Cinema Management System

Software Architecture Document

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 12/12/2020 | 0.1 | Initial document | Ngô Thị Ngọc Thơ |
| 24/12/2020 | 1.0 | Updated document | Ngô Thị Ngọc Thơ |
| 01/01/2021 | 1.0 | Reviewed and updated document | Ngô Thị Ngọc Thơ |

**Mục lục**

[**1.** **Introduction** 4](#_Toc60601804)

[**1.1.** **Purpose** 4](#_Toc60601805)

[**1.2.** **Scope** 4](#_Toc60601806)

[**1.3.** **Definition, Acronyms and Abbreviations** 4](#_Toc60601807)

[**1.4.** **References** 4](#_Toc60601808)

[**2.** **Architectural Goals and Constraints** 4](#_Toc60601809)

[**3.** **Use-Case Model** 5](#_Toc60601810)

[**4.** **Logical View** 5](#_Toc60601811)

[**4.1.** **Overview** 5](#_Toc60601812)

[**4.1.1.** **Systems** 5](#_Toc60601813)

[**4.1.2.** **Layering** 5](#_Toc60601814)

[**4.2.** **Architecturally Significant Design Packages** 6](#_Toc60601815)

[**5.** **Deployment View** 6](#_Toc60601816)

**Software Architecture Document**

1. **Introduction**
   1. **Purpose**

The subject "Cinema management system" is intended to fully describe the features of the Cinema management system. It provides project development participants as well as the customer understanding what the system is capable of and how the process is done.

In fact in the software development process, at each stage a different document is built corresponding to that stage. To minimize the complexity of documents in the teaching process, this material is consistently developed throughout the project development process.

* 1. **Scope**

This document covers all stages of the CMS application development process up until the beginning of the programming process.

* 1. **Definition, Acronyms and Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Description** |
| **SAD** | Software Architecture Document |
| **RCP** | Cinema Management System |

* 1. **References**

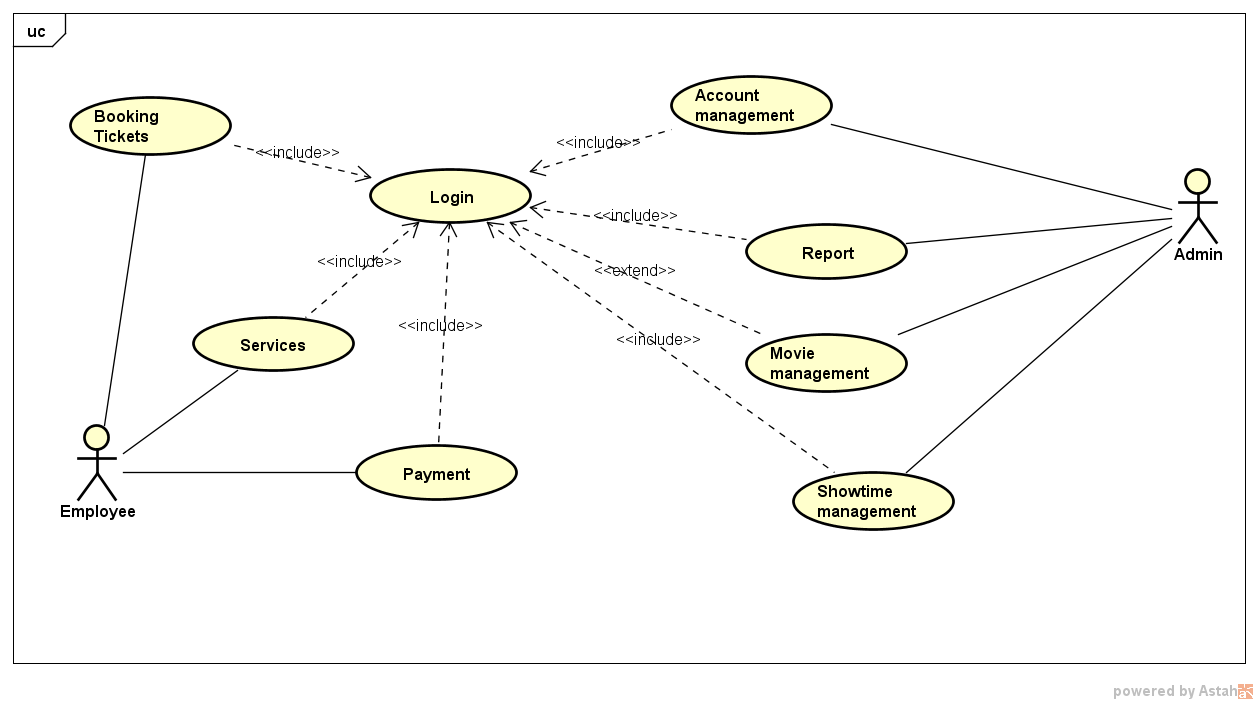
[1] CMS Software Architecture Document

[2] Documents of lecturer Nguyen Trong Minh Hong Phuoc - Saigon International University.

1. **Architectural Goals and Constraints**

There are some key requirements and system constraints that have a significant bearing on the architecture. They are:

1. The system is meant as a proof of concept for a more complete project prediction system to be built in the future. Therefore one of the primary stakeholders in this document and the system as a whole are future architects and designers, not necessarily users as is normally the case. As a result, one goal of this document is to be useful to future architects and designers.
2. The system will be written using Microsoft .NET technologies but will use an open source RDBMS system (MySQL) for data persistence and will be deployed to a Linux webserver running Mono. These special deployment requirements require additional consideration in the development of the architecture.
3. The system must communicate with multiple third-party APIs, Assembla and Google Predictive. Defining how the system interfaces with these third-party systems is a primary concern of the architecture.
4. Section 3.3 of the Software Requirements Specification outlines a number of anticipated changes that the application could face over time. One of the primary goals of the system architecture is to minimize the impact of these changes by minimizing the amount of code that would need to be modified to implement them. The architecture seeks to do this through the use of modularization and information hiding to isolate components that are likely to change from the rest of the system.
5. **Use-Case Model**



1. **Logical View**
   1. **Overview**
      1. **Systems**

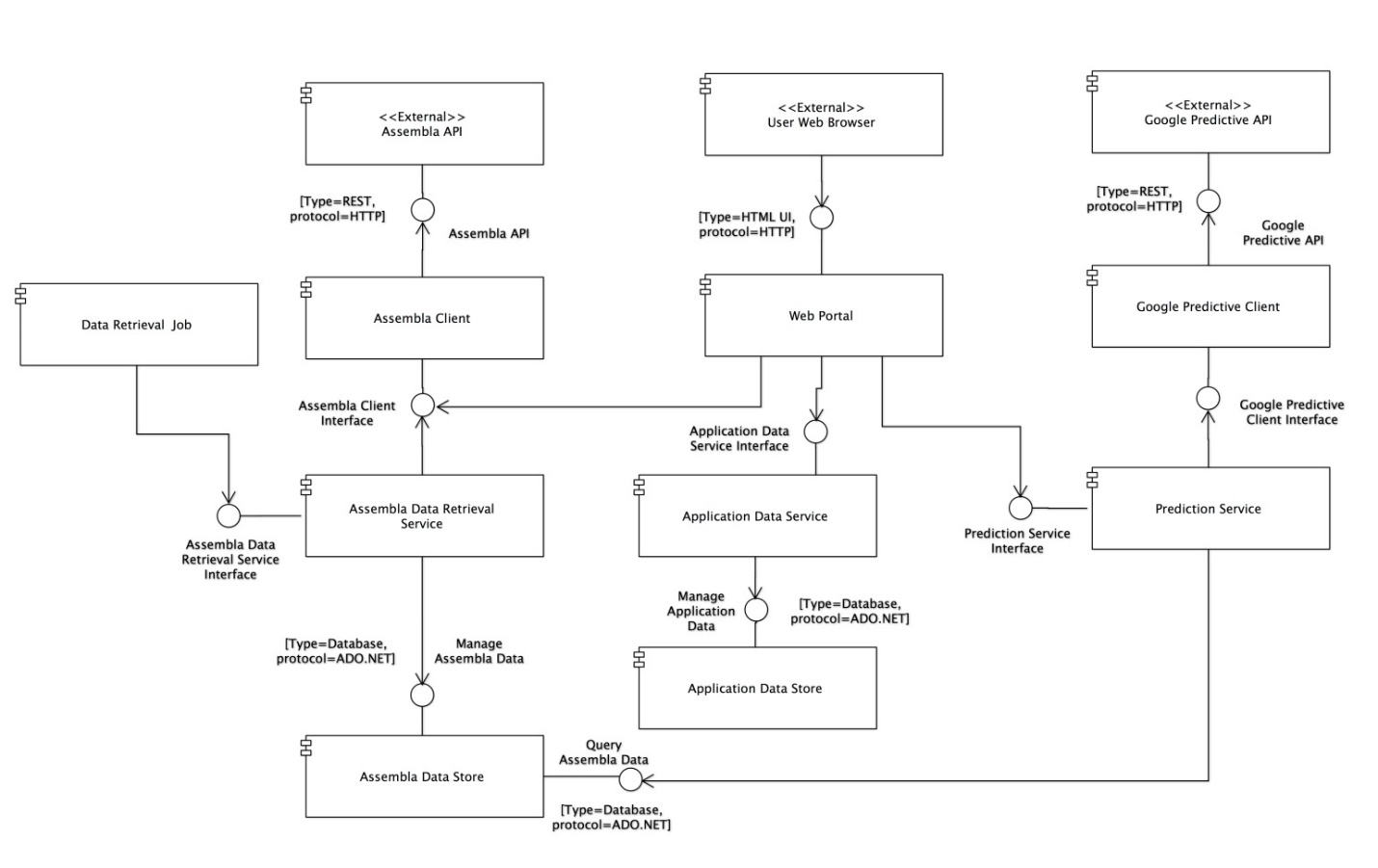
CMS application can be divided into 3 main sub systems.

* Admin management system
* Employee management system
  + 1. **Layering**

CMS application is divided into 3 layers. The layering model of the HMS application is based on a responsibility layering strategy that associates each layer with a particular responsibility. This strategy has been chosen because it isolates various system responsibilities from one another, so that it improves both system development, reusability and maintenance.



* 1. **Architecturally Significant Design Packages**



1. **Deployment View**

The web application will be hosted on a single physical server. An Apache webserver running a mono module will be used to serve the application pages. In addition, a MySql Server instance will also be hosted on the physical server to aid the application in persisting data.

The application will interface with external APIs (Assembla, Google Predictive), of which the deployment scenarios are not known.

The application’s deployment specifics can be seen below.

