ReelEasy - Movie Theater Ticketing System

SDS: Software Design Specifications

Version 1.0

7/22/2025

Group 2
Bobby Bavongkhoun
Caleb Wolf
Gleb Rodin

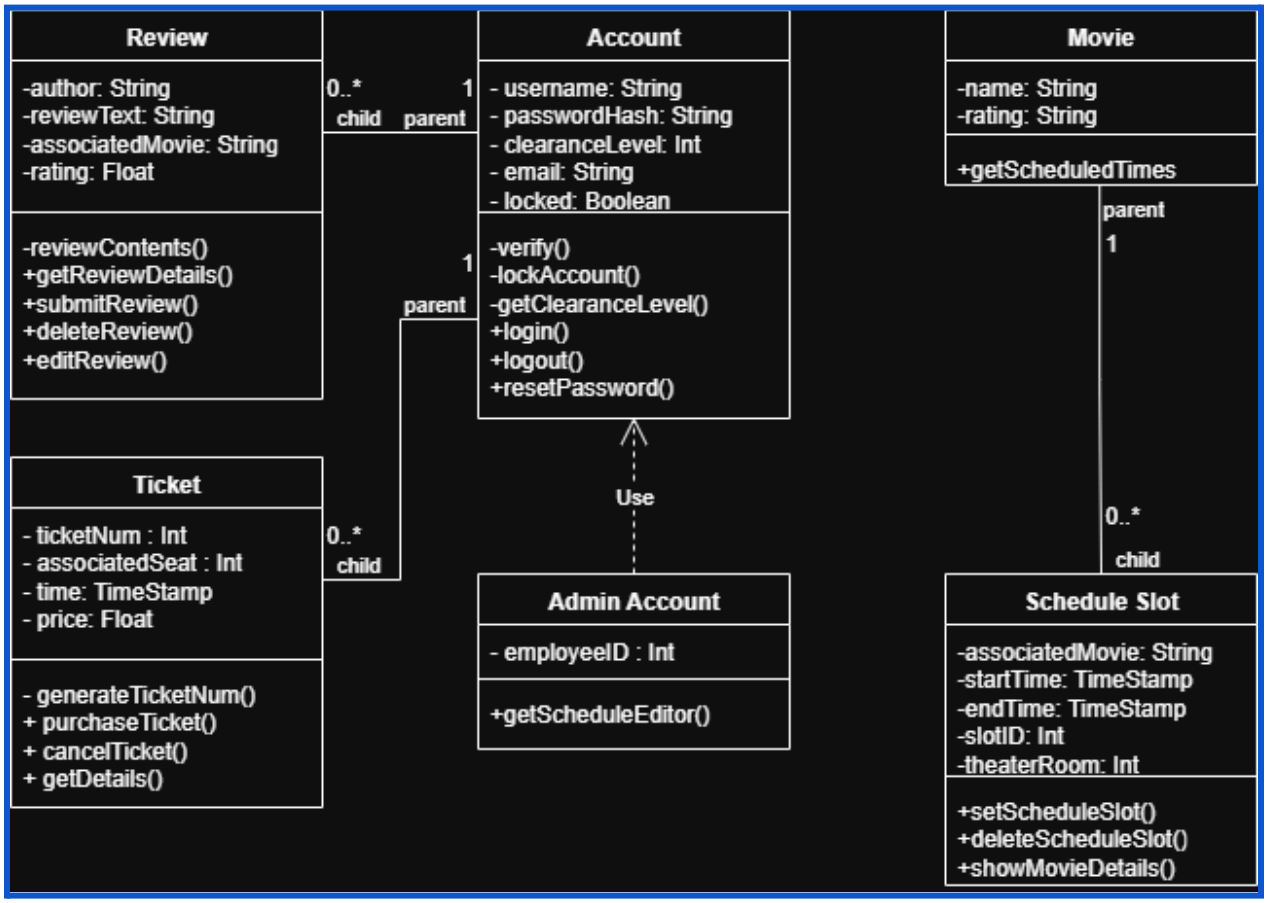
Prepared for
CS 250 - Introduction to Software Systems
Instructor: Gus Hanna, Ph.D.
Summer 2025

1. System Overview

ReelEasy is a secure, multi-platform movie theater ticketing system designed for SDSU's School of Theatre, Television, and Film. The system provides a modern way for students and community members to search movies, select seats, purchase tickets online, and manage their accounts. Administrative staff can update movie listings and manage show schedules through a secure admin portal.

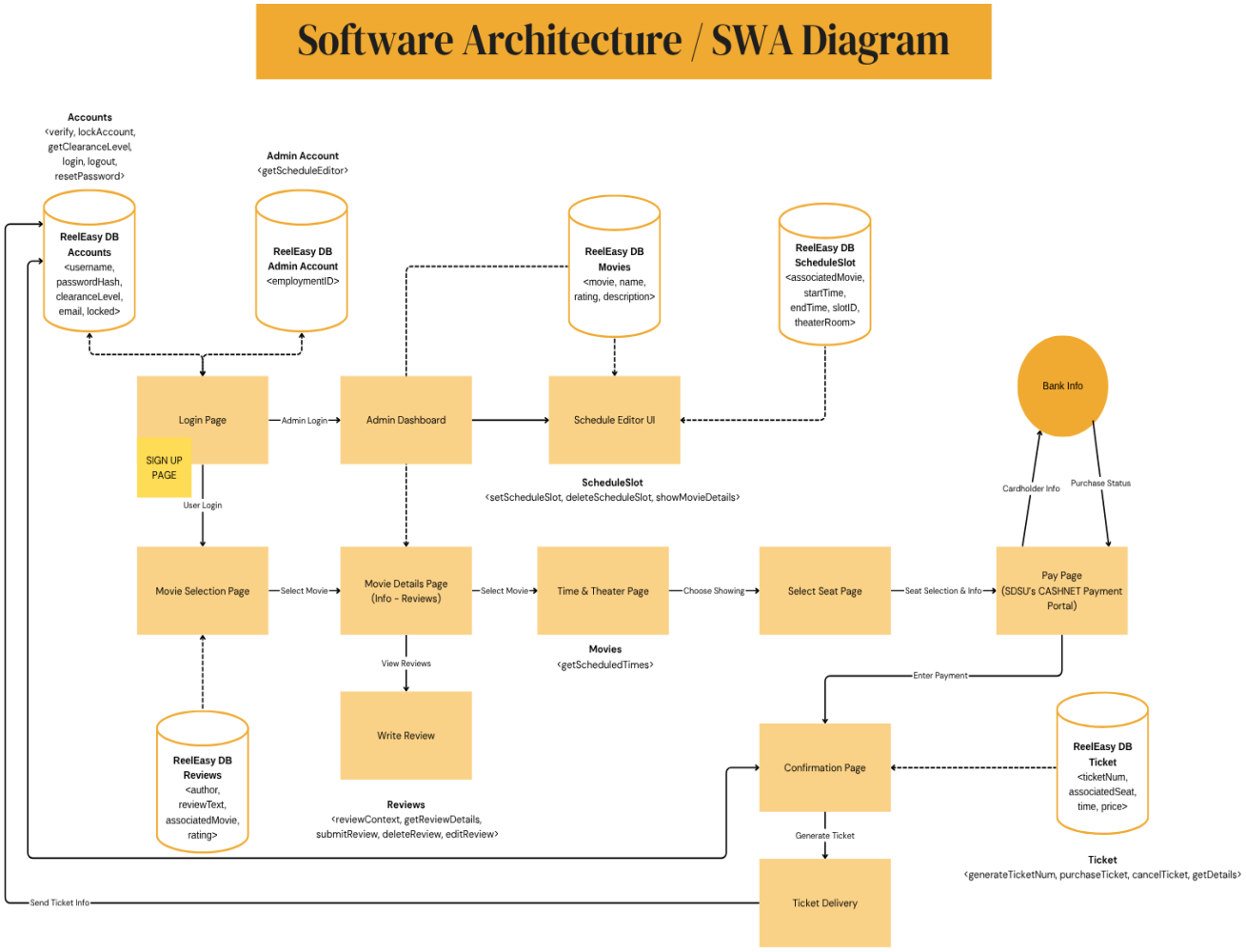
2. Software Architecture Overview

2.1 UML Class



> Include Description

2.2 ReelEasy Software Architecture



Hyperlink: [ReelEasy - Software Architecture Diagram]

2.2.1 Software Architecture Diagram

The ReelEasy Software Architecture Diagram (SWA) illustrates how the entire ticketing system's front-end user interfaces, admin tools, and back-end storage work together to deliver all required functionality for SDSU's School of Theatre, Television, and Film.

The diagram encompasses main components, attributes, and its functions in how they interact in tandem with one another through a single centralized database, which contains all the information needed to manage users, movies, show schedules, tickets, and reviews.

There is one centralized database called the “ReelEasy” database which has different subcategories for storing different information. This includes:

- Accounts: Contain the information and functions for a user to have a functional account in the app.
- Admin Account: Contains the information and functions for staffers to operate and edit the app.
- Movies: Contains the information and functions of movie listings and descriptions.
- ScheduleSlot: Contains the information and function for the location and times for the movie.
- Reviews: Contains the information and functions for users to access and write reviews on said movies.
- Ticket: Contains the information and functions for users to get their ticket and theater information.

2.2.1.1 Login / Sign Up

- Users start at the Login Page or Sign-Up Page.
- Credentials are verified with the ReelEasy Accounts DB, which stores usernames, hashed passwords, clearance levels (admin/user) and email addresses.
- Accounts can be locked automatically if multiple failed logins attempts occur

2.2.1.2 Movie Selection / Browsing

- Logged-in users can access the Movie Selection Page where they can browse all available movies.
- When a movie is selected, the system will then load the Movie Details Page, which shows the movie information retrieved from the Movie DB and also pull reviews from the Reviews DB.

2.2.1.3 Reviews

- On the Movie Details Page, users can view all reviews for their selected movie (getReviewDetails()). If logged in, users can submit (submitReview()), edit (editReview()), or

delete (deleteReview()) their own reviews.

- The reviewContent() allows the system to validate content to prevent inappropriate submissions.

2.2.1.4 Showtimes / Seat Selection

- After selecting a movie, users may proceed to the Time & Theater page where they will display available showtimes retrieved from the scheduleSlot DB.
- The select page allows for the user to choose an available seat for their chosen showtime.

2.2.1.5 Payment & Confirmation

- Pay page verifies payment details and processes the ticket purchase (purchaseTicket())
- The system would generate a ticket number for the user (generateTicketNum()), store it in the tickets table, and create an electronic ticket with the seat, the time, and price information.
- At the confirmation page, it shows the completed transaction, and the Ticket Delivery module sends the ticket details to the user's email or phone.

3. Development & Timeline

3.1 Team Responsibilities

The ReelEasy system is being built by four project teams, each responsible for a major area of the system:

- Front-End Team: Designs and develops all user pages, including login, movie selection, seat selection, payment, and ticket display.
- Back-End Team: Builds server logic and APIs for login, reviews, schedule slots, ticket generation, and database interaction.
- QA Team: Tests each feature (login, movie browsing, payments, etc.), reports bugs, and verifies fixes.
- Project Management: Plans the workflow, monitors deadlines, manages the staging environment, and oversees deployment.

All teams commit to GitHub and coordinate through weekly development meetings every Monday to review progress, plan the week's tasks, and address any issues.

3.2 Development Timeline

Phase: Planning
Dates: July 28 – August 2
Tasks: Define system goals, assign team roles, create GitHub repo, outline user and admin flows
Teams Involved: Project Management
Note: Weekly development meetings begin Monday, July 29

Phase: Design
Dates: August 3 – August 9
Tasks: Finalize database structure, UML/class diagrams, and wireframes for each page
Teams Involved: Front-End, Back-End

Phase: Development Part 1
Dates: August 10 – August 16
Tasks: Build login/signup pages, account verification, admin dashboard access, movie selection and review pages
Teams Involved: Back-End, Front-End

Phase: Development Part 2
Dates: August 17 – August 23
Tasks: Implement ScheduleSlot logic, Time & Theater page, and seat selection flow
Teams Involved: Back-End, Front-End

Phase: Development Part 3
Dates: August 24 – August 30
Tasks: Integrate payment flow (mock SDSU CASHNET), generate ticket, and implement confirmation and ticket delivery
Teams Involved: Back-End, Front-End

Phase: Staging

Dates: August 31 – September 2

Tasks: Push codebase to a staging server for full-system testing

Teams Involved: QA Team, All Teams

Phase: Testing

Dates: September 3 – September 4

Tasks: Conduct end-to-end testing, fix bugs, review performance and security

Teams Involved: QA Team, Back-End

Phase: Deployment

Dates: September 5 – September 6

Tasks: Final deployment to production environment, documentation finalized, and submission prepared

Teams Involved: Project Management, All

3.3 Tools and Workflow

- GitHub: Used for version control and collaboration
- Figma or Canva: Used for design mockups and architecture diagrams
- VS Code: Used as the code editor
- Postman: Used for testing API endpoints
- Discord, Zoom and Slack: Used for communication and weekly check-ins
- Weekly Development Meeting: Held every Monday to review progress and assign tasks
- Staging Server: Used to test the full system before final deployment
- SDSU Mock CASHNET Portal: Simulated payment system used for the Pay Page

Group 2 - Project Assignments

Name	Role	Projects	Contributions
Bobby Bavongkhoun	Developer / Design	Planning	Project planning
Caleb Wolf	Developer / Design	Implementation	Specifications
Gleb Rodin	Developer / Design	Research	Specifications