Design Document: MTS

1. Purpose

- This document will describe the architecture of the MTS and explain its functionality and the design decisions that were made in its creation. It is intended for the client and outlines the requirements, the design implementation, validation and potential modifications that could be made in the future. As this is intended for the Client it will not be as detailed, refer to the SRS for more detail on any part of this document.

2. Client's Requirements and Validation

Client Requirements

1. To provide a user friendly interface for booking and viewing movies.
2. Support user registration, authentication and account management.
3. Include seat selection, payment processing and booking management features.
4. Allow system admin to manage showtimes, movies and tickets.
5. Ensure and plan for system reliability, scalability and security.

Validation:

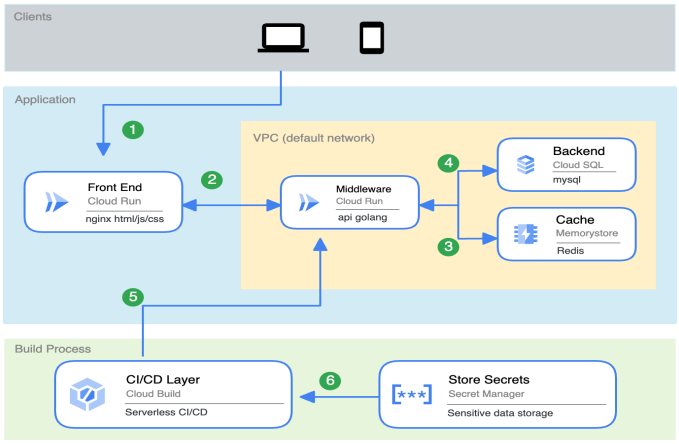
- All requirements were validated by creating functional prototypes, conducting testing and responding to and implementing client feedback.

3. System Design Overview

3.1 System Architecture.

The MTS is has a 3 tier design architecture:

1. **Front End:**
    a. Built using HTML, CSS and JavaScript
    b. Robust interphase for both desktops and mobile devices.
2. **Backend:**
    a. RESTful APIs built using Node.js and Express.js
    b. Is used to handle business logic, user authentication and interactions in the database.
3. **Database**
    a. A relational database (MySQL) to store structured data formatted including users, bookings and movies.
    b. Redis caching layer for performance optimization.



4. Verification and Validation

1. **Testing**:
    - Functional Testing: Verify each feature (e.g., ticket booking).
    - Performance Testing: Ensure system stability under peak loads.
    - Security Testing: Check for vulnerabilities in authentication and data handling.
2. **Validation**:
    - User feedback sessions.
    - Review sessions with the client.

5. Security Measures

- **Authentication**:
    - Password hashing with bcrypt.
    - Multi-factor authentication (2FA).
- **Data Protection**:
    - HTTPS for secure communication.
    - Encrypt sensitive user and payment data.
- **Access Control**:
    - Role-based access (users vs. admins).

6. Potential Modifications and Future Enhancements

1. **Mobile App**: Extend functionality to a dedicated mobile app.
2. **Loyalty Program**: Add reward points for frequent users.
3. **AI Recommendations**: Suggest movies based on user preferences and history.
4. **Multilingual Support**: Enable translations for non-English-speaking users.
5. **Bug Fixes:** As users encounter bugs, there will be updates to fix them.

7. Threats and Countermeasures (CIA framework)

Category	**Threat**	**Countermeasure**
User Confidentiality	Data Breaches	Use of Encryption, and access controls.
Integrity	Unauthorized data modification.	Role-Based access and input validation
Availability	DDoS Attacks	Implement rate limiting and firewalls