

A Penny for Your Thoughts Ltd.
Movie Match for Android
Software Design Document

Matthew Penny

Date: 12/08/2022

TABLE OF CONTENTS

1.	INTRODUCTION	2
1.1	Purpose	2
1.2	Scope	2
1.3	Overview	2
1.4	Reference Material	2
2.	SYSTEM OVERVIEW	2
3.	SYSTEM ARCHITECTURE	2
3.1	Architectural Design	2
3.2	Decomposition Description	2
3.3	Design Rationale	3
4.	DATA DESIGN	3
4.1	Data Description	3
4.2	Data Dictionary	3
5.	COMPONENT DESIGN	3
6.	HUMAN INTERFACE DESIGN	4
6.1	Overview of User Interface	4
6.2	Screen Images	4
6.3	Screen Objects and Actions	4

1. INTRODUCTION

1.1 Purpose

This is a Software Design Document for the capstone project of Matthew Penny, which consists of an Android application that uses an API from *Movie of the Night*, a Swipe-able UI, Firebase's Real-time Database as well as it's Storage function. We will be aiming for devices running Android 7.0 and higher to be able to accommodate as many users as possible. We will use Diagrams.net as the wire frame software, as well as to create any diagrams required, including but not limited to: DFD's (Data Flow Diagrams), UML diagrams, and Use Cases. GitHub will be the VCS and there will be public access for any administrators, and instructors who require it.

1.2 Scope

As mentioned above, the application is for Android devices only and those devices must be capable of running Android version 7.0 or better. This is a requirement of the streaming services and as such, cannot be altered. The programming language is Java, even though Kotlin is the language of Android, as it is the language with which our head programmer is the most comfortable. We will use the RecyclerView as we believe that the application should be accessible to all individuals, as well as the simplest design to implement. There will be a long push menu and a navigation drawer to allow data changes and access to other activities.

With this application, our goals are multiple and, we hope, far reaching. To start, we are aiming to have a one-stop application to find and stream people's favorites across multiple services, as well (if they wish) to match with and chat with others who enjoy the same types of films and series. As our progress continues and the head programmer can continue their learning, we would like the application to "learn" from the user and eventually be able to suggest a movie or series, regardless of the service that the person is currently searching through.

1.3 Overview

We will be using this document to detail the functionalities, context and design of the capstone project. As the sole employee of the company, Matthew Penny is the head designer, programmer, UI/UX developer and will be referred to as "we" throughout the document. As the project manager, he will also be in charge of the changes and diagrams that can be found throughout.

2. SYSTEM OVERVIEW

Give a general description of the functionality, context and design of your project. Provide any background information if necessary.

Develop a modular program structure and explain the relationships between the modules to achieve the complete functionality of the system. This is a high level overview of how

responsibilities of the system were partitioned and then assigned to subsystems. Identify each high level subsystem and the roles or responsibilities assigned to it. Describe how these subsystems collaborate with each other in order to achieve the desired functionality. Don't go into too much detail about the individual subsystems. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together. Provide a diagram showing the major subsystems and data repositories and their interconnections. Describe the diagram if required.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design

3.2 Decomposition Description

Provide a decomposition of the subsystems in the architectural design. Supplement with text as needed. You may choose to give a functional description or an object-oriented description. For a functional description, put top-level data flow diagram (DFD) and structural decomposition diagrams. For an OO description, put subsystem model, object diagrams, generalization hierarchy diagram(s) (if any), aggregation hierarchy diagram(s) (if any), interface specifications, and sequence diagrams here.

3.3 Design Rationale

Discuss the rationale for selecting the architecture described in 3.1 including critical issues and trade-offs that were considered. You may discuss other architectures that were considered, provided that you explain why you didn't choose them.

4. DATA DESIGN

4.1 Data Description

As discussed in the early planning stages, the information for our users, liked movies and series

Explain how the information domain of your system is transformed into data structures. Describe how the major data or system entities are stored, processed and organized. List any databases or data storage items.

4.2 Data Dictionary

Alphabetically list the system entities or major data along with their types and descriptions. If you provided a functional description in Section 3.2, list all the functions and function parameters. If you provided an OO description, list the objects and its attributes, methods and method parameters.

5. COMPONENT DESIGN

In this section, we take a closer look at what each component does in a more systematic way. If you gave a functional description in section 3.2, provide a summary of your algorithm for each function listed in 3.2 in procedural description language (PDL) or pseudocode. If you gave an OO description, summarize each object member function for all the objects listed in 3.2 in PDL or pseudocode. Describe any local data when necessary.

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

Describe the functionality of the system from the user's perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.

6.2 Screen Images

Display screenshots showing the interface from the user's perspective. These can be hand-drawn or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)

6.3 Screen Objects and Actions

A discussion of screen objects and actions associated with those objects.

7. REQUIREMENTS MATRIX

Provide a cross-reference that traces components and data structures to the requirements in your SRS document.

Use a tabular format to show which system components satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS.