**University of Information Technology and Sciences**

**Department of CSE**

# Project Report

**of**

**Movie Ticket Booking System**

|  |  |
| --- | --- |
| Course code : | CSE0613122 |
| Course title : | Object Oriental Programing Lab |
| Assignment No. | 01 |
| Assignment title : | Movie ticket booking system. |
| Submission date : | 16 Dec 2024 |

|  |  |
| --- | --- |
| Submitted by: | Submitted to: |
| Name : Rayhan Hamim    Id : 0432410005101012    Batch : CSE 55    Section : A1 | Name : Al Imtiaz        Dept. of CSE    University of Information Technology and Sciences. |

**Abstract**

This report outlines the development and functionality of the **TICKETSPOT** system, a Java-based ticket booking application. The system includes essential features like user authentication, movie selection, payment processing, and administrative functionalities. Key components include Java source files, class diagrams, data files, and a graphical interface, which provide a comprehensive ticket management solution.

**Objectives**

1. To design and implement a **ticket booking system**.
2. To utilize Java programming for managing user interaction, class structures, and logical flow.
3. To integrate user and admin functionalities for seamless booking operations.
4. To store and manage essential data like user login credentials and movie details.
5. To incorporate graphical components for user-friendly interaction.

**Equipment and Components**

* **Hardware**: Computer system with Java JDK installed.
* **Software**:
  + Java Development Kit (JDK)
  + Integrated Development Environment (IDE): Eclipse/IntelliJ
  + Text Editor: Notepad++
* **Files**:
  + start.java (Main file)
  + Class files (Homepage.java, Login.java, Admin.java, etc.)
  + Data files: data.txt, login data.txt
  + Images: For UI components (avatars, movie posters, payment icons).

**Class Diagram**

The project uses multiple classes for modular functionality:

* **start.java**: Initializes the project with the Homepage class.
* **Homepage.java**: Acts as the main hub for user interaction.
* **Login.java**: Handles user authentication.
* **Admin.java**: Administrative functionalities like managing movies and schedules.
* **Registration.java**: User registration component.
* **payment.java**: Processes payments.
* **showtime.java**: Displays available showtimes.
* **ticketPrice.java**: Manages ticket pricing logic.

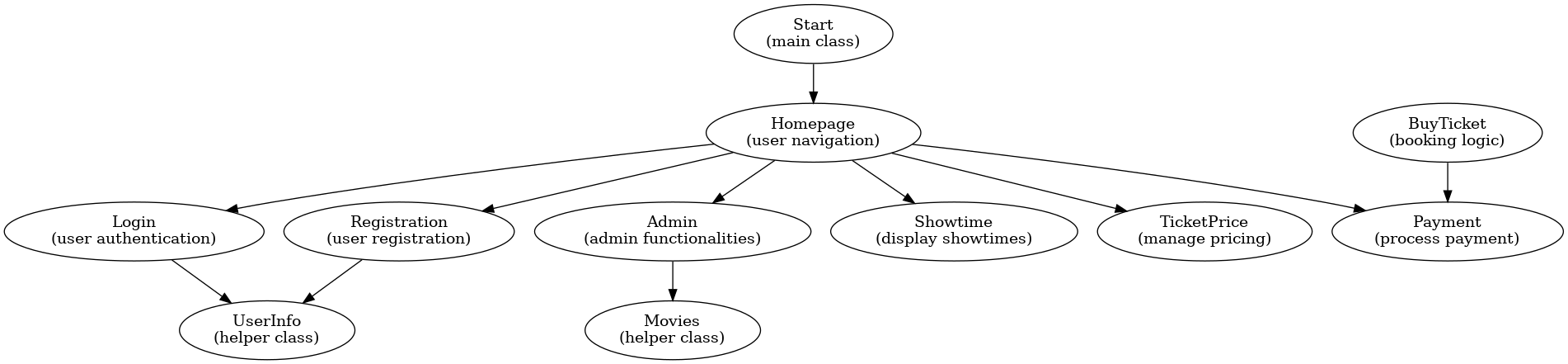


Figure: Class Diagram

Additional helper classes include userinfo.java, movies.java, and buyTicket.java, ensuring a clean and modular code structure.

**Theory**

The **TICKETSPOT** system is built upon the principles of **Object-Oriented Programming (OOP)**. OOP allows the system to break down complex processes into manageable classes and objects, each encapsulating data and behavior. Key concepts include:

* **Classes and Objects**: Every functionality, such as user login, payment processing, and movie management, is organized into distinct classes. Objects created from these classes allow interaction between different components of the system.
* **Inheritance**: Common functionalities can be extended or reused across classes using inheritance, ensuring code reusability and maintainability.
* **Encapsulation**: Data related to user inputs, login credentials, and ticket details is securely handled and hidden within the respective classes.
* **File Handling**: The system uses data.txt and login data.txt for persistent storage, allowing user information and booking details to be saved and retrieved efficiently.
* **Graphical Integration**: Images and UI elements are incorporated to enhance the user experience, making the interface visually appealing and intuitive.

By leveraging OOP, the **TICKETSPOT** system ensures modularity, scalability, and maintainability, providing a strong foundation for future enhancements and updates.

**Methodology**

1. **Project Initialization**:
   * The project begins with the creation of the main file, start.java, which serves as the entry point and initializes the Homepage class.
2. **System Design**:
   * Functionalities such as login, movie selection, and payment were modularized into separate classes like Login.java, Admin.java, and payment.java.
   * The modular approach ensures separation of concerns and easy maintenance.
3. **Data Management**:
   * Essential user credentials and booking records are stored in text files (data.txt and login data.txt).
   * File handling techniques were used to read, write, and manage these records efficiently.
4. **Graphical Integration**:
   * Visual components like images (avatars, movie posters, and icons) were integrated to create an interactive user interface.
5. **Testing**:
   * Each component was compiled and executed separately to validate proper functioning. The interaction between modules was tested to ensure seamless user experience.
6. **Deployment**:
   * The final project was deployed with all required files, including text files for data persistence and images for graphical enhancements.

**Code**

import Classes.Homepage;

public class start {

public static void main(String[] args) {

Homepage h = new Homepage();

}

}

**Additional Files**: [Other classes include Login.java, Admin.java, etc. Refer to project files for complete source code.]

**Observations**

1. The system initializes successfully from the start.java file.
2. The Homepage acts as the navigation hub for users.
3. User login and registration functionality works effectively with data.txt as the backend storage.
4. Movie selection, ticket booking, and payment processes integrate seamlessly.
5. Administrative tools allow movie and showtime management.

**Discussion and Analysis**

The **TICKETSPOT** system demonstrates a modular and scalable design through its well-defined class structure and user-friendly graphical interface. By implementing Object-Oriented Programming principles, the project ensures reusability and easy maintenance of code. The integration of file handling for data storage provides simple yet effective management of user credentials and booking information. However, the system lacks advanced error handling and backend database support, which limits scalability and robustness. Enhancing the project with a relational database (e.g., MySQL) and adding error-logging mechanisms would significantly improve performance and reliability. Overall, the project successfully meets its primary objectives while leaving room for future improvements and enhancements.

**Conclusion**

The **TICKETSPOT** project successfully implements a basic ticket booking system using Java. It demonstrates core concepts of object-oriented programming, file handling, and graphical integration. The system meets the primary objectives but can be further enhanced with database support and improved user experience.

**References**

1. Oracle Java Documentation -<https://docs.oracle.com/javase/>
2. Java OOP Principles - <https://www.w3schools.com/java/>
3. File Handling in Java - <https://www.geeksforgeeks.org/file-handling-java/>