<A BIDV based smart banking app>

Use-Case-Realization Specification: <Book movie ticket online>

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <20/12/2022 > | <1.0> | <Analysis and write use case> | <Đàm Thị Linh> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 2

1.1 Purpose 2

1.2 Scope 2

1.3 Definitions, Acronyms, and Abbreviations 2

1.4 References 2

1.5 Overview 2

2. Flow of Events—Design 2

3. Derived Requirements 2

Use-Case-Realization Specification: <Booking movie ticket online>

# Introduction

## Purpose

This document describes how the use case for booking movie tickets online using a smart banking app is realized in a design model, which is specially illustrated by sequence diagram and class diagram.

## Scope

This document applies to book movie ticket online on a mobile smart banking app

## Definitions, Acronyms, and Abbreviations

User: Customer who uses the service of smart banking app

Cinema view: GUI displayed to show location with corresponding cinema group

Cinema controller: controller to do tasks relating to cinema

Cinema group: contains many cinemas

Now Showing view: GUI to show movies which are showing in some cinemas

Showtimes view: GUI to show time slots with day

Simulation view: simulate a real cinema for user to choose seat(s)

Checkout view: display all the transaction information (ticket)

Now Showing, showtimes, simulation controller: controller which is corresponding to its view

Account: contain user’s money on this bank

Timeslots: some of the timeline that the movie will be displayed on a cinema

## References

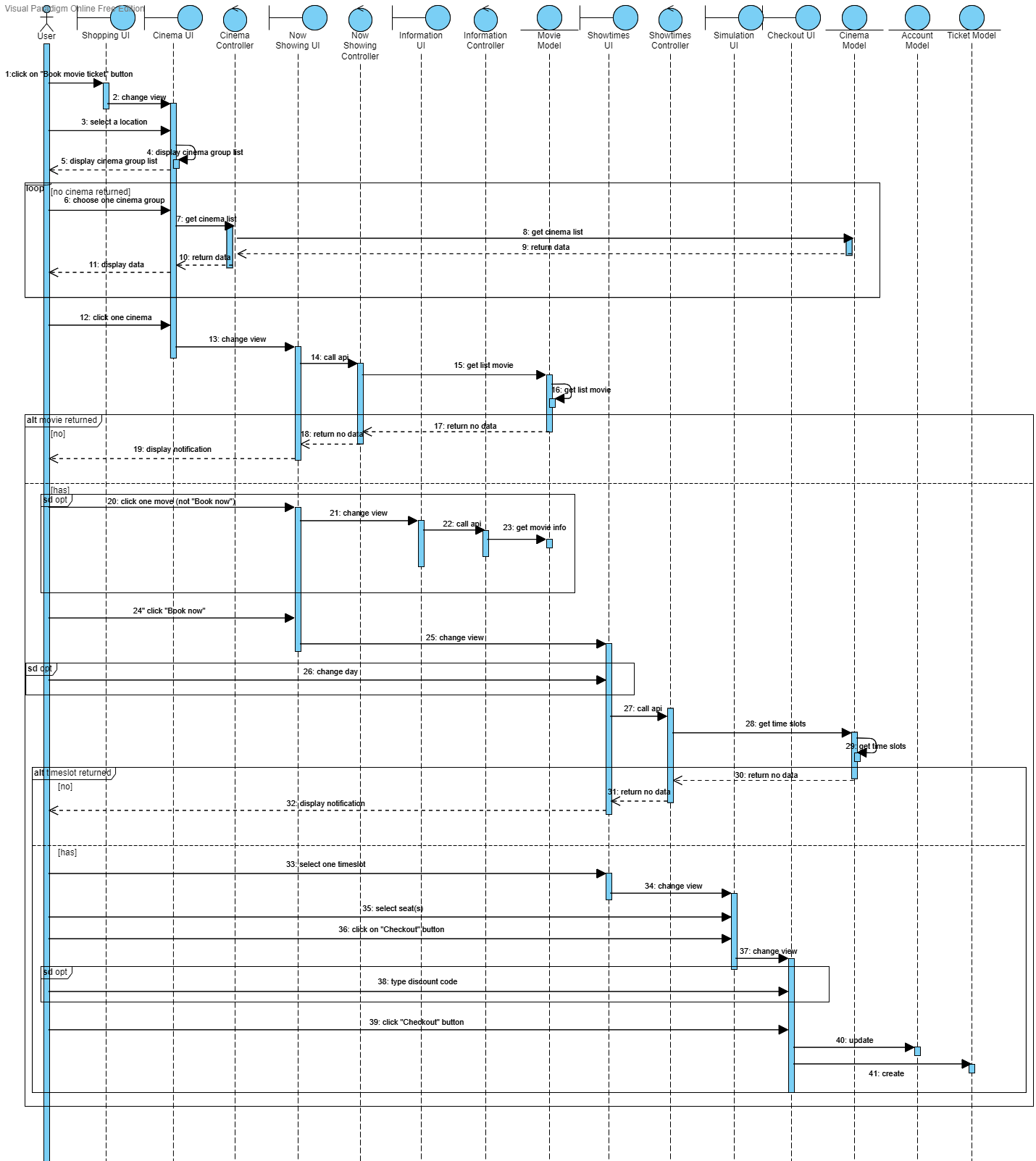
None

## Overview

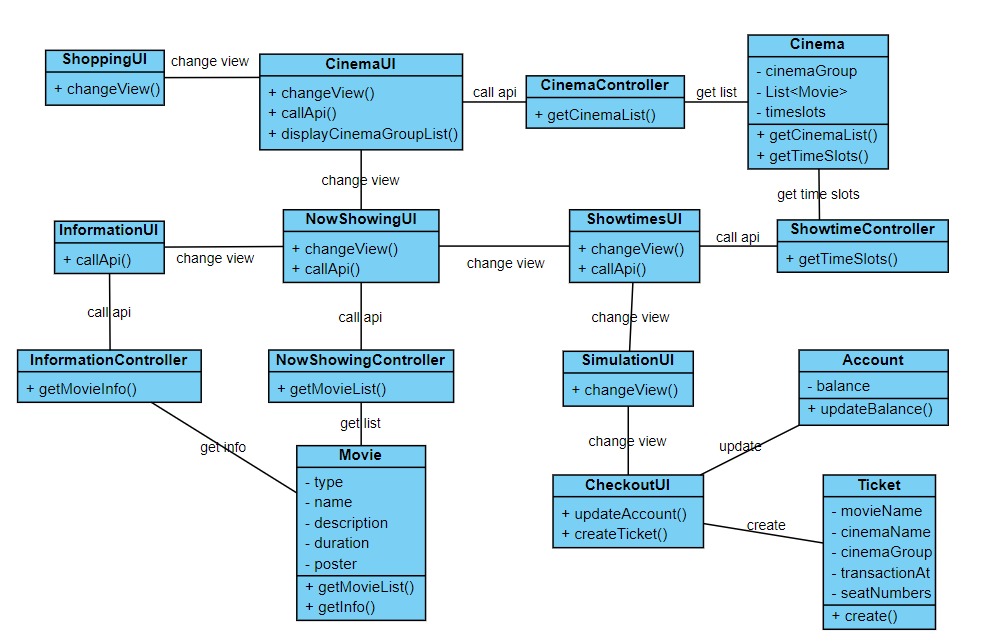
The Use-Case Realization Specification for booking movie tickets online is offered in detail to aid in understanding and implementation. It will be summarized into a textual description in this document. There are sequence and class diagrams that show how the use case is implemented. The final section, "Derived Requirements," specifies all requirements, including non-functional requirements, on use-case realizations that were not taken into account in the design model but that must be addressed during implementation in text.

# Flow of Events—Design

## sequence diagram



## class diagram



# Derived Requirements

3.1. User has to choose at least one seat(s) and that seat(s) have to conform to some rules such as no leftmost or rightmost empty space, not left empty seat between 2 seats that user choose

3.2. On the checkout view, the user can type a piece of discount code

3.3. On now showing view, 10 days are displayed for user to choose