Documentation Report

for

MOBLIMA

Version 1.0 approved

Prepared by

LEE JUIN,

JONATHAN NG HONG JIN,

OI YEEK SHENG,

JERICK LIM KAI ZHENG,

LEE CI HUI

SSP2 Group 1, Nanyang Technological University

2022-11-08

Table of Contents

Preface i

1. Table of Contents ii
2. Revision History iii
3. Introduction 1
   1. Document Conventions 1
   2. Class Model 1
   3. Testing 1
   4. Assumptions & Dependencies
4. Design Standards 4
   1. Software Engineering Principles 4
   2. Object-oriented Design Principles 5
   3. Proposed Features

Appendix A: Data Dictionary 63

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| LEE JUIN | 2022-11-08 | Initial write-up. | 1.0 |

# Introduction

## Document Conventions

This section covers the conventional standards used throughout this document.

**Font**: Times New Roman

**Heading:** Bold, Size 18

**Sub-heading:** Bold, Size 14

**Content:** Italic, Size 12

**Technical Standards:** IEEE 830-1998

## Class Model

The MOBLIMA Application is developed using Object-oriented Design Strategies. To begin development, the team started by constructing a concise and comprehensive class model, which comprises of two sub-models —— Entity Class model and Control Boundary Class model. The former helped the team in developing persistent data relationships while the latter depicted a general overview of the program flow.

The development team proceeds to integrate the constructed models to form the over-arching system architecture. Further details regarding the considerations on the system architecture design can be found under section *2. Design Standards*.

The Class Models are presented as follows. A supplementary copy of each class model diagram is provided independent of this document for clarity.

### Diagram Description automatically generatedEntity Class Model

### Control Boundary Class Model

## Testing

After development, the MOBLIMA Application has undergone several Unit Testing and Integration Testing. This section covers the techniques used during the testing phase. Supplementary copies of the testing results in the form of pictorial screenshots are provided.

### White-box Testing

### Grey-box Testing

## Assumptions & Dependencies

This section covers all the assumptions made by the development team during the development phase. This section also serves as a repository for all dependencies assumed throughout the development of the project.

### Assumptions

* The MOBLIMA Application is designed for no concurrent user. Hence, exceptions such as race condition from multiple users booking the same seat are not handled by the application.
* The MOBLIMA Application is designed with a pseudo-manual notion. No external server is hosting the MOBLIMA Application. Hence, actions such as removing a showing time can only be done manually upon start-up of the application by the administrator.
* The MOBLIMA Application is designed with a simple login feature. No help is provided for lost account retrieval.
* The MOBLIMA Application does not verify the legitimacy of Movie Goers who self-proclaimed as either Student, Child, or Senior Citizen. The verification shall be done by the vendor instead.
* The MOBLIMA Application does not verify the legitimacy of Movie Goers’ payment information.
* The MOBLIMA Application assumes that a movie may have multiple showing types, i.e., a movie can be showed as a 3D movie, or a standard movie.
* The MOBLIMA Application assumes that a movie’s review is independent of its showing type. Hence, all reviews under the same movie title are displayed to the user upon query, regardless of the showing types available.
* The MOBLIMA Application assumes that a movie’s title is always unique. Hence, no exception handling regarding movie of the same name is done.

### Dependencies

|  |  |
| --- | --- |
| Java support | The MOBLIMA Application is developed under Java SE 7. |
| Command-line Interface (CLI) support | The MOBLIMA Application requires a terminal window to operate.  The Command Prompt from Windows or Terminal on MacOS will suffice for the application. |

# Design Standards

## Software Engineering Principles

This section covers all the good Software Engineering principles practiced and applied throughout the development process.

### Model-View-Controller (MVC) Design Pattern

The MOBLIMA Application adopts a Model-View-Controller (MVC) system architectural design pattern. The Model component holds all persistent data of the application, while the Controller component represents all the business logic behind the application. Finally, the View component is the CLI presented to the Movie Goers.

## Object-oriented Design Principles

### Single-Responsibility Principle

### Open-Closed Principle

### Dependency Inversion Principle

## Proposed Features