Software Requirements Specification

for

TicketBooker

Movie Ticket Booking System

Version 1.0 approved

Prepared by the “A-Team”

Ken Cook, George Henning, Nicole Williams, Brian Tuovila, Aidan Smith

CPS298 Sect 01 Winter 2021

May 1, 2021

Table of Contents

1. Introduction

1.1 Purpose

The intended product is web-based software which sells movie tickets online for a small-sized theater complex. Our example has five screens and seven movie titles.

1.2 Product Scope

- This online movie ticket booking system “Ticket Booker” allows registered users to purchase tickets online.
- It is assumed that this will lead to increased sales, due to its convenience. It allows people to purchase tickets from anywhere with an internet connection.
- Most large cinema chains already have online sales systems in operation, so this product will be targeted to smaller cinemas who do not yet sell tickets online.

1.3 References

We are getting some ideas from one of our competitors who seems to have a higher budget.

https://www.movietickets.com/

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

This online ticket sales system 'Ticket Booker' is a *follow-on* member to an in-house ticketing system. It is for demonstration purposes only. If a theater owner likes our product and chooses to purchase the system, a level of support (80 hours max) will be included with the purchase to merge both the in-house and online sales into one relational database management system (RDBMS). If necessary, we will provide an interface to the in-house RDBMS from MySQL, which is what *Ticket Booker* uses.

2.2 Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

- Sign into account, or sign in as 'guest'.
- User Interface displays movie names, times, and prices of five screens in one complex.
- Once a Showing (movie title, date, and time) is clicked, a seating chart will be displayed, showing seats already purchased.
- Once a seat is selected, (the user name is assigned to that specific seat) and is stored in the online database.

2.3 User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

- **Online Customers:** Guests can browse the offerings and add things to a cart, but need to be registered in order to make a purchase and view purchases.

- **System Administrator or** in-house IT personnel. The software allows them to:
    - Handle routine tasks such as adding / removing movie titles, updating ticket prices, and scheduling showings.
    - Report the amount of online sales and other statistics to the theater owner.
    - Troubleshoot any problems or bugs that arise with the UI or the database.
    - Create new features, such as a mailing list or a rewards club.

2.4 Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

It operates on a *DigitalOcean* Droplet running Ubuntu Linux, MySQL, Tomcat 8.5 & 9.0, using JSP and Bootstrap 5.

2.5 User Interface

The user interface will be accessible from almost any modern web browser and will use http protocol.

3. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

3.1 Administration Features

Administrators can view sold tickets and the current movie showing list.

3.2 Online Guests Features

Online guests, users that are not logged into the system, can view movie listings, and select seats adding them to a cart. They can then login.

3.3 Online Customer Features

Online customers are logged into the system.
They can:

1. View movie listings
2. Select seats adding them to a cart
3. Edit cart contents, selecting multiple seats in multiple movies, on different dates.
4. Complete purchase
5. View purchased tickets
6. View scannable ticket

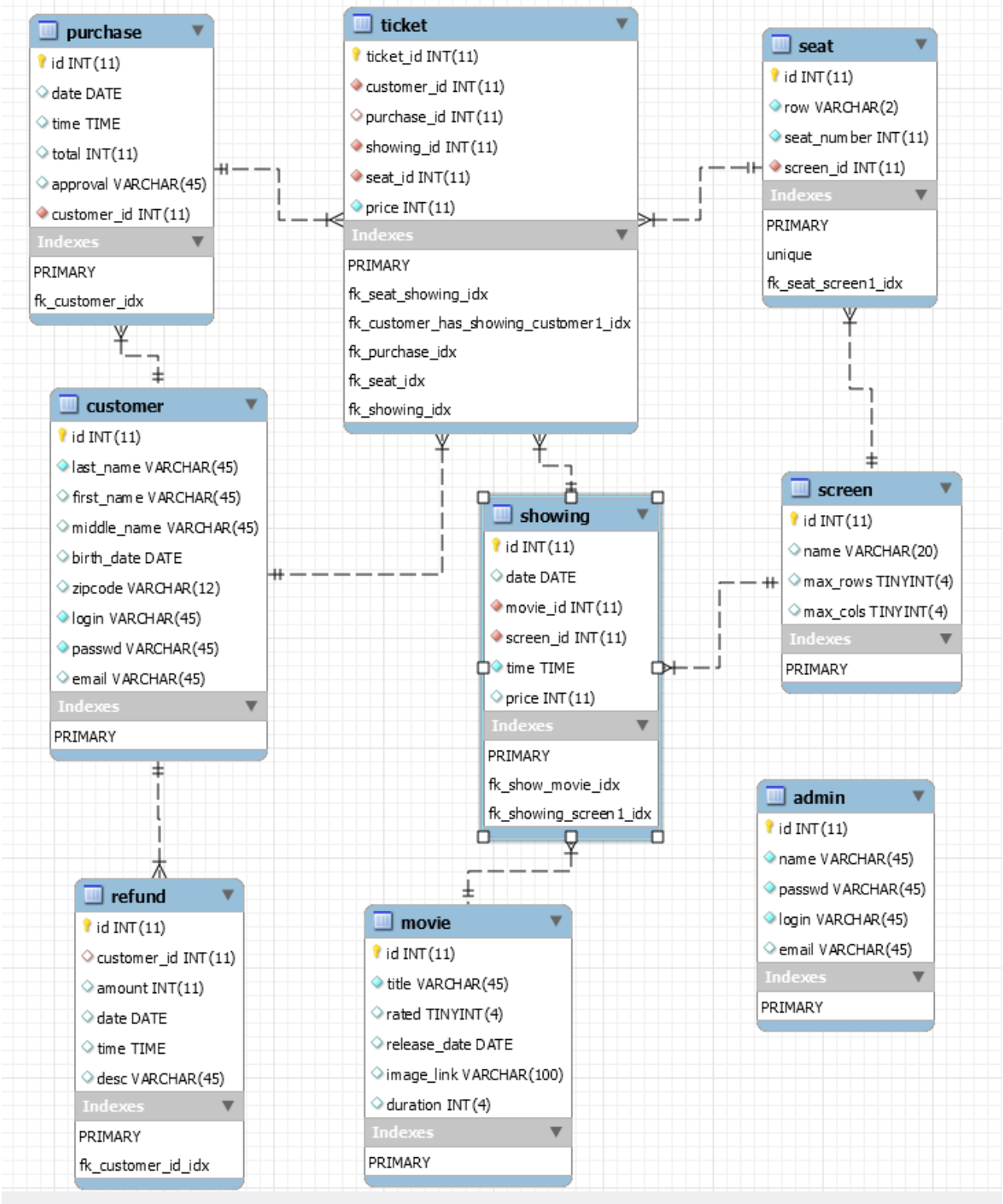
4. Other Nonfunctional Requirements

4.1 Business Rules

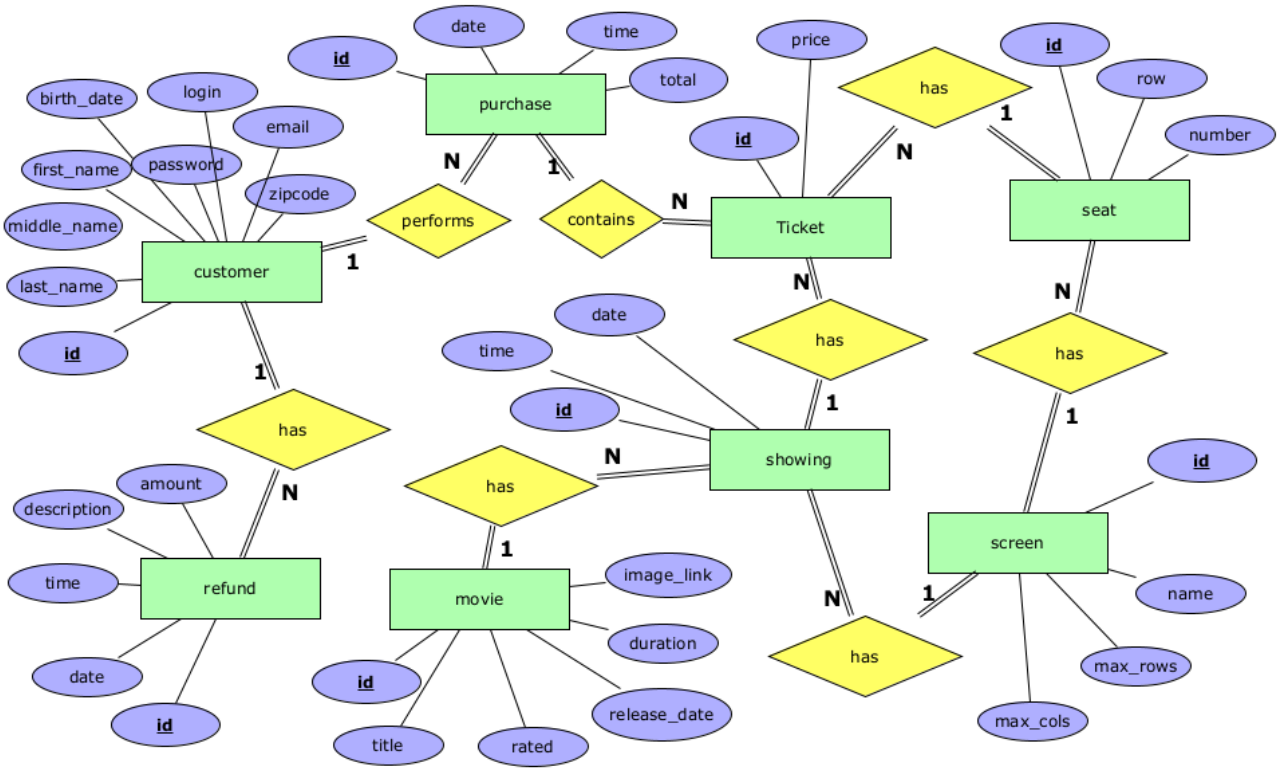
<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

- Each seat is represented by one ticket. For example, if someone intends to purchase four seats, then they would select four tickets, put them in a cart, and then click ‘purchase’.
- Once a purchase number has been created, the seats are color coded as unavailable for purchase by anyone else.

Appendix B: Analysis Models



Database Diagram



TicketBooker ERD