

**Cinema Website**

**Phạm Quốc Nghị(Leader) - CE140179**

**Lê Thành Nhân - CE140044**

**Hứa Thiên Ngân - CE140552**

**Nguyễn Quốc Toàn - CE140110**

**Mentor: Lương Hoàng Hướng**

**Group 10**

1. Trương Thị Thanh Xuân – CE140149
2. Trương Nhật Nam – CE140370
3. Nguyễn Quốc An – CE140024

Mentor Luong Hoang Huong



**Asterism Cinema**

Table of Contents

[I. Problem Definition 3](#_Toc8908)

[II. User Requirement 4](#_Toc19697)

[1. Input 4](#_Toc25941)

[2. Output 4](#_Toc3426)

[3. List of Funtion 4](#_Toc24699)

[III. Hardware and Software Requirement 5](#_Toc26788)

[1. System requirement 5](#_Toc25152)

[2. Required Software 5](#_Toc15316)

[IV. JSP 6](#_Toc9676)

[1. What is JSP? 6](#_Toc26108)

[2. Advantages of JSP over Servlet 6](#_Toc17558)

[3. The Lifecycle of a JSP Page 6](#_Toc16123)

[V. MVC 7](#_Toc29216)

[1. What is MVC? 7](#_Toc745)

[2. The advantages of MVC 7](#_Toc24149)

[VI. Java Spring 8](#_Toc15725)

[1. What is Java Spring? 8](#_Toc15363)

[2. The advantages of Java Spring 8](#_Toc23765)

[VII. Detail 9](#_Toc5137)

[1. Date of project plan 9](#_Toc4858)

[2. project’s Vision/Objective 9](#_Toc4948)

[3. Project Initiation/Requirement Document 9](#_Toc6708)

[4. Plan 9](#_Toc12347)

[VIII. Roles 10](#_Toc26338)

[1. Table roles 10](#_Toc11943)

[2. Gantt 10](#_Toc21084)

[IX. Schedule 11](#_Toc31651)

[X. Design Pattern 12](#_Toc29254)

[1. Data base Diagram 12](#_Toc26599)

[2. Class Diagram 12](#_Toc10135)

[3. Activity diagram 12](#_Toc16188)

[4. Usercase diagram 12](#_Toc2781)

[XI. Program Interface 13](#_Toc27712)

1. Problem Definition

Nowadays, the ongoing development of technology affects human life so much. It is not only changing the thinking, behavior, but also people demand. Specifically, the Internet provides a series of useful utilities for accessing, searching, entertaining and so on... that is reason why we are being attracted by social medias and waste a lot of time. The plenty of company found the above problems, so the development of each business would be parallel with the development of their websites. In this project we are going to develope a website for selling cinema tickets with Java web application foundation. In this website we are trying to design interfaces that attractted user and ensure website information of customer is secure.

1. User Requirement
2. Input

* A movie ticket website, for users to display movie listings, movie showtimes, select seats, sign in to buy tickets and see the bill.
* For admin, there are features such as adding movie, adding showtimes, changing user status, redirecting to an admin's own page when logging in.

1. Output

* When a user makes a reservation all seat information and a reservation are saved and the database.
* When the admin adds a movie, all movie information will be stored in the database and displayed on the website

1. List of Funtion

* User
* Login and register for the user
* See movie information
* See movie showtimes
* Choose a seat to book a ticket
* Fill in the information to pay for tickets
* show bill
* show bill list
* Admin
* Log in for the admin
* See user list
* Change user permissions
* change user status
* Updated movie information
* Extra showtimes for movies

1. Hardware and Software Requirement
2. System requirement
3. Required Software

* NetBeans IDE runs on the Java SE Development Kit (JDK) which consists of the Java Runtime Environment and developer tools for compiling, debugging, and running applications written in the Java language.
* The tested JDK for this release is JDK 8u101 for Windows, Linux, and OS X. The 8.2 version of the IDE cannot be installed or run on the JDK older than JDK 8.
* Note:
* The PHP and C/C++ NetBeans bundles only require the Java Runtime Environment (JRE) 8 to be installed and run.
* Java features in the IDE and JavaFX 8 features require JDK 8.
* Download Tomcat for webserver, XAMPP for database.

1. JSP
2. What is JSP?

* JavaServer Pages (JSP) is a Java standard technology that enables you to write dynamic, data-driven pages for your Java web applications. JSP is built on top of the Java Servlet specification. The two technologies typically work together, especially in older Java web applications. From a coding perspective, the most obvious difference between them is that with servlets you write Java code and then embed client-side markup (like HTML) into that code, whereas with JSP you start with the client-side script or markup, then embed JSP tags to connect your page to the Java backend.
* JSP is also closely related to JSF (JavaServer Faces), a Java specification for building MVC (model-view-controller) web applications. JSP is a relatively simpler and older technology than JSF, which is the standard for Java web frameworks like Eclipse Mojarra, MyFaces, and PrimeFaces. While it is not uncommon to see JSP used as the frontend for older JSF applications, Facelets is the preferred view technology for modern JSF implementations.
* While JSP may not be your first choice for building dynamic web pages, it is a core Java web technology. JSP pages are relatively quick and easy to build, and they interact seamlessly with Java servlets in a servlet container like Tomcat. You will encounter JSP in older Java web applications, and from time to time you may find it useful for building simple, dynamic Java web pages. As a Java developer, you should at least be familiar with JSP.

1. Advantages of JSP over other Technologies

* Active Server Pages (ASP) : ASP is a Microsoft technology. The dynamic part of JSP is written in Java, so it is more powerful and easier to use. Secondly, JSP is platform independent whereas ASP is not.
* Pure Servlets : It is more convenient to write regular HTML than to have println statements that generate HTML. Allows separation of look from the content. In a JSP web designer can design web page separately and servlet programmers can insert the dynamic content separately.
* Server-Side Includes (SSI) : SSI is widely supported technology for including externally defined pieces into a static web page. JSP is better because it lets you use servlets instead of a separate program to generate that dynamic part. Besides, SSI is really only intended for simple inclusions, not for real programs that use form data, make database connection.
* JavaScript : JavaScript can generate HTML dynamically on the client. However, it can only access client environment. JavaScript can't access server-side resources like databases, catalogs, pricing information etc.
* Static HTML : Regular HTML cannot contain dynamic information. JSP is easy and convenient. It is quite feasible to insert small amounts of dynamic data.

1. MVC
2. What is MVC?

* Stands for "Model-View-Controller." MVC is an application design model comprised of three interconnected parts. They include the model (data), the view (user interface), and the controller (processes that handle input).
* The MVC model or "pattern" is commonly used for developing modern user interfaces. It is provides the fundamental pieces for designing a programs for desktop or mobile, as well as web applications. It works well with object-oriented programming, since the different models, views, and controllers can be treated as objects and reused within an application.
* Below is a description of each aspect of MVC:
* Model:

A model is data used by a program. This may be a database, file, or a simple object, such as an icon or a character in a video game.

* View:

A view is the means of displaying objects within an application. Examples include displaying a window or buttons or text within a window. It includes anything that the user can see.

* Controller:

A controller updates both models and views. It accepts input and performs the corresponding update. For example, a controller can update a model by changing the attributes of a character in a video game. It may modify the view by displaying the updated character in the game.

1. The advantages of MVC

* A main advantage of MVC is separation of concern. Separation of concern means we divide the application Model, Control and View.
* We can easily maintain our application because of separation of concern.
* In the same time we can split many developers work at a time. It will not affects one developer work to another developer work.
* It supports TTD (test-driven development). We can create an application with unit test. We can write won test case.
* Latest version of MVC Support default responsive web site and mobile templates.
* We can create own view engine. It is syntax is very easy compare to traditional view engine.

1. Java Spring
2. What is Java Spring?

* The Java Spring framework is very popular among developers because it offers them a lot of assistance with creating web apps, and, generally, it makes them more productive. With this in mind, let’s consider the pros and cons of Spring framework as well as some mistakes. However, before we get into the details, let’s get a brief refresher on what the Spring framework is all about.
* Spring is a Java Framework that makes it much simpler to create enterprise-grade applications. One of the biggest reasons for its popularity is that it supports a lot of scenarios. For example, large companies might have an app that has been around for quite some time now, and they need to run it on a server whose upgraded cycle is beyond the control of developers. Other businesses might have a standalone product that does not require a server. For all of these scenarios and many others, the Spring framework has you covered.

1. The advantages of Java Spring

* Plain Old Java Object – Developers call this POJO. The reason this is so beneficial is that it means developers do not have to use a server or any other enterprise container. This makes the entire framework extremely lightweight, which is a significant advantage when developing web applications.
* Flexible Configurations – Developers have the option to choose either XML or Java-based annotations for configuration purposes. Having such an option makes the jobs of developers a lot simpler.
* The AOP Module – Developers can have different compilation units or a separate class loader.
* Testing is Easier – The Spring Dependency injection helps developers insert test data.

1. Detail
2. Date of project plan

* Start: 01/06/2020
* Finish: 23/07/2020

1. Project’s Vision/Objective

* Create a website for the Asterism Cinema, on the site will show the films of the currently playing theater, and soon.
* The movie will automatically delete from the show date without a schedule.
* Users can view movie information, movie showtimes and book tickets. To make a reservation, a login is required, when the ticket is displayed, all seats selected must be colored so that the user is aware and cannot choose. After choosing a chair, go to the payment information page, after payment the data will be updated into the database and the selected seat will switch to the purchased state for others to see.

1. Project Initiation/Requirement Document

The website has 2 different directions after logging in:

* For admin account:
* There are features like viewing movie listings, updating movie information, adding movie showtimes, adding movies.
* See the list of all ordered invoices.
* View user list, toggle user status.
* For customer account:
* View the list of currently playing movies, movie details, movie showtimes for a week, select tickets, book tickets and display the bill.

1. Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Phạm Quốc Nghị | Lê Thành Nhân | Hứa Thiên Ngân | Nguyễn Quốc Toàn |
| Design wireframe |  |  |  | ✓ |
| Design Logo |  |  | ✓ |  |
| Add data |  | ✓ |  |  |
| Create model |  |  | ✓ | ✓ |
| MVC config | ✓ |  |  |  |
| Design user interface |  |  |  | ✓ |
| Code HomeController | ✓ | ✓ |  |  |
| Code FilmController |  |  |  | ✓ |
| Code BillController |  |  | ✓ |  |
| Code UserController |  | ✓ |  |  |
| Code-view: Customer-header |  |  |  | ✓ |
| Code-view: Customer-footer |  |  | ✓ |  |
| Code-view: Home page |  |  |  | ✓ |
| Code-view: ShowTime page | ✓ |  |  |  |
| Code-view: Film Detail |  |  |  | ✓ |
| Code-view: Login page |  | ✓ |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Code-view: Choose seat | ✓ |  |  |  |
| Design wireframe of admin |  |  | ✓ | ✓ |
| Code Controller: RoomController | ✓ |  |  |  |
| Code-view: Film list |  |  | ✓ |  |
| Code-view: User list |  |  | ✓ |  |
| Code-view: User update |  | ✓ |  |  |
| Code-view: Dashboard bar |  |  | ✓ |  |
| Code-view: Bill list (for customer) |  |  |  | ✓ |
| Code-view: Bill page |  |  |  | ✓ |
| Code-view: Bill Form |  |  |  | ✓ |
| Code-view: Film list (for admin) | ✓ |  |  |  |
| Code-view: register page |  | ✓ |  |  |
| Debug code | ✓ | ✓ |  |  |

1. Roles
2. Table roles

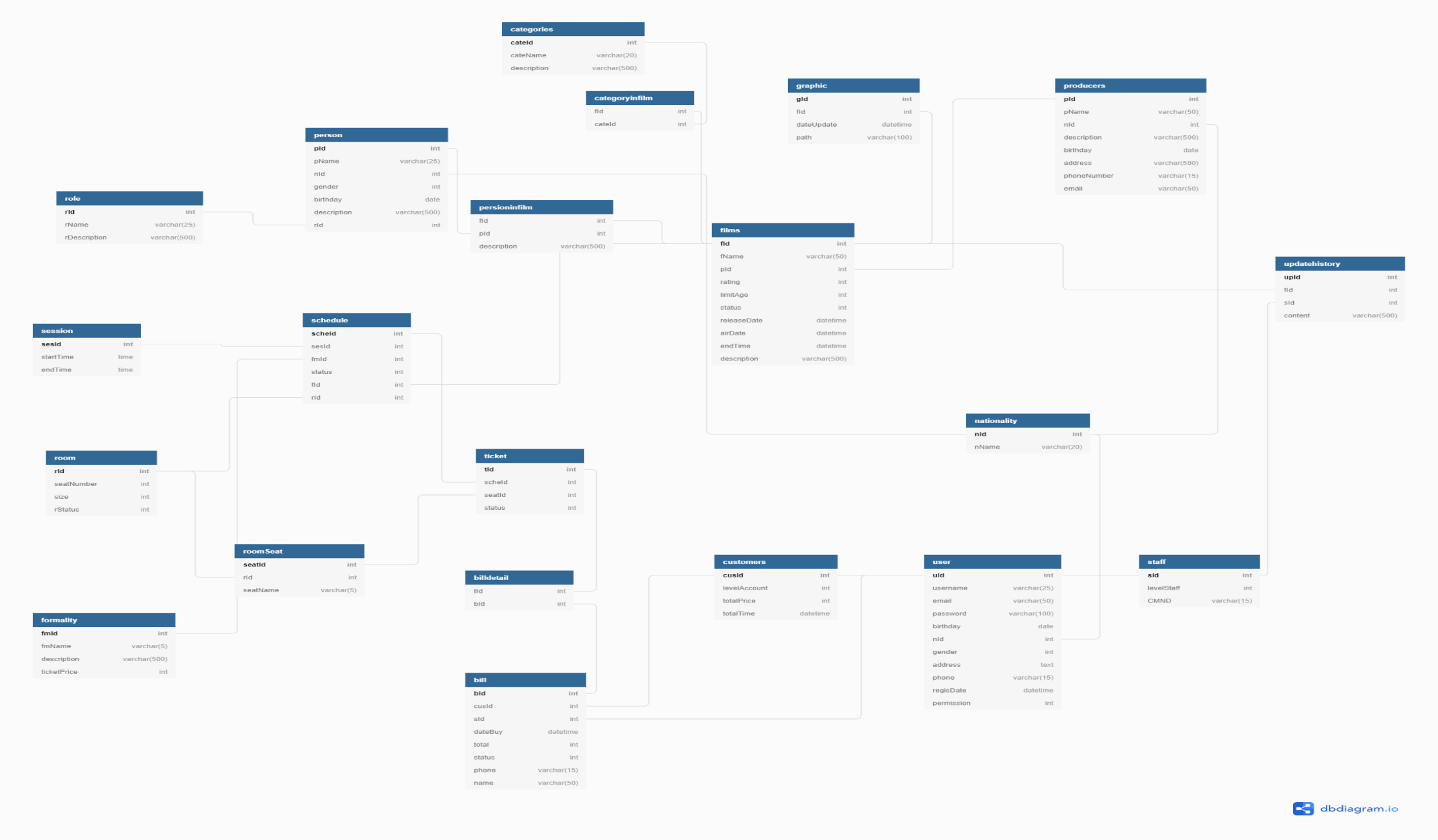
|  |  |
| --- | --- |
| Member | Roles |
| Phạm Quốc Nghị | Leader, System Manager |
| Lê Thành Nhân | Back-end Developer |
| Hứa Thiên Ngân | Full-Stack Developer |
| Nguyễn Quốc Toàn | Full-Stack Developer |

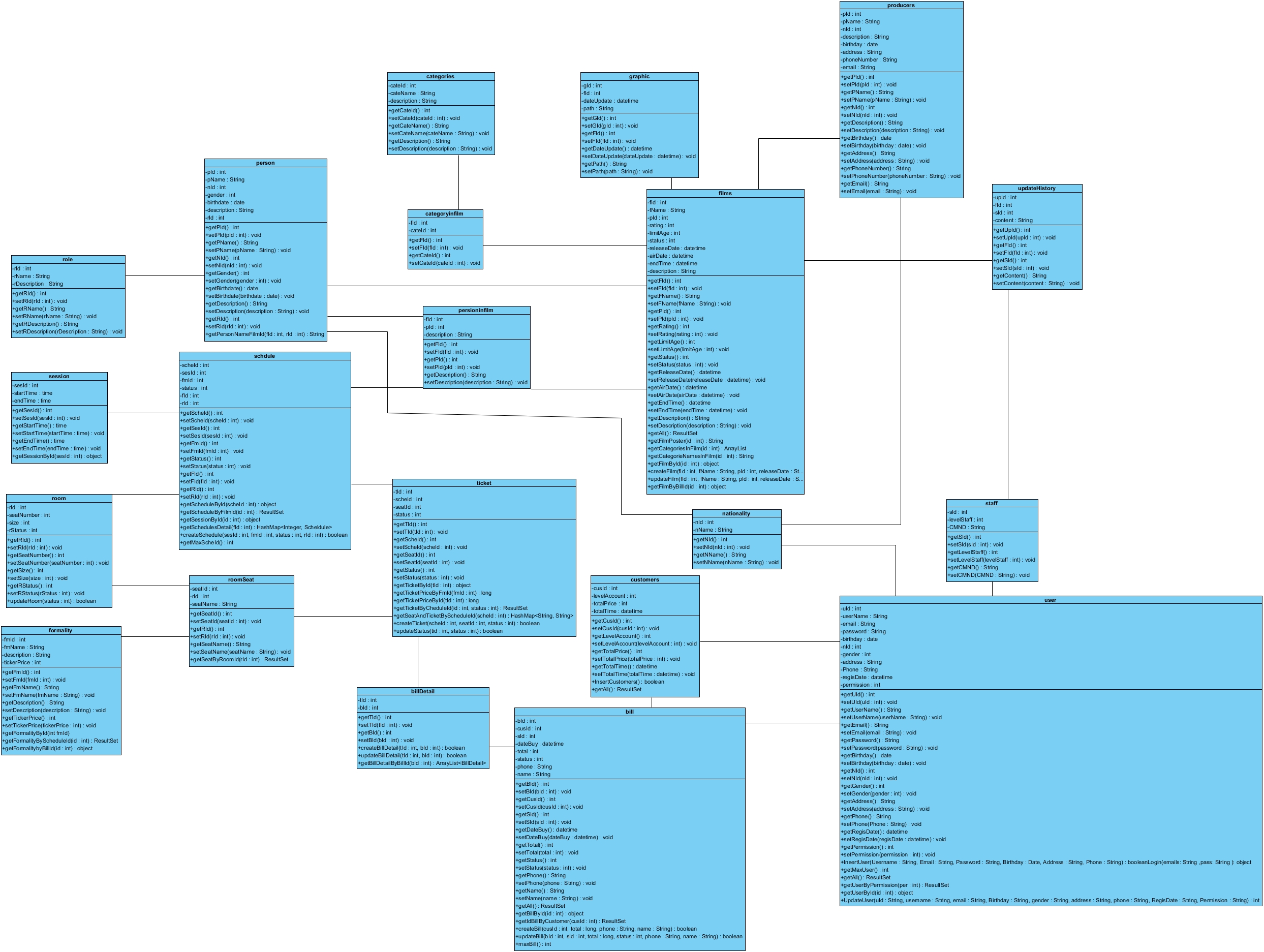
1. Gantt

1. Schedule

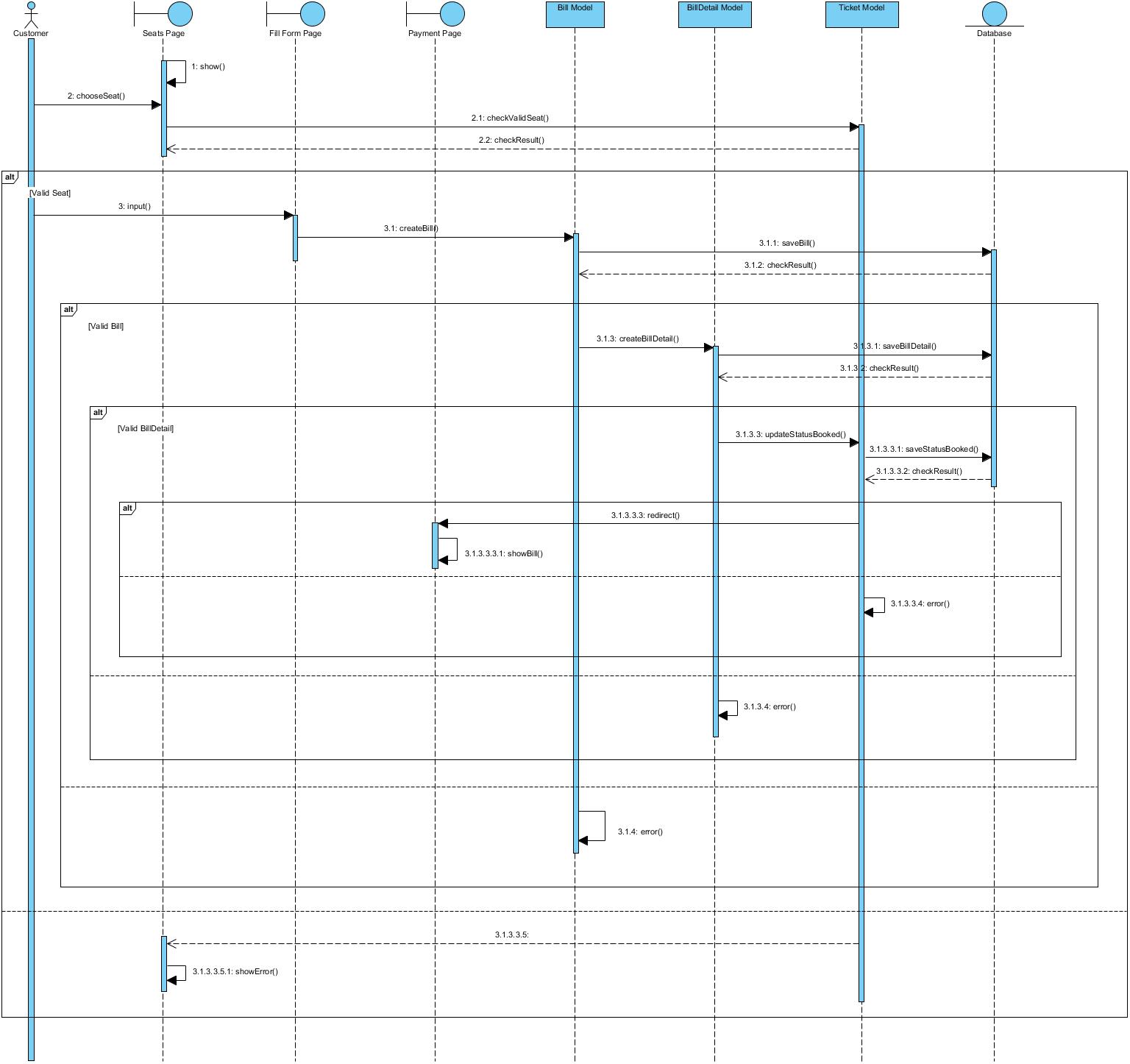
**Lịch họp offline**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***STT*** | ***Ngày họp*** | ***Thời gian họp*** | ***Mục đích cuộc họp*** | ***Các thành viên tham gia*** | ***Hình Thức họp*** |
| 1 | 23/06/2020 | 14:00 | Xác định dự án và phân chia nhiệm vụ | Full | Offline |
| 2 | 06/07/2020 | 14:00 | Báo cáo tiến độ và Kiểm tra sản phẩm | Full | Offline |
| 3 | 11/07/2020 | 14:00 | Báo cáo tiến độ và Kiểm tra sản phẩm | Full | Offline |
| 4 | 17/07/2020 | 14:00 | Báo cáo tiến độ và Kiểm tra sản phẩm | Full | Offline |
| 5 | 22/07/2020 | 14:00 | Kiểm tra sản phẩm | Full | Offline |

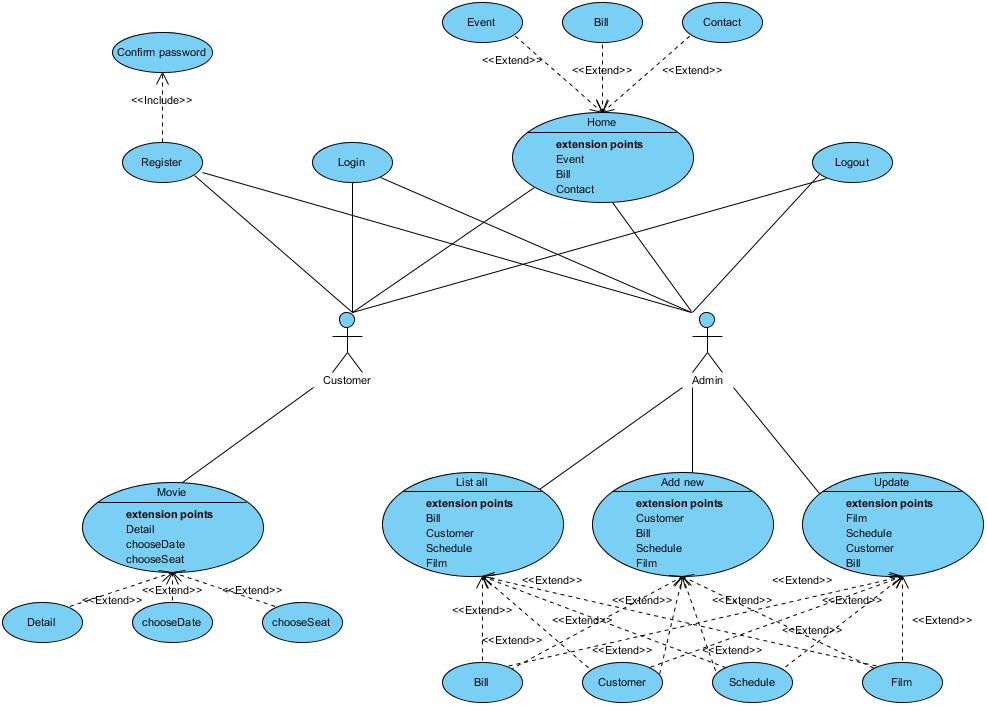
1. Design Pattern
2. Data base Diagram
3. Class Diagram



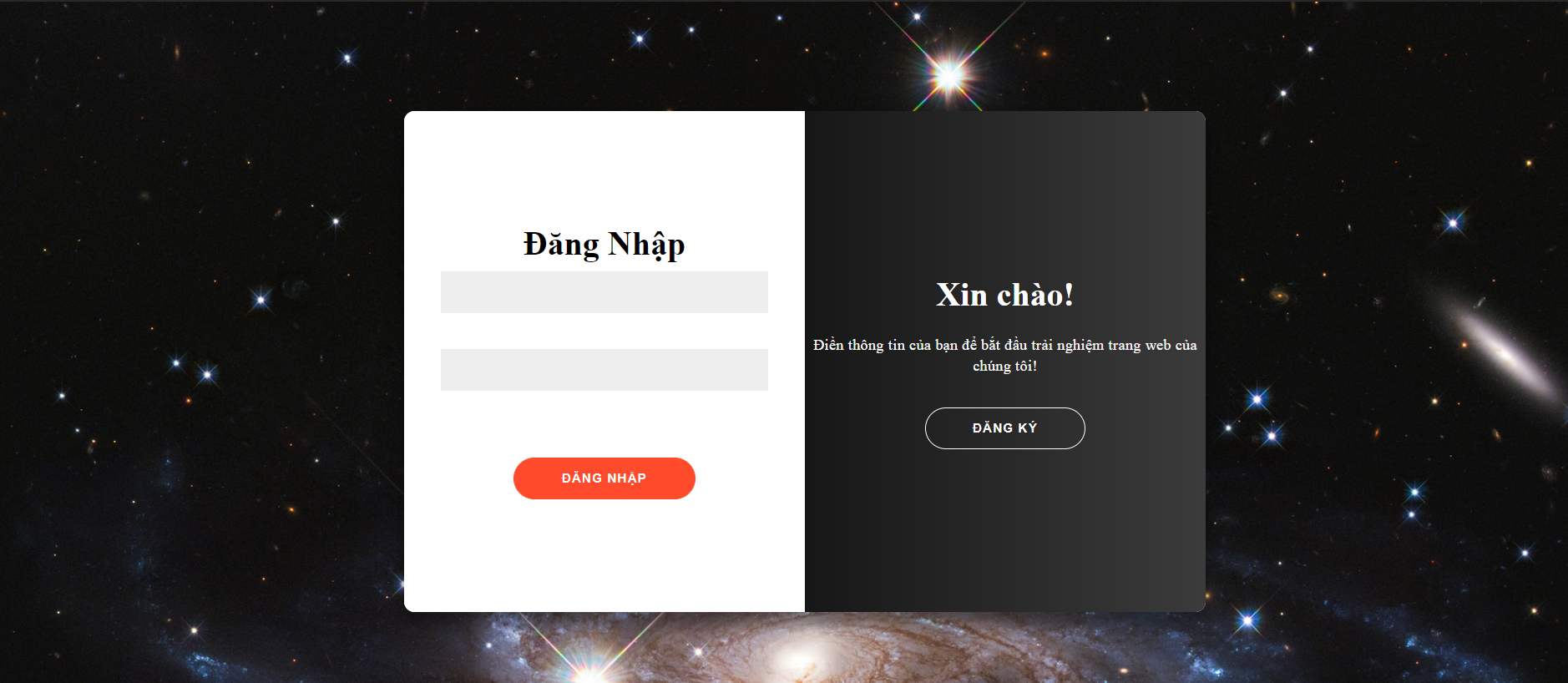
1. Sequence diagram



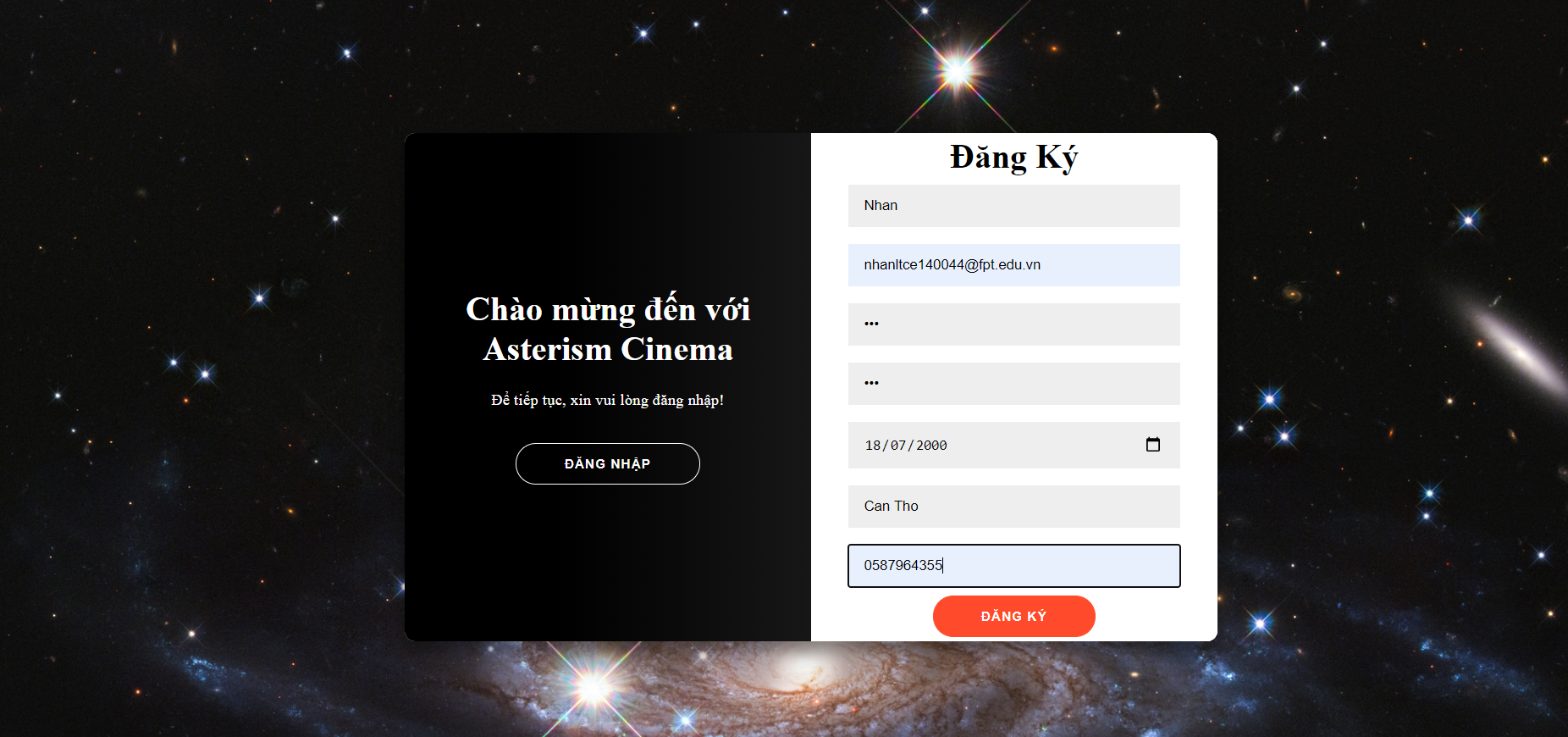
1. Usercase diagram



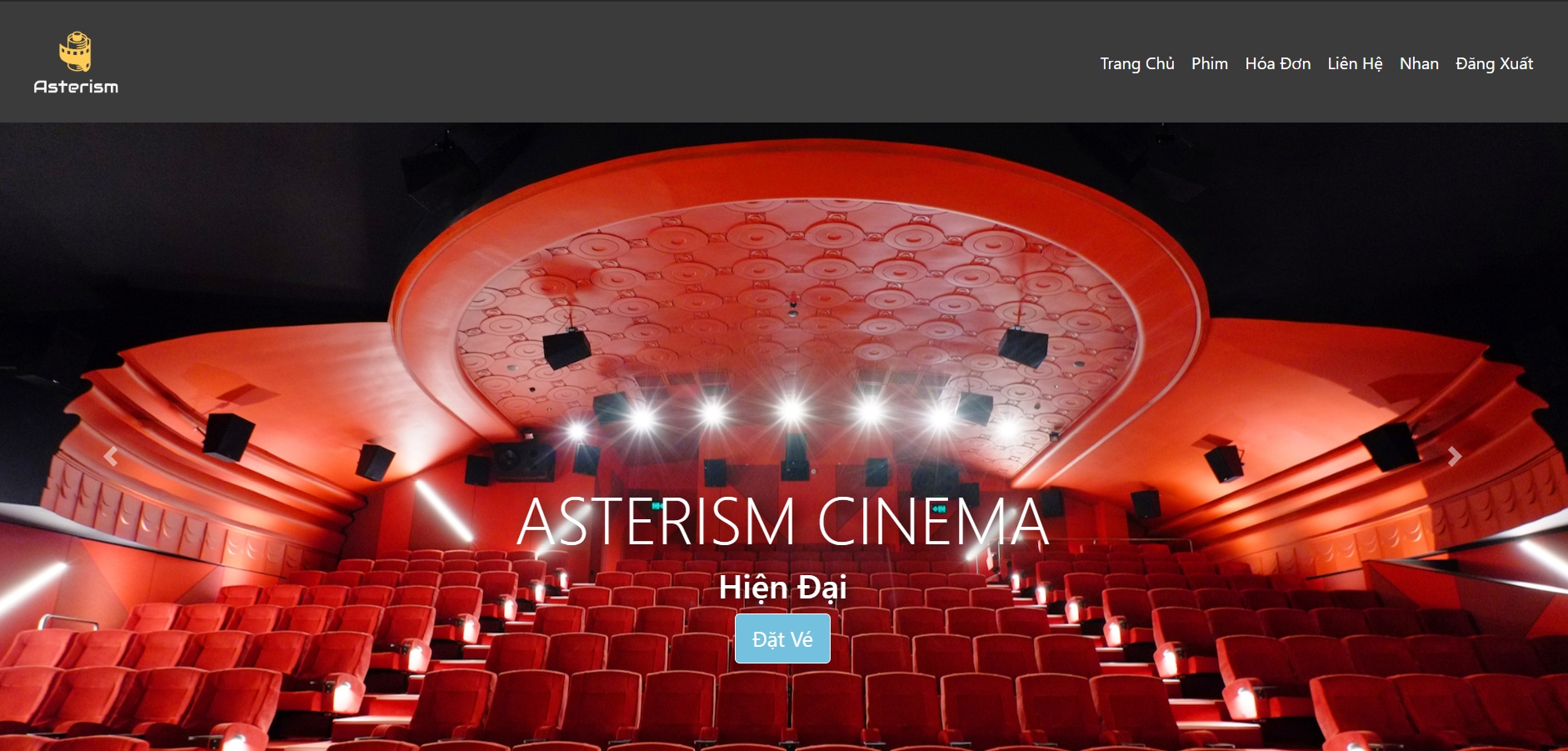
1. Program Interface
2. User View



Login Page



Register Page



Home Page