Tiny Tix

Software Requirements Specification

1.3

3/27/2024

Group 10
Audio Antuna, Logan Moreno, Jake de los Reyes

Prepared for
CS 250- Introduction to Software Systems
Instructor: Gus Hanna, Ph.D.
Spring 2024

Revision History

Date	Description	Author	Comments
2/14/2024	Version 1.0	Jake de los Reyes, Audio Antuna, Logan Moreno	First draft through 3.6
2/28/2024	Version 1.1	Jake de los Reyes, Audio Antuna, Logan Moreno	First draft 3.6 through 4.3
3/13/2024	Version 1.2	Jake de los Reyes, Audio Antuna, Logan Moreno	Minor changes to UML
3/27/2024	Version 1.3	Jake de los Reyes, Audio Antuna, Logan Moreno	Added 4.4 Data Management

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

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references, and an overview of the SRS. The aim of this document is to gather and analyze the functions of the Tiny Tix software system and give an in-depth insight by defining the problem statement in detail. Additionally, it also focuses on the capabilities required by stakeholders and their needs while defining high-level, functional product features. The detailed requirements of the Tiny Tix software are provided in this document.

1.1 Purpose

The purpose of this document is to aggregate the necessary design elements of a ticketing software system, ensuring that all parts of the ticket purchasing process are accounted for. Additionally, it assists developers and designers in the software delivery lifecycle.

1.2 Scope

All relevant processes related to viewing available seats, purchasing seats, and receiving confirmation as well as proof of purchase of seats are included in this scope. This SRS will describe all important design requirements and will help developers to thoroughly understand possible niche errors that may arise. This will help to prevent accidental programming mistakes from hinderring the user experience and will help to deter users with negative intentions from exploiting flaws in the system.

1.3 Definitions, Acronyms, and Abbreviations

Purchase Window	A short time during which the user has a temporary reservation for the tickets that they are viewing and these tickets show up as unavailable to other users.

1.4 References

No references as of version 1.0.

1.5 Overview

The rest of this SRS describes specific hardware and software features that must be incorporated into the end product. It provides constraints as well as justification for these constraints that need to be followed to create a successful product. Section 2 describes the product from both a user and developer perspective, section 3 includes the aforementioned requirements, and section 4

provides all important models of analysis related to the product. Section 5 describes the necessary process to change this SRS and the appendix provides supplementary information for the reader.

2. General Description

This document contains the problem statement that the current system is facing which is hampering the growth opportunities of the company. It further contains a list of the stakeholders and users of the proposed solution. It also illustrates the needs and wants of the stakeholders that were identified in the brainstorming exercise as part of the requirements workshop. It further lists and briefly describes the major features and a brief description of each of the proposed systems. The following SRS contains the detailed product perspective from different stakeholders. It provides the detailed product functions of E-Store with user characteristics permitted constraints, assumptions and dependencies, and requirements subsets.

2.1 Product Perspective

This website will be similar to other sites for purchasing tickets. It will be comparable to online shopping sites in the fact that it will show all available products (seats, in this case) and will have a confirmation page for users once they've selected products similar to a shopping cart page.

2.2 Product Functions

This website will have 3 key functions: displaying available movies and showtimes based on location, displaying available tickets for said movies, and providing a purchasing process for selected tickets during the purchase window. The site will display all show times for currently playing movies on the home page. Clicking on a showtime will pull up all available seats in a map-format such that users are able to easily tell the location of seats in reference to the screen of the theater. The process of purchasing tickets will take place after users have selected seats and have entered the purchase window. During this time, users are given the option to input personal information (name/email) and payment information in order to confirm their purchase, after which they will be sent a confirmation for this purchase via email.

2.3 User Characteristics

The main demographic of users who will utilize this website are those who want to view and purchase movie tickets online. Due to its simplistic and easy-to-use interface, users will be able to save time, stress, and inconvenience of buying their tickets in-person. Customers continue to use Tiny Tix because of its generous reward system. After a certain quantity of tickets are purchased, users will obtain benefits including, but not limited, to discounted and free movie theater items.

2.4 General Constraints

The Tiny Tix software will operate as a website accessible through major web browsers such as Chrome, Firefox, Microsoft Edge, etc. The program must be efficient enough to be used over the web without local download, and secure enough to manage users' private information such as credit card numbers. As a website, the Tiny Tix software should be simple enough to run on smartphones and other underpowered devices.

2.5 Assumptions and Dependencies

This website will use services provided by Authorize.net to process transactions. This will allow for easier transactions and will provide a trustworthy middleman for users unfamiliar with the site.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Software will implement a pleasing and easy-to-navigate user interface, with navigational buttons to explore movies, showtimes, theatre locations, customer support, and account information.

3.1.2 Hardware Interfaces

Given that this site needs to be updated in real time, all servers will be run through ethernet to provide the most reliable connection for users and to minimize latency.

- Individual performance will vary depending on the user's device as the site will be designed for a multitude of devices

3.1.3 Software Interfaces

1. The website will communicate with the user information network to confirm the identity of users.
2.The website will communicate with the company availability network to check for seat availability.
3.The website will communicate with Tax system to dynamically calculate tax based on location and local tax laws.
4.The website will communicate with Authorize.net to confirm user purchases.
5.The website will communicate with GPS services to get the user's location.

3.1.4 Communications Interfaces

HTTPS will be used for all communication.

3.2 Functional Requirements

3.2.1 Purchasing a Ticket

3.2.1.1 Introduction
-After finding their desired film and showtime, users will be able to purchase their tickets for the show.
3.2.1.2 Inputs
-Select a movie
-Select a showtime
-Select seats
-Account information
-(Optional) Promo code
-Payment information via Authorize.net
3.2.1.3 Processing
-Check for seat availability
-Authorize payment information
-Mark tickets as unavailable
3.2.1.4 Outputs
-Return a successful purchase screen to the user
- Virtual ticket is converted into a QR code
- Option to display, print, or share the virtual ticket through email or SMS
3.2.1.5 Error Handling
-If seats are no longer available, return user to the seat selection screen with an error message

3.2.2 Creating an Account

3.2.2.1 Introduction
-Users will be able to create and log into their accounts in order to store payment and personal information, as well as purchase history and currently active tickets.
3.2.2.2 Inputs
-Email Address and/or Phone Number
-Password
-Personal Information (Name, Age, Address, Gender)
3.2.2.3 Processing
-If an account already exists for the given email/phone number, log in.
-If no account exists, create a file in the software's cloud servers containing the personal information of the user.
3.2.2.4 Output
-Bring the user to their account page after creating/logging into their account.
3.2.2.5 Error Handling

-If an account already exists and the input password is incorrect, send the user an error message and prompt them to enter a different password.

3.2.3 Holding a Ticket for Purchase

3.2.3.1 Introduction
-During the ticket purchasing process, the user is prompted to select their seats, this feature would hold the ticket from being selected by other users for a limit of 10 minutes to avoid multiple users from trying to purchase the same seats.
3.2.3.2 Inputs
-The selected movie, showtime, and seats of the user.
3.2.3.3 Processing
-Temporarily mark the desired seats as sold
-Begin a 10-minute timer before marking them as available (unless the user follows through with the purchase).
3.2.3.4 Output
-Notify user that their seats are reserved for exactly 10 minutes.
-Provide a countdown timer in the user interface.
3.2.3.5 Error Handling
-If two users attempt to reverse the same seats at the exact same time, return an error message to both users and prompt them to select again.

3.3 Use Cases

3.3.1 Use Case #1

With just a simple desktop or mobile device, users can purchase tickets ahead of time for themselves or others. Tiny Tix is available on all major operating systems and web browsers through a search engine. Users who purchase their tickets through Tiny Tix will receive a virtual rendition in the form of a QR code. Once the individual arrives at their selected movie theater, they are able to scan the given QR code to verify admission.

3.3.2 Use Case #2

Users can browse the available movies by using the navigation features displayed on screen. Due to the website having an auto-update feature, the availability of movie times and theater seats will be relative to current time. Users are free to look through the entire selection of options based on their location.

3.3.3 Use Case #3

Those who create their own account on Tiny Tix have access to a rewards program. Users will automatically obtain points by purchasing movie tickets on the software. When viewing the personal profile section, users may check their points balance or choose to redeem their points on movie theater items.

3.4 Classes / Objects

3.4.1 User Account

3.4.1.1 Attributes
-User's personal information
-Contact information
-Log in details
3.4.1.2 Functions
-Store user's personal information
-Store purchase history
-Store currently active tickets for usage
-Store user's payment information
-Update user information
-Update payment information
-Update log in information
Required for Functional Requirement of **Creating an Account**

3.4.2 Theater

3.4.2.1 Attributes
-Max number of seats
-Showtimes
-Specific seats available for each showtime
-Price of each showing
-Type of showing (e.g. 3D, regular, subtitled, etc)
3.4.2.2 Functions
-Determine if seats are available for each showtime, how many, and where each seat is
-Provides all necessary information for **Purchasing a Ticket**

3.5 Non-Functional Requirements

3.5.1 Performance

The software must be running on a system that can withstand the use of ~1000 active users, with the ability to process every request ranging from logging in to purchasing and reserving tickets. Additionally, the system must contain enough storage to store the information of all tickets, and the personal files for every user account containing their personal information.

3.5.2 Reliability

The software must be running on a system in which the measured downtime does not exceed 5 minutes per day, otherwise the system must be upgraded for the sake of business and user experience.

3.5.3 Availability

The software will be made available through the internet and therefor will be accessible to all users regardless of location.

3.5.4 Security

With the Tiny Tix software having access to the sensitive personal information of its users, the servers used to store this information must be adequately protected and at the very least within the scope of legal requirements.

3.5.5 Maintainability

The Tiny Tix software should be run via a server that is easy to upgrade and maintain in the event of more-than-anticipated user traffic.

3.5.6 Portability

Being a website, the Tiny Tix software does not need to be run on a portable device and can therefore be run via a standard server.

3.6 Inverse Requirements

1. The site must be secure and not vulnerable to data breaches and malicious attempts by users.
2. The site must be able to accurately validate user information.
3. The site must be able to provide accurate error messages to users.
4. The site must provide accurate permissions to users and to administrators.
5. The site must be efficient for the vast majority of users.

3.7 Design Constraints

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

1. The software assumes that users are on a device that has access to an internet connection and has the system requirements run a web browser.
2. The software assumes that the user limit will not be exceeded and cause a server overload.
3. The software must satisfy the policies stated in the General Data Protection Regulation (GDPR) to ensure data privacy.
4. The software requires a collection of databases with relevant movie theater data to operate.

3.8 Logical Database Requirements //Don't do this yet

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

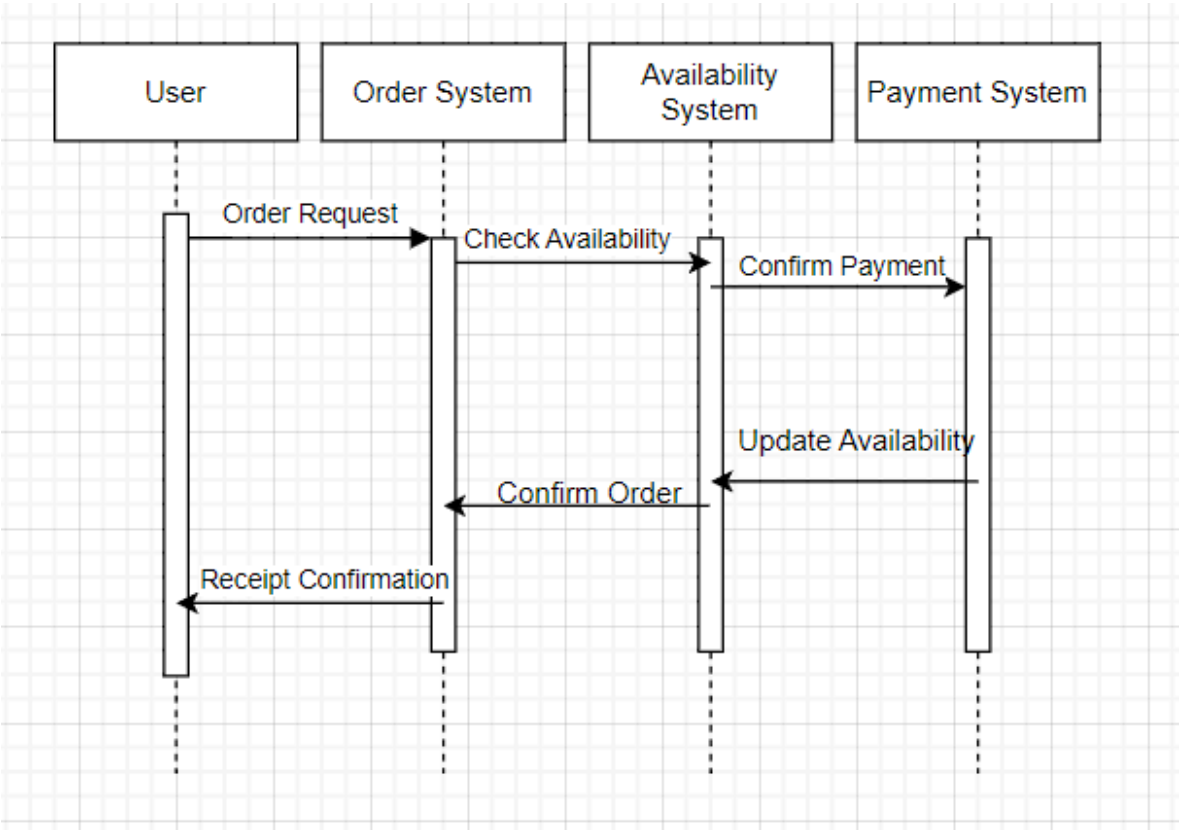
3.9 Other Requirements //Don't do this yet

Catchall section for any additional requirements.

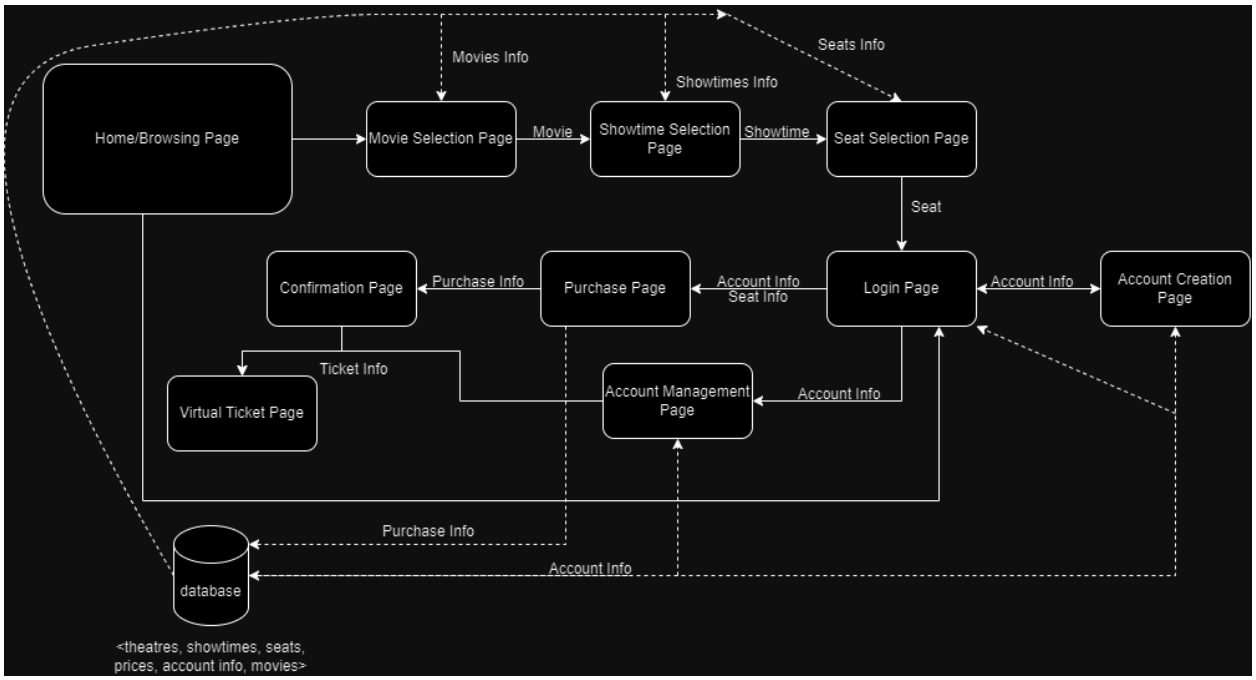
4. Analysis Models

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 be traceable the SRS's requirements.

4.1 Sequence Diagram

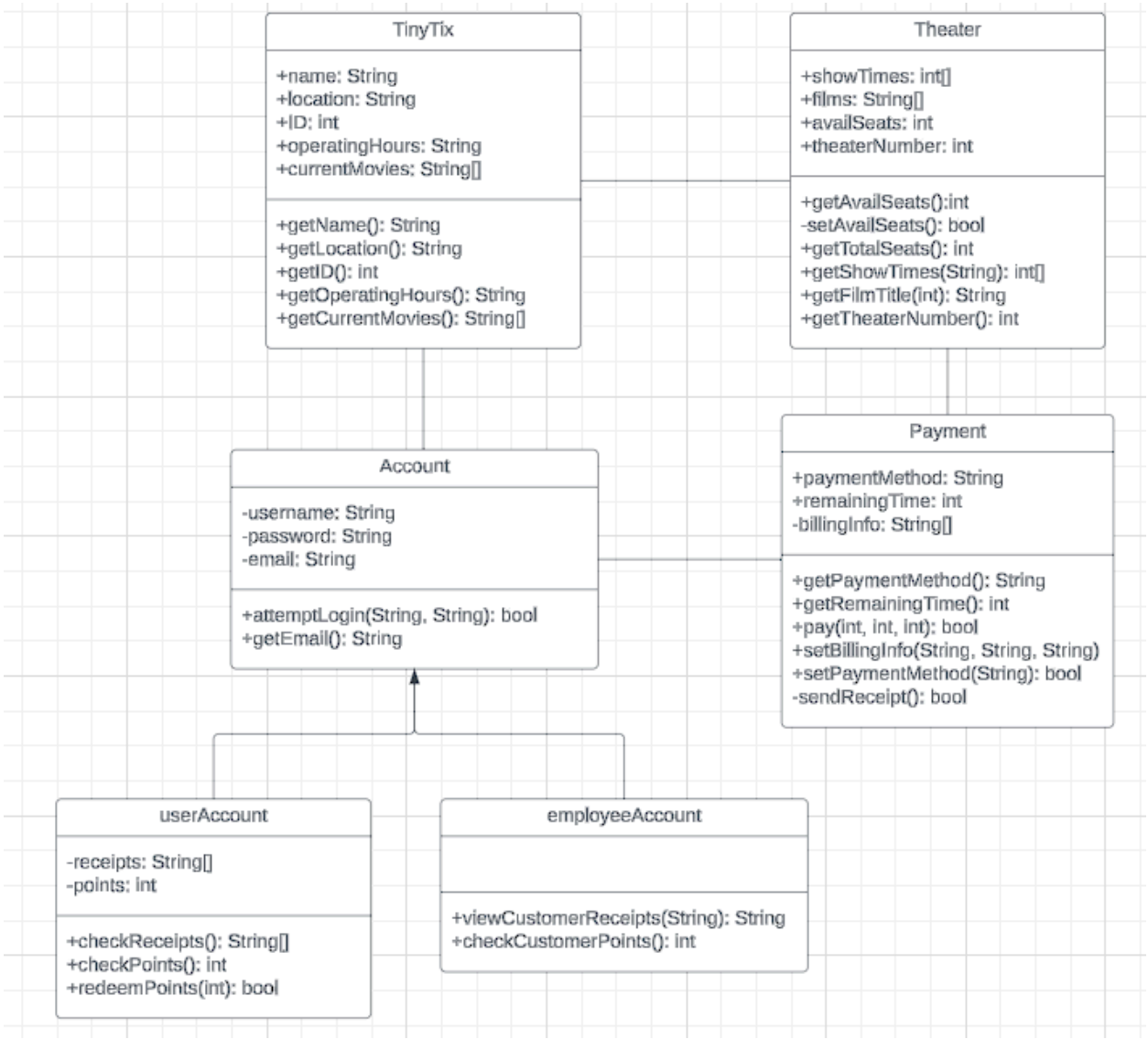


4.2 Software Architecture Diagram (SWA)



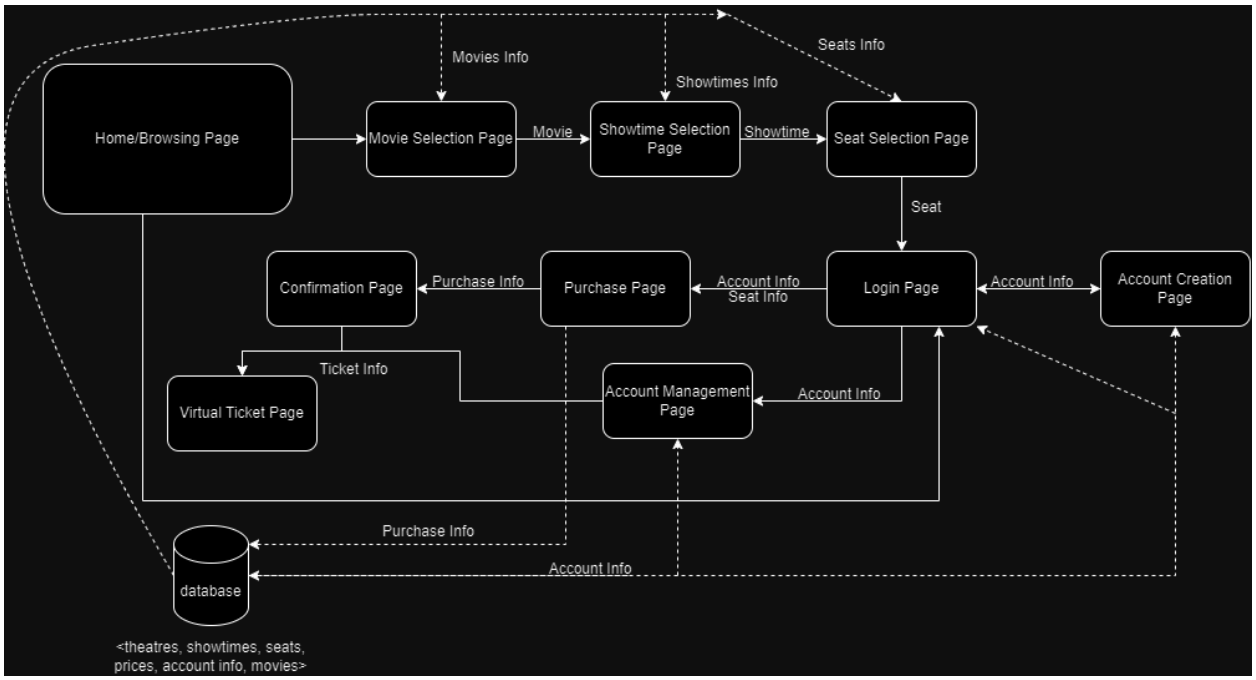
The software architecture diagram (SWA) above depicts the basic architecture of the Tiny Tix program. It begins with the Home/Browsing page, from which the user can access the movie selection and login pages. The Movie Selection page pulls Movie information from the database and after user selection, sends the movie information to the next page, the Showtime Selection Page. This page pulls showtime information from the database and after user selection, sends that information to the next page. On the Seat Selection Page, seat information is retrieved from the database, and after user selection, moves to the login page, sending the seat information with it. If the user does not have an account they will be directed to the account creation page. The account creation page stores the user's information in the database and returns the user to the login page. On the login page the user would input their login information and the page would retrieve account information from the database. After login, the user would be directed to the purchase page, retrieving account and seat information from the previous page, and storing the purchase information in the database. After purchase it would move to the confirmation page, and subsequently the virtual ticket page. Alternatively, after the login page, the user can access an Account Management Page which can be used to access previously purchased virtual tickets.

4.3 UML Class Diagram



This UML diagram shows the way that data is stored, accessed, and modified within the website. TinyTix is the “hub” of the program. It stores basic information about the theater such as its location and operating hours. Account class stores login information from the user and is an abstract class with subclasses userAccount and employeeAccount that allow users to access their own receipts/points and employees to check customer receipts/points, respectively. Theater class holds information about each theater such as showtimes and available seats (in this case, “theater” is referring to the individual rooms that films are shown in). Payment class contains all the information necessary to allow a customer to make a payment and calls methods in other classes in order to send out a receipt following a payment (for example getEmail() within Account class to find where to send the receipt and getTheaterNumber() within Theater class to tell the customer where their showing is).

4.4 Data Management



This diagram is reused from section 4.2. For our use-case, we found it unnecessary to make significant changes to the diagram. Solid lines show how users can navigate through the website and dotted lines show how data travels through it. Additionally, solid lines can double as a data channel and are indicated via labels for the data traveling during user navigation. In this case, the account creation and account management pages provide account information to the database and the purchase page provides purchase data to the database which is used to update seat availability which is sent to the seat selection page to display it to the customer. All data will be held in one database and will be managed using SQL through Salesforce. Only one database is necessary with how small the business currently is, however, more databases may be considered with future growth. Salesforce was chosen because it is a much more cost-effective choice than other options such as Microsoft Dynamics (the comparable option for small businesses) which provide excessive ways to access and modify data that are unnecessary given the current scale of the company and would add unnecessary complexity (Microsoft Dynamics Business Central Essentials starts at $70/user/month whereas salesforce starts at $25/user/month).

Especially with the number of data needed to be processed, a scalable and efficient database is a must. To accommodate the abundance of data involved in the e-commerce website, SQL was chosen over nonSQL. The tradeoff for this is that nonSQL would typically out-scale and be more cost-effective than SQL, but this is negated because the business only uses one database. If needed, a hybrid approach would be considered in the future.

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