BUKIDNON STATE UNIVERSITY

COLLEGE OF TECHNOLOGIES

INFORMATION TECHNOLOGY DEPARTMENT

CineMatch: Movie Recommendation App

[VERSION NUMBER]

[DATE AND REVISION HISTORY]

By:

Name	Roles	Contact Information
Baldivino, Reynante O.	Lead Developer	09268100382
Gallardo, Jessa Y.	Documentation	09945286529
Lagare, Angel Mae L.	Project Manager	09979258119

Table of Contents

1. Document Overview

1.1 Scope

The CineMatch project covers many important parts of creating a mobile application. It includes the design of the app, the features it offers, and how it works on Android devices. The project also explains the tools used to build the app, such as Firebase for saving data and TMDb for movie information. Testing is part of the process to make sure the app works smoothly and gives users a good experience. However, creating an iOS version and promoting the app to the public are not part of this project.

This document explains how users can log in using a Google account, view recommended movies, search for titles, and watch trailers. It also describes how the app helps users save or bookmark their favorite movies for easy access later. The design is simple, clear, and easy to use, following Android's standard look and feel. Movie information is updated using trusted sources, so users always see the latest details. The app was built with the goal of making movie searching faster and more personalized.

1.2 Audience

The intended readers of this document include instructors, students, and anyone interested in how a mobile app is made. Developers can use the information to understand the app's features and how it was created. Designers can see how each screen was planned and organized. Testers can learn what parts of the app need to be checked to make sure everything works. Future students may also find this document useful when creating their own mobile app projects.

2. Project Overview

2.1 Executive Summary

CineMatch is a mobile app designed to make choosing a movie easier, faster, and more enjoyable. Many people today spend too much time scrolling through endless options on streaming platforms without finding something they truly want to watch. CineMatch solves this by giving personalized movie suggestions based on the user's interests, past choices, and favorite genres. The app creates a more focused experience by helping users quickly find films they're more likely to enjoy. It is built for Android devices and uses simple screens, clean layouts, and easy navigation to ensure that anyone can use it without confusion. This app brings together movie summaries, trailers, and search features all in one place, making it convenient and efficient. Users can also save their favorite movies to revisit them later.

The recommendation system is based on real user behavior, so it feels more accurate and tailored than general movie apps. CineMatch uses trusted sources for movie information and trailers, so users always see up-to-date content. The goal is not only to entertain, but also to make movie discovery a better and smarter experience for users of all ages, especially students, young adults, and film lovers who often want fast, quality suggestions.

Objectives

One of the main goals of CineMatch is to help users avoid wasting time when trying to choose a movie. By offering suggestions based on personal preferences, the app makes sure that the recommendations match what the user is actually interested in. Another goal is to provide a simple and clean design that users can easily understand, even if they are using the app for the first time. The app also aims to connect with well-known movie databases and services to provide accurate and current movie details. Making the user experience smooth, fast, and stress-free is an important part of this objective. Also to create a platform that supports safe and secure login through Google, which helps users feel comfortable saving their favorites. The app should also be able to load content quickly and respond well, even when many people are using it. It is also designed to work well on different types of Android devices, including both phones and tablets. CineMatch supports educational and personal development by helping users learn how recommendation systems work.

Main Features

CineMatch includes many helpful features that work together to create a smooth and enjoyable experience. The app starts by letting users sign in using their Google account, which makes the login process fast, secure, and easy. After logging in, users are welcomed with a home screen filled with recommended movies that match their viewing history and selected genres. A search feature allows users to quickly find specific movies by title or by selecting a category, such as action, romance, or comedy. This helps users who already have a movie in mind as well as those just exploring. Each movie comes with important details like a short summary, the rating, genre, and cast members, so the user can decide if it's worth watching. CineMatch also lets users watch trailers without leaving the app by using YouTube integration. This saves time and gives users a better idea of what the movie looks like before watching. If the user likes a movie, they can save it to a favorites list for future viewing. All of these features are designed to work together so the user can find, explore, and enjoy movies with as little effort as possible.

2.2 Problem Statement

Many users spend too much time trying to find the right movie to watch. Existing movie apps often show suggestions that do not match the user's taste or are not personalized. This leads to frustration and wasted time, especially when users are tired or unsure of what to pick. CineMatch solves this problem by giving recommendations that are based on the user's actual preferences and previous choices. This makes it easier to discover enjoyable movies without the usual hassle. In addition to the need for better recommendations, users also want simple and quick access to movie details and trailers. Some apps make users switch between different platforms just to see a trailer or learn more about a film.

User Needs

Users are looking for a movie app that understands their personal preferences and viewing habits. They want an app that can suggest films that feel right for their mood, without needing to scroll endlessly. A good recommendation system should learn from what users like and offer movie choices that match their favorite genres, past selections, and ratings. Users also expect an app to be quick, responsive, and easy to use, especially when they are short on time. Convenience, simplicity, and relevance are the most important things users look for in a movie app.

Beyond just recommendations, users also want access to helpful features in one place. They prefer to watch trailers, view cast details, and read short summaries without leaving the app. A strong user need is the ability to bookmark favorite movies and return to them later. Many also want to explore new movies based on what they already enjoy, not random or trending content. Overall, users need a smart, easy, and personal experience that helps them enjoy more movies and spend less time searching.

Market Analysis

The main users of CineMatch are students, movie lovers, and young adults who often use mobile apps to find entertainment. These users usually look for fast and accurate recommendations because they don't want to waste time scrolling through long lists of random movies. While other apps like Netflix or IMDb offer recommendations, they are not always personal, and they don't combine all features like login, search, trailers, and bookmarking favorites in a single, easy-to-use app. Some apps also require too much setup or show content that does not match the user's taste, leading to a poor experience.

CineMatch stands out by being simple, smart, and focused on the user's actual preferences. It uses data from trusted movie sources to keep content updated and relevant. The app also offers a cleaner layout and a smoother experience than many existing apps, especially those that are filled with ads or confusing options. Its main strength is giving personalized movie suggestions while keeping the app light and easy to use. With more people looking for convenient ways to choose movies, CineMatch fits perfectly into the growing need for smarter entertainment tools.

3. Functional Specifications

3.1 Feature list

Login with Google

This feature allows users to easily and securely log into the app using their existing Google account. It removes the need to create a new username and password, saving time and effort for the user. Logging in with Google also helps the app personalize movie suggestions based on user identity and preferences. It provides a seamless way to start using the app with just a few taps. This feature also helps keep user data protected and organized within a trusted platform.

User Flow

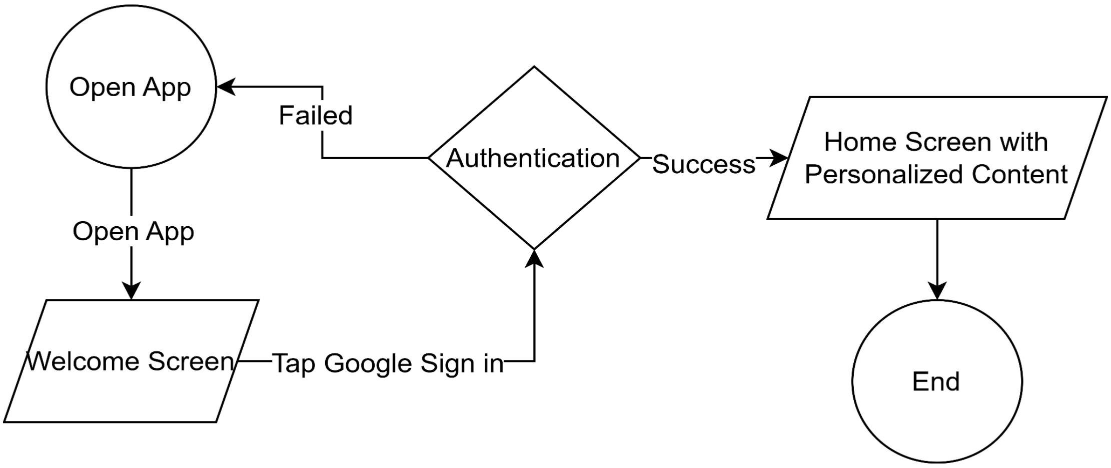

Figure 1: User Flow Login With Google

Use Case

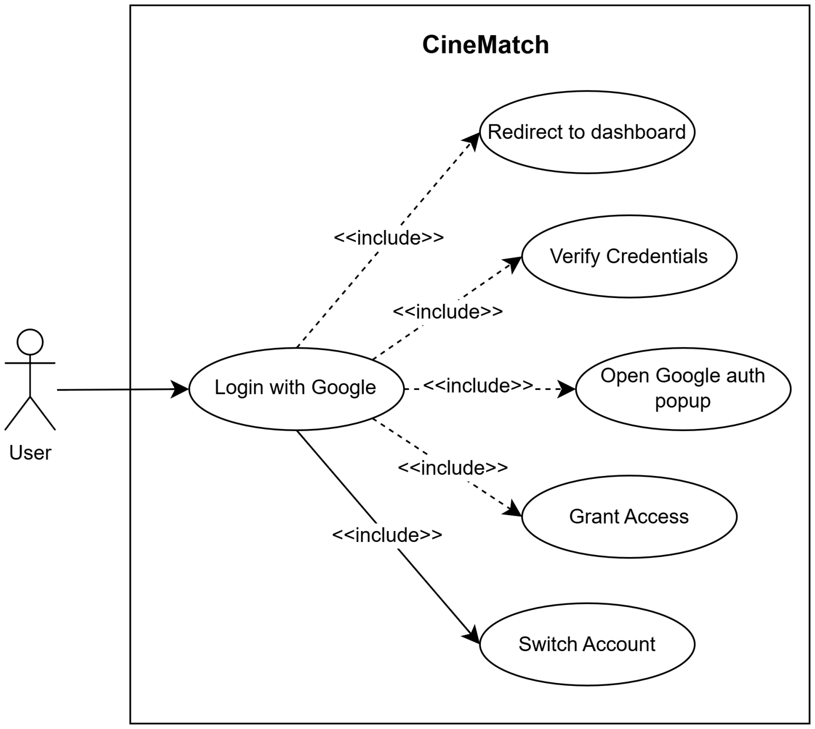

Figure 2: Primary Flow Login With Google

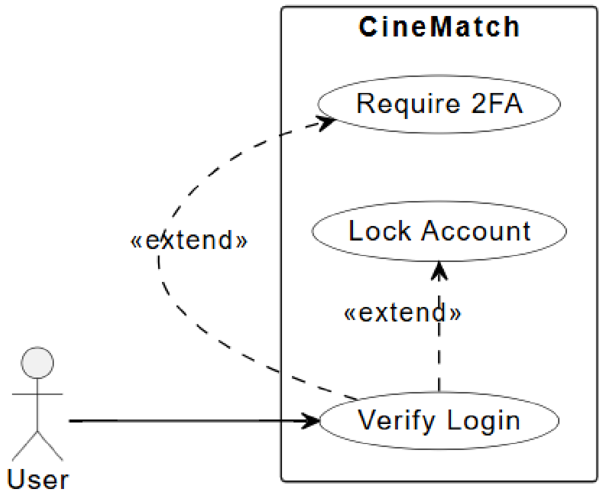

Figure 3: Alternative Flow Login With Google

Personalized Movie Recommendation/ Suggestion

After logging in, users are shown a list of movies specially selected based on what they like. These suggestions are not random; they're picked using the user's viewing history, favorite genres, and saved bookmarks. This feature helps users quickly find something they're more likely to enjoy, instead of scrolling endlessly through unrelated titles. The recommendations update regularly to give users new options each time they use the app. It is the heart of CineMatch, designed to save time and improve the overall movie-picking experience.

User Flow

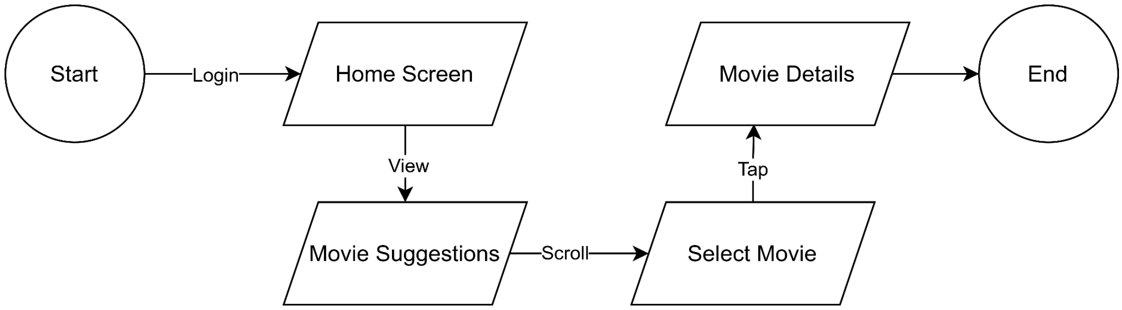

Figure 4: User Flow Personalized Movie Recommendation

Use Case

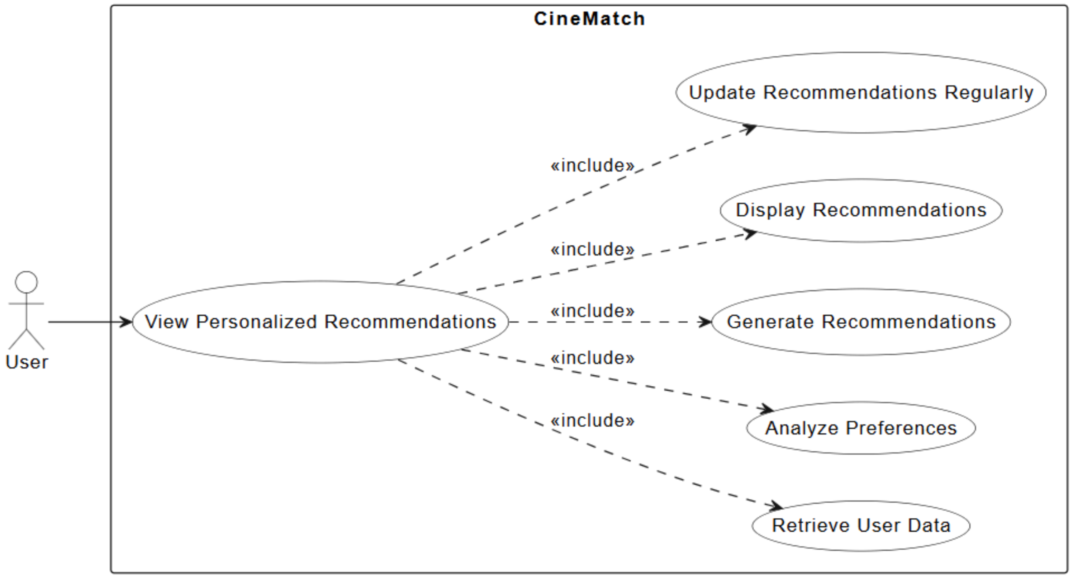

Figure 5: Primary Flow Movie Recommendation

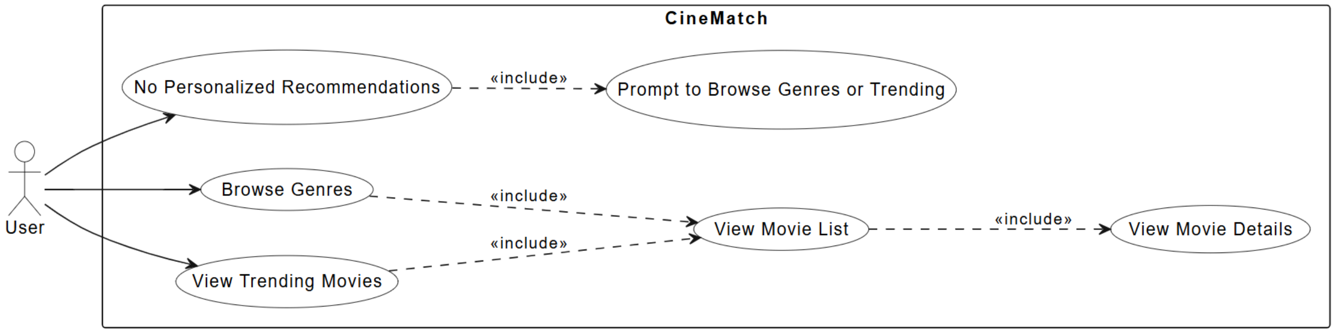

Figure 6: Alternative Flow Movie Recommendation

Movie Search

The search feature allows users to find specific movies by typing the title, a keyword, or even a genre. It is useful when the user already has a movie in mind or wants to explore something related to a certain theme or topic. The search results appear instantly as the user types, making it quick and responsive. This feature makes it easy to look up older films, new releases, or even movies based on mood. It helps users take control of what they want to watch instead of relying only on suggestions.

User Flow

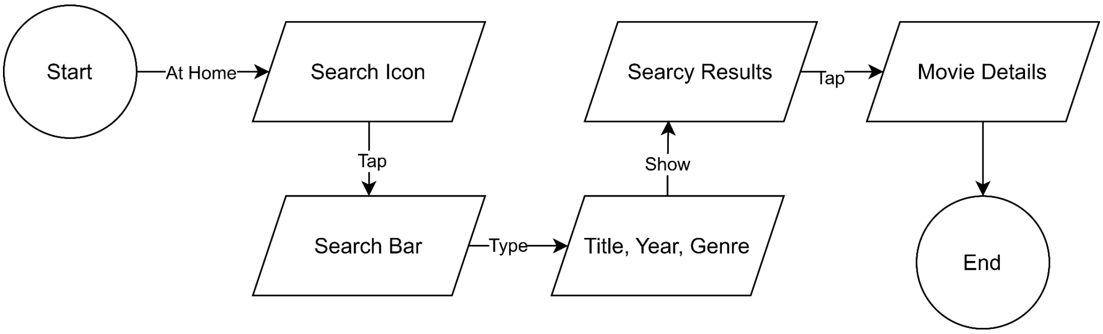

Figure 7: User Flow Movie Search

Use Case

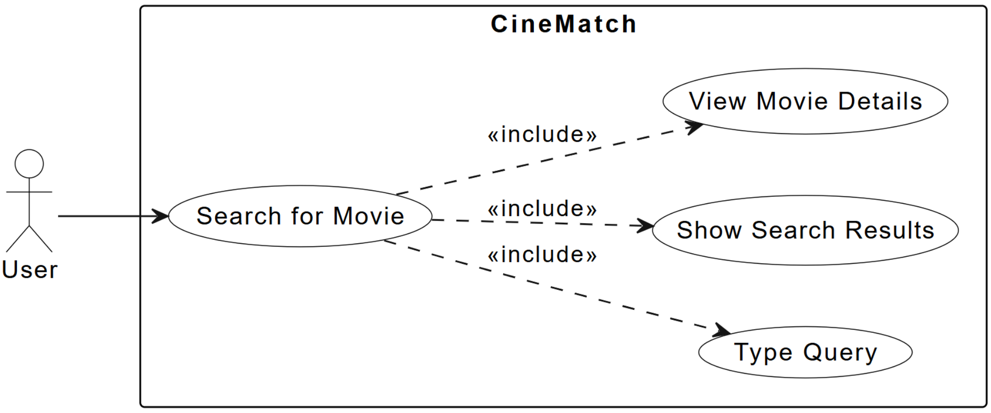

Figure 8: Primary Flow Movie Search

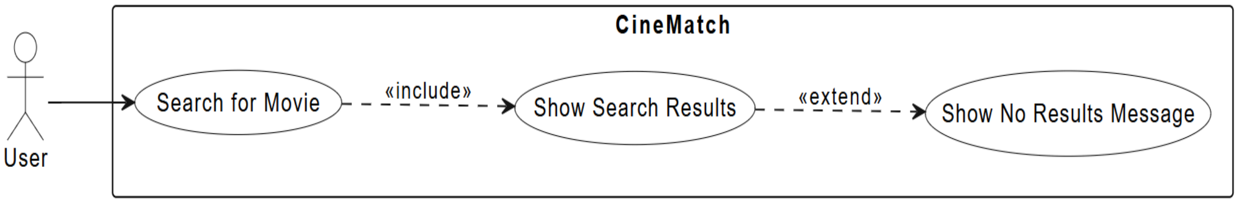

Figure 9: Alternative Flow Movie Search

Movie Details

Each movie listed in CineMatch includes a detailed page that shows everything the user needs to know before deciding to watch. This includes a short plot summary, the movie's rating, genre, cast, release year, and more. It helps the user understand the story, tone, and main characters without needing to search elsewhere. This saves time and helps with decision-making, especially for users who want to know more before committing to a film. It also gives quick access to related features like bookmarking and trailers.

User Flow

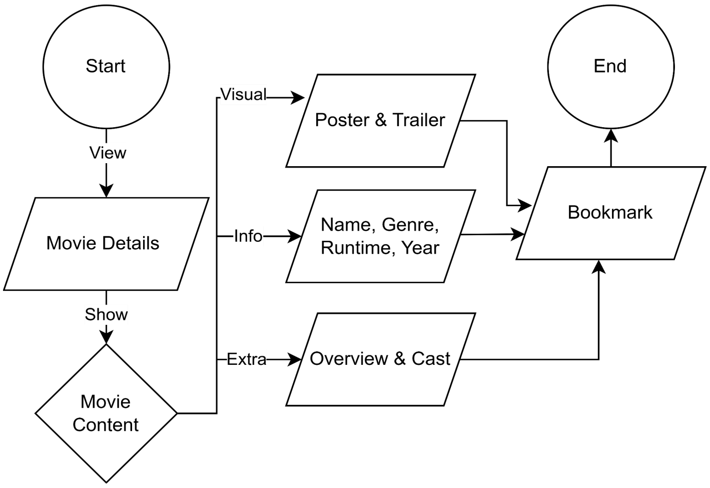

Figure 10: User Flow Movie Details

Use Case

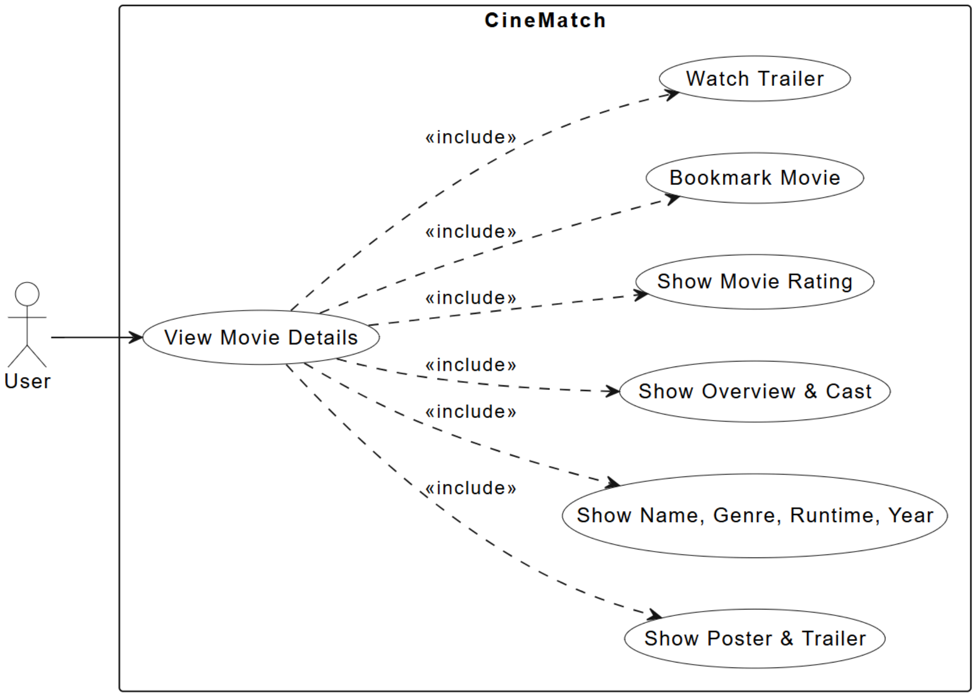

Figure 11: Primary Flow Movie Details

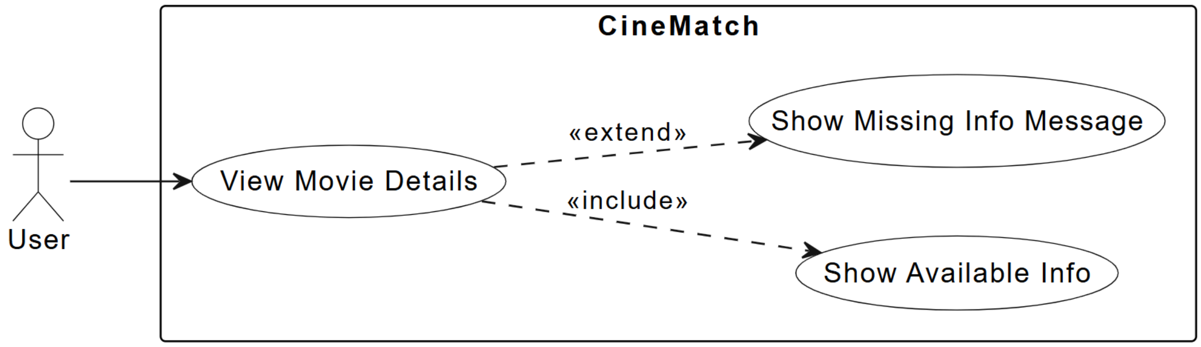

Figure 12: Alternative Flow Movie Details

Watch Trailers

CineMatch lets users watch movie trailers directly within the app, using a built-in YouTube player. This means there's no need to switch to another app or browser just to see what a movie looks like. The trailer feature gives users a better sense of the film's tone, visuals, and storyline before they watch the full movie. It also adds excitement and helps users decide whether to bookmark the film. It's quick, convenient, and works smoothly within the app.

User Flow

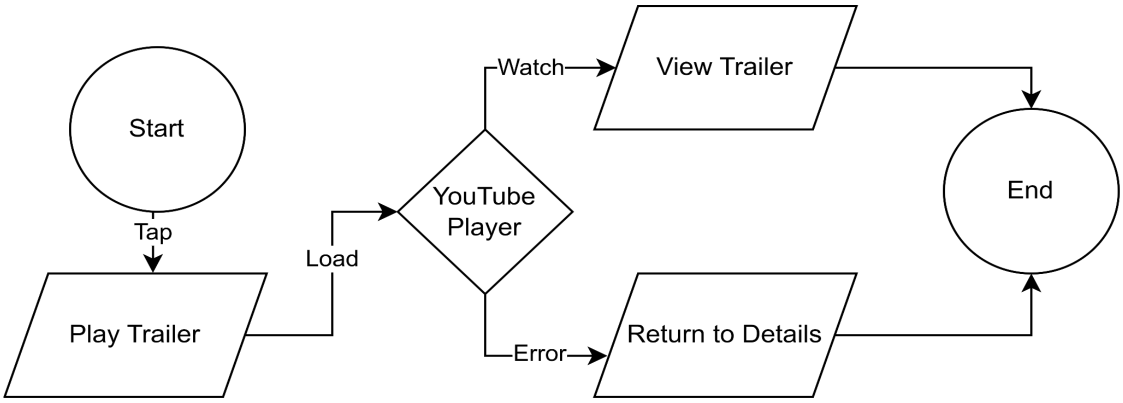

Figure 13: User Flow Watch Trailers

Use Case

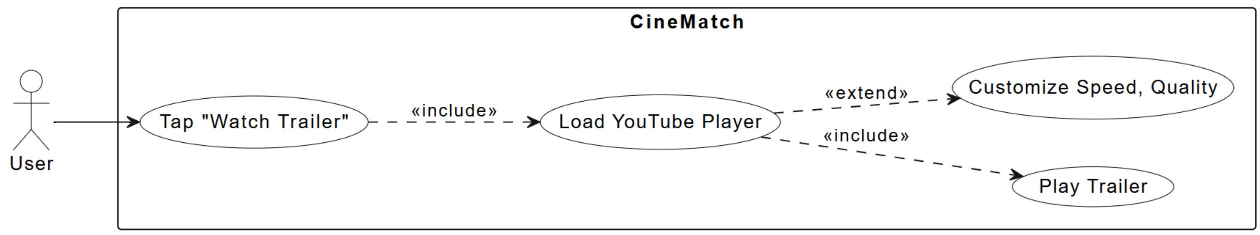

Figure 14: Primary Flow Watch Trailers

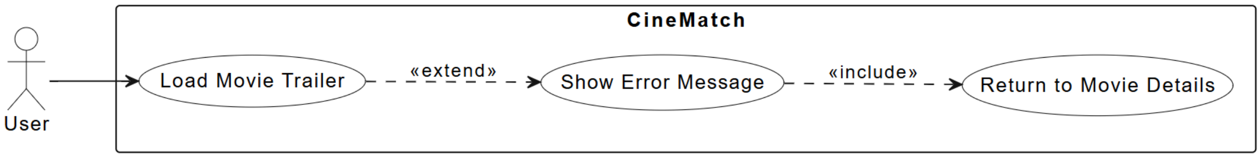

Figure 15: Primary Flow Watch Trailers

Bookmark Movies

This feature allows users to save the movies they like or plan to watch later by tapping a simple bookmark icon. It's helpful for users who want to keep track of interesting titles without having to search for them again. Bookmarked movies are stored in a separate list for easy access anytime. Whether users are planning a movie night or just collecting favorites, this feature keeps everything organized. Bookmarks are saved securely and stay available as long as the user is logged in.

User Flow

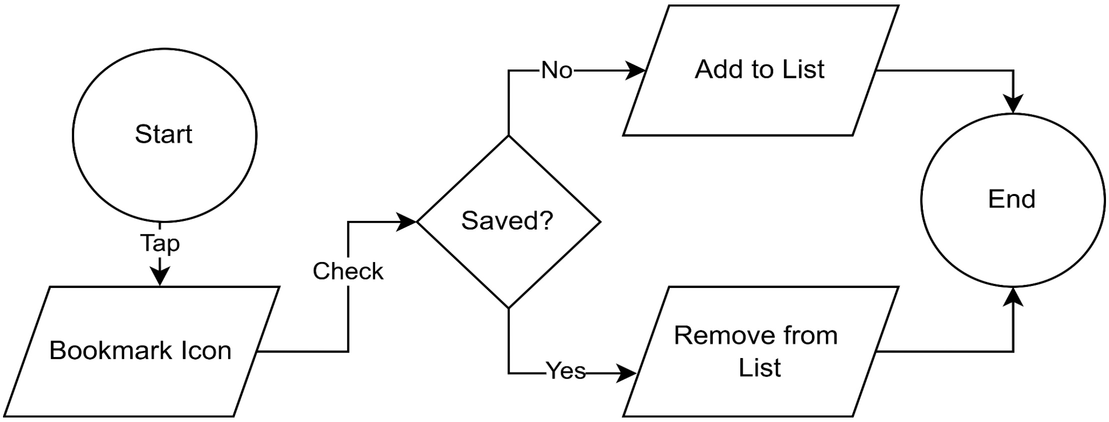

Figure 16: User Flow Bookmark Movies

Use Case

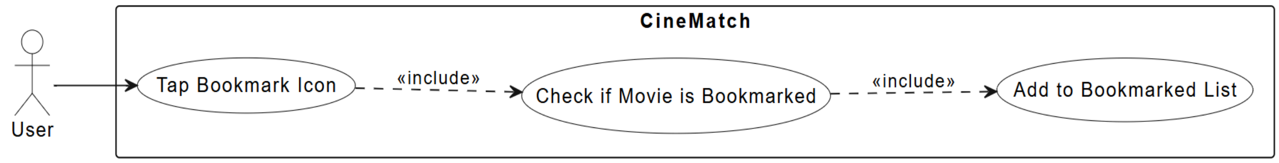

Figure 17: Primary Flow Bookmark Movies

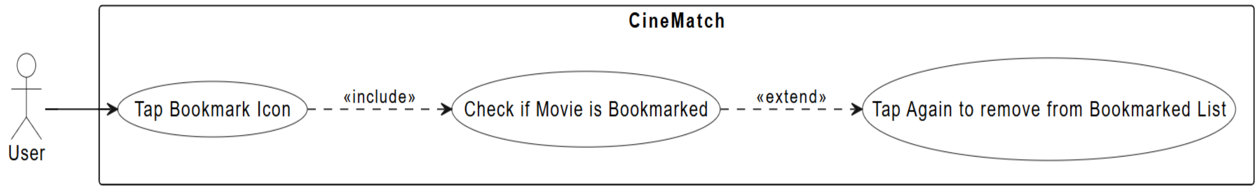
CineMatch
User
Tap Bookmark Icon
«include»
Check if Movie is Bookmarked
«extend»
Tap Again to remove from Bookmarked List

Figure 18: Alternative Flow Bookmark Movies

View Bookmarked List

Users can view all the movies they have bookmarked in one dedicated section of the app. This feature makes it easy to revisit saved movies without needing to remember their titles or search again. Each bookmarked movie can be tapped to see details, play the trailer, or decide to watch it. It acts like a personal movie shelf, helping users plan future watchlists or remember past favorites. The list can grow and update as users continue to explore the app.

User Flow

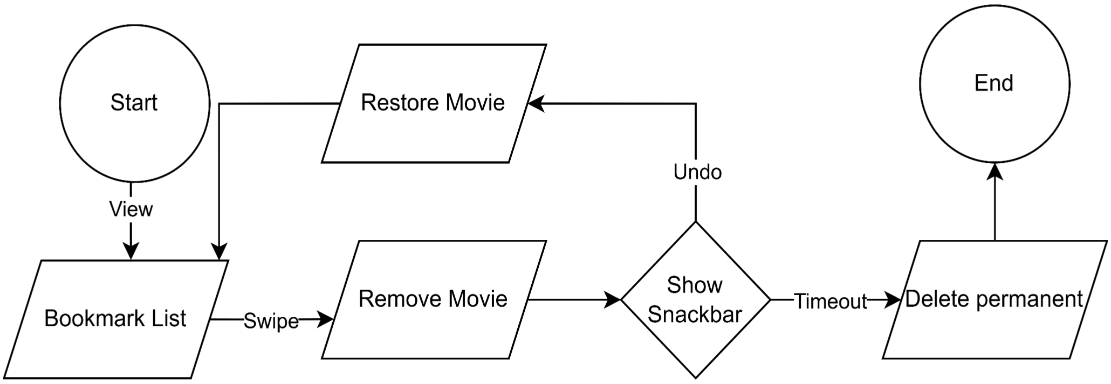

Figure 19: User Flow View Bookmarked List

Use Case

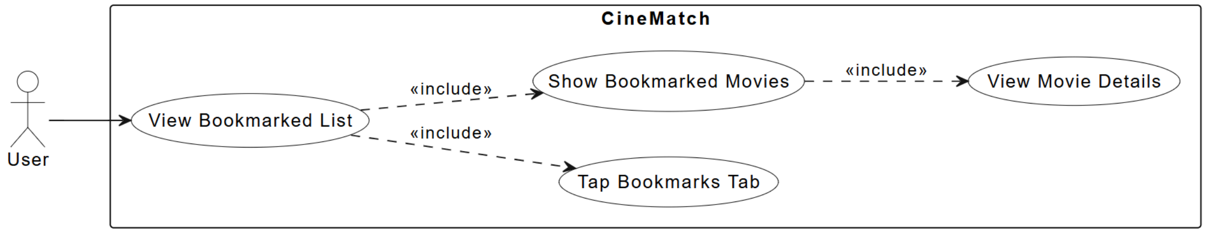

Figure 20: Primary Flow View Bookmarked List

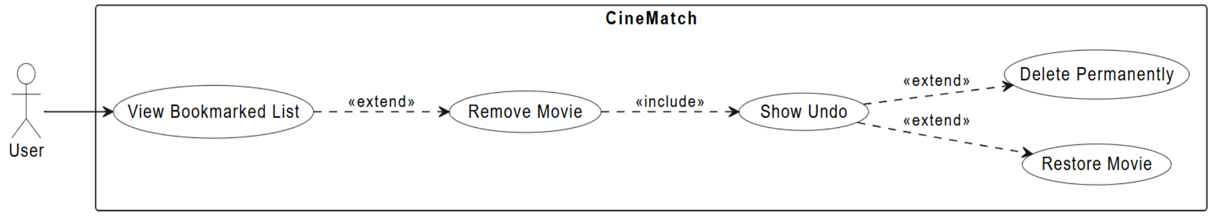

Figure 20: Alternative Flow View Bookmarked List

Profile Customization

Users can manage their personal profile information in a dedicated settings section of the app. This feature allows users to update their profile picture and username, while also displaying their email address for reference. The settings page serves as a personal hub where users can maintain their account information. Profile changes are saved automatically and reflected across the app, ensuring a consistent user experience. The interface is designed to be intuitive, making it easy for users to keep their profile information up to date.

User Flow

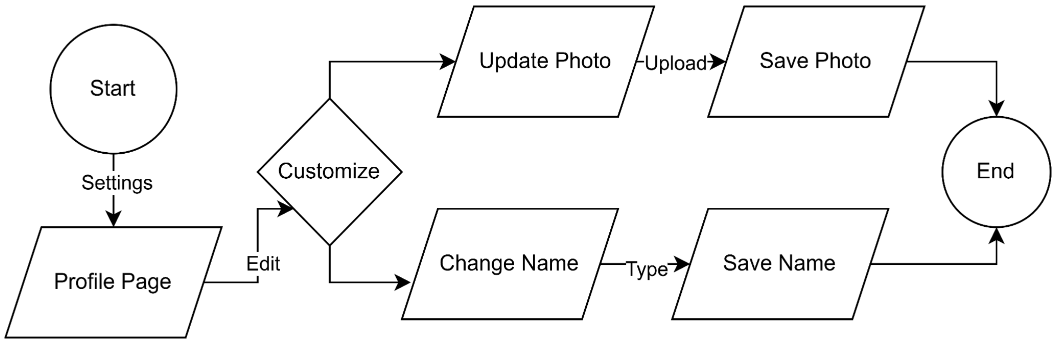

Figure 22: User Flow Profile Customization

Use Case

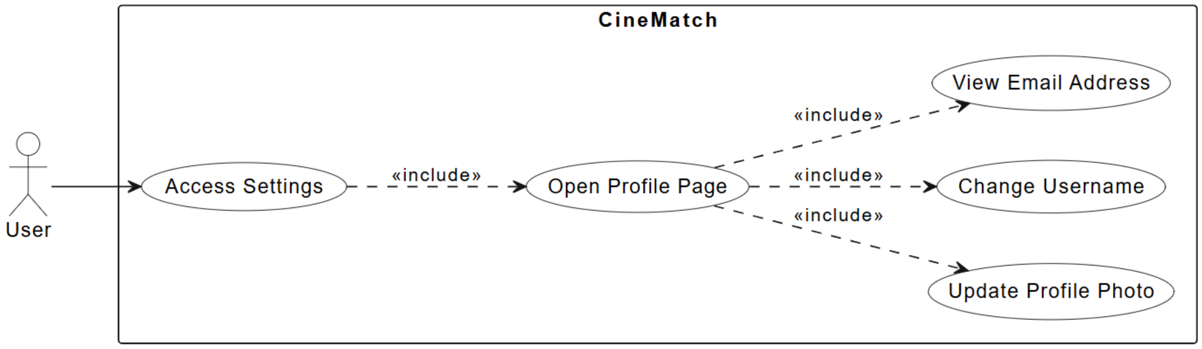

Figure 23: Primary Flow Profile Customization

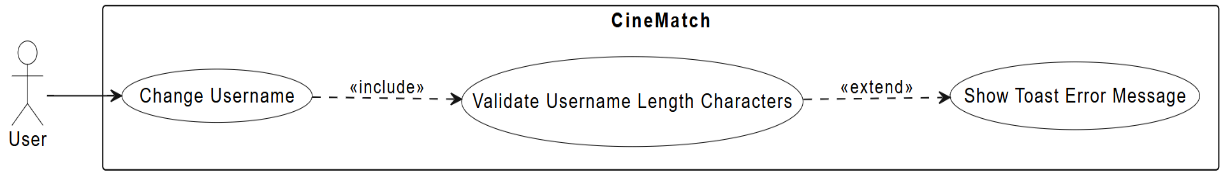

Figure 24: Alternative Flow Profile Customization

3.2 User Stories & Requirements

User Stories

1. **Google Login**

- "As a user, I want to log in with my Google account so that I can quickly access the app without creating new credentials."

- “As a user, I want to see my Google profile information so that I can verify I'm using the correct account.”
- “As a user, I want to be able to switch Google accounts so that I can use the app with different profiles.”
- “As a user, I want to maintain my login session so that I don't have to log in repeatedly.”

2. **Personalized Movie Recommendations**
   - "As a user, I want to filter movies by genre so that I can find movies matching my current preferences."
   - “As a user, I want to view details of recommended movies so that I can decide if I want to watch them.”
3. **Movie Search**
   - "As a user, I want to search for movies by exact title so I can quickly find specific films I'm looking for."
   - "As a user, I want to filter search results by release year so I can find movies from a particular time period."
   - "As a user, I want to browse movies by genre so I can discover films that match my current mood."
4. **Movie Details**
   - "As a user, I want to see complete movie information (title, genre, language, runtime, release year, and rating) displayed prominently so I can evaluate it at a glance."
   - "As a user, I want to read a concise overview so I can understand the movie’s story without spoilers."

- "As a user, I want to view the cast list so I can recognize actors I enjoy."
- "As a user, I want to see the movie poster and trailer so I can visually assess its style and quality."

5. **Watch Trailers**
    - "As a user, I want to play trailers instantly within the app so I don’t get redirected to YouTube or other platforms."
    - "As a user, I want the trailer to play with essential controls (play/pause, skip forward/back, mute, subtitles) so I can watch comfortably."
    - "As a user, I want the option to lower video quality when my internet is unstable so I can watch without buffering."
6. **Bookmark Movies**
    - "As a user, I want to bookmark movies from the details page so I can save them for later viewing."
    - "As a user, I want to see visual confirmation (e.g., icon change) when I bookmark a movie so I know it was saved successfully."
    - "As a user, I want to remove bookmarks with one tap so I can keep my list up to date."
7. **Bookmarked List**
    - "As a user, I want to access all my bookmarked movies in a dedicated tab so I can manage my watchlist in one place."
    - "As a user, I want to tap any bookmarked movie to view its details or trailer so I can decide to watch it now or later."

- "As a user, I want my bookmarks to sync across devices so I can access them anywhere."
- "As a user, I want to unbookmark movies directly from my saved list so I can keep it relevant."
- "As a user, I want to unbookmark movies from the details page so I can remove them without returning to my list."

8. **Profile Customization**
   - "As a user, I want to update my profile picture so I can personalize my account."
   - "As a user, I want to edit my display name or username so I can represent myself how I prefer."

Acceptance Criteria

1. Google Login
   - User is redirected to Google's OAuth screen when 'Sign in with Google' is tapped.
   - App displays the user's Google profile name/email on the settings page after login.
   - 'Switch Account' option appears in settings, triggering Google's account picker.
   - Login session persists for 30 days unless manually logged out.
2. Personalized Movie Recommendations
   - Homepage shows at least 5 recommendations based on user's watch history/genres.
   - Genre filter updates results instantly when toggled (e.g., 'Comedy').
   - Each recommendation card displays title, poster, and rating (★4.2).
3. Movie Search
   - Genre tags ('Horror') appear below the search bar for quick filtering.

- Users can search by full or partial movie titles ('Aveng' shows 'Avengers').
- Genre chips ('Action', 'Romance') appear below the search bar.
- Shows 'No results' if the search query returns no matches.

4. Movie Details

- Details page shows: Title, Genre, Language, Runtime, Year, Rating.
- Cast section displays 5+ actors with character names and thumbnails.
- Trailer thumbnail plays full trailer in-app when tapped.

5. Watch Trailers

- Trailer starts within 2 seconds with controls: Play/Pause, ±10s Skip, Mute, Subtitles.

6. Bookmark Movies

- Bookmark icon toggles between outlined (unsaved) and filled (saved) states.
- Toast message: "Added to Bookmarks"

7. Bookmarked List

- 'My Bookmarks' tab loads all saved movies within 1 second.
- Bookmark removal shows "Undo" toast (5s timeout).

8. Profile Customization

- Upload from gallery or camera (supports JPG/PNG).
- Editable field (3 to 15 characters maximum).
- Shows connected Google account email (read-only).

Non-functional Requirements

1. Performance Requirements

- Response time: "Movie search results must load within 1.5 seconds"
- API latency: "API endpoints must respond within 500ms"

   - Page load time: "Initial page load must complete within 2 seconds"

2. Security Requirements
   - User authentication: "Secure user authentication using JWT tokens"
   - API security: "All API endpoints must be protected against CSRF attacks"
   - Input validation: "All user inputs must be sanitized to prevent XSS attacks"
3. Reliability Requirements
   - Availability: "Application must maintain 99.5% uptime"
   - Data consistency: "User preferences must be consistently saved across sessions"
   - Error handling: "Error handling for API failures"
   - Recovery: "Automatic recovery from database connection issues"
4. Scalability Requirements
   - User scaling: "System must handle 10,000 concurrent users"
   - Database scaling: "Database must support growing movie catalog"
   - Cache management: "Implement caching for frequently accessed movie data"
   - Load balancing: "Distribute traffic across multiple servers"

5. Maintainability Requirements
   - Code organization: "Modular architecture for easy feature additions"
   - Documentation: "API documentation using Swagger/OpenAPI"
   - Testing: "80% code coverage for critical components"
   - Logging: "Comprehensive logging for debugging and monitoring"
6. Usability Requirements
   - Error messages: "User-friendly error messages for failed operations"
   - Loading states: "Clear loading indicators for async operations"
7. Compatibility Requirements

- Browser support: "Support for Chrome, Firefox, Safari, and Edge"
- Mobile support: "Responsive design for iOS and Android"
- API compatibility: "RESTful API with versioning support"
- Third-party integration: "Support for movie database APIs"

4. Technical Specifications

4.1 Architecture

- Overview Diagram: A high-level system architecture diagram that outlines components such as the mobile client, backend services, APIs, databases, and third-party integrations.
- Design Patterns: Discuss selected patterns (e.g., MVVM, MVC, VIPER) and explain their roles in organizing code.

4.2 Platform-Specific Considerations

iOS and Android Guidelines

1. iOS Guidelines:

UI/UX Standards:

- Follow Apple's Human Interface Guidelines
- Use iOS-specific navigation patterns
- Implement iOS-style gestures and animations
- Adhere to iOS typography and spacing guidelines
- Support both portrait and landscape orientations

Design Elements:

- Use SF Pro or system fonts
- Follow iOS spacing conventions
- Implement iOS-style tab bars and navigation
- Support iOS dark mode
- Use iOS-specific UI components

2. Android Guidelines:

UI/UX Standards:

- The app follows Material Design principles with custom styling
- Uses Android-specific components like:
- NestedScrollView for scrollable content
- ConstraintLayout for flexible layouts
- TabLayout for content organization
- BottomNavigationView for main navigation
- Custom color scheme with dark theme support
- Portrait orientation enforced for certain activities (Privacy, Help, About)

Design Elements:

- Custom fonts (Argentum Sans)

- Consistent padding and margins (16dp, 24dp)
- Standard text sizes (14sp, 16sp, 18sp, 24sp)
- Custom tab styling with dark background
- Responsive layouts using weight and constraints

Hardware & OS Compatibility

Android Compatibility:

- OS Requirements:
- Minimum SDK: 24 (Android 7.0 Nougat)
- Target SDK: 34 (Android 14)
- Compile SDK: 34
- NDK Version: 25.1.8937393

Device Requirements:

- Compatible with devices running Android 7.0 and above
- Supports both phones and tablets
- Optimized for portrait orientation
- Requires internet connectivity for movie data and authentication

Performance Considerations:

- Memory management optimized for Android
- Efficient image loading and caching
- Background processing for data fetching

- Firebase integration for authentication and data storage

iOS Compatibility

OS Requirements:

- iOS 13.0 and above
- Supports iPhone and iPad
- Optimized for both portrait and landscape orientations

Device Requirements:

- Compatible with iPhone and iPad devices
- Supports various screen sizes and resolutions
- Requires internet connectivity
- Supports both light and dark mode

Performance Considerations:

- Efficient memory management
- Optimized image loading
- Background processing capabilities
- Firebase integration for cross-platform consistency

4.3 Data Management

- Data Flow Diagrams: Show how data moves between the mobile app, local storage, and back-end systems.z
- Database Schemas: For both local and server-side databases.
- API Endpoints: List and describe RESTful or GraphQL endpoints, including sample requests and responses.

4.4 Security & Privacy

This framework adheres to industry standards (Firebase, OAuth 2.0, GDPR) and ensures user data is secure, minimal, and transparently managed.

Authentication & Authorization

Authentication Standards

- **Primary Provider**: Firebase Authentication.
- **Supported Methods**:
    - Email/Password authentication.
    - Google Sign-In (OAuth 2.0).
    - JWT-based authentication via Firebase Auth tokens.
- **Session Management**:
    - Firebase Auth state persistence.
    - Automatic token refresh.

Implementation Details

- Firebase Authentication SDK integration.
- Google Sign-In with secure token handling.
- Secure credential management (e.g., hashed passwords).
- Password reset functionality.
- Account creation with email verification.

Data Encryption

In-Transit Encryption

- All API communications use **HTTPS/TLS**.
- Firebase services enforce encrypted connections.
- Secure WebSocket connections for real-time updates.

At-Rest Encryption

- Firebase Firestore and Storage data encryption (AES-256).
- Secure local storage (e.g., Android Keystore for sensitive data).

Compliance Requirements

GDPR Compliance

- **Transparency**: Disclose data collection (e.g., privacy policy).
- **Data Minimization**: Collect only essential user data.
- **User Rights**:
    - Access, delete, or export data (via in-app settings).
    - Consent management for optional data.
- **Data Protection**:
    - Regular security audits.
    - Breach notification procedures (72-hour reporting).

User Data & Privacy Policies

Data Collection

- **Essential Data**:
    - Email, display name, profile photo (optional).
    - Movie preferences, watchlist, authentication logs.
- **Analytics**:
    - Anonymous usage statistics (e.g., Firebase Analytics).

Data Storage & Management

- **Storage**:
    - Firebase Firestore (user data).

    - Firebase Storage (profile images).
- **Controls**:
    - User data access/deletion tools.
    - Retention policies (e.g., "Data deleted after account closure").

Privacy Features

- Private profiles and watchlists.
- Secure authentication (OAuth 2.0 + JWT).
- Encrypted data transmission (HTTPS/TLS).

Security Measures

- **Authentication**:
    - Strong password requirements (if using email/password).
    - Session timeout (30 days inactivity, Page 18).
- **Infrastructure**:
    - Regular security updates.
    - API rate limiting to prevent abuse.

Additional Security Features

Account Security

- Password strength enforcement (min. 8 chars, special symbols).
- Account recovery via email.
- Secure logout (token invalidation).

Data Protection

- End-to-end encryption for sensitive operations.
- Restricted API key access.

Privacy Controls

- In-app privacy settings (e.g., profile visibility).
- Opt-out for analytics tracking.

4.5 Third-Party Integration

- SDKs & Libraries:
    - Firebase: Authentication, Firestore, Storage
    - Google: Play Services Auth, Identity Services
    - UI: Material Components, ConstraintLayout, Navigation, ViewPager2
    - Networking: Retrofit2, OkHttp3, Gson
    - Reactive: RxJava2, RxAndroid
    - Image: Glide, Picasso
    - Database: Room, SharedPreferences
    - Architecture: Lifecycle (ViewModel/LiveData), Paging Library
    - Media: YouTube Player, Custom Blur Engine

Integration Details:

Component Interconnections:

1. Firebase Services:
    - Auth for Google/email login → Firestore for user data → Storage for media
2. TMDB API:
    - Retrofit + RxJava fetches data → Gson parses → LiveData updates UI

3. UI Framework:
    - Material Components + Navigation → ViewPager2 + ConstraintLayout
4. Data Flow:
    - Remote (Firestore/TMDB) ↔ Local (Room) ↔ UI (ViewModel/LiveData)
5. Media Pipeline:
    - Glide loads images → YouTube Player embeds trailers → Custom effects

5. UI/UX Design Specifications

5.1 Wireframes & Mockups

- Screens Overview: Wireframes for each screen or major component.

- High-Fidelity Designs: Detailed mockups demonstrating the final look-and-feel.

5.2 Navigation & Flow

- User Flow Diagrams: Outline of possible user journeys through the app.
- Accessibility Guidelines: Information on how the app meets accessibility standards.

6. Deployment & Maintenance

6.1 Deployment Plan

- Release Process:

The release process for the CineMatch movie recommendation app involves both mobile and web versions designed for academic demonstration. The Android application is configured with version 1.0 in the build settings, targeting devices running Android 7.0 or later. Firebase services handle authentication and data storage, ensuring seamless synchronization between the mobile and web platforms. The mobile app undergoes code optimization and thorough testing of core features such as Google login, movie searches, and bookmark functionality. The web version, developed using React.js, connects to the same Firebase backend to maintain data consistency. Instead of traditional app store distribution, the Android version is packaged as a signed APK for direct installation, while the web application is deployed via Firebase Hosting with a shareable URL. Comprehensive documentation accompanies the project, including setup instructions for both platforms, enabling easy evaluation by instructors and peers.

Testing occurs in multiple phases to ensure reliability across platforms. Initial developer testing verifies core functionality on various devices and browsers. Peer testing follows, with classmates evaluating both versions to identify bugs and assess the recommendation system's effectiveness. Stress testing simulates multiple users accessing the web version simultaneously to check performance under load. For distribution, the Android APK is shared with installation guidelines, while the web version remains accessible via a hosted URL. API keys and sensitive data are secured using environment variables, and a demo account option allows quick access without registration. The final submission includes source code, database schemas, test cases, and a demonstration video showcasing both versions. This approach ensures the project meets academic requirements while demonstrating professional development practices for cross-platform applications.

- Rollout Strategy:
    1. Initial Verification- Internal quality checks by the development team to validate core functionality and crash reporting over 3 days.

2. Limited User Evaluation- Extended to a select group of classmates (25% of potential users) for 1 week, focusing on usability feedback and performance metrics.
3. Web Deployment- The web version is progressively made available to all users over 48 hours, with performance monitored via Firebase Analytics.
4. Mobile Deployment- The Android APK is released in controlled batches, with each group receiving installation instructions

7. Appendices

7.1 Glossary

- API (Application Programming Interface)- A set of protocols and tools that allow different software systems to communicate with each other. CineMatch uses APIs to fetch movie data from TMDb and integrate YouTube for trailers.
- Authentication - The process of verifying the identity of a user. Typically implemented through login credentials.
- Android - A mobile operating system developed by Google, used as the primary platform for CineMatch.
- Backend - The server-side part of the app that handles data processing, storage, and business logic (e.g., Firebase for CineMatch).
- Bookmarking - A feature that lets users save movies to a personal list for later access.

- CSRF (Cross-Site Request Forgery)- A security attack where unauthorized commands are sent from a trusted user. CineMatch implements protections against this.
- Database- An organized collection of structured information or data, typically stored electronically. CineMatch uses a database to store user data and movie preferences.
- Deployment - The act of making an application available for use. Can be done on web servers or app stores
- Firebase- A Google-backed platform providing backend services like authentication, databases, and analytics, used in CineMatch for user data management.
- Frontend - The client-side part of the app that users interact with (e.g., UI screens and navigation).
- MVVM (Model-View-ViewModel) - A design pattern separating UI logic from business logic, used in CineMatch's architecture.
- TMDb (The Movie Database)- A third-party movie database API providing CineMatch with movie details, ratings, and metadata.
- UI/UX (User Interface/User Experience)- Refers to the design and usability of the app's screens and interactions.
- YouTube Integration- A feature allowing trailers to be played within the app using YouTube's embedded player.

7.2 References & Resources

- Related Documentation: Other documents or external references that provide additional context.
- External Links: Useful resources and links to design guidelines (e.g., Apple Human Interface Guidelines, Material Design).