CinemaGO

Analysis and Design Document

Student: Ruști Maria

**Group: 30238**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <23/mar/19> | <1.0> | <details> | Ruști Maria |
| <16/may/19> | <1.0> |  | Ruști Maria |
| <03/june/19> | <1.0> |  | Ruști Maria |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

This project represents an online application which will allow users to register in order to buy tickets and reserve cinema seats and the administrator to manage the operations available.

Users will have to create accounts, login and choose the movie they want to watch. They can then buy tickets for that movie and reserve seats if they have preferences. Otherwise, the seats will be automatically generated for them.

The one who keeps the app up to date is the administrator, who will be able to add and delete movies or update the schedule and prices.

# Elaboration – Iteration 1.1

# Domain Models

*[Define the domain model and create the conceptual class diagrams]*

Models: User, Movie, Ticket, Cart, Order

# Architectural Design

## Conceptual Architecture

*[Define the system’s conceptual architecture; use an architectural style and pattern - highlight its use and motivate your choice.]*

Layered Architecture Pattern

A **layer** is a group of classes that have the same set of link-time module dependencies to other modules.In other words, a layer is a group of reusable components that are reusable in similar circumstances.

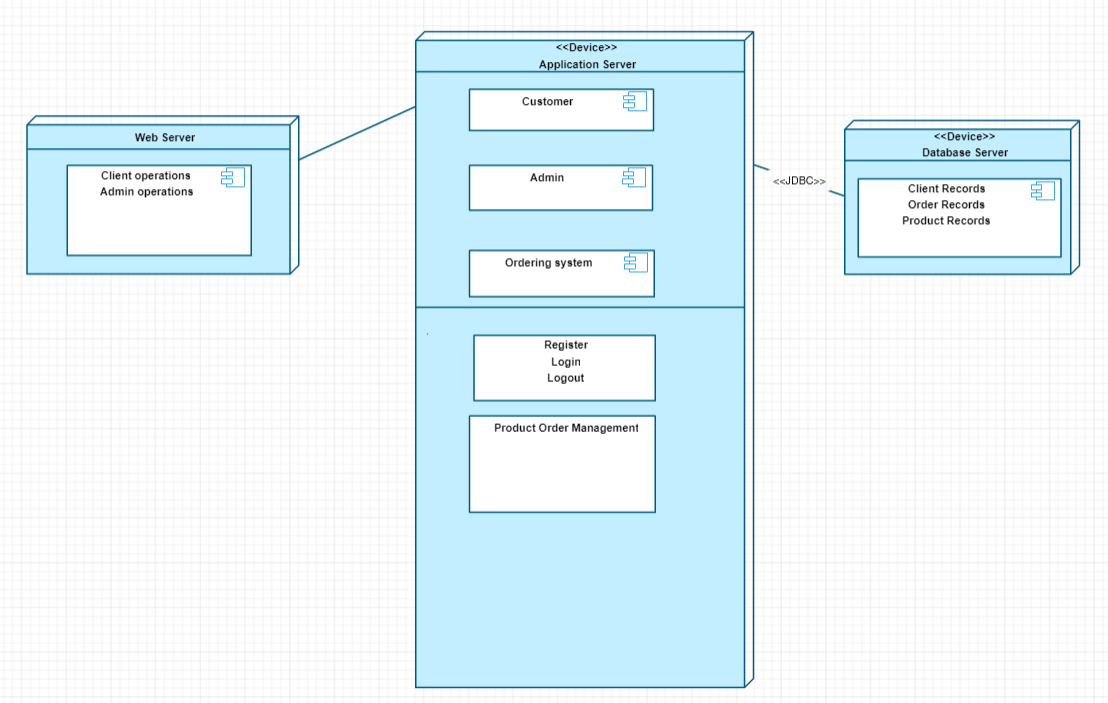
One of the powerful features of the layered architecture pattern is the separation of concerns among components. Components within a specific layer deal only with logic that pertains to that layer.

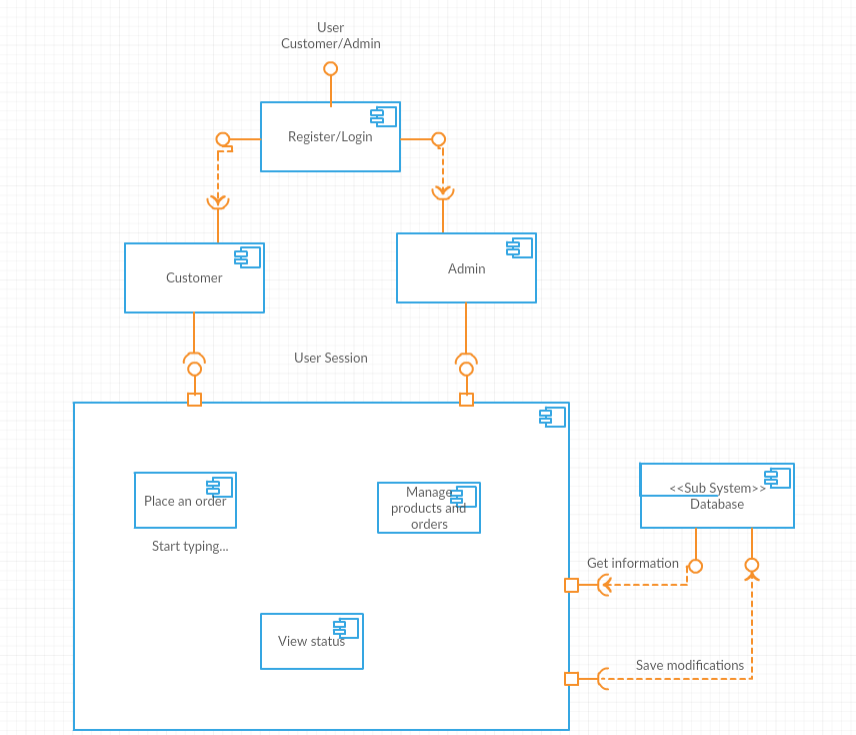
## Package Design

*[Create a package diagram]*

## Component and Deployment Diagrams

*[Create the component and deployment diagrams.]*



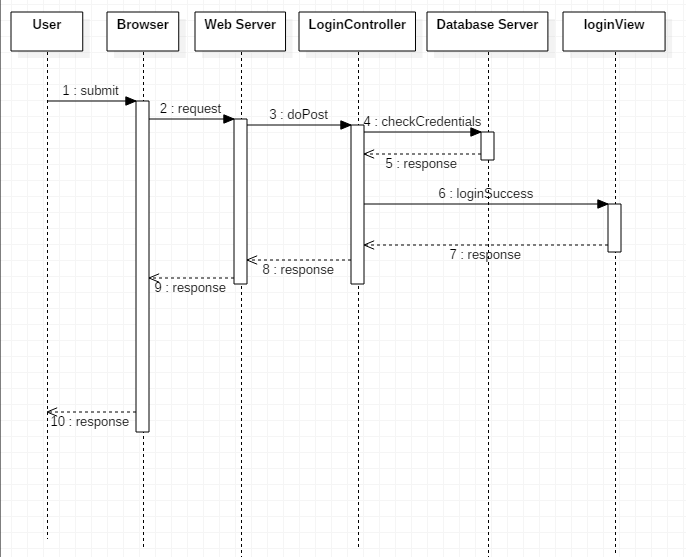


# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

*[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]*

**

## Class Design

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

Design Patterns used:

* Observer
* Mediator

# Data Model

*[Create the data model for the system.]*

# Unit Testing

*[Present the used testing methods and the associated test case scenarios.]*

The system was tested using Junit.

The following scenarios were tested: register user, buy ticket, add movie, delete movie

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

*[Describe how you applied integration testing and present the associated test case scenarios.]*

# Future improvements

The system can be improved by adding more functionalities and making the users' interaction with the application more simple and as efficient as possible, in terms of accomplishing their goals.

# Bibliography

<https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html>

<https://en.wikipedia.org/wiki/Factory_method_pattern>

<https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html>

<https://www.thymeleaf.org/doc/articles/springmvcaccessdata.html>