FPT INTERNATIONAL ACADEMY

International Programmers Training FPT-Aptech Center

--o0o—

eProject Document

Online Shop DVDS

Group 2		
Group Member	**Phung Van Thuc**	**Student510977**
	Vu Hoang Chien	**Student540242**
	Dinh Ngoc Hung	**Student510955**
Instructor	**Le Thi Hao**	
Batch	**C1007I**	
Semester	**4**	

- Ha Noi, 10/2012 -

Contents

1. Introduction

Music, Game, Movie is an important part of our life. Nowadays, listening to music online is hobby of most modern people. So you need to build website where users will get relevant information about the latest music, game, movie releases in the market. This website has web pages that give information about the new releases in the field of music, game, and movie in the National market.

2. Problem Definition

2.1. Problem Abstraction

The site will allow users to view all kinds of music, game, movie - includes National and International, listen to some music, see trailers game, movie, and download them for free some but not full, vote for an album, game, movie and send feedback about the site, etc. In addition, this website also provides a forum that users could discuss. The website has search and browse functions too.

2.2. The Current System

The site will allow users to view all kinds of music, game, movie - includes National and International, listen to some music, see trailers game, movie, and download them for free some but not full, vote for an album, game, movie and send feedback about the site, etc. In addition, this website also provides a forum that users could discuss. The website has search and browse functions too..

2.3. The Proposed System

When users need to know information about recent releases of music, game, movie they can open this website to see or search some information, and naturally, they can buy some DVDs if they want.

If users want to know information regarding music, game, movie or they wish to buy a new DVDs, they need not visit a shop at all. Instead of this, they can just open the website to refer to information about music displayed on the web pages and place an order.

Finally, what's thing that we want customers: come to see, and buy new releases albums.

FUNCTIONAL REQUIREMENTS :
ADMIN

Admin is the manager of this website. Admin manages these main functions:

- Manage Categories :

This function contains view categories, adding a new category, deleting, editing, search.

- Manage Artists:

Artist is a composer or a singer or both.

This function contains view artists, adding a new artist, deleting, editing, search.

- Manage Albums:

This function contains view albums, adding a new album, deleting, editing, search.

- Manage Songs, games, movies:

This function contains view, adding a new, deleting, editing, search.

- Manage Users:
    - User can be a guest or a member or a customer.

- A user hasn't registered at website is a Guest.
- A user has already registered at website is a Member.
- Member is called a Customer if he buys album from the website or he had an account at Website.
- This function contains view users, adding a new user, deleting, editing, search and recovery user.

- Manage Feedback:

This function contains view, deleting, reply and search feedback of users.

- Manage Suppliers:

Supplier is company which supplies music's discs for website.

This function contains view suppliers, adding a new supplier, deleting, editing, search and recovery supplier.

- Manage Producers:

Producer is a company which publishes.

This function contains view producers, adding a new producer, deleting, editing, search and recovery producer.

- Manager Permissions:

Permission of website is divided into many module functions. Each module function has own functions such as view, adding, deleting and editing.

This function contains view permissions of an administrator, assigning permissions, unassigned permissions and editing permission.

- Manage Products:

A product is an album which is sold on the website.

This function contains view products, adding a new product, deleting, editing, search.

- Manage Purchasing Invoices:

This function contains view purchasing invoices, adding a new purchasing invoice, deleting, editing, search.

- Manage Orders:

This function contains view orders, deleting, editing, search.

- Manage Reports:

This function helps admin can view reports.

- Manage Advertising:

This function contains changing banner of website, and changing all ads of website. The Ads can be placed in any position.

- Manage News:

This function contains view, adding, deleting, editing, search and recovery news of music, game, movie. Include National and International news.

- Manage Administrators:

Admin manages website and manage other administrators.

This function contains view administrators, adding a new administrator, deleting, editing, search and recovery administrator.

- Manage Album Reviews:

Admin manages all reviews about album of users.

This function contains adding, deleting, editing and search album reviews.

- Manage Promotion:

Admin manages promotion on website.

This function contains view all promotions and adding new promotion.

GUEST

- Guest is a user who doesn't register at website. He can use these below functions:
- Read news about music, game, movie on website.
- Browse or search information about an artist, an album, a song or a lyric, or see trailer game, movie or information content about them.
- Listen to free trailer movie, and some music online (but not full DVD)
- Join the forum of website.

MEMBER

- Member is a user who registered at website. Signing in the website, a Member can use these below functions:
- All functions of Guest.
- Download music, trailer movie, or movie game.
- Buy DVD on website:
- Write the review about album, movie, game or rating.
- Change profile (view profile, edit profile, change password, cancel account)

- Send feedback.
- Member can manage his Website Account (such as Check Balance, View Transactions Log).
- Member can manage history orders (such as view history orders …).

2.4. The System Requirement

- Java virtual machine
- GlassFish v3 server
- IE 8 /Chrome 21 / Firefox 12
- Microsoft SQL Sever 2008

2.5. Development Enviroment

- Microsoft Windows 7
- Microsoft SQL Sever 2008
- Netbean 7.1.2
- GlassFish Server

3. Requirements and Business Flow

3.1. System Administrator

a. Login to system*:*

Purpose:

  Allow use declared identification to used application.

Object used:

  Administrator

Functional request:

  - Allow user input username and password and validate information.

  - The program will display error if username or password is invalid.

Execute condition:

b. Manage Categories:

Purpose:

Allow admin to manage Categories: view categories, adding a new category, deleting, editing, search.

Object used:

Administrator

Functional request:

The application must allow admin to view categories, adding anew categories, edit, delete, search.

Execute condition:

Administrator has to log in to the system

c. Manage Artists:

Purpose:

Allow admin to manage composer or a singer or both

Object used:

Administrator

Functional request:

The application must allow admin to view Artists, adding anew Artists, edit, delete, search.

Execute condition:

Administrator has to log in to the system

d. Manage Artists:

Purpose:

Allow admin to manage Albums

Object used:

Administrator

Functional request:

The application must allow admin to view Albums, adding anew Albums, edit, delete, search.

Execute condition:

Administrator has to log in to the system

e. Manage Songs, games, movies

Purpose:

Allow admin to manage Songs, games, movies

Object used:

Administrator

Functional request:

The application must allow admin to view**,** adding anew, edit, delete, search.

Execute condition:

Administrator has to log in to the system

f. Manage Users

Purpose:

Allow admin to manage Users

Object used:

Administrator

Functional request:

The application must allow admin to insert, edit, delete, search user's information

The application allow admin to recovery users

Execute condition:

Administrator has to log in to the system

g. Manage FeedBack

Purpose:

Allow admin to manage Feedback

Object used:

Administrator

Functional request:

This function contains view, deleting, reply and search feedback of users

Execute condition:

Administrator has to log in to the system

h. Manage Suppliers

Purpose:

Allow admin to manage Suppliers

Object used:

Administrator

Functional request:

This function contains view suppliers, adding a new supplier, deleting, editing, search and recovery supplier

Execute condition:

Administrator has to log in to the system

i. Manage Producers

Purpose:

Allow admin to manage Producers

Object used:

Administrator

Functional request:

This function contains view Producers, adding a new Producers, deleting, editing, search and recovery Producers

Execute condition:

Administrator has to log in to the system

j. Manage Permissions

Purpose:

Allow admin to manage Permissions

Object used:

Administrator

Functional request:

Permission of website is divided into many module functions

Execute condition:

Administrator has to log in to the system

k. Manage Products

Purpose:

　　　Allow admin to manage Products

Object used:

　　　Administrator

Functional request:

　　　This function contains view Products, adding a new Products, deleting, editing, search

Execute condition:

　　　Administrator has to log in to the system

l. Manage Purchasing Invoices

Purpose:

　　　Allow admin to manage Purchasing Invoices

Object used:

　　　Administrator

Functional request:

　　　This function contains view purchasing invoices, adding a new purchasing invoice, deleting, editing, search.

Execute condition:

　　　Administrator has to log in to the system

m. Manage Orders

Purpose:

　　　Allow admin to manage Orders

Object used:

　　　Administrator

Functional request:

　　　This function contains view orders, deleting, editing, search

Execute condition:

　　　Administrator has to log in to the system

n. Manage Reports

Purpose:

Allow admin to manage Reports

Object used:

Administrator

Functional request:

This function helps admin can view reports

Execute condition:

Administrator has to log in to the system

o. Manage Advertising

Purpose:

Allow admin to manage Advertising

Object used:

Administrator

Functional request:

This function contains changing banner of website, and changing all ads of website. The Ads can be placed in any position

Execute condition:

Administrator has to log in to the system

p. Manage News

Purpose:

Allow admin to manage news

Object used:

Administrator

Functional request:

This function contains view, adding, deleting, editing, search and recovery news of music, game, movie. Include National and International news.

Execute condition:

Administrator has to log in to the system

q. Manage Administrators

Purpose:

Allow admin to manage Administrators

Object used:

Admin

Functional request:

Admin manages website and manage other administrators.

This function contains view administrators, adding a new administrator, deleting, editing, search and recovery administrator.

Execute condition:

Admin has to log in to the system

t. Manage Album reviews

Purpose:

Allow admin to manage Album reviews

Object used:

Administrator

Functional request:

This function contains adding, deleting, editing and search album reviews.

Execute condition:

Administrator has to log in to the system

u. Manage Promotion

Purpose:

Allow admin to manage Promotion

Object used:

Administrator

Functional request:

This function contains view all promotions and adding new promotion.

Execute condition:

Administrator has to log in to the system

3.2. Website
a. Register

Purpose:

> Allow visitor to register to be an user so that they can post their information

Object used:

> Visitor

Functional request:

> Allow visitor to fill Register form then submit their information

Execute condition:

> Visitor has to input real information and it must be in right format

b. Search information

Purpose:

> Allow visitor, user to find any information

Object used:

> Visitor

Functional request:

> Allow visitor and user to search information about an artist, an album, a song or a lyric, or see trailer game, movie or information content about them.

Execute condition:

> Visitor and member

c. Manage feedback

Purpose:

> Allow member to take a feedback about website and services

Object used:

> Member

Functional request:

> The website allow member to take a feedback about website and services

Execute condition:

> Member has login to website

d. Download music, trailer movie, or movie game

Purpose:

Allow member to download music, trailer movie, or movie game

Object used:

Member

Functional request:

The website allow member to Allow member to download music, trailer movie, or movie game

Execute condition:

Member has login to website

e. Buy DVD

Purpose:

Allow member to buy DVD on website

Object used:

Member

Functional request:

The website allow member to buy DVD online

Execute condition:

Member has login to website

f. Write the review about album, movie, game or rating.

Purpose:

Allow member to write the review about album, movie, game or rating.

Object used:

Member

Functional request:

The website allow member to write the review about album, movie, game or rating.

Execute condition:

Member has login to website

f. Change profile.

Purpose:

    Allow member to change profile

Object used:

    Member

Functional request:

    The function allow member to view, edit, change password, cancel account.

Execute condition:

    Member has login to websit

3.3 Activity Diagram
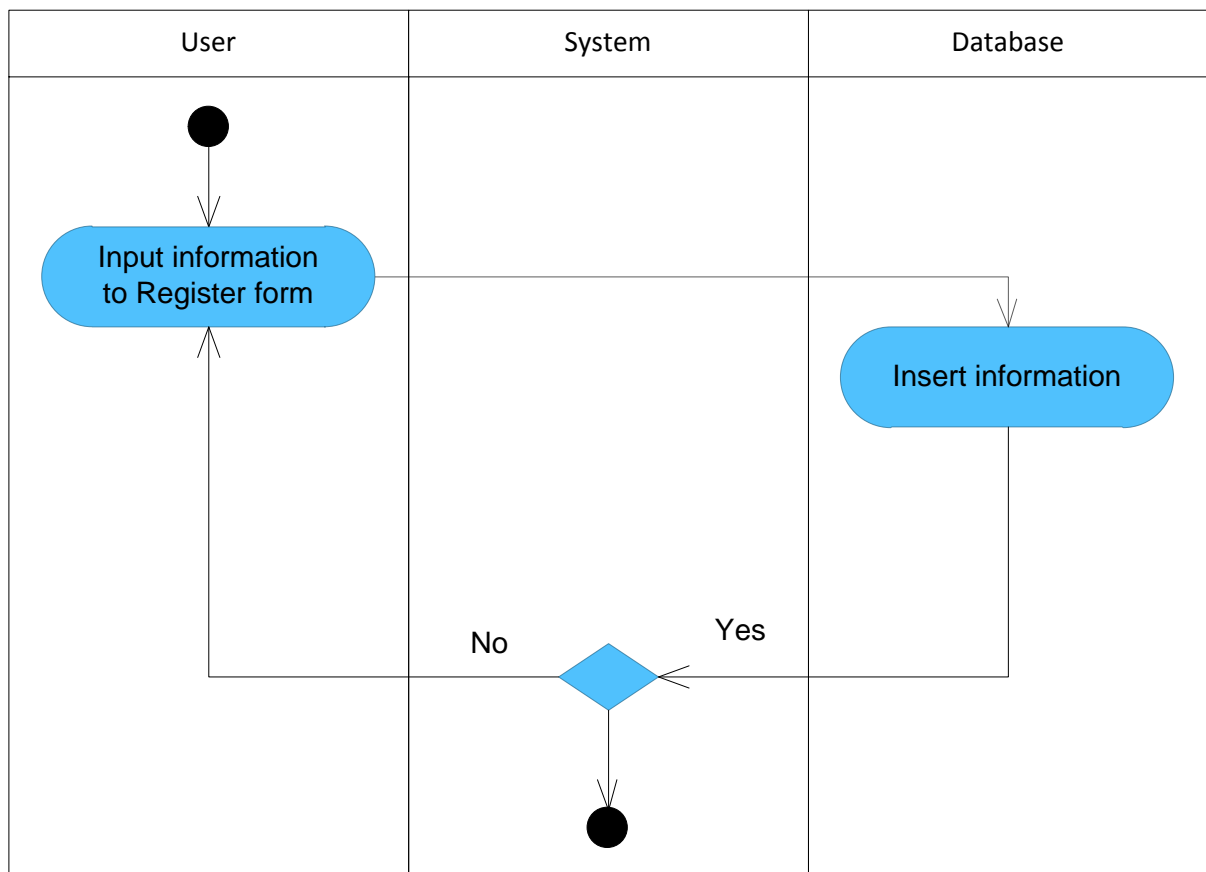
Figure 1 : Register Information

User	System	Database
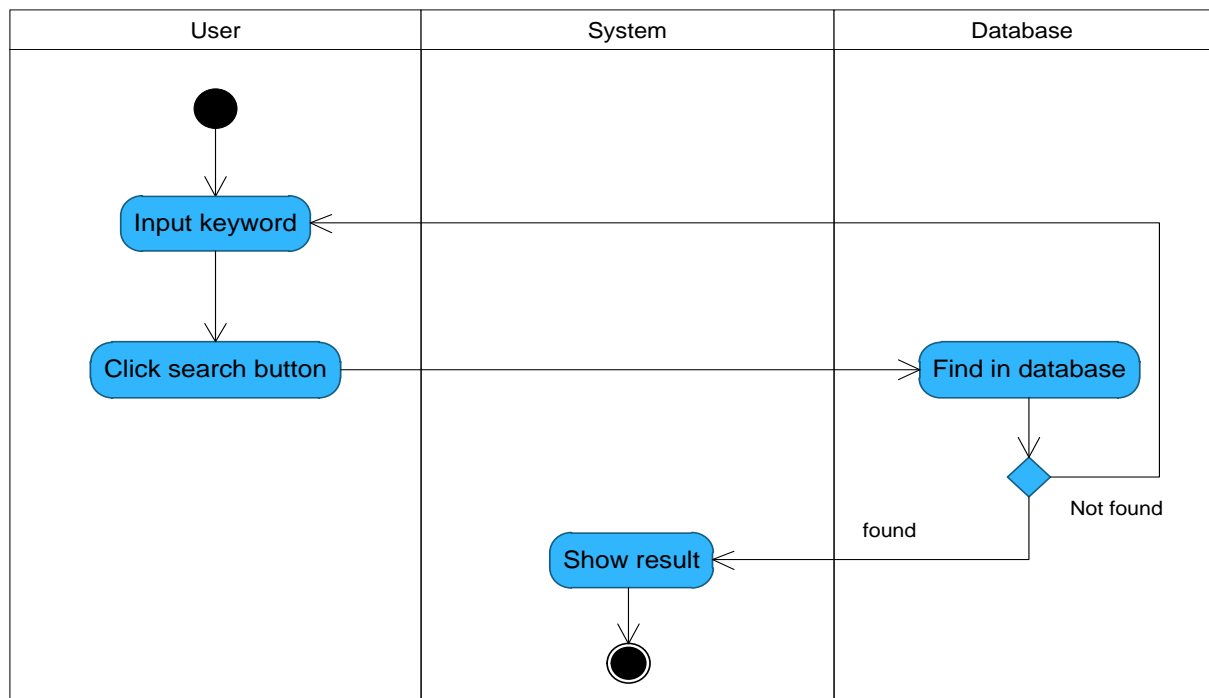
Figure 2 : Search infomation

User	System	Database
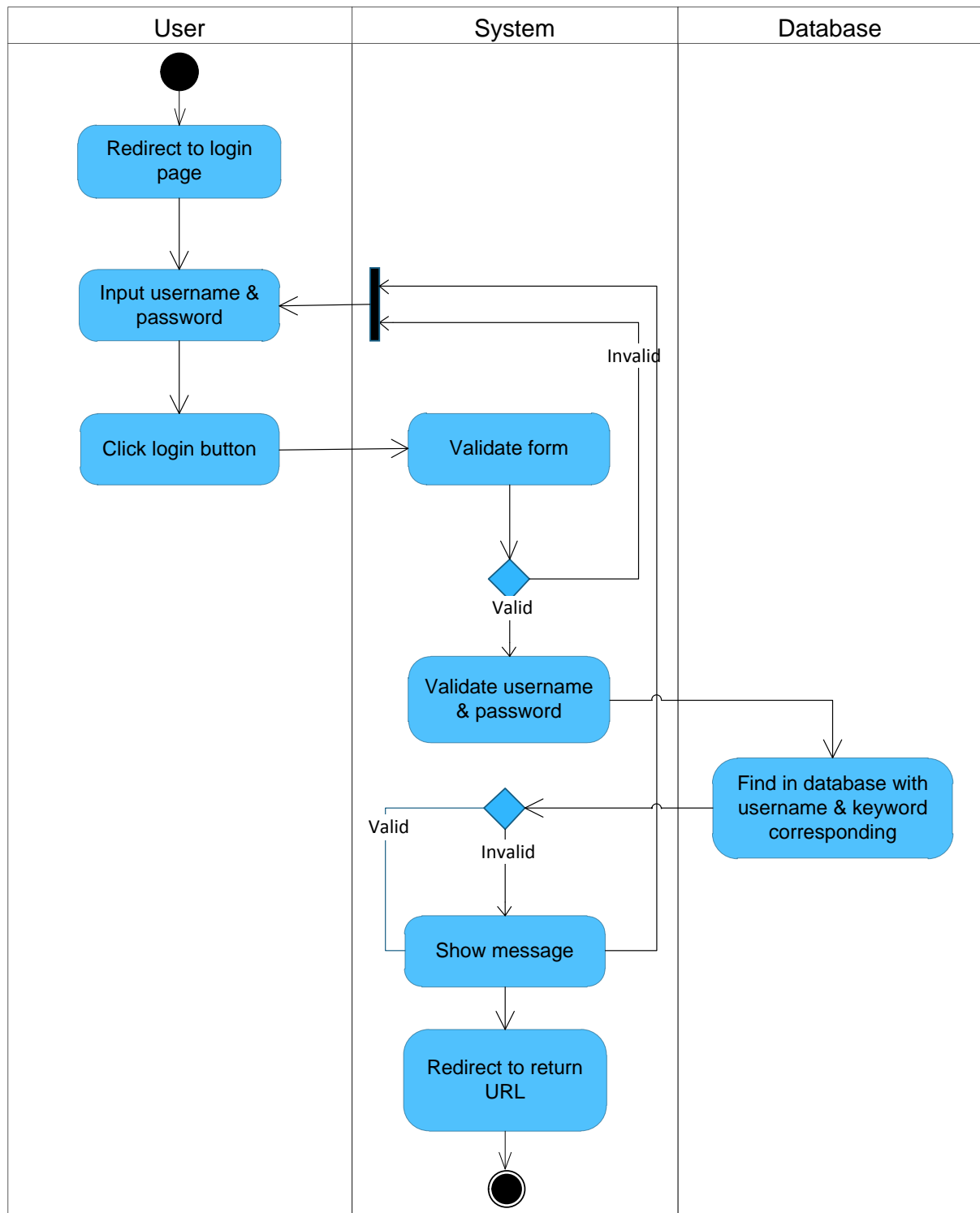
Figure 3 : Login to System
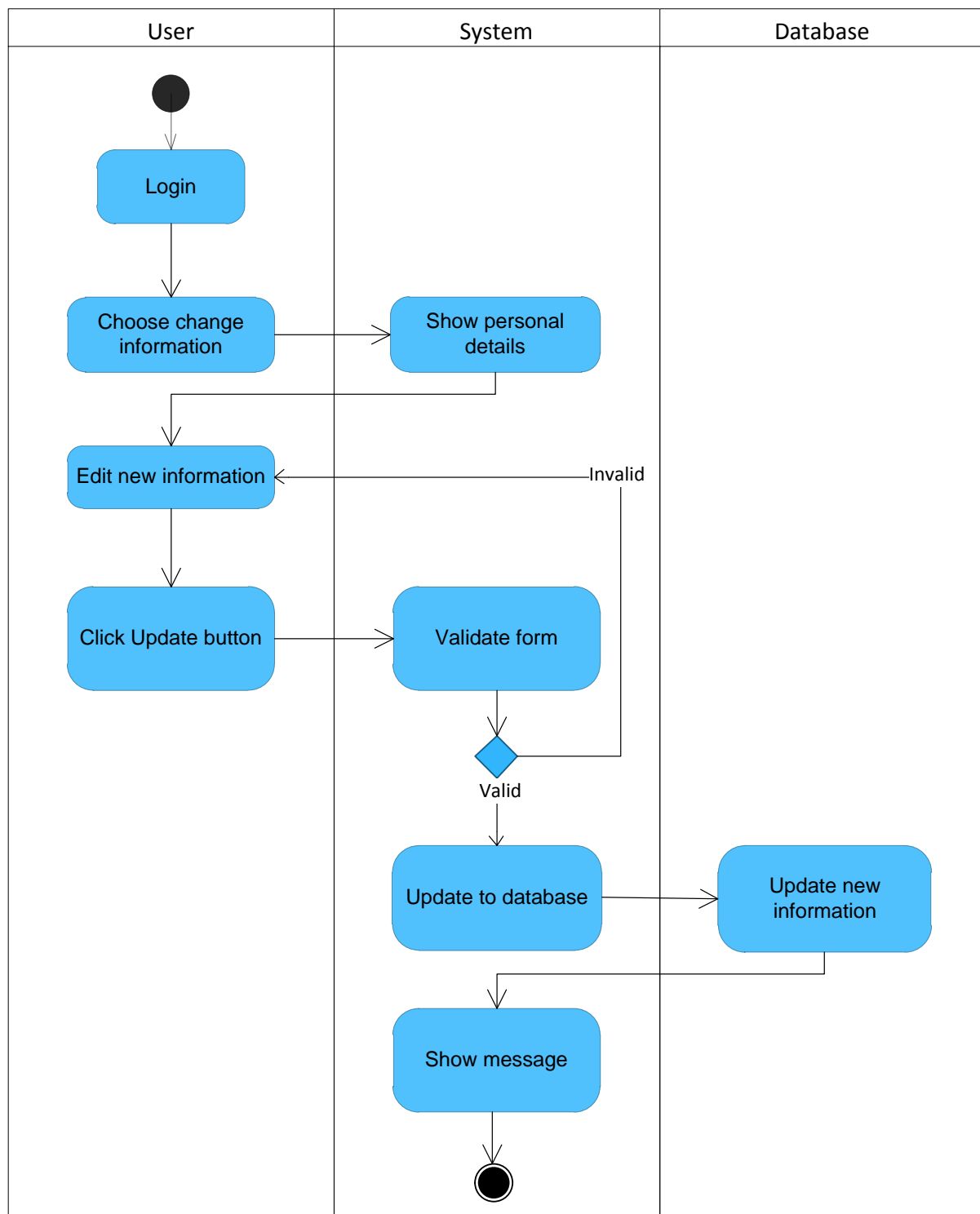
Figure 4 : Change Information
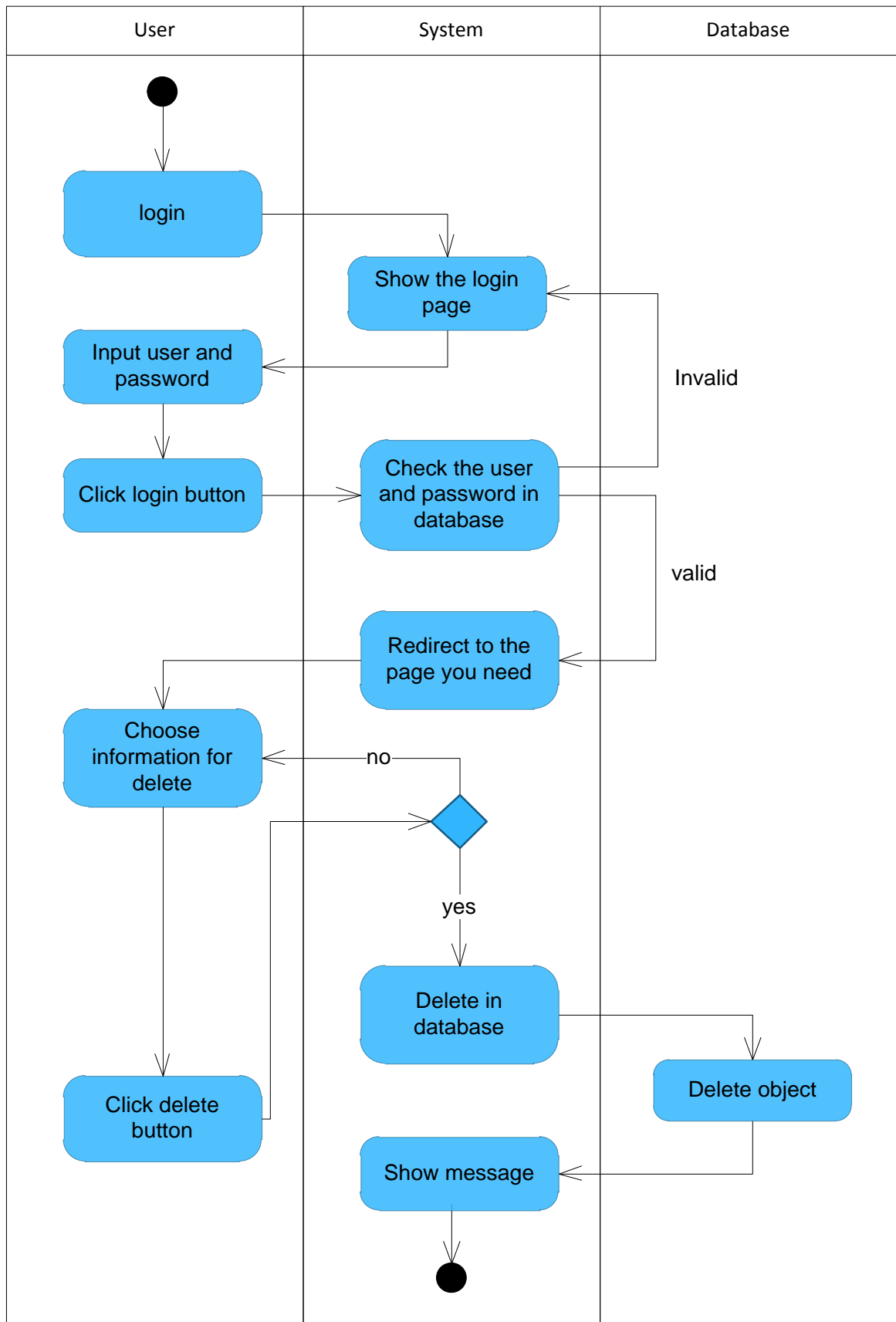
Figure 5 : Delete information

3.4 Use Case Diagram
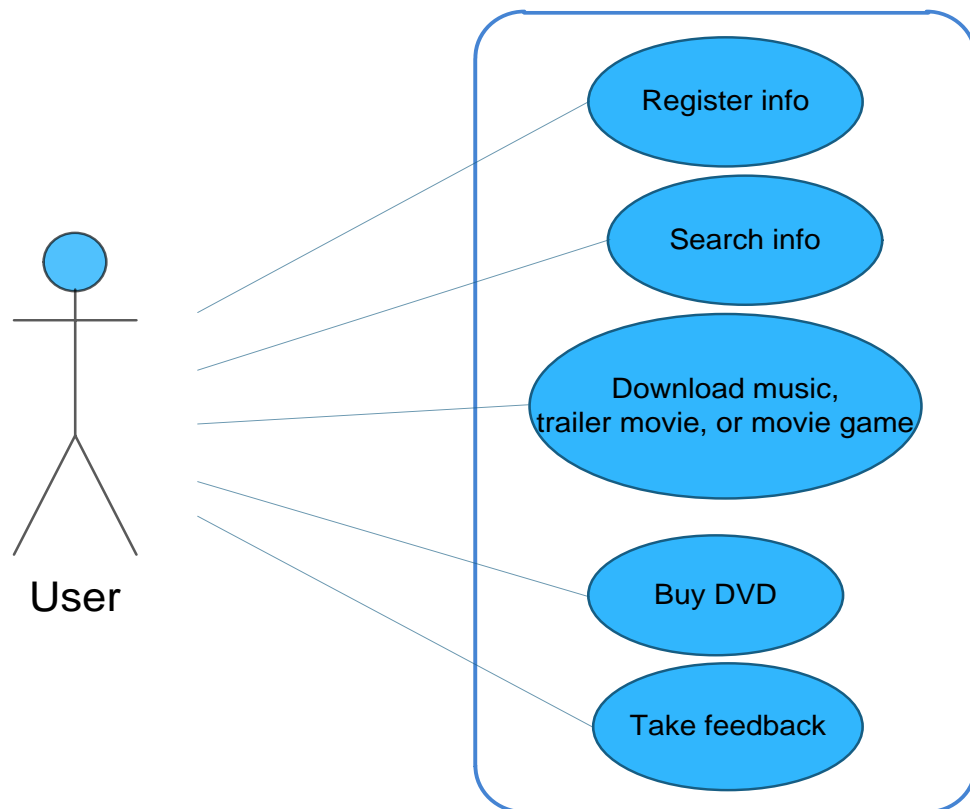
Figure 1: Use Case Diagram User
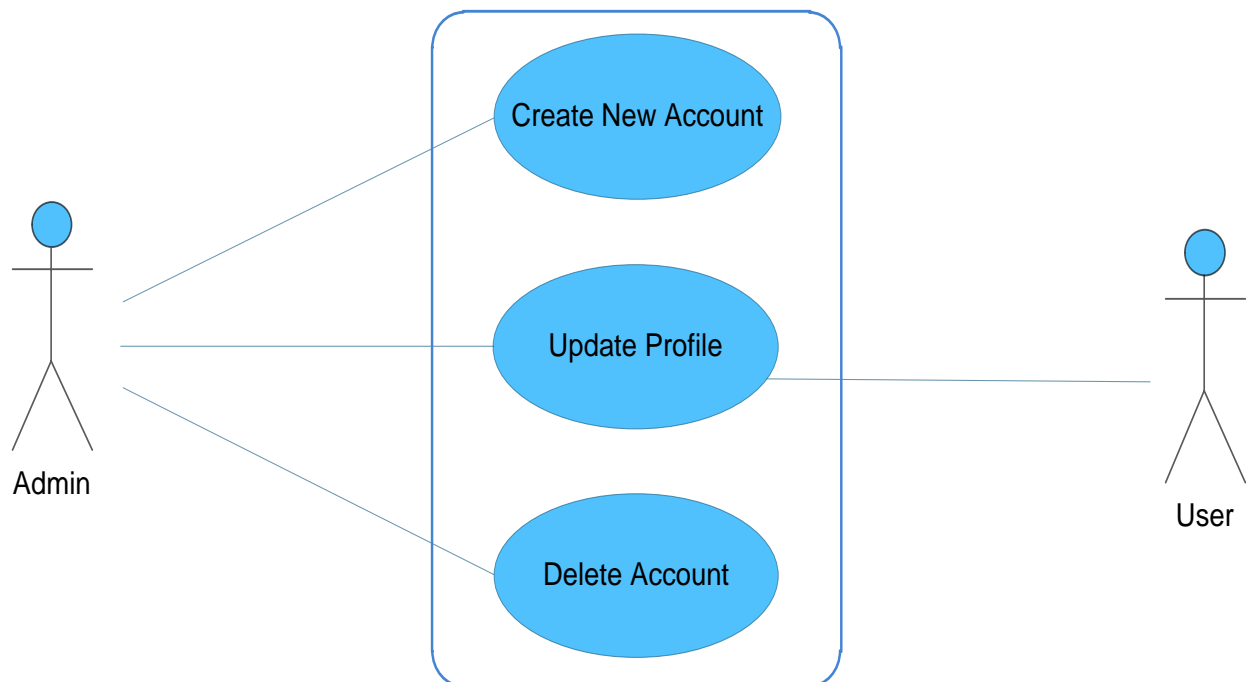
Figure 2: Account Management
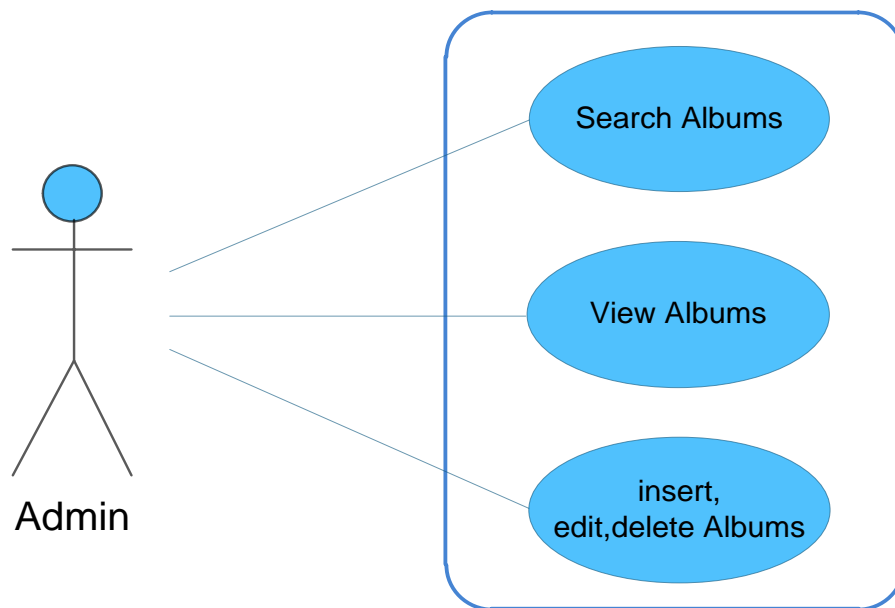
Figure 4: Albums Management
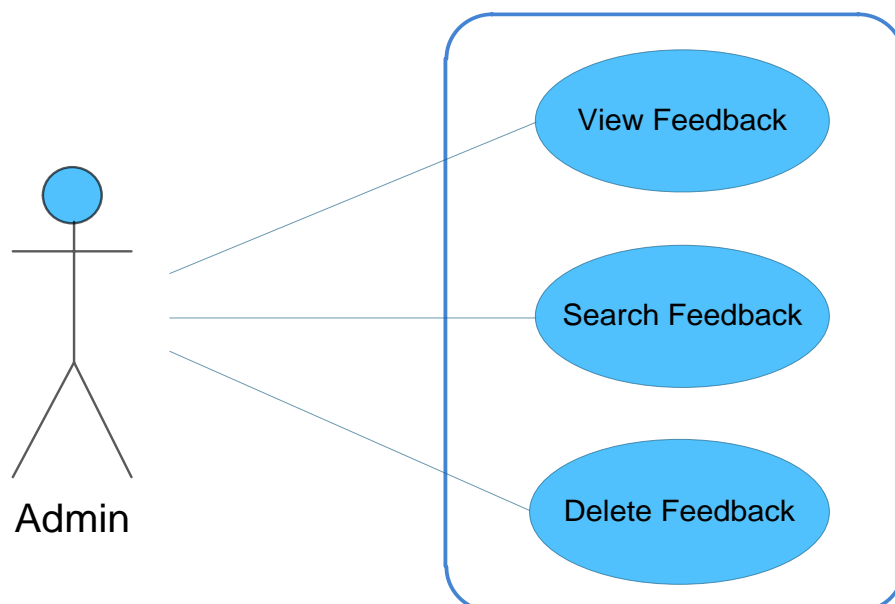
Figure 6: Feedback Management
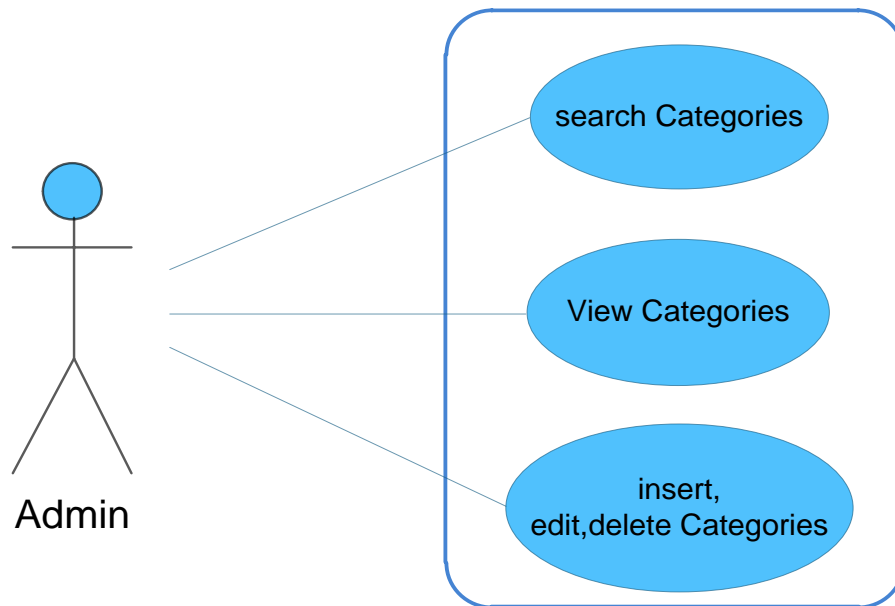
7. Manage Categories
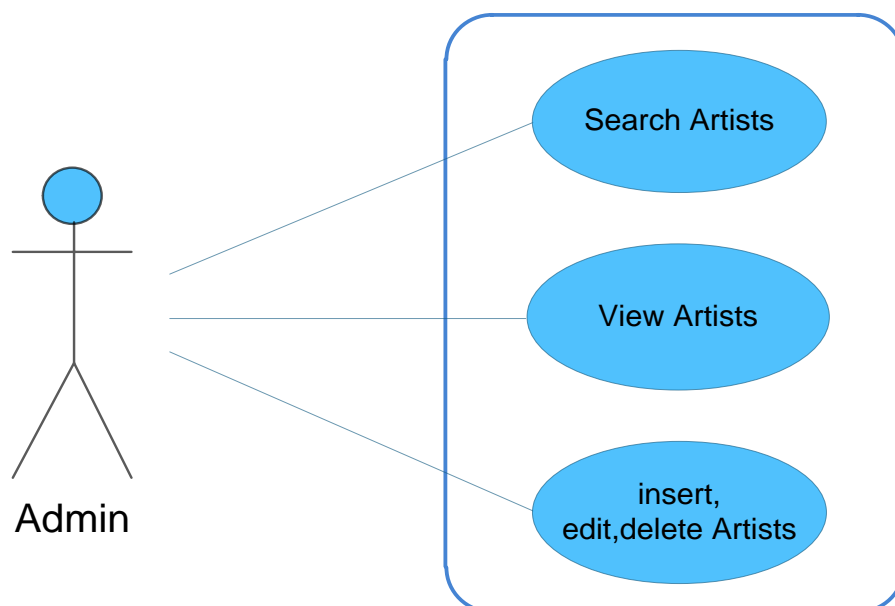

8.Manage Artists
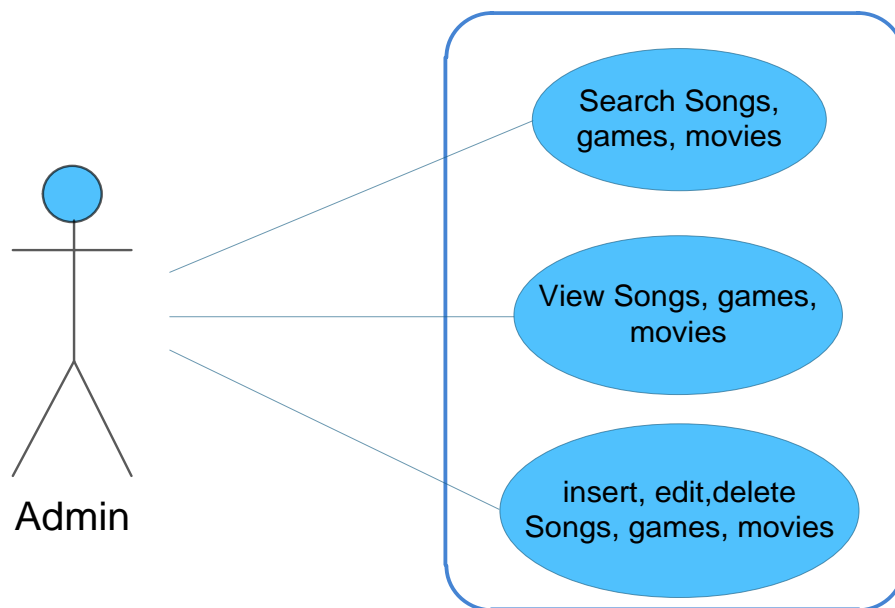
9.Manage Songs, games, movies
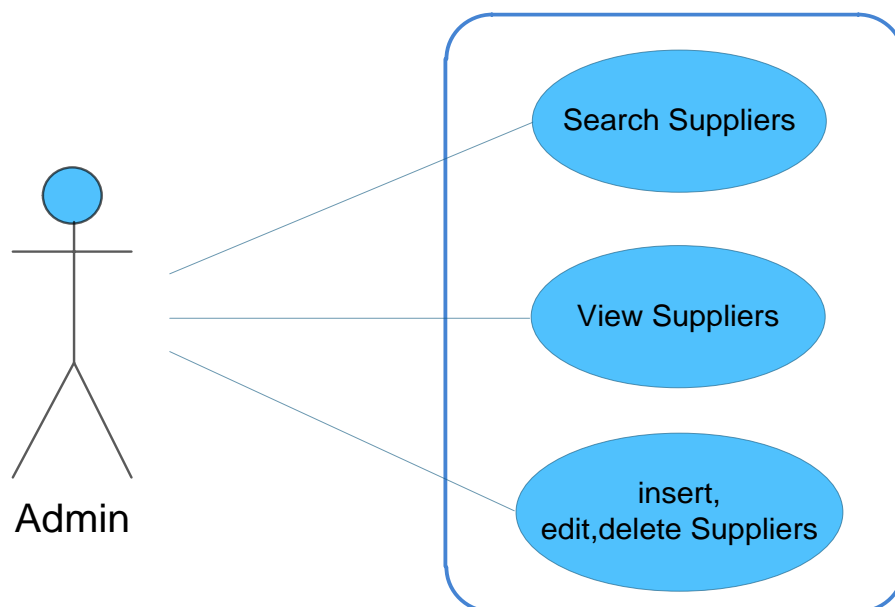
10. Manage Suppliers
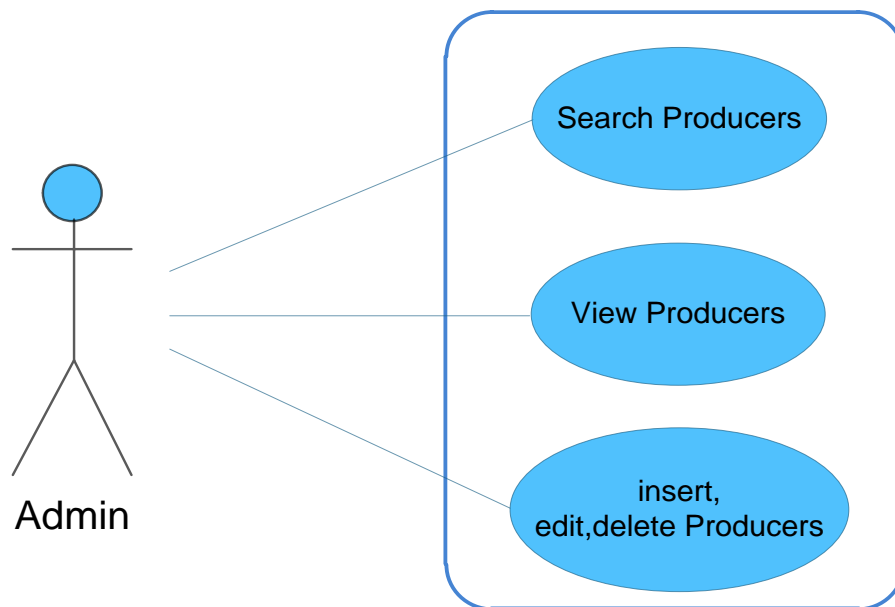
**10. Manage Producers**
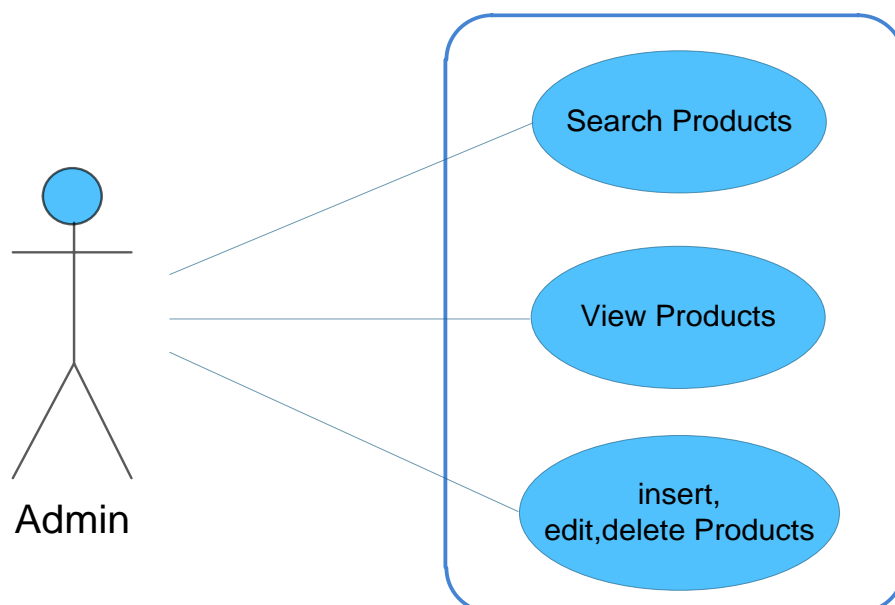

11. Manager Products
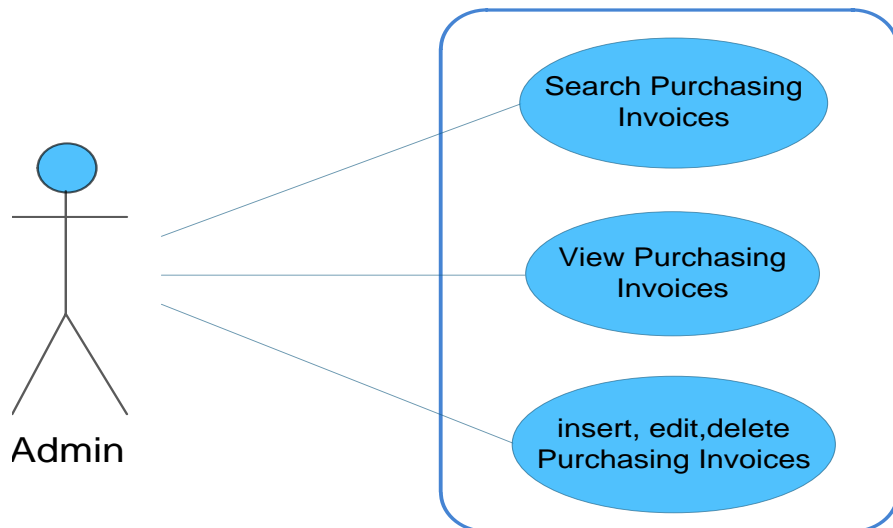
12. Manage Purchasing Invoices
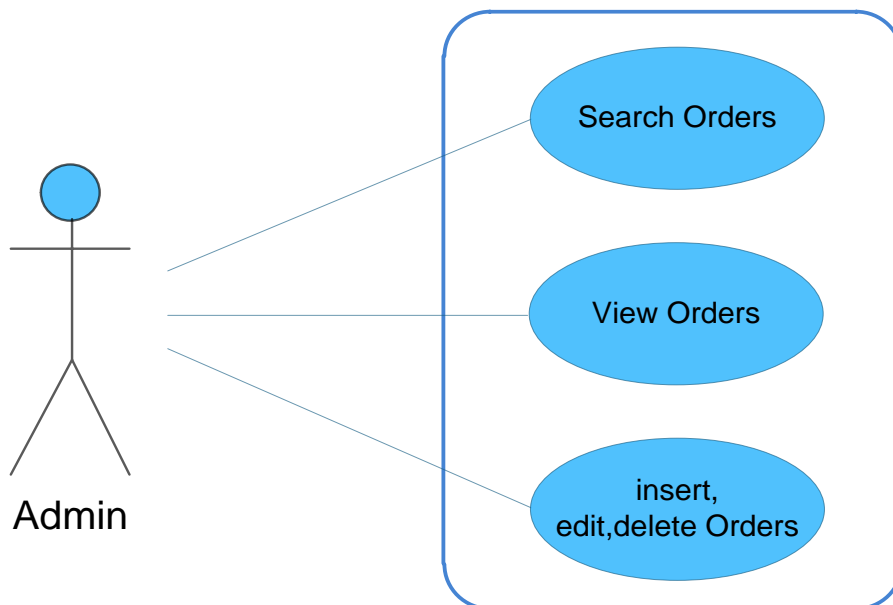
13. Manage Orders
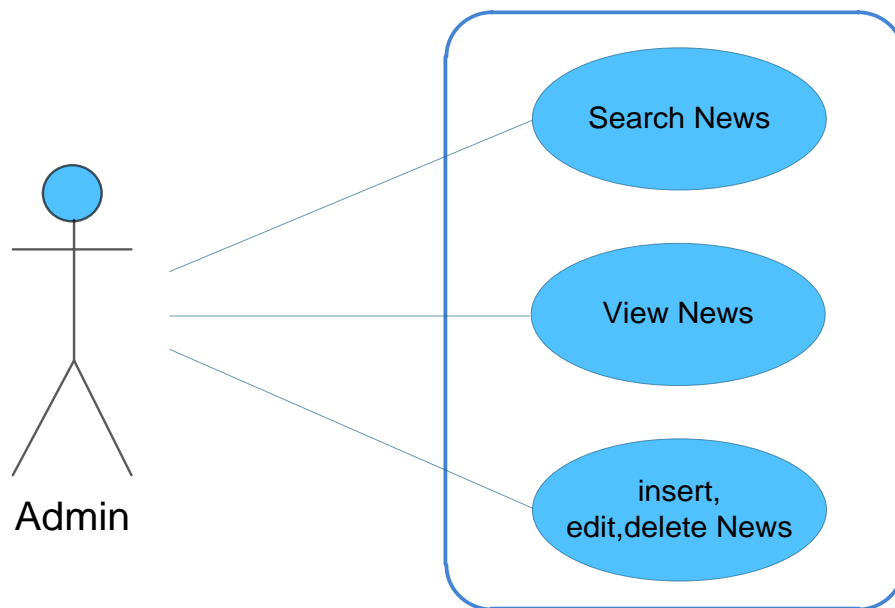
14. Manager News
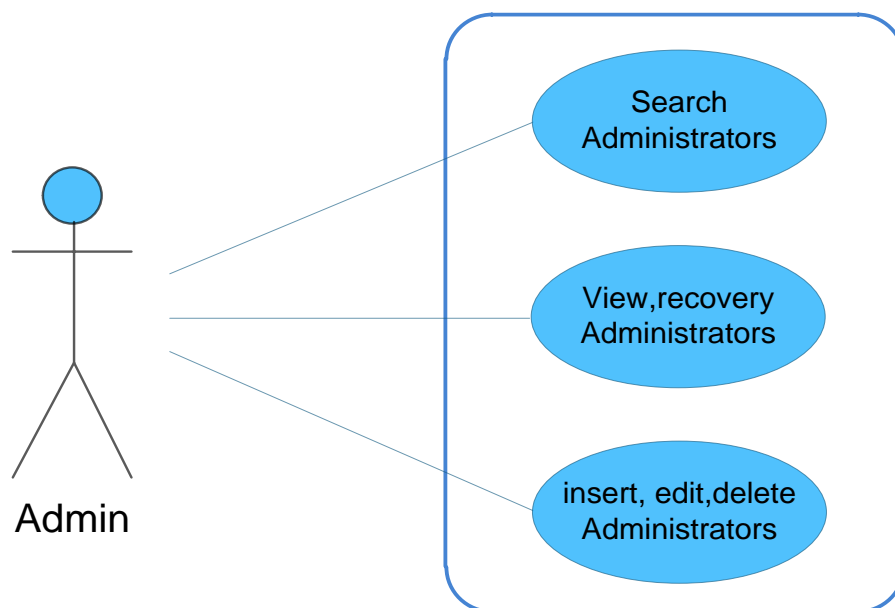
15. Manage Administrators
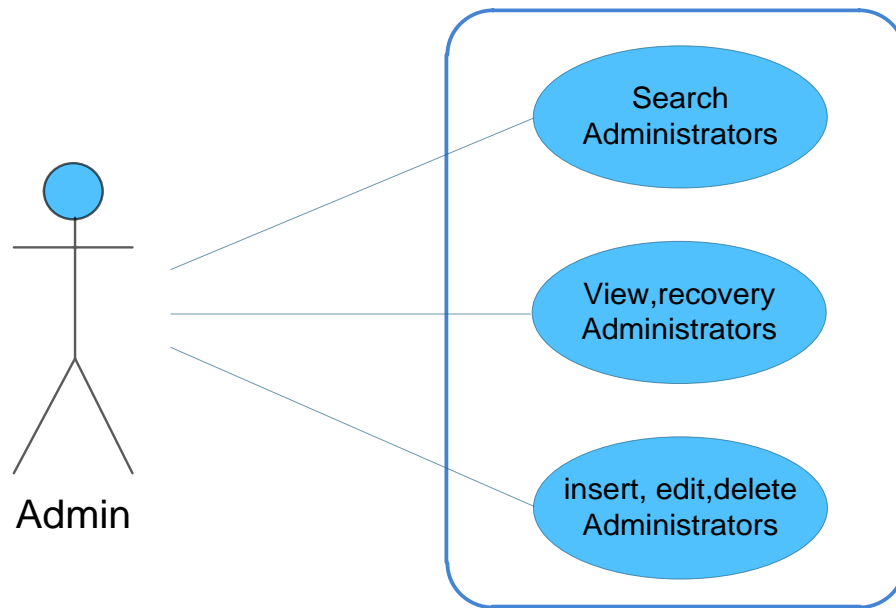
16. Manage Album Reviews

3.5. Use Case Specification

USE CASE #	UC01	
USE CASE Name	Register	
ACTOR	Visitor	
Purpose (1 phrase)	Register Information	
Overview and scope		
Level	Primary	
Preconditions		
Post conditions in words		
Trigger	User clicks **Register** button	
Included Use Cases	None	
Extended Use Cases	None	
MAIN SUCCESSFUL SCENARIO	**Actor Action**	**System Action**
	1. User enter information to Register Form	
		2. System will check if that information is correct or not. If it is correct, display a message Register successfully If it is not correct, ask User to enter again
		3. Display message
OTHER SUCCESSFUL SCENARIOS	**Step**	**Branching Action**
UNSUCCESSFUL SCENARIOS	**Conditions**	**Actions**
Developer		
Creation date and last modified date		

USE CASE #	UC02	
USE CASE Name	Login	
ACTOR	Admin, Users	
Purpose (1 phrase)	Login to system	
Overview and scope		
Level	Primary	
Preconditions	1. User has already had an account . 2. User needs to register.	
Post conditions in words		
Trigger	User click Login button	
Included Use Cases	None	
Extended Use Cases	None	
MAIN SUCCESSFUL SCENARIO	**Actor Action**	**System Action**
	1. User logins to system	
		2. System validates account
	3. User logins successfully	
		3. System shows a message to user
OTHER SUCCESSFUL SCENARIOS	**Step**	**Branching Action**
UNSUCCESSFUL SCENARIOS	**Conditions**	**Actions**
	1. User hasn't had an account yet. 2. User puts wrong username or password	1. User needs to register. 2. User retypes username or password
Developer		
Creation date and		

last modified date	

USE CASE #	UC03
USE CASE Name	listen to music, watch online trailer
ACTOR	Visitor, user
Purpose (1 phrase)	Allow user to play music, trailer movie, trailer game
Overview and scope	
Level	Primary
Preconditions	
Post conditions in words	
Trigger	Allow user to play music, trailer movie, trailer game
Included Use Cases	None
Extended Use Cases	None

	Actor Action	System Action
	1. vistor, users will click to the song, trailer game, trailer movie to watch.	
MAIN SUCCESSFUL SCENARIO		2. system will display the songs,trailer movie,trailer game that users want to see
OTHER SUCCESSFUL SCENARIOS	Step	Branching Action
UNSUCCESSFUL SCENARIOS	Conditions	Actions
Developer		
Creation date and last modified date		

USE CASE #	UC04	
USE CASE Name	Login	
ACTOR	Admin, Member	
Purpose (1 phrase)	Login to system	
Overview and scope		
Level	Primary	
Preconditions	1. User has already had an account . 2. User needs to register.	
Post conditions in words		
Trigger	User click Login button	
Included Use Cases	None	
Extended Use Cases	None	
MAIN SUCCESSFUL SCENARIO	**Actor Action**	**System Action**
	1. User logins to system	
		2. System validates account
	3. User logins successfully	
		3. System shows a message to user
OTHER SUCCESSFUL SCENARIOS	**Step**	**Branching Action**
UNSUCCESSFUL SCENARIOS	**Conditions**	**Actions**
	1. User hasn't had an account yet. 2. User puts wrong username or password	1. User needs to register. 2. User retypes username or password
Developer		
Creation date and		

last modified date	
USE CASE #	UC05
USE CASE Name	Change personal information
ACTOR	Admin
Purpose (1 phrase)	Allow to change personal information existing in the system
Overview and scope	
Level	Primary
Preconditions	1. User needs to login before changing information 2. User chooses information which are wanted to be change
Post conditions in words	System displays information which have been changed and updated to the database.
Trigger	User click Update button
Included Use Cases	None
Extended Use Cases	None

	Actor Action	**System Action**
	1. User logins to the system	
		2. System validates account
	3. User chooses information which are wanted to be changed	
MAIN SUCCESSFUL SCENARIO		4. System shows Edit information form
	5. User puts new information	
	6. User clicks Update button	
		7. System updates new information to the database
		8. System displays a message to User.
OTHER SUCCESSFUL SCENARIOS	**Step**	**Branching Action**

	Conditions	Actions
UNSUCCESSFUL SCENARIOS	System fails to authorize User account	System redirects to Login page and User needs to login to the system
	Information in form is invalid	User needs to retype information
Developer		
Creation date and last modified date		

USE CASE #	UC06
USE CASE Name	Delete information
ACTOR	Admin
Purpose (1 phrase)	Delete information
Overview and scope	
Level	Primary
Preconditions	1. Have account 2. Have insert all information about object
Post conditions in words	System display details information about object existed
Trigger	Clicks **Delete** button
Included Use Cases	None
Extended Use Cases	None

	Actor Action	System Action
MAIN SUCCESSFUL SCENARIO	1. login to system	
		2. System validates account
	3. go to information management	
	4. chooses an object to delete	.
	5. click delete button	
		6. System deletes information in database and return message
		7. System display result about object
OTHER SUCCESSFUL SCENARIOS	**Step**	**Branching Action**

UNSUCCESSFUL SCENARIOS	Conditions	Actions
Developer		
Creation date and last modified date		

USE CASE #	UC07	
USE CASE Name	Search information	
ACTOR	Admin, Member, Visitor	
Purpose (1 phrase)	Search information	
Overview and scope		
Level	Primary	
Preconditions	1. User choose options to search	
Post conditions in words		
Trigger	User click **Search** button	
Included Use Cases	None	
Extended Use Cases	None	
MAIN SUCCESSFUL SCENARIO	**Actor Action**	**System Action**
	1. In the search box, user choose options to search	
	2. User click Search button	

		3. System search information in database and return records corresponding.
		4. System display result
OTHER SUCCESSFUL SCENARIOS	**Step**	**Branching Action**
UNSUCCESSFUL SCENARIOS	**Conditions**	**Actions**
Developer		
Creation date and last modified date		

USE CASE #	UC08	
USE CASE Name	Manage order	
ACTOR	Admin	
Purpose (1 phrase)	Admin manage order	
Overview and scope		
Level	Primary	
Preconditions		
Post conditions in words	.	
Trigger	Users click **Send** button	
Included Use Cases	None	
Extended Use Cases	None	
MAIN SUCCESSFUL SCENARIO	**Actor Action**	**System Action**
	Admin click on the users order	system will have the functionality to manage the user's orders

OTHER SUCCESSFUL SCENARIOS	**Step**	**Branching Action**
UNSUCCESSFUL SCENARIOS	**Conditions**	**Actions**
	1. The information that Users entered is not correct	System display a message Information is not validate, so that Users have to enter again
Developer		
Creation date and last modified date		

4. Design

The application will be made of Web-based distributed three-tier architecture to support multiple user transaction at the same time.
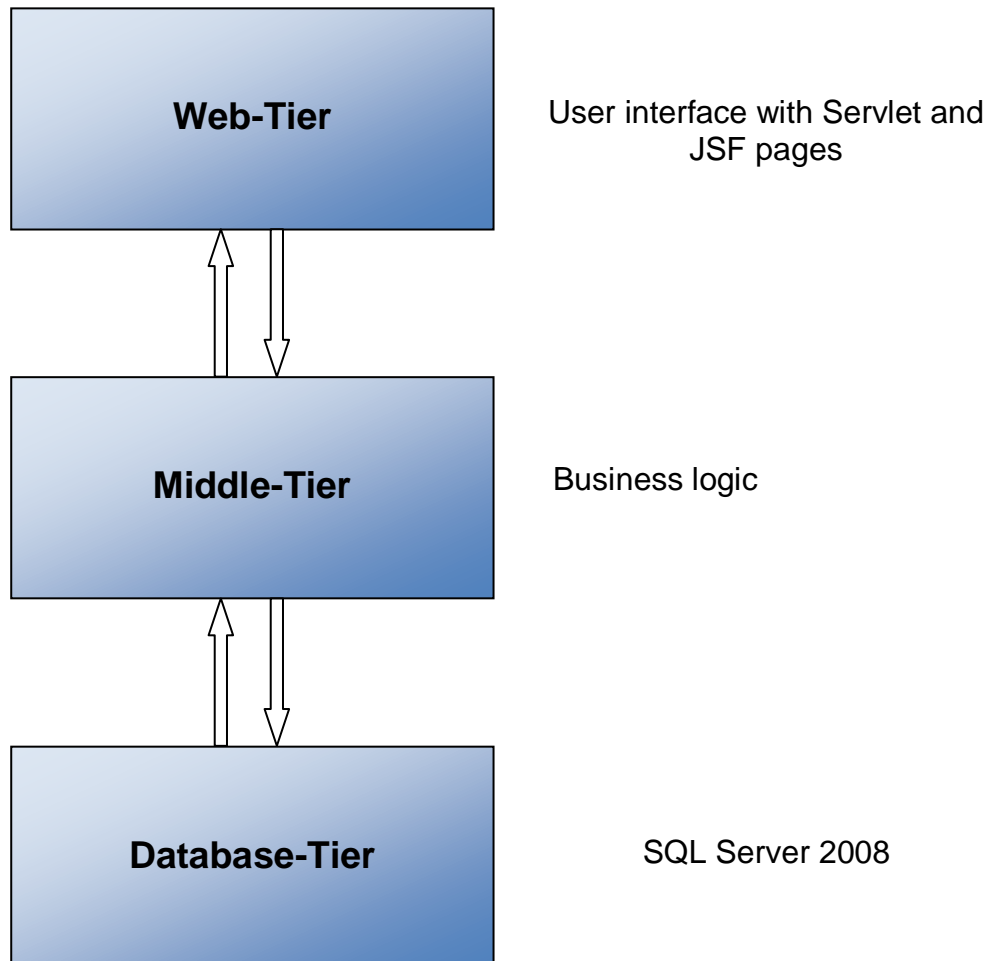
Web-Based Distributed 3-Tier Architecture of the Project

4.1. Database Diagram (DBD)
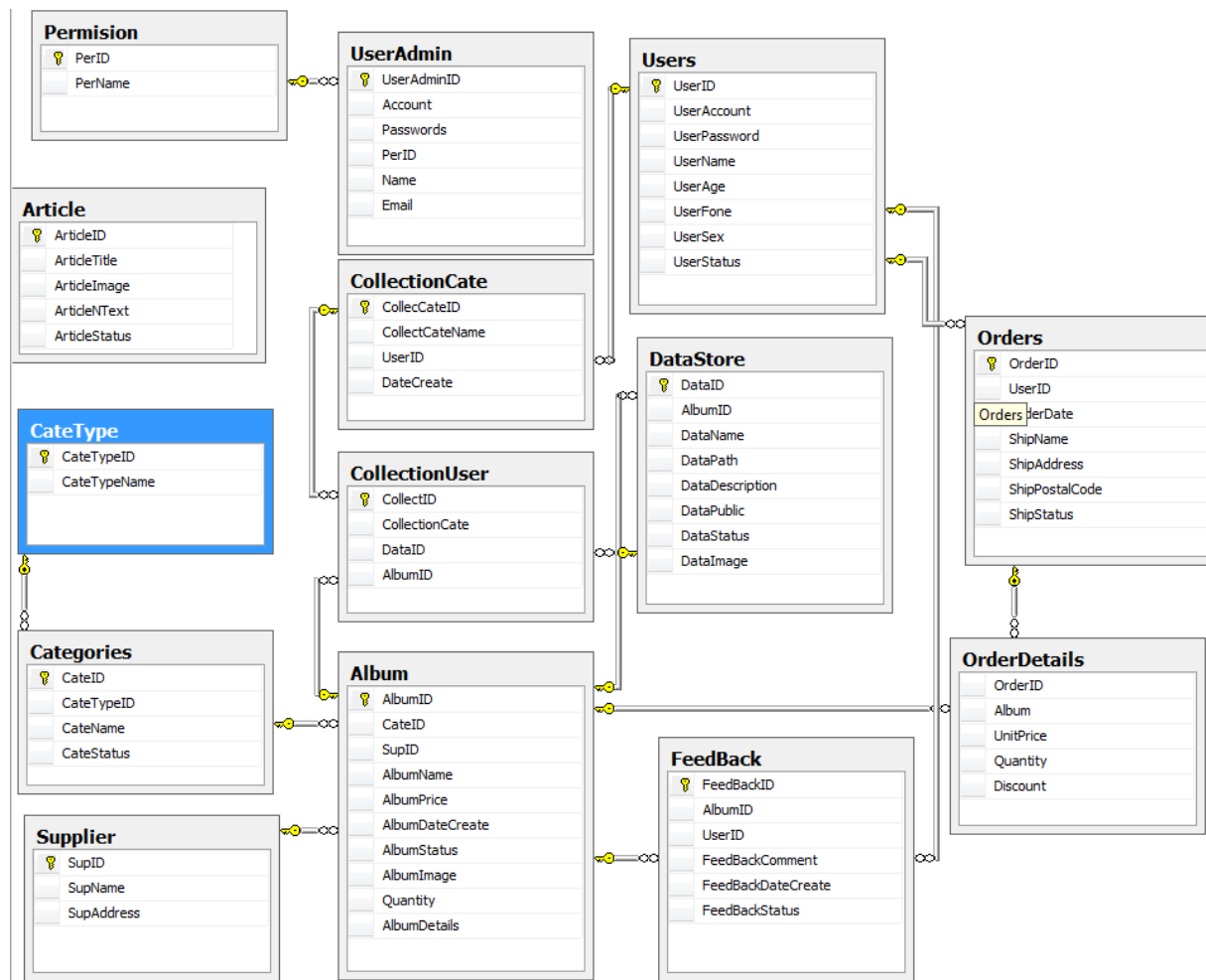
4.2. Database Design

Table Design

Table **CateType**			
File Name	**Data Type**	**Constraint**	**Description**
CateTypeID	int	Primary Key	ID
CateTypeName	Nvarchar(10)		Name of categories type

Table **Categories**			
File Name	**Data Type**	**Constraint**	**Description**
CateID	int	Primary key	ID
CateTypeID	Int	Foreign key	ID of Cate type
CateName	Nvarchar(35)		Name of categories
CateStatus	Bit		Status of categories

Table **Article**			
File Name	**Data Type**	**Constraint**	**Description**
ArticleID	int	Primary key	ID
ArticleTitle	Varchar(100)		Title of article
ArticleImage	NVARCHAR(100)		Images of Article
ArticleNText	NVARCHAR(Max)		
ArticleStatus	BIT		Status of article

Table **Supplier**

File Name	Data Type	Constraint	Description
SupID	Int	Primary key	Supplier Type
SupName	Varchar(50)		Supplier Name
SupAddress	Varchar(50)		Supplier address

Table **Album**

File Name	Data Type	Constraint	Description
AlbumID	Int	Primary key	Album ID
CateID	Int	Foreign key	Categories ID
SupID	Int	Foreign key	Supplier ID
AlbumName	NVARCHAR(200)		Album Name
AlbumPrice	DECIMAL		Album Price
AlbumDateCreate	NVARCHAR(30)		Album Date Create
AlbumStatus	BIT		Album Status
AlbumImage	NVARCHAR(100)		Album Image
Quantity	INT		Quantity
AlbumDetails	NVARCHAR(MAX)		Album Details

Table **DataStore**

File Name	Data Type	Constraint	Description
DataID	Int	Primary key	Data ID
AlbumID	Int	Foreign key	Album ID
DataName	NVARCHAR(100)		Data Name
DataPath	NVARCHAR(100)		Data Path
DataDescription	NVARCHAR(MAX)		Data Description
DataPublic	BIT		Data Public
DataStatus	BIT		Data Status
DataImage	NVARCHAR(200)		Data Image

Table **Users**

File Name	Data Type	Constraint	Description
UserID	Int	Primary key	User ID
UserAccount	NVARCHAR(20)		User Account
UserPassword	NVARCHAR(70)		User Password
UserName	NVARCHAR(30)		User Name
UserAge	NVARCHAR(5)		User Age
UserFone	NVARCHAR(20)		User Fone
UserSex	NVARCHAR(10)		User Sex
UserStatus	BIT		User Status

Table **CollectionCate**

File Name	Data Type	Constraint	Description
CollecCateID	Int	Primary key	CollectionCate ID
CollectCateName	NVARCHAR(100)		CollectionCate Name
UserID	Int	Foreign key	User ID
DateCreate	NVARCHAR(30)		Date Create

Table **CollectionUser**

File Name	Data Type	Constraint	Description
CollectID	Int	Primary key	Collection user ID
CollectCateID	Int	Foreign key	CollectionCate ID
DataID	Int	Foreign key	Data ID
AlbumID	Int	Foreign key	AlbumID

Table **Feedback**

File Name	Data Type	Constraint	Description
FeedBackID	INT	Primary key	FeedBack sID
AlbumID	INT	Foreign key	FeedBack ID
UserID	INT	Foreign key	User ID
FeedBackComment	NVARCHAR(MAX)		FeedBack Comment
FeedBackDateCreate	NVARCHAR(30)		FeedBack Date Create
FeedBackStatus	BIT		FeedBack Status

Table **Orders**

File Name	Data Type	Constraint	Description
OrderID	INT	Primary key	Order ID
UserID	INT	Foreign key	User ID
OrderDate	NVARCHAR(30)		Order Date
ShipName	NVARCHAR(30)		Ship Name
ShipAddress	NVARCHAR(100)		Ship Address
ShipPostalCode	NVARCHAR(20)		Ship Postal Code
ShipStatus	INT		Ship Status

Table OrderDetails

File Name	Data Type	Constraint	Description
OrderID	Int	Primary key	Order ID
AlbumID	Int	Foreign key	Album ID
UnitPrice	DECIMAL		Unit Price
Quantity	Int		Quantity
Discount	DECIMAL		Discount

Table Permission

File Name	Data Type	Constraint	Description
PerID	Int	Primary key	Permission ID
PerName	NVARCHAR(20)		Permission Name

Table UserAdmin

File Name	Data Type	Constraint	Description
UserAdminID	INT	Primary key	User Admin ID
Account	NVARCHAR(25)		Account
Passwords	NVARCHAR(100)		Passwords
PerID	INT	Foreign key	Permission ID
Name	NVARCHAR(50)		Name
Email	NVARCHAR(50)		Email

4.3. Algorithms

```
          ┌──────────────┐
          │    Start     │
          └──────┬───────┘
                 │
                 ▼
          ╱──────────────╲
         │    Enter       │◄──────────────────────────┐
         │ usename and    │                           │
         │  password      │                           │
          ╲──────────────╱                            │
                 │                                     │
                 ▼                                     │
              ◇ Check ◇        ▷ Invalid     ┌─────────────────┐
              ◇database◇ ─────────────────►  │ Show an error   │
              ◇       ◇                      │    message      │
                 │                           └─────────────────┘
                 ▼
              △ Valid
                 │
                 ▼
          ╱──────────────╲
         │    Access      │
         │    System      │
          ╲──────────────╱
                 │
                 ▼
          ┌──────────────┐
          │     End      │
          └──────────────┘
```

Flow Chart for Login

Flow Chart for Regiser Information

```
                          ┌─────────────┐
                          │    Start    │
                          └─────────────┘
                                 │
                                 ▼
                        ╱──────────────╲
                       ╱  Select object ╲◄──────────────────┐
                       ╲                ╱                    │
                        ╲──────────────╱                     │
                                 │                           │
                                 ▼                           │
                       ┌──────────────────┐                 │
                       │ Update information│                 │
                       └──────────────────┘                 │
                                 │                           │
                                 ▼                           │
                              ◇◇◇◇◇                          │
                           ◇         ◇                       │
                          ◇   Check    ◇    ──►  ▲Invalid  ──► ┌──────────────┐
                          ◇ information ◇                     │ Show an error │
                           ◇         ◇                        │   message     │
                              ◇◇◇◇◇                           └──────────────┘
                                 │
                                 ▼
                               ▲
                              Valid
                                 │
                                 ▼
                        ╱──────────────╲
                       ╱     Update      ╲
                       ╲   successful     ╱
                        ╲──────────────╱
                                 │
                                 ▼
                          ┌─────────────┐
                          │     End     │
                          └─────────────┘
```
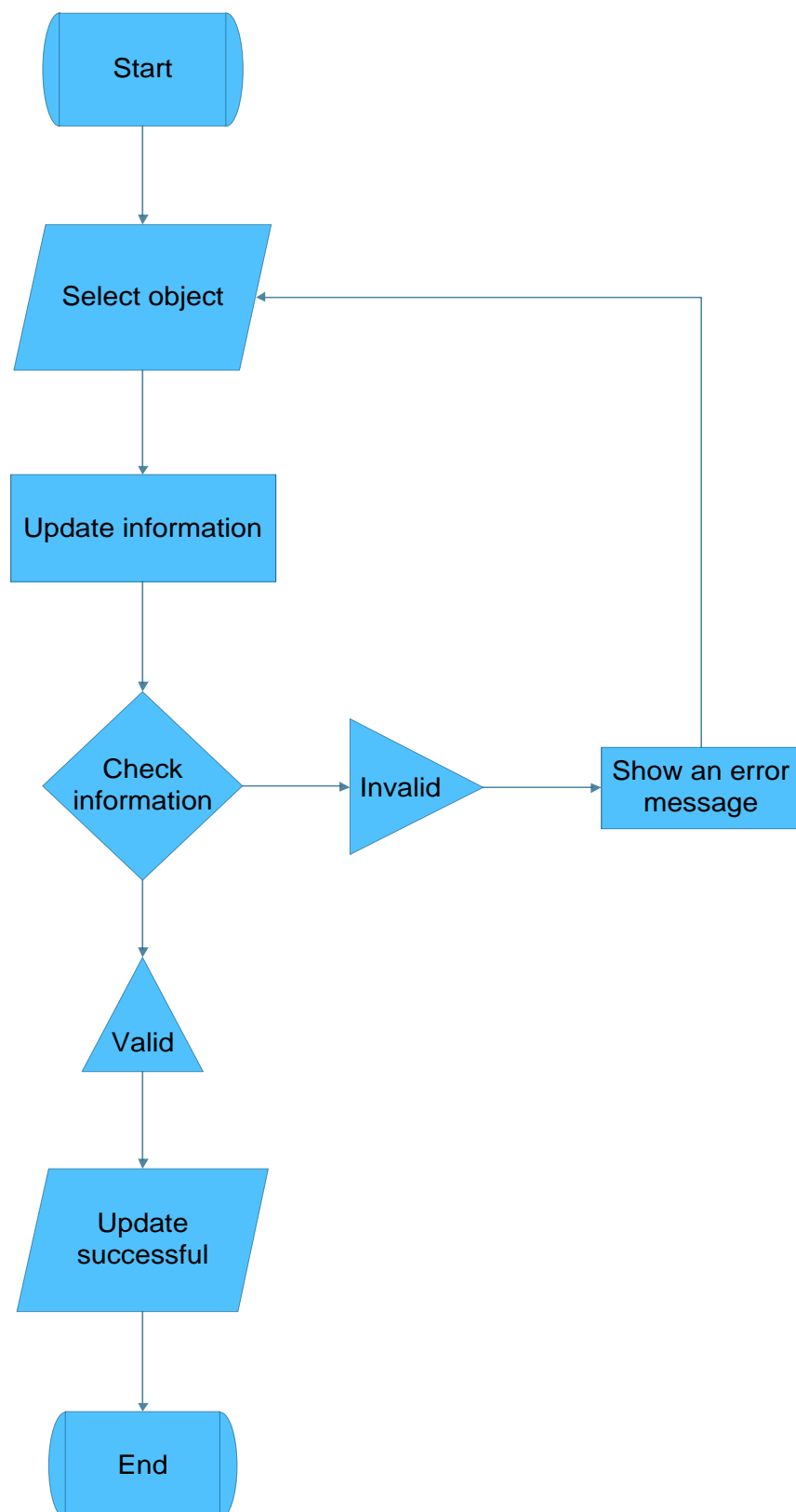
Flow Chart for Update Information

Start

Select object

Find Information to delete

Check information

Invalid

Show an error message

Valid

Delete successful

End
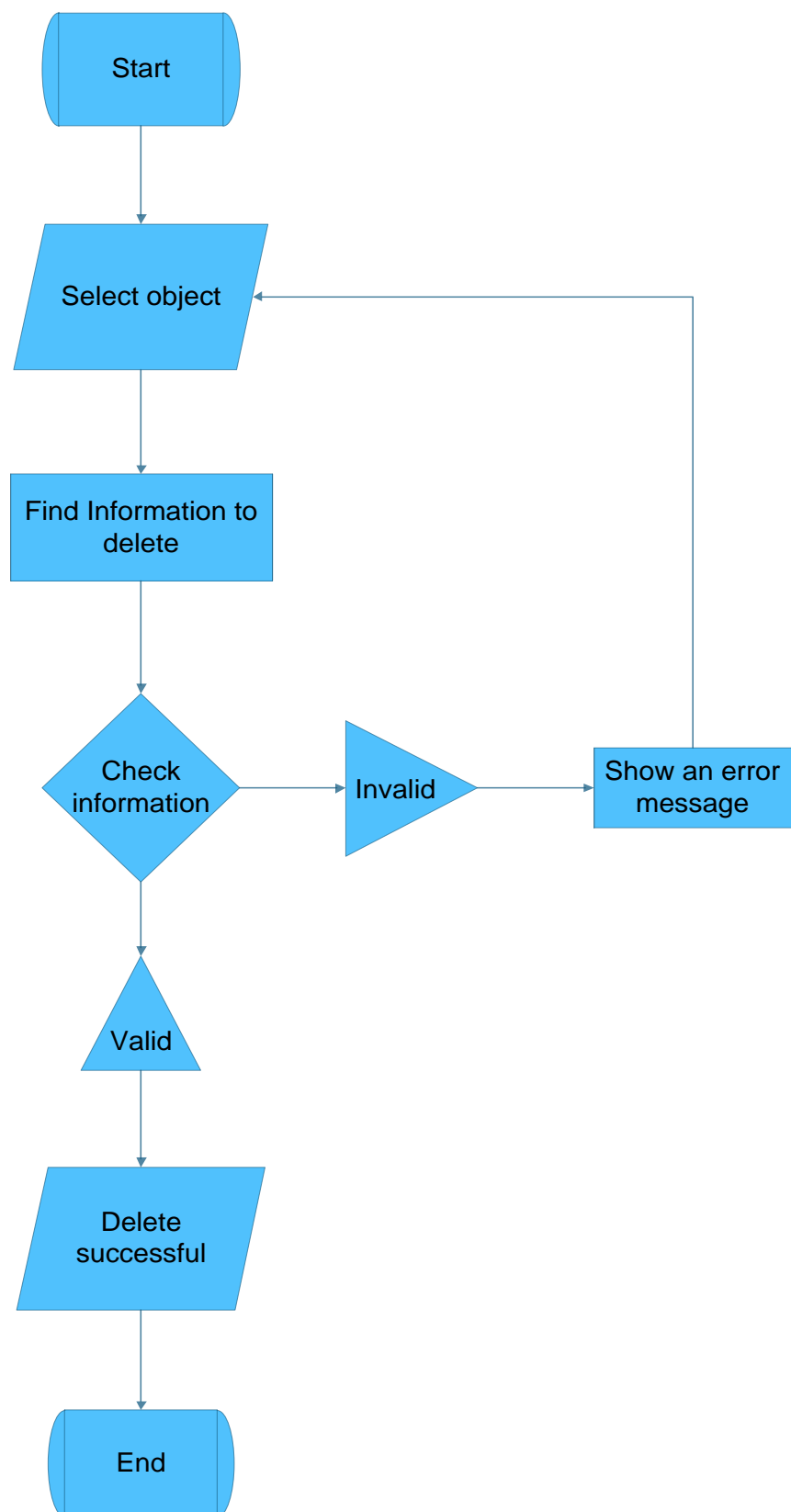
Flow Chart for Delete Information
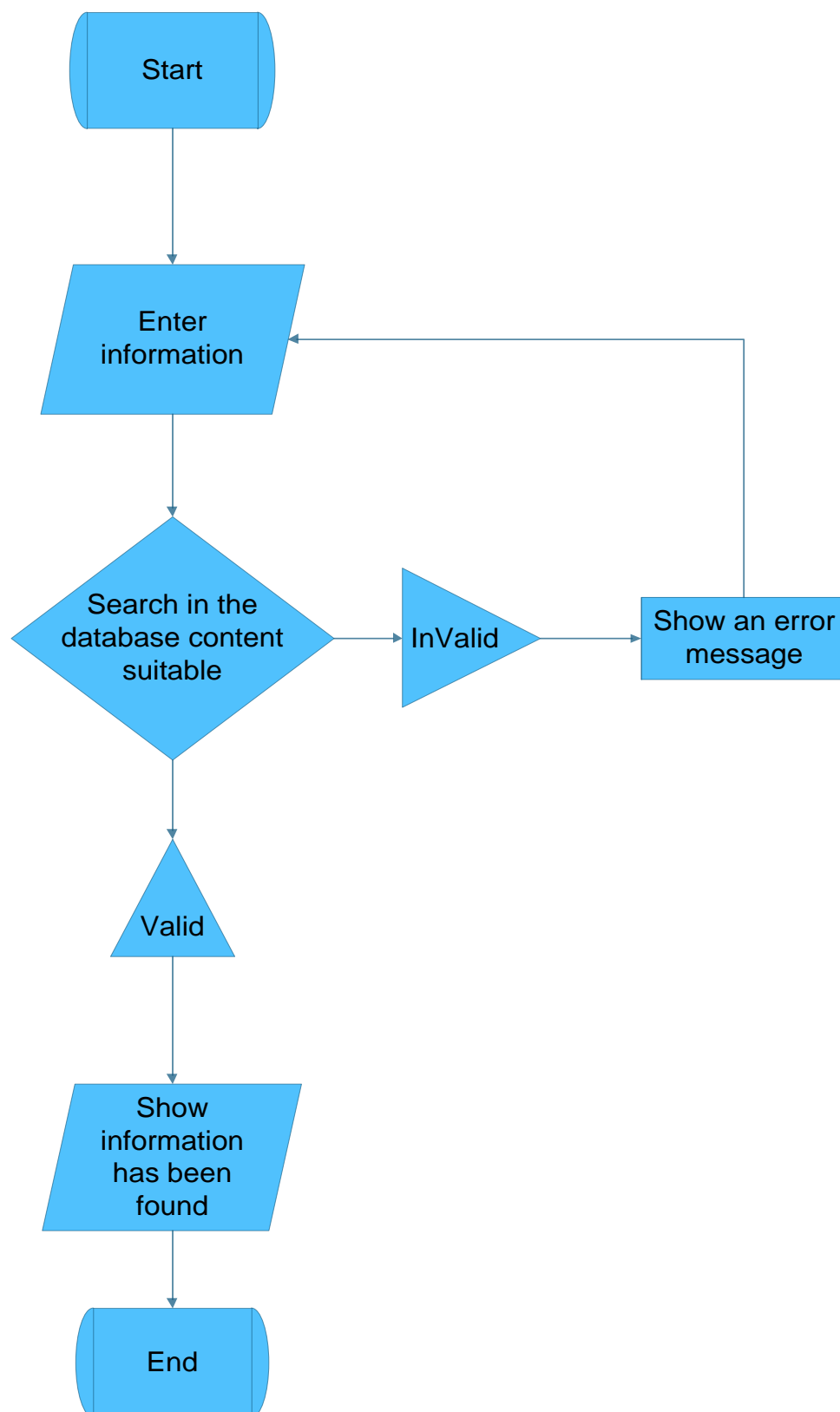
Flow Chart for Search Informatio

Start

Input information into feedback form

Check info → Invalid → Show an error message

Valid

Add feedback to the database

Display information has been sended

End
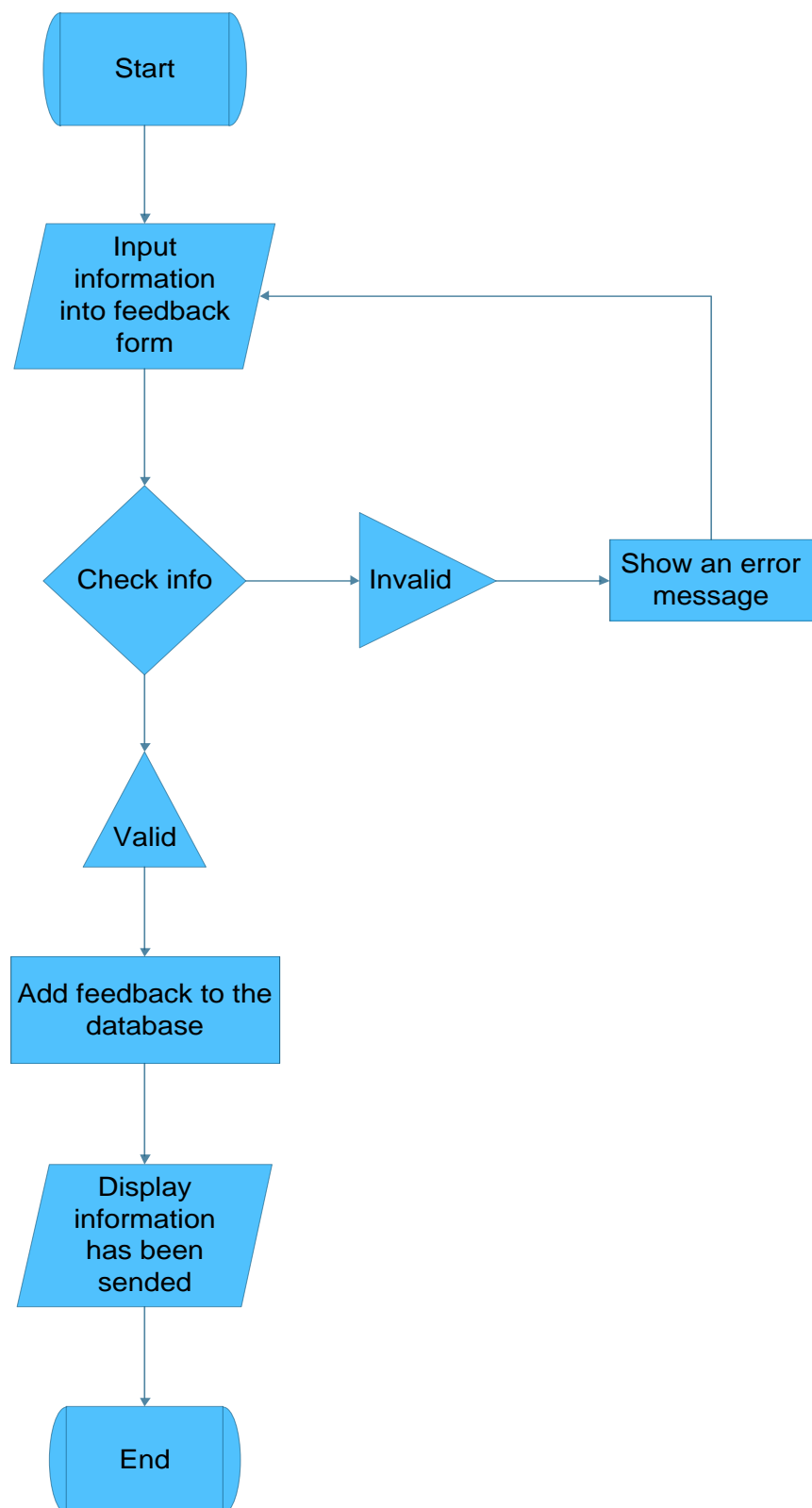
Flow Chart for Take Feedback

5. Management and Project Planning

5.1. Management Approach

The group members using PMS to manage the project code, assign tasks to team members and TortoiseSVN tool. When meeting in class, members tell the difficult issues and work together to solve that. In addition, members also use skype and yahoo to discuss with each other.

5.2. Task Sheet

Stt	Member Name	Task
1	Phung Van Thuc	
2	Vu Hoang Chien	
3	Dinh Ngoc Hung	

6. Screenshots

Figure: Home Page

Figure: Register Page

Figure: Default card

Figure : Personal Authentication Information

Figure : Preview Order

Figure : History Order

Figure : Home dashboard

Figure : Categories

Figure : data edit

Figure : Show data

Figure : Add new dics

Figure : Edit disc

Figure : Disc show

Figure : show orders

Figure : View orders details

Figure : View users

Figure : Users

Figure : Login

Figure : notice order pending