**Topic Description**

An application for maintaining a movie theater reservation system. The system has the following minimum requirements:

• The theater has several (differently sized) halls

• The theater shows several movies at the same time

• Each hall has several screenings a day

• A customer can reserve a seat for any show (provided there is room in the hall)

• The administrator can add movies and screenings as well as browse reservations

**Solution Principle**

Users interact with the system via graphical interface.

The system uses an object-oriented design, abstracting entities into classes.

Data persistence is achieved through file storage system.

**Project Structure**

Data Model Module: core system entities

|  |  |
| --- | --- |
| **Entities** | **Comment** |
| Cinema |  |
| ScreeningRoom |  |
| Movie |  |
| Screening |  |
| Reservation |  |

Business Logic Module: implement core functions

* ReservationManager: Handle reservation-related operations
* AdminManager: Handles administrator operations

Data Storage Module:

|  |  |
| --- | --- |
| **Entities** | **Comment** |
| DataStorage | Responsible for loading and saving data |

User Interface Module:

* UserInterface

**External Libraries Used**

|  |  |
| --- | --- |
| **Usage** | **Library** |
| Data Storage | json |
| Graphical User Interface | tkinter |
| DataTime Handling | datetime |
| Data Validating | re |

**User Process**

User Reservation Process

User → Select Movie → Select Screening → Choose Seats → Confirm Reservation → System Validation → Generate Reservation Code → Save Data

Administrator Add Screening Process

Administrator → Login → Select "Add Screening" → Input Movie, Time, Screening Room → System Validation → Save Data