 **MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Cinema Ticket**

|  |  |
| --- | --- |
| **Group 2** | |
| **Group members** | Lê Công Danh– Team Leader – SE61769  Lưu Lạc – Team Member – SE62056  Hồ Sĩ Hoàng – Team Member – SE61397  Đặng Xuân Huy – Team Member – SE61318 |
| **Supervisor** | Ms. Nguyễn Thị Cẩm Hương |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | OCBS |

- Ho Chi Minh City, 08/2017 -*This page is intentionally left blank*

***ACKNOWLEDGEMENTS***

We would like to take this opportunity to express our profound gratitude towards all the people who have contributed to this project: Our teachers for their advices and participation in the final review, our friends for their valuable technical support.

We would also like to express our appreciation to Ms. Nguyễn Thị Cẩm Hương, our research supervisor for her professional guidance and the useful, constructive recommendations throughout the course of this project.

Finally, we would like to thank our Families and Friends, for their unconditional support and tolerance during the time of doing the project.

# Table of Contents

[Table of Contents 4](#_Toc517994879)

[List of Tables 6](#_Toc517994880)

[Definitions, Acronyms, and Abbreviations 9](#_Toc517994881)

[A. Introduction 10](#_Toc517994882)

[1. Project Overview 10](#_Toc517994883)

[2. Introduction 10](#_Toc517994884)

[3. Current Situation 10](#_Toc517994885)

[4. Problem Definition 10](#_Toc517994886)

[5. Proposed Solution 11](#_Toc517994887)

[**5.1** **Featured functions** 11](#_Toc517994888)

[**5.2** **Possible drawbacks** 11](#_Toc517994889)

[6. Functional Requirements 11](#_Toc517994890)

[7. Roles and Responsibilities 12](#_Toc517994891)

[B. Software Project Management Plan 13](#_Toc517994892)

[1. Problem Definition 13](#_Toc517994893)

[1.1 Name of this Capstone Project 13](#_Toc517994894)

[1.2 Problem Abstract 13](#_Toc517994895)

[1.3 Project Overview 13](#_Toc517994896)

[2. Project organization 16](#_Toc517994897)

[2.1 Software Project Model 16](#_Toc517994898)

[2.2 Roles and Responsibilities 16](#_Toc517994899)

[2.3 Tools and Techniques 17](#_Toc517994900)

[3. Project Management Plan 18](#_Toc517994901)

[3.1 Product Backlog 18](#_Toc517994902)

[3.2 Sprint Backlog 20](#_Toc517994903)

[3.3 Deliverables 25](#_Toc517994904)

[3.4 All Meeting Minutes 25](#_Toc517994905)

[4. Coding Convention 25](#_Toc517994906)

[C. Software Requirement Specifications 27](#_Toc517994907)

[1. User Requirement Specification 27](#_Toc517994908)

[1.1 Admin Requirement 27](#_Toc517994909)

[1.2 Partner Requirement 27](#_Toc517994910)

[1.3 Cinema Manager Requirement 27](#_Toc517994911)

[1.4 User Requirement 27](#_Toc517994912)

[1.5 Scheduler Requirement 28](#_Toc517994913)

[2. Software Requirement Specification 28](#_Toc517994914)

[2.1 External Interface Requirement 28](#_Toc517994915)

[2.2 System Overview Use Case 28](#_Toc517994916)

[2.3 List of Use Case 29](#_Toc517994917)

[3. Software Requirement Specification 104](#_Toc517994923)

[4. Conceptual Diagram 105](#_Toc517994924)

[D. Software Design Description 107](#_Toc517994925)

[1. Design Overview 107](#_Toc517994926)

[2. System Architectural Design 107](#_Toc517994927)

[2.1 Web Application Architecture Description 107](#_Toc517994928)

[2.2 Web Service Application Architecture Description 109](#_Toc517994929)

[2.3 Mobile Application Architecture Description 109](#_Toc517994930)

[3. Component Diagram 110](#_Toc517994931)

[4. Detailed Description 112](#_Toc517994932)

[4.1 Class Diagram 112](#_Toc517994938)

[4.2 Class Diagram Explanation 113](#_Toc517994939)

[5. Interface 119](#_Toc517994940)

[5.1 Web application Design 119](#_Toc517994941)

[5.2 Mobile application Design 125](#_Toc517994942)

[6. Database Design 134](#_Toc517994943)

[6.1 Entity Relationship Diagram (ERD) 134](#_Toc517994944)

[6.2 Data Dictionary 135](#_Toc517994945)

[7. Algorithms 135](#_Toc517994946)

[E. System Implementation & Test 136](#_Toc517994947)

[1. Introduction 136](#_Toc517994948)

[2. Database Relationship Diagram 136](#_Toc517994949)

[3. Implementation 136](#_Toc517994950)

[4. Test Plan 136](#_Toc517994951)

[5. System Testing Test Case 136](#_Toc517994952)

[F. Software User’s Manual 136](#_Toc517994953)

[1. Installation Guide 136](#_Toc517994954)

[2. User Guide 136](#_Toc517994955)

[G. Appendix 136](#_Toc517994956)

# List of Tables

[*Table 1: Roles and Responsibilities* 12](#_Toc517994700)

[*Table 2: Hardware requirement for Server* 15](#_Toc517994701)

[*Table 3: Hardware requirement for Web* 15](#_Toc517994702)

[*Table 4: Hardware requirement for Mobile* 15](#_Toc517994703)

[*Table 5: Software requirement* 15](#_Toc517994704)

[*Table 6: Roles and Responsibilities* 17](#_Toc517994705)

[*Table 7: Tools and Techniques* 18](#_Toc517994706)

[*Table 8: Product Backlog* 20](#_Toc517994707)

[*Table 9: Sprint 1 Development* 21](#_Toc517994708)

[*Table 10: Sprint 2 Development* 22](#_Toc517994709)

[*Table 11: Sprint 3 Development* 23](#_Toc517994710)

[*Table 12: Sprint 4 Development* 24](#_Toc517994711)

[*Table 13: Sprint 5 Development* 24](#_Toc517994712)

[*Table 14: Sprint 6 Development* 25](#_Toc517994713)

[*Table 15: Deliverables* 25](#_Toc517994714)

[Table 16: USE CASE - UC\_AD001 - <Admin> Login 32](#_Toc517994715)

[Table 17: USE CASE - UC\_AD002 - <Admin> View all group cinema 33](#_Toc517994716)

[Table 18; USE CASE -UC\_AD003 - <Admin> Add new store 35](#_Toc517994717)

[Table 19: USE CASE - UC\_AD004 - <Admin> Edit Group Cinema Information 37](#_Toc517994718)

[Table 20: USE CASE - UC\_AD005 - <Admin> Edit Category 39](#_Toc517994719)

[Table 21: USE CASE - UC\_AD006 - <Admin> Create account for Partner 40](#_Toc517994720)

[Table 22: USE CASE - UC\_AD007 - <Admin> Edit partner account 42](#_Toc517994721)

[Table 23: USE CASE - UC\_AD008 - <Admin> Disable partner account 43](#_Toc517994722)

[Table 24: USE CASE - UC\_AD009 - <Admin> Add Film 45](#_Toc517994723)

[Table 25: USE CASE - UC\_AD010 - <Admin> Add Film 47](#_Toc517994724)

[Table 26: USE CASE - UC\_AD011 - <Admin> Edit Film 50](#_Toc517994725)

[Table 27: USE CASE - UC\_P001 - <Partner> Partner Login 52](#_Toc517994726)

[Table 28: USE CASE - UC\_P002 - <Partner> View all cinema manager account 53](#_Toc517994727)

[Table 29: USE CASE - UC\_P003 <Partner> Add Cinema Manager Account 55](#_Toc517994728)

[Table 30: USE CASE - UC\_P004 <Partner>Delete Cinema Manager Account 56](#_Toc517994729)

[Table 31: USE CASE - UC\_P005 <Partner> Edit Cinema Manager Account 58](#_Toc517994730)

[Table 32: USE CASE - UC\_P006 <Partner> View all cinema 60](#_Toc517994731)

[Table 33: USE CASE - UC\_P007 <Partner> Arrange seat 61](#_Toc517994732)

[Table 34: USE CASE - UC\_P008 <Partner> Add Room for Cinema 63](#_Toc517994733)

[Table 35: USE CASE - UC\_P009 <Partner> Add New Cinema 65](#_Toc517994734)

[Table 36: USE CASE - UC\_P010 <Partner> Edit cinema 67](#_Toc517994735)

[Table 37: USE CASE UC\_P011 <Partner> View Room List 69](#_Toc517994736)

[Table 38: USE CASE - UC\_P012 <Partner> View Room Detail 70](#_Toc517994737)

[Table 39: USE CASE - UC\_U001 <User> User Login 72](#_Toc517994738)

[Table 40: USE CASE - UC\_U002 <User> View Profile 74](#_Toc517994739)

[Table 41: USE CASE - UC\_U003 <User> Edit profile 75](#_Toc517994740)

[Table 42: : USE CASE - UC\_U004 <User> View Booking History 77](#_Toc517994741)

[Table 43: USE CASE - UC\_U005 <User> Sign out 78](#_Toc517994742)

[Table 44: USE CASE - UC\_C001 <Customer> Register 80](#_Toc517994743)

[Table 45: USE CASE - UC\_C002 <Customer> Find Schedule 82](#_Toc517994744)

[Table 46: USE CASE - UC\_C003 <Customer> Choose type of seat and quality 84](#_Toc517994745)

[Table 47: USE CASE - UC\_C004 <Customer> Choose Seat 85](#_Toc517994746)

[Table 48: USE CASE - UC\_C005 <Customer> Checkout 87](#_Toc517994747)

[Table 49: USE CASE - UC\_C006 <Customer> Resell ticket 89](#_Toc517994748)

[Table 50: USE CASE - UC\_C007 <Customer> Return ticket 91](#_Toc517994749)

[Table 51: USE CASE - UC\_M001 <Cinema manager > Manager Login 94](#_Toc517994750)

[Table 52: USE CASE - UC\_M002 <Cinema manager > Add movie schedule 95](#_Toc517994751)

[Table 53: USE CASE - UC\_M003 <Cinema manager > Add promotion 97](#_Toc517994752)

[Table 54: USE CASE - UC\_M004 <Cinema manager > View report 99](#_Toc517994753)

[Table 55: USE CASE - UC\_S001 <System> Craw film automatic 101](#_Toc517994754)

[Table 56: USE CASE - UC\_S002 <System> Notification about schedule 103](#_Toc517994755)

[Table 57: USE CASE - UC\_S003 <System> suggest film for user 104](#_Toc517994756)

[Table 58: Conceptual Diagram Data Dictionary 107](#_Toc517994757)

[Table 59: Component Dictionary 111](#_Toc517994758)

[Table 60: Class Diagram Dictionary 113](#_Toc517994759)

[Table 61: User class attributes explanation 113](#_Toc517994760)

[Table 62: User class methods explanation 113](#_Toc517994761)

[Table 63: Admin class attributes explanation 114](#_Toc517994762)

[Table 64: Admin method methods explanation 114](#_Toc517994763)

[Table 65: Customer class attributes explanation 114](#_Toc517994764)

[Table 66: Customer method methods explanation 114](#_Toc517994765)

[Table 67: Partner class attributes explanation 114](#_Toc517994766)

[Table 68: Partner class methods explanation 114](#_Toc517994767)

[Table 69: Cinema Manager class attributes explanation 115](#_Toc517994768)

[Table 70: Cinema Manager class methods explanation 115](#_Toc517994769)

[Table 71: Film class attributes explanation 115](#_Toc517994770)

[Table 72: Film class methods explanation 115](#_Toc517994771)

[Table 73: Movie class attributes explanation 116](#_Toc517994772)

[Table 74: Movie class methods explanation 116](#_Toc517994773)

[Table 75: Show time class attributes 116](#_Toc517994774)

[Table 76: Show time class method explanation 116](#_Toc517994775)

[Table 77: Room class attributes explanation 116](#_Toc517994776)

[Table 78: Room class methods explanation 116](#_Toc517994777)

[Table 79: Digital type class attributes explanation 116](#_Toc517994778)

[Table 80: Digital Type class methods explanation 117](#_Toc517994779)

[Table 81: Seat class attributes explanation 117](#_Toc517994780)

[Table 82: Seat class methods explanation 117](#_Toc517994781)

[Table 83: Type of Seat class attributes explanation 117](#_Toc517994782)

[Table 84: Type of Seat class methods explanation 117](#_Toc517994783)

[Table 85: Group Cinema class attributes explanation 117](#_Toc517994784)

[Table 86: Group Cinema class methods explanation 118](#_Toc517994785)

[Table 87: Cinema class attributes explanation 118](#_Toc517994786)

[Table 88: Cinema class methods explanation 118](#_Toc517994787)

[Table 89: Promotion class attributes explanation 118](#_Toc517994788)

[Table 90: Promotion class methods explanation 118](#_Toc517994789)

[Table 91: Ticket class attributes explanation 118](#_Toc517994790)

[Table 92: Ticket class methods explanation 119](#_Toc517994791)

[Table 93: Booking ticket class attributes explanation 119](#_Toc517994792)

[Table 94: Booking ticket class methods explanation 119](#_Toc517994793)

[Table 95: Booking class attributes explanation 119](#_Toc517994794)

[Table 96: Booking class methods explanation 119](#_Toc517994795)

[Table 97: Define <Unauthorized User> Home Page Android 127](#_Toc517994796)

[Table 98: Define <Unauthorized User> FilmDetail Page Android 129](#_Toc517994797)

[Table 99: Define <Unauthorized User> ChooseTicket Page Android 130](#_Toc517994798)

[Table 100: Define <Unauthorized User> ChooseSeat Page Android 132](#_Toc517994799)

**List of Figures**

[Figure 1: Scrum Process 15](#_Toc517357524)

[Figure 2: System Overview Use Case 28](#_Toc517357525)

[Figure 3: <Admin> Overview Use Case 29](#_Toc517357526)

[Figure 4: <Admin> Login admin(UC\_AD001) 29](#_Toc517357527)

[Figure 5: <Admin> View All Group Cinema (UC\_AD002) 31](#_Toc517357528)

[Figure 6: <Admin> Add new group cinema (UC\_AD003) 32](#_Toc517357529)

[Figure 7: <Admin> Edit group cinema information(UC\_AD04) 34](#_Toc517357530)

[Figure 8: <Admin> View All Partner (UC\_AD005) 37](#_Toc517357531)

[Figure 9: <Admin> Create Account for Partner (UC\_AD006) 38](#_Toc517357532)

[Figure 10: <Admin> Edit partner account (UC\_AD007) 40](#_Toc517357533)

[Figure 11: <Admin> Delete partner account (UC\_AD008) 41](#_Toc517357534)

[Figure 12: <Admin> View list film (UC\_AD009) 43](#_Toc517357535)

[Figure 13: <Admin> Add Film(UC\_AD010) 44](#_Toc517357536)

[Figure 14: <Admin> Edit Film (UC\_AD011) 47](#_Toc517357537)

[Figure 15: <User> Overview Use Case 49](#_Toc517357538)

[Figure 16: <Partner> Partner login (UC\_P001) 49](#_Toc517357539)

[Figure 17: <Partner> View all cinema manager account (UC\_P002) 51](#_Toc517357540)

[Figure 18: <Partner> Add Cinema Manager Account (UC\_P003) 52](#_Toc517357541)

[Figure 19: <Partner>Delete Cinema Manager Account (UC\_P004) 54](#_Toc517357542)

# Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| SCP | Shopping Clothes with Pictures |
| UI | User Interface |
| UX | User Experience |
| MVVM | Model – View – View Model |
| ORM | Object Relation Mapping |
| TDD | Test Driven Development |

# Introduction

## Project Overview

* Official name: Online cinema booking system
* Vietnamese name: Hệ thống đặt vé xem phim trực tuyến
* Abbreviation: OCBS
* Product Type: Website and Mobile Applications
* Start Date: 21/05/2018
* End Date: 31/08/2018

## Introduction

The demand for seeing movie at cinema is growing fast, along with the rapid rising of smartphones and websites. Our team decided to develop a system that would allow users to book movie tickets easily from all our partners. The system also helps partners manage better their business by our revenue report’s system.  We allow the customer to control their business on our system. Besides that, we also to make our users more satisfied with us by trying to improve old function. We also add some new features for example: we allowing customer to resell their ticket to other customer in case they are busy and want to resell it.

## Current Situation

In ordinary life, when you want to watch movies in the cinema, you have to buy tickets but you do not have time. Remember the time when you had to stand in a long queue outside the theatre to book movie tickets.

How do you not go to the cinema but have the ticket and have the best seats?

Nowadays, the force is strong to jump online or use highly accessible mobile apps to book tickets to the latest trending film. However, while many cinemas in the market of Vietnam have not general system for booking tickets. Customers must buy ticket offline or find another many website of film company to buy it. Therefor, they still look for a solution to enable moviegoers to purchase tickets online.

Nobody can deny that today’s society is one of convenience. With the technology sector continually looking to humor our desire to save time and increase efficiency we’ve become used to having access to the world through our fingertips. But any website do not enough demand of customer. We’ve all heard about the highly disgruntled moviegoers about some problem for this buy the tickets online.

## Problem Definition

Current systems doesn't allow customer to resell ticket when they are busy.

The current system does not allowed to return ticket in a time interval (example: Before 4 hours of movie screening, users can return tickets to the system and get back the accumulated points. Accumulated points can be used to purchase products in the store).

The current system makes promotional announcements for general customer, but do not made promotion announcements for specific customer (for example, people in the month purchased more than 30 movie tickets).

## Proposed Solution

To the problems that existing systems have, we are developing a system that help customers who are having trouble finding a compatible service in booking cinema tickets. Customers who use our system will have more choices in how they can handle their purchased tickets.

Cinemas which use our service will have less empty seats due to how our system works. The cinemas are able to not only give promotion to all customers but they can also give promotion to specific customers.

* 1. Featured functions
* Web application:
  + Provides the ability to resell or return the tickets for customers who cannot come.
  + Cinemas can arrange seats for the room in case of seats being too disconnected.
  + Cinemas have the ability to give promotion to specific customer.
* Mobile application:
  + QR code will be provided for each ticket purchased and checked at the movie room gate
  1. Possible drawbacks
* Since there are QR code verifications, staffs will have to be trained.
* Customers who want to buy the tickets that are being sold by other customers have to contact the seller themselves to make the transaction.

## Functional Requirements

The functional requirements are listed as below:

* Customer component:
  + Booking ticket: customers can choose the ticket base on film, cinema, time and check out the ticket.
  + Resell ticket: customers can resell ticket for the other when they are busy.
  + Return ticket: customers can return ticket and receive money online.
  + Register: customers can register new account to become a user
* User component:
  + User can do whatever customer can.
  + Login: User can login to the system.
  + Manage profile: User can view and edit their profile, view booking history.
* Partner component:
  + Partner Login: partner can login to manage their cinema group
  + Add Cinema: partner can add new cinema
  + Add Room: partner can add room for cinema that they have created.
  + Arrange Seat: partner can arrange seat for the room in cinema
  + View statistic and report: partner can see the user statistic and revenue report.
  + Add promotion: partner can add promotion and for specific user or all customer.
* Admin component:
  + Admin login: Admin can login to manage partner and the system.
  + Add partner: Admin can create new group of cinema of new partner
  + Add Film: Admin can add new film for partner to create schedule
  + Manage Partner Account: Admin can add, disable, change password of partner’s account.
* System component:
  + Save report: automatic save partner report each month and each 6 months.
  + Notify: the system will notify to user when cinema add new promotion that related to them.

## Roles and Responsibilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full Name** | **Role** | **Position** | **Contact** |
| 1 | Nguyễn Thị Cẩm Hương | Project Manager | Supervisor | huongntc2@fpt.edu.vn |
| 2 | Lê Công Danh | Developer | Leader | danhlcse61769@fpt.edu.vn |
| 3 | Lưu Lạc | Developer | Member | laclse62056@fpt.edu.vn |
| 4 | Hồ Sĩ Hoàng | Developer | Member | hoanghsse61397@fpt.edu.vn |
| 5 | Đặng Xuân Huy | Developer | Member | huydxse61318@fpt.edu.vn |
| *Table 1: Roles and Responsibilities* | | | | |

# B. Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

* Official name: Online cinema booking system
* Vietnamese name: Hệ thống đặt vé xem phim trực tuyến
* Abbreviation: OCBS

### Problem Abstract

* The demand for watching movies at the cinema is rising. Customer want a fast and convenient way to buy tickets.
* We offer a convenient online booking movie ticket application for both web and mobile.  We always want the best service for our users. So, we have added some new features to help them buying tickets easier. For example, we allow them to pay or resell tickets in case they are busy.
* We also care a lot about our partner. We provide more support to our partners by offering statistical reports that help them better manage their cinemas.

### Project Overview

#### Current diagnostic diseases

Below are the problems encountered in this project:

* **Lack of UI, UX knowledge:** Our team members all study Information System and no one have good understanding about UI and UX of a standard E-Commerce application. That is the reason why spending time to study popular websites UI, UX is quite important.
* **Many similar systems on the market:** There are many systems like us in the market. The challenge is we have to do better than what they already have. Besides that, we need to analyses their system’s weaknesses and add new feature to overcome it.
* **We do not know a lot about our future partner:** We need to find out what our partner need and provide it. we analyses information collect from the system to support for our partner. The biggest purpose for partner is to increase their revenue.

#### The Proposed System

Our system includes three main subsystems: online websites for customer and user to booking ticket, for partner to manage their movie theater, for admin to manage the system. For convenient, we also have mobile application system for user and customer to booking ticket.

##### Web Application

Website is a common system for users (customers), partner, cinema manager and admin. Website provide following features:

* For Customer:
  + Customer can booking ticket online: they can choose which film, cinema, time, seat that they like and then checkout the get the ticket.
  + Customer can resell ticket when they are busy or got in to urgent situation.
* For User:
  + User can login to the system.
  + User can manage their profile: user can view and edit profile.
  + User can see their booking history.
* For Cinema Manager:
  + Cinema Manager can login to the system.
  + Cinema Manager view their cinema report.
  + Cinema Manager can add schedule for their cinema.
  + Cinema Manager can add promotion for their cinema.
* For Partner:
  + Partner can do what cinema Manager can do.
  + Partner can manage the cinema in their system.
  + Partner can add new cinema: they can add new cinema and then add room and arrange seat the this room.
* For Admin:
  + Admin can login to admin ’s system.
  + Admin can add and manage film list.
  + Admin can Manage partner account: They can add new partner and add new partner type of seat. They also can disable change password of partner’s account.

##### Mobile Application

The mobile application mostly for users and customer to:

* + Search films and booking ticket.
* Manage profile only for user

##### API Application

#### Boundaries of the System

* Everybody who interact with the system has to equip enough device as follow:
  + Computer with internet connection.
  + An android smartphone with android 4.0 or above.
* The language of this system is Vietnamese
* The complete product includes:
  + Website application for customer, users, partner, cinema manager and admin
  + Mobile application for customer and users.
* Every Cinema partner can corporate with us to manage their cinema better and improve their marketing.

#### Future plans

* System use machine learning to find out which film the user would like to see and then recommend it to them.
* We will integrated with GPS finding cinema near user on mobile application.
* We will integration with more platform of payment.
* We will integrate a forum about movie on this website.

#### Development Environment

##### Hardware requirement

* **For server:**

|  |  |  |
| --- | --- | --- |
| **Hardware** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | Cable (4 Mbps) | Cable (8 Mbps) |
| **Operating System** | Windows Vista, 7, 8 | Windows 7, 8 |
| **Computer Processor** | Intel® Core i3 1.4GHz | Intel® Core(TM) i5 CPU , M 460 @ 2.53GHz |
| *Table 2: Hardware requirement for Server* | | |

* **For web:**

|  |  |  |
| --- | --- | --- |
| **Hardware** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | **512Kbps** | **8 Mbps** |
| **Operating System** | Windows Vista, 7, 8 | Windows 7, 8 |
| **Computer Processor** | 1 GHz | Intel® Core(TM) i5 CPU , M 460 @ 2.53GHz |
| **Computer Memory** | 1GB of RAM | 3GB of RAM or more |
| *Table 3: Hardware requirement for Web* | | |

* **For mobile:**

|  |  |  |
| --- | --- | --- |
| **Hardware** | **Minimum Requirements** | **Recommended** |
| **Internet Connection** | 512Kbps | Wi-Fi Connection 12MB |
| **Operating System** | Android 4.0 | Android 4.0 |
| **Hardware** | NFC supported | NFC supported |
| **Memory** | 128MB of RAM | 1GB of RAM or more |
| *Table 4: Hardware requirement for Mobile* | | |

##### Software requirement

|  |  |
| --- | --- |
| **Software** | **Name / Version** |
| Operating system | Windows 7 or above |
| Environment | NET Framework |
| Modeling tool | Microsoft Visual 2013 |
| IDE | Visual studio 2013, Android studio 3.1.2 |
| DBMS | Microsoft Sql Server 2012 |
| Source control | Github Desktop |
| Web browser | Chrome 42 or above |

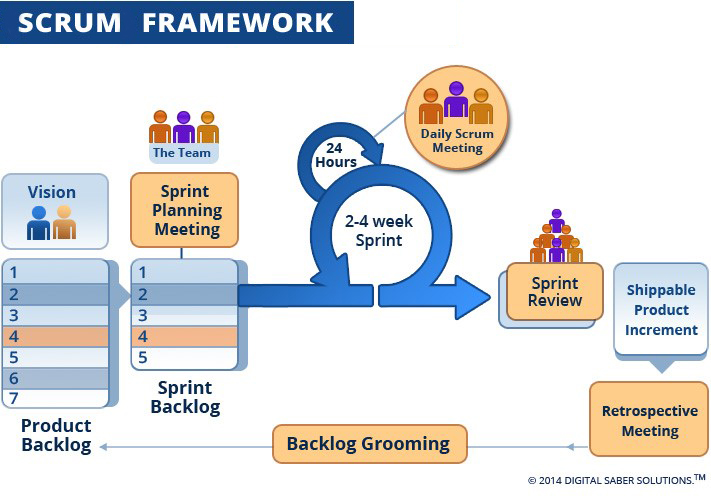
*Table 5: Software requirement*

## Project organization

### Software Project Model

This project is developed using Scrum model – part of an agile framework for Software development project. Our team choose Scrum model because of the following reasons:

* We had constantly changing requirements, as soon as we got something on paper, the stakeholder would make a change.
* Prototypes are delivered frequently for evaluation, usually weekly, rather than months.
* Take fewer risks when there is a change in requirement.
* All members must work together in order to avoid misunderstanding or miscommunication.
* Able to study new skills or knowledge at the same time as developing.



*Figure 1: Scrum Process*

(<http://www.digitalsaber.com/process/agile-scrum-framework/>)

### Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full Name** | **Role in Group** | **Responsibilities** |
| 1 | Nguyễn Thị Cẩm Hương | Product owner | * Specify scope and user requirement. * Supervise the development progress. * Provide professional techniques and business analysis support. |
| 2 | Lê Công Danh | Scrum master | * Managing process * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Testing |
| 3 | Lưu Lạc | Team member | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Testing |
| 4 | Hồ Sĩ Hoàng | Team member | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Testing |
| 5 | Đặng Xuân Huy | Team member | * Designing database * Clarifying requirements * Prepare documents * GUI design * Create test plan * Coding * Testing |
| *Table 6: Roles and Responsibilities* | | | |

### Tools and Techniques

|  |  |  |
| --- | --- | --- |
|  | **Tools** | **Techniques** |
| Front-end | Bracket | * HTML5 * CSS3 * ES6 * NodeJS * ReactJS |
| Back-end | Visual studio 2013 | * AJAX * JSON |
| Web server |  | * N/A |
| Mobile Application | Android studio | * Java 8 * Android SDK |
| Database management system | SQL SERVER 2012 | N/A |
| *Table 7: Tools and Techniques* | | |

## Project Management Plan

### Product Backlog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Story ID** | | **Features** | **Task ID** | **Task description** | **Sprint** |
| 1 | | Create Product Backlog | 1 | Create Product Backlog | 1 |
| 2 | | Write Introduction document | 2 | Write Introduction document | 1 |
| 3 | | Write project management plan | 3.1 | Problem definition | 1 |
| 3.2 | Project organization | 1 |
| 3.3 | Project management plan | 1 |
| 3.4 | Coding convention | 1 |
| 3.5 | Review document | 1 |
| 4 | | Building system structure | 4.1 | Building back-end structure | 1 |
|  |  |
| 5 | | Write software requirements | 5.1 | User requirement specification | 2 |  |  |
| 5.2 | Use case diagram | 2 |  |  |
| 6 | | Write software design and documentation | 6.1 | System architectural design | 2 |  |  |
| 6.2 | Component diagram | 2 |  |  |
| 6.3 | Class diagram | 2 |  |  |
| 6.4 | Sequence diagram | 2 |  |  |
| 6.5 | Database design | 2 |  |  |
| 6.6 | Entity relationship diagram | 2 |  |  |
| 6.7 | Design user interface | 2 |  |  |
| 6.8 | Test plan and test case | 2 |  |  |
| 7 | | Implementation | 7.1 | User registration | 3 |  |  |
| 7.2 | User login | 3 |  |  |
| 7.3 | User logout | 3 |  |  |
| 7.4 | User change them information | 3 |  |  |
| 7.5 | Booking ticket | 3 |  |  |
| 7.6 | Check out | 3 |  |  |
| 7.7 | Resell ticket | 3 |  |  |
| 7.8 | Return ticket | 3 |  |  |
| 7.9 | User cancel ticket | 3 |  |  |
| 7.10 | Displays a list of cinemas | 3 |  |  |
| 7.11 | Display a news page | 3 |  |  |
| 7.12 | Search movie by title, theater, date, showtime | 3 |  |  |
| 7.13 | Display movie (now show, coming soon) | 3 |  |  |
| 7.14 | Displays a movie's schedule for a specific movie | 3 |  |  |
| 7.15 | Displays theater’s information for a specific theater | 4 |  |  |
| 7.16 | Displays a list of movie schedule in a specific theater | 4 |  |  |
| 7.17 | Partner login | 4 |  |  |
| 7.18 | Partner create cinema | 4 |  |  |
| 7.19 | Partner add room to cinema | 4 |  |  |
| 7.20 | Partner arrange seat in room | 4 |  |  |
| 7.21 | Partner add film schedule | 4 |  |  |
| 7.22 | Partner add promotion | 4 |  |  |
| 7.23 | Partner view report | 4 |  |  |
| 7.24 | Partner view user statistic | 4 |  |  |
| 7.25 | Admin login | 4 |  |  |
| 7.26 | Amin add partner | 4 |  |  |
| 7.27 | Admin add partner type of seat | 4 |  |  |
| 7.28 | Admin manage the list of cinemas | 4 |  |  |
| 7.29 | Admin manage the list of movies | 4 |  |  |
| 7.30 | Admin manage the list of news and promotion | 4 |  |  |
| 8 | | Software testing | 8.1 | Unit test | 5 |  |  |
| 8.2 | Integration test | 5 |  |  |
| 8.3 | System test | 5 |  |  |
| 9 | | Quality assurance | 9.1 | Quality assurance for back-end | 5 |  |  |
| 9.2 | Quality assurance for web | 5 |  |  |
| 9.3 | Quality assurance for mobile | 5 |  |  |
| 10 | | Software user’s manual | 10.1 | Installation guide | 6 |  |  |
| 10.2 | User’s guide | 6 |  |  |
|  | | | | | |  |  |

***Table 8: Product Backlog***

### Sprint Backlog

#### Sprint 1 (14/5/2018 -27/5/2018) Project Initiation

##### Goal

* Sprint 1 must complete following tasks:
* Create Product Backlog
* Create Introduction document:
* Write Introduction document
* Review Introduction document
* Create project management plan:
* Problem definition
* Project organization
* Project management plan
* Coding convention
  + - Review document
* Build system structure:
* Backend Structure
* Web Structure
* Mobile Structure

##### Development

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | | **Task** | **Implement** | **Reviewer** | |
| 1 | | Create Product Backlog | LacL | DanhLC | |
| 2 | | Write Introduction document | HuyDX | DanhLC | |
| 3 | | Problem definition | DanhLC, HuyDx | LacL | |
| 4 | | Project organization | HoangHS | HuyDX,DanhLC | |
| 5 | | Project management plan | DanhLC | LacL | |
| 6 | | Coding convention | HuyDX, HoangHS | LacL | |
| 7 | | Review document | DanhLC | HuongNTC | |
| 8 | | Backend Structure | DanhLC, HoangHS | LacL | |
| 9 | | Web Structure | DanhLC | HuongNTC | |
| 10 | | Mobile Structure | LacL | HuongNTC | |
| 11 | | Mobile prototype | LacL | HuongNTC | |
| *Table 9: Sprint 1 Development* | | | |

#### Sprint 2 (28/5/2018 - 3/6/2018) Software requirement and basic features

##### Goal

* Sprint 2 must complete the following tasks:
* Write software requirements:
* User Requirement Specification
* Use case diagram
* Write software design and documentation:
* System architectural design
* Component diagram
* Class diagram
* Sequence diagram
* Database design
* Entity relationship diagram
* Design user interface
* Test plan and test case

##### Development

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | User Requirement Specification | DanhLc, LacL, HuyDX | HuongNTC | |
| 3 | Use case diagram | DanhLC | HuongNTC | |
| 5 | System architectural design | DanhLC, LacL | HuongNTC | |
| 6 | Component diagram | DanhLC | HuongNTC | |
| 7 | Class diagram | LacL, HoangHS, HuyDX | HuongNTC | |
| 8 | Sequence diagram | LacL, HoangHS, HuyDX | HuongNTC | |
| 9 | Database design | DanhLC | HuongNTC | |
| 10 | Entity relationship diagram | LacL | HuongNTC | |
| 11 | Design user interface | HuyDX(web), LacL(mobile) | HuongNTC | |
| 12 | Test plan and test case | LacL | HuongNTC | |
| 1 | User Requirement Specification | DanhLc, LacL, HuyDX | HuongNTC | |
| 3 | Use case diagram | DanhLC | HuongNTC | |
| 5 | System architectural design | DanhLC, LacL | HuongNTC | |
| *Table 10: Sprint 2 Development* | | | |

#### Sprint 3 (4/6/2018 - 17/6/2018) Software Design Description and main features

##### Goal

* Sprint 3 must complete the following tasks:
* Implementation:
* User registration
* User login
* User logout
* User change them information
* Booking ticket
* Check out
* Resell ticket
* Return ticket
* User cancel ticket
* Displays a list of cinemas
* Display a news page
* Search movie by title, theater, date, showtime
* Display movie (now show, coming soon)
  + - * Displays a movie's schedule for a specific movie

##### Development

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Task** | **Implement** | **Reviewer** |
| 1 | User registration | HoangHS | LacL |
| 2 | User login | HoangHS | LacL |
| 3 | User logout | HoangHS | DanhLC |
| 4 | User change them information | HuyDX | LacL |
| 5 | Booking ticket | DanhLC | LacL |
| 6 | Check out | DanhLC | LacL |
| 7 | Resell ticket | DanhLC | LacL |
| 8 | Return ticket | DanhLC | LacL |
| 9 | User cancel ticket | HuyDX | LacL |
| 10 | Displays a list of cinemas | HuyDX | DanhLC |
| 11 | Display a news page | LacL | DanhLC |
| 12 | Search movie by title, theater, date, showtime | LacL | DanhLC |
| 13 | Display movie (now show, coming soon) | LacL | DanhLC |
| 14 | Displays a movie's schedule for a specific movie | LacL | DanhLC |
| *Table 11: Sprint 3 Development* | | | |

#### 3.2 Sprint 4 (18/6/2018 – 8/7/2018) Finish up coding

##### Goal

* Sprint 4 must complete following tasks:
* Implementation:
* Displays theater’s information for a specific theater
* Displays a list of movie schedule in a specific theater
* Partner login
* Partner create cinema
* Partner add room to cinema
* Partner arrange seat in room
* Partner add film schedule
* Partner add promotion
* Partner view report
* Partner view user statistic
* Admin login
* Amin add partner
* Admin add partner type of seat
* Admin manage the list of cinemas
* Admin manage the list of movies
* Admin manage the list of news and promotion

##### Development

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Task** | **Implement** | **Reviewer** | |
| 1 | Displays theater’s information for a specific theater | HuyDX | LacL | |
| 2 | Displays a list of movie schedule in a specific theater | HuyDX | LacL | |
| 3 | Partner login | HoangHS | LacL | |
| 4 | Partner create cinema | HoangHS | LacL | |
| 5 | Partner add room to cinema | DanhLC | LacL | |
| 6 | Partner arrange seat in room | DanhLC | LacL | |
| 7 | Partner add film schedule | LacL | DanhLC | |
| 8 | Partner add promotion | DanhLC | LacL | |
| 9 | Partner view report | HuyDX | LacL | |
| 10 | Partner view user statistic | HoangHS | LacL | |
| 11 | Admin login | HoangHS | DanhLC | |
| 12 | Amin add partner | LacL | DanhLC | |
| 13 | Admin add partner type of seat | LacL | DanhLC | |
| 14 | Admin manage the list of cinemas | LacL | DanhLC | |
| 15 | Admin manage the list of movies | LacL | DanhLC | |
| 16 | Admin manage the list of news and promotion | LacL | DanhLC | |
| *Table 12: Sprint 4 Development* | | | |

#### Sprint 5 (8/7/2018 – 22/7/2018) Software Test Document and Quality Assurance

##### Goal

* Sprint 5 must complete the following tasks:
* Software testing:
* Unit test
* Integration test
* System test
* Quality assurance:
* Quality Assurance for Backend

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Task** | **Implement** | **Reviewer** | |
| 1 | Unit test | All | Self review | |
| 2 | Integration test | LacL | DanhLc | |
| 3 | System test | DanhLC | LacL | |
| 4 | Quality Assurance for Backend | HoangHS | DanhLC | |
| 5 | Quality Assurance for Web | HuyDX | DanhLC | |
| 6 | Quality Assurance for Mobile | LacL | DanhLC | |
| *Table 13: Sprint 5 Development* | | | |

* Quality Assurance for Web
* Quality Assurance for Mobile

##### Development

#### Sprint 6 (22/7/2018 – 5/8/2018) Software User’s Manual

##### Goal

* Sprint 6 must complete the following tasks:
* Installation Guide
* User’s Guide

##### Development

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Task** | **Implement** | **Reviewer** | |
| 1 | Installation Guide | LacL, HoangHS, HuyDX | DanhLC | |
| 2 | User’s Guide | DanhLC | LacL | |
| *Table 14: Sprint 6 Development* | | | |  |

### Deliverables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Deliverable** | **Deliverable date** | **Deliverable location** | **Note** |
| 1 | Introduction Document, Task list |  | FU - LMS | Report No. 1 |
| 2 | Software Project Management Plan |  | FU – LMS | Report No. 2 |
| 3 | Software Requirements Specification |  | FU – LMS | Report No. 3 |
| 4 | Software Design Description |  | FU – LMS | Report No. 4 |
| 5 | Software Test Documentation Guide Implementation (Coding) |  | FU – LMS | Report No. 5 |
| 6 | Software User’s Manual |  | FU - LMS | Report No. 6 |
| *Table 15: Deliverables* | | | | |

* For each Sprint, deliverables are potentially shippable products, which can be a part of document or prototype implemented based on the project’s core flow without any constraints.
* Each Sprint has a fixed duration of two weeks.

### All Meeting Minutes

All meeting documents could be found [here](https://drive.google.com/drive/folders/0B9QIp5ehkI6rXzVtdE9OMVNtTjg)

## Coding Convention

C#.NET : Using for backend website:

* Naming convention:
* In short examples that do not include [using directives](https://github.com/DanhlcGitHub/CinemaTicket), use namespace qualifications. If you know that a namespace is imported by default in a project, you do not have to fully qualify the names from that namespace. Qualified names can be broken after a dot (.) if they are too long for a single line.
* You do not have to change the names of objects that were created by using the Visual Studio designer tools to make them fit other guidelines.
* Indentation:
* These formatting rules concern the use of indentation to format code.
* The following table shows the rule names, applicable languages, default values, and first supported version of Visual Studio.

JavaScript (ES6): Using for frontend website

* Naming convention:
* Use camel-case style for variables and functions name (Lowercase for first letter)
* Use camel-case for classes (Uppercase for first letter)
* Indentation:
* Use two spaces per indentation level
* Do not use “;” for separate statements and expressions (ES6)
* Avoid single-line method
* Classes:
* Classes should be declared using class keyword for ES6. Using function to declares classes in ES6 is bad
* Use constructor function for initialize a class
* Use export and import to reuse modules

-    Use **this** to refer current class instance

Java (Android): Using for mobile development

* Naming convention:
* Use **camel-case** style for variables and functions name (Lowercase for first letter)
* Use **camel-case** for classes (Uppercase for first letter)
* Indentation:
* Use four **spaces** per indentation level
* User eight spaces to continue indent
* Use “;” for separate statements and expressions
* Classes:
* Classes should be declared using class keyword
* Using public, private, protected to declare protection level of instance variable and function
* -    Use **this** to refer current class instance

References:

* [1] C#.NET: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions
* [2] ES6: <https://github.com/elierotenberg/coding-styles/blob/master/es6.md>
* [3] Java: <http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

# Software Requirement Specifications

## User Requirement Specification

### Admin Requirement

Admin is the person who manages all the films and partners in the system. Admin can do the following functions:

* Manage films:
* Add new films
* Edit films
* View list films
* Manage Partner:
* Add new partner
* Edit partner information
* Disable partner account

### Partner Requirement

Partner is the person who manages a group of cinemas that belong to a brand. Partner can do the following functions:

* Add new cinema
* Manage cinema account:
* Add cinema manager account
* Edit cinema manager account

### Cinema Manager Requirement

Cinema manager is the person who manages 1 cinema of the cinemas that the partner own. Cinema manager can do the following functions:

* Add movie schedule
* Add promotion
* View Report
* View user Statistic

### User Requirement

Users are the people who use the service that have already registered. User can do the following functions:

* Manage user profile:
* View profile
* Edit profile
* View booking history

### Scheduler Requirement

Customers are the people who use the service that may or may not have registered. Customer can do the following functions:

* Register
* Book ticket

## Software Requirement Specification

### External Interface Requirement

#### User Interface

* The user interface uses Vietnamese as main language for users and English for Staff, Manager and Admin on both Mobile and Web application.
* The user interface displays best on 1024x768 and above screen size

#### Hardware Interface

* Android Smartphone: Nexus 4 or later.

#### Software Interface

* Web application: work with Firefox (v30 or above), Chromes (v25 or above)
* Mobile Application: Android 4 (or later)

#### Communication Protocol

* Use HTTP protocol 1.1 for communication between the web browser and the web server.
* Use HTTP protocol 1.1 for communication between the mobile application and the web service.
* Use HTTP protocol 1.1 for communication between the server and the Microsoft service.

### System Overview Use Case

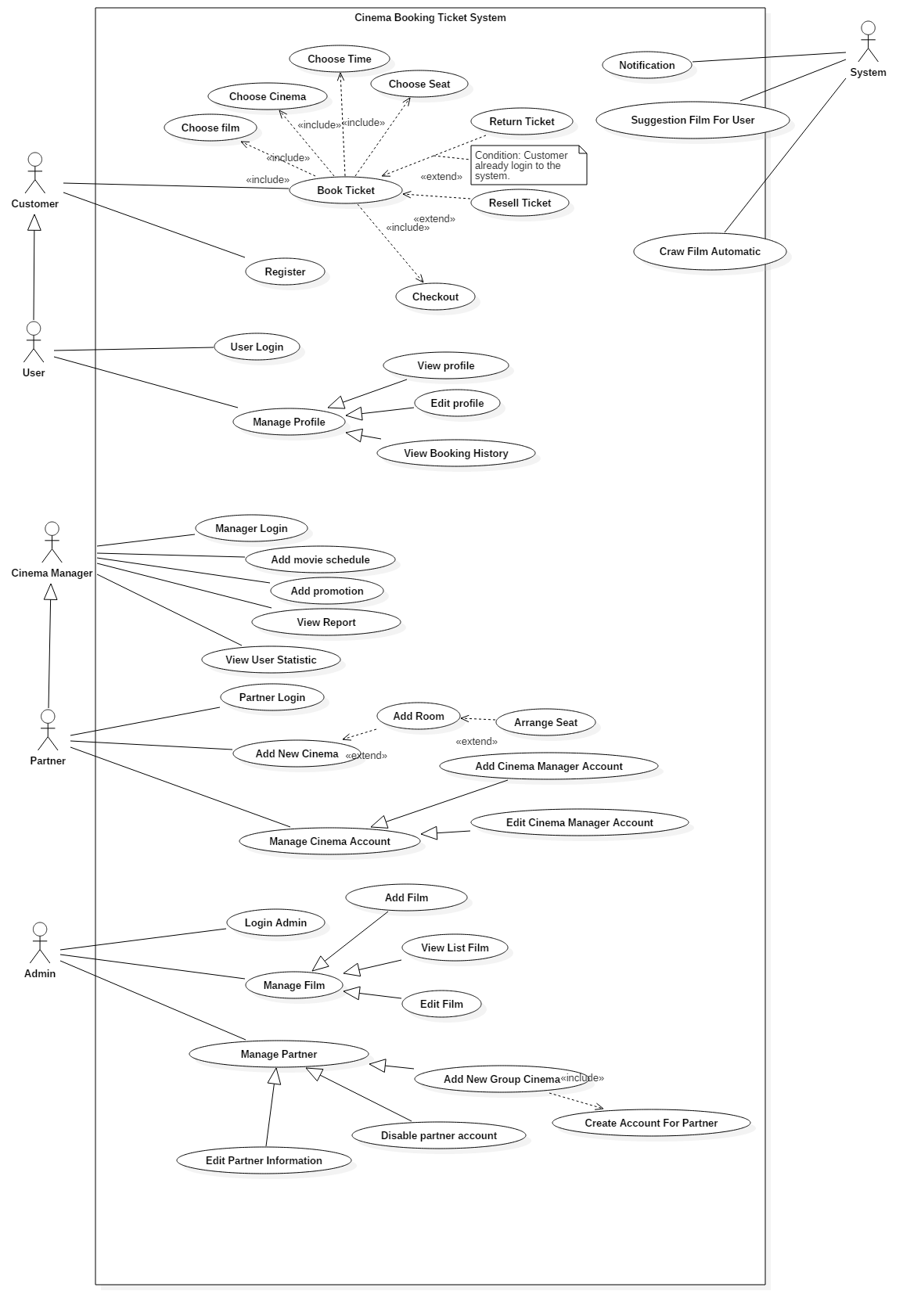


Figure 2: System Overview Use Case

### List of Use Case

#### <Admin> Overview Use Case

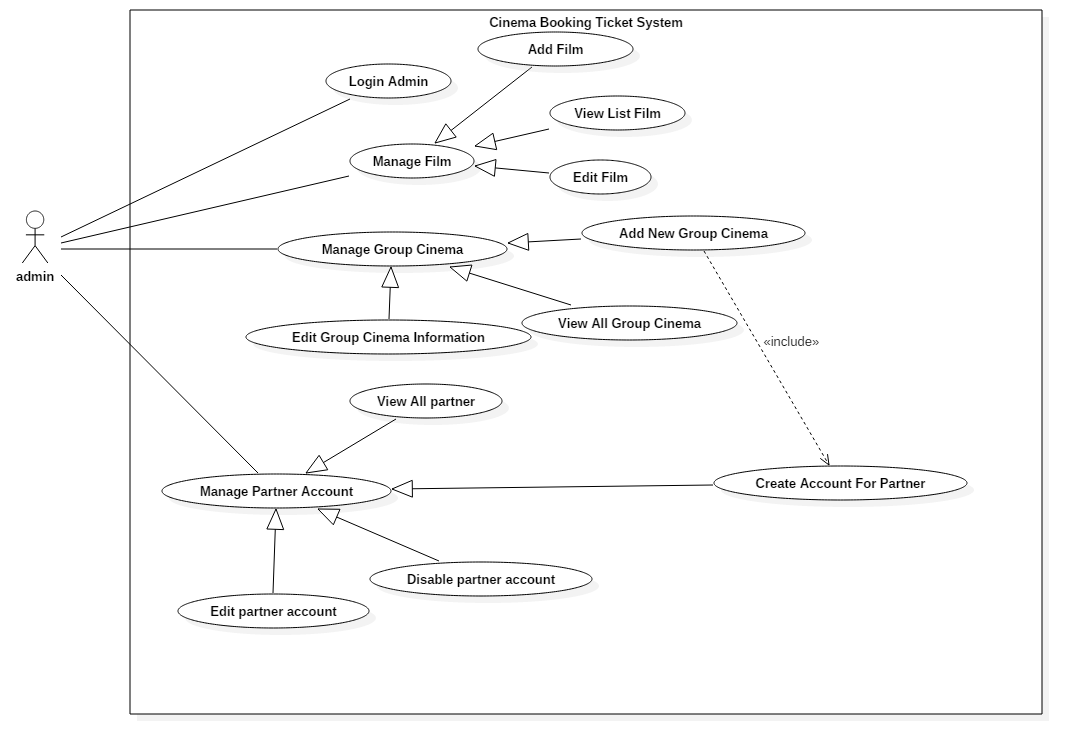


Figure 3: <Admin> Overview Use Case

##### <Amin> Login Admin (UC\_AD001)



Figure 4: <Admin> Login admin(UC\_AD001)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – AD\_011** | | | |
| **Use Case No.** | AD\_001 | **Use Case Version** | 2.0 |
| **Use Case Name** | Login Admin | | |
| **Author** | DanhLC | | |
| **Date** | 19/06/2018 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case helps actor signs in to their account.   **Goal:**   * Actor is signed in to the system.   **Triggers:**   * Actor sends sign in command.   **Preconditions:**   * Actor’s account must be in the system.   **Post Conditions:**   * **Success:** Actor sign in successfully * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor goes to login view. | System requires identity information from actor: - **Admin username**: free text input, required, regex: /^[A-Za-z0-9]+(?:[ \_-][A-Za-z0-9]+)\*$/ - **Password**: free text input, required, length 8 - 40 | | 2 | Actor inputs information |  | | 3 | Actor sends command to sign in to the system  [Alternative 1] | System validates inputted information  [Exception 1] | | 4 |  | System signed admin in |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Actor enters wrong identity information | Wrong identity information, system shows error message: “username or password is not correct”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid admin username format | System shows error message: “Password cannot be blank and must be 8 – 40 characters long”. | | 2 | Invalid password format | System shows error message: “Password cannot be blank and must be 8 – 40 characters long”. |   **Relationships:** N/A  **Business Rules:**   * Password is encrypted at server. * System will compare user’s encrypted string with the one stored in the system. If matched, user will be signed in with their admin rights. | | | |
|  | | | |

Table 16: USE CASE - UC\_AD001 - <Admin> Login

##### <Admin> View All Group Cinema (UC\_AD002)

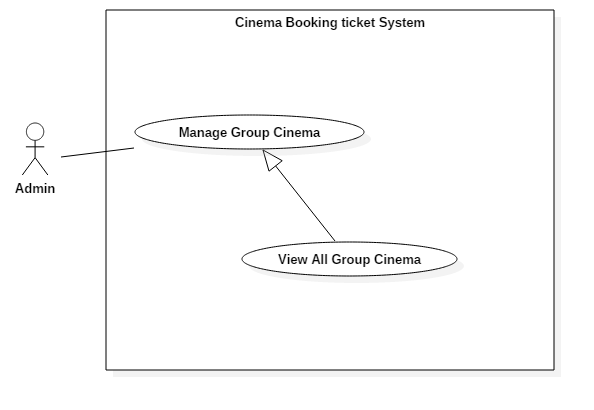


Figure 5: <Admin> View All Group Cinema (UC\_AD002)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_AD002** | | | |
| **Use Case No.** | UC\_AD002 | **Use Case Version** | 2.0 |
| **Use Case Name** | View all group cinema | | |
| **Author** | DanhLC | | |
| **Date** | 19/06/2018 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows actor to view all group cinema which already exist in the system. The list of all group cinema will display in table format, each row of the table will display basic information of it.   **Goal:**   * Display all group cinema   **Triggers:**   * Actor sends command to view all group cinema   **Preconditions:**   * Actor must login as Admin role.   **Post Conditions:**   * **Success:** Display list of all group cinema information. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor click “Manage group cinema” button. | System will display list of all group cinema as table, each row of table will display following information:   * **name**: text, read only. * **phone**: text, read only. * **Email**: text, read only. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:** N/A | | | |
|  | | | |

Table 17: USE CASE - UC\_AD002 - <Admin> View all group cinema

##### <Admin> Add new Group Cinema (UC\_AD003)

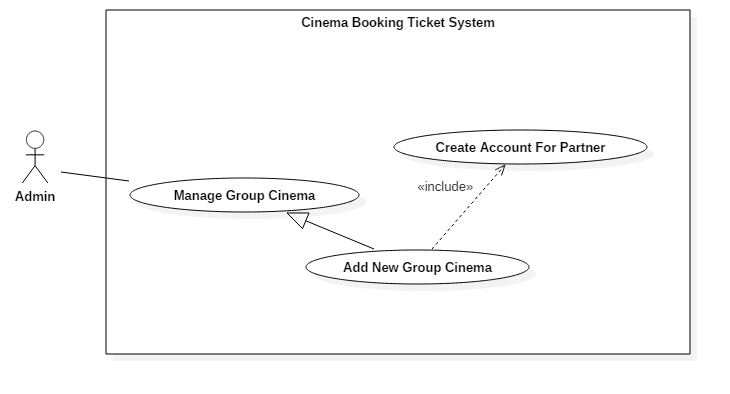


Figure 6: <Admin> Add new group cinema (UC\_AD003)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_AD003** | | | |
| **Use Case No.** | UC\_AD003 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add new group cinema | | |
| **Author** | DanhLC | | |
| **Date** | 19/06/2018 | **Priority** | High |
| **Actor:**   * Admin   **Summary:**   * This use case allows actor to add a new group cinema to the system. And when a cinema group was created, a Partner account will be created to manage this group cinema.   **Goal:**   * This function helps actor create a new group cinema and also create a partner account to manage this group cinema.   **Triggers:**   * Actor sends request to add new group cinema.   **Preconditions:**   * Actor must login admin first.   **Post Conditions:**   * **Success:** New group cinema was added to the system with its partner account. * **Fail:** System will display message below   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User click “manage group cinema” | System will display manage group cinema screen. | | 2 | User click “Add new group cinema” | System requires information from admin:   * **Logo Picture**: hidden free text input, required, length 8–40 characters. * **Group cinema name:** free text input, required, length 8–40 characters. * **Phone**: free text input, required, length 9-11 digits. * **Address**: free text input, required, length 8-500 characters.   **Email**: free text input, required, length 8-40 characters. | | 3 | Actor inputs information. |  | | 4 | Actor sends command to save.  [Alternative 1] | System validates information.  [Exception 1, 2, 3, 4, 5] | | 5 |  | System adds a new group cinema with inputted information and notify manager via email. |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Actor sends request to cancel. | System cancels adding new group cinema and redirect to home page. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid Group cinema name format. | System shows error message: “group cinema name cannot be blank and must be 8-40 characters long”. | | 2 | Invalid Address format. | System shows error message: “Address cannot be blank and must be 8-40 characters long”. | | 3 | Invalid Email format | System shows error message: “Email cannot be blank and must be 8-40 characters long”. | | 4 | Invalid Phone format. | System shows error message: “Phone cannot be blank or contain alphabetical characters and must be 9-11 digits, ”. | | 5 | Invalid Logo Picture format | System shows error message: “Picture cannot be empty and must be in jpeg/jpg/png format”. |   **Relationships:**   * Include Add new partner account: Actor has to manually add a partner account whenever a new cinema group was created.   **Business Rules:**   * Partner’s password will be randomly generated. * Randomly generated password includes alphanumeric, uppercase and lowercase. | | | |

Table 18; USE CASE -UC\_AD003 - <Admin> Add new store

##### <Admin> Edit group cinema information account (UC\_AD004)

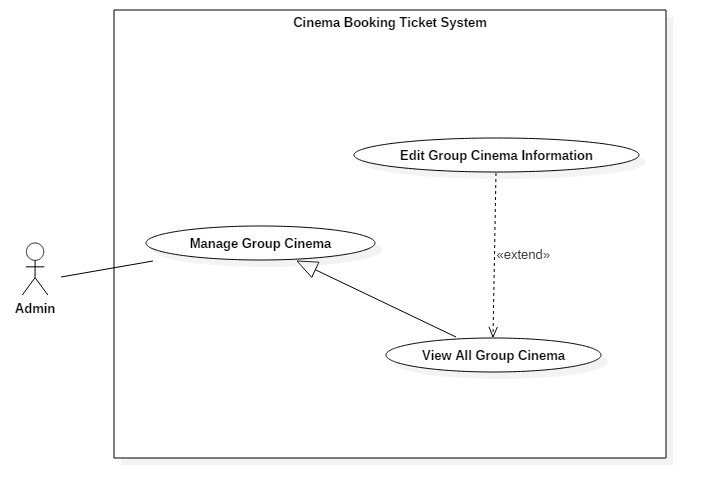


Figure 7: <Admin> Edit group cinema information(UC\_AD04)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_AD004** | | | |
| **Use Case No.** | UC\_AD004 | **Use Case Version** | 2.0 |
| **Use Case Name** | Edit group cinema information | | |
| **Author** | DanhLC | | |
| **Date** | 19/06/2018 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case helps actor to edit group cinema information, which have already been existed in the system.   **Goal:**   * To change to information of this group cinema.   **Triggers:**   * Actor sends edit group cinema request.   **Preconditions:**   * User login into system as Admin. * Group cinema must already be existed.   **Post Conditions:**   * **Success:** Group cinema will be change as the information that they has inputed. * **Fail:** System will display below messages   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home page, actor click “Manage group cinema” button | System show list group cinema in table format. | | 2 | Actor click “detail” button next to the group cinema want to edit. | System requires information from admin:   * **Logo Picture**: hidden free text input, required, length 8–255 characters. * **Group cinema name:** free text input, required, length 8–40 characters. * **Phone**: free text input, required, length 9-11 digits. * **Address**: free text input, required, length 8-500 characters. * **Email**: free text input, required, length 8-40 characters. | | 3 | Actor click “save” button.  [Alternative 1] | System validates information.  [Exception 1, 2, 3, 4, 5] | | 4 |  | System edit that group cinema with inputted information to database. |   **Alternative Scenario:**  [Alternative 1]   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Actor sends command to cancel. | System cancels edit group cinema action. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid Group cinema name format. | System shows error message: “Store name cannot be blank and must be 8-40 characters long”. | | 2 | Invalid Address format. | System shows error message: “Address cannot be blank and must be 8-500 characters long”. | | 3 | Invalid Email format | System shows error message: “Email cannot be blank and must be 8-40 characters long”. | | 4 | Invalid Phone format. | System shows error message: “Phone cannot be blank or contain alphabetical characters and must be 9-11 digits, ”. | | 5 | Invalid Picture format | System shows error message: “Picture cannot be empty and must be in jpeg/jpg/png format”. |   **Relationships:** N/A  **Business Rules:** N/A | | | |
|  | | | |
|  | | | |

Table 19: USE CASE - UC\_AD004 - <Admin> Edit Group Cinema Information

##### <Admin> View all Partner (UC\_AD05)

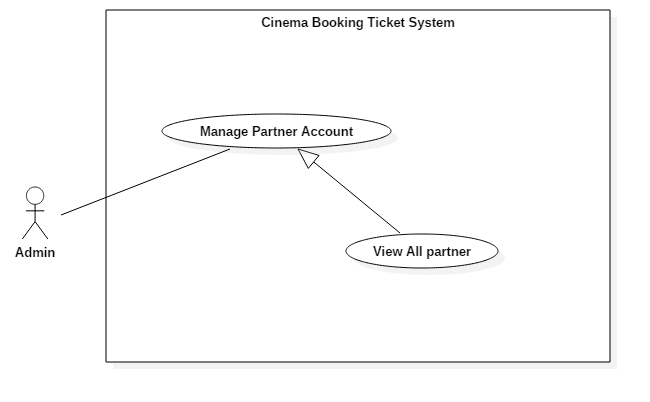


Figure 8: <Admin> View All Partner (UC\_AD005)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_AD005** | | | |
| **Use Case No.** | UC\_AD005 | **Use Case Version** | 2.0 |
| **Use Case Name** | View all Partner | | |
| **Author** | DanhLC | | |
| **Date** | 19/06/2018 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows actor to view all partner account that existing in the system. List of partners will be display as table form, each row will display the basic information of a specific partner account.   **Goal:**   * Successfully actor can view all list of partner account.   **Triggers:**   * Actor sends command to view all partner account.   **Preconditions:**   * Actor must login as Admin role.   **Post Conditions:**   * **Success:** Actor sends command to view all partner account. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home screen actor click “manage partner account” | System display a list of all partner account in a table form. Each row will display flowing information:   * **name**: text, read only. * **phone**: text, read only. * **Email**: text, read only. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:**   * Only category alias can be edited to prevent conflict with database. | | | |

Table 20: USE CASE - UC\_AD005 - <Admin> Edit Category

##### <Admin> Create Account for Partner (UC\_AD006)

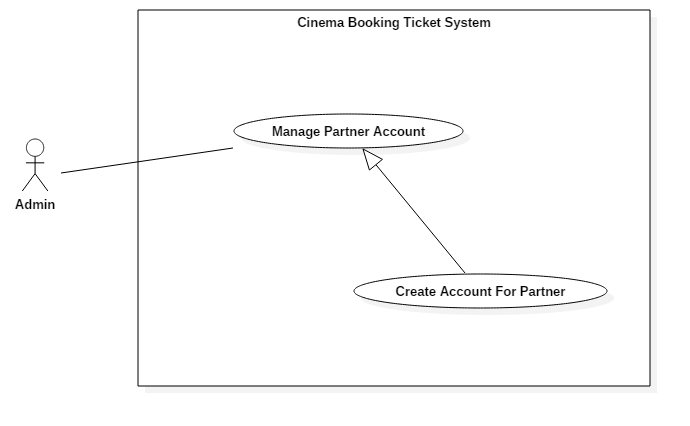


Figure 9: <Admin> Create Account for Partner (UC\_AD006)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_AD006** | | | |
| **Use Case No.** | UC\_AD006 | **Use Case Version** | 2.0 |
| **Use Case Name** | Create account for partner | | |
| **Author** | DanhLC | | |
| **Date** | 19/06/2018 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows actor to create new account for partner. This account will associate with one and only one group cinema. The account has been created will be used to manage that group cinema.   **Goal:**   * Successfully added new Partner account will be add to the system.   **Triggers:**   * Actor sends request to add Partner account.   **Preconditions:**   * Actor must login as Admin role.   **Post Conditions:**   * **Success:** New partner account will be added to the system. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home page, actor click “Manage partner account” button. | System will display manage partner view. | | 2 | Actor click “add partner” button. | System will have required input following field:   * **Partner username:** free text input, required, length 8–40 characters. * **Phone**: free text input, required, length 9-11 digits. * **Email**: free text input, required, length 8-40 characters. * **Group cinema Id**: select a cinema from the dropdown list | | 3 | Click button “add”.  [Alternative 1] | System validates information.  [Exception 1, 2, 3, 4, 5] | | 4 |  | System adds new partner account to database. |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Actor sends command to cancel. | System cancels adding new partner account. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid Partner username format. | System shows error message: “Store name cannot be blank and must be 8-40 characters long”. | | 2 | Invalid Email format | System shows error message: “Email cannot be blank and must be 8-40 characters long”. | | 3 | Invalid Phone format. | System shows error message: “Phone cannot be blank or contain alphabetical characters and must be 9-11 digits, ”. | | 4 | Partner username already exist | System shows error message: “This username already exist, please input another username”. |   **Relationships:** N/A  **Business Rules:**   * Partner account’s password will be auto generated by the system and send back to partner via email. | | | |
|  | | | |

Table 21: USE CASE - UC\_AD006 - <Admin> Create account for Partner

##### <Admin> Edit partner account (UC\_AD007)

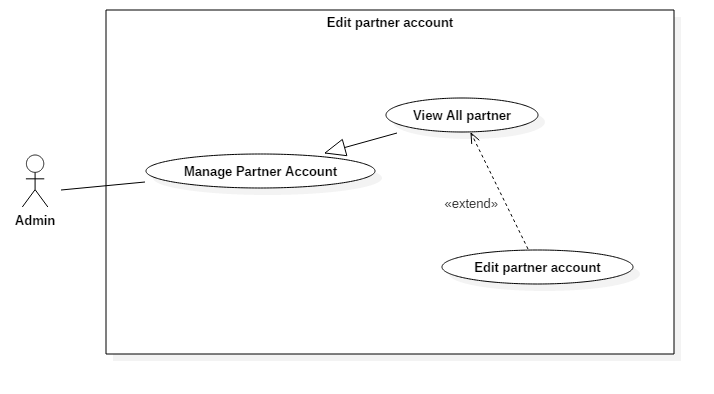


Figure 10: <Admin> Edit partner account (UC\_AD007)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_AD007** | | | |
| **Use Case No.** | UC\_AD007 | **Use Case Version** | 2.0 |
| **Use Case Name** | Edit partner account | | |
| **Author** | DanhLC | | |
| **Date** | 19/06/2018 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows actor to edit partner account information.   **Goal:**   * Successfully the information of specific partner account will be edited.   **Triggers:**   * Actor sends command to edit partner account.   **Preconditions:**   * Actor must login as Admin role.   **Post Conditions:**   * **Success:** Actor have to full fill information to edit partner account. * **Fail:** N/A.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home screen, Actor click “manage partner account” | System requires display list of partner account as table. | | 2 | Actor choose what account want to edit and click “detail” next to this account. | System will display a form. That form will full fill current information of this partner account. System requires information from admin:   * **Phone**: free text input, required, length 9-11 digits. * **Address**: free text input, required, length 8-500 characters. | | 3 | Actor click “save” button. | System validates information.  [Exception 1, 2, 3, 4, 5] | | 4 |  | System edit that partner account with inputted information to database. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid Email format | System shows error message: “Email cannot be blank and must be 8-40 characters long”. | | 2 | Invalid Phone format. | System shows error message: “Phone cannot be blank or contain alphabetical characters and must be 9-11 digits, ”. |   **Relationships:** N/A  **Business Rules:** N/A | | | |
|  | | | |

Table 22: USE CASE - UC\_AD007 - <Admin> Edit partner account

##### <Admin> Delete partner account (UC\_AD007)

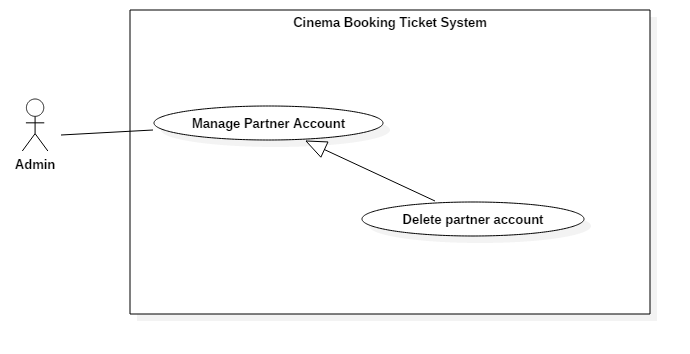


Figure 11: <Admin> Delete partner account (UC\_AD008)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_AD008** | | | |
| **Use Case No.** | UC\_AD008 | **Use Case Version** | 2.0 |
| **Use Case Name** | Delete partner account | | |
| **Author** | DanhLC | | |
| **Date** | 19/06/2018 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows actor to delete partner account that already exist to the system. When account was delete, partner cannot login to the system.   **Goal:**   * Successfully this partner account cannot be used to login Partner.   **Triggers:**   * Actor sends command to delete partner account.   **Preconditions:**   * Actor must login as Admin role.   **Post Conditions:**   * **Success:** N/A * **Fail:** N/A.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home screen, Actor click “Manage partner account” | System requires display list of partner account as table. | | 2 | Actor choose what account want to delete and click “delete” next to this account. | System change status of this account to unavailable. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 23: USE CASE - UC\_AD008 - <Admin> Disable partner account

##### <Admin> View List Film (UC\_AD009)

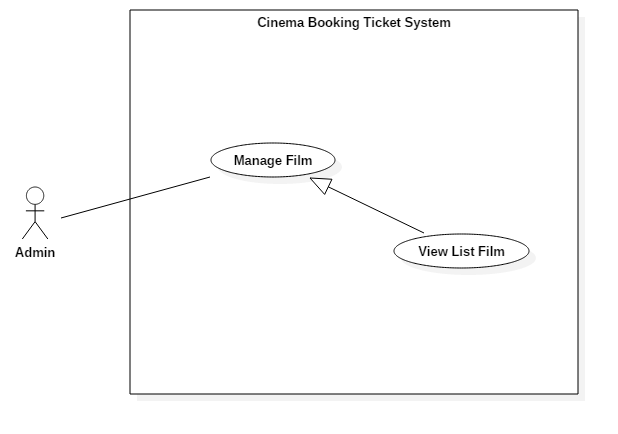


Figure 12: <Admin> View list film (UC\_AD009)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_AD009** | | | |
| **Use Case No.** | UC\_AD009 | **Use Case Version** | 2.0 |
| **Use Case Name** | View List Film | | |
| **Author** | DanhLC | | |
| **Date** | 19/06/2018 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows actor to view all list film that exist in the system. The start screen will show only current time film. You can select the time the see the older film.   **Goal:**   * Successfully Actor can view the list of film that exist in the system.   **Triggers:**   * Actor sends command to see all list film.   **Preconditions:**   * Actor must login as Admin role.   **Post Conditions:**   * **Success:** N/A * **Fail:** N/A.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home screen, Actor click “Manage Film”  *[Alternative 1]* | System will display list of all film as table, each row of table will display following information:   * **Film name**: text, read only. * **imdb**: text, read only. * **Release date**: text,read only. * **Author**: text, read only. * **Film status**: text, read only. |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | At home screen, Actor click “Manage Film” | System will display list of current time film and showing film for upcoming 6 months as table, each row of table will display following information:   * **Film name**: text, read only. * **imdb**: text, read only. * **Release date**: text, read only. * **Author**: text, read only. * **Film status**: text, read only. | | 2 | User select duration (from – to) that they want to search and click “search” button. | System will display list of all fill that matching with this duration. |   **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 24: USE CASE - UC\_AD009 - <Admin> Add Film

##### <Admin> Add Film (UC\_AD010)

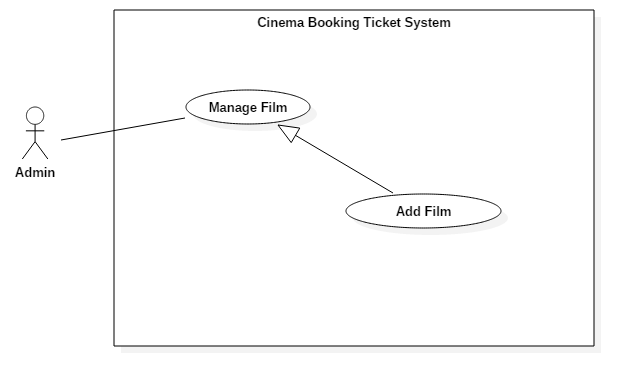


Figure 13: <Admin> Add Film(UC\_AD010)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_AD010** | | | |
| **Use Case No.** | UC\_AD010 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add Film | | |
| **Author** | DanhLC | | |
| **Date** | 19/06/2018 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows actor to add new film to the system. All the cinema can add schedule for this film if they have the right to show it.   **Goal:**   * Successfully Actor can edit the film that exist in the system.   **Triggers:**   * Actor sends command to add new film.   **Preconditions:**   * Actor must login as Admin role.   **Post Conditions:**   * **Success:** N/A * **Fail:** N/A.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home screen, Actor click “Manage Film” | System will display manages film screen. | | 2 | Click “Add Film” button | Input form will display, request admin input following information:   * **Film Name**: hidden free text input, required, length 8–40 characters. * **Date release:** date input required. * **Restricted**: select from dropdown following digit (13,16,18) * **Film length**: digit number. * **Imdb**: decimal number required. The number is limit from 0 -> 10. * **Digital type**: select digital type from check box. * **Author**: free text input. * **Movie genre**: free text input. * **Film content**: free text input, limit to 1000 character. * **Actor list**: free text input. * **Countries**: free text input. * **Trailer link**: hidden free text input, required, length 8–255 characters. * **Poster picture:** hidden free text input, required, length 8–255 characters. * **Addition picture:** hidden free text input, required, length 8–255 characters. * **Film Status:** hidden text input. Select from dropdown list. Default is showing. | | 3 | Actor inputs information. |  | | 4 | Actor sends command to save.  [Alternative 1] | System validates information.  [Exception 1, 2, 3, 4] | | 5 |  | System adds a new film with inputted information to the database and immediately update information for current screen film list. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid Group film name | System shows error message: “film name cannot be blank and must be 8-40 characters long”. | | 2 | Invalid release date | System shows error message: “release date cannot be blank”. | | 3 | Invalid poster picture | System shows error message: “Picture cannot be empty and must be in jpeg/jpg/png format”. | | 4 | Invalid addition picture | System shows error message: “Picture cannot be empty and must be in jpeg/jpg/png format”. |   **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 25: USE CASE - UC\_AD010 - <Admin> Add Film

##### <Admin> Edit Film (UC\_AD011)

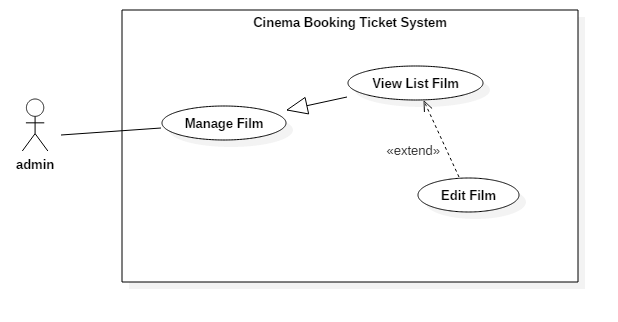


Figure 14: <Admin> Edit Film (UC\_AD011)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_AD011** | | | |
| **Use Case No.** | UC\_AD011 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add Film | | |
| **Author** | DanhLC | | |
| **Date** | 19/06/2018 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case allows actor to edit film that already exist in the system.   **Goal:**   * Successfully the film information will be edited.   **Triggers:**   * Actor sends command to edit film.   **Preconditions:**   * Actor must login as Admin role.   **Post Conditions:**   * **Success:** N/A * **Fail:** N/A.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home screen, Actor click “Manage Film” | System will display manages film screen. | | 2 | Click “Detail” button | System will display a form. That form will full fill current information of selected film. System requires information from admin:   * **Film Name**: hidden free text input, required, length 8–40 characters. * **Date release:** date input required. * **Restricted**: select from dropdown following digit (13,16,18) * **Film length**: digit number. * **Imdb**: decimal number required. The number is limit from 0 -> 10. * **Digital type**: select digital type from check box. * **Author**: free text input. * **Movie genre**: free text input. * **Film content**: free text input, limit to 1000 character. * **Actor list**: free text input. * **Countries**: free text input. * **Trailer link**: hidden free text input, required, length 8–255 characters. * **Poster picture:** hidden free text input, required, length 8–255 characters. * **Addition picture:** hidden free text input, required, length 8–255 characters. * **Film Status:** hidden text input. Select from dropdown list. Default is showing. | | 3 | Actor inputs information. |  | | 4 | Actor sends command to save. | System validates information.  [Exception 1, 2, 3, 4] | | 5 |  | System edit the film that was selected with inputted information, save it to the database and immediately update information for current screen film list. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid Group film name | System shows error message: “film name cannot be blank and must be 8-40 characters long”. | | 2 | Invalid release date | System shows error message: “release date cannot be blank”. | | 3 | Invalid poster picture | System shows error message: “Picture cannot be empty and must be in jpeg/jpg/png format”. | | 4 | Invalid addition picture | System shows error message: “Picture cannot be empty and must be in jpeg/jpg/png format”. |   **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 26: USE CASE - UC\_AD011 - <Admin> Edit Film

#### <Partner> Overview Use Case

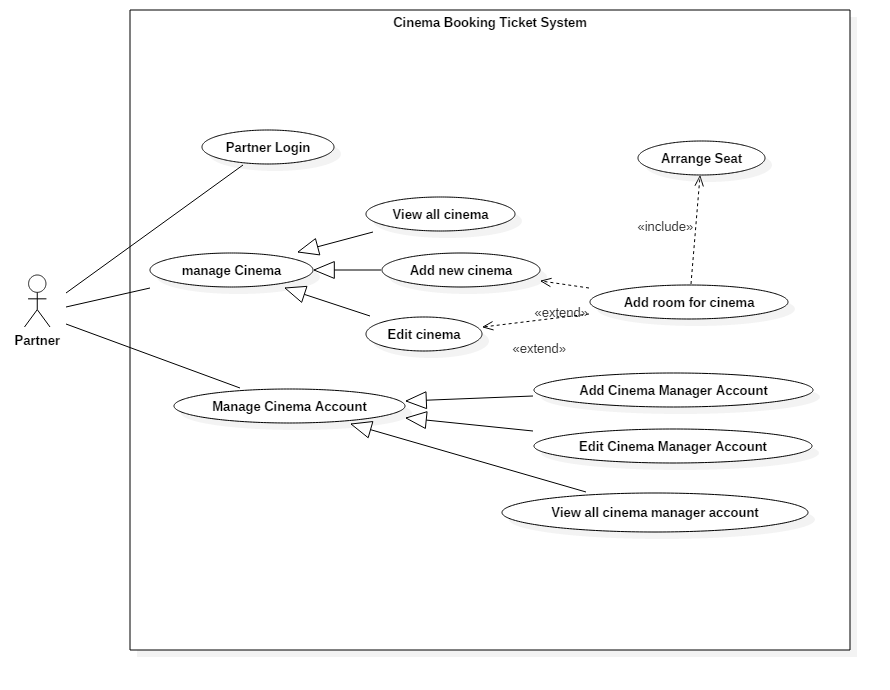


Figure 15: <User> Overview Use Case



##### <Partner> Partner login (UC\_P001)

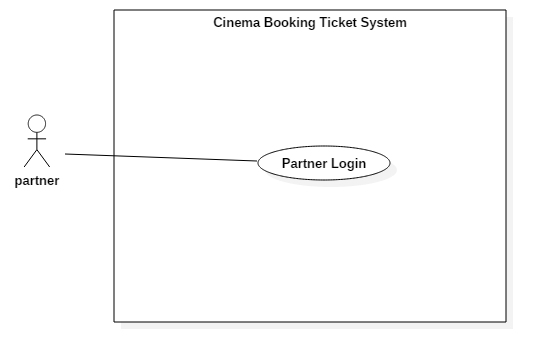


Figure 16: <Partner> Partner login (UC\_P001)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_P001** | | | |
| **Use Case No.** | UC\_P001 | **Use Case Version** | 2.0 |
| **Use Case Name** | Partner Login | | |
| **Author** | DanhLC | | |
| **Date** | 19/06/2018 | **Priority** | Normal |
| **Actor:**   * Admin   **Summary:**   * This use case helps actor signs in to their account.   **Goal:**   * Actor is signed in to the system.   **Triggers:**   * Actor sends sign in command.   **Preconditions:**   * Actor’s account must be in the system.   **Post Conditions:**   * **Success:** Actor sign in successfully * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor goes to login view. | System requires identity information from actor: - **Partner username**: free text input, required, regex: /^[A-Za-z0-9]+(?:[ \_-][A-Za-z0-9]+)\*$/ - **Password**: free text input, required, length 8 - 40 | | 2 | Actor inputs information |  | | 3 | Actor sends command to sign in to the system  [Alternative 1] | System validates inputted information  [Exception 1,2] | | 4 |  | System signed partner in |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Actor enters wrong identity information | Wrong identity information, system shows error message: “partner username or password is not correct”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid partner username format | System shows error message: “Password cannot be blank and must be 8 – 40 characters long”. | | 2 | Invalid password format | System shows error message: “Password cannot be blank and must be 8 – 40 characters long”. |   **Relationships:** N/A  **Business Rules:**   * Password is encrypted at server. * System will compare user’s encrypted string with the one stored in the system. If matched, user will be signed in with their partner rights. | | | |

Table 27: USE CASE - UC\_P001 - <Partner> Partner Login

##### <Partner>View All Cinema Manager (UC\_P002)

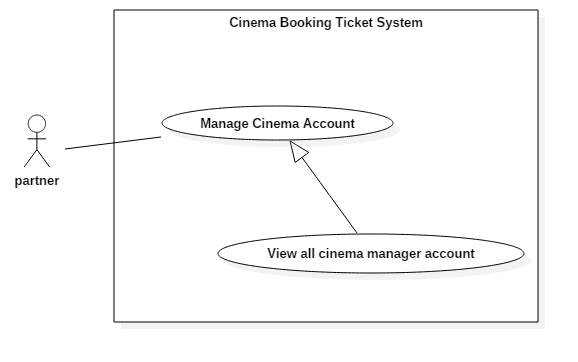


Figure 17: <Partner> View all cinema manager account (UC\_P002)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_P002** | | | |
| **Use Case No.** | UC\_P002 | **Use Case Version** | 2.0 |
| **Use Case Name** | View all cinema manager account | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | Normal |
| **Actor:**   * Partner   **Summary:**   * This use case allows actor to view list of all cinema manager as table form.   **Goal:**   * Actor can view list of all cinema manager.   **Triggers:**   * Actor sends command to view list of all cinema manager.   **Preconditions:**   * Actor must login as Partner.   **Post Conditions:**   * **Success:** The action is completed. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home screen, Actor click “Manage Cinema Manager Account” | System will display list of all cinema manager as table, each row of table will display following information:   * **Account username**: text, read only. * **Cinema name**: text, read only. * **phone**: text, read only. * **email**: text, read only. |   **Alternative Scenario: N/A**  **Exceptions: N/A**  **Relationships: N/A**  **Business Rules:** N/A | | | |

Table 28: USE CASE - UC\_P002 - <Partner> View all cinema manager account

##### <Partner> Add Cinema Manager Account (UC\_P003)

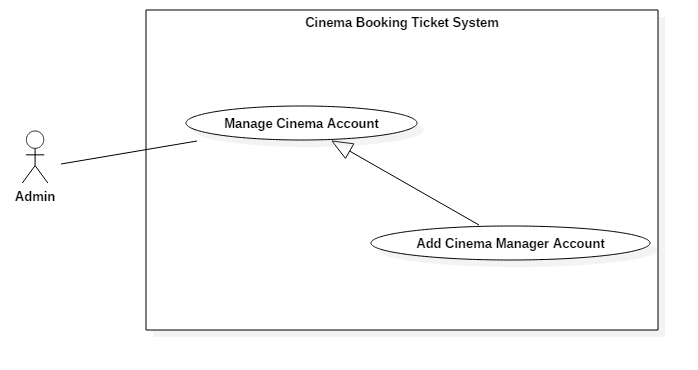


Figure 18: <Partner> Add Cinema Manager Account (UC\_P003)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_P003** | | | |
| **Use Case No.** | UC\_P003 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add Cinema Manager Account | | |
| **Author** | DanhLC | | |
| **Date** | 19/06/2018 | **Priority** | Normal |
| **Actor:**   * Partner   **Summary:**   * This use case allows actor to add new cinema manager account. Cinema manager is considered as child of partner. If partner manage a group of cinemas. Cinema manager will manage each cinema in this group cinema.   **Goal:**   * New cinema manager account will be added to the system.   **Triggers:**   * Actor sends command to add new cinema manager account.   **Preconditions:**   * Actor must login as Partner role.   **Post Conditions:**   * **Success:** Actor must full fill the information in right format and click “save”. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home screen, Actor click “Manage Cinema Manager Account” | System will display manages cinema manager account screen. | | 2 | Click “Add Cinema Manager Account” button | System will have required input following field:   * **Cinema manager username:** free text input, required, length 8–40 characters. * **Cinema manager password:** free text input, required, length 8–40 characters. * **Phone**: free text input, required, length 9-11 digits. * **Email**: free text input, required, length 8-40 characters. * **CinemaId**: select from dropdown list. Dropdown list will contain all cinema of current group cinema. | | 3 | Click button “add”. | System validates information.  [Exception 1, 2, 3, 4, 5] | | 4 |  | System adds new cinema manager account to database. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid cinema manager username format. | System shows error message: “cinema manager username cannot be blank and must be 8-40 characters long”. | | 2 | Invalid cinema manager password format | System shows error message: “password cannot be blank and must be 8-40 characters long”. | | 3 | Invalid email password format | System shows error message: “Email cannot be blank and must be 8-40 characters long”. | | 4 | Invalid Phone format. | System shows error message: “Phone cannot be blank or contain alphabetical characters and must be 9-11 digits, ”. | | 5 | Partner username already exist | System shows error message: “This username already exists, please input another username”. |   **Relationships:** N/A  **Business Rules:**   * After create, the account information will be automatic send back to cinema manager via email. | | | |

Table 29: USE CASE - UC\_P003 <Partner> Add Cinema Manager Account

##### <Partner>Delete Cinema Manager Account (UC\_P004)

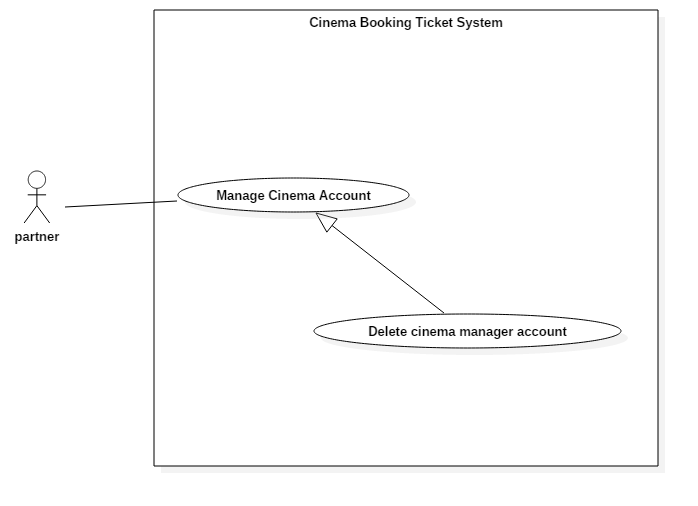


Figure 19: <Partner>Delete Cinema Manager Account (UC\_P004)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_P004** | | | |
| **Use Case No.** | UC\_P004 | **Use Case Version** | 2.0 |
| **Use Case Name** | Delete cinema manager account | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | Normal |
| **Actor:**   * Partner   **Summary:**   * This use case allows actor to delete cinema manger account. When cinema manager account has been deleted. Cinema manager cannot login with this account into the system. Partner also cannot see this account when click “view all cinema manager account”.   **Goal:**   * Delete cinema manager account.   **Triggers:**   * Actor sends command to delete cinema manager account.   **Preconditions:**   * Actor must login as partner role.   **Post Conditions:**   * **Success:** delete cinema manager account. * **Fail:** N/A.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home screen, Actor click “Manage Cinema Manager Account” | System will display list of cinema manager account in table form. | | 2 | Actor choose what cinema manager account want to delete and click “delete” next to this account. | System change status of this account to unavailable. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 30: USE CASE - UC\_P004 <Partner>Delete Cinema Manager Account

##### <Partner> Edit Cinema Manager Account (UC\_P005)

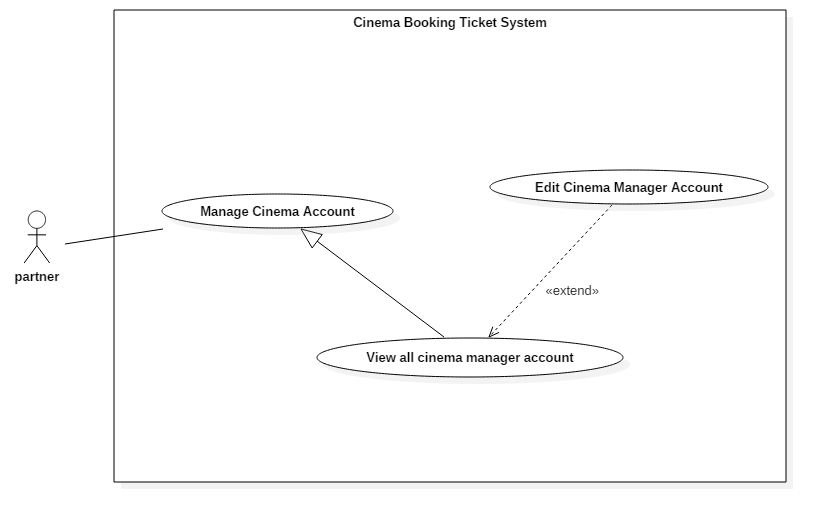


Figure 20: <Partner> Edit Cinema Manager Account (UC\_P005)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_P005** | | | |
| **Use Case No.** | UC\_P005 | **Use Case Version** | 2.0 |
| **Use Case Name** | Edit Cinema Manager Account | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | Normal |
| **Actor:**   * Partner   **Summary:**   * This use case allows actor to edit manager account information.   **Goal:**   * Update new information for selected cinema manager account.   **Triggers:**   * Actor sends command to edit cinema manager account.   **Preconditions:**   * Actor must login as Partner role.   **Post Conditions:**   * **Success:** Partner fill new information of cinema manager in right format. * **Fail:** System will display error messages below.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home screen, Actor click “Manage Cinema Manager Account” | System will display list of cinema manager account in table form. | | 2 | Click “Detail” button | System will show form fill current cinema manager account that was selected. System allow actor edit following field of this form:   * **Cinema manager password:** free text input, required, length 8–40 characters. * **Phone**: free text input, required, length 9-11 digits. * **Email**: free text input, required, length 8-40 characters. * **CinemaId**: select from dropdown list. Dropdown list will contain all cinema of current group cinema. | | 3 | Click button “save”. | System validates information.  [Exception 1, 2, 3] | | 4 |  | System update new information for the selected cinema manager account to database. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid cinema manager password format | System shows error message: “password cannot be blank and must be 8-40 characters long”. | | 2 | Invalid email password format | System shows error message: “Email cannot be blank and must be 8-40 characters long”. | | 3 | Invalid Phone format. | System shows error message: “Phone cannot be blank or contain alphabetical characters and must be 9-11 digits, ”. |   **Relationships:** N/A  **Business Rules:** After create, the account information will be automatic send back to cinema manager via email. | | | |

Table 31: USE CASE - UC\_P005 <Partner> Edit Cinema Manager Account

##### <Partner> View all cinema (UC\_P006)

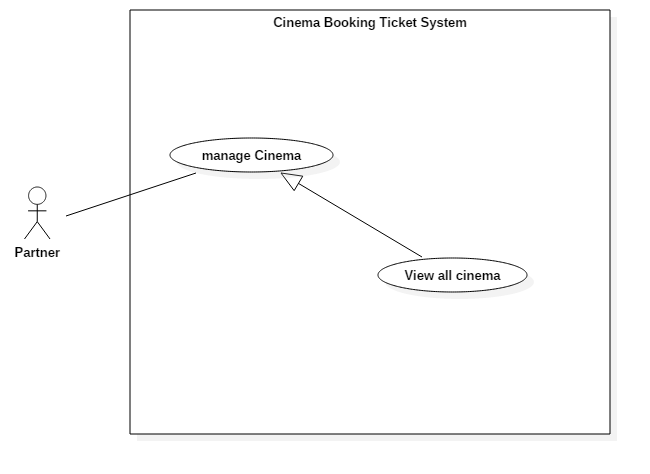


Figure 21: <Partner> View all cinema (UC\_P006)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_P006** | | | |
| **Use Case No.** | UC\_P006 | **Use Case Version** | 2.0 |
| **Use Case Name** | View All Cinema | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows actor to view all list of cinemas that belong to this group cinema. Each cinema will manage by cinema manager.   **Goal:**   * Actor can view all cinema belong to their cinema group.   **Triggers:**   * Actor sends command to view all cinema group.   **Preconditions:**   * Actor must login as Partner role.   **Post Conditions:**   * **Success:** Actor click “view all cinema”. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home screen, Actor click “Manage Cinema” | System will display list of all cinema as table, each row of table will display following information:   * **Cinema name**: text, read only. * **Cinema address**: text, read only. * **phone**: text, read only. * **email**: text, read only. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 32: USE CASE - UC\_P006 <Partner> View all cinema

##### <Partner> Arrange seat (UC\_P007)

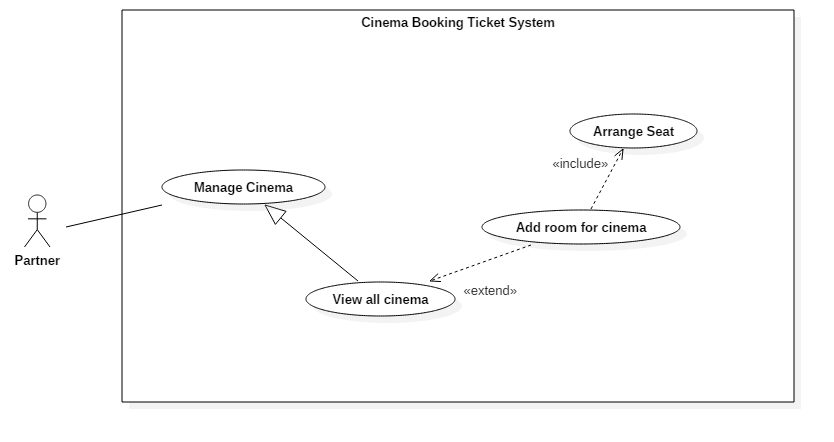


Figure 22: <Partner> Arrange seat (UC\_P007)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_P007** | | | |
| **Use Case No.** | UC\_P007 | **Use Case Version** | 2.0 |
| **Use Case Name** | Arrange seat | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | Normal |
| **Actor:**   * Partner   **Summary:**   * When partner create a cinema. They may want to add room to their cinema. And when a room is created, we allow partner to arrange seat for this room. The strategy for this is as following. First of all, when create room, we ask them for input dimension x and y and capacity of this room. We will create a room template match with what partner has inputted above. The room will contain zero seat available. After that, partner can add seat to this template. If there is no seat add in an area, this area will become an empty area.   **Goal:**   * To arrange seat for a room of specific cinema.   **Triggers:**   * Actor sends command to add arrange seat for cinema.   **Preconditions:**   * Actor must login as Partner.   **Post Conditions:**   * **Success:** The action is completed. * **Fail:** System will display error messages below.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor choose which seat that they want to add on the seat matrix template.  [Alternative 1] | A seat was added at the position that they click. The color of this seat will change from gray to orange meaning available. | | 2 | Partner continue repeat step 1 until the number of seat full fill more than 50% of total capacity of the room. |  | | 3 | Partner click “save” button | System validates information.  [Exception 1, 2, 3, 4] | | 4 |  | System will save the matrix of seat in to the database. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Actor choose which seat that they want to remove on the seat matrix template. | A seat was remove at the position that they click. The color of this seat will change from orange to gray meaning unavailable. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Number of seat choose not full fill more than 50% of total capacity of the room. | System shows message: “The seat you choose must more than 50% of total capacity of this room”. |   **Relationships:**   * Include with add room to cinema: when user want to add room to cinema they must to arrange seat for it.   **Business Rules:** Number of seat must full fill more than 50% of total capacity of the room. | | | |

Table 33: USE CASE - UC\_P007 <Partner> Arrange seat

##### <Partner> Add Room for Cinema (UC\_P008)

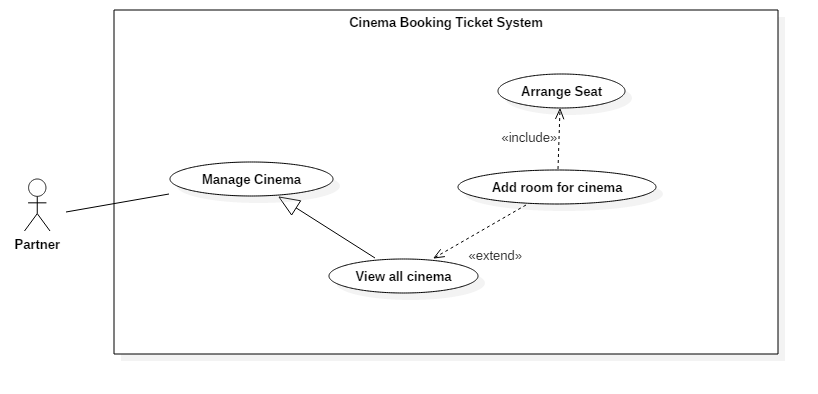


Figure 23: <Partner> Add Room for Cinema (UC\_P008)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_P008** | | | |
| **Use Case No.** | UC\_P008 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add room for cinema | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | Normal |
| **Actor:**   * Partner   **Summary:**   * This use case allows actor to add room to specific cinema. When list user show, they can extend to add room for this cinema.   **Goal:**   * This use case help actor to add a room to specific cinema.   **Triggers:**   * Actor sends command to add room for cinema.   **Preconditions:**   * Actor must login as Partner.   **Post Conditions:**   * **Success:** The action is completed. * **Fail:** System will display error messages below.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home screen, actor click button “manage cinema”. | System display screen for manage cinema. The list of cinemas will show as a table. | | 2 | Select a row represent for a cinema and click “add room” button at the end of this row. | System will show form for add room and required following input:   * **Capacity:** free number input, value is from 50 to 400. * **Room’s width**: free number input, required, value is from 5 to 20. * **Room’s height**: free number input, required, value is from 5 to 20. | | 3 | Actor sends command to save. | System validates information.  [Exception 1, 2, 3, 4] | | 4 |  | System adds a new room for the selected cinema with inputted information. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Invalid capacity input. | System shows message: “capacity must be number and value is between 50 and 400”. | | 2 | Invalid Room’s width input | System shows message: “dimension X must be number and value is between 5 and 20” | | 3 | Invalid Room’s height input | System shows message: “dimension X must be number and value is between 5 and 20” | | 4 | Capacity input is less than multiple of width and height. | System shows message: “capacity must less than multiple of width and height”. |   **Relationships:**   * Extend from view all cinema: when user want to show list of cinemas, they can extend for add room for this cinema. * Include with arrange seat. When a room is created they must arrange seat for it.   **Business Rules:**   * Once you click save, you cannot edit seat position any more. | | | |

Table 34: USE CASE - UC\_P008 <Partner> Add Room for Cinema

##### <Partner> Add New Cinema (UC\_P009)



Figure 24: <Partner> Add New Cinema (UC\_P009)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_P009** | | | |
| **Use Case No.** | UC\_P009 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add new cinema | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | Normal |
| **Actor:**   * Partner   **Summary:**   * This use case allows actor to add new cinema for current group cinema.   **Goal:**   * To add new cinema for current group cinema.   **Triggers:**   * Actor sends command to add new cinema.   **Preconditions:**   * Actor must login as Partner role.   **Post Conditions:**   * **Success:** The action is completed. * **Fail:** N/A.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home, click button “mange cinema”. | System display list of cinemas in table form. | | 2 | Click “add new” button. | System will have required input following field:   * **Cinema name:** free text input, required, length 8–40 characters. * **Cinema address:** free text input, required, length 8–255 characters. * **Phone**: free text input, required, length 9-11 digits. * **Email**: free text input, required, length 8-40 characters. * **Introduction**: free text input, required, length 0-1000 characters. * **Profile picture:** hidden free text input, required, length 8–255 characters. | | 3 | Actor inputs information. |  | | 4 | Actor sends command to save. | System validates information.  [Exception 1, 2, 3, 4,5] | | 5 |  | System adds a new cinema with inputted information to the database and immediately update information for current screen cinema list. |   **Alternative Scenario:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid Address format. | System shows error message: “Address cannot be blank and must be 8-40 characters long”. | | 2 | Invalid Email format | System shows error message: “Email cannot be blank and must be 8-40 characters long and must match email format”. | | 3 | Invalid Phone format. | System shows error message: “Phone cannot be blank or contain alphabetical characters and must be 9-11 digits, ”. | | 4 | Invalid Name format. | System shows error message: “Name cannot be blank and must be 8-40 characters long”. | | 5 | Invalid profile picture | System shows error message: “Picture cannot be empty and must be in jpeg/jpg/png format”. |   **Relationships: N/A**  **Business Rules:** N/A | | | |

Table 35: USE CASE - UC\_P009 <Partner> Add New Cinema

##### <Partner> Edit cinema (UC\_P010)

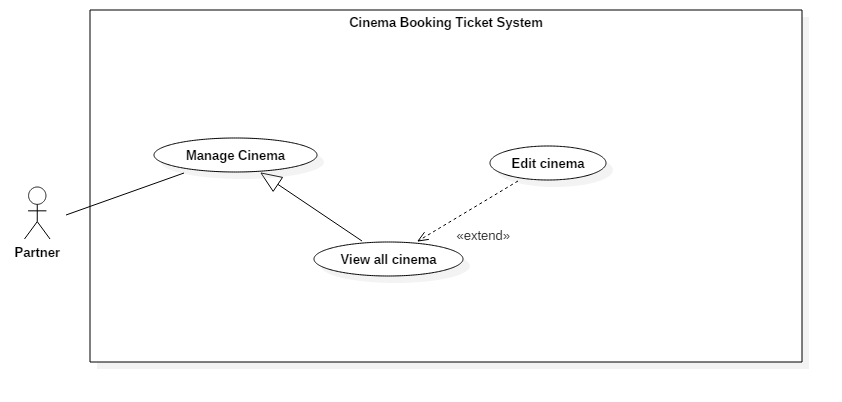


Figure 25: <Partner> Edit cinema (UC\_P010)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_P010** | | | |
| **Use Case No.** | UC\_P010 | **Use Case Version** | 2.0 |
| **Use Case Name** | Edit cinema | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | Normal |
| **Actor:**   * Partner   **Summary:**   * This use case allows actor to edit cinema information.   **Goal:**   * Update new information for selected cinema.   **Triggers:**   * Actor sends command to edit cinema information.   **Preconditions:**   * Actor must login as Partner.   **Post Conditions:**   * **Success:** The action is completed. * **Fail:** Show error message as below.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home screen, Actor click “Manage Cinema” | System will display manages cinema screen. | | 2 | Click “Detail” button | System will display a form. That form will full fill current information of selected cinema. System requires information from partner:   * **Cinema name:** free text input, required, length 8–40 characters. * **Cinema address:** free text input, required, length 8–255 characters. * **Phone**: free text input, required, length 9-11 digits. * **Email**: free text input, required, length 8-40 characters. * **Introduction**: free text input, required, length 0-1000 characters. * **Profile picture:** hidden free text input, required, length 8–255 characters. | | 3 | Actor inputs information. |  | | 4 | Actor sends command to save. | System validates information.  [Exception 1, 2, 3, 4] | | 5 |  | System update information for selected cinema with inputted information to the database and immediately update information for current screen cinema list.. |   **Alternative Scenario:**  **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid Address format. | System shows error message: “Address cannot be blank and must be 8-40 characters long”. | | 2 | Invalid Email format | System shows error message: “Email cannot be blank and must be 8-40 characters long and must match email format”. | | 3 | Invalid Phone format. | System shows error message: “Phone cannot be blank or contain alphabetical characters and must be 9-11 digits, ”. | | 4 | Invalid Name format. | System shows error message: “Name cannot be blank and must be 8-40 characters long”. | | 5 | Invalid profile picture | System shows error message: “Picture cannot be empty and must be in jpeg/jpg/png format”. |   **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 36: USE CASE - UC\_P010 <Partner> Edit cinema

##### <Partner> View Room List (UC\_P011)

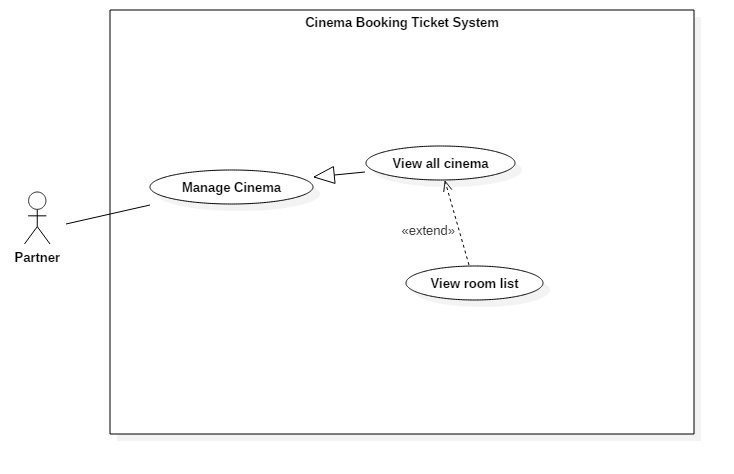


Figure 26: <Partner> View Room List (UC\_P011)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_P011** | | | |
| **Use Case No.** | UC\_P011 | **Use Case Version** | 2.0 |
| **Use Case Name** | View room list | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | Normal |
| **Actor:**   * Partner   **Summary:**   * This use case allows actor view all room of specific cinema.   **Goal:**   * View all room of specific cinema.   **Triggers:**   * Actor sends command to view all room.   **Preconditions:**   * Actor must login as Partner.   **Post Conditions:**   * **Success:** The action is completed * **Fail:** N/A.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home screen, Actor click “Manage Cinema” | System will display list of all cinema as table. | | 2 | Actor click “detail” button at the end of each row | System show all room belong to this cinema as table format. This table will stay below cinema list table. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 37: USE CASE UC\_P011 <Partner> View Room List

##### <Partner> View Room Detail (UC\_P012)

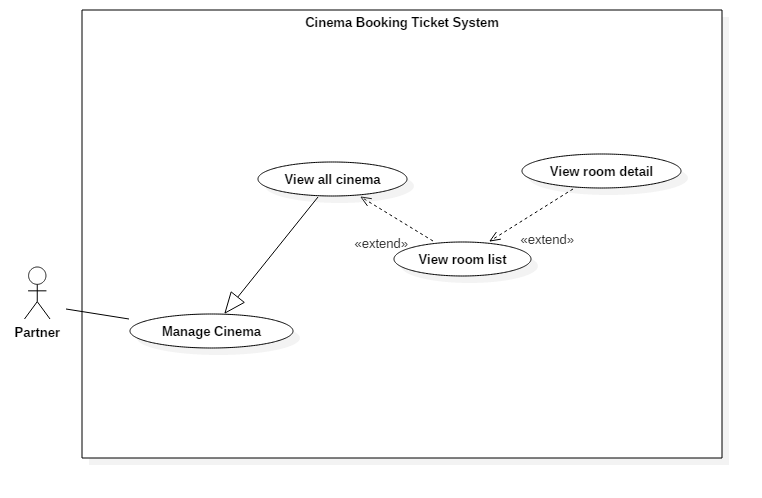


Figure 27: <Partner> View Room Detail (UC\_P012)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_P012** | | | |
| **Use Case No.** | UC\_P012 | **Use Case Version** | 2.0 |
| **Use Case Name** | View room detail | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | Normal |
| **Actor:**   * Partner   **Summary:**   * This use case allows actor view all room of specific cinema.   **Goal:**   * View all room of specific cinema.   **Triggers:**   * Actor sends command to view all room.   **Preconditions:**   * Actor must login as Partner.   **Post Conditions:**   * **Success:** The action is completed * **Fail:** N/A.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home screen, Actor click “Manage Cinema” | System will display list of all cinema as table. | | 2 | Actor click “detail” button at the end of each row of cinema table. | System show all room belong to this cinema as table format. This table will stay below cinema list table. | | 3 | Actor click “detail” button at the end of each row of room table. | System give the specific view about the room which user has added before. | | 4 | Actor click “finish view” | Go back to manage cinema screen. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 38: USE CASE - UC\_P012 <Partner> View Room Detail

#### <User> Overview Use Case

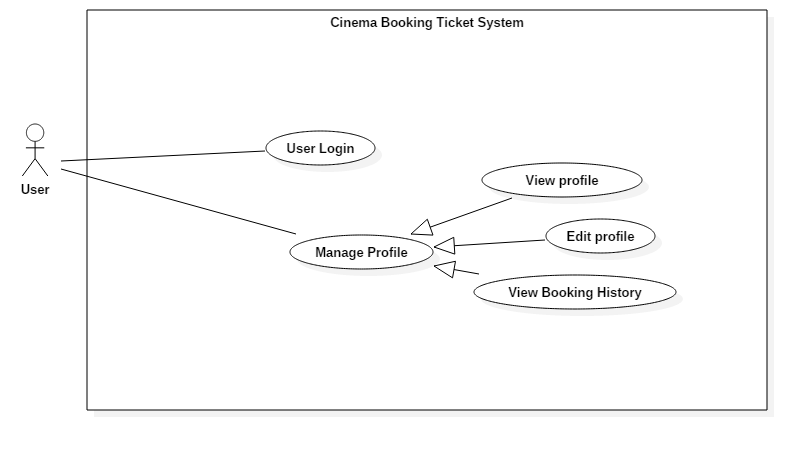


Figure 28: <User> Overview Use Case

##### <User> User Login (UC\_U001)

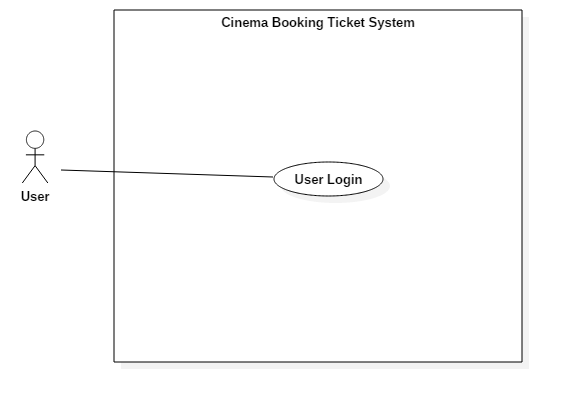


Figure 29: <User> User Login (UC\_U001)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – U001** | | | |
| **Use Case No.** | UC\_U001 | **Use Case Version** | 2.0 |
| **Use Case Name** | User Login | | |
| **Author** | DanhLC | | |
| **Date** | 19/06/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case helps actor signs in to their account.   **Goal:**   * Actor is signed in to the system.   **Triggers:**   * Actor sends sign in command.   **Preconditions:**   * Actor’s account must be in the system.   **Post Conditions:**   * **Success:** Actor sign in successfully * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor goes to login view. | System requires identity information from actor: - **Admin username**: free text input, required, regex: /^[A-Za-z0-9]+(?:[ \_-][A-Za-z0-9]+)\*$/ - **Password**: free text input, required, length 8 - 40 | | 2 | Actor inputs information |  | | 3 | Actor sends command to sign in to the system  [Alternative 1] | System validates inputted information  [Exception 1] | | 4 |  | System signed user in |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Actor enters wrong identity information | Wrong identity information, system shows error message: “username or password is not correct”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid admin username format | System shows error message: “Password cannot be blank and must be 8 – 40 characters long”. | | 2 | Invalid password format | System shows error message: “Password cannot be blank and must be 8 – 40 characters long”. |   **Relationships:** N/A  **Business Rules:**N/A | | | |

Table 39: USE CASE - UC\_U001 <User> User Login

##### <User> View Profile (UC\_U002)

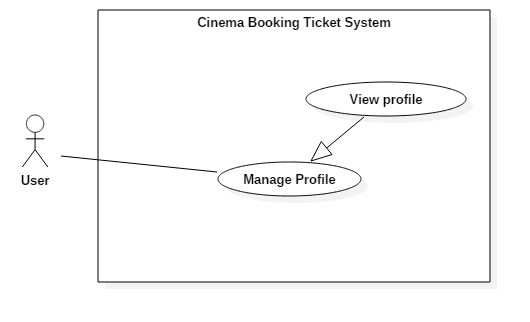
****

Figure 30: <User> View Profile (UC\_U002)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_U002** | | | |
| **Use Case No.** | UC\_U002 | **Use Case Version** | 2.0 |
| **Use Case Name** | View Profile | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows actor to view their information.   **Goal:**   * View profile information.   **Triggers:**   * Actor sends command to view profile.   **Preconditions:**   * Actor must login as User.   **Post Conditions:**   * **Success:** The action is completed. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home screen, Actor click “view profile” | System will display their profile information. The information will include:   * **username**: text, read only. * **phone** text, read only. * **email**: text, read only. * **tix**: text, read only. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 40: USE CASE - UC\_U002 <User> View Profile

##### <User> Edit profile (UC\_U003)

****

Figure 31: <User> Edit profile (UC\_U003)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_U003** | | | |
| **Use Case No.** | UC\_U003 | **Use Case Version** | 2.0 |
| **Use Case Name** | Edit Profile | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows actor to edit their user profile.   **Goal:**   * Edit profile information for user.   **Triggers:**   * Actor sends command to edit profile.   **Preconditions:**   * Actor must login as User.   **Post Conditions:**   * **Success:** Update current user information with newly input data. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor go to view profile | System will display their profile information. The information will include:   * **username**: text, read only. * **phone** text, read only. * **email**: text, read only. * **tix**: text, read only. | | 2 | Actor click “edit” button. | System requires information from partner:.   * **Phone**: free text input, required, length 9-11 digits. * **Email**: free text input, required, length 8-40 characters. | | 3 | Actor inputs information |  | | 4 | Actor sends command to edit staff | System validates information.  [Exception 1, 2, 3, 4] | | 5 |  | System updates staff with newly inputted information |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid Email format | System shows error message: “Email cannot be blank and must be 8-40 characters long and must match email format”. | | 2 | Invalid Phone format. | System shows error message: “Phone cannot be blank or contain alphabetical characters and must be 9-11 digits, ”. |   **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 41: USE CASE - UC\_U003 <User> Edit profile

##### <User> View Booking History (UC\_U004)

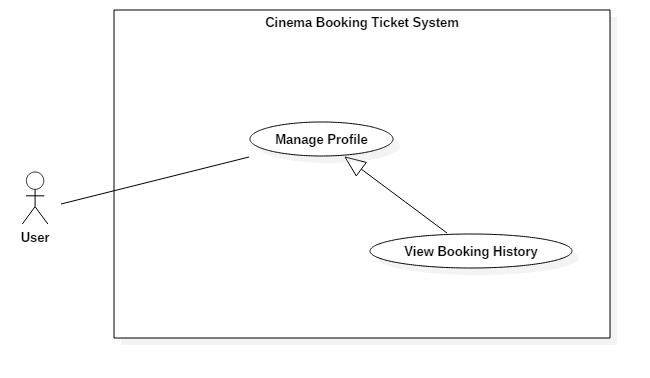


Figure 32: <User> View Booking History (UC\_U004)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_U004** | | | |
| **Use Case No.** | UC\_ | **Use Case Version** | 2.0 |
| **Use Case Name** | View Booking History | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case allows actor to view their booking history.   **Goal:**   * Actor can view their booking history.   **Triggers:**   * Actor sends command to view booking history.   **Preconditions:**   * Actor must login as User.   **Post Conditions:**   * **Success:** The action is completed. * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor goes to profile view. | System display profile screen. | | 2 | Actor click “view booking history” button. | System will display their booking history. The information will display in table form. The information will include:   * **Booking date**: text, read only. * **Film Name:** text, read only. * **Ticket quantity:** text, read only. * **Total amount**: text, read only. |   **Alternative Scenario:** N/A  **Exceptions:** N/A  **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 42: : USE CASE - UC\_U004 <User> View Booking History

##### <User> Sign out (UC\_U005)



Figure 33: <User> Sign out (UC\_U005)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_U005** | | | |
| **Use Case No.** | UC\_U005 | **Use Case Version** | 2.0 |
| **Use Case Name** | Sign out | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | Normal |
| **Actor:**   * User   **Summary:**   * This use case helps User sign out of their accounts.   **Goal:**   * Actor is signed out.   **Triggers:**   * Actor sends sign out command.   **Preconditions:**   * Actor must be logged in into system.   **Post Conditions:**   * **Success:** Account is signed out successfully * **Fail:** Account is still signed out   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor clicks sign out button.  [Alternative 1] | System signs actor out and redirects to Home page |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 |  | System automatically signs actor out when actor’s session is over |   **Relationships:** N/A  **Business Rules:**   * System destroys actor’s session and redirects to Home page. | | | |

Table 43: USE CASE - UC\_U005 <User> Sign out

#### <Customer> Overview Use Case



Figure 34: <Customer> Overview Use Case

##### <Customer> Register (UC\_C001)

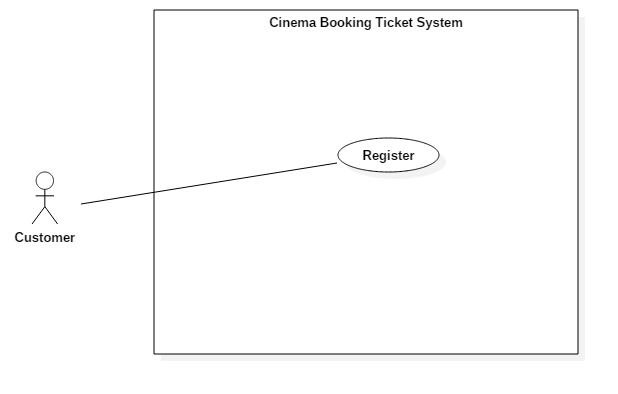


Figure 35: <Customer> Register (UC\_C001)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_C001** | | | |
| **Use Case No.** | UC\_C001 | **Use Case Version** | 2.0 |
| **Use Case Name** | Register | | |
| **Author** | DanhLC | | |
| **Date** | 17/01/2016 | **Priority** | Normal |
| **Actor:**   * Customer   **Summary:**   * This use case helps Customer registers a new account.   **Goal:**   * Actor can create a new account in the system.   **Triggers:**   * Actor sends sign up command.   **Preconditions:**   * N/A   **Post Conditions:**   * **Success:** Account is created successfully * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor goes to sign up view. | System requires some information from actor:  - **Username**: free text input, required, length 8 - 40  **- Email**: free text input, required, length 8 – 40, regex: /\A([\w+\-].?)+@[a-z\d\-]+(\.[a-z]+)\*\.[a-z]+\z/i - **Password**: free text input, required, length 8 - 40  -**Confirm** **password**: free text input, required, length 8 - 40 | | 2 | Actor inputs information |  | | 3 | Actor sends sign up command | System validates inputted information.  [Exception 1, 2, 3] | | 4 |  | System creates an account with Actor’s email |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | username already exists in the system | System shows error message “Account already exists”. | | 2 | Invalid username format. | System shows error message: “Username cannot blank and must be 8 – 40 character longs”. | | 3 | Invalid email format | System shows error message: “Not a valid email format”. | | 4 | Invalid password format | System shows error message: “Password cannot be blank and must be 8 – 40 characters long”. | | 5 | Confirm password does not match. | System shows error message: “These passwords do not match”. |   **Relationships:** N/A  **Business Rules:**   * After sign in to the system, actor will be redirected to home page. | | | |

Table 44: USE CASE - UC\_C001 <Customer> Register

##### <Customer> Find Schedule (UC\_C002)

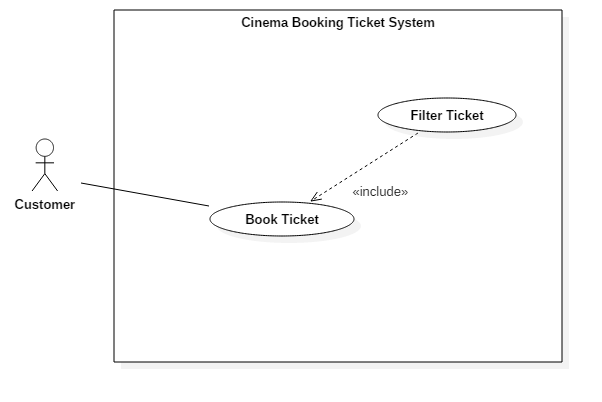


Figure 36: <Customer> Find Schedule (UC\_C002)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_C002** | | | |
| **Use Case No.** | UC\_C002 | **Use Case Version** | 2.0 |
| **Use Case Name** | Filter Ticket | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | High |
| **Actor:**   * Customer   **Summary:**   * This use case helps Actor to filter ticket by film, by group cinema, by time and date. After that user can easy choose the ticket that they want to buy.   **Goal:**   * Actor can easily choose suitable ticket.   **Triggers:**   * Actor can send request to filter ticket.   **Preconditions: N/A**  **Post Conditions:**   * **Success:** The action is completed * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Customer choose the film that they want to buy ticket. | System goes to detail film page. At the end of the page will display quick navigate to select. | | 2 | Click on to group cinema on quick navigate. | System will display all cinema of this group cinema on quick navigate. | | 3 | Click on cinema on list cinema response by previous step. | System recommend all Show Time filter by current film, cinema. | | 4 | Click on date on the top of film ‘s quick navigate | System recommend all Show Time filter by current film, cinema, date selected. |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | In the quick filter at home page. User choose the group cinema. | System will display all cinema of this group cinema on quick navigate. | | 2 | Click on cinema on list cinema response by previous step. | System recommend all Show Time and film filter by cinema of current day. |   **Exceptions:**  **Relationships:** Include with book ticket.  **Business Rules:** | | | |

Table 45: USE CASE - UC\_C002 <Customer> Find Schedule

##### <Customer>Choose type of seat and quality (UC\_C003)

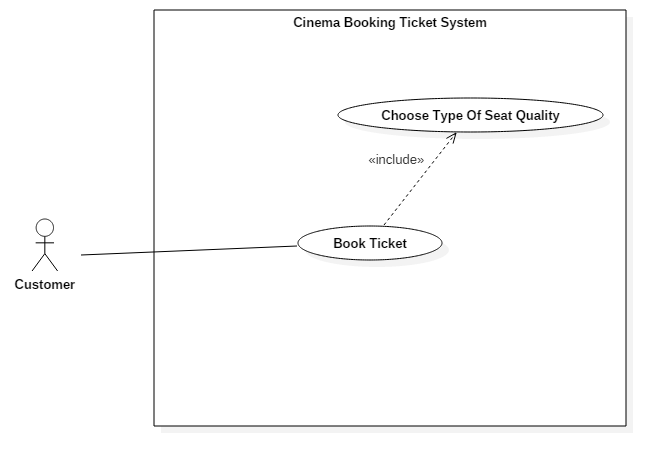


Figure 37: <Customer>Choose type of seat and quality (UC\_C003)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_C003** | | | |
| **Use Case No.** | UC\_C003 | **Use Case Version** | 2.0 |
| **Use Case Name** | Choose type of seat quality | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | normal |
| **Actor:**   * Staff   **Summary:**   * This use case helps Actor to choose the type of seat and number of ticket that they want to buy.   **Goal:**   * Actor can choose the type of seat and number of ticket that they want to buy.   **Triggers:**   * Actor sends filter command to choose ticket by click in show time of the ticket that they want to buy.   **Preconditions:**   * Actor must choose the ticket that they want to buy success.   **Post Conditions:**   * **Success:** The action is completed * **Fail:** N/A   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor click on the show time of the ticket that they want to buy | System will show the screen with all type of seat. | | 2 | Actor click on “+” / “-“ button next to type of seat that they want to buy | Increate/decrease the ticket quality with that type of seat. The ticket quality for all type of seat will limit to 10.  [Exception 1] |   **Alternative Scenario: N/A**  **Exceptions: N/A**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Number of seat is more than 10. | System shows error message: “Bạn chỉ được chọn 10 vé”. | | 2 | Number of seat is equal 0 | System will automatic disable “chọn ghế” button. So they cannot go to the next step to book ticket. |   **Relationships:** Include with book ticket.  **Business Rules:** | | | |

Table 46: USE CASE - UC\_C003 <Customer> Choose type of seat and quality

##### <Customer> Choose Seat (UC\_C004)

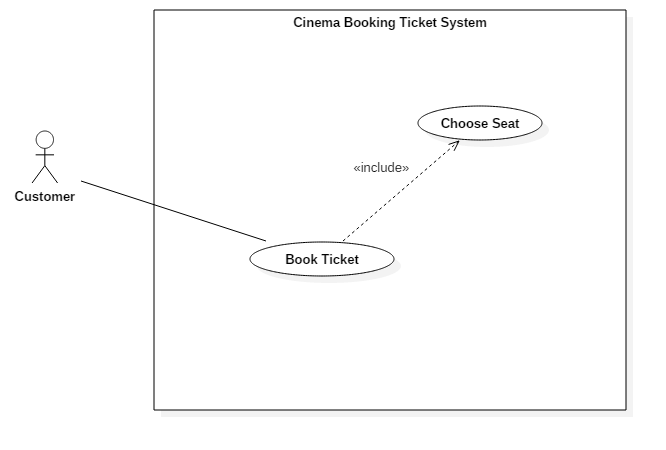


Figure 38: <Customer> Choose Seat (UC\_C004)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_C004** | | | |
| **Use Case No.** | UC\_C004 | **Use Case Version** | 2.0 |
| **Use Case Name** | Choose Seat | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | High |
| **Actor:**   * Customer   **Summary:**   * This use case helps actor can choose their seat position in the room that they will see the film.   **Goal:**   * Actor can choose their seat position in the room that they will see the film.   **Triggers:**   * Actor sends choose seat command.   **Preconditions:**   * Users must choose type of seat quality success.   **Post Conditions:**   * **Success:** The action is completed. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | After choose type of seat and ticket quality success. Actor now move to choose seat screen. | 2D room cinema will show. This screen will allow user to choose the position of seat that they want to buy. | | 2 | User click on the available seat with number of seat that they select in previous step. | System will select that seat and change it go green for user.  [Exception 1] |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | The actor cannot leave a seat empty between 2 selected seats. | System shows alert message: “Bạn không thể bỏ ghế ở giữa”. | | 2 | The actor not select seat equal with the quality that they selected in previous step. | Sytem show alert message: “Bạn chưa chọn đủ ghế”. |   **Relationships:** Include with book ticket.  **Business Rules:**N/A | | | |

Table 47: USE CASE - UC\_C004 <Customer> Choose Seat

##### <Customer> Checkout (UC\_C005)

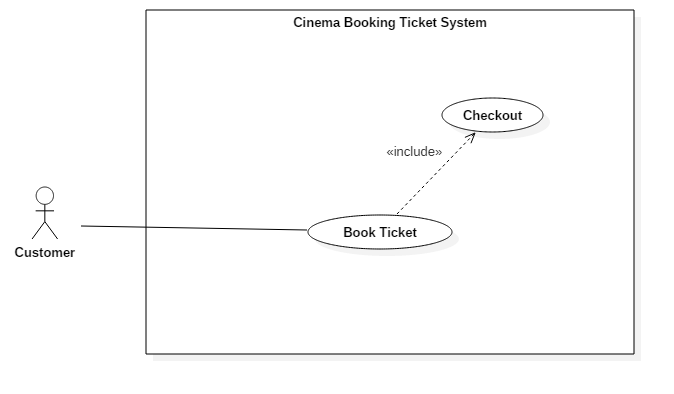


Figure 39: <Customer> Checkout (UC\_C005)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_C005** | | | |
| **Use Case No.** | UC\_C005 | **Use Case Version** | 2.0 |
| **Use Case Name** | Checkout | | |
| **Author** | DanhLC | | |
| **Date** | 20/06/2018 | **Priority** | High |
| **Actor:**   * Customer   **Summary:**   * This use case helps actor to check out the ticket that they want to buy.   **Goal:**   * Customer will own the ticket after this use case success.   **Triggers:**   * Actor sends check out command.   **Preconditions:**  **Post Conditions:**   * **Success:** The action is completed. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At choose seat screen. | System requires information from actor:   * **Phone**: free text input, required, length 9-11 digits. * **Email**: free text input, required, length 8-40 characters. | | 2 | Actor inputs information. |  | | 3 | Actor sends command to check out. | System validates information.  [Exception 1, 2] | |  | Confirm ticket dialog will open | System display confirm ticket information that user has choose. | | 4 | User click paypal button  [Alternative 1] | Paypal window will open and require user to specify their paypal’s account.  [Exception 3, 4] | | 5 |  | System create order and automatic send ticket code and order code for customer via email |   **Alternative Scenario:**  [alternative 1]   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | User click close button | Confirm dialog are close. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid Email format | System shows error message: “Email cannot be blank and must be 8-40 characters long and must match email format”. | | 2 | Invalid Phone format. | System shows error message: “Phone cannot be blank or contain alphabetical characters and must be 9-11 digits, ”. | | 3 | User has 8 minute for payment after confirm dialog are display. If time out the payment can’t complete | System shows error message: “Bạn đã hết thời gian đặt vé, vui lòng đặt vé lại ”. |   **Relationships:** include with booking ticket  **Business Rules: N/A** | | | |

Table 48: USE CASE - UC\_C005 <Customer> Checkout

##### <Customer> Resell ticket (UC\_C006)

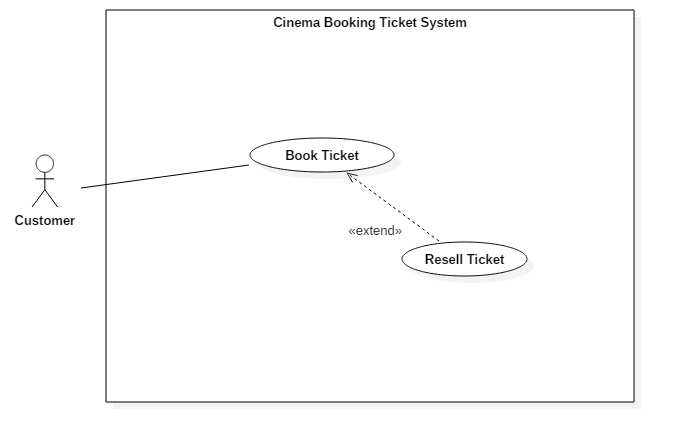


Figure 40: <Customer> Resell ticket (UC\_C006)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_C006** | | | |
| **Use Case No.** | UC\_C006 | **Use Case Version** | 2.0 |
| **Use Case Name** | Resell Ticket | | |
| **Author** | DanhLC | | |
| **Date** | 21/01/2016 | **Priority** | High |
| **Actor:**   * Staff   **Summary:**   * This use case can help customer and user can resell their ticket that they already buy. When they resell seat success. The seat will change color and information of buyer will show on that seat. The other customer wants to re-buy that seat can contact with them.   **Goal:**   * Customer can resell their ticket or the other customer.   **Triggers:**   * Customer send request to resell ticket.   **Preconditions:**  **Post Conditions:**   * **Success:** The action is completed. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | After click choose ticket. User will go to choose number of seat screen. At here click “bán lại vé/tra ve” button. | 2D room cinema will show. This screen will allow user to choose the seat that they already buy. | | 2 | They are remembered the seat that they already buy and click on it. | Drop down list will show offer you 2 options:  - return ( trả vé )  - resell (bán lại vé) | | 3 | Actor click “bán lại vé” option. | System display form request input ticket code for identity their seat. | | 3 | Customer input ticket code. Click “Xác thực”. | System validate ticket information.  [Exception 1] | | 4 |  | System display a form with following information   * Textbox: request seller input buyer email. * “Hoàn tất, chờ người mua liên hệ” button * “bán lại vé” button. | | 4 | Actor click “Hoàn tất, chờ người mua liên hệ”  [alternative 1] | System change seat status to “resell” and change current seat color to blue. Other customer can see the blue seat and information of seller when they are booking ticket. |   **Alternative Scenario:**  [alternative 1]   |  |  |  | | --- | --- | --- | | Step | Cause | System Response | | 1 | Actor click “Hoàn tất, chờ người mua liên hệ” | System validate ticket information.  [Exception 2] | |  |  | System automatic send new ticket code for the email that actor has inputted and cancel old ticket code. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid ticket code | System shows error message: “Mã vẽ bạn vừa nhập sai!”. | | 2 | Invalid email format | System shows error message: “email chưa đúng định dạng”. |   **Relationships:** extend with booking ticket  **Business Rules:**   * System will auto generate new ticket code, send it for buyer, cancel old code. | | | |

Table 49: USE CASE - UC\_C006 <Customer> Resell ticket

##### <Customer> Return ticket (UC\_C007)

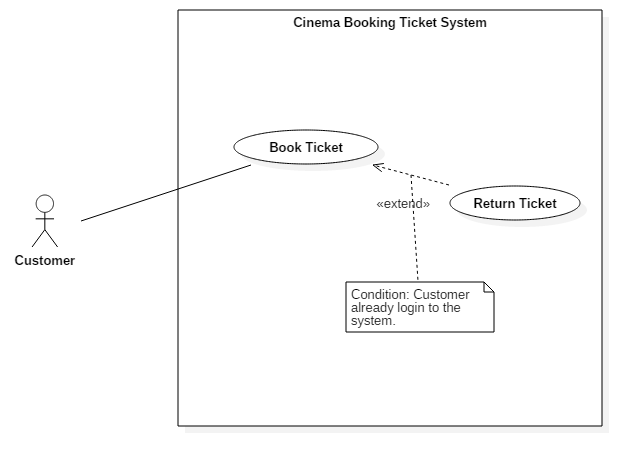


Figure 41: <Customer> Return ticket (UC\_C007)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_C007** | | | |
| **Use Case No.** | UC\_C007 | **Use Case Version** | 2.0 |
| **Use Case Name** | Return Ticket | | |
| **Author** | DanhLC | | |
| **Date** | 21/01/2016 | **Priority** | High |
| **Actor:**   * Staff   **Summary:**   * This use case can help customer and user can return ticket and receive money online. This online money can be used for ticket payment in future.   **Goal:**   * Customer can return ticket and receive money online.   **Triggers:**   * Customer send request to return ticket.   **Preconditions:**  **Post Conditions:**   * **Success:** The action is completed. * **Fail:** Show error message.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | After click choose ticket. User will go to choose number of seat screen. At here click “bán lại vé/tra ve” button. | 2D room cinema will show. This screen will allow user to choose the seat that they already buy. | | 2 | They are remembered the seat that they already buy and click on it. | Drop down list will show offer you 2 options:  - return ( trả vé )  - resell (bán lại vé) | | 3 | Actor click “trả vé” option. | System display form request input ticket code for identity their seat. | | 4 | Customer input ticket code. Click “Xác thực”. | System validate ticket information.  [Exception 1] | | 5 |  | System will change selected ticket status to available, send money online for user. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid ticket code | System shows error message: “Mã vẽ bạn vừa nhập sai!”. |   **Relationships:** extend with booking ticket  **Business Rules:**   * System will auto generate new ticket code, send it for buyer, cancel old code. | | | |

Table 50: USE CASE - UC\_C007 <Customer> Return ticket

#### <Cinema manager> Overview User Case

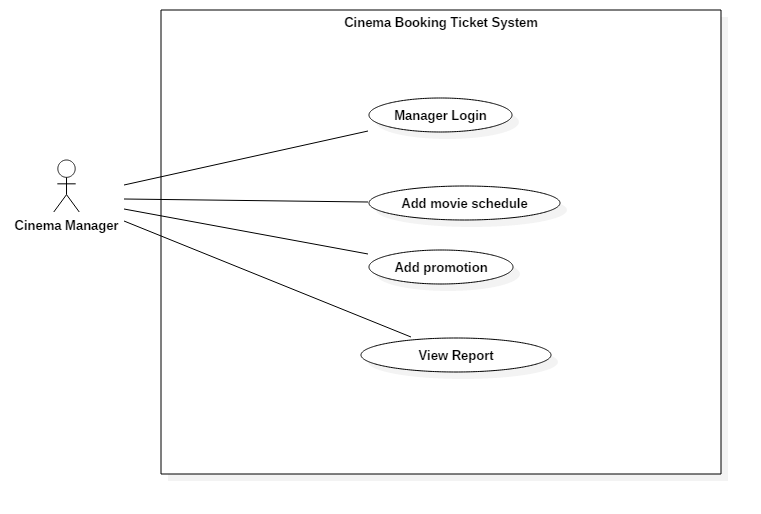


Figure 42: <Cinema Manager> Overview Use Case

##### < Cinema manager > Manager Login (UC\_M001)

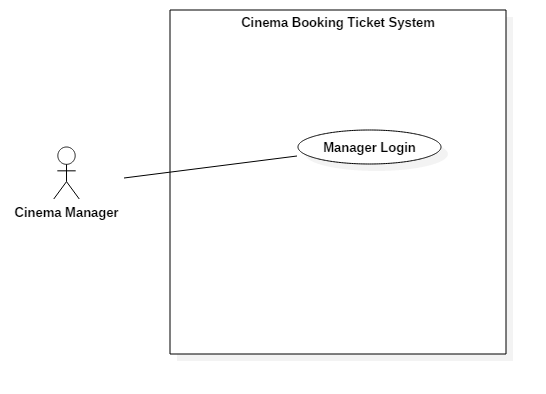


Figure 43: <Cinema manager > Manager Login (UC\_M001)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_M001** | | | |
| **Use Case No.** | UC \_M001 | **Use Case Version** | 2.0 |
| **Use Case Name** | Manger Login | | |
| **Author** | DanhLC | | |
| **Date** | 19/06/2018 | **Priority** | Normal |
| **Actor:**   * Cinema manager   **Summary:**   * This use case helps actor signs in to their account.   **Goal:**   * Actor is signed in to the system.   **Triggers:**   * Actor sends sign in command.   **Preconditions:**   * Actor’s account must be in the system.   **Post Conditions:**   * **Success:** Actor sign in successfully * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Actor goes to login view. | System requires identity information from actor: - **cinema manager username**: free text input, required, regex: /^[A-Za-z0-9]+(?:[ \_-][A-Za-z0-9]+)\*$/ - **Password**: free text input, required, length 8 - 40 | | 2 | Actor inputs information |  | | 3 | Actor sends command to sign in to the system  [Alternative 1] | System validates inputted information  [Exception 1] | | 4 |  | System signed user in |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Actor enters wrong identity information | Wrong identity information, system shows error message: “username or password is not correct”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 2 | Invalid cinema manager username format | System shows error message: “username cannot be blank and must be 8 – 40 characters long”. | | 3 | Invalid password format | System shows error message: “password cannot be blank and must be 8 – 40 characters long”. |   **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 51: USE CASE - UC\_M001 <Cinema manager > Manager Login

##### <Cinema manager > Add movie schedule (UC\_M002)

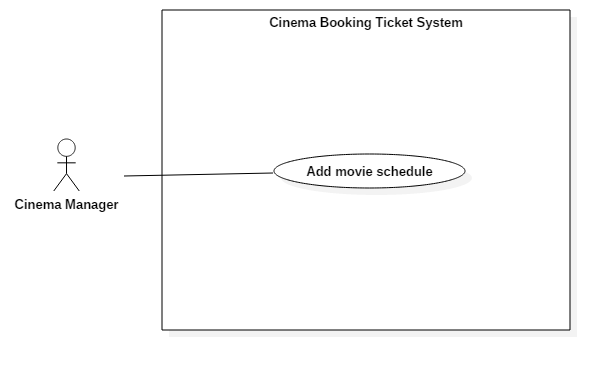


Figure 44: <Cinema manager > Add movie schedule (UC\_M002)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_M001** | | | |
| **Use Case No.** | UC\_M002 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add movie schedule | | |
| **Author** | DanhLC | | |
| **Date** | 21/06/2018 | **Priority** | Normal |
| **Actor:**   * Cinema manager   **Summary:**   * This use case helps actor to add movie schedule for their cinema. The schedule after that can update on customer view, so customer can book ticket for that schedule.   **Goal:**   * Actor can add schedule for their cinema.   **Triggers:**   * Actor sends sign in command.   **Preconditions:**   * Actor must login with cinema manager role.   **Post Conditions:**   * **Success:** Action is completed. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home page, Actor click “add schedule” button. | System request following input.   * Schedule date: date bigger or equal current date. * Select a film: select a film from dropdown list. | | 2 | Actor inputs information | System validates inputted information  [Exception 1] | | 3 | Actor click button “suggest schedule”  [Alternative 1] | System automatic suggest | | 4 | Actor click button “add” | System add schedule with given information. |   **Alternative Scenario:**  *[Alternative 1]*   |  |  |  | | --- | --- | --- | | Step. | Actor Action | System Response | | 1 | Actor manually input time and room for schedule and click add. | System show warning message “the schedule may not good when you add manual, do you want to continue”.  [Exception 1] | | 2 | Actor click “yes” | System add schedule with given information. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Date input not valid | System shows error message: “wrong date input, inputted date must equal or bigger than today”. | | 2 | Invalid schedule | System shows error message: “Can’t add schedule, duplicate time for this schedule”. |   **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 52: USE CASE - UC\_M002 <Cinema manager > Add movie schedule

##### <Cinema manager > Add promotion(UC\_M003)

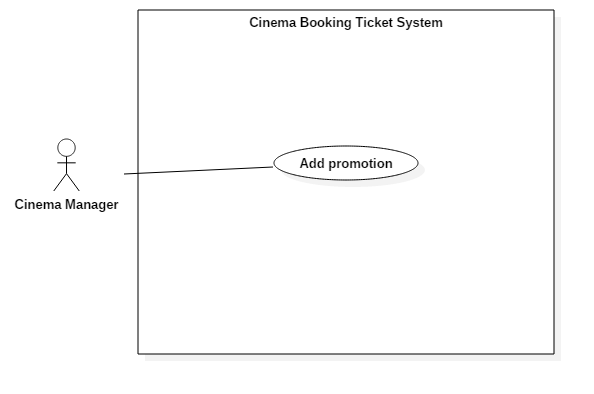


Figure 45: <Cinema manager > Add promotion (UC\_M003)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_M003** | | | |
| **Use Case No.** | UC\_M003 | **Use Case Version** | 2.0 |
| **Use Case Name** | Add promotion | | |
| **Author** | DanhLC | | |
| **Date** | 21/06/2018 | **Priority** | Normal |
| **Actor:**   * Cinema manager   **Summary:**   * This use case helps actor to add promotion news for current cinema. The promotion just applies when customer buy ticket at the cinema, not for online system. This use case is for marketing the cinema of partner.   **Goal:**   * Actor can add promotion news for current cinema.   **Triggers:**   * Actor sends add promotion command.   **Preconditions:**   * Actor must login with cinema manager role.   **Post Conditions:**   * **Success:** Action is completed. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home page, Actor click “add promotion” button | System request following input   * **Header**: free text input, required, length 8 – 40. * **Content**: free text input, required, length 50 – 1000. * **Picture**: hidden free text input, required, length 8–255 characters. * **Date:** date time input. * **Duration :** digit number. | | 2 | Actor inputs information | System validates inputted information.  [Exception 1,2,3] | | 3 | Actor click button “save” | System automatic save content to xml format and save the link to this xml file to the data base. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid header format. | System shows error message: “Address cannot be blank and must be 8-40 characters long”. | | 2 | Invalid content format | System shows error message: “Email cannot be blank and must be 8-40 characters long and must match email format”. | | 3 | Invalid picture format | System shows error message: “Picture cannot be empty and must be in jpeg/jpg/png format”. | | 4 | Invalid date | System show error message “Date cannot empty and must bigger or equal than current date”. |   **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 53: USE CASE - UC\_M003 <Cinema manager > Add promotion

##### <Cinema manager > View report (UC\_M004)

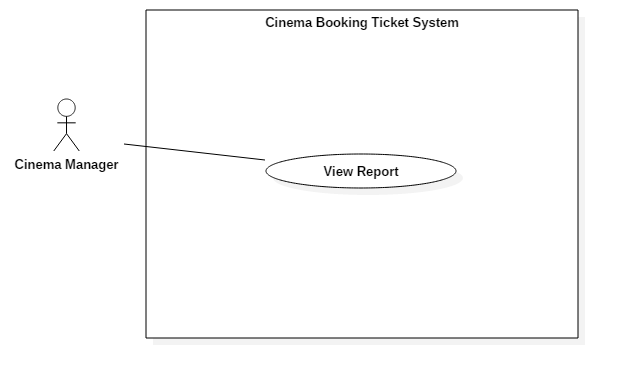


Figure 46: <Cinema manager > View report (UC\_M004)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_M004** | | | |
| **Use Case No.** | UC\_M004 | **Use Case Version** | 2.0 |
| **Use Case Name** | View report | | |
| **Author** | DanhLC | | |
| **Date** | 21/06/2018 | **Priority** | Normal |
| **Actor:**   * Cinema manager   **Summary:**   * This use case helps actor view report filter by date, film, room. System will statistic by number of ticket sold, revenue, number of ticket return.   **Goal:**   * Actor can view their cinema report.   **Triggers:**   * Actor sends view report command.   **Preconditions:**   * Actor must login with cinema manager role.   **Post Conditions:**   * **Success:** Action is completed. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | At home page, Actor click “view report” button | System show filter form request following input   * **Film**: select from drop down list * **Room**: Select from drop down list. * **Show Time**: Select from drop down list. * **From date:** date input. * **To date:** date input. * **Report by revenue:** check box. * **Report by ticket:** check box. * **Report by return ticket:** check box. | | 2 | Actor inputs information | System validates inputted information.  [Exception 1,2,3] | | 3 | Actor click button “save” | System automatic save content to xml format and save the link to this xml file to the data base. |   **Alternative Scenario: N/A**  **Exceptions:**   |  |  |  | | --- | --- | --- | | No. | Cause | System Response | | 1 | Invalid from date format. | System shows error message: “From date cannot null and must bigger or equal current date”. | | 2 | Invalid “to date” format | System shows error message: “To date cannot null and must bigger than from date”. | | 3 | You must be select at least one check box. | System shows error message: “You must be select at least one check box.”. | | 4 | Invalid date | System show error message “Date cannot empty and must bigger or equal than current date”. |   **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 54: USE CASE - UC\_M004 <Cinema manager > View report

#### Initial <System> Overview Use Case

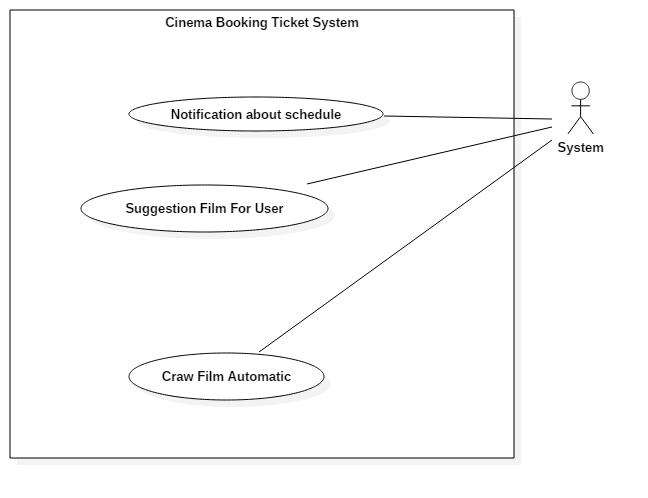


Figure 47: Initial <System> Overview Use Case

##### <System> Craw film automatic (UC\_S001)

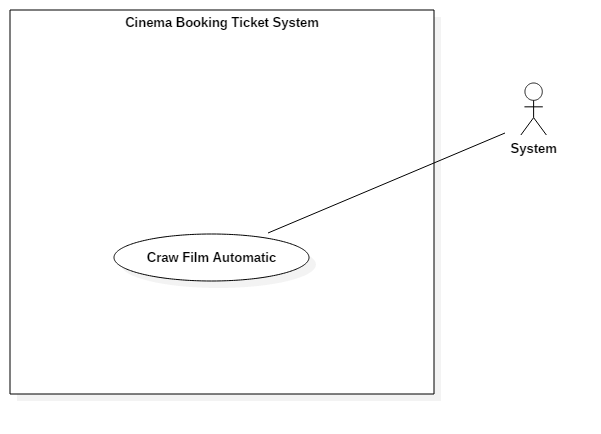


Figure 48: <System> Craw film automatic (UC\_S001)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_S001** | | | |
| **Use Case No.** | UC\_S001 | **Use Case Version** | 2.0 |
| **Use Case Name** | Craw film automatic | | |
| **Author** | DanhLC | | |
| **Date** | 21/06/2018 | **Priority** | Normal |
| **Actor:**   * System   **Summary:**   * System will automatic craw film data and add it to database.   **Goal:**   * New film will automatic add to database.   **Triggers:**   * System run success.   **Preconditions:**   * Actor must be logged in into system.   **Post Conditions:**   * **Success:** Action complete. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | System deploy | System run initial function for craw data automatically on web. |   **Alternative Scenario: N/A**  **Exceptions: N/A**  **Relationships:** N/A  **Business Rules:** N/A. | | | |

Table 55: USE CASE - UC\_S001 <System> Craw film automatic

##### <System> Notification about schedule (UC\_S002)

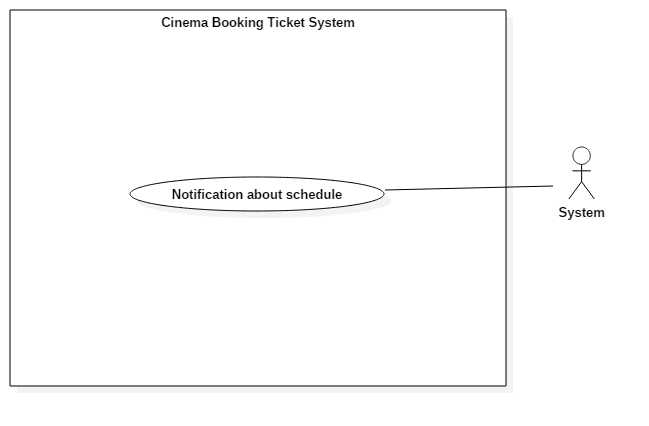


Figure 49: <System> Notification about schedule (UC\_S002)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_S002** | | | |
| **Use Case No.** | UC\_S002 | **Use Case Version** | 2.0 |
| **Use Case Name** | Notification about Schedule | | |
| **Author** | DanhLC | | |
| **Date** | 21/06/2018 | **Priority** | Normal |
| **Actor:**   * System   **Summary:**   * System will automatic notification for mobile user before the film show time 30 minutes. The purpose is reminded them about the ticket that they have bought.   **Goal:**   * Notification for user about schedule.   **Triggers:**   * System run success.   **Preconditions:**   * System deployed success.   **Post Conditions:**   * **Success:** Action complete. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | System deploy | System run initial function notification for mobile user. |   **Alternative Scenario: N/A**  **Exceptions: N/A**  **Relationships:** N/A  **Business Rules:** N/A. | | | |

Table 56: USE CASE - UC\_S002 <System> Notification about schedule

##### <System> suggest film for user (UC\_S003)

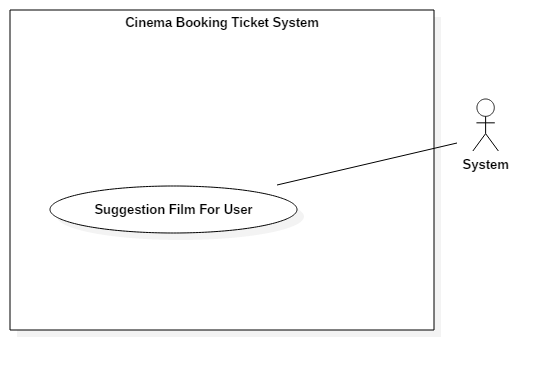


Figure 50: <System> suggest film for user (UC\_S003)

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_S003** | | | |
| **Use Case No.** | UC\_S003 | **Use Case Version** | 2.0 |
| **Use Case Name** | Suggest film for user | | |
| **Author** | DanhLC | | |
| **Date** | 21/06/2018 | **Priority** | Normal |
| **Actor:**   * System   **Summary:**   * System will automatic suggest film for user. When they go to home page list 4 films suitable with their favorite will show on top. Each week we will send mail to suggest film for them.   **Goal:**   * System automatic suggest film for user base on their favorite.   **Triggers:**   * System run success.   **Preconditions:**   * System deploy success.   **Post Conditions:**   * **Success:** Action complete. * **Fail:** Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | System deploy | System run initial function for suggest film for user |   **Alternative Scenario: N/A.**  **Exceptions: N/A.**  **Relationships:** N/A.  **Business Rule: N/A.** | | | |

Table 57: USE CASE - UC\_S003 <System> suggest film for user

## Software Requirement Specification

* 1. **Usability**
* Admin interface is written in English including all dialogs and messages.
* Partner, cinema manager and user interface are written in Vietnamese including all dialogs and messages.
* The system needs no more than 2 days of training for cinema manager.
  1. **Reliability**
* The system can allow ticket buyers to resell or return their ticket 4 hour prior to the movie show time in case they cannot go to the cinema.
* QR code is saved in cell phone so even without internet customer can use QR code to get into the cinema room.
  1. **Availability**
* Server has back-up method to protect and restore necessary data as soon as possible whenever there is a problem.
  1. **Security**.
* Each role of user has a specific permission to interact with the system.
* System always checks for authorization and authentication before doing anything.
* Only Admin can create partner account, only partner can create cinema account.
  1. **Maintainability**
* The system is divided into separated modules.
  1. **Portability**
* User can use the mobile application on devices running Android 4 or later.
* Web application can be run on Chrome browser version 42 or later.
  1. **Performance**
* Client System run fast and returns results in less than 2 seconds under 4Mbps bandwidth.
* Admin System run fast and returns results in less than 2 seconds under 4Mbps bandwidth.

## Conceptual Diagram

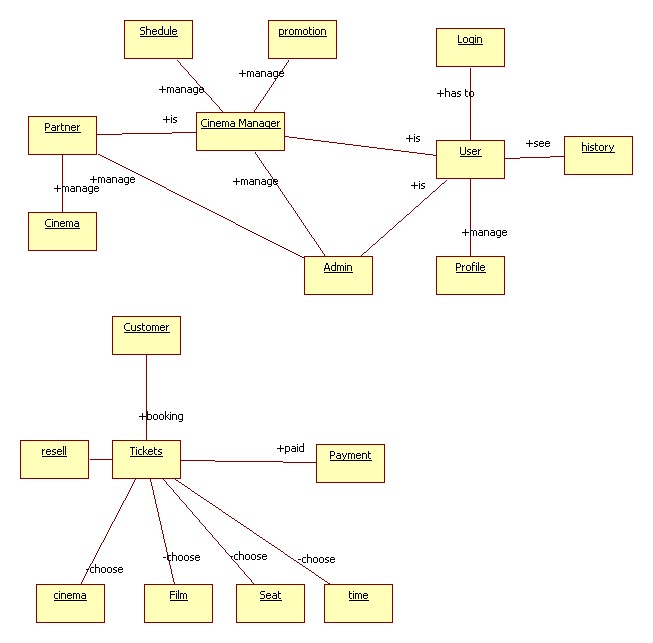


Figure 51: Conceptual Diagram

|  |  |  |
| --- | --- | --- |
| **Entity Name** | **Description** | |
| Admin | Person who manages the system. | |
| Cinema Manager | Person who manages the schedule and promotion of the cinema. | |
| Partner | Person who managers the cinema in their system | |
| User | Person who manages their profile and see their booking history | |
| Customer | Person who booking tickets. | |
| Payment | Contains all payment methods. | |
| Resell | Contains all information’s ticket return. | |
| Tickets | Contains information that customer booking about cinema, film, seat and time | |
| Schedule | Contains all information’s schedule | |
| Promotion | Contains all promotion of cinema. | |
| Profile | Contains all information’s user | |
| History | Contains a list history pay tickets. | |
|  | |

Table 58: Conceptual Diagram Data Dictionary

# Software Design Description

## Design Overview

* The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.
* The detailed design describes static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.
* The database design describes the relationships between entities and details of each entity.
* Document overview:
* Section 1: gives an overall description of the system architecture design.
* Section 2: gives component diagrams that describe the connection and integration of the system.
* Section 3: gives the detail design description, which includes class diagram, class explanation, and sequence diagram to details the application functions.
* Section 4: gives the interface design description, which includes component interface, web application interface and mobile application design.
* Section 5: describe a fully attributed Entity Relationship Diagram.
* Section 6: describe the algorithms that apply to the system.

## System Architectural Design

### Web Application Architecture Description

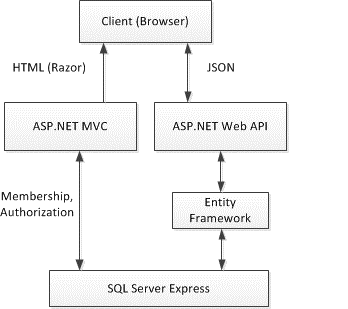


Figure 52: Web Application Architecture

We choose this architecture for a couple of reasons:

* To maximize the render performance of web client. Our application may need most of the time to do training
* Easy to develop, this architecture separates frontend out of backend. It helps developers can work independently with each other
* The architecture divides the application into components. This increases reusability and testability
* Reduce side effects in application

### Web Service Application Architecture Description

Backend Architecture

*Figure 53: Backend Application Architecture*

The Web Service Application Architecture will base on MVC architecture. We choose this architect because of following advantages:

* The application will be divided into three clear part
* Testing each component in the architecture become easier
* The existing of Model or Business Layer leverage testability and decoupling to the View
* Active Record is a pattern to abstract the data layer and manipulate the database; it also acts as an ORM. It combines both data and domain logic into object so that user can easily know how to update the data

### Mobile Application Architecture Description

Mobile Architecture

*Figure 54: Mobile Application Architecture*

Mobile application architecture will use MVVM. There is couple of reasons choosing this architect:

* It minimizes the number of dependency, helps increase loose coupling between application
* The View is passive and will be notified to update data
* The View Model will contain complex view logic; it also has no information about View. This help increase reusability of View Model
* The Model is where business logic sit in; it also performs API requests to backend application

Develop Mobile Application using MVVM have increase productivity and reduce side effects

## Component Diagram

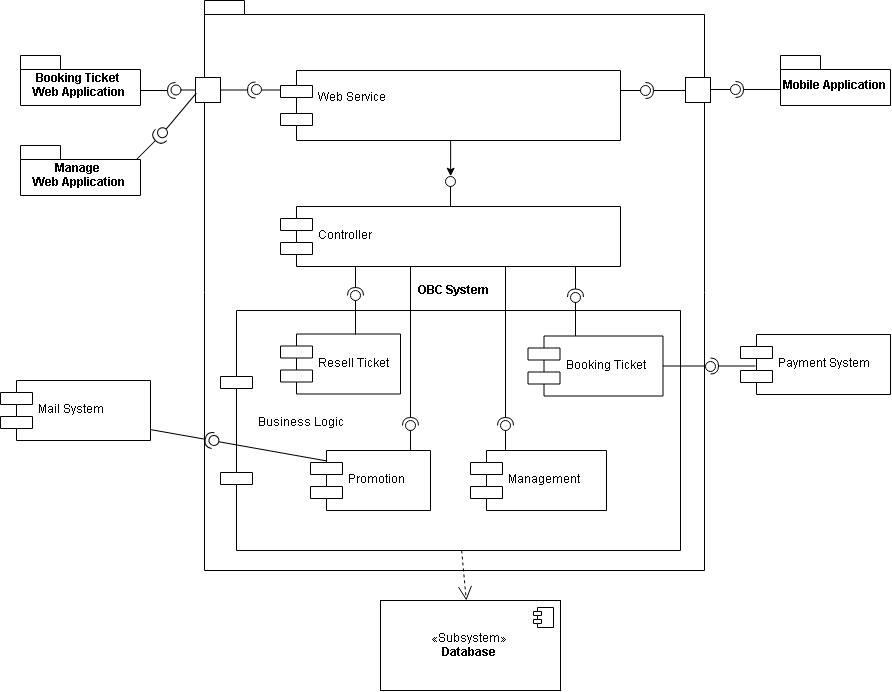


Figure 55: Component Diagram

|  |  |
| --- | --- |
| **Name** | **Description** |
| OBS System | Online cinema booking system |
| Booking Ticket Web Application | Package application to use system |
| Manage Web Application | Package application to manage system |
| Mobile Application | Package application for mobile device to use system |
| Web Service | Component is used to provide API for mobile application to interact with the system. |
| Business Logic | Common services are used to handle system’s business operations |
| Controller | Component to handle HTTP request |
| Management | Component to manage system data |
| Resell Ticket | Component to return ticket by resell |
| Booking Ticket | Component to choosing ticket |
| Promotion | Component to support customer about promotion |
| Payment System | External component for handle purchasing |
| Mail System | External component for message to send email |
| Active Record | Abstract data layer to interact with database system |
| Database System | Where application data is manipulated and stored |
|  | |

Table 59: Component Dictionary

## Detailed Description



### Class Diagram

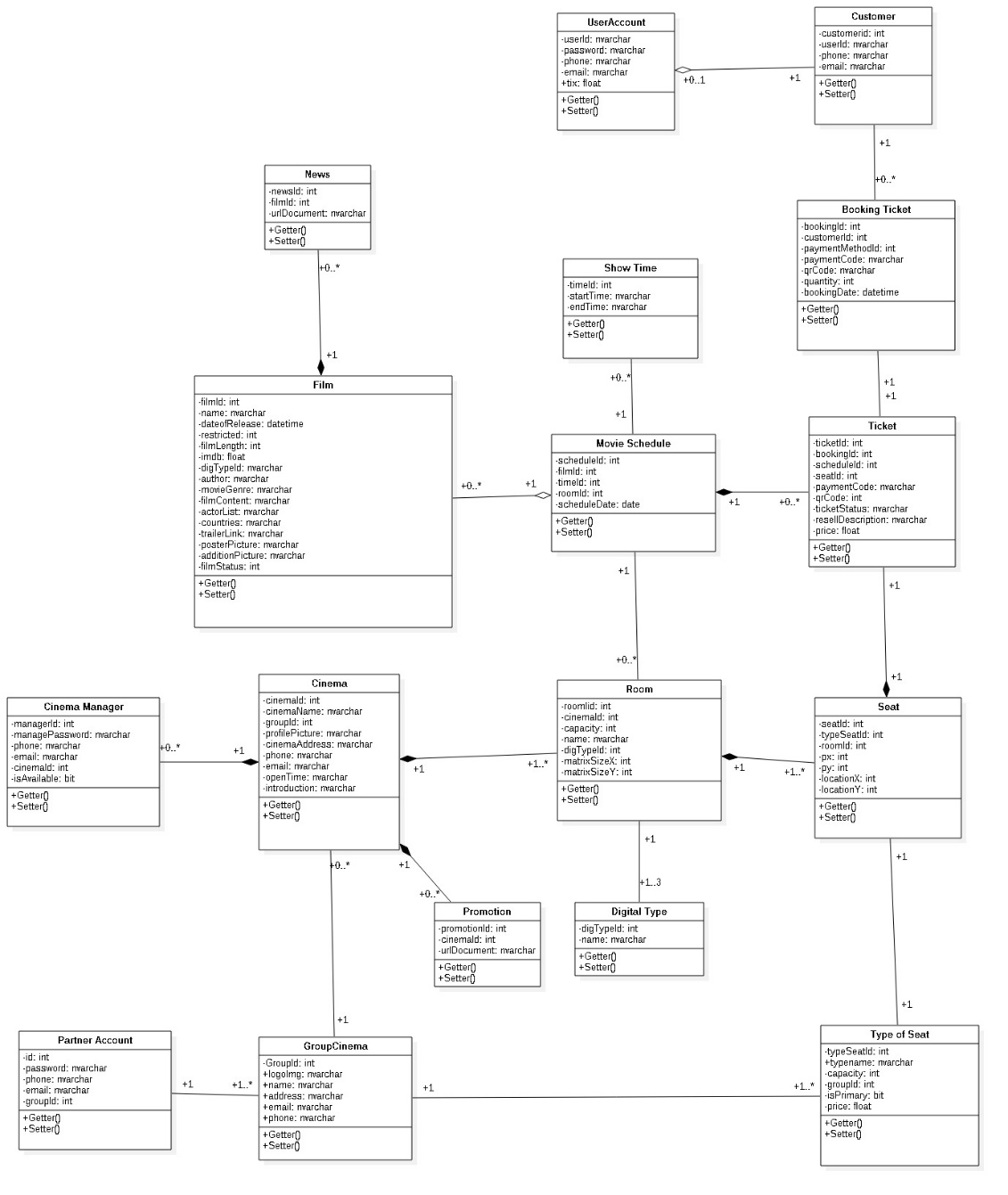


Figure 56: Class Diagram

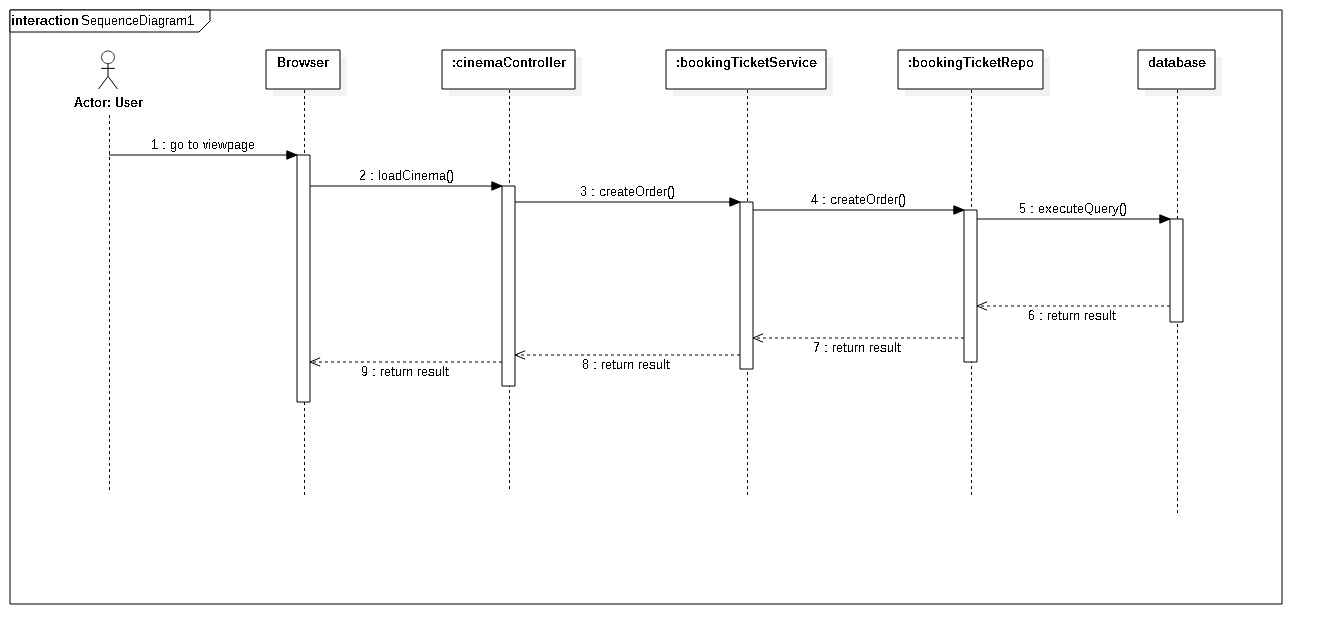
### Class Diagram Explanation

|  |  |  |
| --- | --- | --- |
| **Class name** | **Mapping column with conceptual diagram** | **Description** |
| User Entity | User Account | Contains the information of users in the system. |
| Admin Entity | Admin Account | Contains the information of admin in the system. |
| Customer Entity | Customer | Contains the information of customers in the system. |
| Partner Entity | Partner Account | Contains the information of partners in the system. |
| Cinema Manager  Entity | Cinema Manager | Contains the information of cinema managers in the system. |
| Film Entity | Film | Contains the detailed information of all films in the systems. |
| Movie schedule Entity | Movie schedule | Contains the information of all movie schedules in the system. |
| Show Time Entity | Show Time | Contains the detailed information of all film show time in the systems. |
| Room Entity | Room | Contains the detailed information of all room in the systems. |
| Digital Type Entity | Digital Type | Contains the information of all the type of digital in the systems. |
| Seat Entity | Seat | Contains the information of all seats in the systems. |
| Type of Seat Entity | Type of Seat | Contains the information of all the types of seat in the systems. |
| Group Cinema Entity | Group Cinema | Contains the information of all the groups of cinema in the systems. |
| Cinema Entity | Cinema | Contains the information of all the cinemas in the systems. |
| Promotion Entity | Promotion | Contains the information of all the promotions in the systems. |
| Ticket Entity | Ticket | Contains the information of all the tickets in the systems. |
| Booking Ticket Entity | Booking Ticket | Contains the information of all the transactions when booking ticket. |
| News Entity | News | Contains the news information of all the film |
|  | | |

### Interaction Diagram

#### Sequence Diagram

##### <User>Booking Ticket

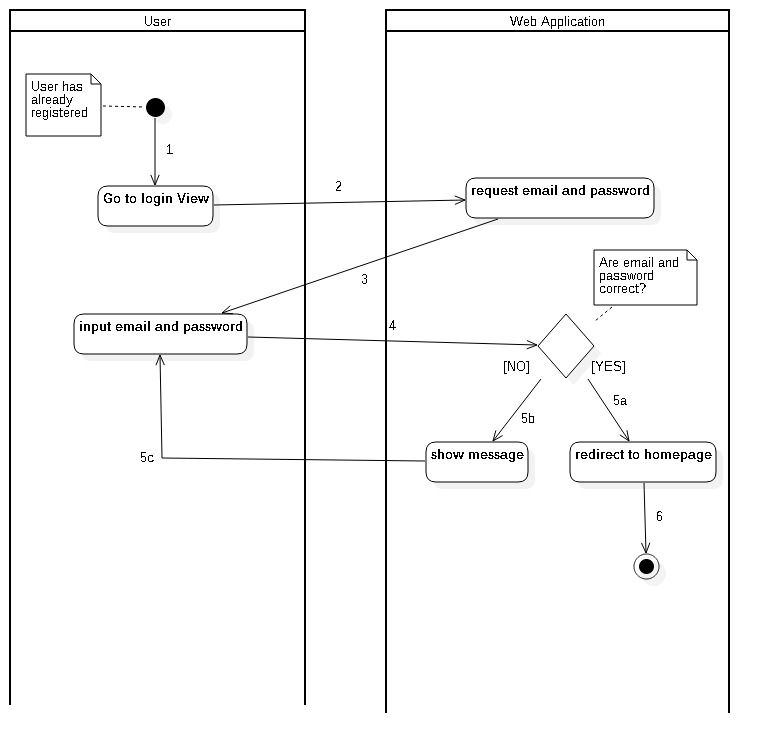


##### <Staff> Add time

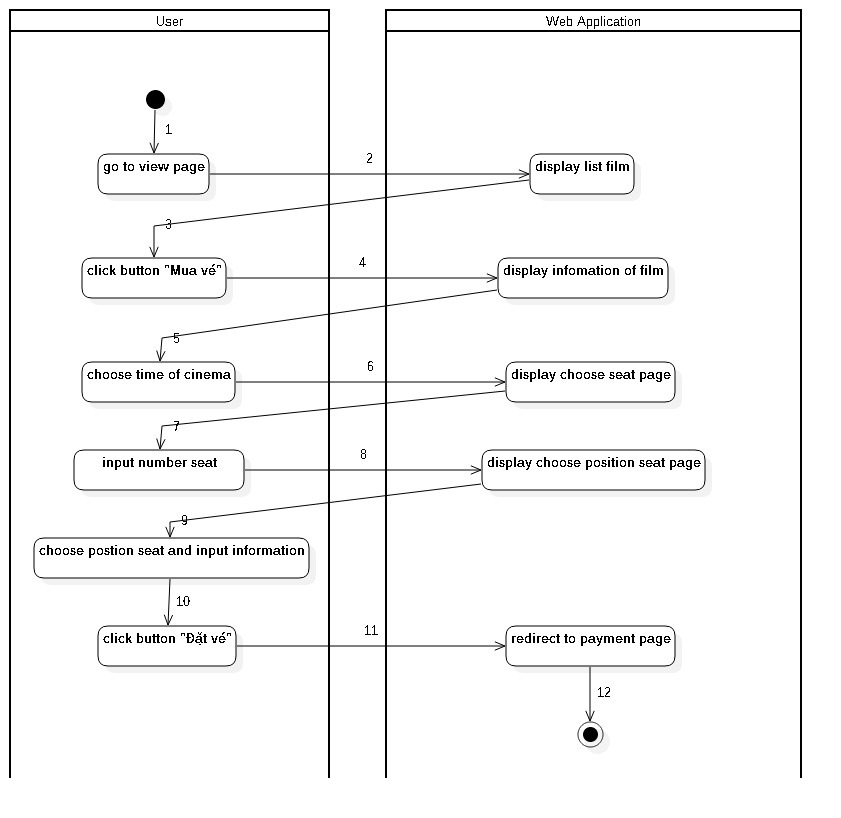
##### <Staff> Add promotion

#### Activity Diagram

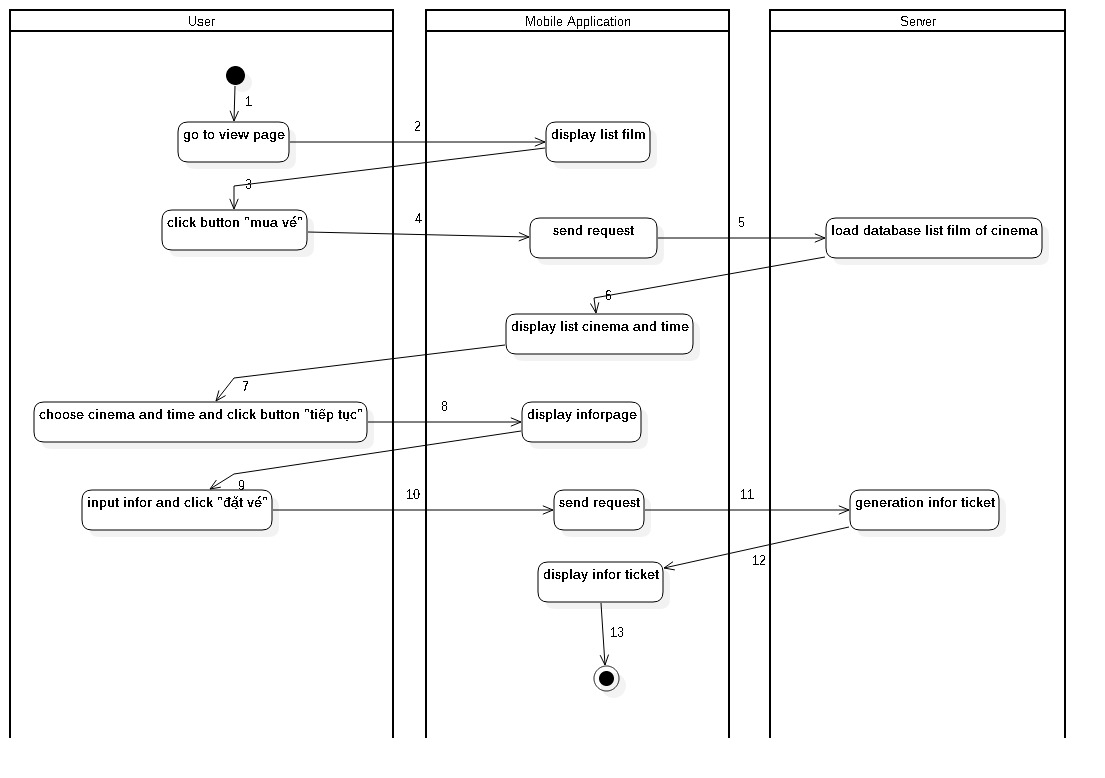
##### Login



##### Booking Ticket Web Application



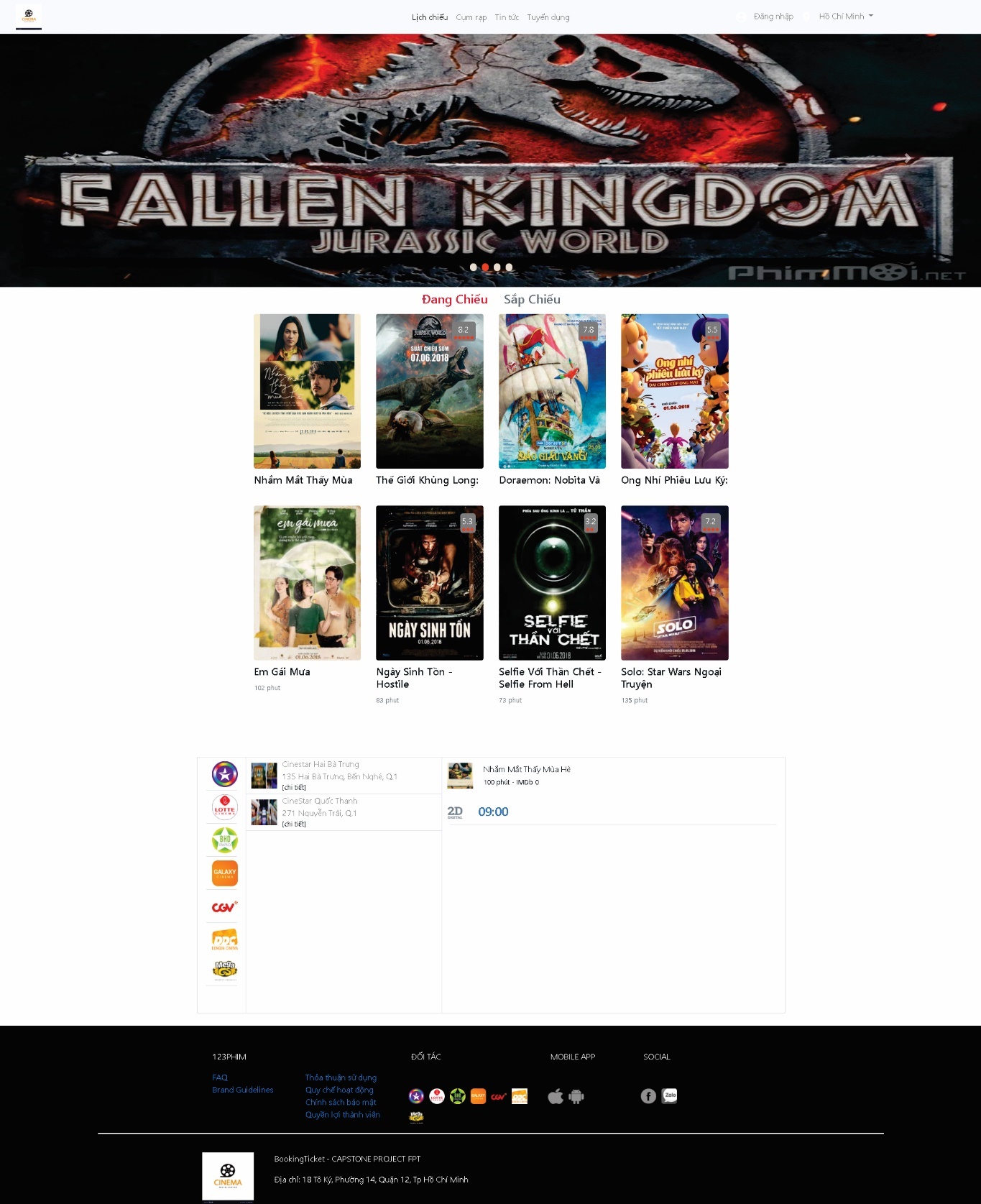
##### Booking Ticket Mobile Application



## Interface

### Web application Design

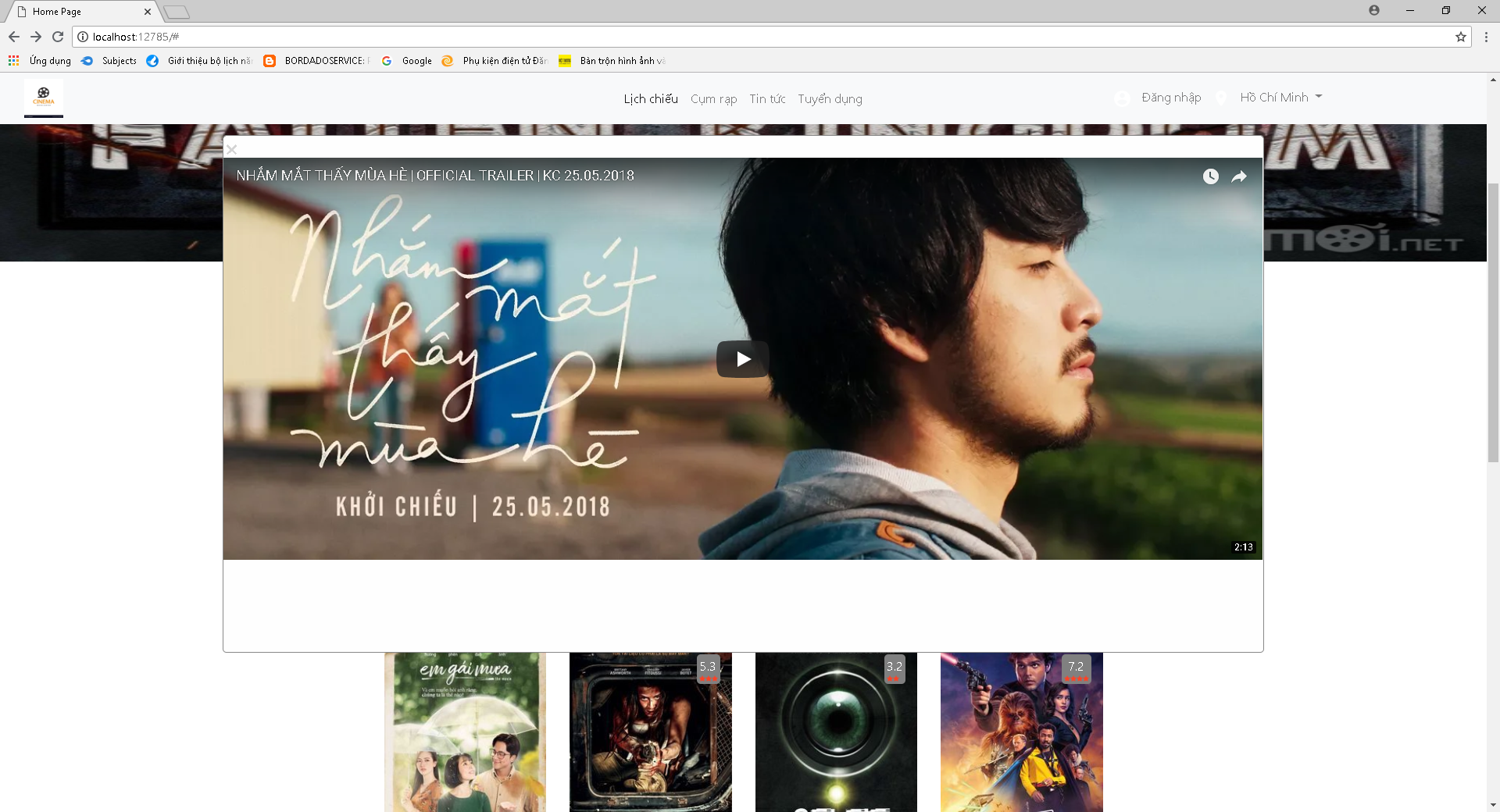
#### <Unauthorized User> Home Page



**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Đăng nhập | Log-in to the system | N/A | Transfer to specific page with specific account |
| 2 | Đang Chiếu | Show all the movies being shown on the system | N/A | Show all movies |
| 3 | Sắp Chiếu | Show all the movies will showed on the system | N/A | Show all movies |
| 4 | Image | Click on image’s movies will allow see trailer’s movies | N/A | Show window contain trailer’s movies. |
| 5 | Logo cinema | Click on image’ logo cinema will show movies schedule | N/A | Show movies schedule |

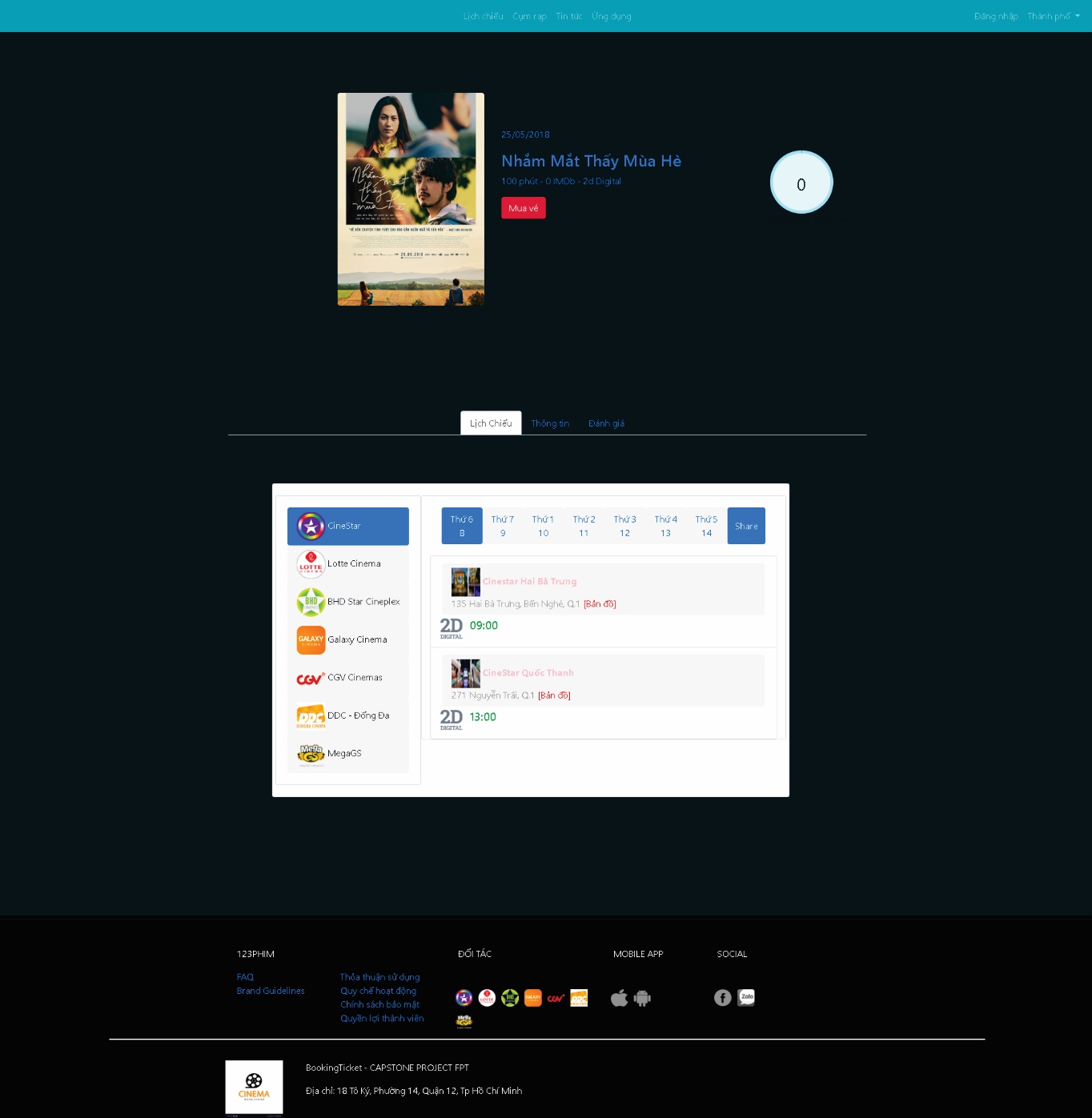
##### Trailer



**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Play | Click play button to run the trailer | N/A | The trailer is run |

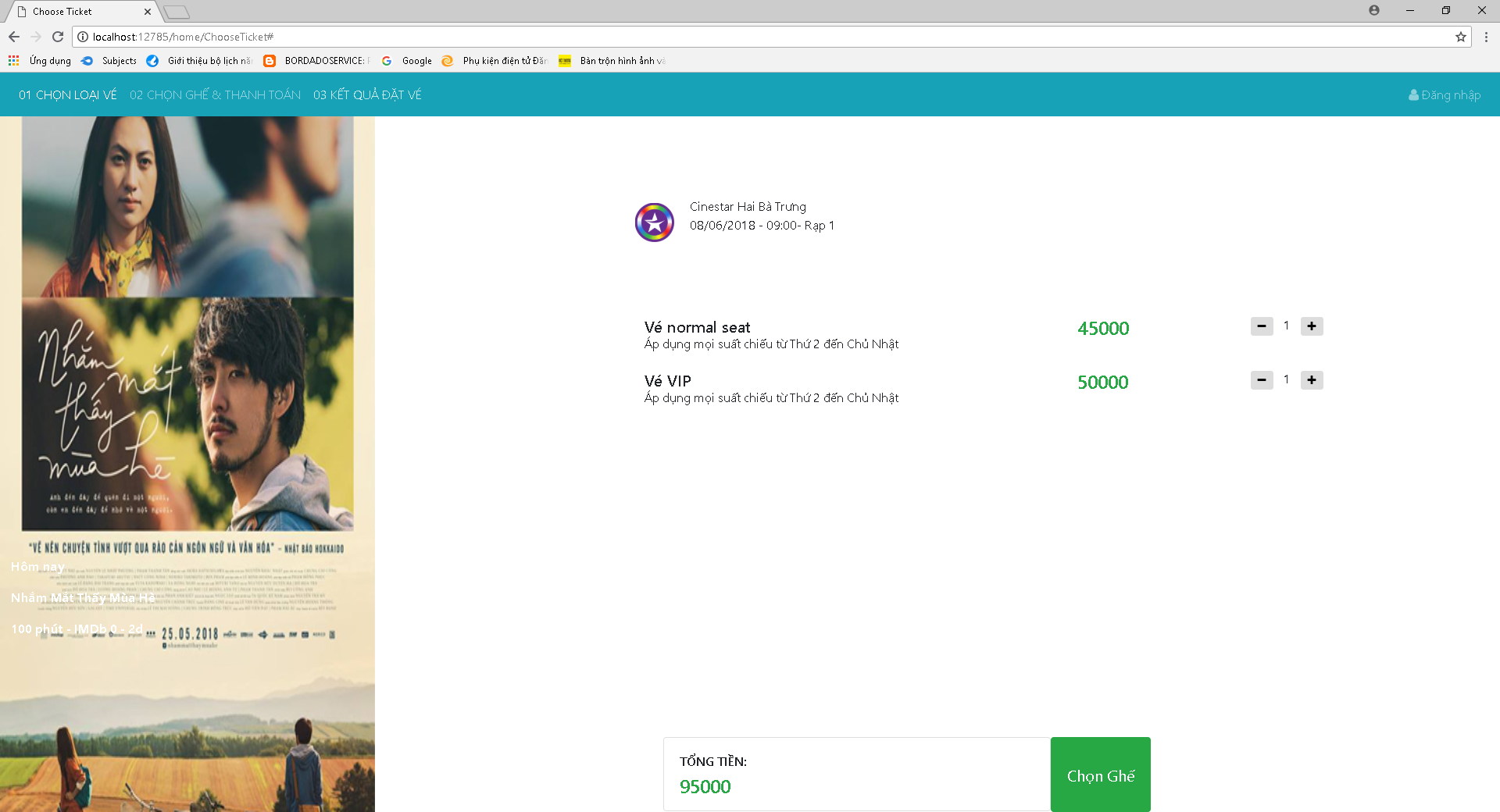
#### <Unauthorized User> Film Detail Page



**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Mua vé | N/A | N/A | N/A |
| 2 | Lịch chiếu | Click choose hyperlink “Lịch chiếu” will show all the movies being shown on the movies schedule follow cinema | N/A | Show all movies follow cinema |
| 3 | Thông tin | Click choose hyperlink “Thông tin” will Show information’s the movies | N/A | Show information’s the movies |
| 4 | Đánh giá | Click choose hyperlink “Đánh giá” will show feedback of customer follow this movies | N/A | Show list feedback of customer |
| 5 | Logo cinema | Click on image’ logo cinema will show movies schedule | N/A | Show movies schedule |
| 6 | 09:00 | Click on time in film detail will redirect the page to ChooseTicket | N/A | Show ChooseTicket page |

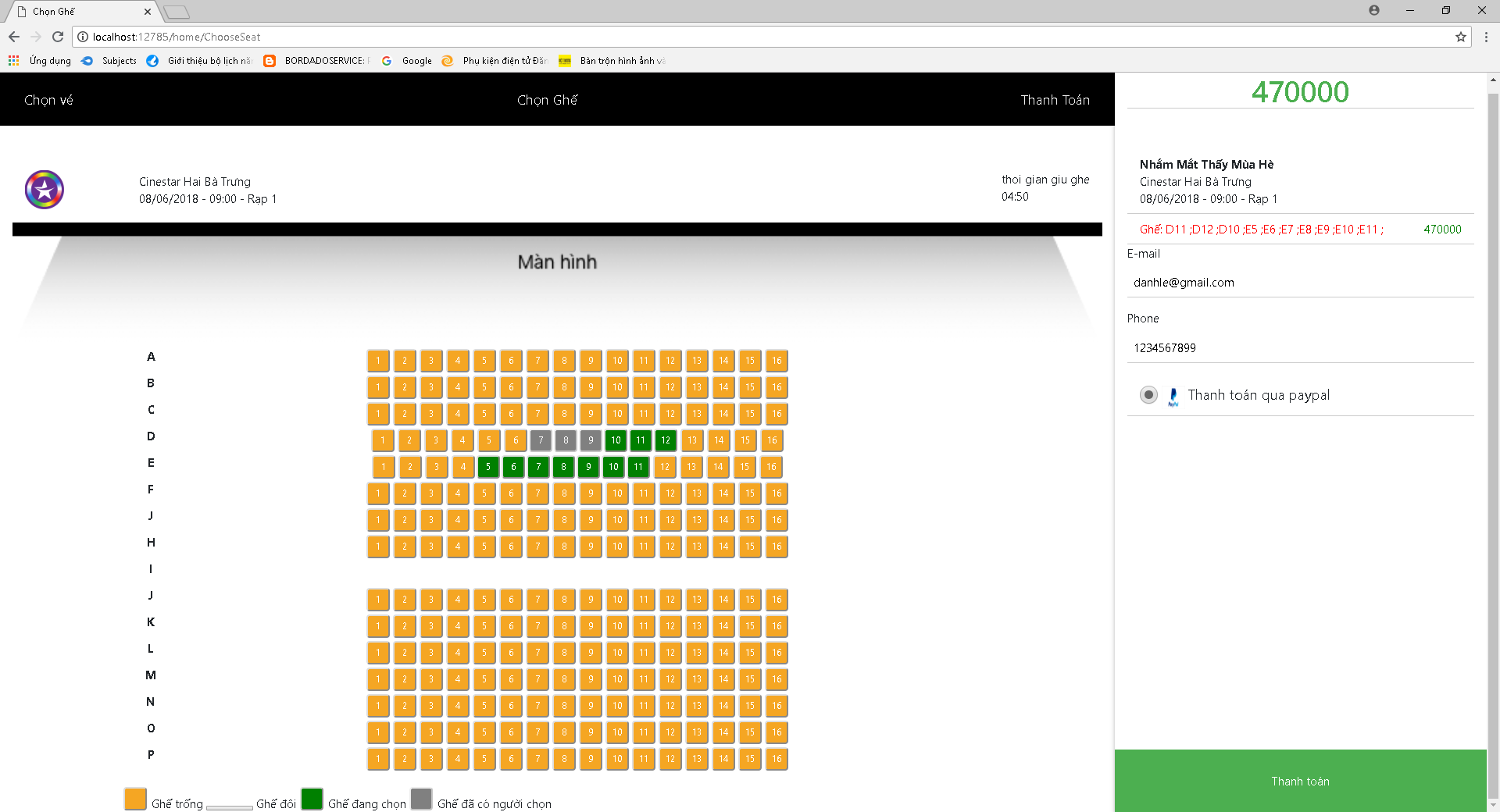
#### <Unauthorized User> ChooseTicket Page



**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | - | Decreases seats | N/A | Decreases 1 or more seats |
| 2 | + | Increase seats | N/A | Increases 1 or more seats |
| 3 | Chọn ghế | Complete process choose ticket | N/A | Show ChooseSeat page |

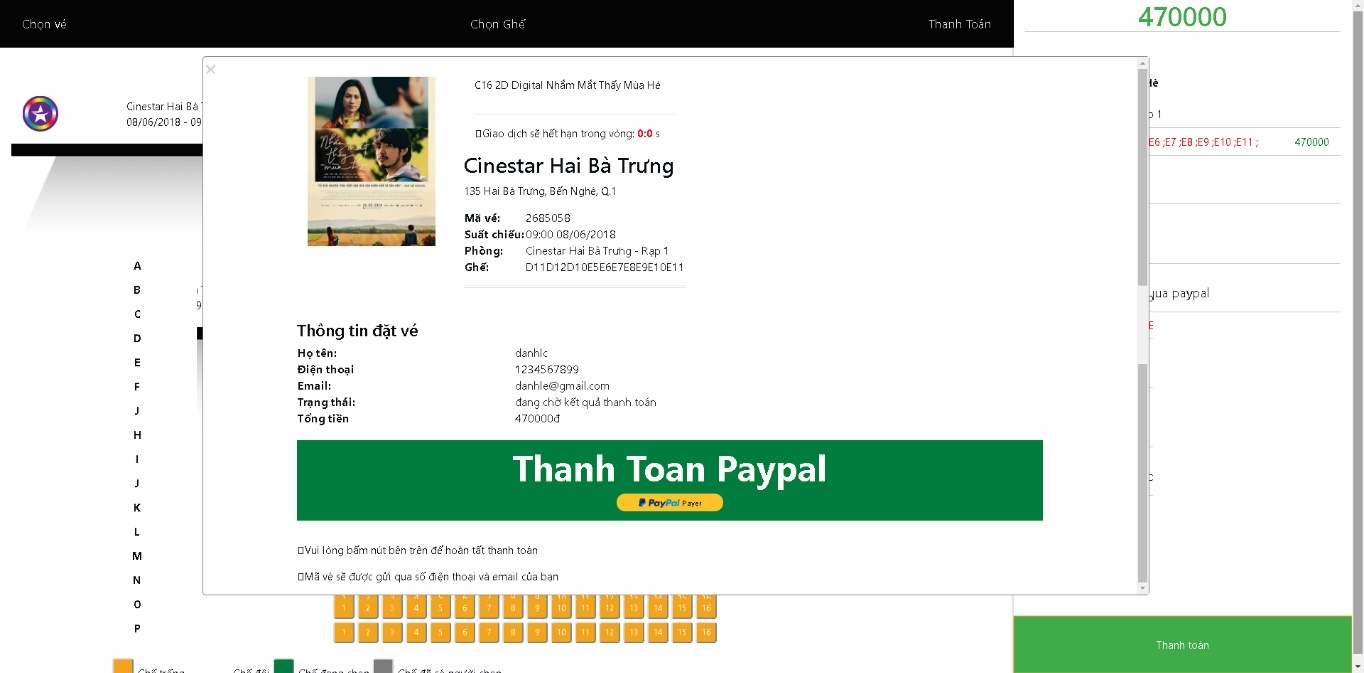
#### <Unauthorized User> ChooseSeat page



**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Seats | Allow choose seats allow number of ticket | Không thể để trống ghế ở giữa | Displays the selected seat and turns green. |
| 2 | Thanh toán | Complete process choose seat | N/A | Show notification about ticket |

##### Paypal notification



**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Thanh Toan Paypal | Complete process booking ticket and redirect to payment channel | N/A | redirect to payment channel page. |
| 2 | X | Button cancel, click this button will be returned ChooseSeat page |  | Show ChooseSeat page |

#### <Unauthorized Admin> Home Page

### Mobile application Design

#### <Unauthorized User> Home Page Android

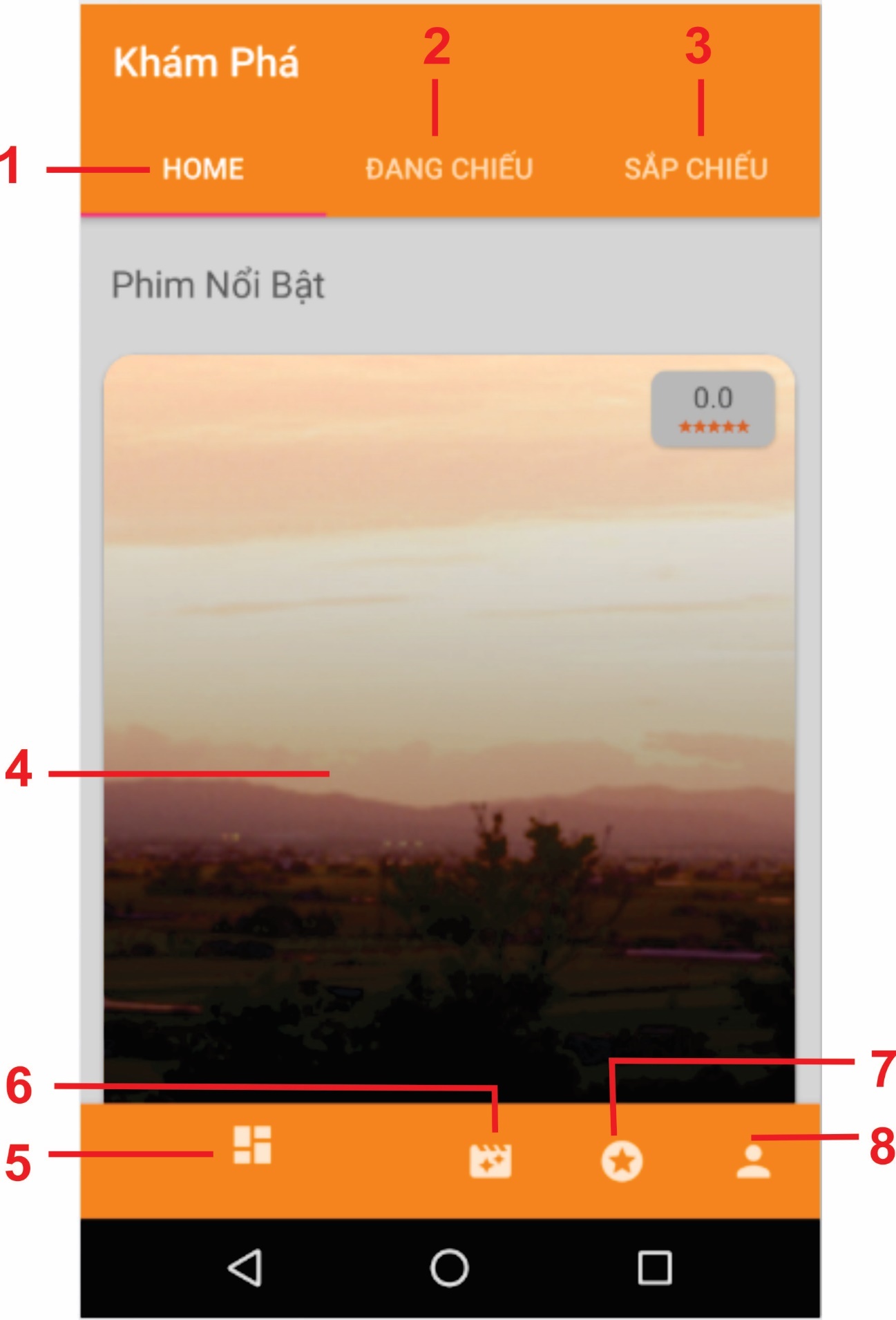


Figure 57: <Unauthorized User> Home Page Android

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Home | Show all information of application | N/A | Show all information of application |
| 2 | Đang Chiếu | Show all the movies being shown on the system | N/A | Show all movies |
| 3 | Sắp Chiếu | Show all the movies will showed on the system | N/A | Show all movies |
| 4 | Image’s movies | Click on “Đặt vé” button or image’s cinema will show movies schedule | N/A | Show movies schedule |
| 5 | Button | Click on button simple will show home page | N/A | Show home page |
| 6 | Button | Click on button simple will show cinema page | N/A | Show cinema page |
| 7 | Button | Click on button simple will show news page | N/A | Show news page |
| 8 | Button | Click on button simple will show login page | N/A | Show login page |

Table 97: Define <Unauthorized User> Home Page Android

#### <Unauthorized User> FilmDetail Page Android

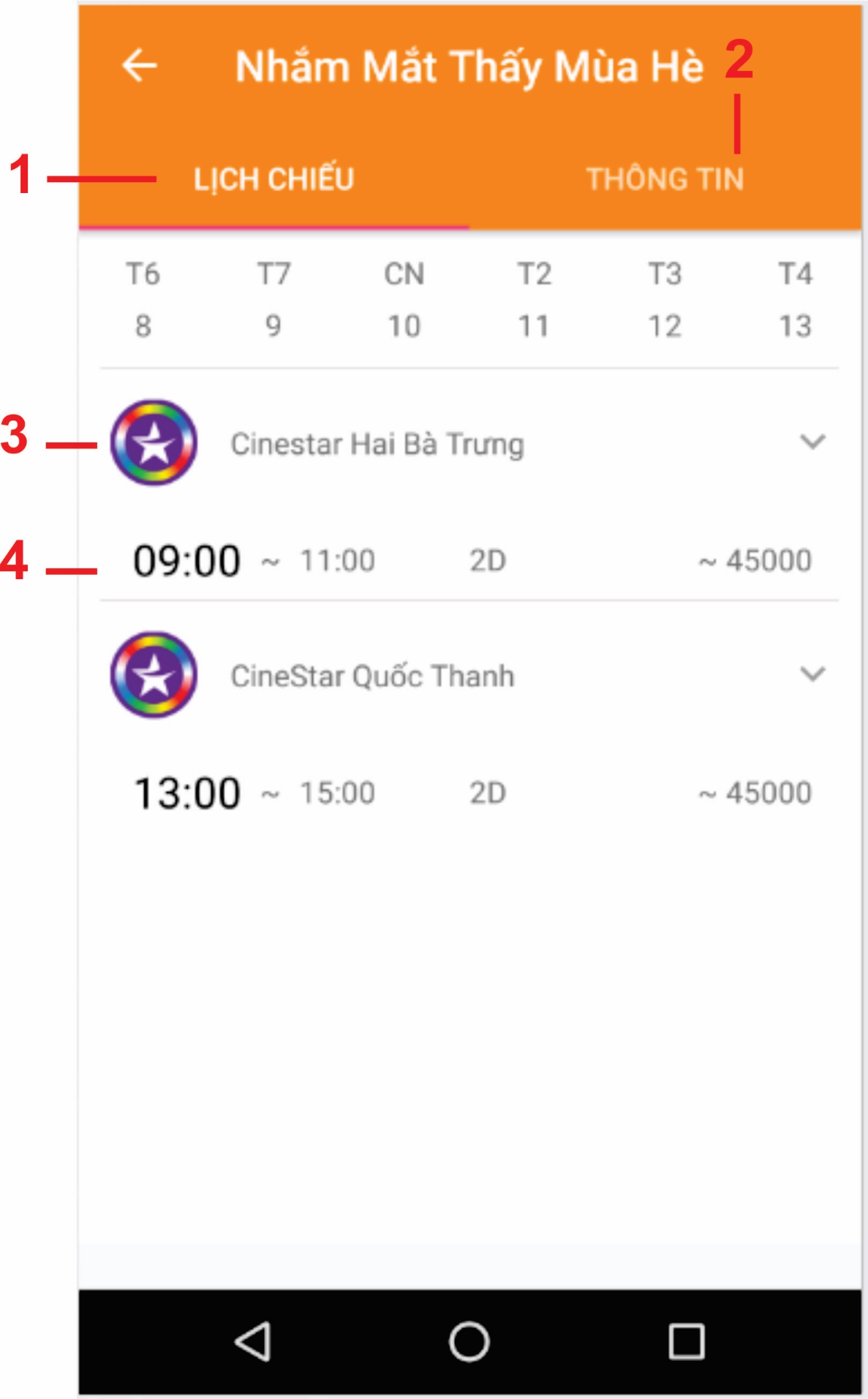


Figure 58: <Unauthorized User> FilmDetail Page Android

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Lịch chiếu | Click choose hyperlink “Lịch chiếu” will show all the movies being shown on the movies schedule follow cinema | N/A | Show all movies follow cinema |
| 2 | Thông tin | Click choose hyperlink “Thông tin” will Show information’s the movies | N/A | Redirect to “Thông tin” page |
| 3 | Name cinema | Click choose name cinema will show time’s information | N/A | Show list time’s information |
| 4 | Information of a movie | Click on a row container movie’s information will show chooseTicket page | N/A | Show ChooseTicket page |

Table 98: Define <Unauthorized User> FilmDetail Page Android

#### <Unauthorized User> ChooseTicket Page Android

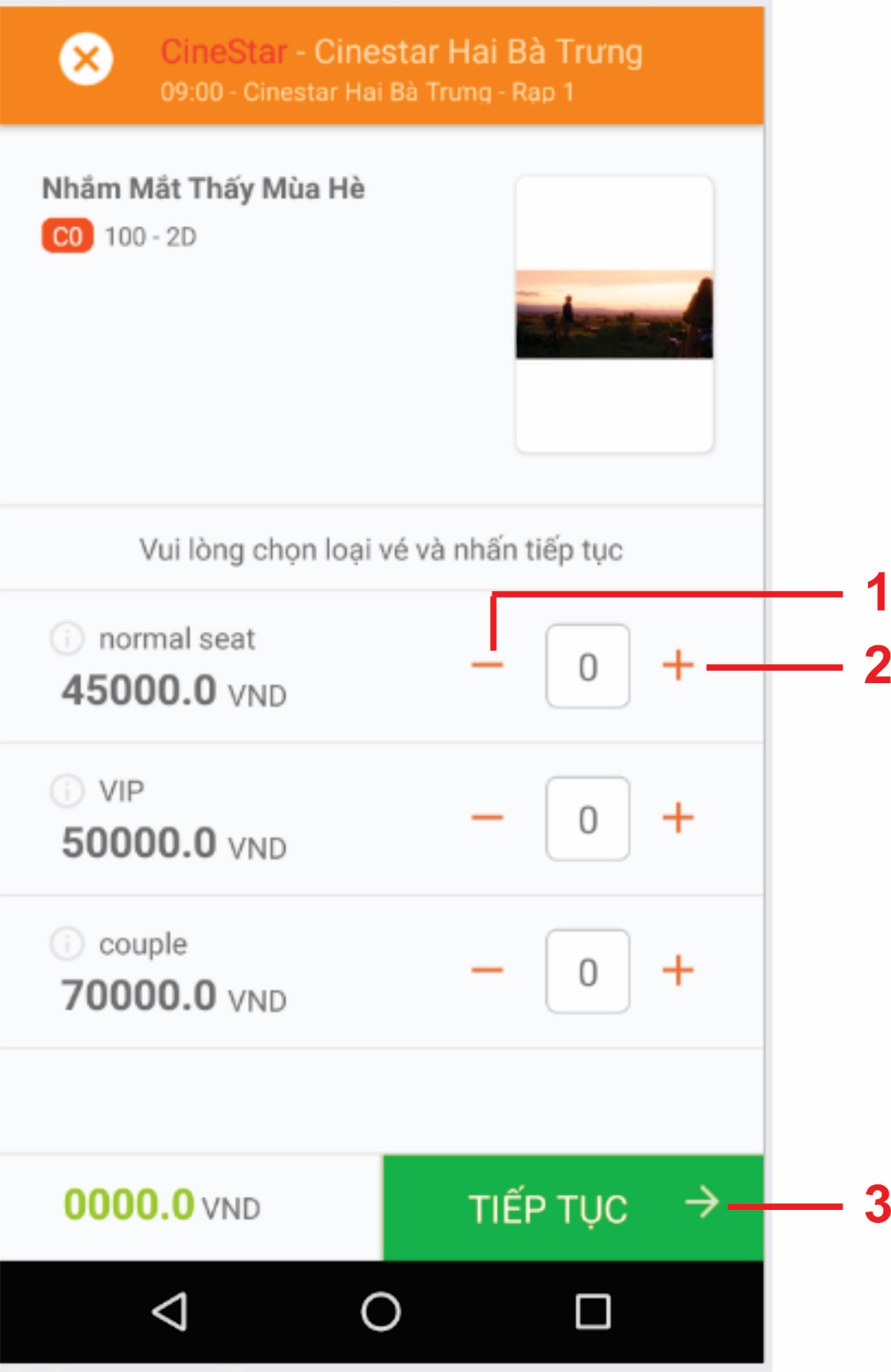


Figure 59: <Unauthorized User> ChooseTicket Page Android

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | - | Decreases seats | N/A | Decreases 1 or more seats |
| 2 | + | Increase seats | N/A | Increases 1 or more seats |
| 3 | Tiếp tục | Complete process choose ticket | N/A | Show ChooseSeat page |

Table 99: Define <Unauthorized User> ChooseTicket Page Android

#### <Unauthorized User> ChooseSeat Page Android

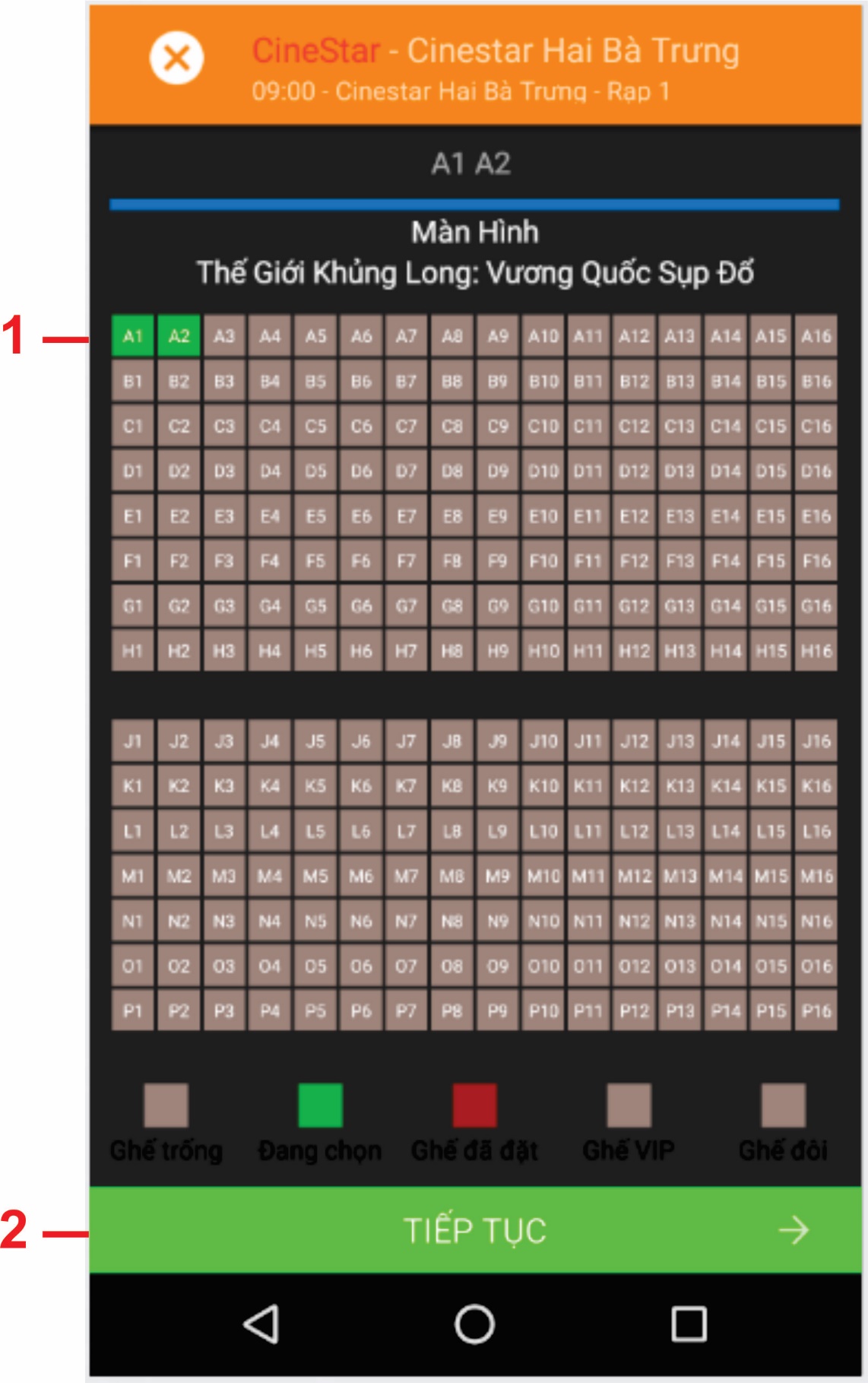


Figure 60: <Unauthorized User> ChooseSeat Page Android

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Seats | Allow choose seats allow number of ticket | N/A | Displays the selected seat and turns green. |
| 2 | Tiếp tục | Complete process choose seat | N/A | Show notification about ticket |

Table 100: Define <Unauthorized User> ChooseSeat Page Android

#### <Unauthorized User> InformationTicket Page Android

**Fields**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field name** | **Description** | **Read only** | **Mandatory** | **Control type** | **Data type** | **Length** |
| 1 | Email | Fill in Email | No | Yes | Textbox | String | N/A |
| 2 | Số điện thoại | Fill in telephone number | No | Yes | Number | String | N/A |
|  | | | | | |  |  |

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 3 | Type Payment | Click choose payments | N/A | N/A |
| 4 | Đặt vé | Complete process booking ticket | N/A | Redirect user to payments channel page. |

#### <Unauthorized User> InformationMovie Page Android



**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Play | Click play button to run the trailer | N/A | The trailer is run |

#### <Unauthorized User> InformationCinema Page Android

**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Logo image | Click on logo image will show detail time’s movie | N/A | Show all time’s movie of cinema |

#### <Unauthorized User> News Page Android

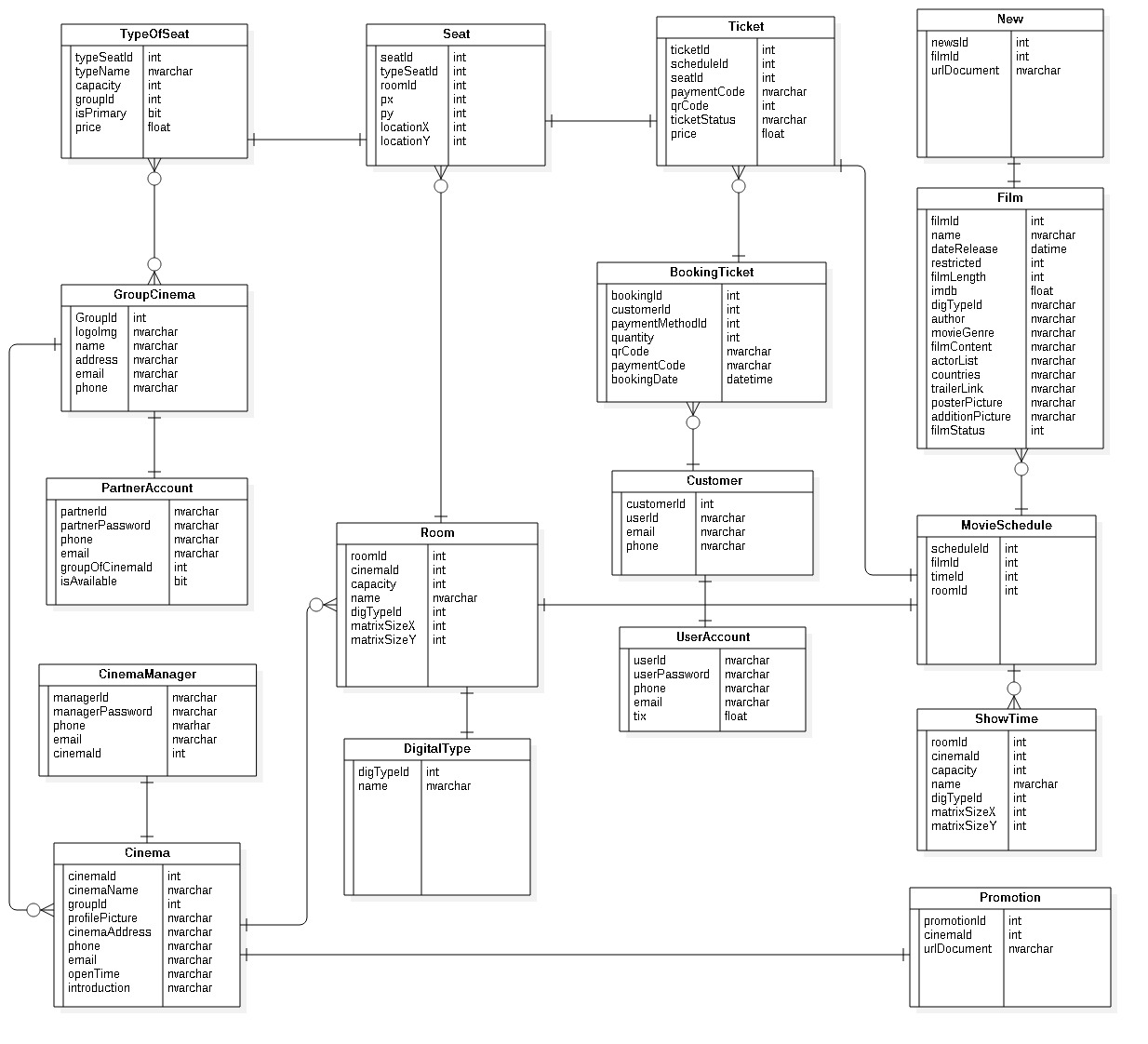


**Buttons/Hyperlinks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Detail News | Click on image or title of news will show detail news | N/A | Show news page detail |

## Database Design

### Entity Relationship Diagram (ERD)



*Figure 61: Entity Relationship Diagram (ERD)*

### Data Dictionary

|  |  |
| --- | --- |
| **Entity Data Dictionary: describe contents of all entities** | |
| **Entity Name** | **Description** |
| Film | Contains information of all films in the system. |
| Digital type | Contains information of all digital type in the system. |
| Group Cinema | Contains information of all group cinema in the system. |
| Partner Account | Contains information of all partner account |
| Cinema | Contains a list of cinemas in the system |
| Cinema Manager | Contains information of all cinema manager account |
| Show Time | Contains a list of show time in the system. |
| Room | Contains information of all room |
| Seat | Contains information of all seat |
| Type of seat | Contains all information of type of seat |
| Ticket | Contains all information of ticket |
| Booking Ticket | Contains a list of order made by users. |
| News | Contains a list news of cinema. |
| ShowTime | Contains all time of movie. |
| Customer | Contains information of all customer |
| User Account | Contains information of all user account. |
| Promotion | Contains information of all promotion. |

#### User

**Attributes**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| userId | Unique id of user. |
| userPassword | Password of each user. |
| phone | User’s phone number. |
| email | Email of user. |
| tix |  |

Table 61: User class attributes explanation

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 62: User class methods explanation

#### Customer

**Attributes**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| customerId | Unique id of customer. |
| userId | Id of the account the customer has. |
| phone | Phone number of customer. |
| email | Email of customer. |

Table 65: Customer class attributes explanation

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 66: Customer method methods explanation

#### Partner Account

**Attributes**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| partnerId | Unique id of partner. |
| partnerPassword | Password of admin. |
| phone | Phone number of each partner. |
| email | Email of each partner. |
| groupOfCinemaId | Unique id of the group of cinema each partner belong to. |
| IsAvailable |  |

Table 67: Partner class attributes explanation

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 68: Partner class methods explanation

#### Cinema Manager

**Attributes**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| managerId | Unique id of cinema manager. |
| managerPassword | Password of cinema manager. |
| phone | Phone number of cinema manager. |
| email | Email of cinema manager. |
| cinemaId | Id of the group of cinema the cinema manager belong to. |
| isAvailable |  |

Table 69: Cinema Manager class attributes explanation

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 70: Cinema Manager class methods explanation

#### Film

**Attributes**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| filmId | Unique id of film. |
| name | Film’s name. |
| dateRelease | Film’s release date. |
| restricted | Film’s age restriction. |
| filmLength | Film’s length. |
| imdb | Film’s imdb score. |
| digTypeId | Film’s shown in 2D or 3D |
| author | Film’s author |
| movieGenre | Film’s genre |
| filmContent | Film’s summary |
| actorList | Film’s list of actors |
| countries | Countries that the film is shown. |
| trailerLink | Film’s trailer |
| posterPicture | Film’s poster |
| additionPicture | Film’s additional picture |
| filmStatus | Film’s status of whether it is not shown anymore, being shown or is coming in the future. |

Table 71: Film class attributes explanation

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 72: Film class methods explanation

#### Movie schedule

**Attributes**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| scheduleId | Unique id of movie schedule. |
| filmId | Id of film that is shown on the movie schedule. |
| timeId | Id of the time the film is shown |
| roomId | Id of the room the film is shown in. |
| scheduleDate | Date of the movie schedule. |

Table 73: Movie class attributes explanation

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 74: Movie class methods explanation

#### Show Time

**Attributes**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| timeId | Unique id of the time the film is shown. |
| startTime | Time when the movie start. |
| endTime | Time when the movie end. |

Table 75: Show time class attributes

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 76: Show time class method explanation

#### Room

**Attributes**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| roomId | Unique room’s id. |
| cinemaId | Id of the cinema the room belong to. |
| capacity | Maximum number of customer the room can take in. |
| name | Room’s name. |
| digTypeId | Room is a 2D room, 3D room or iMax room |
| matrixSizeX | Room’s width. |
| matrixSizeY | Room’s Length. |

Table 77: Room class attributes explanation

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 78: Room class methods explanation

#### Digital Type

**Attributes**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| digTypeId | Unique id of digital type. |
| name | Type of digital(2D,3D,iMax). |

Table 79: Digital type class attributes explanation

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 80: Digital Type class methods explanation

#### Seat

**Attributes**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| seatId | Unique Seat’s id. |
| typeSeatId | Id of the type of seat. |
| roomId | Id of the room that have the seat. |
| px | Seat’s horizontal pixel. |
| py | Seat’s vertical pixel. |
| locationX | Seat’s horizontal location. |
| locationY | Seat’s vertical location. |

Table 81: Seat class attributes explanation

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 82: Seat class methods explanation

#### Type of Seat

**Attributes**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| typeSeatId | Unique type of seat’s id. |
| typeName | Name of the type of seat. |
| capacity | Maximum number of customer the type of seat can take in. |
| groupId | Id of the group of cinema the type of seat belong to. |
| price | Type of seat’s price. |

Table 83: Type of Seat class attributes explanation

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 84: Type of Seat class methods explanation

#### Group Cinema

**Attributes**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| groupId | Unique Id of the group of cinema. |
| logoImg | Logo of the group of cinema. |
| name | Name of the group of cinema. |
| mail | Mail of the group of cinema. |
| address | Address of the group of cinema. |
| phone | Phone of the group of cinema. |

Table 85: Group Cinema class attributes explanation

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 86: Group Cinema class methods explanation

#### Cinema

**Attributes**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| cinemaId | Unique Cinema id. |
| cinemaName | Cinema’s name. |
| groupId | Id of the group of cinema the cinema belong to. |
| profilePicture | Cinema’s profile picture. |
| cinemaAddress | Cinema’s address. |
| phone | Cinema’s phone number. |
| email | Cinema’s email. |
| openTime | The time when cinema open. |
| introduction | Cinema’s introduction. |

Table 87: Cinema class attributes explanation

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 88: Cinema class methods explanation

#### Promotion

**Attribute**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| promotionId | Unique Id of the promotion. |
| cinemaId | Id of cinema. |
| urlDocument | The document that contains the detail of the promotion. |

Table 89: Promotion class attributes explanation

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 90: Promotion class methods explanation

#### Ticket

**Attribute**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| ticketId | Unique Ticket id. |
| bookingId | Id of the booking that is on the ticket. |
| scheduleId | id of the transaction when the customer is booking a ticket. |
| seatId | Id of the seat that is on the ticket. |
| paymentCode | Code of the payment. |
| qrCode | QR’s code. |
| ticketStatus | Ticket’s status. |
| resellDecription | Decription info ticket resell |
| price | Ticket’s price. |

Table 91: Ticket class attributes explanation

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 92: Ticket class methods explanation

#### Booking Ticket

**Attribute**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| bookingId | Unique id of the transaction when the customer is booking a ticket. |
| customerId | Id of the customer. |
| paymentMethodId | Id of the method the customer is paying. |
| paymentCode | Code of the payment. |
| qrCode | QR’s code. |
| quantity | Quantity of how many tickets the customer is buying. |
| bookingDate | The date the customer book the ticket(s). |

Table 93: Booking ticket class attributes explanation

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 94: Booking ticket class methods explanation

#### Promotion

**Attributes**

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| promotionId(PK) | Unique Id of the promotion. |
| cinemaId | Id of cinema. |
| urlDocument | The document that contains the detail of the promotion. |

Table 95: Booking class attributes explanation

**Methods**

|  |  |
| --- | --- |
| **Method** | **Description** |
| Getter | Get attribute value |
| Setter | Set value of attribute |

Table 96: Booking class methods explanation

## Algorithms

# System Implementation & Test

## Introduction

### Overview

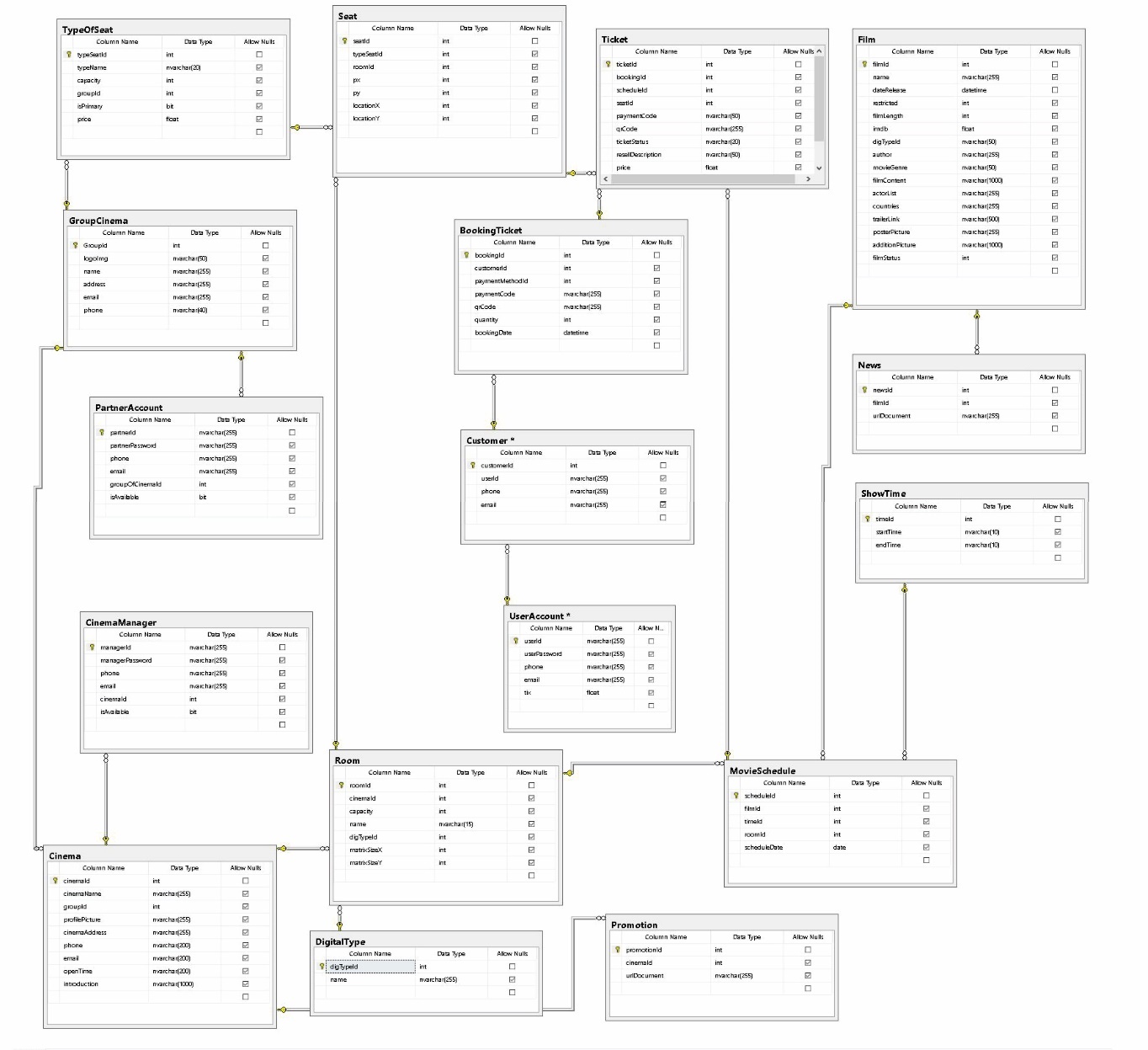
This section describes the approach and methodologies used by our group to plan, organize and manage the testing of OCB system. It provides in detail all necessary information about the implementation and testing procedure of the system included test plans, test cases, test result, test environments, pass/fail criteria and risks estimations as well as a checklist to cover all possible cases.

### Test Approach

* Goal: Test all features in the whole OCB system based on the core flow.
* Method: black-box testing
* Technique: check list

## Database Relationship Diagram

### Physical Diagram



### Data Dictionary

|  |  |
| --- | --- |
| **Entity Data Dictionary: describe contents of all entities** | |
| **Entity Name** | **Description** |
| Film | Contains information of all films in the system. |
| Digital type | Contains information of all digital type in the system. |
| Group Cinema | Contains information of all group cinema in the system. |
| Partner account | Contains information of all partner account |
| Cinema | Contains a list of cinemas in the system |
| Cinema Manager | Contains information of all cinema manager account |
| Show Time | Contains a list of show time in the system. |
| Room | Contains information of all room |
| Seat | Contains information of all seat |
| Type of seat | Contains all information of type of seat |
| Ticket | Contains all information of ticket |
| Booking Ticket | Contains a list of order made by users. |
| News | Contains a list news of cinema. |
| ShowTime | Contains all time of movie. |
| Customer | Contains information of all customer |
| User Account | Contains information of all user account. |
| Promotion | Contains information of all promotion. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | Attributes | Description | Domain | Null |
| User Account | userId(PK) | Unique id of user. | String | No |
| userPassword | Password of each user. | String | Yes |
| phone | User’s phone number. | String | Yes |
| email | Email of user. | String | Yes |
| tix |  | float | Yes |
| Admin Account | Id(PK) | Unique id of admin. | Integer | No |
| password | Password of admin. | String | Yes |
| phone | Phone number of admin. | String | Yes |
| email | Email of admin. | String | Yes |
| Customer | customerId(PK) | Unique id of customer. | Integer | No |
| userId | Id of the account the customer has. | String | Yes |
| phone | Phone number of customer. | String | Yes |
| email | Email of customer. | String | Yes |
| Cinema Manager | managerId(PK) | Unique id of cinema manager. | String | No |
| managerPassword | Password of cinema manager. | String | Yes |
| phone | Phone number of cinema manager. | String | Yes |
| email | Email of cinema manager. | String | Yes |
| cinemaId | Id of the cinema manager belong to. | Integer | Yes |
| IsAvailable |  | bit | Yes |
| Film | filmId(PK) | Unique id of film. | Integer | No |
| name | Film’s name. | String | Yes |
| dateRelease | Film’s release date. | Datetime | No |
| restricted | Film’s age restriction. | Integer | Yes |
| filmLength | Film’s length. | Integer | Yes |
| imdb | Film’s imdb score. | Float | Yes |
| digTypeId | Film’s shown in 2D or 3D | String | Yes |
| author | Film’s author | String | Yes |
| movieGenre | Film’s genre | String | Yes |
| filmContent | Film’s summary | String | Yes |
| actorList | Film’s list of actors | String | Yes |
| countries | Countries that the film is shown. | String | Yes |
| trailerLink | Film’s trailer | String | Yes |
| posterPicture | Film’s poster | String | Yes |
| additionPicture | Film’s additional picture | String | Yes |
| filmStatus | Film’s status of whether it is not shown anymore, being shown or is coming in the future. | Integer | Yes |
| Movie Schedule | scheduleId(PK) | Unique id of movie schedule. | Integer | No |
| filmId | Id of film that is shown on the movie schedule. | Integer | Yes |
| timeId | Id of the time the film is shown | Integer | Yes |
| roomId | Id of the room the film is shown in. | Integer | Yes |
| scheduleDate | Date of the movie schedule. | Date | Yes |
| Show time | timeId(PK) | Unique id of the time the film is shown. | Integer | No |
| startTime | Time when the movie start. | String | Yes |
| endTime | Time when the movie end. | String | Yes |
| Room | roomId(PK) | Unique room’s id. | Integer | No |
| cinemaId | Id of the cinema the room belong to. | Integer | Yes |
| capacity | Maximum number of customer the room can take in. | Integer | Yes |
| name | Room’s name. | String | Yes |
| digTypeId | Room is a 2D room, 3D room or iMax room | Integer | Yes |
| matrixSizeX | Room’s width. | Integer | Yes |
| matrixSizeY | Room’s Length. | Integer | Yes |
| Digital Type | digTypeId(PK) | Unique id of digital type. | Integer | No |
| name | Type of digital(2D,3D,iMax). | String | Yes |
| Seat | seatId(PK) | Unique Seat’s id. | Integer | No |
| typeSeatId | Id of the type of seat. | Integer | Yes |
| roomId | Id of the room that have the seat. | Integer | Yes |
| px | Seat’s horizontal pixel. | Integer | Yes |
| py | Seat’s vertical pixel. | Integer | Yes |
| locationX | Seat’s horizontal location. | Integer | Yes |
| locationY | Seat’s vertical location. | Integer | Yes |
| Type of Seat | typeSeatId(PK) | Unique type of seat’s id. | Integer | No |
| typeSeatName | Name of the type of seat. | Integer | Yes |
| capacity | Maximum number of customer the type of seat can take in. | Integer | Yes |
| groupId | Id of the group of cinema the type of seat belong to. | Integer | Yes |
| price | Type of seat’s price. | Float | Yes |
| Group Cinema | groupId(PK) | Unique Id of the group of cinema. | Integer | No |
| logoImg | Logo of the group of cinema. | String | Yes |
| name | Name of the group of cinema. | String | Yes |
| mail | Mail of the group of cinema. | String | Yes |
| address | Address of the group of cinema. | String | Yes |
| phone | Phone of the group of cinema. | String | Yes |
| Cinema | cinemaId(PK) | Unique Cinema id. | Integer | No |
| cinemaName | Cinema’s name. | String | Yes |
| groupId | Id of the group of cinema the cinema belong to. | Integer | Yes |
| profilePicture | Cinema’s profile picture. | String | Yes |
| cinemaAddress | Cinema’s address. | String | Yes |
| phone | Cinema’s phone number. | String | Yes |
| email | Cinema’s email. | String | Yes |
| openTime | The time when cinema open. | String | Yes |
| introduction | Cinema’s introduction. | String | Yes |
| Promotion | promotionId(PK) | Unique Id of the promotion. | Integer | No |
| cinemaId | Id of cinema. | Integer | Yes |
| urlDocument | The document that contains the detail of the promotion. | String | Yes |
| Ticket | ticketId(PK) | Unique Ticket id. | Integer | No |
| bookingId | Id of the booking that is on the ticket. | Integer | Yes |
| scheduleId | id of the transaction when the customer is booking a ticket. | Integer | Yes |
| seatId | Id of the seat that is on the ticket. | Integer | Yes |
| paymentCode | Code of the payment. | String | Yes |
| qrCode | QR’s code. | String | Yes |
| ticketStatus | Ticket’s status. | String | Yes |
| resellDecription | Decription info ticket resell | String | Yes |
| price | Ticket’s price. | float | Yes |
| Booking Ticket | bookingId(Pk) | Unique id of the transaction when the customer is booking a ticket. | Integer | No |
| customerId | Id of the customer. | Integer | Yes |
| paymentMethodId | Id of the method the customer is paying. | Integer | Yes |
| paymentCode | Code of the payment. | String | Yes |
| qrCode | QR’s code. | String | Yes |
| quantity | Quantity of how many tickets the customer is buying. | Integer | Yes |
| bookingDate | The date the customer book the ticket(s). | Datetime | Yes |

## Implementation

### Backend Architecture Implementation

Backend Architecture

*Figure 76: Backend Architecture Implementation*

|  |  |
| --- | --- |
| **Name** | **Description** |
| Clarifai | Image Recognition Service |
| Query | Contains complex queries |
| Active Record | Object relational mapping |
| Model | Business Logic Layer |
| Controller | Receives input from client |
| View | Outputs the display of information |
| Client | Web application users |
| Database | Contains a collection of schemas, table, queries |

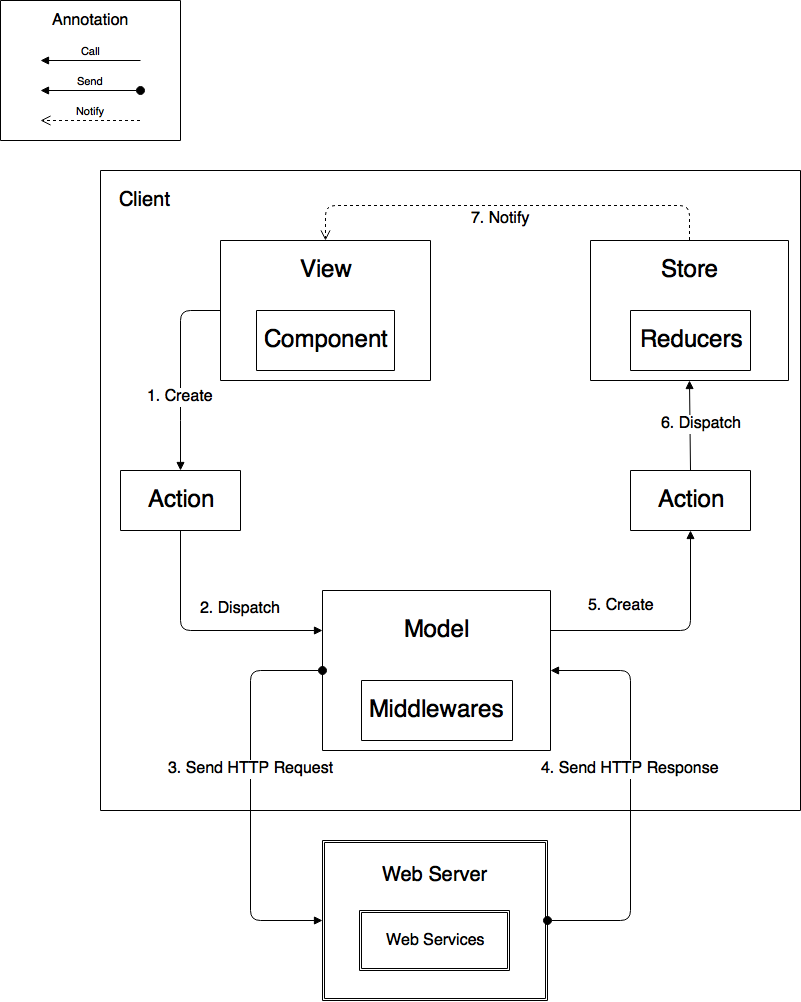
### Mobile Application Architecture Implementation

Mobile Architecture

*Figure 77: Mobile Application Architecture Implementation*

|  |  |
| --- | --- |
| **Name** | **Description** |
| Controller | Control and handle event of a screen |
| View Model | View properties and commands |
| Model | Business Logic Layer |
| Web Services | Interface to communicate with web application |
| View | Outputs the display of information |

### Web Architecture Implementation



*Figure 78: Web Architecture Implementation*

|  |  |
| --- | --- |
| **Name** | **Description** |
| View | Output display of information |
| Components | Encapsulates and reuses part of View |
| Store | Contains application states |
| Reducers | Are functions to manipulate states |
| Action | Are commands to communicate with Model and Store |
| Model | Data Access Layer |
| Middleware | Contains business logic |
| Web Server | Contains Web services |
| Web Services | Interface to communicate with web application |

### Test Driven Development

We will use Test Driven Development (TDD) to implement web services and backend logic. We choose TDD for couple of reason:

- To protect our implementation against future breaks

- To make sure we understand the requirement before writing any line of code

- To do quality assurance to our code base

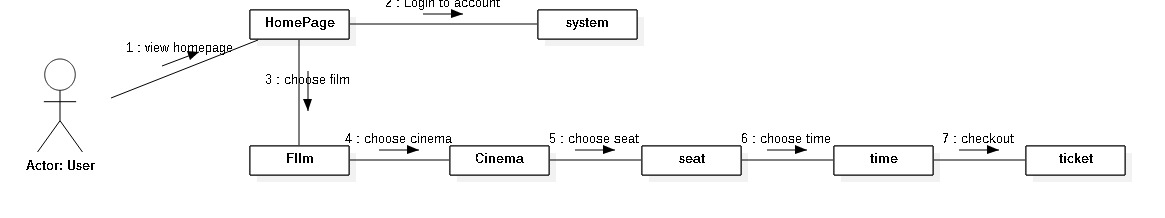
- Applying TDD brings confidence when we do refactoring our code base

The test result could be found [here](https://drive.google.com/open?id=0B0EgBSrhB9kfOHhfMUZ1Yk02U28)

## Test Plan

### Communication Diagram

#### User



#### Staff

## System Testing Test Case

### Web Application Test Cases

##### <User> Booking Ticket Web Application

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Result** | **Test Date** |
| **OSB01** | Login with account | 1. Go to homepage  2. click login  3. fill username and password  4. Click “login” button. | Web application shows “connect with account” | Pass: 9  Fail: 2 | From 5/7/2018  To 15/7/2018 |
| **OSB 02** | Login with facebook | 1. Go to homepage  2. click login  3. click “Login with Facebook”  4. choose or fill account Facebook | Web application shows “connect with account facebook” | Pass: 9  Fail: 1 | From 5/7/2018  To 15/7/2018 |
| **OSB 03** | Login with Google mail | 1. Go to homepage  2. click login  3. click “Login with google mail”  4. choose or fill account google mail | Web application shows “connect with account google mail” | Pass: 9  Fail: 1 | From 5/7/2018  To 15/7/2018 |
| **OSB 04** | Booking a ticket | 1. Go to homepage  2. click choose “mua vé”  3. click choose time of film in cinema  4. input number seat  5. click “chọn ghế”  6. input fields:  - email: “[huydxse61318@gmail.com](mailto:huydxse61318@gmail.com)”  - phone: “0908853025”  7. click button “đặt vé” | Web application shows “info page ticket” | Pass: 7  Fail: 3 | From 5/7/2018  To 15/7/2018 |
| **OSB 05** | Booking a ticket | 1. Go to homepage  2. click choose “mua vé”  3. click choose time of film in cinema  4. input number seat  5. click “chọn ghế”  6. input fields:  - email: “[huydxse61318](mailto:huydxse61318@gmail.com)”  - phone: “0908853025”  7. click button “đặt vé” | Web application show notification “email không đúng định dạng” | Pass: 9  Fail: 1 | From 5/7/2018  To 15/7/2018 |
| **OSB 06** | Booking a ticket | 1. Go to homepage  2. click choose “mua vé”  3. click choose time of film in cinema  4. input number seat  5. click “chọn ghế”  6. input fields:  - email: “[huydxse61318@gmail.com](mailto:huydxse61318@gmail.com)”  - phone: “3025”  7. click button “đặt vé” | Web application show notification “Số điện thoại không hợp lệ hoặc đầu số không được hỗ trợ.” | Pass: 9  Fail: 1 | From 5/7/2018  To 15/7/2018 |
| **OSB 07** | Choose position seat | 1. Go to homepage  2. click choose “mua vé”  3. click choose time of film in cinema  4. input number seat  Input number: 2  5. click “chọn ghế”  Position A1 and A2 | Web application continues booking ticket | Pass: 9  Fail: 1 | From 5/7/2018  To 15/7/2018 |
| **OSB 08** | Choose position seat | 1. Go to homepage  2. click choose “mua vé”  3. click choose time of film in cinema  4. input number seat  Input number: 2  5. click “chọn ghế”  Position A1 and A4 | Web application continues booking ticket | Pass: 9  Fail: 1 | From 5/7/2018  To 15/7/2018 |
| **OSB 09** | Choose position seat | 1. Go to homepage  2. click choose “mua vé”  3. click choose time of film in cinema  4. input number seat  Input number: 2  5. click “chọn ghế”  Position A1 and A3 | Web application show notification “Bạn không thể bỏ 1 ghế trống ở giữa” | Pass: 9  Fail: 1 | From 5/7/2018  To 15/7/2018 |
| **OSB 10** | Choose position seat | 1. Go to homepage  2. click choose “mua vé”  3. click choose time of film in cinema  4. input number seat  Input number: 2  5. click “chọn ghế”  Position A2 and A3 | Web application show notification “Bạn không thể bỏ trống 1 ghế ở đầu mỗi dãy” | Pass: 9  Fail: 1 | From 5/7/2018  To 15/7/2018 |
|  | | | | | |

##### <User>Booking Ticket Mobile Application

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Test Case Description** | **Test case procedure** | **Expected output** | **Result** | **Test Date** |
| **MSCP01** | Booking a ticket | 1. Go to homepage  2. click choose “mua vé”  3. click choose time of film in cinema  4. input number seat  5. click “Tiếp tục”  6. choose position seat  7. click “Tiếp tục”  6. input fields:  - email: “[huydxse61318@gmail.com](mailto:huydxse61318@gmail.com)”  - phone: “0908853025”  7. click button “Đặt vé” | Web application shows “info page ticket” | Pass: 8  Fail: 2 | From 5/7/2018  To 15/7/2018 |
| **MSCP02** | Booking a ticket | 1. Go to homepage  2. click choose “mua vé”  3. click choose time of film in cinema  4. input number seat  5. click “Tiếp tục”  6. choose position seat  7. click “Tiếp tục”  6. input fields:  - email: “[huydxse61318](mailto:huydxse61318@gmail.com)”  - phone: “0908853025”  7. click button “Đặt vé” | Web application shows notification “xin vui long kiểm tra email hoặc số điện thoại” | Pass: 8  Fail: 2 | From 5/7/2018  To 15/7/2018 |
| **MSCP03** | Booking a ticket | 1. Go to homepage  2. click choose “mua vé”  3. click choose time of film in cinema  4. input number seat  5. click “Tiếp tục”  6. choose position seat  7. click “Tiếp tục”  6. input fields:  - email: “[huydxse61318@gmail.com](mailto:huydxse61318@gmail.com)”  - phone: “090885”  7. click button “Đặt vé” | Web application shows notification “xin vui long kiểm tra email hoặc số điện thoại”” | Pass: 8  Fail: 2 | From 5/7/2018  To 15/7/2018 |
| **MSCP04** | Choose position seat | 1. Go to homepage  2. click choose “mua vé”  3. click choose time of film in cinema  4. input number seat  Input number: 2  5. click “chọn ghế”  Position A2 and A3 | Web application show notification “Vui long không để trống vị trí đầu dãy hoặc giữa hai ghế đã được đặt” | Pass: 9  Fail: 1 | From 5/7/2018  To 15/7/2018 |
| **MSCP05** | Choose position seat | 1. Go to homepage  2. click choose “mua vé”  3. click choose time of film in cinema  4. input number seat  Input number: 2  5. click “chọn ghế”  Position A1 and A3 | Web application show notification “Vui long không để trống vị trí đầu dãy hoặc giữa hai ghế đã được đặt” | Pass: 9  Fail: 1 | From 5/7/2018  To 15/7/2018 |
|  | | | | | |

# Software User’s Manual

## Installation Guide

### Setting up environment at server side

The specifications are based on the dependencies requirements and performance test result from previous section of this document.

#### Hardware Requirements

|  |  |
| --- | --- |
| **Hardware** | **Specification** |
| Internet Connection | 4 Mbps |
| Computer Processor | Intel® Core i3 1.4GHz or more |
| Computer Memory | 2GB of RAM or more |
| Hard Drive | 30GB or more |
| *Table 94: Installation Guide - Hardware Requirements* | |

#### Software Requirements

|  |  |
| --- | --- |
| **Software** | **Application name/version** |
| Operating System | Ubuntu 12.04 LTS or later |
| Ruby | RubyMine |
| DBMS | PostgresSQL |
| *Table 95: Installation Guide - Software Requirement* | |

#### Mobile Requirements

|  |  |
| --- | --- |
| **Hardware** | **Specification** |
| CPU | Quad-core 1500 MHz, Krait |
| Memory | 2 GB |
| Storage | 16 GB |
| Model | Google Nexus 4 |
| *Table 96: Installation Guide - Mobile Requirement* | |

## User Guide

# Appendix

1. UML standard diagrams:

<https://www.tutorialspoint.com/uml/uml_standard_diagrams.htm>

1. Component diagram notation:

<https://www.ibm.com/developerworks/rational/library/dec04/bell/>

1. algorithm:

1. UML Diagrams:

<http://www.uml-diagrams.org/>