Absolute Cinemas

Software Requirements Specification

Version 1.0

9/18/25

Group 3
Renz Ryan Santillana
Luis Meza
Fabian Llamas

Prepared for
CS 250- Introduction to Software Systems
Instructor: Gus Hanna, Ph.D.
Fall 2023

Revision History

Date	Description	Author	Comments
9/18/25	Version 0.0	Group 3	First draft for Asst 1
9/25/25	Version 1.0	Group 3	Final draft for Asst 1

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Dr. Gus Hanna	Instructor, CS 250	

Table of Contents

Revision History ii

Document Approval ii

1. Introduction 1

1.1 Purpose 1
1.2 Scope 1
1.3 Definitions, Acronyms, and Abbreviations 1
1.4 References 1
1.5 Overview 1

2. General Description 2

2.1 Product Perspective 2
2.2 Product Functions 2
2.3 User Characteristics 2
2.4 General Constraints 2
2.5 Assumptions and Dependencies 2

3. Specific Requirements 2

3.1 External Interface Requirements 3
3.1.1 User Interfaces *3*
3.1.2 Hardware Interfaces *3*
3.1.3 Software Interfaces *3*
3.1.4 Communications Interfaces *3*
3.2 Functional Requirements 3
3.2.1 Feature #1: Search *3*

3.2.2 Feature #2: Payment Processing............ 3
3.3 Use Cases............ 3
3.3.1 Use Case #1............ 3
3.3.2 Use Case #2
3.3.3 Use Case #3............ 3
3.4 Classes / Objects............ 3
3.4.1 Object#1: Movie............ 3
3.4.2 Object#2: Ticket
3.4.3 Object#3: Seat............ 3
3.5 Non-Functional Requirements............ 4
3.5.1 Performance............ 4
3.5.2 Reliability............ 4
3.5.3 Availability............ 4
3.5.4 Security............ 4
3.5.5 Maintainability............ 4
3.5.6 Portability............ 4
3.6 Inverse Requirements............ 4
3.7 Design Constraints............ 4
3.8 Logical Database Requirements............ 4
3.9 Other Requirements............ 4

4. Analysis Models............ 4

4.1 Sequence Diagrams............ 5
4.3 Data Flow Diagrams (DFD)............ 5
4.2 State-Transition Diagrams (STD)............ 5

5. Change Management Process............ 5

A. Appendices............ 5

A.1 Appendix 1............ 5
A.2 Appendix 2............ 5

1. Introduction

The introduction to the Software Requirement Specification (SRS) document should provide an overview of the complete SRS document. While writing this document please remember that this document should contain all of the information needed by a software engineer to adequately design and implement the software product described by the requirements listed in this document. (Note: the following subsection annotations are largely taken from the IEEE Guide to SRS.)

1.1 Purpose

The purpose of "Absolute Cinemas" is to create a simple and reliable system for purchasing movie tickets online and through a mobile application and website implementation. The system will allow users to browse movie listings, view schedules, and securely purchase tickets. The SRS will guide the design and implementation process by describing functional and non-functional requirements.

1.2 Scope

*(*1) Absolute Cinemas will be the software product to be produced. This encompasses the mobile app versions to be released on major operating systems and the development of https://AbsoluteCinemas.com for supported browsers.

(2) Absolute Cinemas Software will grant users access to ticketing features for movies and digital media. These features will include:

- Browsing, Searching, and Previewing.
- Selection of tickets based on location, date, and time.
- Reservation of in-theater seating for a selected number of tickets.
- Confirm and Purchase selected ticket products.

This software will include general features unrelated to digital media, such as Sign-up and Login to user accounts, options to use Membership rewards, and caching of unfinished purchase progress.

(3) The application of Absolute Cinemas software catalogs and displays options for digital media to users. This provides users with benefits prior to visiting physical locations.

(a) The main objective of this software is to provide users with up-to-date information on tickets and seating, presenting benefits for users to encourage their purchase ahead of time.

(b) This software will provide purchase capabilities up to 1 hour before show times when available at the user's location.

(c) This software's goal is to effectively and efficiently process transactions from users to provide ticketing information.

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 appendices in the SRS or by reference to other documents.

- UI - User Interface

- DBMS - Database Management System
- API - Application Programming Interface
- QR Code - Quick Response Code used for ticket scanning
- 2FA - Two-Factor Authentication
- SRS - Software Requirement Specification

1.4 References

The references are:

- Absolute Cinemas Structural Model
- Absolute Cinemas Software and Electronic Model
- Absolute Cinemas Database Management System Model
- Absolute Cinemas Application Programming Interface Model
- Absolute Cinemas Vision Draft

1.5 Overview

From this point onwards, this SRS document will cover two major sections in the form of General Description and Specific Requirements.
As an overview, the General Description will inform the reader of the aspects of Absolute Cinemas software that can be described on paper. This includes the perspective from which the software is designed, the functionality of each software, the intended user base, and the constraints and assumptions accounted for. The Specific Requirements describe the data requirements needed to account for both Functional and Non-Functional Requirements that comprise Absolute Cinemas.

2. General Description

This section of the SRS should describe the general factors that affect the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.

2.1 Product Perspective

Absolute Cinemas is an independent software product that will connect with existing theater hardware and software. This software is dependent on systems such as scanners, payment processors, computers, and operating systems. Internet access is necessary for cloud-based servers to handle reservations, payments, and ticket verification.

2.2 Product Functions

The functions of all products will be the following:

1. Search and display movies with showtimes.
    a. Display movies based on user criteria and search terms.
    b. If no criteria are specified, display a default state.
2. Provide seat selection and availability updates.

    a. Allow for the selection of multiple seats.
3. Support secure ticket purchasing using multiple payment methods.
4. Generate and deliver digital tickets through email or app notifications.
    a. Delivery occurs within 1 minute of purchase.
5. Maintain customer profiles with order history preferences.

2.3 User Characteristics

This subsection of the SRS should describe those general characteristics of the eventual users of the product that will affect the specific requirements. (See the IEEE Guide to SRS for more details.)

1. The primary users are customers between the ages of 13-65 with basic smartphone or computer knowledge.
    a. Users will intend to browse, review, and purchase tickets from the application.
2. Secondary users include theater staff who will scan tickets and manage schedules.
    a. Secondary users will be allowed the basic features of primary users.
    b. Users can review customer and primary user information upon request.
3. 3. All users are expected to have internet access and a modern browser or mobile device.

2.4 General Constraints

This subsection of the SRS should provide a general description of any other items that will limit the developer's options for designing the system. (See the IEEE Guide to SRS for a partial list of possible general constraints.)
The general constraints of this application lie in the software's dependency on data. These are the identified constraints:

1. The application must run on Android, iOS, and web browsers.
    a. This application is subject to updates based on changing mobile platforms and the few supported browsers.
2. The system depends on third-party payment gateways.
    a. This application must always support at least one payment method at all times.
    b. Visa and Mastercard debit types will take priority among payment methods.
3. Movie data must be updated daily from theater management systems.

2.5 Assumptions and Dependencies

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

The dependencies of this software can be in one of two categories: First-Party and Third-Party. Assumptions for the success of this hardware will also be grouped in this section as they become relevant.
First-Party Dependencies are delegated to the appropriate physical location and property of the customer. Such dependencies are:

- Theaters will provide accurate movie schedules and seat availability.
- Theaters will maintain access to the ticketing database to validate tickets upon use.

Third-party dependencies are taken care of by partnered services and primary users.

- Users will have stable internet connectivity.
- Payment providers will maintain secure and reliable services.

3. Specific Requirements

This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project's software design, implementation, and testing.

Each requirement in this section should be:

- *Correct*
- *Traceable (both forward and backward to prior/future artifacts)*
- *Unambiguous*
- *Verifiable (i.e., testable)*
- *Prioritized (with respect to importance and/or stability)*
- *Complete*
- *Consistent*
- *Uniquely identifiable (usually via numbering like 3.4.5.6)*

Attention should be paid to carefully organizing the requirements presented in this section so that they may be easily accessed and understood. Furthermore, this SRS is not the software design document; therefore, one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interfaces are defined by the devices accessing the application. Accounting for these differences, mobile devices will implement a different interface from web browsers.

1. Mobile Devices
    a. The interface will provide simple navigation for browsing movies and purchasing tickets.
    b. Mobile and web versions will use a consistent design on mobile devices.
    c. Mobile app versions will display simple notifications in the app and push notifications when enabled.
    d. Customers will view digital QR codes for admission.
2. Web Browser
    a. Interface optimized for web browsers.
    b. Prompt users for desktop notifications.

3.1.2 Hardware Interfaces

- Ticket scanners must support QR code recognition
- Payment terminals must integrate with the theater's financial system

3.1.3 Software Interfaces

- The system will connect with third-party payment APIs

- It will use the theater's scheduling database for showtime updates
- It will integrate with email/SMS gateways for ticket confirmations.

3.1.4 Communications Interfaces

- All communication between client devices and the server will use HTTPS
- Internal server communication will use TCP/IP protocols

3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.2.1 Feature #1: Search

3.2.1.1 Introduction: This feature allows the user to type in criteria for a search in the movie database. Search will sort by similarity and exact matches.
3.2.1.2 Inputs: user keywords or filter (title, genre, time).
3.2.1.3 Processing: query database for matches.
3.2.1.4 Outputs: list of movies with details.
3.2.1.5 Error Handling: display "no results found" if no match exists.

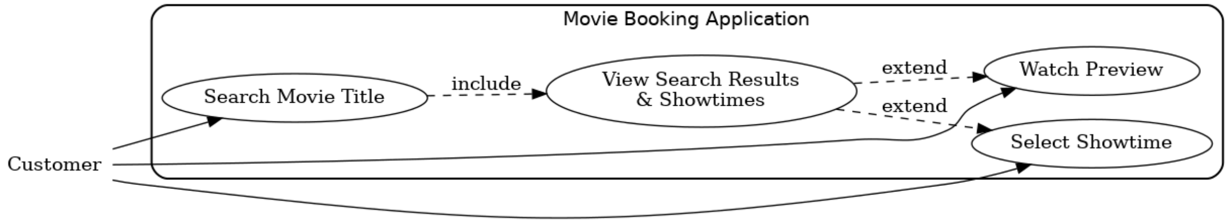
3.2.2 Feature #2: Payment Processing

3.2.2.1 Introduction: Payment Processing is a feature initiated by the user after selecting to check-out. Payment processing connects the user with third-party payment processing services.
3.2.2.2 Inputs: Confirm payment method used for purchase.
3.2.2.3 Processing: Verify the availability of seats in a showing through UX.
3.2.2.4 Output: Confirm payment through email with a QR code attached for check-in.
3.2.2.5 Error Handling: Reject payment if the selected seats are already occupied and or an invalid payment method.

3.3 Use Cases

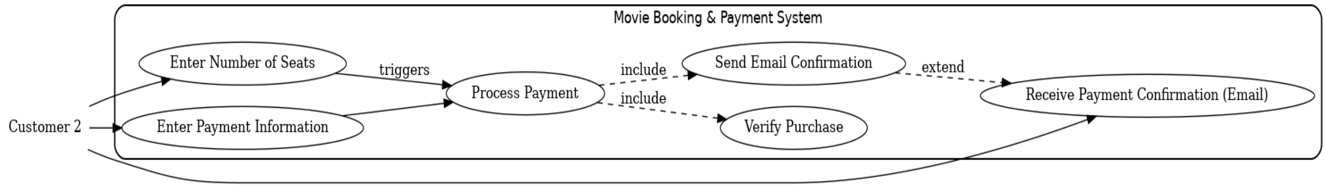
3.3.1 Use Case #1:

Actor: Customer 1
Description: The customer opens the application and enters a movie title in the search bar. The application displays search results and available showtimes to the user. The customer watches a preview of the movie and selects a desired showtime.



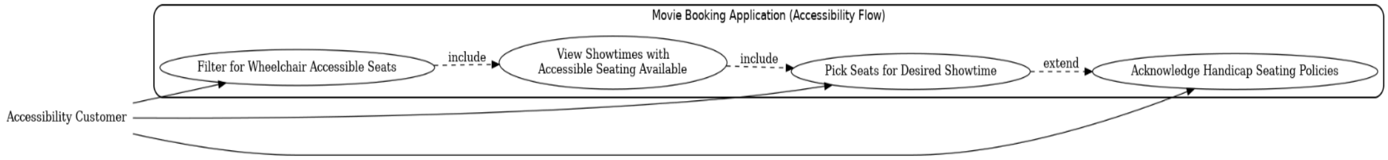
3.3.2 Use Case #2:

Actor: Customer 2
Description: The customer enters a valid amount of available seats to pay for with the appropriate payment information entered by the customer and will receive a payment confirmation through email once the purchase is processed and verified.



3.3.3 Use Case #3:

Actor: Accessibility Customer
Description: The customer filters a search for wheelchair accessible seats, application displays showtimes with handicap seating still available, customer picks seats for the desired showtime, customer is prompted to acknowledge handicap seating policies.


Movie Booking Application (Accessibility Flow)
Filter for Wheelchair Accessible Seats
include
View Showtimes with Accessible Seating Available
include
Pick Seats for Desired Showtime
extend
Acknowledge Handicap Seating Policies
Accessibility Customer

3.4 Classes / Objects

3.4.1 Object #1: Movie

3.4.1.1 Attributes: title, genre, duration, rating, showtimes
3.4.1.2 Functions: displayInfo(), getSchedule(), getLocation()
Reference to functional requirements and/or use cases

3.4.2 Object#2: Ticket

3.4.2.1 Attributes: ticketID, movie, seat, showtime, QRCode
3.4.2.2 Functions: generateQRCode(), validate()

3.4.2 Object#3: Seat

3.4.2.1 Attributes: rowNum, seatNum, accessible, availability
3.4.2.2 Functions: getNumber(), getRow(), getAvailability(), isAccessable(),

3.5 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often, these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transactions shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30-day MTBF value, etc).

3.5.1 Performance

The application will be considered a light workload for all supported devices. Focused on delivering through mobile devices, the application will not be memory-intensive or computationally complex. The website version of the software will be consistent with the performance requirements of the mobile application. Therefore, performance requirements are as follows:

- 95% of search queries shall return results within 2 seconds.
- Filtering search results will occur automatically during a search.
- Ticket confirmation shall be generated within 10 seconds after payment.
- Generating QR codes occurs within seconds of request.

3.5.2 Reliability

The movie database server will be responsible for maintaining reliability across all on-site servers. Redundancy is also planned within servers in the form of data backups and uninterruptible power supplies to deploy when needed. This implementation will allow the application to maintain these features:

- System uptime must be at least 99.5%
- Daily data backups will be maintained through a separate database with cloud transfer.

3.5.3 Availability

- The system must be accessible 24/7 with minimal planned downtime; such planned downtimes must be off-hours.

3.5.4 Security

- All sensitive data shall be encrypted during transfer and storage.
- Users must log in with a password and optional Two-Factor Authentication(2FA).
- All passwords have an expiration date in a month from password creation.
- New passwords must be unique and cannot be old passwords used in the past.
- Two-Factor Authentication will be sent through mobile notifications.

3.5.5 Maintainability

- Code shall follow a modular design for easier updates.
- Commentation in classes is a requirement to avoid confusion when accessing code.
- Documentation will be maintained for developers.
- Senior programmers must create a master key for the program to be created and secured through a physical or digital safe.

3.5.6 Portability

- The application must run on Android, iOS, and major browsers (Chrome, Firefox, Safari, Edge, and Brave).
- Database and server components should be deployable on cloud platforms such as AWS or Azure.

3.6 Inverse Requirements

*State any *useful* inverse requirements.*

3.7 Design Constraints

Specify design constraints imposed by other standards, company policies, hardware limitations, etc., that will impact this software project.

3.8 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, and data integrity? Etc.

3.9 Other Requirements

Catchall section for any additional requirements.

4. Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should trace the SRS's requirements.

4.1 Sequence Diagrams

4.3 Data Flow Diagrams (DFD)

4.2 State-Transition Diagrams (STD)

5. Change Management Process

Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved?

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

A.2 Appendix 2