CineLibra

Software Requirements Specification

07.04.2024

Tolga Fehmioğlu 150120022
Mehmet Toprak Balıkçı 150121032
Enes Torluoğlu 150121002
Muhammed Enes Gökdeniz 150121538

Prepared for
CSE3044 Software Engineering Term Project

Table of Contents

1. Introduction

This Software Requirements Specification document serves as a guide detailing the requirements and specifications necessary for the development of Cinelibra, a mobile application designed to manage and track movies and books. This document encapsulates the essential information required by developers to design, develop, and implement the Cinelibra application.

1.1 Purpose

The primary purpose of this SRS document is to explain the functional and non-functional requirements, constraints, and dependencies of the Cinelibra application. It aims to provide clarity and direction to the development team, and help the creation of this mobile app. Additionally, this document serves as a reference point for everyone involved in the development process, and aims to make clear the project objectives and user expectations.

1.2 Scope

(1) The software product to be produced is a mobile application called "Cinelibra."

(2) The Cinelibra mobile application will:

- *Provide users with an archive of books and movies.*
- *Include recommendation algorithms to suggest movies or books based on user preferences, viewing history, ratings, and other relevant factors.*
- *Allow users to mark favorites and create personalized lists of preferred books and movies.*
- *Offer a user-friendly interface for browsing and searching the archive.*
- *Support user authentication and secure access to personal data.*
- *Provide options for users to rate and review books and movies.*
- *Provide a comment section to integrate with other user's comments.*

The Cinelibra mobile application will not:

- *Allow users to buy or watch books or movies directly.*
- *Include original content produced specifically for the application, such as exclusive short stories or films.*
- *There will be no chat section for discussing something.The possible interaction between users can be provided by the comment section.*

- *In general it will not include features unrelated to the management and recommendation of books and movies.*

(3) Application of the Cinelibra software:

- *The Cinelibra mobile application targets users who enjoy books and movies and seek personalized recommendations in these categories.*
- *Key objectives and goals include:*
    - *Providing users with a convenient platform to discover new books and movies based on their interests and preferences.*
    - *Enhancing user engagement through personalized recommendations and social sharing features.*
    - *Offering a unique user experience for browsing, searching, and managing favorite content.*

1.3 Definitions, Acronyms, and Abbreviations

This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.

1.3.1 Definitions

Firebase	*FireBase is an app development platform that helps you build and grow apps games users love.Backed by Google and trusted by millions by businesses around the world*
React Native	*React Native is an open - source mobile application framework created by Facebook. It allows developers to build mobile applications using JavaScript and React*
System Sequence Diagram	*is a type of UML (Unified Modeling Language) diagram used in software engineering to depict the interactions between external actors and a system*
UML Class Diagram	*A UML (Unified Modeling Language) class diagram is a type of structural diagram used in software engineering to visualize the structure of a system's classes, their attributes, methods, and relationships.*
Wireframe Diagram	*A wireframe diagram is a visual representation of a user interface or web page layout.*
Data Flow Diagram	*DFD is a graphical representation of the flow of data within a system.*

Use Case	*A use case is a concept used in software engineering to describe the behavior of a system or software application from the perspective of an external user or actor*
Mobile Device	*Mobile device refers to a handheld computing device that is designed for portable use and can be carried by an individual.*

1.3.2 Acronyms and Abbreviations

OS	Operating system
CPU	Central Processing Unit
RAM	Random Access Memory
GUI	Graphical User Interface
API	Application Programming Interface
GB	Gigabyte
TB	Terabyte
MBPS	Megabits per second

1.4 References

1. https://www.lucidchart.com/pages/uml-use-case-diagram
2. https://blog.hubspot.com/marketing/data-flow-diagram

1.5 Overview

The SRS is organized under three headers. “General Description”, “Specific Requirements”, “UML Diagrams”:

The “General Description” will encompass a high level overview and the general constraints of the CineLibra mobile programme.

The “Specific Requirements” header will elaborate more in depth to the requirements of the CineLibra mobile programme .

The “UML Diagrams” part will consist of the diagrams created for the CineLibra mobile programme such as; Use case and Class diagrams, Sequence diagrams, Data Flow diagrams and the State Transition diagrams.

2. General Description

2.1 Product Perspective

2.1.1 System Interfaces

To be able to use this app the client side requirements are:

Specification	Description
Mobile Device	Any mobile device running on android 12/ ios 15 or higher firmware.

2.1.2 Interfaces

- Registration is mandatory in order to use the product.
- User has to provide valid credentials for the user name, password and email fields.
- Users may alter their credentials provided beforehand in the app after a successful login.

2.1.3 Hardware Interfaces

Specifications	**Particulars**
OS	**Windows 11 2024 64-Bit**

CPU	**Intel Core i5 12400 6 - cores**
RAM	**16 GB**
Storage	**2 TB Hard Drive**
Bandwidth	**14-18 MBPS**

2.1.4 Software Interfaces

1. **Software**
    a. **React Native v0.73 or, Higher**
2. **Database**
    a. **Firebase v13.1 or, Higher**

2.1.5 Communications Interfaces

The proposed application shall be distributed through application stores of the native os' such as apple app store and android google play store.

2.1.6 Memory Constraints

Please refer to the section 2.1.3 Hardware Interfaces.

2.2 Product Functions

Function No.	Function Name
1	User Registration
2	User Login
3	User Logout
4	User Profile
5	App Home Page
6	App Navigation Menu
7	Item Search Page
8	Item Information Page

9	Commenting on Item
10	Rating Item
11	“Watched/Read” and “Watch/Read Later” List Page
12	“Favorites” List Page

2.3 User Characteristics

The user characteristics for Cinelibra covers a broad demographic, ranging from teenagers to adults, with varying levels of technological proficiency. Since this app is about movies and books it is expected that most of the users will be interested in books and cinema. While some users may be tech-savvy and prefer advanced features, others may require a more straightforward interface, therefore to reach a larger user base Cinelibra aims for a more understandable and easy to use user interface. These characteristics will guide the development of Cinelibra to ensure it meets the needs of its user base effectively.

2.4 General Constraints

- Technological Constraints: Cinelibra must be developed using React Native framework due to project requirements and compatibility with cross-platform development.
- Budgetary Constraints: There is no development budget for Cinelibra so that only free to use APIs and softwares should be used during development of Cinelibra.
- Time Constraints: The project has a strict timeline of a few months for development and deployment, requiring efficient project management and timely delivery.
- Resource Constraints: Limited availability of human resources and limited access to external APIs may impact the development timeline and feature implementation.
- Compatibility Constraints: Cinelibra must be compatible with a range of mobile devices and operating systems,iOS and Android, necessitating thorough testing and optimization.

2.5 Assumptions and Dependencies

The compatibility of the Cinelibra application with various mobile operating systems (iOS, Android) is assumed.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- CineLibra will have a modern and easy-to-use design, making it simple for users to find what they need.
- It will work well on different devices like tablets, and phones, so everyone can use it comfortably.

3.1.2 Hardware Interfaces

CineLibra will work on many devices like computers, tablets, and phones, including iOS, and Android.

3.1.3 Software Interfaces

- Firebase Integration: CineLibra will integrate with Firebase for backend services, including user authentication, real-time database storage, and This integration will ensure efficient data management and synchronization across devices.
- Third-Party APIs: CineLibra may integrate with third-party APIs for functionalities such as movie data retrieval from external databases.

3.1.4 Communications Interfaces

- Network Connectivity: CineLibra will require reliable internet connectivity for communication between the client application and Firebase servers.

  Encryption: Communications between the CineLibra application and Firebase servers will be encrypted using secure protocols such as HTTPS, ensuring the privacy and integrity of user data during transmission over the internet.

3.2 Functional Requirements

Function No.	Function Name	Description	Sub-Functionality
1	User Registration	A GUI in the app shall be required for the user registration process.	When the app is launched for the first time the registration page button shall show up and navigate to the Registration page when pressed.
			The GUI shall include basic fields such

			as: UserName, Email and password.
			Register with google shall be included.
2	User Login	A GUI in the app shall be required for the user login process.	When the app is launched for the first time the Login page button shall show up and navigate to the login page when pressed.
			The GUI shall include required information fields such as: UserName, Password.
			Login with google shall be present.
3	User Logout	A GUI in the app shall be required for the logout process	When the slide over menu is present a button for logout shall be present and shall log out of the account when pressed.
4	User Profile	A GUI in the app shall be required for the user to see their profile information after login.	When the slide over menu is present a profile button shall be present and shall lead to the profile page with the profile name and watch lists contained within.
5	App Home Page	A GUI shall be required that displays core functionalities	When the login process is finished a home page should show up consisting of: Search bar, Recommended; Latest , Popular item lists and a slide over menu button.
6	App Navigation Menu	A Navigation Menu shall be required to navigate inside the app.	A navigation menu in the form of a slide over view shall be present when the menu button is pressed on the home page.
			A logout button, a movie section button, a book section button and a profile button shall be present inside the menu.
7	Item Search Page	A Search page shall be required to let the user search an item.	When the search bar is pressed a search page shall come up showing relevant information alongside relevant items.
8	Item Information Page	An item information page shall be required to show	When an item is pressed anywhere in the GUI the items information page shall

		the information of that item.	come up.
			Favorite, watch later, watched list buttons to add; A rating indicator, the item description and information field, comment section of other users.
9	Commenting on Item	A commenting functionality shall be required.	When a comment button is pressed in an item page a text field shall come up to allow the user to comment.
10	Rating Item	A rating functionality shall be required.	The user shall be able to rate an item from the items page by pressing the rating indicator.
11	“Watched/Read” and “Watch/Read Later” List Page	Watch lists shall be required with sub-Functionalities.	A watched/Read list shall come up when the corresponding button is pressed in the slide over page and shall contain the items in a GUI slide over representation as a list of the items the user has bookmarked.
			A to be watched/To be read list shall come up when the corresponding button is pressed in the slide over page and shall contain the items in a GUI slide over representation as a list of the items the user has bookmarked.
12	“Favorites” List Page	A Favourites list page shall be required.	A to be Favorites list shall come up when the corresponding button is pressed in the slide over page and shall contain the items in a GUI slide over representation as a list of the items the user has bookmarked.

3.3 Non-Functional Requirements

3.3.1 Performance

- Response Time: Most user actions, like loading pages, should take less than 2 seconds.
- Throughput: The system should handle at least 100 users at once without slowing down.

3.3.2 Reliability

- Mean Time Between Failures : The system should work without issues for more than 30 days on average.
- Error Rate: The system should have fewer than 1 error per 100 transactions.

3.3.3 Availability

- Uptime: The system should be available 99.9% of the time, allowing for less than 9 hours of downtime per year.

3.3.4 Security

- Data Encryption: All sensitive information like passwords should be hidden using strong codes.
- Access Control: Only certain people should have access to important parts of the system.

3.3.5 Maintainability

- Code Documentation: The code should be explained well for easier understanding by developers.
- Modularity: Changing one part of the system shouldn't mess up the rest of it.

3.3.6 Portability

- Cross-Platform Compatibility: The system should work on different browsers and devices that support ios or android platforms.

3.4 Inverse Requirements

3.4.1 Sign Up

- *The system shall not allow users to sign up with invalid email addresses or usernames.*

3.4.2 Log In

- *The system shall not grant access to users with incorrect login credentials without appropriate authentication.*

3.4.3 Adding a Movie to Watch Later List

- *The system shall not allow users to add the same movie to the watch later list multiple times.*

3.4.4 Marking a Movie as Watched

- *The system shall not remove movies from the watch later list if they have not been marked as watched.*

3.4.5 Searching Movies

- *The system shall not display irrelevant or unrelated search results based on user queries.*

3.4.6 Comment Section

- The system should not allow users to edit or delete comments posted by other users

3.4.7 User Profile Management

- *The system shall not allow users to delete their profiles without proper authentication and confirmation.*

3.4.8 Commenting and Rating Movies

- *The system shall not allow users to comment or rate movies without first watching them.*

3.4.9 Security Measures

- *The system shall not store sensitive user information, such as passwords, in plain text format.*

3.4.10 Performance Limitations

- *The system should maintain its speed and performance even when many users are using it at the same time.*

3.4.11 Data Integrity

- *The system shall not allow unauthorized users to modify or delete data without proper access privileges.*

3.5 Design Constraints

Platform Compatibility: The software must work well on smartphones, ensuring that users can access it regardless of the device they're using.
Budget and Time Constraints: The development team has limited sources which are free to develop a mobile app and a deadline of approximately 8 weeks to complete the project,

which may influence decisions about which features to prioritize and how much time can be spent on development.
Use of Third-Party APIs: The software relies on a third-party movie database API to fetch information about movies, meaning that the availability and reliability of this API will affect the functionality of the software.
Technology Stack: The development team needs to have a well known level of reading and writing capability in JavaScript and prefers to use the React framework for building the user interface, for that reason it can be a hard task to integrate the system to other frameworks which can be also used as alternatives.

3.6 Logical Database Requirements

Database Usage

- *Firebase will be used as the main database for storing CineLibra's data.*

Data Formats

- *Data will be stored in JSON format, which is easy to work with.*

Storage Capabilities

- *Firebase can store different types of data like user profiles, movie info, comments, and ratings.*

Data Retention

- *We'll keep data in the database as long as it's needed. User data will stay forever, while other data might get deleted over time.*

Data Integrity

- *Firebase has security and validation rules to make sure data is safe and correct.*

Real-Time Updates

- *Changes made to the database will show up instantly in the app, so users always see the latest info.*

Scalability

- *Firebase can handle more users and data as the app grows, without slowing down.*

3.7 Other Requirements

We have listed all the necessary functional requirements to implement the system up to this point. However, there may be future needs that require the system to expand. Therefore, scalability and modularity are the key points of our system's design.

4. UML Diagrams

4.1 Use Cases

USE CASE 1: Sign Up
BASIC FLOW:

1. User enters credentials like username, email and password.
2. System checks if the given email and username are already in use then registers the user if credentials are not in use.
3. User is signed up and directed to log in.

EXTENSIONS:

1-

A. Using a google account to sign up.
    1.User is directed to an external google page to log in to google account.

2-

A. Entered email is in use.
    1. User is required to enter a new email to continue signing up.
B. Chosen username is in use.
    1. User is required to enter a new username.

USE CASE 2: Log In
BASIC FLOW:

1. User enters email and password to login to CineLibra.
2. System checks if credentials are valid.
3. User logs in to the home page of CineLibra.
4. User chooses to get into the movies tab or books tab.

EXTENSIONS:

1-

A. User with the given email doesn’t exist.
    1. System directs the user to the sign up screen.
B. In case of wrong password, the user is required to re-enter password.

USE CASE 3: Adding a movie to watch later list
BASIC FLOW:

1. User chooses a movie from home page
2. User checks the comment and rating of the movie to decide whether to watch it.
3. User adds the movie to the watch later list.

EXTENSIONS:

3-

A. Movie is already marked as watched, it cannot be added to watch later.
    1.System shows a message that the user has already watched the movie.

USE CASE 4: Marking a movie as watched

BASIC FLOW:

1. User enters to profile page, chooses the movie added to watch-later list.
2. User marks the movie as watched.
3. System removes the movie from watch-later list.

EXTENSIONS:

2-

A. Commenting and rating the movie.
    1. If the movie is not marked as watched, user can neither comment nor rate the movie
    2. System shows a message that a user should watch before commenting or rating a movie.
B. Adding the movie to favorites list

USE CASE 5: Searching movies

BASIC FLOW:

1. User uses the filters, like release date, genre etc.
2. System returns relevant results appropriate to the filters.
3. User chooses and goes to the page of the movie they are looking for.

EXTENSIONS:

1-

A. Searching by the name of a movie.
    1. User searches for a movie by the name.
    2. System returns a movie with the searched name if found.
B. Using personalized recommendation algorithm.
    1. User goes to the recommendation tab.
    2. System shows recommended movies chosen for the user by the algorithm.

USE CASE 6: Switching movies/books tabs

BASIC FLOW:

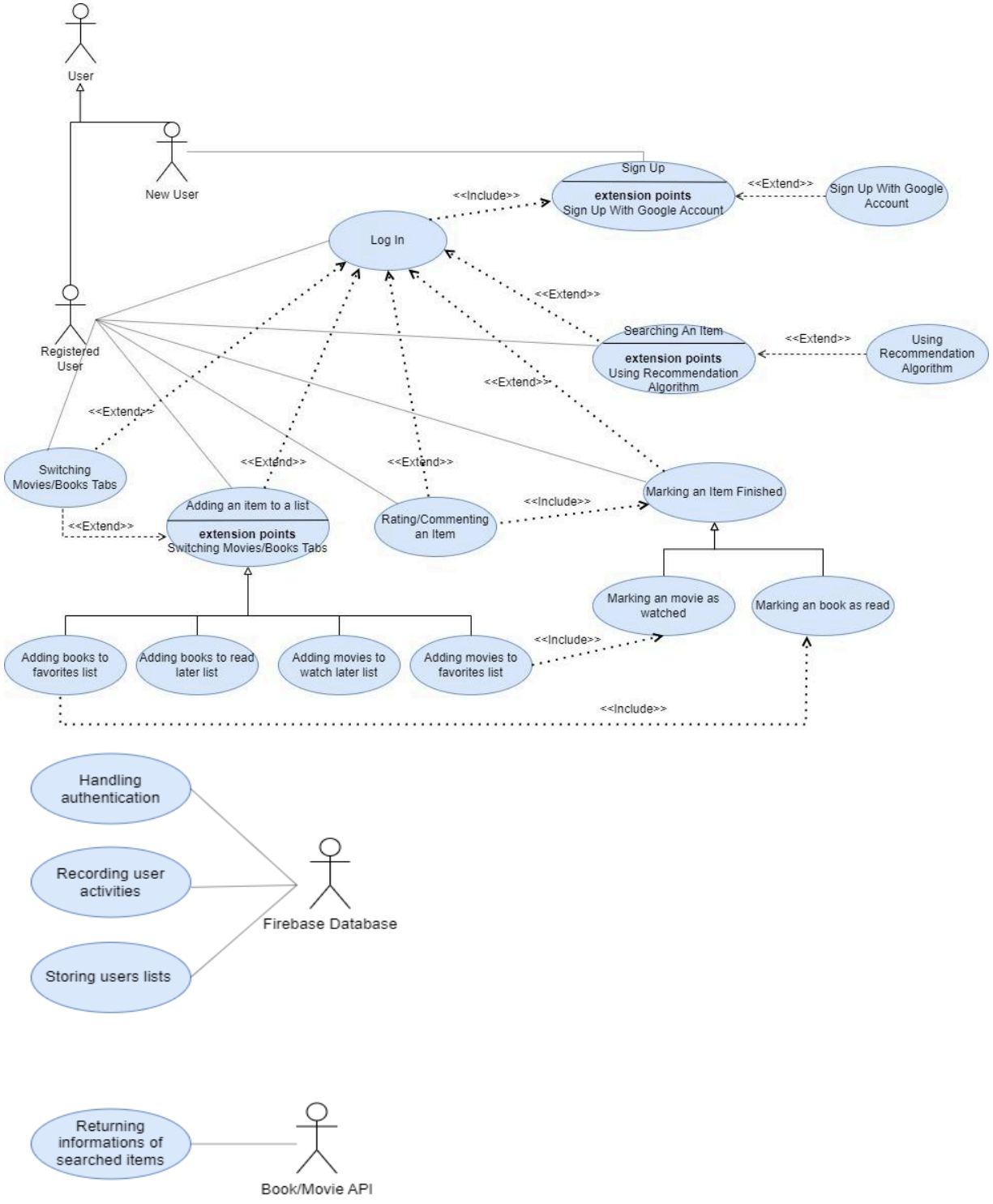
1. User can switch the movies tab to books tab or vice versa.

EXTENSIONS:

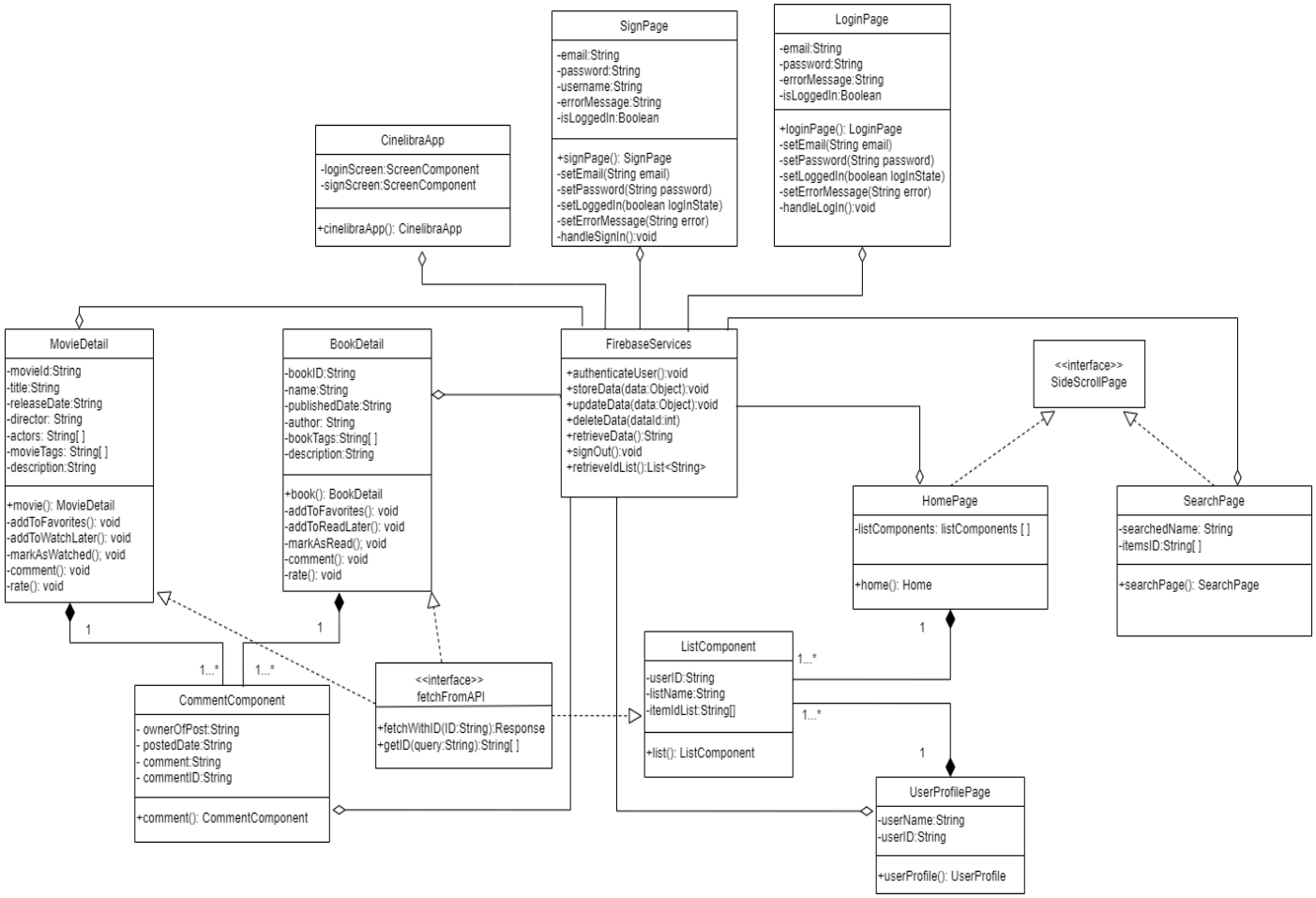
1-

A. User does all the same actions with books too like rating, commenting, marking as read, adding to favorites list.

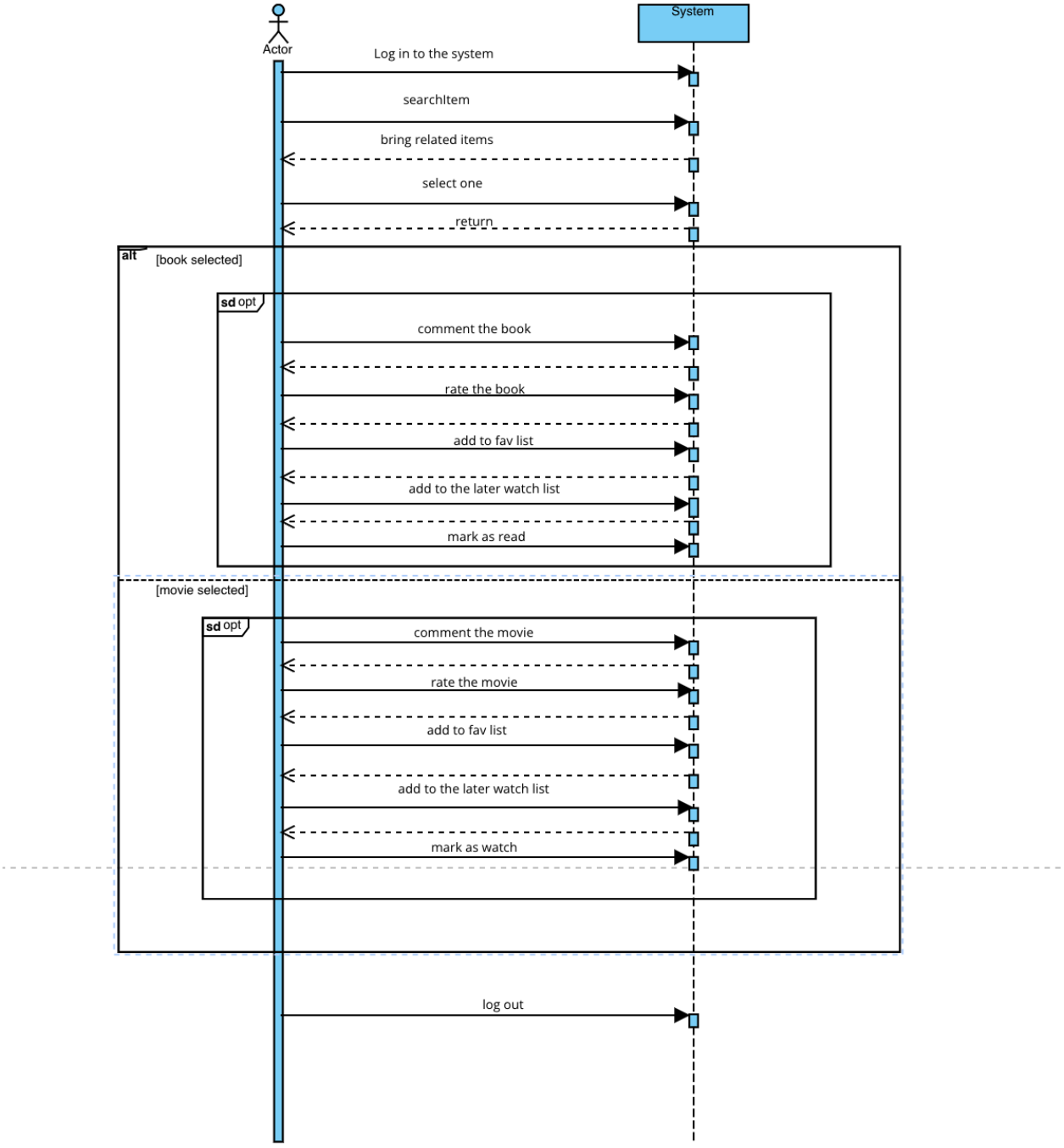
USE CASE DIAGRAMS:



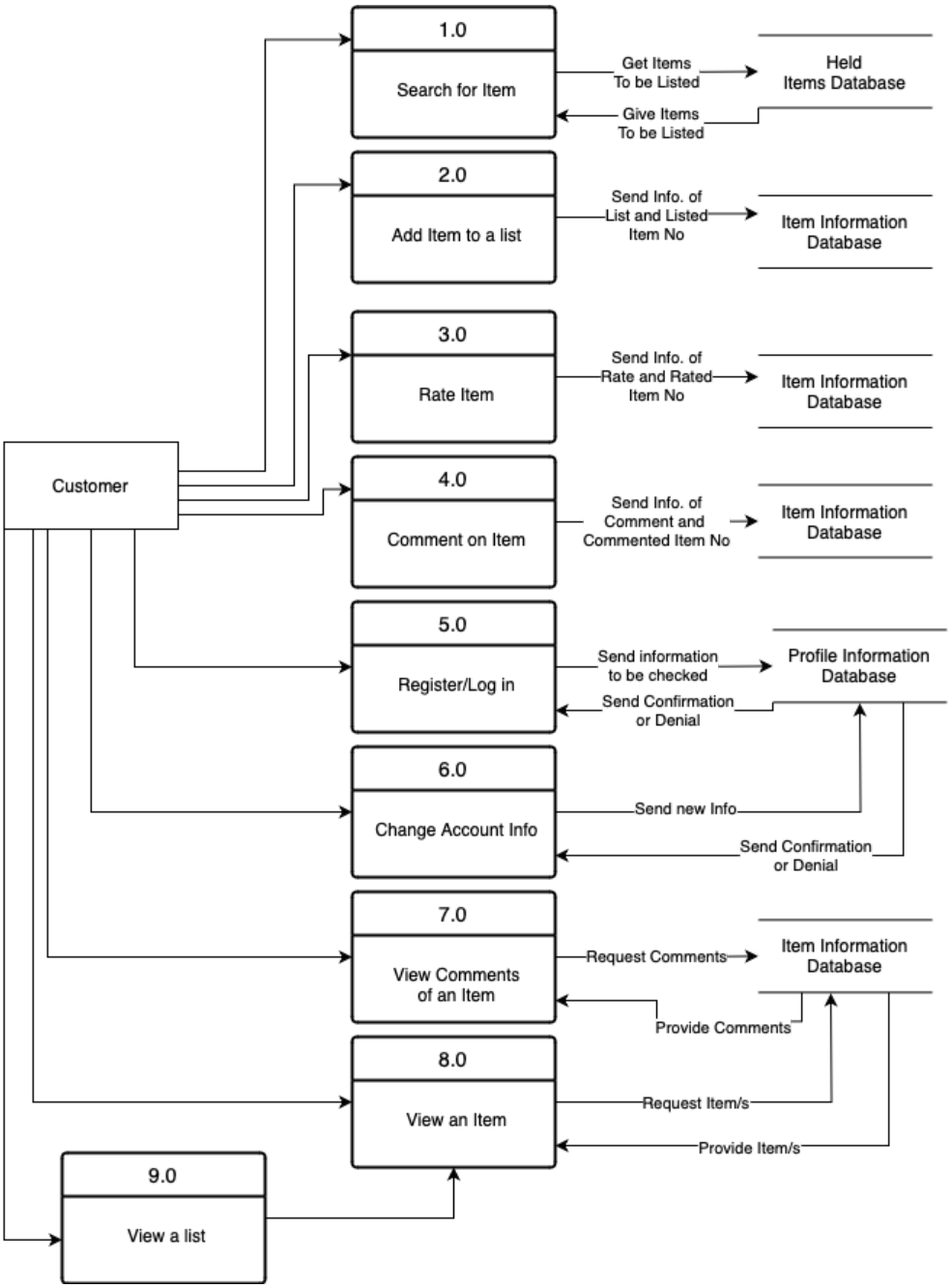
4.2 Classes / Objects



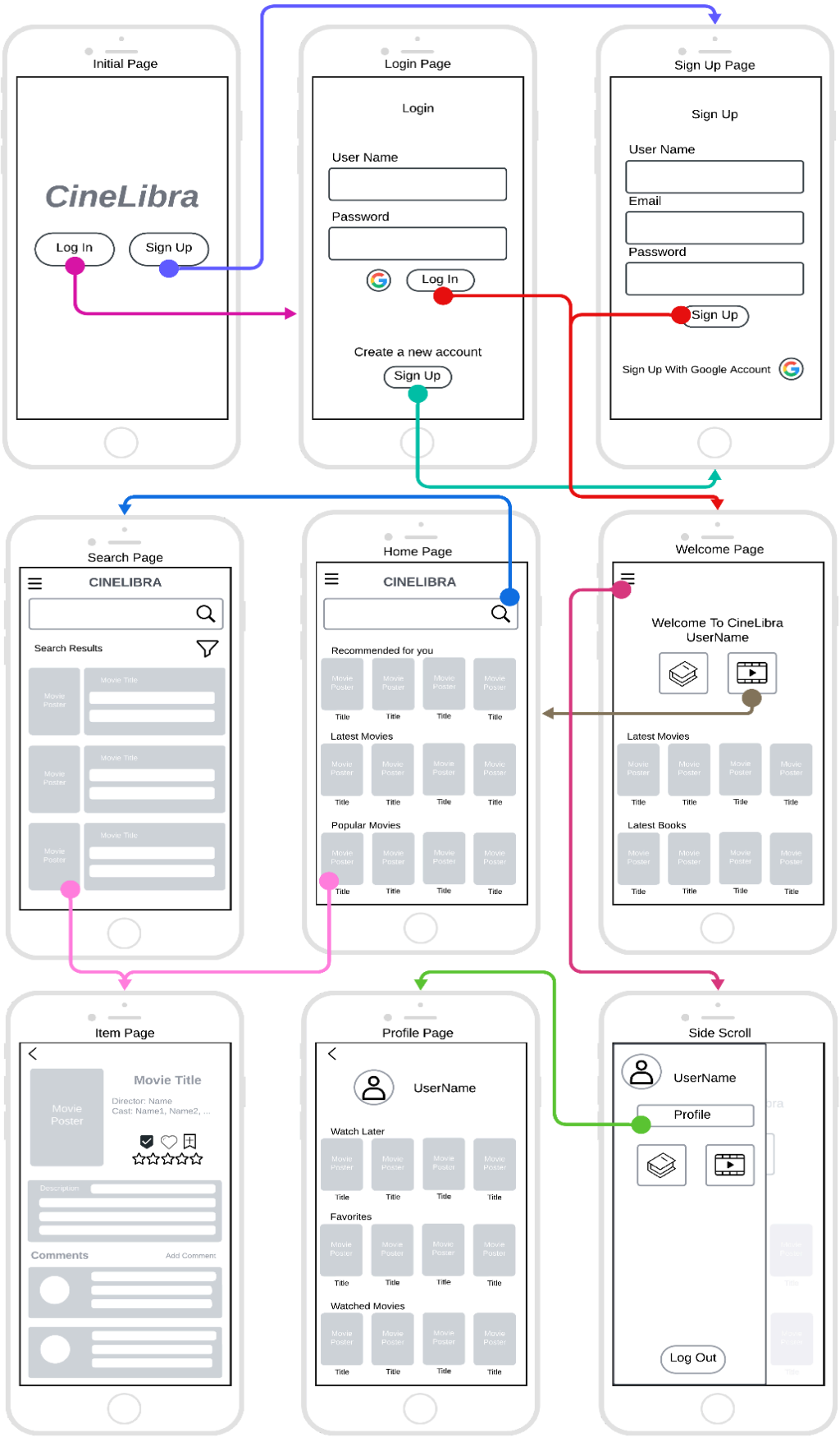
4.3 Sequence Diagrams



4.4 Data Flow Diagrams (DFD)



4.5 Wireframe Diagram



A. Appendices
Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS's overall set of requirements.

Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.

A.1 Appendix 1

Appendix 1: Work distribution

WORK DISTRIBUTION

Group Members	Task
Tolga Fehmioğlu	Scope,assumptions and dependencies,external interface requirements,non-functional requirements,inverse requirements,design constraints,logical database requirements,other requirements,system sequence diagram,uml class diagram
Mehmet Toprak Balıkçı	Data flow diagram, References, Overview, Product perspective, System interfaces, Interfaces, Hardware interfaces, Software interfaces, Communication interfaces, Functional Requirements, Product Functions.
Enes Torluoğlu	Introduction,Purpose,User Characteristics,General Constraints,Use Cases, Use Case Diagram, UML Class Diagram, Wireframe Diagram
Muhammed Enes Gökdeniz	Definitions, Acronyms, and Abbreviations , UML class diagram , General Description

A.2 Appendix 2

Appendix 2: Project Timeline

This appendix outlines the planned schedule for the development of the Cinelibra mobile application.Note that Week 1 represents the starting point when the project is first assigned.

Project Timeline:

Phase	Timeline
Requirements Gathering	Week 1 - Week 3
Initial Design	Week 4 - Week 8
Development	Week 9 - Week 13
Testing and Debugging	Week 13 - Week 14
Finalization and Deployment	Week 15