THIS DOCUMENT CONTAINS DETAILINGS OF A TICKETING SITE, NAMELY AN SRS.
SCROLL TO VIEW

Ticketing System

Software Requirements Specification

Ticket Hub

January 31, 2024

Group #9
Amelia Grevin, Elliot Gambale, Hannah Ferdows

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

Revision History

Date	Description	Author	Comments
<date>	<Version 1>	<Your Name>	<First Revision>
02/14/24	<version 1>	<Amelia Grevin, Hannah Ferdows, Elliot Gambale>	<first revision>
02/24/24	<version 2>	<Amelia Grevin, Hannah Ferdows, Elliot Gambale>	<second revision>

Document Approval

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

Signature	Printed Name	Title	Date
	<Your Name>	Software Eng.	
	Dr. Gus Hanna	Instructor, CS 250	

Table of Contents

1. Introduction

This SRS document lays out the process to create an online platform for movie theater guests. You will find a step-by-step description for each engineering team to help construct their specific part of the site.

1.1 Purpose

This SRS is intended for the software developers tasked with creating a user-friendly platform to buy movie tickets.

1.2 Scope

Our products name: Ticket Hub
Product functions:

- display available movie options
- allow for movie selection
- display available seats
- select up to 20 seats
- select movie time
- select movie location
- select child/student/adult/senior
- press checkout button
- put in credit info
- email ticket
- be able to handle 1000+ users
- run in web browser
- block bots with over 20+ ticket requests
- customer feedback
- admin
- take in rotten tomatoes

Ticket booth is a user-friendly, accessible website that enables users to buy tickets to movies while being able to see reviews of given movies.

All clients of the theaters under the umbrella company have access to the website.

1.3 Definitions, Acronyms, and Abbreviations

SQL: Structured Query Language
API: Application Programming Interface
MacOS: Macintosh Operating System
MTBF: Mean time between failures
SRS: Software Requirements Specifications

1.4 References

Software development life cycle – Author: Gus Hanna
Source: Canvas Slides

useCases.ppt – Author: Gus Hanna
Source: Canvas Slides

How to Write a Software Requirements Specification

– Author: Gerhard Kruger & Charles Lane

Date: 1/17/23
Source: (Perforce.com)
https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document

1.5 Overview

The rest of this SRS contains detailed information of the functionality of the website as well as any constraints within the design process. It will also include the distribution of work among the software engineering team assigned to this project.

You can search below for needed information via the following sections of the SRS below: general overview, product functionality, requirement specifications, and use case models.

2. General Description

Find the following descriptions below: product perspective, product functions, user characteristics, general constraints, assumptions and dependencies.

2.1 Product Perspective

This site should work independently from other ticketing systems. It works similarly to popular sites such as amctheatres.com and fandango.com, but a more thorough description of “Ticket booth” site is described below:

Operation Inside Various Constraints:

System interfaces: System Interfaces should allow the site to open on any different hardware and adjust the display to fit the screen and function properly.
User interfaces: The user interfaces are split among 3 different screens; home, selection, and payment. The home screen will display all the available movies, the theater location, and a search bar. The selection screen will appear after a movie and location are chosen and the movie time

and the seat selection screen will appear. Finally, after these selections, the payment screen will appear and the user will be asked to type in credit card/payment information.
Hardware interfaces: The hardware interface is different from user to user. A website such as this one can be opened on a mobile phone, computer, ipad, ect…
Software interfaces:The primary software interfaces to be used are: intellij IDEA(a javaScript coding IDE)...
Communications interfaces:Communication interfaces include – communication between website and gps in order to locate nearest theater, communication between website and pre-existing movie trailers (in order to display on site), communication between site and online review sites (such as rotten tomatoes), communication between site and secure payment site (PayPal), communication between site and wifi and data, and communication between site and captcha (the authorizing service against bots). It should also be noted that communication with cloud databases will be likely and necessary in order to access any stored data regarding the movies and/or showtimes.
Memory: Gives user option to remember option, also gives an option to remember location when using, will need storage for all showtimes, seats, and who is buying what seat. Keep track of how many tickets are sold with a maximum of 20 tickets per theater. Also needs dedicated storage for reviews of movies as well as website experience. Needs to have dedicated memory to display reviews from critics of available movies.
Operations:User operations include the selection of tickets, selection of movie times, inputting payment info, selecting a theater location, as well as a seat.
Administration operations include updating available movies, updating movie times, updating available seats and unavailable seats, debugging any issues with the site.
Site adaptation requirements:In order to ensure the site to run on different platforms we plan on testing it and looking to implement the ability for the site to work properly under any wifi provider. Other means of extra precautions needed to be kept in mind depending on different adaptations of the site includes implementing a strategy for our programming language to cooperate with other possible ones it may need to work with, as well as to ensure that the site is not plagiarizing or copying without allowed access.

2.2 Product Functions

Product functionality depends on the user's needs. The main purpose of the platform is to allow users to purchase tickets via an online platform. This however, is not the only function of Ticket Hub. Ticket Hub provides users a platform to buy tickets, browse available movies in theaters, view online ratings, and view theater locations. The functionality of each individual stage of the buying process is described further in other sections of this SRS.

2.3 User Characteristics

This site should be accessible and available to all users regardless of their technical expertise and experience. It is meant to be extremely user friendly and intuitive.

2.4 General Constraints

Must follow all regulatory policies for selling tickets such that laws or illegal activity is broken or taking place. Devices must be able to have access to a browser as well as the internet. Website will require cookies from the user in order to process transactions. The website will also log activity by each user, logging what they buy and when they buy it. The website will also ask the user for their location in order to determine nearby theaters. All admins to the website will have access to these logs. Users on the website will be able to determine what movie they want to see and what kind of ticket they will purchase. This website should be able to operate with at least 1000 users at a time and still operate functionally. If the software were to fail, this would be costly to revenue as each minute is a potentially ticket lost. This website will not store the user's data, for safety to the user. It will also be programmed in such a way to keep the software decrypted from unwanted users, such as bots. This can be achieved by captchas to see if users are human.

2.5 Assumptions and Dependencies

Be weary of any changes in the way the website functions depending on what web browser is being used, and account for them if any. Also be wary of how website display may change depending on the hardware the user is using.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

HTML, CSS, JavaScript and React framework will help build a user interface and front end.

3.1.2 Hardware Interfaces

Mobile view, Laptop/computer view, iPad (tablet) are the most likely means of hardware interfaces we must look to work around and be compatible with.

3.1.3 Software Interfaces

This site will be built using Python and JavaScript. JavaScript will primarily be used to design the website itself, and Python will be used with all communication aspects.

- Django framework for maintenance, integration efficiency and code reusability

3.1.4 Communications Interfaces

MONGODB &/or MySQL – (outside databases communication languages in order to retract information and data). Another communication interface we may need to plan for is building or using a pre-existing API to access cloud computed information/

3.2 Functional Requirements

Find, in this section, specific features of the software project.

3.2.1 Extracting Review Data from Outside Sites

3.2.1.1 Introduction
Due to our website's unique planning to display reviews as seen on sites such as Rotten Tomatoes, below we will discuss in detail the logistics of doing so.

3.2.1.2 Inputs
The inputs necessary to extract are a rating out of 5 stars, and the top 2 reviews.

3.2.1.3 Processing
In order to get this data and process it, we must scrape it with an SQL server, and upload it to our cloud for easy access through APIs.

3.2.1.4 Outputs
Our website will therefore output said rating, an image displaying the stars visually, and the 2 reviews as quotes to reference at the bottom of the movie page, for those seeking such.

3.2.1.5 Error Handling
Some errors which may be intercepted during this process are a movie with no reviews, in which case will be caught by the error handling message saying “No reviews currently available”. Another is the movie not being on the second party website at all, where the same message could handle it. Lastly, it shall display this message again in case the data fails to load.

3.2.2 Updating Seating Maps for Tickets Purchased in Person

3.2.2.1 Introduction
Due to the need to consider that some tickets may be purchased in person, while the seating map may fill according to users’ choices online, we must also implement the functionality for employees to fill the map as tickets are sold at the theater.

3.2.2.2 Inputs
The inputs will be a seat number, and seat row to fill for each ticket.

3.2.2.3 Processing
Given that the site will need an administrative side, in order to make any adjustments, updates, etc. there should also be an employee view where seat maps can be filled and adjusted.

3.2.2.4 Outputs
The seat map should ideally update live to the website as employees adjust and fill it, showcasing seats that are taken (by grayed out colors and other signifiers).

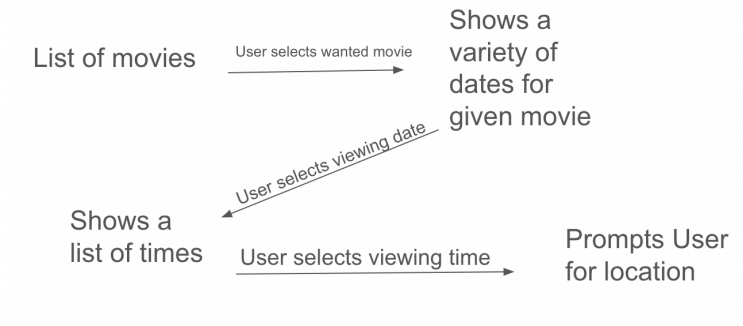
3.2.2.5 Error Handling
In the case in which something crashes, or parts are not communicating well with each other, the system is to display, “Seat map currently not available. Check with your local theater.” This way,

customers will have to be seated in person until the map is up and running again, and server connection is restored. One other error to handle is wanting to purchase a certain number of tickets, when there are less seats available than tickets desired. In this case as well, there will be an error message prompting the user to return to the last screen and try again with a different number of tickets, or to be redirected to the movies page in order to choose a new showing.

3.3 Use Cases

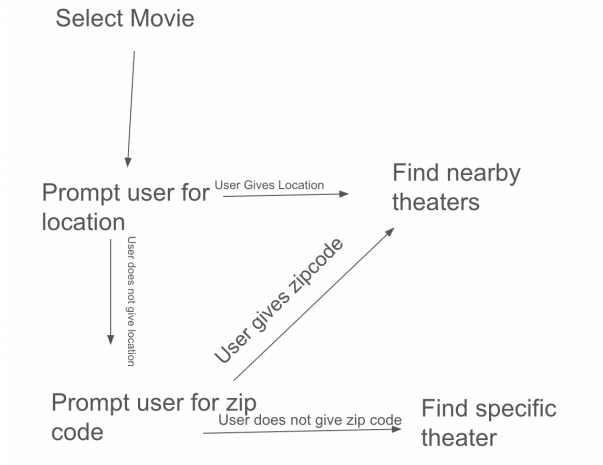
3.3.1 Use Case #1

The first use case for the website would be to display all available movies, and their showtimes each day into a month in advance, so customers can plan ahead and come accordingly.



3.3.2 Use Case #2

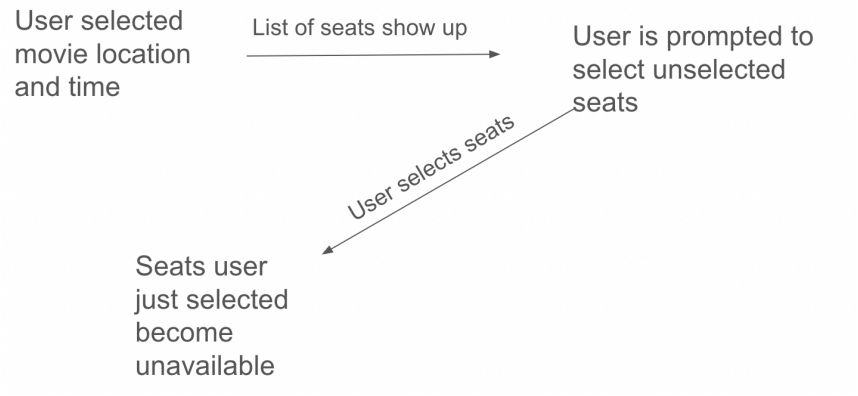
The second use case for our ticketing website would be for purchasing a ticket – users will need specifications! like which location they're looking at and choice of movie.



3.3.3 Use Case #3

The final use case for the user is for them to view specific seats/ a seating map and whether certain seats that they'd like are open for them to buy or sit in. (Also, if there are a certain

number of seats open together in the case in which someone wants to come and sit with their friends)



…

3.4 Classes / Objects

3.4.1 User / User Profile

The class ‘user’ will consist of the class object userProfile, which will be used to access methods and *functions* regarding view available tickets, purchase tickets, view upcoming showtimes, all locations available, and seat maps for a specific movie and time as chosen by the user.

Movie Details / Movie

The class ‘movieDetails’ will consist of the class object movie, which will be used to access method and *functions* such as the date of the specific movie, its showtimes, the name, number of seats available, whether the movie has started yet or not (after a showtime is chosen) and its reviews – which will be externally extracted from movie review websites and stored as one of its components.

<Reference to functional requirements and/or use cases>

…

3.5 Non-Functional Requirements

3.5.1 Performance

Our application should be able to run and to load tickets efficiently, regardless of the user count. During normal run time, our site should take no more that 85% of CPU usage, ignoring any errors that may occur during high capacity. To lessen the chance of this happening, the implementation of a 1000 user cap will be placed so that the performance will not be drastically affected.

3.5.2 Reliability

Ensure the website remains functioning while the user is active. Also, assign a specific period of maintenance time for the admin to ensure that the site is up and functioning all other times. Make sure the site can signal admin to fix any bugs/errors. The site should not crash any user's device. We look to add a cue to ensure that no user is kicked from the site in case of too many users, and there is an orderly line.

3.5.3 Availability

application and web browser available, should be available on as close to 100% of browser applications as possible
– (browsers: Google, Safari, Firefox, Bing – research more browsers that are commonly used and can be necessary to achieve ~100%)

3.5.4 Security

Secured checkout, and basket/cart available for the user to add their ticket to. Can use 3rd party security measures – such as secured paypal checkout. Otherwise data can be encrypted and stored in a cloud database. Also authorizations such as reCaptcha should be involved to ensure identities and human users.

3.5.5 Maintainability

The application should have an automatic check for updates as well as automatically administer them across all platforms (live editing)

- Has routine checkup dates (ex. every 2 weeks required to check on functionality and software versions)
- Regular updates to the website in order to tweak minor bugs and keep the website modern and maintained.

3.5.6 Portability

Website will be designed on macOS, but will be compatible with all operating systems.

3.6 Inverse Requirements

Website will have no more than 1000 users at a given time, if so, an error message occurs and the queue will begin forming.
Website will not sell more than 20 tickets for a given movie, if exceeded will portray a sold out message.

Section 1-3 conclusion…

The above SRS describes the baseline needed to start an online ticketing system. You will find, in this document, information provided for; the creation, functionality, and the platforms needed to construct the site. Section 1 explains more thoroughly the contents of the SRS sections 2 and 3. Section 2 describes in much more detail, the functions of the site, and an overview of all the site operations. Finally, Section 3 explains more dynamics of the site, specifically, what it can and cannot do and what outside information/connections are allowing it to successfully run smoothly and efficiently.

3.7 Design Constraints

Ticket hub will run into a few constraints imposed by other standards, these standards include; company policies and hardware limitations. Below are constraints listed for each of these standards.

Company policies:

- Privacy Policy: this policy ensures that the privacy of the user remains intact, that no credit card information is at risk of obstruction and personal information should remain private.
- Terms of use: When users access the site, it should be noted that their information should be truthful and accurate. The password information should not be shared and is in the hands of the user, it is not the responsibility of the admin. Ensure good user conduct, there should be no abuse of the site. In any of these cases there should be a user ban for a predetermined time.
- ticket policy: return policy for tickets should be declared and rules should be strictly followed by users and enforced by administration. Simple ticket rules should ensure there is only one ticket given out for each seat and a maximum number of tickets should be given

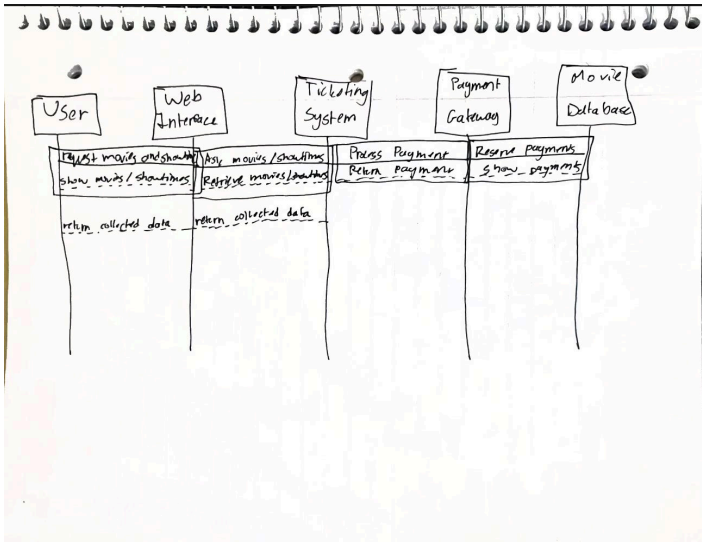
Hardware Limitations:

Hardware limitations often depend on the user and admin devices. The site should not overwhelm a user's device/cpu and should be scaled to adjust to any different model/brand/company's device. Software should be formatted to fit any screen size/ be available in different formats for phone vs computers. Hardware constraints during production period/admin accessing, may include; memory storage space, internet connection, battery life, and heat management. One major hardware restriction for both admin and users is wifi/connectivity. The user should be connected to the internet to access the site.

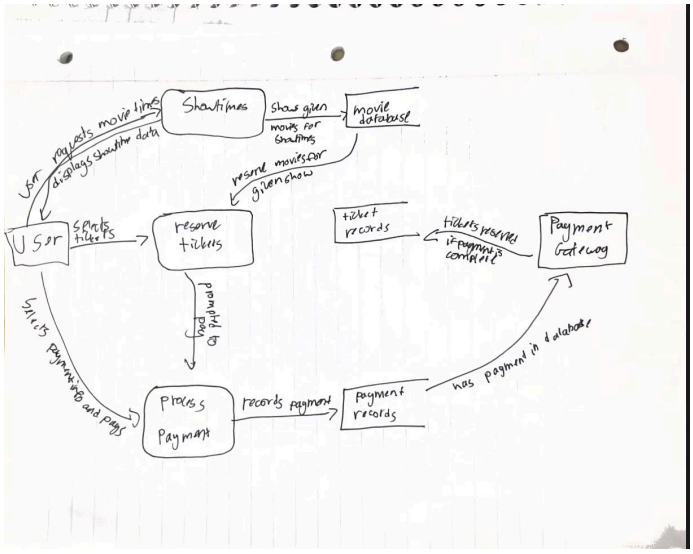
4. Analysis Models

…

4.1 Sequence Diagrams

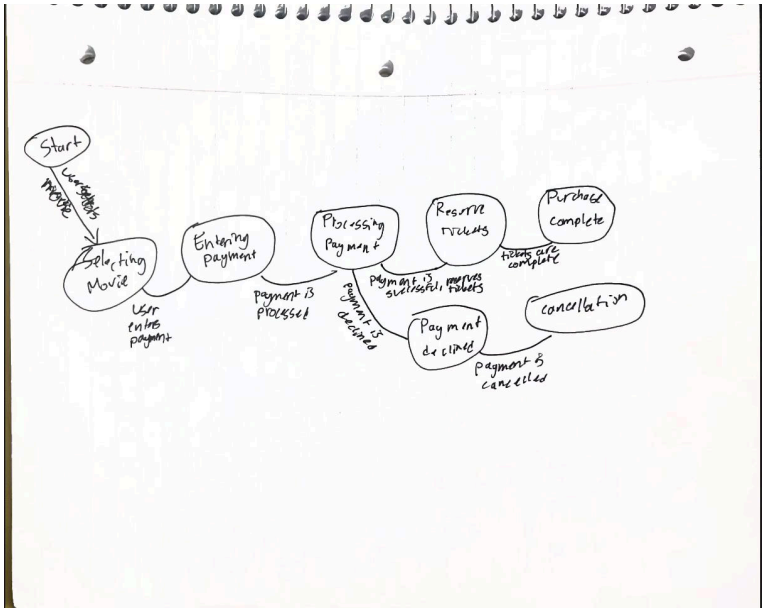


4.3 Data Flow Diagrams (DFD)



4.2 State-Transition Diagrams (STD)

Ticket Hub



END