|  |  |  |
| --- | --- | --- |
|  |  |  |

Yesteryear

By Aryan Bataan

CSC3150: Systems Design

Bataan Aryan - System Specification - Yesteryear - v1

Professor Andy Cameron

# May 31, 2024

A logo with text on it

Description automatically generated

Table of Contents

[1.0. Executive Summary 4](#_Toc168563669)

[2.0. Introduction 5](#_Toc168563670)

[2.1: Problem Statement/Project Vision 5](#_Toc168563671)

[2.2: System Capabilities 7](#_Toc168563672)

[2.3: Non-functional Requirements and Design Constraints 8](#_Toc168563673)

[2.3.1: Non-functional Requirements 8](#_Toc168563674)

[2.3.2: Constraints 9](#_Toc168563675)

[2.4: System Evolution 9](#_Toc168563676)

[2.4.1: Version 2 Changes 10](#_Toc168563677)

[2.4.2: Version 3 and Beyond Changes 11](#_Toc168563678)

[2.5: Document Outline 14](#_Toc168563679)

[3.0: Structural Model 14](#_Toc168563680)

[3.1: Model Introduction 14](#_Toc168563681)

[3.2: Class Diagrams 15](#_Toc168563682)

[3.3: Metadata 15](#_Toc168563683)

[3.3.1: Film 16](#_Toc168563684)

[3.3.2: FilmProfile 18](#_Toc168563685)

[3.3.3: Review 20](#_Toc168563686)

[3.3.4: Review 22](#_Toc168563687)

[3.3.5: Watchlist 24](#_Toc168563688)

[4.0: Architectural Design 25](#_Toc168563689)

[4.1: Architecture Overview 25](#_Toc168563690)

[4.2: Infrastructure Model 27](#_Toc168563691)

[4.2.1: Deployment Diagram 1 – Architecture Overview 27](#_Toc168563692)

[4.2.2: Deployment Diagram 2 – Nodes and Artifacts 28](#_Toc168563693)

[4.3: Hardware and Software Requirements 29](#_Toc168563694)

[4.3.1: Hardware Components 29](#_Toc168563695)

[4.3.2: Software Components 29](#_Toc168563696)

[4.4: Security Plan 31](#_Toc168563697)

[4.4.1: Security Overview 31](#_Toc168563698)

[5.0: User-Interface 34](#_Toc168563699)

[5.1: User-Interface Requirements and Constraints 34](#_Toc168563700)

[5.1.1: User-Interface Guiding Principles 34](#_Toc168563701)

[5.1.2: User-Interface Constraints 34](#_Toc168563702)

[5.2: Window/Screen Navigation Diagram 36](#_Toc168563703)

[5.3: UI Wireframes 37](#_Toc168563704)

[5.3.1: Starting Page 39](#_Toc168563705)

[5.3.2: Lobby 40](#_Toc168563706)

[5.3.3: Create Account 41](#_Toc168563707)

[5.3.4: Log In 42](#_Toc168563708)

[5.3.5: Forgot Password 43](#_Toc168563709)

[5.3.6: Settings 44](#_Toc168563710)

[5.3.7: Accessibility Options 45](#_Toc168563711)

[5.3.8: Adjust Preferences 46](#_Toc168563712)

[5.3.9: Audio Settings 47](#_Toc168563713)

[5.3.10: Spatial Audio Settings 48](#_Toc168563714)

[5.3.11: Manage Profile 49](#_Toc168563715)

[5.3.12: Dashboard 50](#_Toc168563716)

[5.3.13: Film Details 51](#_Toc168563717)

[5.3.14: Choose Film Format 52](#_Toc168563718)

[5.3.15: Pre-Buffering Screen 53](#_Toc168563719)

[5.3.16: Playback Screen 54](#_Toc168563720)

[5.3.17: Offline Viewing Downloads 55](#_Toc168563721)

[6.0: Appendices 56](#_Toc168563722)

[6.1: Glossary 56](#_Toc168563723)

[6.2: References/Bibliography 57](#_Toc168563724)

## 1.0. Executive Summary

Yesteryear is a software application optimized for the Apple Vision Pro headset, envisioned to offer users an immersive cinematic experience from the comfort of their homes. The Yesteryear project strives to address the need for a more engaging and personalized movie-watching experience than modern theatres.

The primary audience for this artifact includes developers, stakeholders, and potential investors interested in the technical and strategic aspects of the project. The customer base encompasses film enthusiasts and tech-savvy individuals seeking a premium home entertainment solution.

The project has progressed through the initial design and planning phases, focusing on creating an intuitive user interface, integrating advanced film formats such as IMAX and Dolby Atmos, and ensuring seamless functionality on the Apple Vision Pro headset. The system supports various film formats, allows for screen size adjustments, and provides uninterrupted playback with pre-buffering features.

Looking forward, Yesteryear aims to evolve through multiple versions. Version 1 will deliver the core functionalities required for a high-quality immersive experience. Version 2 will introduce advanced personalization and social viewing features, while Version 3 will integrate augmented reality elements and enhance user engagement.

This document outlines Yesteryear's system capabilities, non-functional requirements, design constraints, structural model, architectural design, and user interface specifications. The following steps involve finalizing the design, implementing the core features, and preparing for the public launch slated for July 31, 2025.

By focusing on user comfort, technological innovation, and accessibility, Yesteryear is poised to revolutionize the home movie-watching experience, providing a theater-like atmosphere at an affordable cost.

## 2.0. Introduction

Yesteryear is a software application optimized for the Apple Vision Pro AR headset, striving to provide users with an immersive cinematic experience in the comfort of their homes. The application leverages advanced film formats like IMAX, IMAX 70MM, 70MM, and 35MM, along with Dolby Atmos surround sound technology, to deliver a high-quality, engaging viewing experience. The main objective is to offer an alternative to the traditional movie theatre experience, addressing the need for more customer care in the current theatre environment by creating a rewarding experience at a one-time cost.

Yesteryear’s user interface will imitate the user’s familiarity with iOS applications, ensuring ease of use and efficiency. The application will allow users to seamlessly switch between numerous film formats, adjust screen sizes effortlessly, and nourish undisturbed viewing with pre-buffering features. The system will also note that spatial audio enhances rather than disrupts the immersive experience.

Developers can refer to the System Proposal document for a more detailed account of the project’s vision, scope, and requirements. Specific sections such as the “Problem Statement” and “Project Vision and Scope” provide in-depth insights into the motivations and goals behind Yesteryear. In contrast, the sections “Functional Requirements” and “Non-functional Requirements” outline the technical specifications and constraints of the system.

By focusing on user comfort and satisfaction, Yesteryear aims to transform home movie watching into a premium, theatre-like experience backed by state-of-the-art technology and thoughtful design.

## 2.1: Problem Statement/Project Vision

Yesteryear strives to address the underwhelming experience of modern movie theatres by providing a fully immersive, home-based cinematic experience. Today’s movie-going audience often finds the theatre experience impersonal and transactional, lacking the care and dedication there once was. Yesteryear is designed to offer a high-quality, immersive viewing experience using the Apple Vision Pro headset, allowing users to nourish films in various formats, such as IMAX and Dolby Atmos, from the unparalleled comfort of their homes. This solution aims to reintroduce the satisfaction and personal connection missing in the current movie-going experience.

Yesteryear envisions creating a software application that provides a premium cinematic experience at an affordable price, magnifying user comfort. The application will support multiple film formats, including IMAX, IMAX 70MM, 70MM, and 35MM, and will utilize Dolby Atmos for surround sound. The project’s scope includes developing an intuitive interface similar to iOS applications, ensuring users’ ease of use and familiarity. The application will allow immaculate switching between film formats, screen size adjustments, and uninterrupted playback through pre-buffering techniques. The vision is to create a superior, theatre-like experience made to order for individual preferences within a user-friendly digital environment.

Critical stakeholders in the Yesteryear project include film studios, filmmakers, investors, developers, Apple, and content providers. As mentioned, the stakeholders aim to ensure the application’s functionality, broad audience reach, and commercial success. Film studios and filmmakers are hands-on about their content's proper display and quality. Investors and developers are concerned with the technical feasibility, market potential, and financial returns. Apple and content providers want to integrate and distribute the application on their platforms. The expected benefits to all stakeholders include a profitable, innovative product that amplifies user engagement and satisfaction, potentially leading to increased revenue and market share in the home entertainment sector.

## 2.2: System Capabilities

The following section showcases all of Yesteryear’s system capabilities by the first version. It briefly describes each capability with their corresponding Use Case names and ID numbers. For a more detailed account of functional requirements and use case descriptions, analyze Section 4.0, System Proposal, and Section 5.0, System Proposal.

1. Create Account (UC-1): Users need to be able to create new accounts for themselves by providing their personal information and selecting peculiar usernames and passwords.

2. Choose Film Format (UC-2): Users need to be able to select a film format to their liking from the options provided.

3. Maximize/Minimize Screen Size (UC-3): Users need to be able to control the screen size for a better viewing experience.

4. Pre-Buffering and Playback (UC-4): Users need to be presented with a seamless viewing experience with the help of buffering before the film is played.

5. Offline-Viewing Downloads (UC-5): Users need to be able to download their favorite films for offline viewing.

6. Navigate Virtual Environment (UC-6): Users must assimilate to the virtual environment for a better experience.

7. Explore Films (UC-7): Users need to be able to browse the application’s catalog of films to choose from.

8. Manage Profile (UC-8): Users need to be able to update their personal information, preferences, and settings within their profile hub.

9. Log In (UC-9): Users need to be able to log into their accounts to access their personalized features.

10. Provide Content (UC-10): Content providers must be able to upload and manage the content they provide for the platform.

## 2.3: Non-functional Requirements and Design Constraints

Yesteryear strives to be an entertaining, tenable, and user-friendly application. The following section briefly lists and describes the system’s non-functional requirements. The section also presides over the system’s constraints to define Yesteryear’s scope. For further information regarding Yesteryear’s criteria and limitations, check Section 1.0, System Proposal, and Section 4.0, System Proposal.

## 2.3.1: Non-functional Requirements

1. Performance

* Yesteryear should load quickly and respond promptly to user commands.
* Playback should be smooth sailing and undisturbed, with the least buffering or delays.
* The application should handle many concurrent users without notable performance degradation.

2. Operationality

* The application is prepared to launch to the public by July 31, 2025.
* No additional hardware is necessary to develop the application; it will be developed specifically for the Apple Vision Pro headset.
* The only payment accepted will be through Apple Pay, which is processed when the user initiates the application.
* The interface should prioritize user comfort and ensure an uninterrupted user experience.

3. Accessibility

* Yesteryear should be accessible to users with disabilities, complying with standards such as the Web Content Accessibility Guidelines.
* Features like screen readers, keyboard navigation, and alternative input methods (e.g., eye-movement tracers) should be supported.

## 2.3.2: Constraints

1. Securing content licenses and cooperating with copyrights is a crucial constraint. This includes negotiations with film studios and content providers to ensure legal distribution rights.

2. The application must abide by regional and market-specific regulations, including data protection laws (e.g., GDPR, CCPA), content rating requirements, and consumer protection laws.

3. The application must integrate seamlessly with the Apple Vision Pro headset and fully utilize its capabilities, including visionOS, Xcode, and Reality Composure Pro.

4. Ensuring compatibility and high performance across different film formats (IMAX, IMAX 70MM, 70MM, and 35MM) is essential.

## 2.4: System Evolution

In Version 1 (Minimum Viable Product), Yesteryear will focus on delivering the core functionalities necessary for a high-quality, immersive home theatre experience. This includes user authentication and account management, fundamental content discovery and playback, support for multiple film formats (IMAX, IMAX 70MM, 70MM, and 35MM), and the integration of Dolby Atmos for surround sound. Users can adjust screen sizes and nourish uninterrupted playback without interruptions due to buffering. Additionally, core virtual environment navigation and accessibility features will be integrated to ensure that a wide range of users will use the application. Further details about each capability of Version 1 are in Section 5, System Proposal.

In Version 2, Yesteryear will introduce advanced functionalities to enhance user engagement and personalization. The features above will include advanced search and filtering capabilities, enhanced social features such as user groups and live chat, interactive timelines and maps, and machine learning-driven content personalization. Tools for collaboration and shared memories, as well as native mobile applications with offline functionalities, will also be developed. This version of Yesteryear will focus on improving the user experience and expanding the application's capabilities based on user feedback and evolving market demands.

In Version 3, Yesteryear will dedicate its facilities to further expand the application’s ecosystem and integrate cutting-edge technologies to deliver a more immersive and interactive experience. This version of Yesteryear will introduce features such as eye-movement tracers and haptic feedback to amplify users' sensory engagement. In addition, support for virtual events and premiers will be included, allowing filmmakers and studios to hold events directly within the application. Advanced analytics tools will be developed for content providers to better understand user preferences and viewing patterns. This version will explore integrating augmented reality (AR) elements to create a more dynamic and interactive viewing experience and potential collaborations with other VR and AR platforms to expand content accessibility.

## 2.4.1: Version 2 Changes

The viewing experience is the ultimate drawing factor for Yesteryear to be the exceptional outlier as an application. The following information provides additional functional capabilities for Version 2 to satisfy the system’s value:

1. Integrated Social Viewing Experience

* Users can now create virtual rooms where they can watch movies with friends and family in real time, regardless of their physical location.
* This feature supports synchronized playback, allowing all participants to view the film simultaneously with shared audio and visual effects.
* Users can interact via voice chat and on-screen reactions.
* The system synchronizes playback across all participants’ devices.

2. Advanced Content Personalization

* Machine learning algorithms are leveraged to offer highly tailored content recommendations.
* The system analyzes user viewing habits, preferences, and feedback to suggest films and shows that align closely with their interests.
* The system can retrieve user data and view history.

3. Virtual Reality Themed Environments

* Users are allowed to choose from a variety of themed virtual environments to enhance their viewing experience.
* Themes include classic cinema, outdoor drive-in, futuristic theater, and more.
* Each environment is designed to amplify immersion and can be selected or changed during playback.
* The system renders the selected environment around the user’s viewing screen.

## 2.4.2: Version 3 and Beyond Changes

During version 3, Yesteryear will expand its reach and focus on the technicalities of augmented reality. The following information presents Version 3’s extra functionalities to attain the expansion:

1. Augmented Reality (AR) Integration for Interactive Experiences

* Yesteryear will incorporate augmented reality to provide interactive and dynamic movie experiences.
* This use case allows users to interact with AR elements that extend the movie experience beyond the screen, creating a more immersive and appealing environment.
* The system loads AR elements and synchronizes them with the movie playback.
* Users then interact with the AR elements using hand gestures or voice commands.

2. Interactive Storytelling and Choose-Your-Own-Adventure

* Yesteryear will support interactive storytelling where users can influence the movie's plot by making choices at critical points.
* During critical points, the system presents choices for the user.
* The system continues the movie based on the user’s selection.
* The system will select a default option and continue playback if users don't choose.

3. Enhanced Social Features: Virtual Reality (VR) Theatres and Events

* Social features will be expanded to include VR theatres where users can host or attend virtual screenings and events.
* This feature supports real-time interaction and communication within a shared virtual space, amplifying the communal movie-watching experience.
* Users interact with each other through avatars and voice chat.
* The system synchronizes playback for all participants.
* The user creates or joins a virtual theatre or event.

4. Adaptive Content Recommendation System

* The content recommendation system will be enhanced to use advanced machine learning that adapts to user preferences in real time.
* This will provide more accurate and personalized content suggestions based on user interactions and feedback.
* Users receive personalized content suggestions.
* The system uses general recommendations based on popular content if there needs to be more data.

5. Enhanced Accessibility Features

* Features like text-to-speech, voice control, and support for various assistive technologies will be the next frontier of innovation for Yesteryear.
* The accessibility settings will be where users can access these features.
* If a setting fails to apply, the system reverts to the previous state and notifies the user.

Beyond Version 3, Yesteryear strives to make the best use of user data in terms of security and further expansion.

1. Multi-language and Localization Support

* Future versions of Yesteryear will contain extensive multi-language and localization support.
* Users can experience the application in their native language and provide localized content.
* Users can change the application’s preferred language by accessing the settings and selecting a language.
* If the selected language is not fully supported, the system defaults to the closest supported language and lets the user know.

2. Virtual Tours and Behind-the-Scenes Content

* Yesteryear will offer virtual tours and behind-the-scenes content for movies, allowing users to explore film sets and production processes in a virtual environment.
* This feature will enhance user engagement and provide educational content for children.
* If the virtual content fails to load, the system notifies the user and renders a list of alternative content options.

3. Gamification and Achievement Systems

* To boost user engagement, future versions of Yesteryear will feature gamification elements such as achievements, rewards, and leaderboards.
* Users can earn points and badges for watching movies, interacting with content, and participating in social features.
* The system will also be programmed to keep track of achievements and display progress on a leaderboard.

## 2.5: Document Outline

This system specification document provides an in-depth game plan for Yesteryear, recounting its evolution, structural model, architectural design, and user interaction. The document starts with a summary of the system’s progression through different versions, highlighting new features and improvements. It then improves upon functional and non-functional requirements, data needs, and constraints previously listed in the System Proposal. The Structural Model section includes class diagrams and metadata, accentuating relationships and associations. The Architectural Design section outlines the system’s infrastructure, underlining hardware and software requirements, security measures, and compliance with regulations. Lastly, the User Interface (UI) section underscores end-user interaction requirements, providing a navigation map and UI wireframes to illustrate the user experience. This all-inclusive document aspires to ensure that all aspects of Yesteryear’s design and functionality are clearly defined and aligned with user needs and system capabilities.

## 3.0: Structural Model

## 3.1: Model Introduction

This section comprises an overview of the class diagrams and their metadata. The Class Diagrams section utilizes the Unified Modeling Language (UML) class model to better delineate the system’s structure. The diagram accents the names of the numerous classes, features, techniques, and connections.

The Metadata section is responsible for providing extra information about the attribute components and operations. It outlines the processing logic for the different courses of action and class behavior in optimal circumstances.

## 3.2: Class Diagrams

A diagram of a class

Description automatically generated

[Class Diagram](https://lucid.app/lucidchart/ee769d74-156b-46ff-899d-afec4660e498/edit?viewport_loc=279%2C-3175%2C3135%2C3194%2C0_0&invitationId=inv_ec08b915-7326-4bde-b863-41f26f1e523d)

## 3.3: Metadata

This section chronicles each class's data elements and operations in the previous master diagram. With this as a tool, developers should be able to efficiently understand and conceive the possible outcome of the implementation. Listed below are the page numbers for each class in alphabetical order.

1. Film (Page 13)

2. FilmProfile (Page 14)

3. Review (Page 15)

4. User (Page 16)

5. Watchlist (Page 17)

## 3.3.1: Film

A screen shot of a computer

Description automatically generated

Description: The Film class is responsible for illustrating the characteristics of each film the user selects at a given time.

Visibility: Public

Abstract (Yes/No): No

Additional Information:

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Read Only? | Multiplicity |
| filmID | Unique identifier of each film | Yes | 1 |
| title | Title of the film | No | 1 |
| genre | Genre of the film | No | 1 |
| director | Director of the film | No | 1 |
| release | Release date of the film | No | 1 |
| format | Format of the film (e.g., IMAX, 35MM) | No | 1 |
| duration | Duration of the film | No | 1 |

**Operations:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Is Query? | Is Polymorphic? |
| playFilm | Plays the film | No | No |
| pauseFilm | Pauses the film | No | No |
| stopFilm | Stops the film | No | No |
| addToWatchlist | Adds the film to the user's watchlist | No | No |

3.3.2: FilmProfileA screenshot of a computer

Description automatically generated

Description: The FilmProfile class represents a film profile within the Yesteryear system.

Visibility: Public

Abstract (Yes/No): No

Additional Information:

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Read Only? | Multiplicity |
| profileID | Unique identifier for each film profile | Yes | 1 |
| profileName | Name of the film profile | No | 1 |
| creationDate | Date the profile was created | Yes | 1 |
| profileOwner | Owner of the profile | Yes | 1 |
| profileMembers | List of members in the profile | No | 0..\* |
| watchHistory | List of films watched in the profile | No | 0..\* |

**Operations**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Is Query? | Is Polymorphic? |
| addMember | Adds a member to the profile | No | No |
| removeMember | Removes a member from the profile | No | No |
| updateProfile | Updates the profile name | No | No |
| viewWatchHistory | Displays the watch history of the profile | Yes | No |

## 3.3.3: Review

**A close-up of a review

Description automatically generated**

Description: The Review class represents a film review within the Yesteryear system.

Visibility: Public

Abstract (Yes/No): No

Additional Information:

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Read Only? | Multiplicity |
| reviewID | Unique identifier for each review | Yes | 1 |
| film | The film being reviewed | Yes | 1 |
| user | The user who wrote the review | Yes | 1 |
| rating | Rating given to the film | No | 1 |
| comment | User's comments about the film | No | 1 |
| reviewDate | Date the review was written | Yes | 1 |

**Operations**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Is Query? | Is Polymorphic? |
| submitReview | Submits a new review for a film | No | No |
| viewReview | Displays the details of the review | Yes | No |

## 3.3.4: Review

**A screenshot of a computer

Description automatically generated**

Description: The User class Represents a user within the Yesteryear system.

Visibility: Public

Abstract (Yes/No): No

Additional Information:

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Read Only? | Multiplicity |
| userID | Unique identifier for the user | Yes | 1 |
| username | Username chosen by the user | Yes | 1 |
| password | Password for the user's account | No | 1 |
| firstName | User's first name | No | 1 |
| lastName | User's last name | No | 1 |
| dateOfBirth | User's date of birth | No | 1 |
| emailAddress | User's email address | No | 1 |
| phoneNumber | User's phone number | No | 1 |
| ownedProfiles | List of profiles owned by the user | No | 0..\* |
| joinedProfiles | List of profiles joined by the user | No | 0..\* |

**Operations**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Is Query? | Is Polymorphic? |
| createUserAccount | Creates a new user account | No | No |
| verifySignIn | Verifies user credentials | Yes | No |
| signOut | Signs the user out of their account | No | No |
| createFilmProfile | Creates a new film profile | No | No |
| viewOwnedProfiles | Displays profiles owned by the user | Yes | No |
| viewJoinedProfiles | Displays profiles joined by the user | Yes | No |
| leaveProfile | Leaves a profile | No | No |
| deleteProfile | Deletes a profile | No | No |
| updateUserAccount | Updates user account information | No | No |

## 3.3.5: Watchlist

**A screen shot of a watch list

Description automatically generated**

Description: The User class Represents a user's watchlist within the Yesteryear system.

Visibility: Public

Abstract (Yes/No): No

Additional Information:

**Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Read Only? | Multiplicity |
| watchlistID | Unique identifier for the watchlist | Yes | 1 |
| user | The user who owns the watchlist | Yes | 1 |
| films | List of films in the watchlist | No | 0..\* |

**Operations**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Is Query? | Is Polymorphic? |
| addFilm | Adds a film to the watchlist | No | No |
| removeFilm | Removes a film from the watchlist | No | No |
| viewWatchlist | Displays the films in the watchlist | Yes | No |

## 4.0: Architectural Design

## 4.1: Architecture Overview

This section provides a comprehensive high-level overview of the architecture for Yesteryear, an immersive cinematic experience application designed for the Apple Vision Pro headset. It describes the essential components, interactions, and the overall system design. Readers will learn about the infrastructure model, hardware and software requirements, and the security plan necessary to support and secure the Yesteryear system.

The architectural design of Yesteryear is based on a 3-tier client-server model to ensure scalability, flexibility, and performance. This architecture includes a presentation layer, application layer, and data layer. The presentation layer handles the user interface and interactions; the application layer processes business logic, and the data layer manages data storage and retrieval. To implement this architecture, purchasing two new servers is recommended: one dedicated to handling application services and the other to managing data storage and database operations.

In addition, Yesteryear will leverage iCloud for seamless data synchronization and storage. By integrating with iCloud, users will benefit from consistent and secure access to their profiles, watchlists, and viewing history across multiple Apple devices. This integration will simplify data backup and recovery processes, enhancing system reliability and user experience.

The following subsections include the Infrastructure Model, Hardware and Software Requirements, and Security Plan:

* The Infrastructure Model subsection includes two deployment diagrams for Yesteryear. The first diagram comprehensively illustrates the system's overall structure, detailing the various components and their interactions. The second diagram highlights the physical and logical entities, their connections, and the distribution of elements across the infrastructure.
* The Hardware and Software Requirements subsection outlines the specific hardware and software required for Yesteryear’s longevity. It includes the specifications for servers, network devices, and client devices.
* The Security Plan subsection presents the security measures and protocols to protect the Yesteryear system. It includes data encryption, user authentication, and authorization processes to ensure secure access and, most importantly, data integrity.

## 4.2: Infrastructure Model

## 4.2.1: Deployment Diagram 1 – Architecture Overview

A screen shot of a computer

Description automatically generated

[**Deployment Diagram 1 - Architecture Overview**](https://lucid.app/lucidchart/f5c4f8b8-7380-43bf-898f-bbe783ebaaae/edit?invitationId=inv_d5d1e603-84ad-4865-b1d6-0f3b3d668cde)

## 4.2.2: Deployment Diagram 2 – Nodes and Artifacts

A diagram of a computer server

Description automatically generated

[Deployment Diagram 2 - Node and Artifacts](https://lucid.app/lucidchart/91d77ed4-a1c3-4c18-8184-c34eda619208/edit?viewport_loc=1369%2C-1123%2C2907%2C2962%2C0_0&invitationId=inv_2853a39b-3dd2-4df5-a321-c71e80094931)

## 4.3: Hardware and Software Requirements

Yesteryear does not require any additional hardware for its development. Also, the system architecture requires specific hardware components for client devices and servers, along with various software components for development, deployment, and operation.

## 4.3.1: Hardware Components

The system architecture requires a mixture of local hardware for client devices and cloud-based services for servers and data management. External services like Apple Push Notification Services (APNS) and iCloud do not require additional hardware.

1. Client Devices: Users will need the Apple Vision Pro Headset that supports Yesteryear.

2. Application Server: Yesteryear uses cloud-based servers such as AWS Elastic Compute Cloud (EC2) to host the application logic and API services. For scalability, resources can be quickly scaled based on demand.

3. Database Server: Yesteryear utilizes cloud-based database services such as AWS Relational Database Service (RDS) with PostgreSQL to host user data and application data.

4. Backup Database Server: For data redundancy and disaster recovery, Yesteryear uses a backup database server like the AWS RDS Multi-Az, which provides automatic backup and data replication.

Users will need the Apple Vision Pro headset that runs on visionOS 1.1 or later to install and use Yesteryear. No additional software is required to use the application; however, users must download the application from the Apple App Store available in the headset.

## 4.3.2: Software Components

An iOS/macOS development environment is required for Yesteryear's system implementation. This includes necessary development tools and services for building, testing, and deploying the application.

1. Xcode: The development and testing will be conducted using Xcode, Apple’s integrated development environment. Xcode includes all necessary tools, such as compilers, debuggers, and performance analysis tools for building iOS, visionOS, and macOS applications.

2. visionOS: Yesteryear will run on visionOS, the operating system designed for Apple Vision Pro. VisionOS provides the framework and tools to create immersive AR/VR experiences.

3. Reality Composure Pro: This tool will design and build 3D content and AR/VR experiences for Yesteryear. Reality Composer Pro allows for creating complex scenes and interactions, ensuring a high-quality immersive experience.

4. Figma: Figma will be used to design the user interface and create wireframes and mockups. Figma supports real-time collaboration and provides a range of plugins to enhance productivity.

5. Jenkins/GitHub Actions: Implementing a CI/CD pipeline using tools like Jenkins or GitHub actions will automate the testing and deployment processes, ensuring that new updates are quickly and reliably deployed.

6. Encryption Protocols: Implementing robust security measures, including AES-256 for data-at-rest encryption and TLS 1.3 for data-in-transit encryption.

7. Compliance Tools: Ensuring compliance with data protection regulations like GDPR and CCPA by integrating tools and services that provide these features into the development and operational workflows.

8. Authentication and Authorization: Multi-factor authentication (MFA) will be implemented to enhance the security of user logins. Use Role-Based Access Control (RBAC) will also be set in place to restrict access to sensitive data and functionalities based on user roles and responsibilities.

## 4.4: Security Plan

## 4.4.1: Security Overview

Security is of paramount concern to Yesteryear, given the sensitive nature of user data and the need to protect the integrity of immersive cinematic experiences. This section outlines potential security threats, essential security requirements, and recommended security software to mitigate risks and ensure robust protection.

1. Data Breaches: Unauthorized access to user data, including personal information, viewing history, and preferences, poses a significant threat. Data breaches can lead to identity theft, privacy violations, and loss of user trust.

2. Unauthorized Access: Potential unauthorized access to the system, either through compromised credentials or system vulnerabilities, can result in data manipulation, service disruption, and exploitation of system resources.

3. Data Integrity Attacks: Attacks aiming to alter or corrupt data, such as user profiles or film metadata, can degrade the user experience and damage the system’s credibility.

4. Service Denial Attacks: Distributed Denial of Service (DDoS) attacks can severely deluge system resources, causing service outages and disruptions in user access.

5. Insider Threats: Employees or contractors with legitimate access may intentionally or unintentionally compromise data security.

6. Data Encryption: Encrypt all stored data using AES-256 to ensure that data remains secure even if storage devices are compromised. TLS 1.3 encrypts data transmitted between client devices and servers, preventing eavesdropping and man-in-the-middle attacks.

7. Authentication and Authorization: Multi-factor authentication (MFA) will be implemented to enhance the security of user logins. Use Role-Based Access Control (RBAC) will also be set in place to restrict access to sensitive data and functionalities based on user roles and responsibilities.

**4.4.2: Security Plan**

Below is the system architecture threats table. It is responsible for outlining every possible threat while coexisting with each component of the system architecture. This section also coincides with the plan for controlling each danger associated with an element.

Physical Network App File U.S

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Threats/Components | Loss or Theft | Eavesdropping | Data Breaches | Vulnerability | Corruption | SUC | UAA | LPIPV |
| Apple Vision Pro (User Headset) | 1 | 2 |  |  |  |  |  |  |
| Internet Connection |  | 3 |  | 4 |  |  |  |  |
| Yesteryear Application Server |  |  |  | 5, 6 |  |  |  | 7 |
| External API Services |  |  |  | 8 |  |  |  |  |
| Database Server |  |  |  | 9, 10 | 11 |  |  |  |
| Backup Database Server |  |  |  | 12 | 13 |  |  |  |

**Key**

\* SUC: Stolen User Credentials

\* UAA: Unauthorized Access to Accounts

\* LPIPV: Leakage of Personal Information and Privacy Violation

**Controls**

1. Disaster Recovery Plan: Establish a comprehensive disaster recovery plan to ensure system continuity and data recovery in case of catastrophic events.

2. Fire Suppression Systems: Install halon fire systems in server rooms and sprinklers throughout the building to protect hardware from fire damage.

3. Elevated Server Rooms: Place server rooms on higher floors to reduce the risk of flood damage.

4. Uninterruptible Power Supply (UPS): Deploy UPS systems on all central network servers to protect against power loss and ensure continuous operation.

5. Virus Checking Software: Implement antivirus software across all servers and client devices to detect and mitigate malware threats.

6. Contract Guarantees: Secure contract guarantees from interexchange carriers for reliable communication services.

7. Fiber Backbone Redundancy: To prevent network failure, Lay extra backbone fiber cables in different conduits between central servers.

8. User Training on Security: Conduct extensive training programs for users on recognizing and preventing social engineering attacks and emphasize the importance of security in monthly newsletters.

9. Strong Password Policies: Enforce strong password policies and use password management software to protect user accounts.

10. Multi-Factor Authentication (MFA): Implement MFA to enhance login security and protect against unauthorized access.

11. Call-Back Modem System: Utilize a call-back modem system to verify user identity during remote access.

12. Application Layer Firewall: Deploy application layer firewalls to monitor and filter incoming traffic, preventing unauthorized access and data breaches.

## 5.0: User-Interface

## 5.1: User-Interface Requirements and Constraints

In designing the user interface for Yesteryear, our primary goal is to create an immersive, intuitive, and engaging experience for users of the Apple Vision Pro headset. The interface should be visually appealing, simple to navigate, and responsive to user inputs, ensuring an uninterrupted interaction with the cinematic experience.

## 5.1.1: User-Interface Guiding Principles

1. User-Centric Design: Prioritize the needs and preferences of our users, ensuring the interface is intuitive and requires a minimal learning curve. The design should accommodate new and experienced users, providing a smooth and enjoyable experience.

2. Consistency: Maintain consistency in design elements such as icons, colors, and typography across all screens and interactions. This helps users build familiarity and predictability, enhancing their overall experience.

3. Accessibility: Ensure the interface is accessible to all users, including those with disabilities. This involves implementing voice commands, adjustable text sizes, and visual contrasts that cater to diverse user needs.

4. Performance: Optimize the interface for performance, ensuring quick load times and smooth transitions. The goal is to minimize delays or interruptions that could detract from the immersive experience.

5. Security and Privacy: Integrate robust security and privacy measures within the interface. Users should feel confident that their personal information is protected at all times.

## 5.1.2: User-Interface Constraints

1. Limited Screen Space: The Apple Vision Pro headset provides a limited visual area, which can constrain the amount of information displayed simultaneously.

2. Navigation Complexity: Navigating through a 3D interface can be challenging for users unfamiliar with VR environments, leading to a steeper learning curve.

3. Input Method Constraints: Users are limited to the input methods supported by the Vision Pro, such as gestures, voice commands, and limited physical controls, which might not be as precise as traditional input devices.

4. User Fatigue: Extended headset use can lead to physical and mental fatigue, reducing the duration for which users can interact with the application comfortably.

5. Accessibility Challenges: Ensuring accessibility for users with disabilities can be more complex in a VR environment than traditional interfaces.

6. Performance Limitations: High-quality graphics and real-time rendering required for an immersive experience can strain the device's processing capabilities, potentially affecting performance.

7. Latency Issues: Network latency can impact the application's responsiveness, especially in real-time interactions or data retrievals.

8. Battery Life: The battery life of the Vision Pro headset limits the duration of continuous use, affecting long sessions of engagement with Yesteryear.

9. Environment Dependence: The effectiveness of the VR experience can be influenced by the user's physical environment, such as space constraints and lighting conditions.

10. User Disorientation: Prolonged use of VR can lead to disorientation or motion sickness for some users, impacting the overall user experience.

## 5.2: Window/Screen Navigation Diagram

The following diagram showcases the flow and set arrangement of Yesteryear’s core navigation. The diagram offers developers an overview of the connection between UI components and the screen.

A diagram of a computer screen

Description automatically generated

[Window/Screen Navigation Diagram](https://lucid.app/lucidchart/7cfc02c1-22c4-4f6e-9609-5241fea968f9/edit?invitationId=inv_60315cfa-f98b-463c-97b0-4546a051ef81)

## 5.3: UI Wireframes

In this section, detailed wireframes for Yesteryear are presented for the crucial components of Yesteryear, including the Starting Page, Lobby, Dashboard, Explore Films, Film Details, Pre-Buffering Screen, Playback Screen, Manage Profile, Settings, Create Account, Accessibility Options, Adjust Preferences, Audio Settings, Spatial Audio Settings, Choose Film Format, Forgot Password, and Offline Viewing Downloads. Each wireframe focuses on the critical elements and interactions, providing a visual guide for the development team and stakeholders.

The wireframes in question should be considered something other than tools for designers and developers. Still, it is also a means to communicate the application’s intended user experience to stakeholders, ensuring alignment and clarity in the project’s vision. By outlining the navigation paths and interface elements, the wireframes assist in identifying potential usability issues in the infancy stages of the design process, making room for iterative improvements and user-centered design decisions.

A screenshot of a computer screen

Description automatically generated

[UI Wireframes](https://lucid.app/lucidchart/02a6a377-bede-4fde-ac85-186c20d955e3/edit?viewport_loc=-792%2C534%2C2930%2C2985%2C0_0&invitationId=inv_824e93e7-5987-4fc5-a92f-42e759f83037)

## 5.3.1: Starting Page

A screen shot of a computer

Description automatically generated

The “Starting Page” is the first screen users are shown when they access Yesteryear from the Apple Vision Pro homepage. The home page is adorned with the Yesteryear logo, which gives users a rough idea of the style in which Yesteryear will be presented going forward. Near the bottom of the screen is the “Continue” button that takes the user to the lobby.

## 5.3.2: Lobby

A screenshot of a computer

Description automatically generated

The “Lobby” contains three buttons, one of which is the “Create Account” button, which allows a new user to create an account they’ll use to access Yesteryear. The “Log In” button allows existing users to enter their credentials and reenter Yesteryear. Lastly, the “Explore as Guest” button allows non-registered individuals to view the dashboard, where they’ll be presented with different buttons that give them access to Yesteryear’s core features.

## 5.3.3: Create Account

A screenshot of a computer screen

Description automatically generated

Upon clicking the “Create Account” button, the user is directed to the “Create Account” screen, where they’re asked to enter their primary email address, create a password unique to them, and submit the info when finished.

## 5.3.4: Log In

A screenshot of a login screen

Description automatically generated

Suppose the “Log In” button is selected. In that case, the user will be brought to this page, where they’ll be asked to enter their registered email address and password, assuming the individual logging in is an existing user. The next step for the user will be to select submit, but if they’ve forgotten their set password, they will choose the “Forgot Password” button and be redirected to the “Forgot Password.” screen.

## 5.3.5: Forgot Password

A screenshot of a computer screen

Description automatically generated

If the user forgets their password and selects the “Forgot Password” button, they will be sent to this screen where they’ll be asked to enter their email address and new password and submit when done.

## 5.3.6: Settings

A screenshot of a computer

Description automatically generated

The “Settings” wireframe features a clean and straightforward header titled "Settings" with a back button for easy navigation. The main section includes a prominently centered "Adjust Preferences" button, allowing users to customize audio, video, and interface settings. Below it, the "Accessibility Options" button directs users to a dedicated menu for enabling and customizing features like screen readers, motion control, and color contrast adjustments. Upon saving any changes, a confirmation notification appears at the bottom of the screen, simply stating "Settings saved successfully" to inform users that their preferences have been updated.

## 5.3.7: Accessibility Options

A diagram of options for a user

Description automatically generated with medium confidence

The “Accessibility Options” wireframe presents a user-friendly interface to enhance the user experience for individuals with varying needs. At the top, a "Screen Readers and Voice Assistants" button enables users to activate these assistive technologies. Below, a "Text-to-Speech" button allows converting on-screen text to spoken words. The "Gesture and Motion Control" button allows users to enable intuitive navigation through gesture-based interactions, while the "Haptic Feedback" button lets users activate tactile responses for better sensory input. Upon saving any changes, a confirmation notification appears at the bottom, displaying "Accessibility settings saved successfully" to assure users that their adjustments have been applied.

## 5.3.8: Adjust Preferences

A diagram of a device

Description automatically generated with medium confidence

The “Adjust Preferences” wireframe offers a streamlined interface for customizing user settings. At the top, an "Eye-Movement Tracking" button allows users to turn this feature on or off for enhanced interaction. Below it, the "System Themes" button lets users choose between different visual themes to personalize their experience. The "Color Contrast" button provides options to adjust the contrast settings for better readability and visual comfort. An "Audio Settings" button allows users to configure their sound preferences, including volume levels and audio output. After making any adjustments, a confirmation notification at the bottom displays "Preferences saved successfully," reassuring users that their changes have been applied.

## 5.3.9: Audio Settings

A screen shot of a computer

Description automatically generated

The “Audio Settings” wireframe offers a streamlined and intuitive interface for customizing audio preferences. At the top, a "Volume Control" button allows users to adjust the overall sound levels. Below it, the "Audio Output" button lets users select their preferred audio output device, such as headphones or speakers. The "Language and Audio Tracks" button provides options for choosing different languages and audio tracks, enhancing the viewing experience. The "Spatial Audio Settings" button enables users to configure immersive audio features like Dolby Atmos. At the bottom, an "Exit to Film Details" button quickly returns to the film details screen, ensuring seamless navigation.

## 5.3.10: Spatial Audio Settings

A screenshot of a computer

Description automatically generated

The “Spatial Audio Settings” wireframe is designed to provide an immersive and customizable audio experience. At the top, a "Dolby Atmos Toggle" button allows users to turn Dolby Atmos on or off for 3D sound. Below it, a "Virtual Surround" button offers the option to simulate surround sound on stereo devices. The "Equalizer" button provides access to various presets and customizable sliders for fine-tuning audio frequencies. At the bottom, a "Submit" button allows users to save their adjustments, ensuring their preferred audio settings are applied.

## 5.3.11: Manage Profile

A diagram of a user profile

Description automatically generated with medium confidence

The “Manage Profile” wireframe offers a streamlined and intuitive interface for users to age their account details effortlessly. At the top, the "Update Personal Information" button allows users to edit their details, such as name, email, and contact information. Below this, the "Change Password" button provides a secure way for users to update their login credentials. The "Upload Profile Picture" button lets users personalize their profile by uploading a new photo.

## 5.3.12: Dashboard

A screenshot of a dashboard

Description automatically generated

The “Dashboard” wireframe offers a streamlined and intuitive interface, centralizing essential user functions for easy access. At the top, the "Explore Films" button invites users to browse the film library, facilitating discovery and selection of movies. Below, the "Manage Profile" button allows users to update their personal information and preferences effortlessly. The "Settings" button provides access to various customizable options, ensuring a personalized user experience. Finally, the "Log Out" button at the bottom ensures users can securely exit their accounts with a single click.

## 5.3.13: Film Details

A screen shot of a movie

Description automatically generated

The “Film Details” wireframe provides an intuitive and engaging interface for users to explore individual films. It features a "Continue" button at the top, allowing users to resume watching from where they left off. A "Watch Trailer" button offers a film preview, giving users a glimpse of the content before committing to viewing. The "Add to Watch List" button lets users save the film for easy access later. The "Download for Offline Viewing" button lets users download the movie without an internet connection.

## 5.3.14: Choose Film Format

A screen shot of a movie format

Description automatically generated

The “Choose Film Format” wireframe offers a straightforward and visually appealing interface for selecting a preferred film format. Prominently displayed buttons labeled "70MM," "IMAX 70MM," and "35MM" allow users to easily choose their desired viewing experience with a single tap. Below these options, a "Continue" button directs users to the next step in their viewing journey.

## 5.3.15: Pre-Buffering Screen

A screen shot of a screen

Description automatically generated

The “Pre-Buffering Screen” wireframe is designed to ensure a smooth and uninterrupted viewing experience. At the center, a prominent buffering progress bar visually indicates the film's buffering status, providing users with real-time updates. Below the progress bar, the "Wait" button allows users to choose to wait until the buffering is complete. Alternatively, the "Return Later" button lets users exit and resume buffering when they return. Once buffering is complete, the "Playback Screen" button becomes active, enabling users to adjust their playback settings.

## 5.3.16: Playback Screen

A screen shot of a screen

Description automatically generated

The “Playback Screen” wireframe is designed for intuitive and seamless user interaction. At the center, the "Playback Controls" button allows users to play, pause, rewind, and fast-forward through the film. Adjacent to it, the "Screen Size Adjustment" button offers options to maximize or minimize the screen size for an optimal viewing experience. Below, the "Subtitles and Audio Options" button provides access to turn subtitles on or off, select different subtitle languages, and choose audio tracks. Finally, the "Exit to Film Details" button lets users leave the playback screen and return to the film's details page, providing information about the movie and additional options.

## 5.3.17: Offline Viewing Downloads

A screenshot of a computer screen

Description automatically generated

The “Offline Viewing Downloads” wireframe provides an organized and intuitive interface for managing offline content. At the top, the "View Download Progress" button allows users to check the status of ongoing downloads, ensuring they can track their progress easily. Below, the "Manage Downloaded Films" button offers users access to a dedicated section to view, organize, and delete their downloaded films. Additionally, the "Playback Screen (Offline Mode)" button enables users to seamlessly switch to an offline playback mode, ensuring a smooth viewing experience without an internet connection.

## 6.0: Appendices

## 6.1: Glossary

IOS: The operating system utilized by Apple devices.

IMAX: A film format renowned for its larger-than-life screen size and pristine picture quality.

Dolby Atmos: An immersive sound format that channels sound through three dimensions.

Content Rating Requirements: Guidelines to allocate and rate content (films or video games) based on age appropriateness, content themes, and potential impact.

Augmented Reality (AR): A technology that blends virtual objects into the fold of reality, commonly experienced through a smartphone or an AR headset.

Virtual Reality (VR): A technology that places the user in a simulated environment, commonly used in a VR headset to provide a 3D visual experience.

Regulatory Compliance: Compliance with laws, regulations, and standards imposed by governmental authorities or regulatory bodies, ensuring that the product or service meets pertinent laws and industry guidelines.

Apple Vision Pro: A high-end augmented reality headset developed by Apple, which Yesteryear is optimized for.

Film Formats: The types of film presentations available in Yesteryear include IMAX, IMAX 70MM, 70MM, and 35MM.

Pre-buffering: A feature that ensures uninterrupted playback by loading a portion of the film before the user starts watching.

Machine Learning-Driven Content Personalization: Using machine learning algorithms to analyze user behavior and preferences to provide tailored content recommendations.

Screen Readers: Assistive technology that reads the screen's content aloud to aid users with visual impairments.

Gesture Control: An input method that allows users to interact with the system using hand movements.

Haptic Feedback: Tactile response provided by the system to enhance sensory engagement.

Multi-Factor Authentication (MFA): A security system requiring multiple authentication methods to verify the user’s identity.

Role-Based Access Control (RBAC): A method of regulating access to resources based on the roles of individual users within an enterprise.

AES-256: An encryption standard used for securing data at rest.

TLS 1.3: A protocol that provides privacy and data integrity between two communicating applications.

CI/CD Pipeline: Continuous Integration/Continuous Deployment pipeline used to automate the testing and deployment of applications.

Reality Composer Pro: A tool for designing and building 3D content and AR/VR experiences for Yesteryear.

Figma: A web-based design tool used for UI design and wireframing in Yesteryear's development.

## 6.2: References/Bibliography

Larman, C. (2004). *Applying UML and Patterns: An Introduction to Object-  
Oriented Analysis and Design and Iterative Development* (3rd Ed). Prentice Hall Professional Technical Reference.

Cameron, Andrew. Class Lecture, System Design, Seattle Pacific University,  
Seattle, Washington, June 3, 2024.