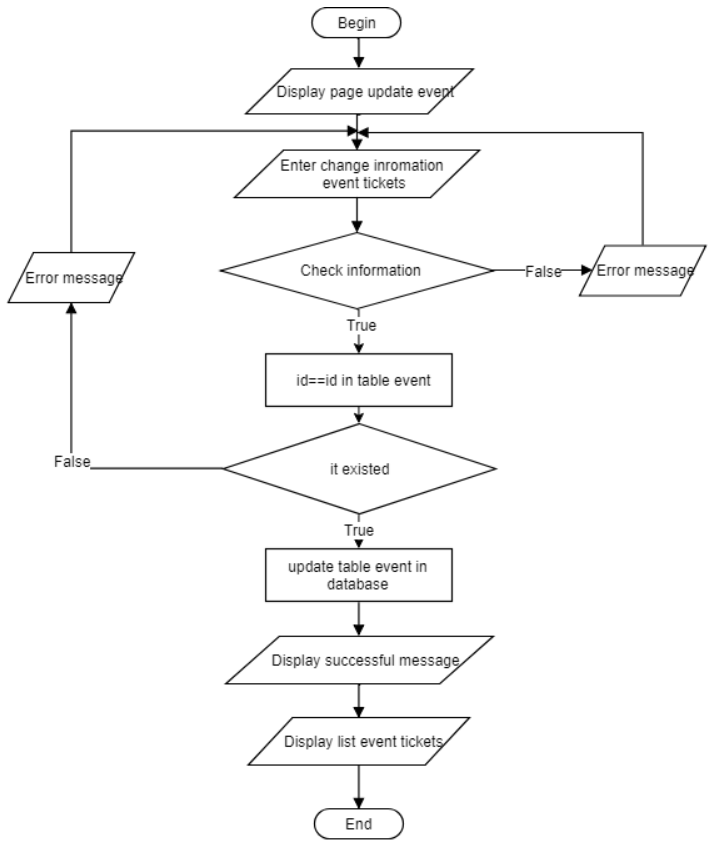
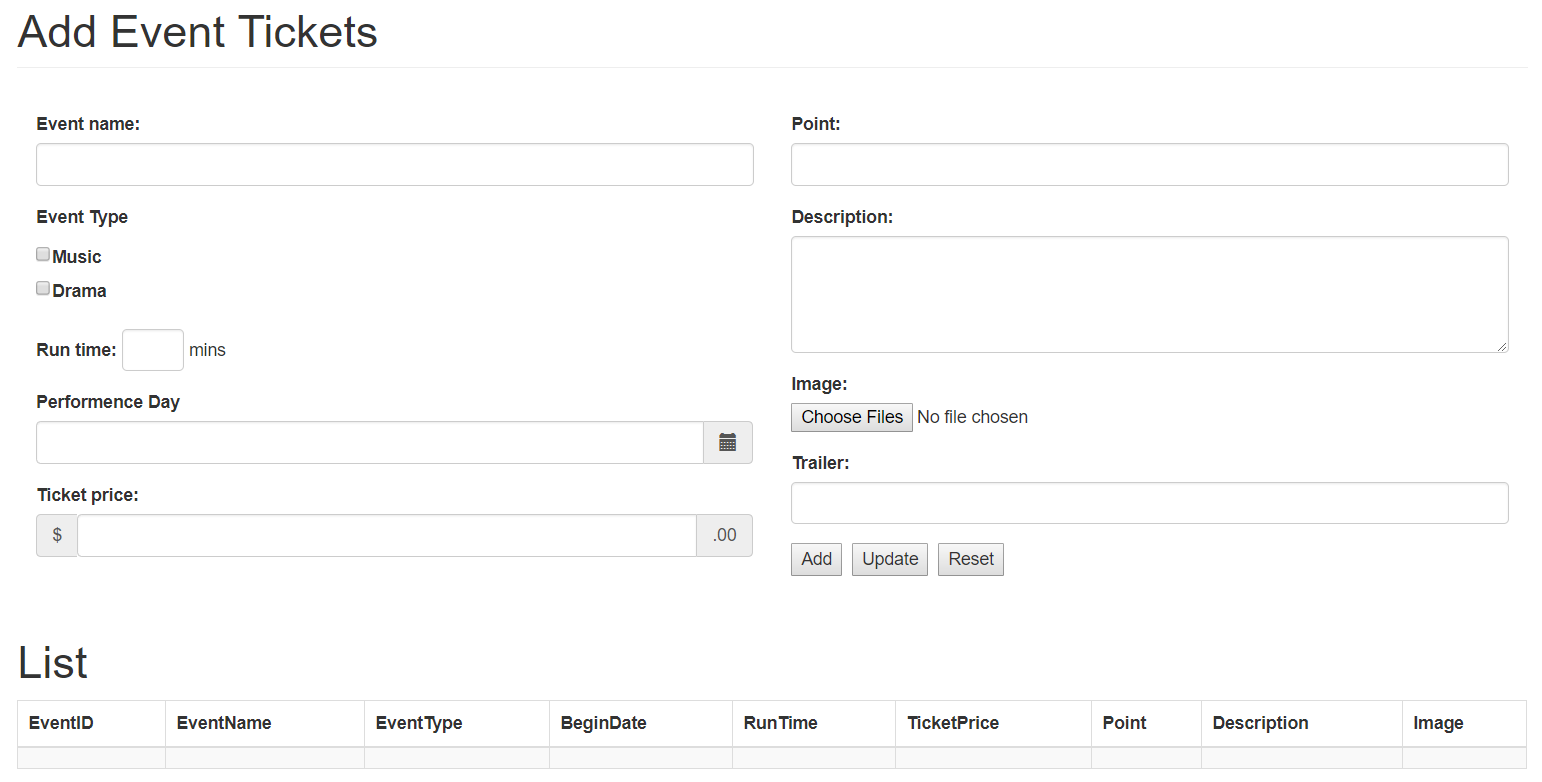
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Time\_Slot\_Movie** | | | | |
| **Fields name** | **Data Type** | **Key** | **Null** | **Description** |
| Time\_Slot\_ID | INT | PK | No | Identity, id of Slot |
| Slot | DATETIME |  | No | Slot info |

**Add Event Tickets**

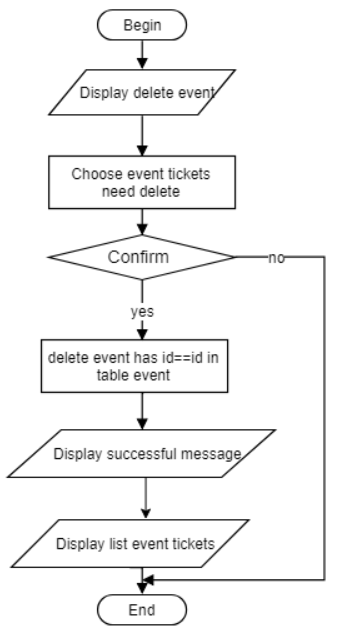


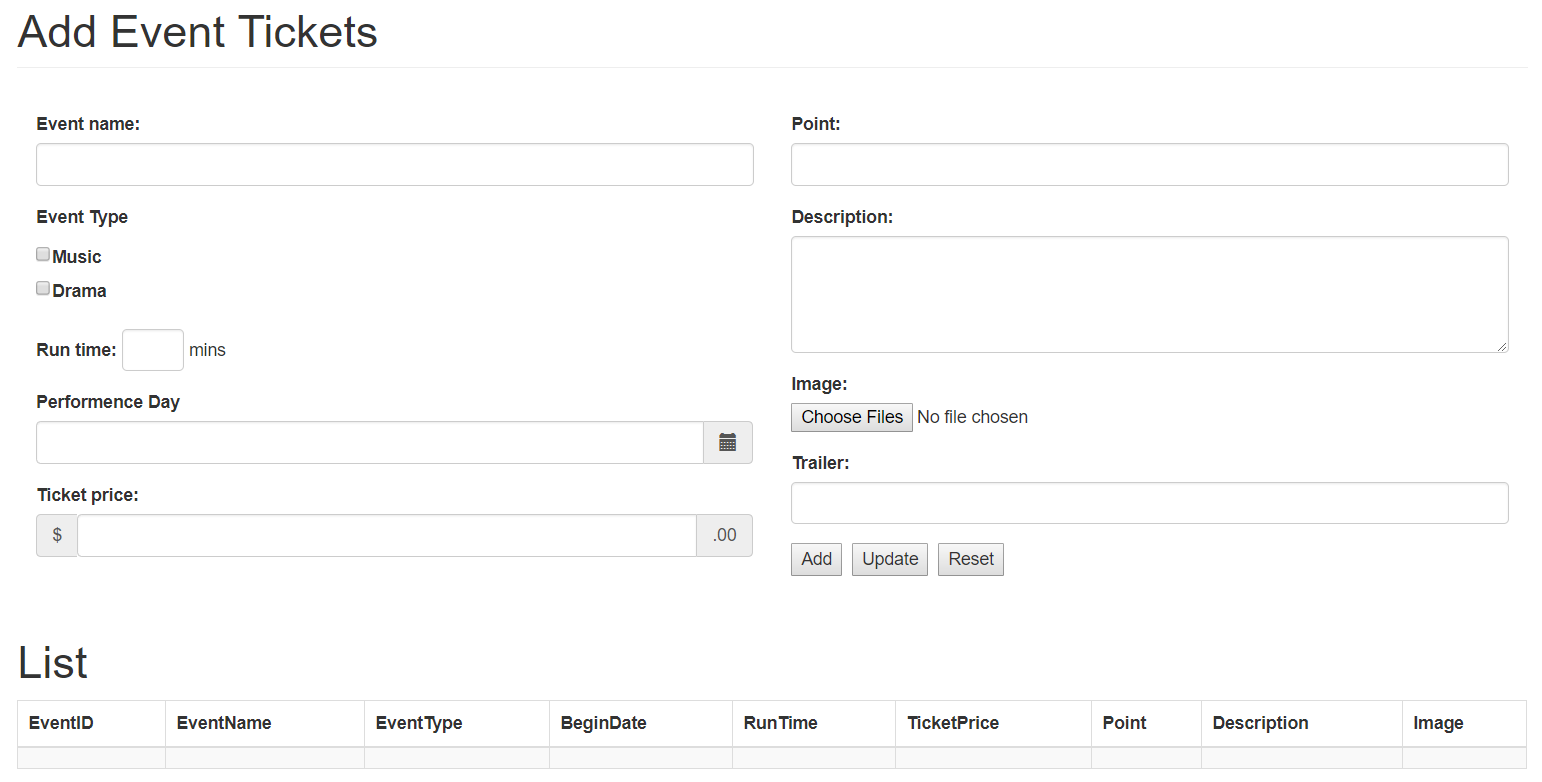


**Update Event Tickets**

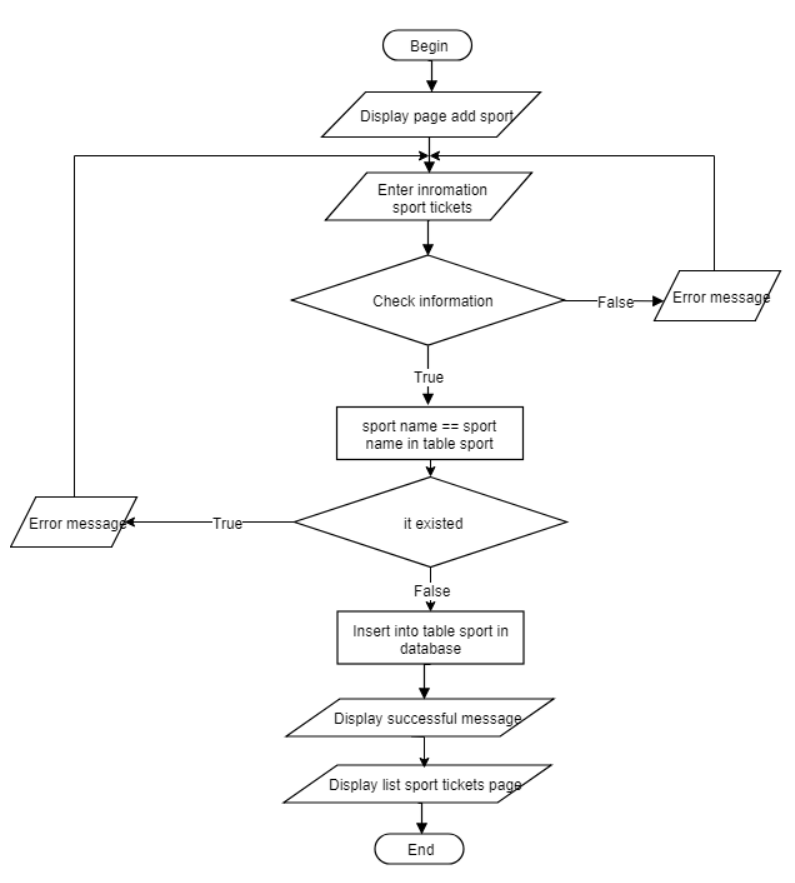


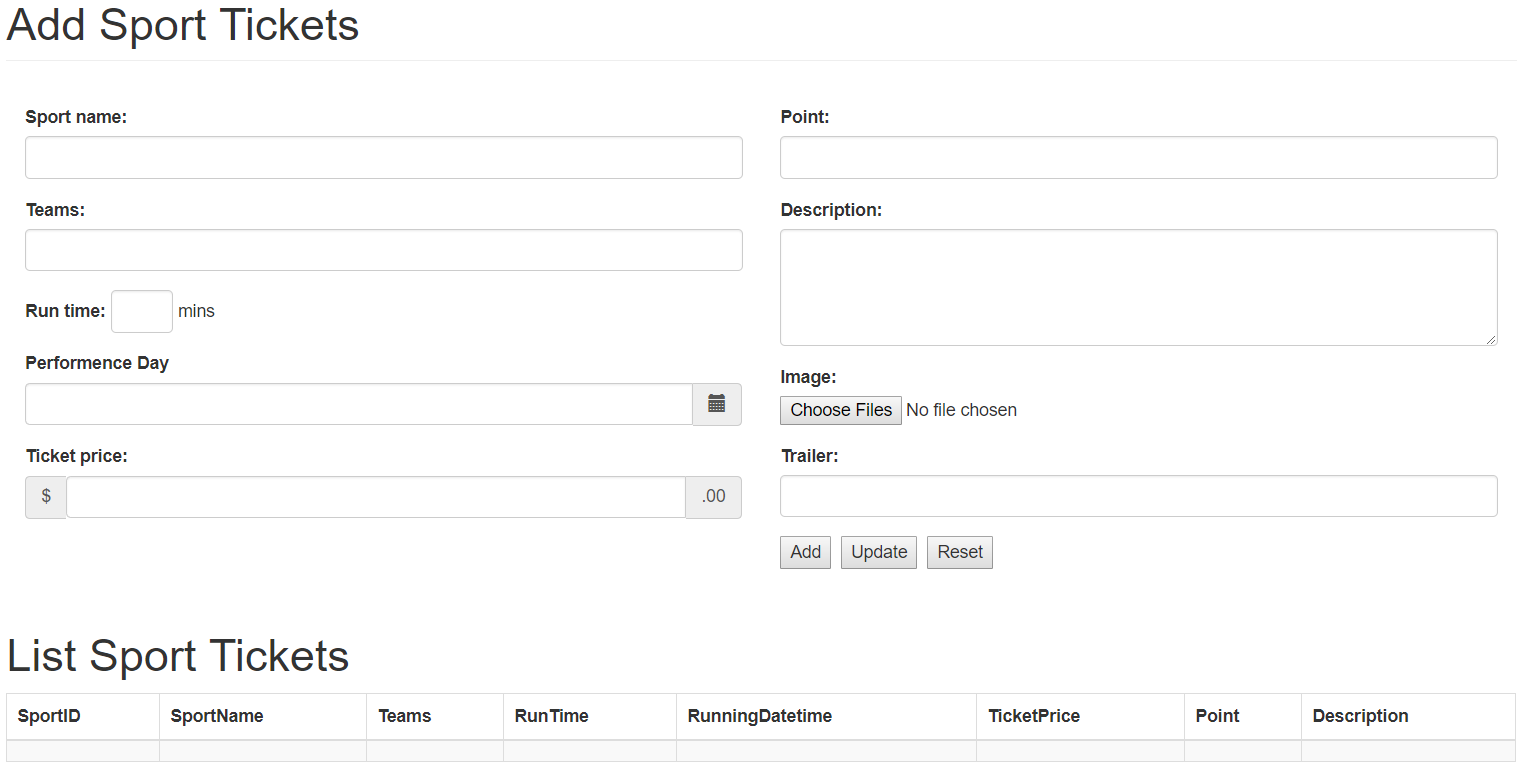
**Delete Event Tickets**



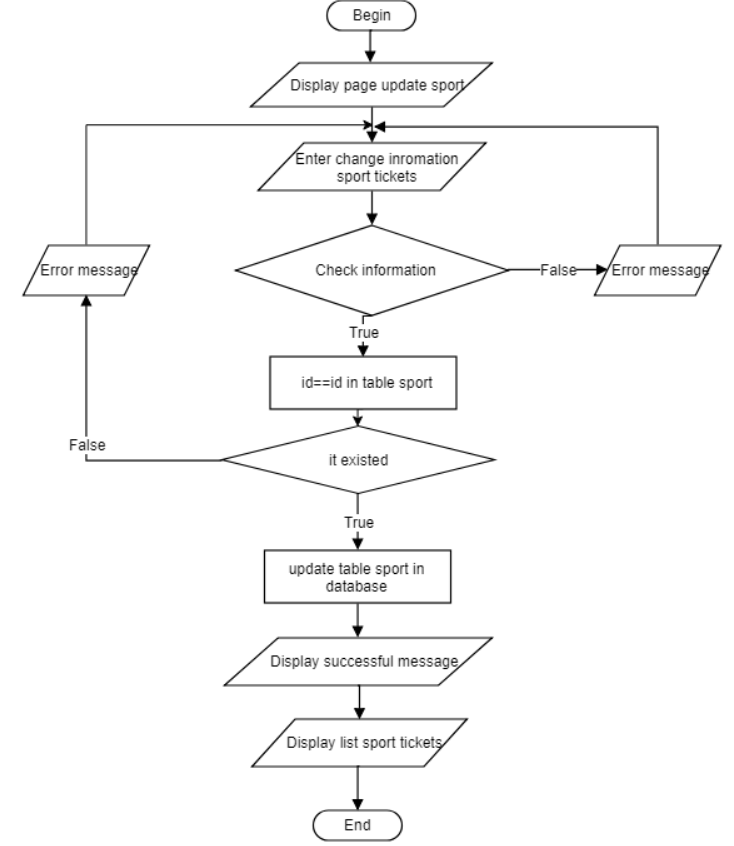
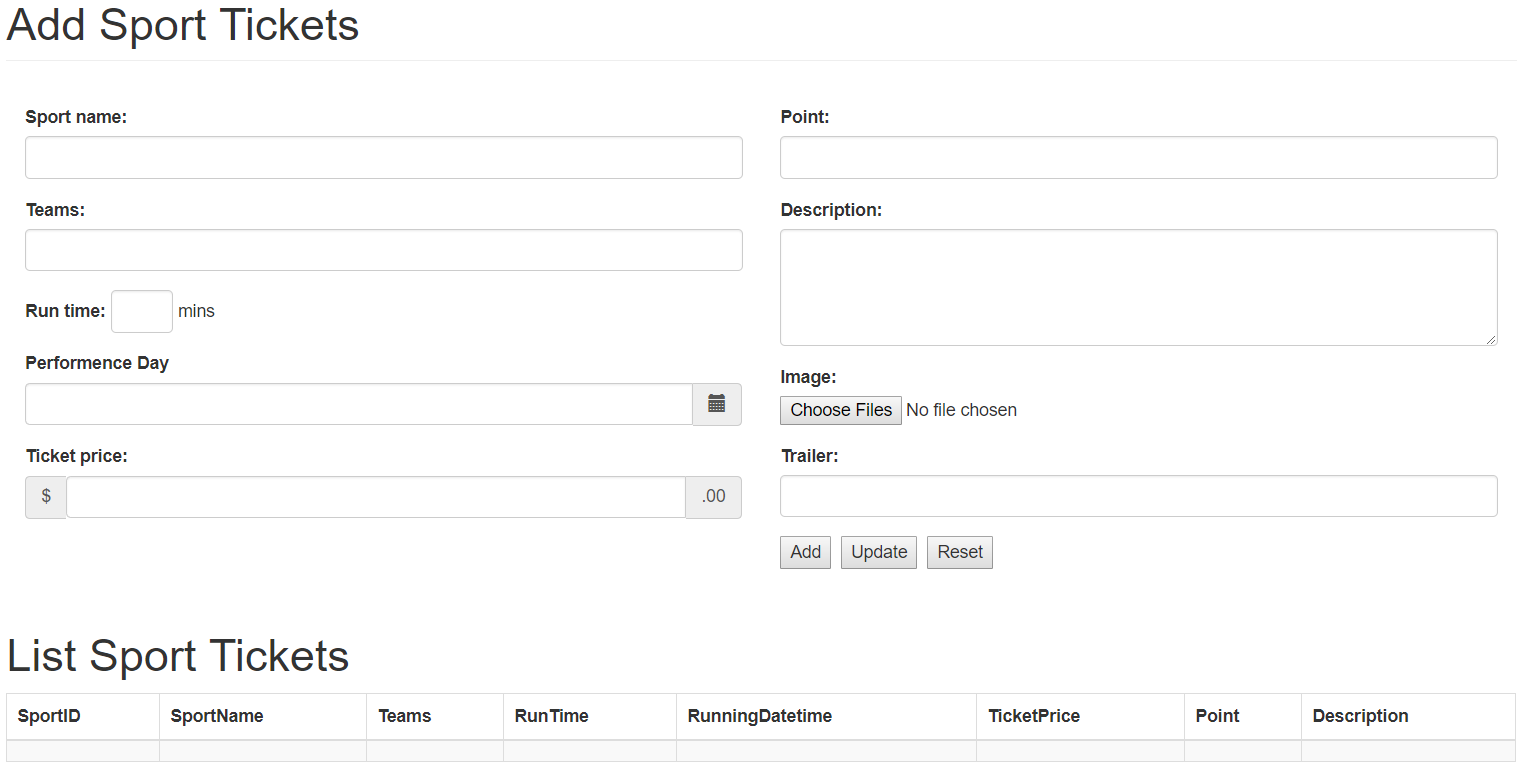


**Add Sport Tickets**

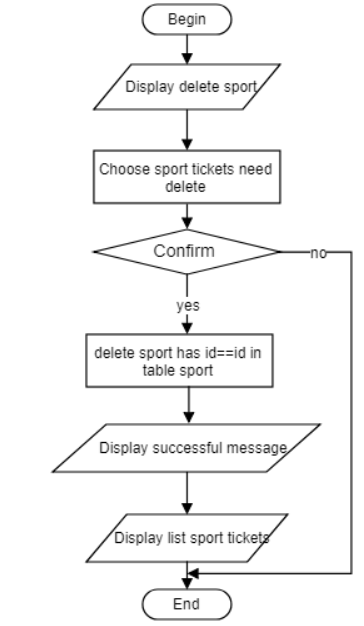


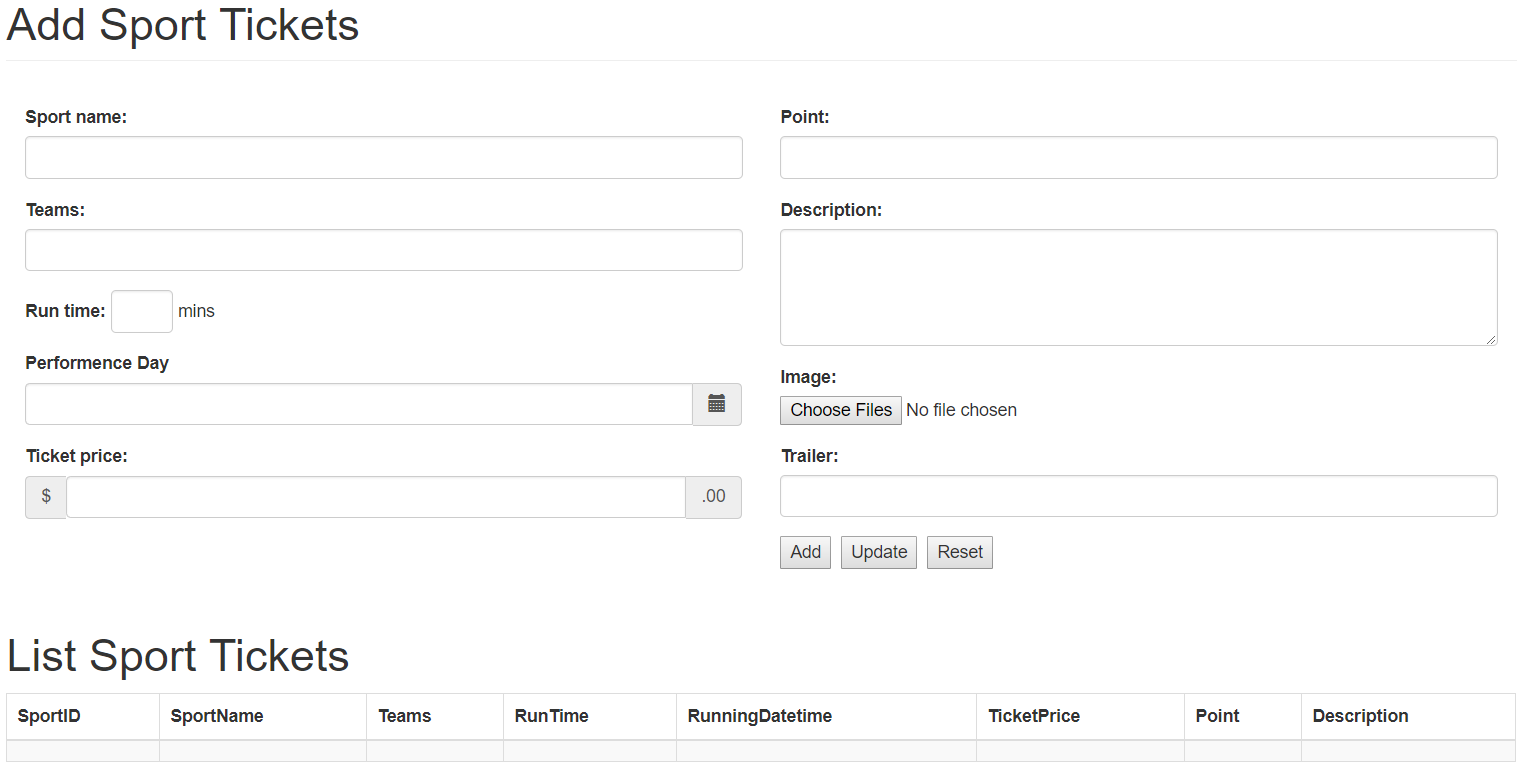


**Update Sport Tickets**

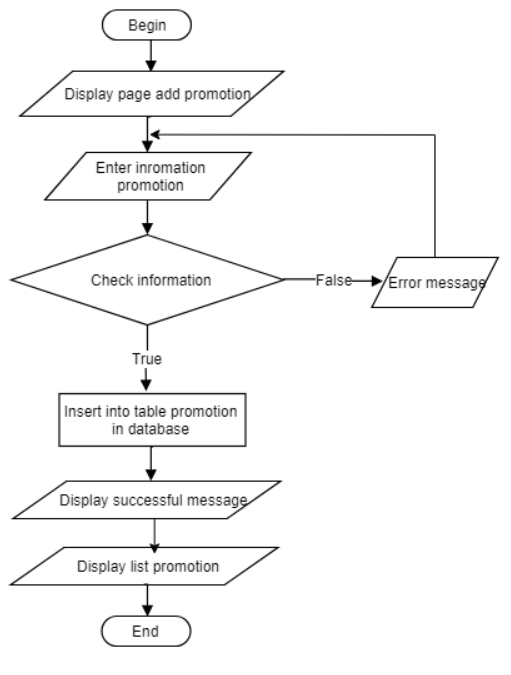


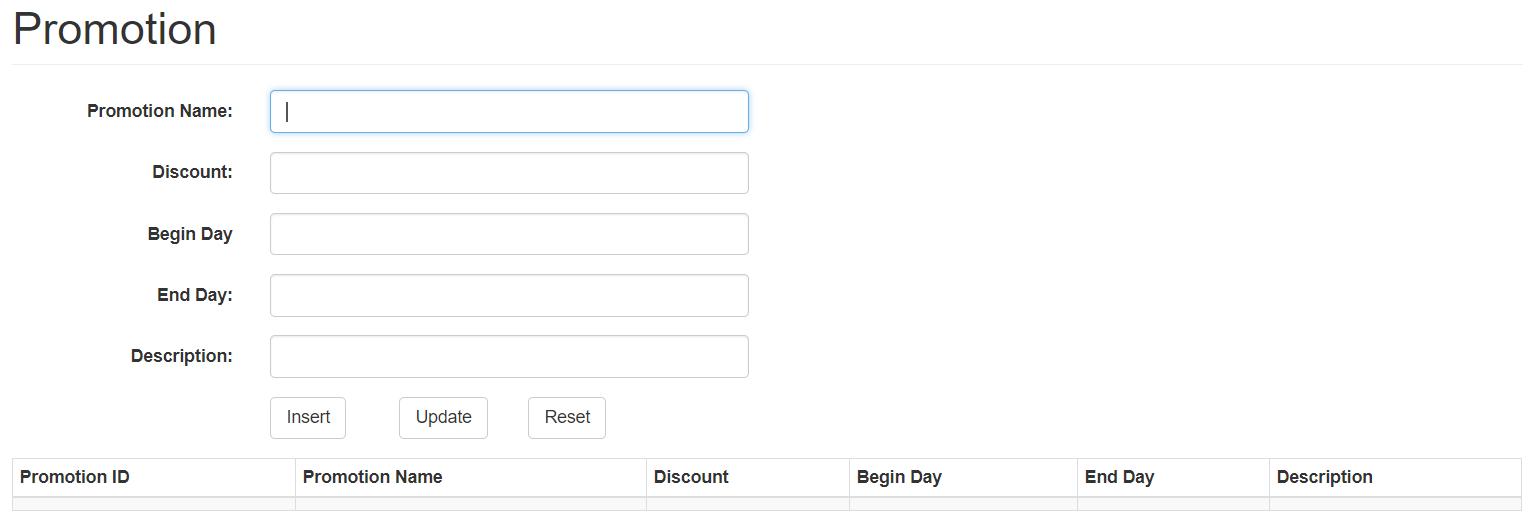
**Delete Sport Tickets**



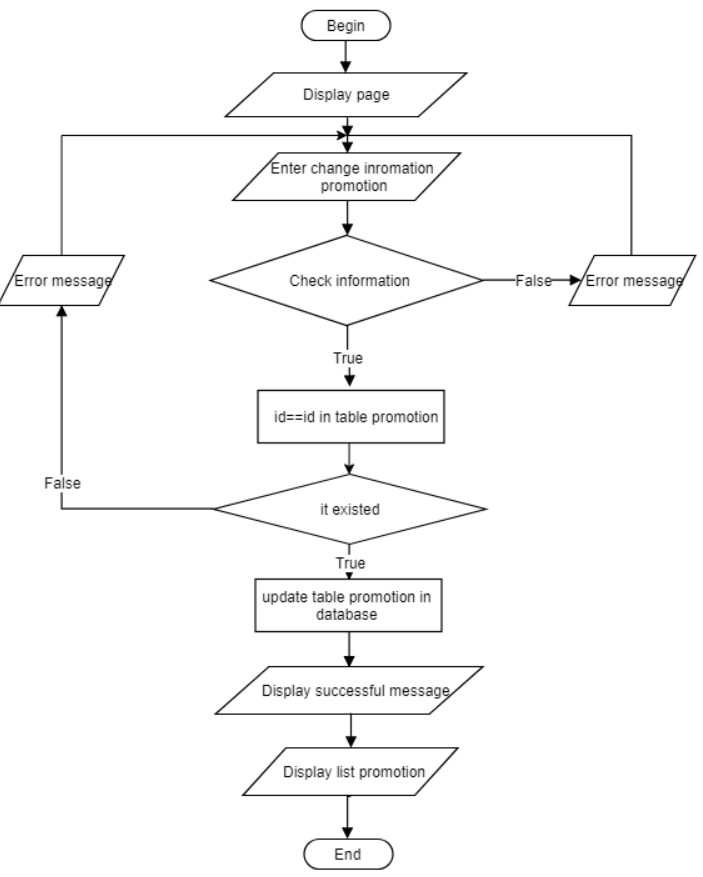
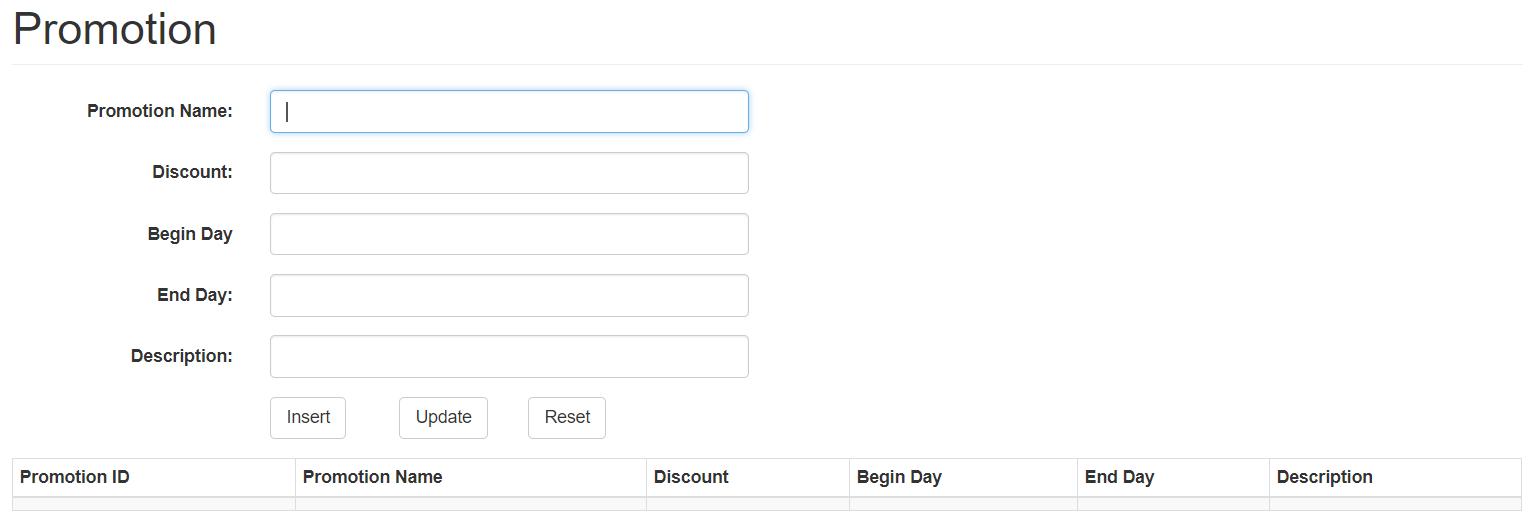


**Add Promotion**

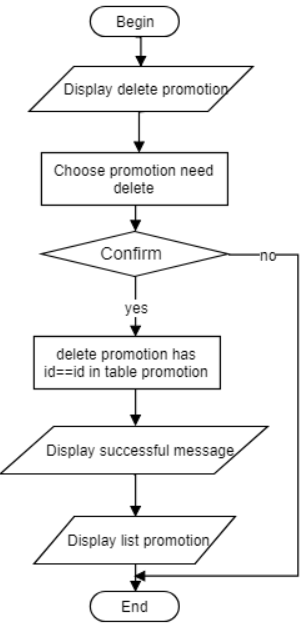


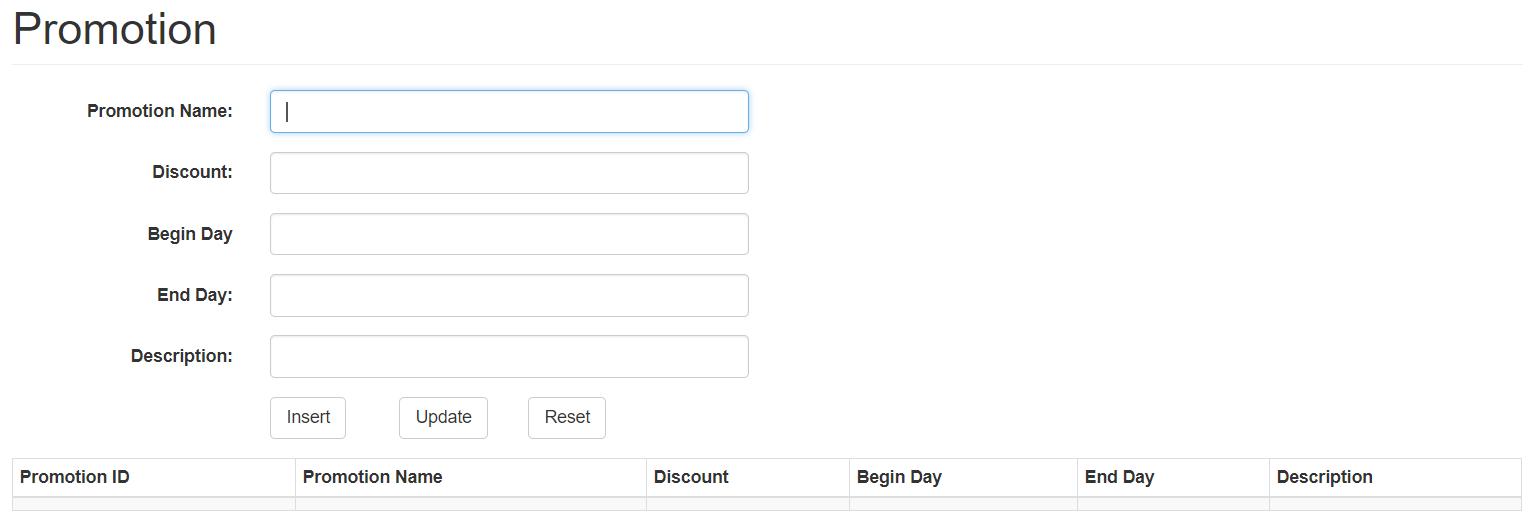


**Update Promotion**

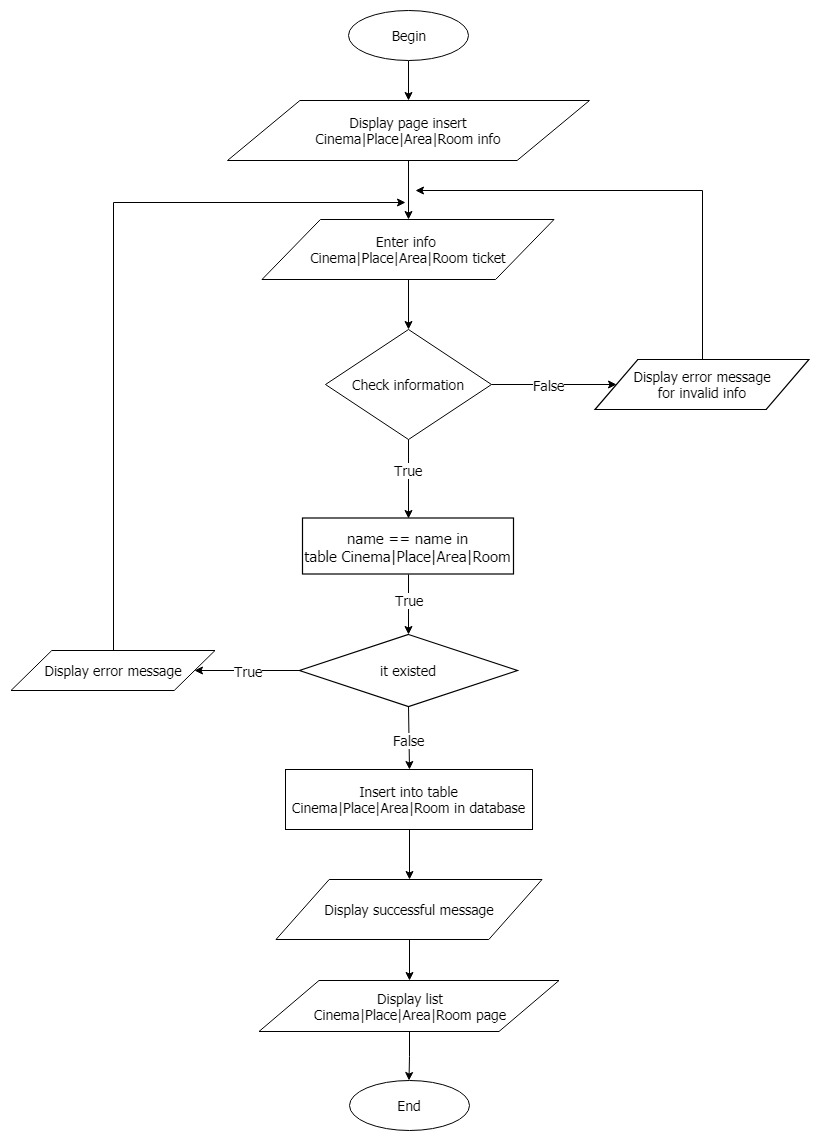


**Delete Promotion**

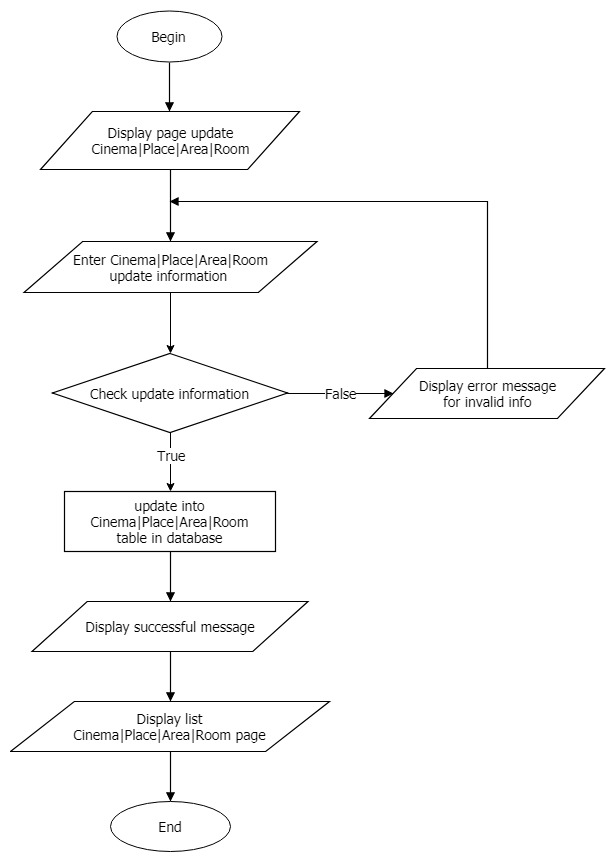




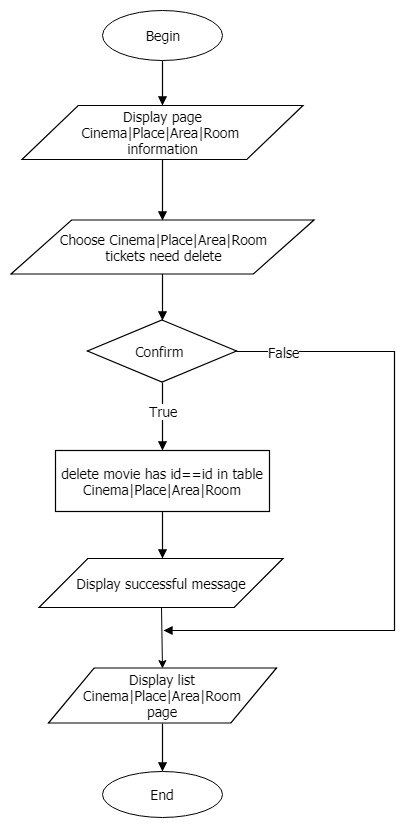
**Add Place**



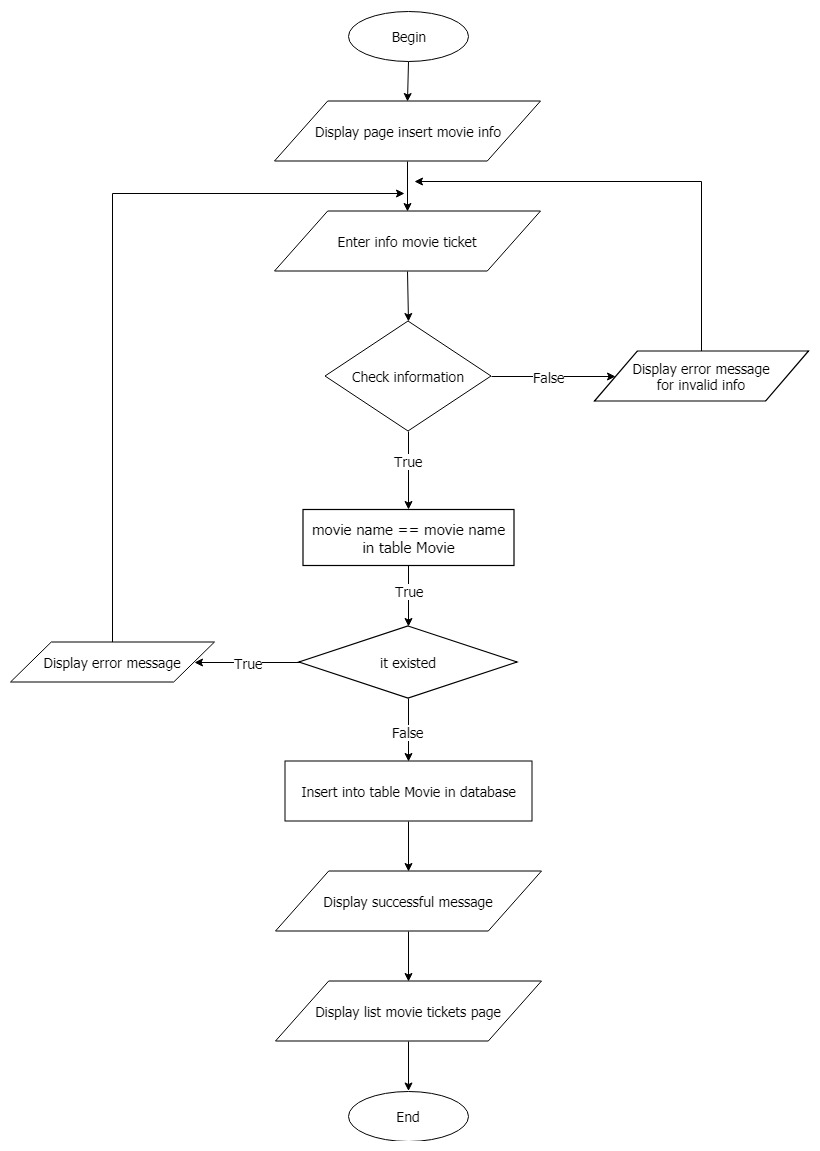
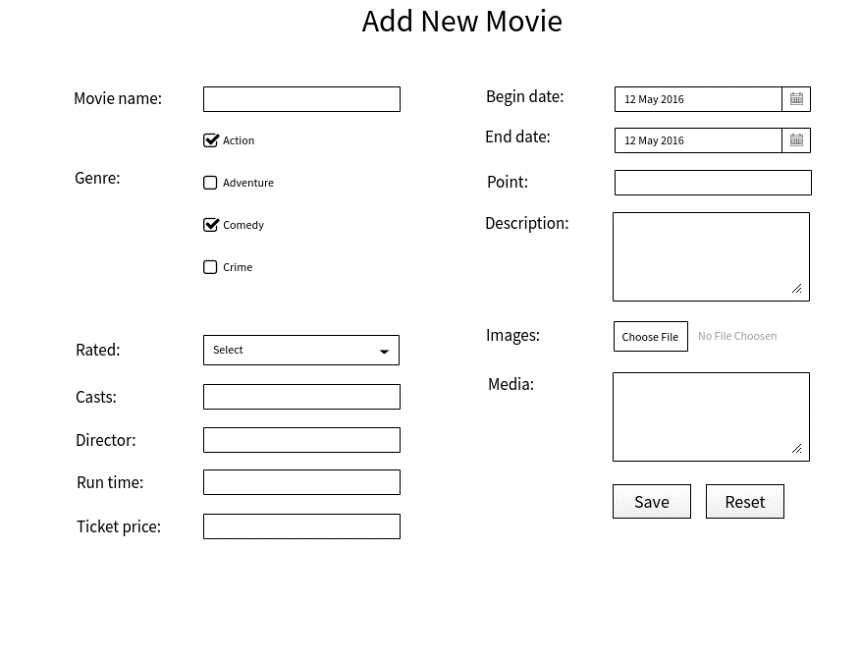
**Update Place**



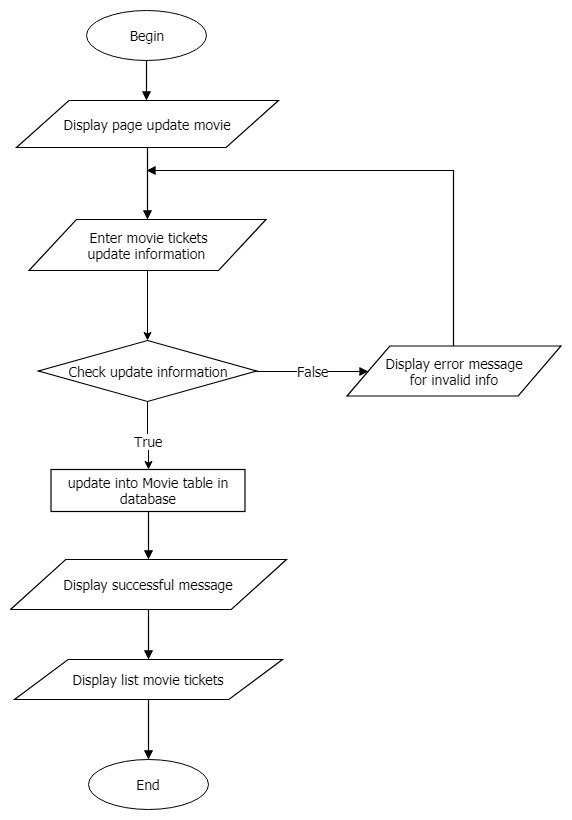
**Delete Place**

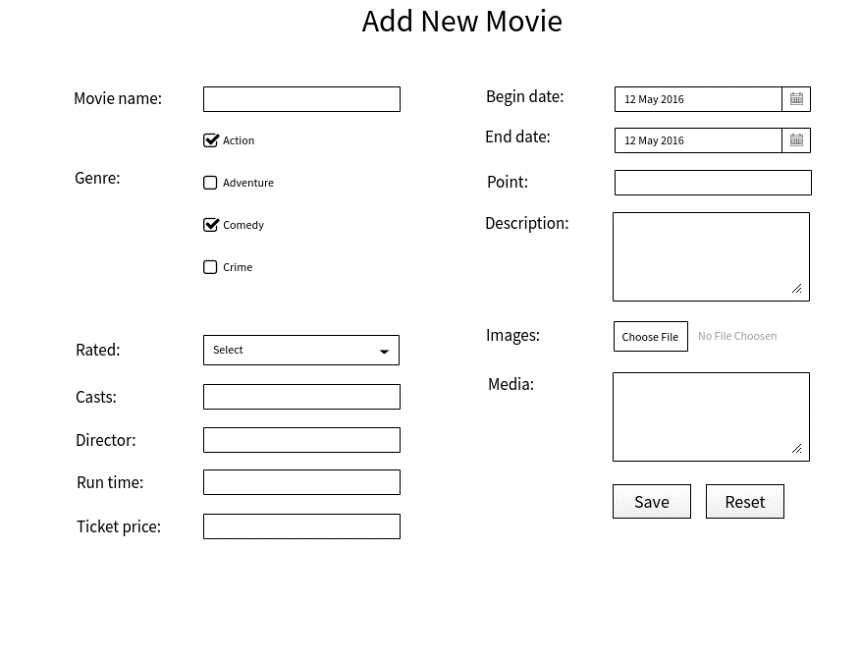


**Add Movie**

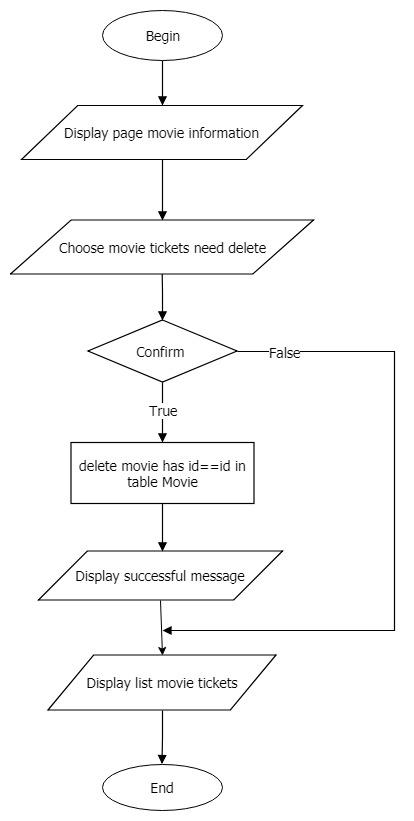


**Update Movie**

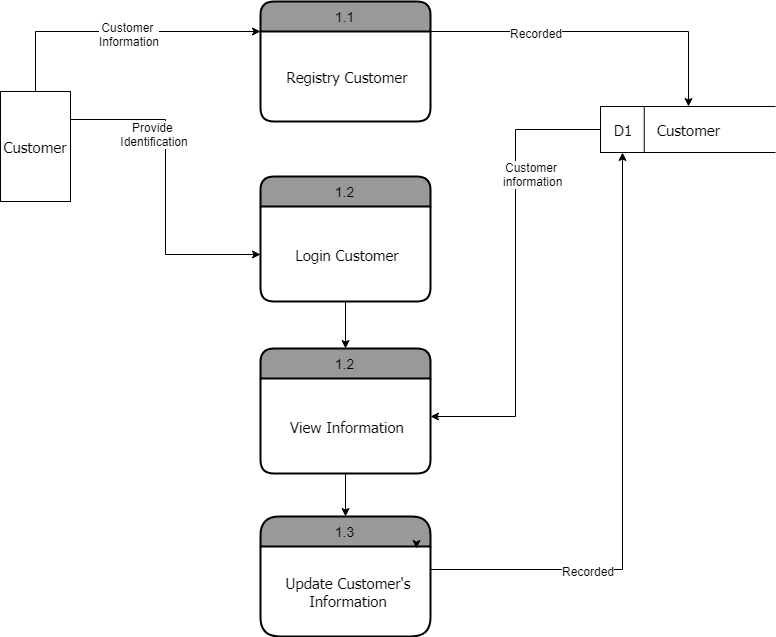




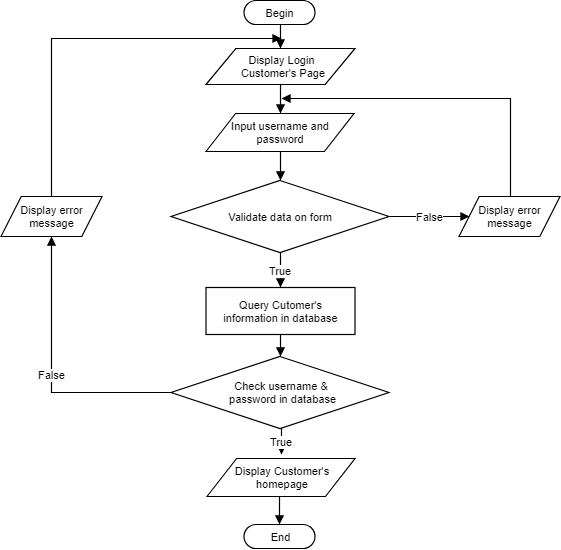
**Delete Movie**

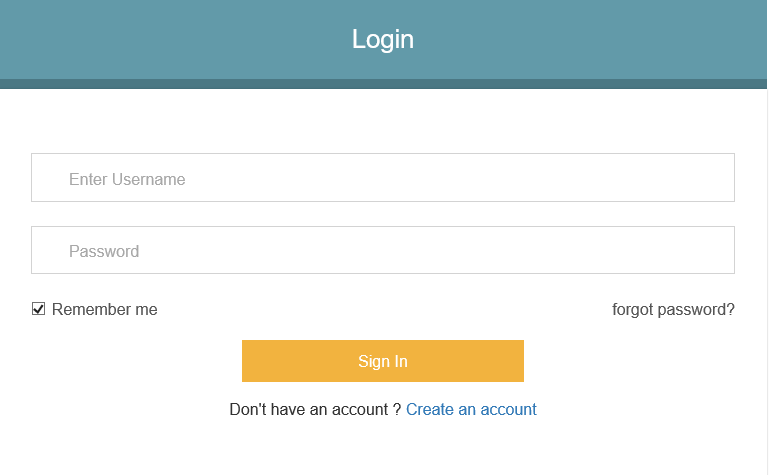


**Customer**

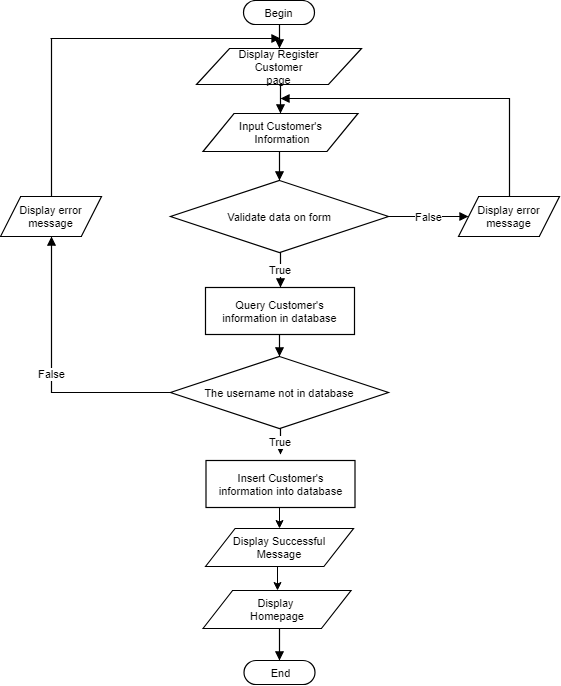
**DFD level 1:**

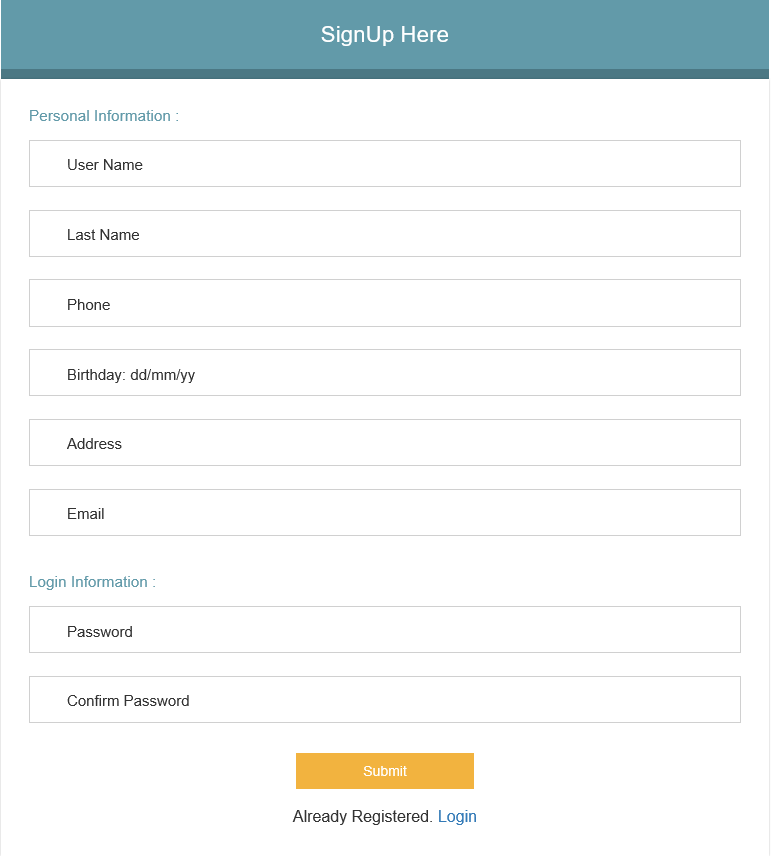
**Flow chart:** *Customer Log In*



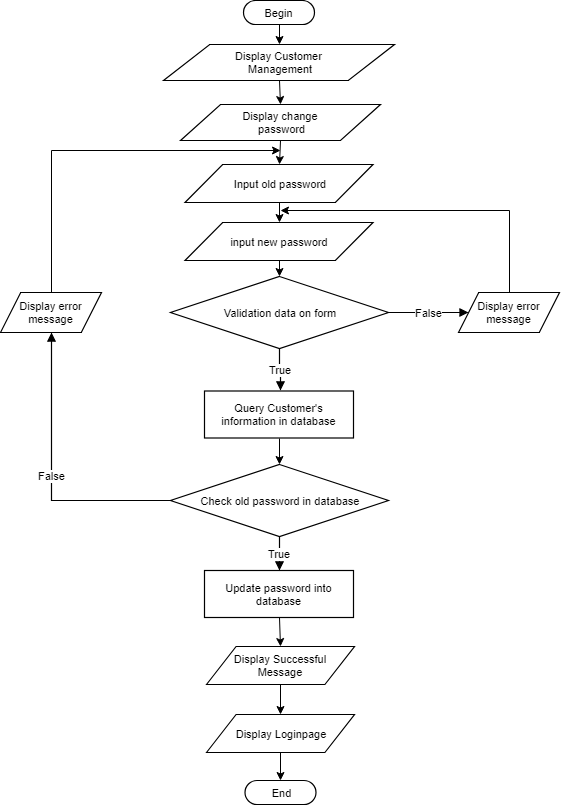
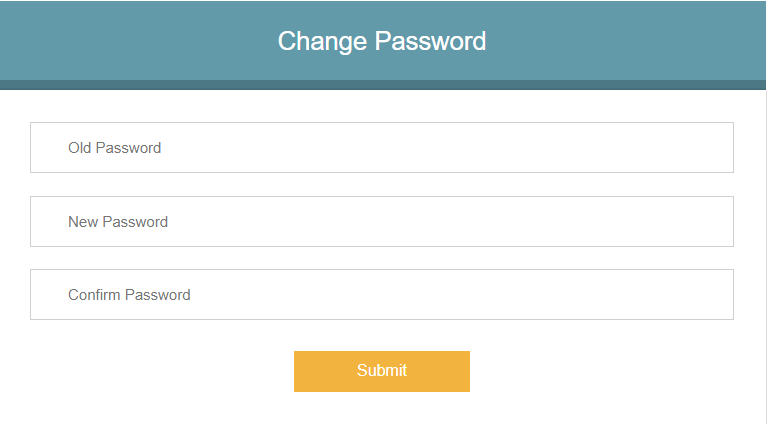


**Flow chart:** *Customer Register*

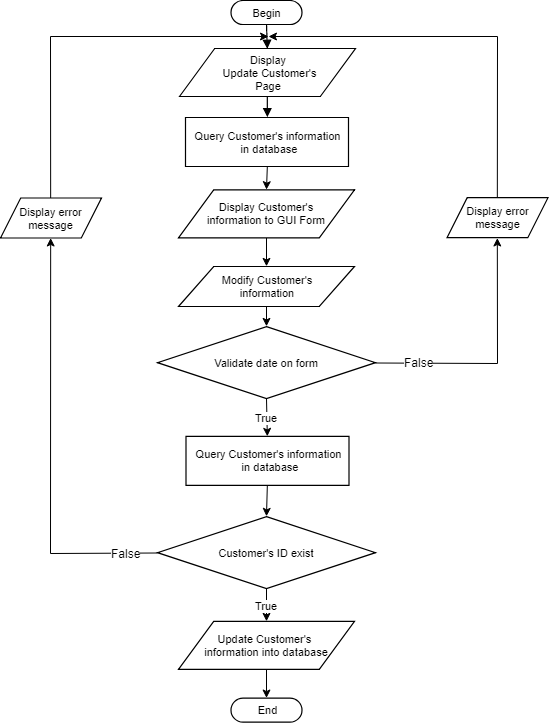
**

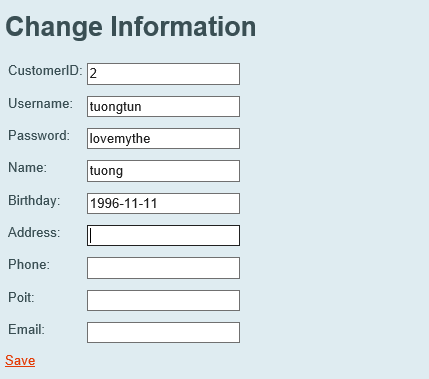


**Flow chart:** *Customer Change Password*

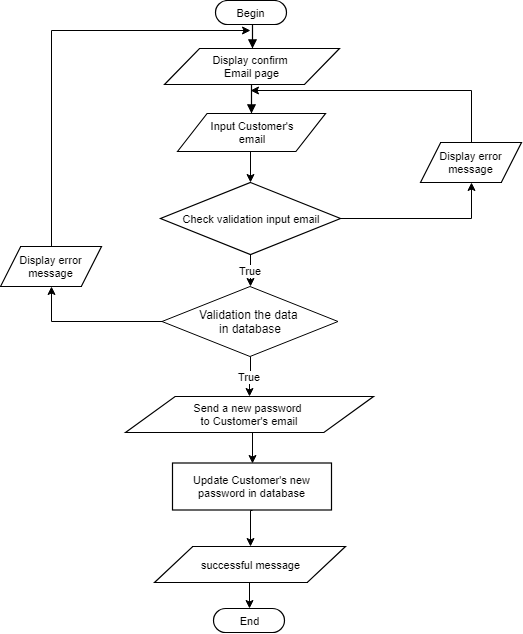
******

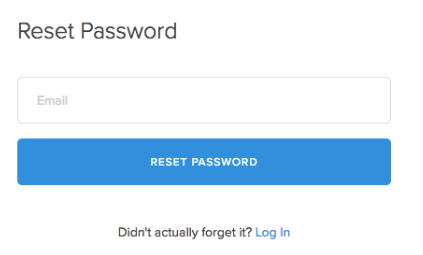
**Flow chart:** Customer *Update Information*



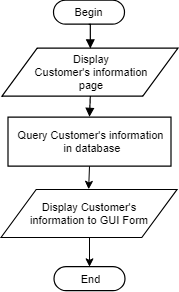


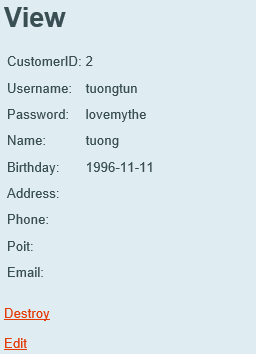
**Flow chart:** *Customer* *Reset Password*

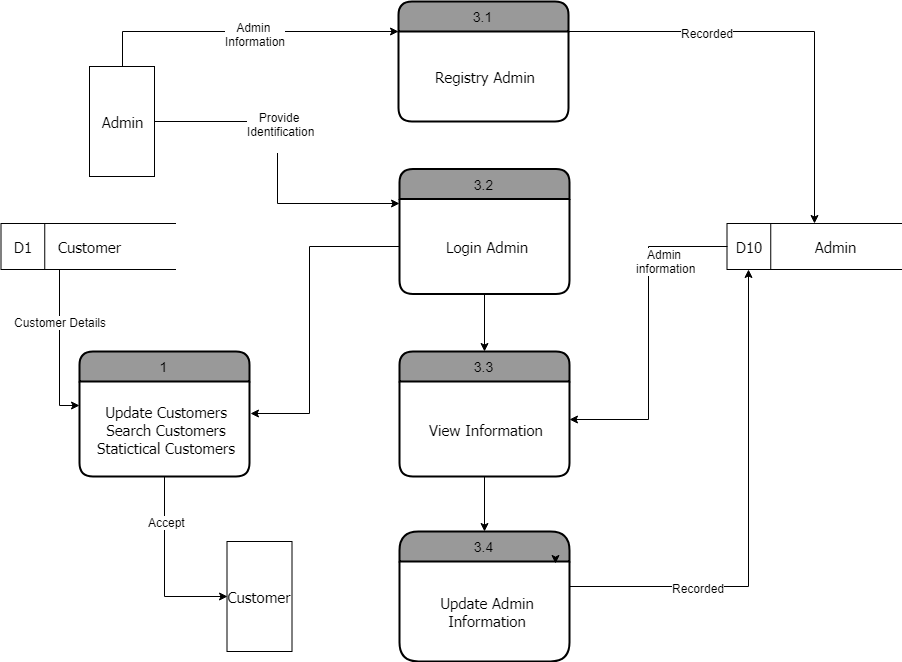
******



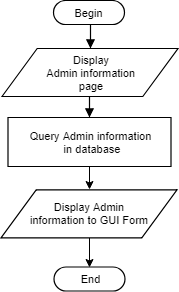
**Flow chart:** *Customer View Management*

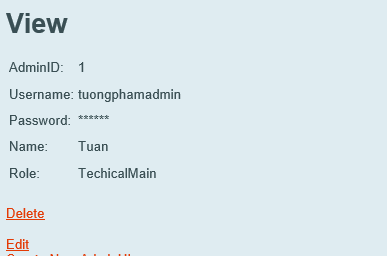
******



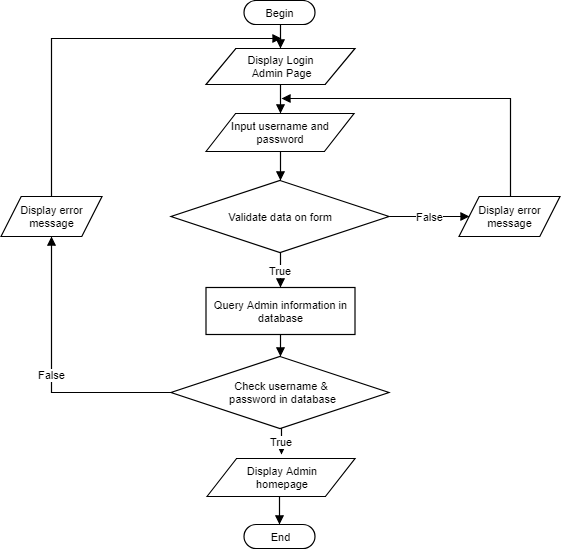
**DFD level 1: Admin**

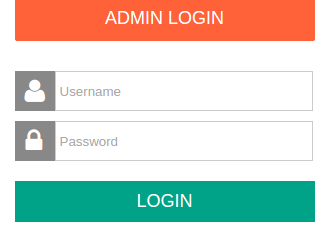
**Flow chart:** *Admin View Management*

******

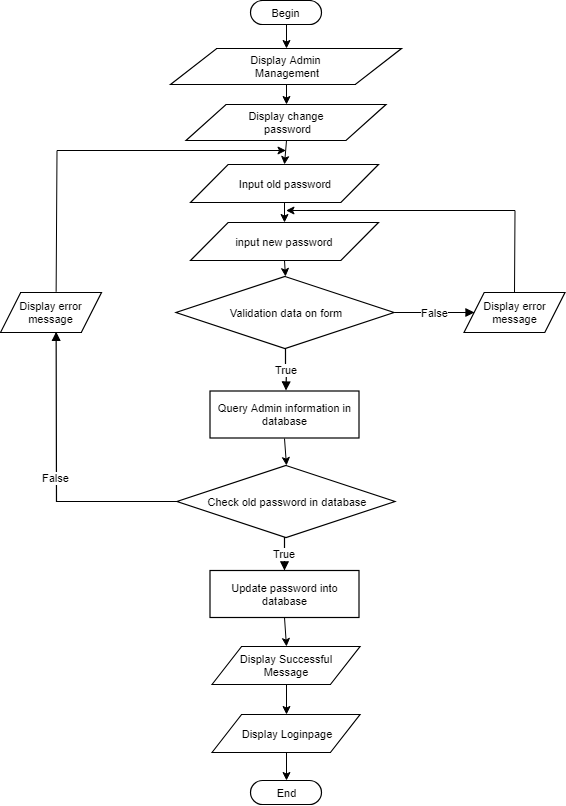
******

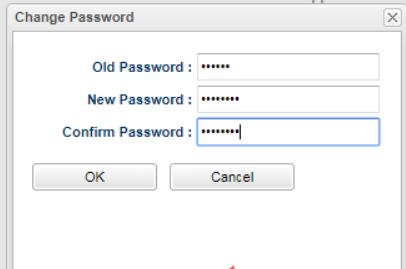
**Flow chart:** *Admin Login*

******

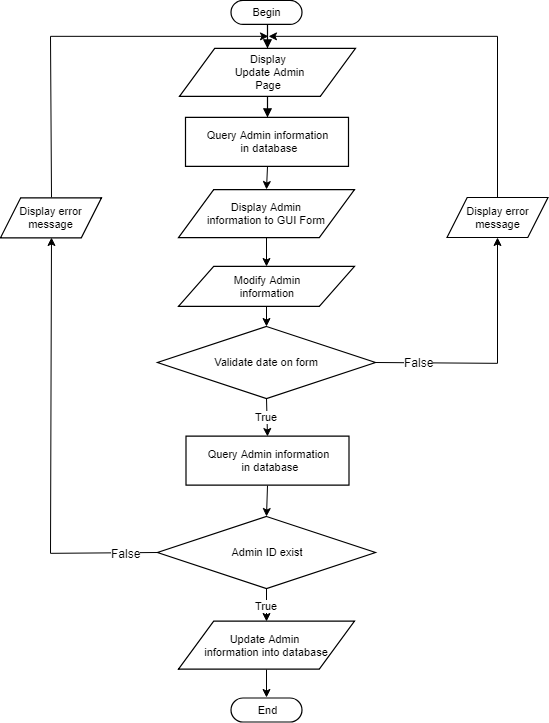
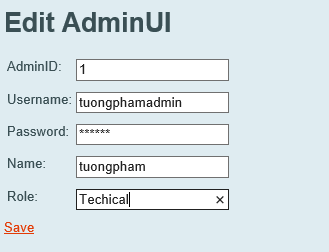


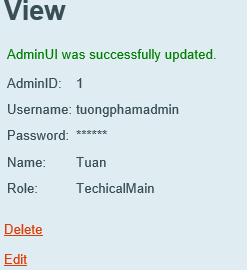
**Flow chart:** *Admin Change Password*

******

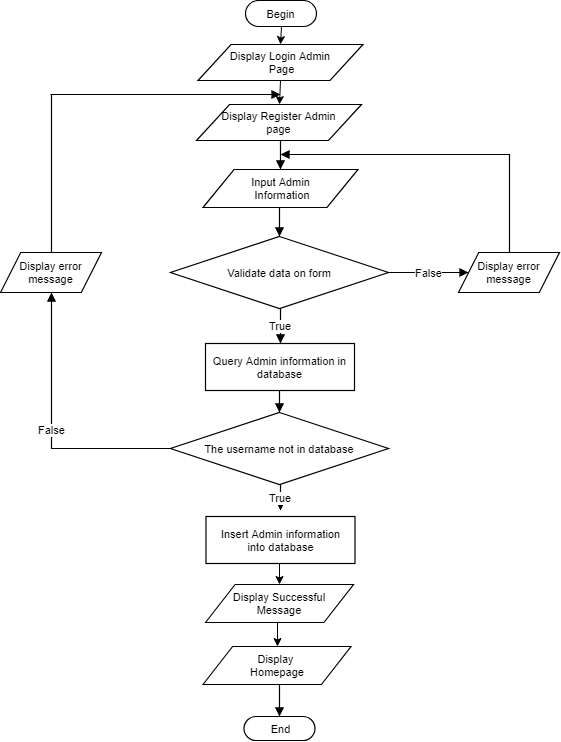


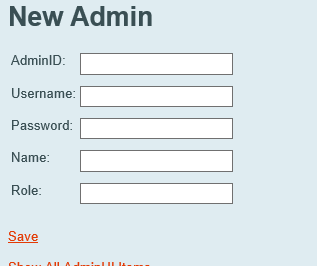
**Flow chart:** *Admin Update Information*



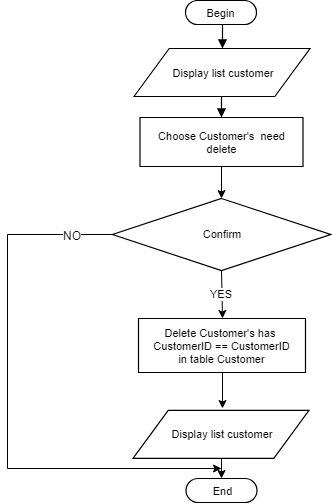


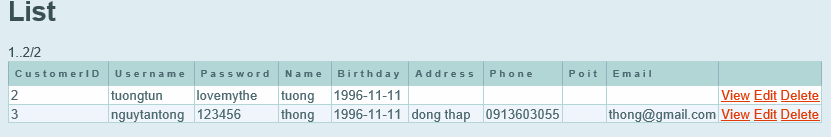
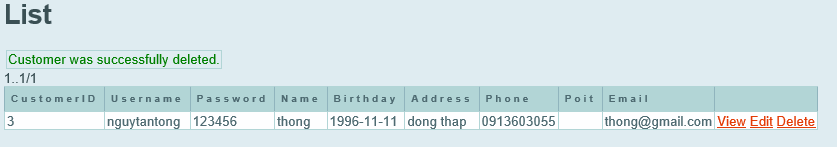
**Flow chart:** *Admin Register*



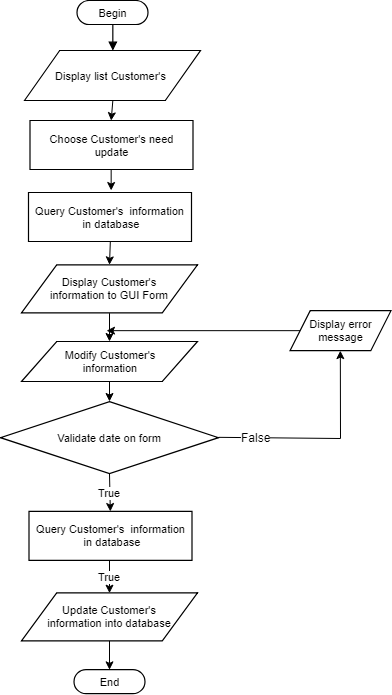


**Flow chart:** *Admin Delete Customer’s*



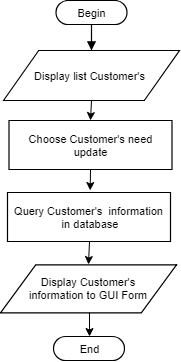


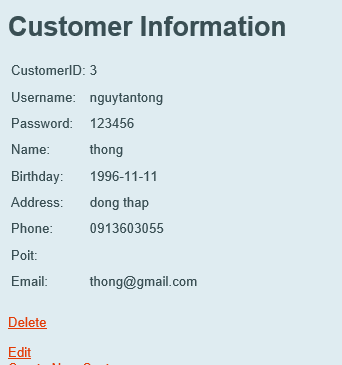
**Flow chart:** *Admin Update Customer’s Information*





**Flow chart:** *Admin View Customer’s Information*





**Programing standards:**

* 1. File name: Files containing Java code must have the extension .java
  2. Create folders to contain images or files.

**Standards for code writing style:**

1. If the body of a loop, if or else statement is a single statement (not a block), indents a tab on a separate line. Although not necessary, single statements can be enclosed in braces {} to form a block. Example :

*if (index < 1)*

*echo “invalid”;*or *if (index < 1) {*

*echo “invalid”;*

*}*

1. Set the else statement of if statement in a separate line, but the indent is the same as if statement. Example:

*if (expr) {*

*statement1;*

*statement2;*

*}*

*else {*

*statement3;*

*statement4;  
}*

and

*if (expr)*

*statement1;*

*else*

*statement2;*

1. Same for nested if-else cases:

*if(expr) {*

*if(expr2) {*

*Statement 1;*

*Statement 2;*

*}*

*else {*

*Statement 1;*

*Statement 2;*

*}*

*}*

*else {*

*if(expr3) {*

*Statement 1;*

*Statement 2;*

*}*

*else {*

*Statement 1;*

*Statement 2;*

*}*

*}*

**Standards for declaring Variables:**

1. Create meaningful variable names. Do not use names that have only a letter like a or i.
2. Create a variable name of a reasonable length
3. Constants must always write all in uppercase, underlined to separate words.
4. The name of the variable is lowercase. For example: name (in case of variable name of word: the first word is lowercase and then the letters of the following words are uppercase, ex: firstName).

**Standards for Function Declarations:**

1. The first character of the method is lowercase, for example: getName () (in case the function name consists of multiple words, the first word is lowercase and then the first letters of the following words are letters flower).
2. Each method should have a comment block to describe the meaning of the function. If you use a complex algorithm, you should include a short description of the algorithm to make the code of the method easy to understand.
3. Method rarely exceed 100 lines. In case it exceeds 100 lines, the objective of the function is unclear. Read it again and try to optimize the source code.

**Note:** Every command line should fit in 80 columns, otherwise split it into multiple lines.

**Other standards**

* 1. Space:
     + Use the space carefully to help programmers easily observe.
     + Set a space after each comma in the parameter list. For example: myMethod (val1, val2).
     + Set a space on either side of a double operator. For example: if (val1 <val2).
     + Do not set the whitespace immediately after the opening or before the closing curly bracket. Ex::

*if ( val1 < val2 ) <= shouldn’t*

* + - Do not put spaces before the semicolon (;) end of the statement.
    - Do not put spaces before the semicolon (;) end of the statement.

*Ex:*

*if (val1 < val2)*

* + - When declaring an array, combine square brackets with element types, separated by variable names, so that it applies to all variables on that row.

*Ex: int[30] list1, list2.*

* 1. Export notice:
     1. Full of information but brief.
     2. Define specific input options in the prompt when appropriate.
     3. Describe the default options in the prompt when appropriate.
  2. Export result:
     + Present information to users in a consistent and understandable way.
  3. Working with the database
     + The SQL statement will be capitalized words, clauses are reserved keywords. Table name, column name will be lowercase. Ex:

*SELECT  
 SUM(sales.units\_sold) FROM sales, customer, time, store, product*

*WHERE  
 sales.customer\_id = customer.customer\_id AND*

*sales.date\_id = time.date\_id AND*

*sales.store\_id = store.store\_id AND*

*sales.product\_id = product.product\_id AND*

*time.year\_id = 2006 AND*

*product.category\_id = "tv"*

*GROUP BY*

*product.brand, store.country\_iso\_i*

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Features Tested** | **Remarks** |
| 1 | Login page |  |
| 2 | Sign up |  |
| 3 | User management |  |
| 4 | Change Password |  |
| 5 | Booking tickets |  |
| 6 | Add, Update, Delete Sport Tickets |  |
| 7 | Add, Update, Delete Event Tickets |  |
| 8 | Add, Update, Delete Movie Tickets |  |
| 9 | Add, Update, Delete Promotion |  |
| 10 | Add, Update, Delete Promotion |  |
| 11 | Add, Update, Delete Place |  |
| 12 | Add, Update, Delete Area |  |
| 13 | Add, Update, Delete Cinema |  |
| 14 | Admin management |  |
| 15 | Add, Delete Admin |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Project plan** | **Description** | **Status** | **Comments** | **Responsibility** |
| Sep 13-2018 | Sep 17-2018 | Collect consumer’s information |  |  | Nguyễn Hoàng Nhạc  Nguyễn Tấn Thông  Phạm Cát Tường  Nguyễn Phạm Bích Phượng |
|  |  | Design |  |  | Nguyễn Hoàng Nhạc  Nguyễn Tấn Thông  Phạm Cát Tường  Nguyễn Phạm Bích Phượng |
|  |  | Program |  |  | Nguyễn Hoàng Nhạc  Nguyễn Tấn Thông  Phạm Cát Tường  Nguyễn Phạm Bích Phượng |
|  |  | Testing & debug |  |  | Nguyễn Hoàng Nhạc  Nguyễn Tấn Thông  Phạm Cát Tường  Nguyễn Phạm Bích Phượng |

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Aspect tested** | **Suggestion/Remarks** |
| 1. | Have all the modules been properly integrated and are they completely. |  |
| 2. | Does each unit meet its objective and purpose?  Are all the validations happening as specified in Process Design? |  |
| 3. | Have all Design and Coding standard been followed and implemented? |  |
| 4. | Is the GUI design consistent all over? |  |
| 5. | Are the codes working as per the specification? |  |
| 6. | Does the application’s functionality resolve the client problem, and satisfy his needs completely? |  |
| 7. | Have the hardware and software been correctly chosen? |  |
| 8. | Additional features and utilities that give value addition to the entire project. |  |