

Project Potholes

CS 442

University of Illinois at Chicago

Spring 2014

Anne Celestino

Walter Dworak

Jeffrey Grandt

Arnold Lee

William Montgomery

Table of Contents

[1. Introduction](#)

[1.1 Purpose](#)

[1.2 Scope](#)

[1.3 Definitions, acronyms, and abbreviations](#)

[1.4 References](#)

[1.5 Overview](#)

[2. Overall description](#)

[2.1. Product perspective](#)

[2.2 Product functions](#)

[2.3 User characteristics](#)

[2.4 Constraints](#)

[2.5 Assumptions and dependencies](#)

[3. Specific Requirements](#)

[3.1 External interface requirements](#)

[3.1.1 User interfaces](#)

[3.1.1.1 User Input](#)

[3.1.1.2 Terminal Window](#)

[3.1.1.3 Command Format](#)

[3.1.1.4 Command Consistency](#)

[3.1.1.5 Web/Graphical User Interface](#)

[3.1.2 Hardware interfaces](#)

[3.1.3 Software interfaces](#)

[3.1.4 Communications interfaces](#)

[3.2 System features](#)

[3.2.1 Search](#)

[3.2.1.1 Introduction/Purpose of feature](#)

[3.2.1.2 Stimulus/Response sequence](#)

[3.2.1.3 Associated functional requirements](#)

[3.2.2 View Movie Data](#)

[3.2.2.1 Introduction/Purpose of feature](#)

[3.2.2.2 Stimulus/Response sequence](#)

[3.2.2.3 Associated functional requirements](#)

[3.2.3 View Actor/Actress Data](#)

[3.2.3.1 Introduction/Purpose of feature](#)

[3.2.3.2 Stimulus/Response sequence](#)

[3.2.3.3 Associated functional requirements](#)

[3.2.4 View Director Data](#)

[3.2.4.1 Introduction/Purpose of feature](#)

[3.2.4.2 Stimulus/Response sequence](#)

[3.2.4.3 Associated functional requirements](#)

[3.2.5 View Writer Data](#)

- [3.2.5.1 Introduction/Purpose of feature](#)
 - [3.2.5.2 Stimulus/Response sequence](#)
 - [3.2.5.3 Associated functional requirements](#)
- [3.2.6 User Account Access](#)
 - [3.2.6.1 Introduction/Purpose of feature](#)
 - [3.2.6.2 Stimulus/Response sequence](#)
 - [3.2.6.3 Associated Functional requirements](#)
- [3.2.7 Create Payment Method](#)
 - [3.2.7.1 Introduction/Purpose of the feature.](#)
 - [3.2.7.2 Stimulus/Response Sequence](#)
 - [3.2.7.3 Associated Functional requirements](#)
- [3.2.8 Make Payment](#)
 - [3.2.8.1 Introduction/Purpose of the feature.](#)
 - [3.2.8.2 Stimulus/Response Sequence](#)
 - [3.2.8.3 Associated Functional requirements](#)
- [3.2.9 Show Orders](#)
 - [3.2.9.1 Introduction/Purpose of the feature](#)
 - [3.2.9.2 Stimulus/Response Sequence](#)
 - [3.2.9.3 Associated Functional requirements](#)
- [3.2.10 Purchase Movie](#)
 - [3.2.10.1 Introduction/Purpose of the feature](#)
 - [3.2.10.2 Stimulus/Response Sequence](#)
 - [3.2.10.3 Associated Functional requirements](#)
- [3.2.11 Rent Movie](#)
 - [3.2.11.1 Introduction/Purpose of the feature](#)
 - [3.2.11.2 Stimulus/Response Sequence](#)
 - [3.2.11.3 Associated Functional requirements](#)
- [3.2.13 Favorites](#)
 - [3.2.13.1 Introduction/Purpose of the feature](#)
 - [3.2.13.2 Stimulus/Response Sequence](#)
 - [3.2.13.3 Associated Functional requirements](#)
- [3.2.14 Currency Types](#)
 - [3.2.13.1 Introduction/Purpose of the feature](#)
 - [3.2.13.2 Stimulus/Response Sequence](#)
 - [3.2.13.3 Associated Functional requirements](#)
- [3.3 Performance requirements](#)
 - [3.3.1 Number of end-users](#)
 - [3.3.1.1 Total number of user accounts](#)
 - [3.3.1.2 Concurrent users](#)
 - [3.3.1.3 Concurrent connections](#)
 - [3.3.1.4 System responsiveness](#)
 - [3.3.2 Database requirements](#)
- [3.4 Logical database requirements](#)

- [3.4.1 Types of Information used by various functions](#)
 - [3.4.1.1 Movie Data](#)
 - [3.4.1.2 User Data](#)
 - [3.4.1.3 Purchase and Rental Information](#)
- [3.4.2 Frequency of use](#)
- [3.4.3 Accessing capabilities](#)
- [3.4.4 Integrity Constraints](#)
- [3.4.5 Data Retention](#)
- [3.5 Design constraints](#)
 - [3.5.1 Size of Database](#)
 - [3.5.2 Memory Usage](#)
- [3.6 Software system attributes](#)
 - [3.6.1 Reliability](#)
 - [3.6.2 Security](#)
 - [3.6.3 Portability](#)
 - [3.6.4 Maintainability](#)
- [3.7 Other requirements](#)
 - [3.7.1 Privacy](#)

1. Introduction

This section provides a description and overview of everything included in this Software Requirements Specification document. This Software Requirement Specifications is written in accordance to the IEEE Std 830-1998.

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the “Potholes” software. It will explain the purpose for the development of the system. It will clarify the system constraints, interface and interactions with external stimuli. This document is primarily aimed towards the customer and a reference for developing the alpha version of the system for the software development team.

1.2 Scope

The system that is being developed is an online movie system. The system allows users to search for movies, actors, directors, or writers. The users are able to register themselves into the database and have the ability to rent or purchase a movie. Registered users can customize their experience by selecting a movie as a favorite. The user also has the ability to change currencies. The project will be referred to internally as “Potholes” to mitigate potential damaging information leaks. The name of the finished product will be decided at a later date.

1.3 Definitions, acronyms, and abbreviations

Word	Definition
Actor	A male performer who acts in the movie.
Actress	A female performer who acts in the movie.
Cast	All actors and actresses in the movie.
Director	A person who directs the movie.
Entity	Distinct objects
Favorite	A movie that has been marked as a favorite by the user.
Stub	A method that mimics the behavior of another component.
User	Someone who interacts with the application.
Writer	A person who written the movie.
Apache Derby	Relational Database Management System

GUI	Graphic User Interface
JavaDoc	Documentation generator for Java
JDBC	Java Database Connector
JVM	Java Virtual Machine
JUnit	Unit testing framework for Java
SQL	Structured Query Language

1.4 References

Software Engineering Standards Committee of the IEEE Computer Society. *IEEE Recommended Practice for Software Requirements Specifications*: Approved 25 June 1998. New York, NY: IEEE, 1998. Print.

1.5 Overview

The Software Requirements Specification document is organized into five chapters: Introduction, Overall Description, Specific Requirements, Appendices, and Index. The second chapter provides an overview of the system's functionality and also mentions the system constraints and assumptions that the application might have. The third chapter offers a look into the requirements in a thorough manner. The fourth chapter is the appendix that will provide supporting information in regards to the application. The final chapter is the index of the Software Requirements Specifications document.

2. Overall description

This chapter will offer an overview of the whole system. The system will be explained in context to how the system interacts with external stimuli and the basic functionality of the system. Finally, the constraints, assumptions, and dependencies for the system will be also provided.

2.1. Product perspective

The system will consist of two parts: a command-line application and a small web GUI application. The command-line interface will be used to find movies, view information about movies and actors, and the ability to rent or purchase a movie. The web application will have a richer user interface that extends upon the command-line interface.

The command-line application and the web GUI application will need to communicate with an Apache Derby database. The Apache Derby database will store information about movies, actors, directors, and writers. It will also store any information in regards to registered users such as their payment methods, their orders, and customized user experience. The functionality provided by the SQL database will be rooted into the application for the user to fully use all available functions of the application.

The system is heavily reliant on data. The Apache Derby database will be used for that to store and retrieve the data. The command-line application and the web GUI application will communicate with the database in the same manner.

2.2 Product functions

With the command-line application and web GUI application, the users will be able to search for movies, actors, directors, or writers. The result that will be shown will be based on the criteria the user inputs. There are several criterion to search for in the system.

The result of the search will be viewed in a list-view. The list view will list all relevant information about the search criteria. In this view, the users will be able to select a specific movie, actor, director or producer to view more information.

If a specific movie was chosen, the users can choose to rent or purchase the movie upon registration. If the user is registered, they can also leave a review or favorite the movie. If they are a returning registered user, they can check their user account to see their order history, make a payment, change the currency of their payment.

2.3 User characteristics

There are two types of users that will interact with the system: users of the command-line application and the web GUI application. The command line users will have full access to the systems capabilities and the web GUI application users will have limited features.

The command-line application and web GUI application users can use the application to find movies, actors, directors, or writers. This means that the users will be able to search for movies, actors, directors, or writers and choose a movie, actor, director or writer from that search to find more information about the chosen result. In order for the user to get a more relevant search result, the user can specify more in their search criteria. In order for users to rent or purchase a movie or to customize their experience, they will need to register within the application.

2.4 Constraints

The command-line application and web GUI application will be constrained by the throughput of the database. Since the database is shared between both the command-line application and the web GUI application, there might be a forced queue for incoming requests that might increase the time it takes to fetch the data and present that data back to the user.

2.5 Assumptions and dependencies

One assumption about the product is that it will either be used on a command-line application or a web GUI application. If the computer does not have enough hardware resources available to run the application, there may be scenarios in which the application does not work as intended or at all. The user is assumed to have at least 4GB of RAM and a dual-core processor to run this application. This application will run on any operating system that runs the Java Virtual Machine.

3. Specific Requirements

3.1 External interface requirements

3.1.1 User interfaces

3.1.1.1 User Input

The user will be required to use a keyboard, and optionally a mouse, to interact with the program via a command line interface using an ASCII character set.

3.1.1.2 Terminal Window

The dimensions of the terminal used is up to the software user, as the only effect on software functionality is the human readability of the software output.

3.1.1.3 Command Format

The command format of the software application will conform to a standard set of command functions, sub functions, and so forth that corresponds to major software features and their subsets while taking user input as required by the invoked function.

3.1.1.4 Command Consistency

The format of these commands, input, and output will conform to a set standard and the decided format will be consistent throughout the program. Changes to the command, input, and out structure will be adjusted as per user feedback regarding the most natural average user experience. Input and output content shall remain static between formats, and will only be adjusted in the event the input and output content is insufficient for the accomplishment of the desired function.

3.1.1.5 Web/Graphical User Interface

The user will also have the option of a basic web GUI that mimics the overall behavior of the command line interface, provides appropriately equal functionality command line interface, and should simply be presenting the user input and output in an alternative format.

3.1.2 Hardware interfaces

The software will make use of either a virtual or physical terminal of the host system as well as any hardware based JVM. The JVM itself will be running the software's Java bytecode.

3.1.3 Software interfaces

Apache Derby version 10.10.1.1, <http://db.apache.org/derby/>, will be used as the applications data storage backend. Database will be stored, added, and edited via SQL query statements sent to the software through a socket interface

3.1.4 Communications interfaces

The software makes use of no means of communication other than standard text input via the standard input pipe.

3.2 System features

3.2.1 Search

3.2.1.1 Introduction/Purpose of feature

The search feature allows a user to interactively search for movies that fit an entered criteria.

3.2.1.2 Stimulus/Response sequence

3.2.1.2.1 The user enters the search criteria into the user interface.

3.2.1.2.2 The system accepts the input and provides a mechanism for selecting a category for submitting a query.

3.2.1.2.3 Optionally, the user selects what category the search should cover (i.e. movies, actors and actresses, directors, etc.). If the user does not make this choice the default is all of the categories.

3.2.1.2.3 The user submits the query.

3.2.1.2.4 The system responds with a list of entities matching the user selected category.

3.2.1.3 Associated functional requirements

3.2.1.3.1 The system shall provide the ability to search for movies.

3.2.1.3.2 The system shall provide the ability to search for actors/actresses.

3.2.1.3.3 The system shall provide the ability to search for directors.

3.2.1.3.4 The system shall provide the ability to search for writers.

3.2.1.3.5 The system shall provide the ability to search using the all category (searches for both movies, actors, directors, and writers).

3.2.1.3.6 The system shall provide a mechanism to search from any point in the user interface.

3.2.1.3.7 The system shall be able to perform a search on a database for matching movies.

3.2.1.3.8 The system shall be able to perform a search on a database for matching actors and actresses.

3.2.1.3.9 The system shall be able to perform a search on a database for matching directors.

3.2.1.3.10 The system shall be able to perform a search on a database for matching writers.

3.2.1.3.11 The system shall be able to perform a search on a database for matching movies, actors or actresses, directors, and writers.

3.2.1.3.12 The system shall be able to display a list of movies to the end user.

3.2.1.3.13 The system shall be able to display a list of actors and actresses to the end user.

3.2.1.3.14 The system shall be able to display a list of directors to the end user.

3.2.1.3.15 The system shall be able to display a list of writers to the end user.

3.2.1.3.15 The system shall be able to display a list of combined movies, actors and/or actresses, directors, and writers to the end user.

3.2.2 View Movie Data

3.2.2.1 Introduction/Purpose of feature

When movies are displayed in a list, the user should be able to select and view information related to each movie.

3.2.2.2 Stimulus/Response sequence

3.2.2.2.1 Precondition: The system has performed a search and displayed the results. The results contain at least one match that is a movie entity.

3.2.2.2.2 The user selects a movie to display.

3.2.2.2.3 The system displays more detailed information about the movie.

3.2.2.3 Associated functional requirements

3.2.2.3.1 The system shall be able to retrieve the year a movie was released.

3.2.2.3.2 The system shall be able to retrieve the cast of the movie.

3.2.2.3.3 The system shall be able to retrieve the average user rating of a movie.

3.2.2.3.4 The system shall be able to retrieve a brief description of a movie.

3.2.2.3.5 The system shall be able to retrieve user reviews of a movie.

3.2.3 View Actor/Actress Data

3.2.3.1 Introduction/Purpose of feature

When actors and actresses are displayed in a list, the user should be able to select and view information related to each actor/actress.

3.2.3.2 Stimulus/Response sequence

3.2.3.2.1 Precondition: The system has performed a search and displayed the results. The results contain at least one match that is an actor or actress entity.

3.2.3.2.2 The user selects an actor or actress to display.

3.2.3.2.3 The system displays more detailed information about the actor or actress.

3.2.3.3 Associated functional requirements

3.2.3.3.1 The system shall be able to retrieve a short biography for an actor or actress from a database.

3.2.3.3.2 The system shall be able to retrieve the year of birth for an actor or actress from a database.

3.2.3.3.3 The system shall be able to retrieve the year of death for an actor or actress from a database. If the actor or actress is still alive, the system shall indicate so.

3.2.3.3.4 The system shall be able to retrieve a list of movies that an actor or actress has appeared in.

3.2.3.3.5 The system shall be able to aggregate biography, year of birth and death, and list of movies of an actor or actress and display the result to the end user.

3.2.4 View Director Data

3.2.4.1 Introduction/Purpose of feature

When directors are displayed in a list, the user should be able to select and view information related to each director.

3.2.4.2 Stimulus/Response sequence

3.2.4.2.1 Precondition: The system has performed a search and displayed the results. The results contain at least one match that is a director entity.

3.2.4.2.2 The user selects a director to display.

3.2.4.2.3 The system displays more detailed information about the director.

3.2.4.3 Associated functional requirements

3.2.4.3.1 The system shall be able to retrieve a short biography for a director from a database.

3.2.4.3.2 The system shall be able to retrieve the year of birth for a director from a database.

3.2.4.3.3 The system shall be able to retrieve the year of death for a director from a database. If the director is still alive, the system shall indicate so.

3.2.4.3.4 The system shall be able to retrieve a list of movies that a director has directed.

3.2.4.3.5 The system shall be able to aggregate biography, year of birth and death, and list of movies of a director and display the result to the end user.

3.2.5 View Writer Data

3.2.5.1 Introduction/Purpose of feature

When writers are displayed in a list, the user should be able to select and view information related to each writer.

3.2.5.2 Stimulus/Response sequence

3.2.5.2.1 Precondition: The system has performed a search and displayed the results. The results contain at least one match that is a writer entity.

3.2.5.2.2 The user selects a writer to display.

3.2.5.2.3 The system displays more detailed information about the writer.

3.2.5.3 Associated functional requirements

3.2.5.3.1 The system shall be able to retrieve a short biography for a writer from a database.

3.2.5.3.2 The system shall be able to retrieve the year of birth for a writer from a database.

3.2.5.3.3 The system shall be able to retrieve the year of death for a writer from a database. If the writer is still alive, the system shall indicate so.

3.2.5.3.4 The system shall be able to retrieve a list of movies that a writer has written.

3.2.5.3.5 The system shall be able to aggregate biography, year of birth and death, and list of movies of a writer and display the result to the end user.

3.2.6 User Account Access

3.2.6.1 Introduction/Purpose of feature

In order to provide access to a user account and the related data in the database, a mechanism must provide a mechanism for logging in and logging out.

3.2.6.2 Stimulus/Response sequence

3.2.6.3.1 The user opens up the user interface.

3.2.6.3.2 The system displays a prompt to login with username and password.

3.2.6.3.3 The user enters the username and password, and submits the request to login.

3.2.6.3.4 The system processes the login request. If the login is not accepted, the system starts over from 3.2.6.3.2. If the login is accepted, the system displays a starting view.

3.2.6.3.6 The user selects logout from a menu.

3.2.6.3.7 The system logs the user out and displays the login screen.

3.2.6.3 Associated Functional requirements

3.2.6.3.1 The system shall limit access to any action needing an account. This includes making a payment, adding a payment method, saving a favorite, or reviewing a movie.

3.2.6.3.2 The system shall provide a mechanism to enter a username and password.

3.2.6.3.3 The system shall provide the ability to authenticate a username and password against a database.

3.2.6.3.4 The system shall grant access to a user's account once that user is authenticated.

3.2.6.3.5 The system shall provide a mechanism that allows the user to logoff.

3.2.7 Create Payment Method

3.2.7.1 Introduction/Purpose of the feature.

To rent or buy a movie, a user must be able to provide a means of payment.

3.2.7.2 Stimulus/Response Sequence

3.2.7.2.1 Precondition: The user must be logged into the system.

3.2.7.2.2 The user selects an option to view his or her payment methods.

3.2.7.2.3 The system responds with a list of saved payment methods and an option to add another payment method.

3.2.7.2.4 The user selects the option to add a new payment method.

3.2.7.2.5 The system displays an interactive form to gather the requisite data.

3.2.7.2.6 The user enters the data.

3.2.7.2.7 The system verifies the data. If the data fails the verification, the system responds with the interactive form from 3.2.7.2.5. Otherwise, the system adds the data to the database and displays an updated list of payment methods.

3.2.7.3 Associated Functional requirements

3.2.7.3.1 The system should provide a mechanism to add payment methods.

3.2.7.3.2 The system should provide a mechanism to verify payment methods. Verification should include credit card number and type verification.

3.2.7.3.3 The system should be able to add a verified card to the database.

3.2.7.3.4 The system should be able to create a list of stored payment methods.

3.2.8 Make Payment

3.2.8.1 Introduction/Purpose of the feature.

When a user wishes to purchase or rent a movie, the system must be able to accept and verify payment.

3.2.8.2 Stimulus/Response Sequence

3.2.8.2.1 Precondition: The user must be logged into the system.

3.2.8.2.2 The end user selects a movie to rent or buy.

3.2.8.2.3 The system displays the user's payment methods. If the user has no payment methods, the system displays a message to add a payment method.

3.2.8.2.4 The user selects a payment method.

3.2.8.2.5 The system processes the payment and adds a record to the database. The system responds with a link to download the movie. If the payment cannot process, the user's payment methods are displayed again as in 3.2.8.2.3.

3.2.8.3 Associated Functional requirements

3.2.8.3.1 The system should provide a mechanism for a user to select to rent a movie.

3.2.8.3.2 The system should provide a mechanism for a user to select to buy a movie.

3.2.8.3.3 The system should be able to process a payment with the stub of a payment processor. Actual processing is not required.

3.2.8.3.4 The system should be able to save a payment in the database.

3.2.8.3.5 The system should be able to display a link to download a movie or provide some way for the user to receive the digital content. The link can be random in nature.

3.2.9 Show Orders

3.2.9.1 Introduction/Purpose of the feature

After a user makes a purchase, the user may want to view his or her payment history.

3.2.9.2 Stimulus/Response Sequence

3.2.9.2.1 Precondition: The user must be logged into the system.

3.2.9.2.2 The user selects the option to show his or her orders.

3.2.9.2.3 The system responds with a list of past payments.

3.2.9.3 Associated Functional requirements

3.2.9.3.1 The system must provide a mechanism for logged in users to select and view the orders.

3.2.9.3.2 The system must be able to retrieve a user's orders from the database.

3.2.9.3.3 The system must be able to display results to the user.

3.2.10 Purchase Movie

3.2.10.1 Introduction/Purpose of the feature

Movies in the systems database will be available for purchase.

3.2.10.2 Stimulus/Response Sequence

3.2.10.2.1 Precondition: The user must be logged into the system.

3.2.10.2.2 The user selects to purchase a particular movie from the movie entity screen.

3.2.10.2.3 The system prompts the user to make a payment and uses functionality described in functional requirement 3.2.8.

3.2.10.2.4 The system responds by delivering the movie content to the user.

3.2.10.2.5 The user is then able to view the movie content.

3.2.10.3 Associated Functional requirements

3.2.10.3.1 The system should provide a mechanism for a user to select to buy a movie.

3.2.10.3.2 The system should be able to display a link to download a movie or provide some way for the user to receive the digital content. The link can be random in nature.

3.2.10.3.3 The content of the movie will be stubbed out as a string that tells the user which movie is being viewed.

3.2.10.3.4 The system will have a medium by which to store movie content and deliver this content to the user.

3.2.10.3.5 The system must be able to accept and verify payment, as described in 3.2.8.

3.2.10.3.6 The user shall have access to view all movie content they have previously purchased without having to repurchase the film.

3.2.11 Rent Movie

3.2.11.1 Introduction/Purpose of the feature

Movies in the systems database will be available for rental.

3.2.11.2 Stimulus/Response Sequence

3.2.11.2.1 Precondition: The user must be logged into the system.

3.2.11.2.2 The user selects to rent a particular movie from the movie entity screen.

3.2.11.2.3 The system prompts the user to make a payment and uses functionality described in functional requirement 3.2.8.

3.2.11.2.4 The system responds by delivering the movie content to the user.

3.2.11.2.5 The user is then able to view the movie content for a specified period of time.

3.2.11.3 Associated Functional requirements

3.2.11.3.1 The system should provide a mechanism for a user to select to rent a movie.

3.2.11.3.2 The content of the movie will be stubbed out as a string that tells the user which movie is being viewed.

3.2.11.3.3 The system will have a medium by which to store movie content and deliver this content to the user.

3.2.11.3.4 The system must be able to accept and verify payment, as described in 3.2.8.

3.2.11.3.5 The user shall have access to view all movie content they have previously rented within the time constraint of the rental period without having to rent the film again.

3.2.11.3.6 The system must maintain the time at which a movie is rented.

3.2.11.3.7 The system must store rental period that will persist throughout the system. The rental period will be 3 days.

3.2.11.3.8 The system must determine which users have access to what content at a given time.

3.2.11.3.9 The user must not be able to view movie content outside of the rental period without renting the title again or purchasing it.

3.2.13 Favorites

3.2.13.1 Introduction/Purpose of the feature

The user will be able to select certain movies as their “favorites”. The user will then be able to view this list at a later time.

3.2.13.2 Stimulus/Response Sequence

3.2.13.2.1 Precondition: The user must be logged into the system.

- 3.2.13.2.2 The user selects the option to favorite a movie from the movie entity screen.
- 3.2.13.2.3 The system responds by adding the movie to the users “favorites” list and displays this updated list to the user.

3.2.13.3 Associated Functional requirements

- 3.2.13.3.1 The system must provide a mechanism for logged in users to view their favorited movies.
- 3.2.13.3.2 The system must store and update a list of favorite movies associated with each user.
- 3.2.13.3.3 The system must provide a medium by which users can add a movie to their favorites list.

3.2.14 Currency Types

3.2.13.1 Introduction/Purpose of the feature

The user will be able to select their preferred currency from a small list of currency options and this currency will be used for all pricing information. The default currency is the US dollar.

3.2.13.2 Stimulus/Response Sequence

- 3.2.13.2.1 Precondition: The user must be logged into the system to change the currency or to use a currency other than the dollar. The default system wide setting will be the US dollar.
- 3.2.13.2.2 The user selects the options in user settings from the main menu to change their currency settings.
- 3.2.13.2.3 The system responds by servicing all future requests for price of an item in a way that is consistent with the users chosen currency. All prices shown to that user will be displayed in the selected currency from that point on until the user changes the currency again or logs out of the system.

3.2.13.3 Associated Functional requirements

- 3.2.13.3.1 The system must store settings for currency on a system wide level.
- 3.2.13.3.2 The system must store settings for currency on a user to user basis.
- 3.2.13.3.3 The system must provide a medium by which to display each price throughout the system in each user’s preferred currency to each respective user.
- 3.2.13.3.4 The system must be able to process payments made in different currencies than the systems default (i.e. US dollars).
- 3.2.13.3.5 The system will store the price of each item for sale in the systems default currency (i.e. US dollars).

3.3 Performance requirements

This subsection should specify both the static and the dynamic numerical requirements placed on the software or on human interaction with the software as a whole. Static numerical requirements may include the following:

3.3.1 Number of end-users

3.3.1.1 Total number of user accounts

The system must support at least a thousand users accounts.

3.3.1.2 Concurrent users

The system shall support 5 concurrent users at a minimum.

3.3.1.3 Concurrent connections

The system shall support 10 simultaneous connections at a minimum.

3.3.1.4 System responsiveness

The system shall respond to 95% of requests for data in under 3 seconds and 99.9% of requests within 10 seconds.

3.3.2 Database requirements

95% of the transactions shall be processed in less than 1 second.

3.4 Logical database requirements

This section is in no way a suggestion of design but rather a logical classification of those data entities as well as their attributes.

3.4.1 Types of Information used by various functions

3.4.1.1 Movie Data

Essential information regarding movie titles, such as name, release date, director, production company, major actor / actress etc.

3.4.1.2 User Data

User data, primarily consist of user account id, name, list of billing address and phone numbers, payment information, and purchase / rental history.

3.4.1.3 Purchase and Rental Information

Purchase and rental information for the overall system, including upcoming due dates for rental return.

3.4.2 Frequency of use

We expect the database will be accessed for every activity that were generated from the Potholes front-end.

3.4.3 Accessing capabilities

There is no requirement from accessing capabilities other than the ability for the Potholes front end to access the Apache Derby backend database.

3.4.4 Integrity Constraints

The database will have to provide data integrity to ensure validity of writable operations, namely making purchases and payment.

3.4.5 Data Retention

No data retention is necessary for the usage of Potholes or the database.

3.5 Design constraints

3.5.1 Size of Database

The total size of Database should not exceed 10 GB.

3.5.2 Memory Usage

The memory usage on the Database server should not exceed 2 GB at all times.

3.6 Software system attributes

3.6.1 Reliability

Measurement of reliability metrics includes the following:

- Test cases or samples that will cover at least 80% of the Pothole front-end source code
- Data validation failures
- Catching of exceptions and “crashes” that are not related to the database

3.6.2 Security

Users of Potholes cannot edit any information on the database other than their own account profile and payment information.

3.6.3 Portability

The system runs on the JVM and supports the environments that are supported by the JVM.

3.6.4 Maintainability

All classes and methods shall be documented with JavaDoc.

All methods shall be unit tested with JUnit.

3.7 Other requirements

3.7.1 Privacy

Users shall not be able to access other users payment information or information regarding previously ordered products.