Cross-Media Recommendation Application

Project Chimera

Change History

1/15/15: Initial skeleton of the document created.

1/22/15: Title of the Development set; Project Chimera.

1/28/15: Added initial Project Scope and Project Goals; Minor format edits.

2/12/15: Functional and Non-Functional Requirements section filled in.

2/15/15: Diagrams and relevant text files entered

Table of Contents

Introduction…………………………………………………………………………………………………………4

Project Description………………………………………………………………………………………………5

Requirements………………………………………………………………………………………………………6

UML Diagrams……………………………………………………………………………………………………..7

**Introduction**

Project Motivation and Purpose

The purpose of Project Chimera is to develop a mobile application and database to allow us to dynamically recommend various forms of media entertainment. To elaborate further, the purpose is to create an intuitive and adaptive User Interface for users to access and retrieve information from a database of compiled information on various forms of entertainment media. This purpose is motivated by the desire to be able to find specific forms of entertainment that appeal to specific personalities rapidly and with little work on part of the individual.

Project Scope

The Project Chimera application this report describes is intended for the Mobile Android system. It will be a widely available application intended to be used by people of all ages for its utility. The main functional pieces of Project Chimera will be based on the ability to rapidly search a database of multiple types of entertainment media and provide suggestions for the user. Users will be able to login and have their own preferences and other pieces of information to increase the efficiency of the database search. The majority of the data will be stored on a server to be accessed when the application is running. Due to the server nature of the database the application will not work correctly when unable to establish a connection to the database.

Project Goals

The main goals of Project Chimera will be staggered with each goal building upon the prior. As such the earliest goals are the most important whereas the farthest away goals may not even be necessary for the working version of Project Chimera. The first and primary goal of Project Chimera is developing a database that contains the cross-media information necessary for any application to use it. This is followed by the creation of an effective means with which to have a regular user interact with the database in order to get recommendations. The last required goal will be the development of a simple and intuitive interface for the user and database to work with for ease of use. Beyond this point goals are all additional features and functionality for the final version. These goals involve features such as determining availability, providing a rating system, and constructing a friends system.

Important Terms

**Cross-Media** – A term used to describe an item that interacts with more than one form of media.

**Project Chimera** – The name of the project to develop and create a cross-media recommendation application.

**Chimera Score** – A three digit integer rating based on user rating, critical rating and popularity.

**Project Description**

Project Features

*Necessary Features*

* Cross Media Database
* Android Application
  + User Interface
* Networking utilization between the database and application
* Method to aggregate information in the database.
* Algorithm to calculate suggestions based on different criteria
* Rating System for titles

*Additional Features*

* Viewing History log for users
* User logic across devices
* Availability of Items suggested

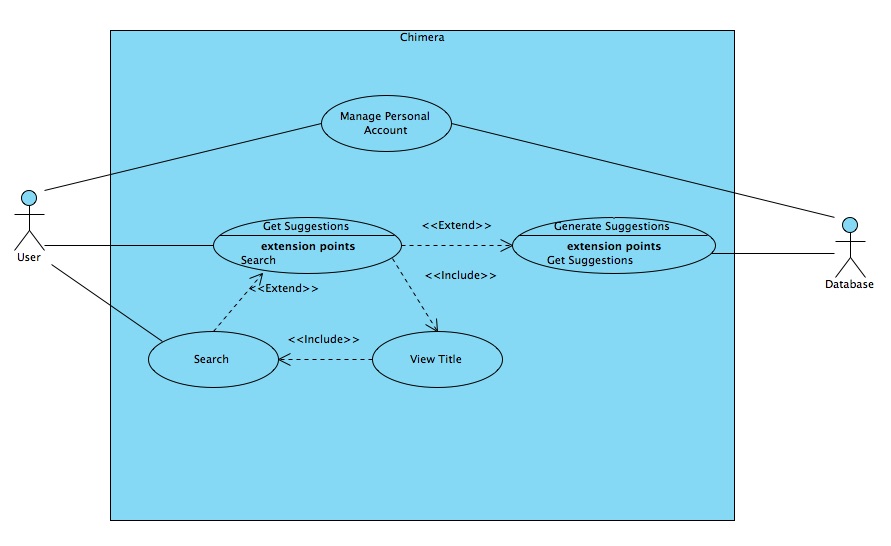
**Requirements**

|  |
| --- |
| Functional Requirements |
| Ability of the user to prompt the database for suggestions |
| Ability for the user to search the database |
| Ability for the user to manage their account |
| Ability for the user to select a title |
| Ability for the user to rate a title |
| Ability for the user to mark a title as seen and/or unseen |
|  |

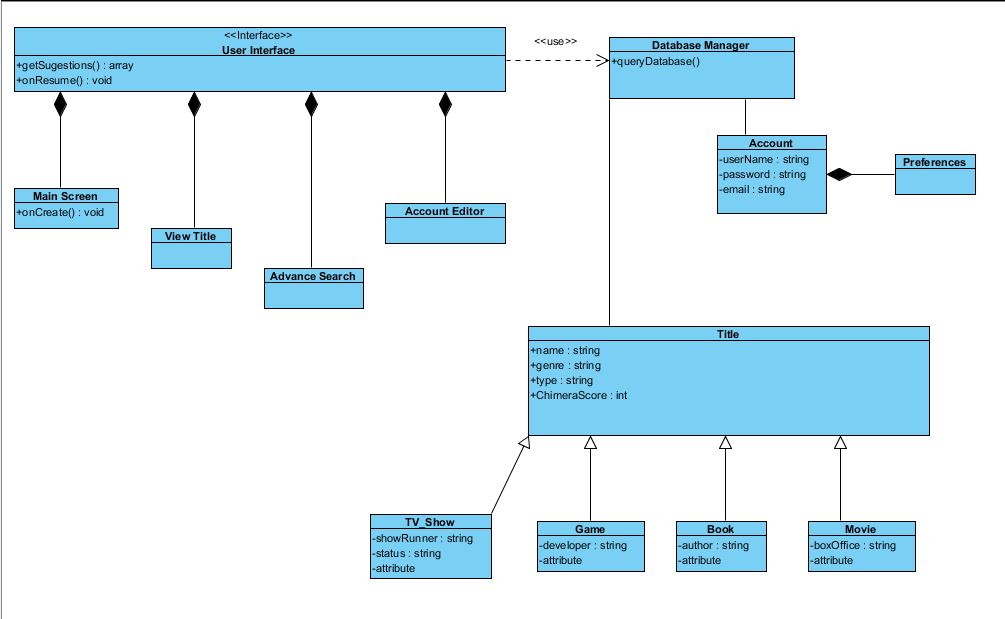
|  |
| --- |
| Non-Functional Requirements |
| Simple and Efficient User Interface |
| Rapid response times across the board |
| Database always online |
| Ability to manage a large group of users simultaneously |
|  |
|  |

UML Diagrams

Use Case Diagram

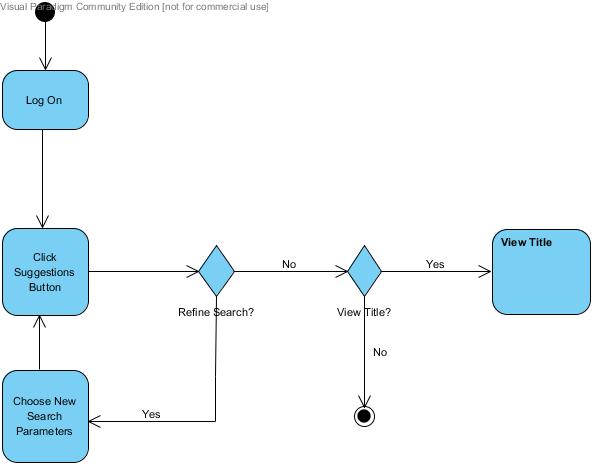


Class Diagram



Activity Diagrams

Get Suggestions



Use Case: Get Suggestions

Context: This use case allows for a user to receive suggestions for certain media from the database.

Actor(s): User, Database

Main Success Scenario:

1. The user logs on.

2. The user clicks the suggestions button.

2.1. The user can click a specific type of media to generate suggestions for.

3. The database generates suggestions of the user.

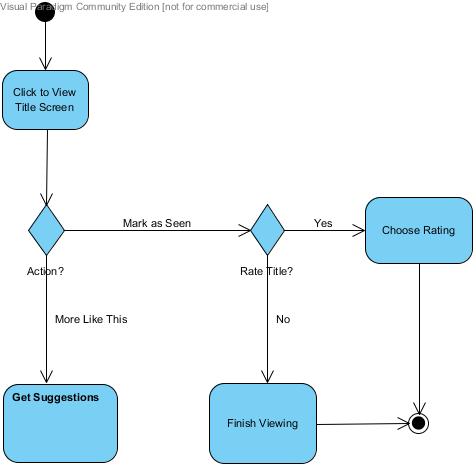
3.1. The user can view a specific title triggering the View Title use case.

3.2. The user can refine their search. (Loops back to step 2.)

3.3. The user can search for a specific title triggering that use case.

3.4. The user can rate the title.

View Titles



Use Case: View Title

Context: A title is chosen to examine bringing up corresponding information.

Actor(s): User, Database

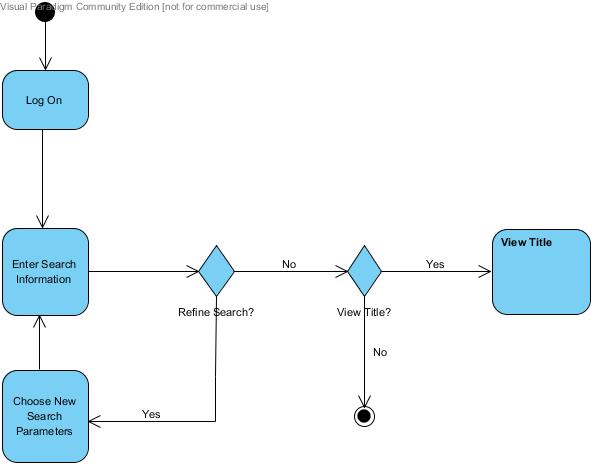
1. User clicks on link to title page.

2. A page generated for that title appears.

2.1. The user chooses to see "More Like This" triggering the Get Suggestions use case

2.2. The user chooses an aspect to search by (director, author, etc.) triggering the Search use case.

Search



Use Case: Search

Context: A user looks for a specific title in the database.

Actor(s): User, Database

1. The user logs on.

2. The user enters the search information.

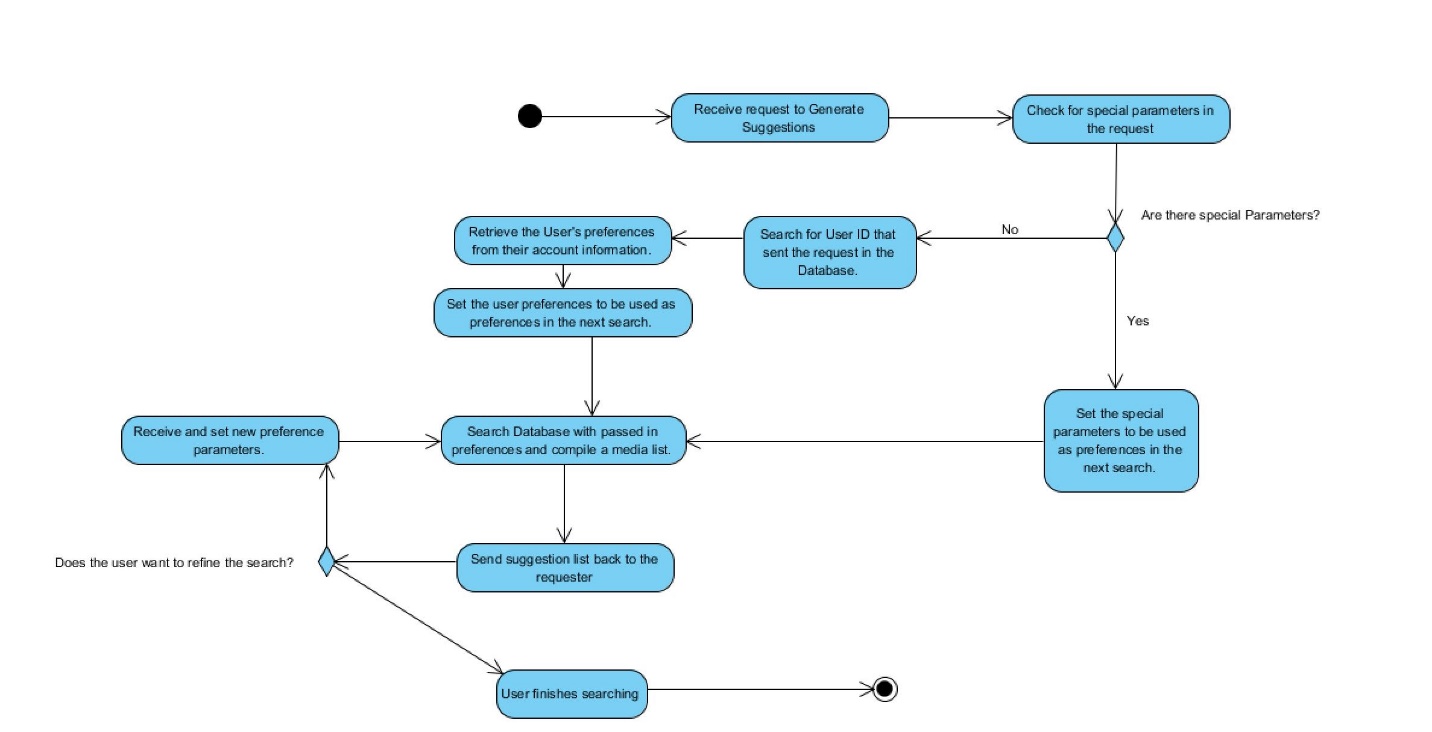
3. The database generates matching results and any close results below that.

3.1. The user can choose to view the title.

3.2. The user can choose to rate the title.

3.3. The user can ask for suggestions

Generate Suggestions



Use Case: Generate Suggestions

Context: The Database is sent a request to generate suggestions, a user ID, and possibly a list of refined search options. The Database searches and generates a list of recommendations based on the user ID preferences or the list of preferences overrides the user ID set preferences.

Actors: Database

Main Success Scenario:

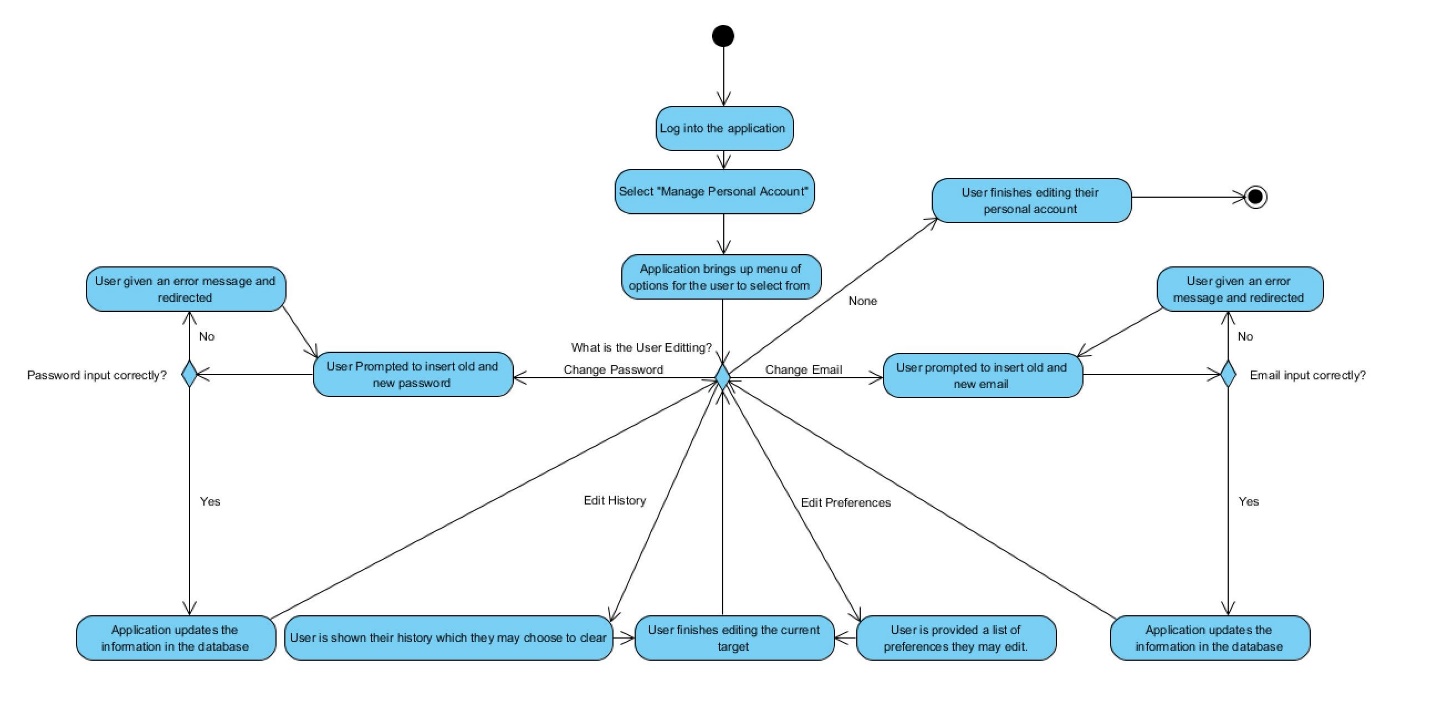
1. Database receives message from application to generate Suggestions
2. Database checks the message for a user ID and search specific preferences
   1. If no Search specific preferences are located then locate the user in the database and use the preferences stored there.
3. Using the preference list, navigate the database and select items that match the user’s preferences.
   1. If the Database finds items that are marked “viewed” in the users account it will not select them.
4. Compile the information of the selected items and send them back to the application that sent the request.

Extensions:

3a No items are selected for recommendation.

1. Send back blank list.

Manage Personal Account



Use Case: Manage Personal Accounts

Context: The User needs to access their account to change important information related directly with the account such as passwords, emails, and preferences.

Actors: User, Database

Main Success Scenario:

1. User logs into the application
2. User selects “Manage Personal Account”
   1. Database pulls up the personal account and displays the information to the user.
   2. A list of options are brought up; Change Email, Change Password, Edit Preferences, Edit History
3. User selects an option to edit
   1. If User selects Change Email
      1. The user is prompted to input their old email and a new one.
      2. User inputs required information and hits “Finish.”
      3. The Database updates the user’s email.
   2. If User selects Change Password
      1. The user is prompted to input their old password and a new one.
      2. User inputs required information and hits “Finish.”
      3. The Database updates the user’s password
   3. If User selects Edit Preferences
      1. Application brings up a list of all the preference settings.
         1. Current preferences are pulled from the database.
      2. User may tap any preference to edit it.
         1. Either tapping the item changes it directly or brings up a smaller list of options to choose from.
      3. User hits “Finish” after performing any changes they wished to do.
      4. The Database updates the user’s preferences.
   4. If user selects Edit History
      1. Application brings up a list of all the titles viewed and/or rated by the user.
         1. Information retrieved from the database.
      2. User can select any title.
         1. User can mark title as unviewed or adjust their rating.
      3. User can select “Clear History”
         1. Database marks all titles unviewed by the user.
      4. User can select “Finish” when done viewing and making edits.
         1. Database makes any edits to the User’s history.
4. User selects “Finish” when done editing their profile.
5. Application returns to the main page.