FLIXMIX

Design Document

11/2/20

Version 1

TeamBYR

Brandon Christensen, Roshani Shiwakoti, Yuta Mogi

TABLE OF CONTENTS

I. Introduction

The main purpose of this application is to connect movie lovers, who can discuss movies, share recommendations and create reviews. The user can enter movies that they have watched, into a database and tag them per genre, ratings, and other tags that they may feel is necessary. The user can create a 'watch list' for movies that they are anticipating to watch, movies they have already watched. The system will also generate movies by recommendation, based on the user's preferences. Additionally, the user is able to share movies in a discussion forum where many other users get together to discuss, review and recommend movies. The user can add movies to their 'watch list' based on another user's post. We hope to use this application to connect with avid movie watchers, critics, and general cinema enthusiasts.

This document will discuss various aspects of the software design process:

- Section II includes Architecture Design.
- Section III includes Design Details and Implementations
- Section IV includes Testing Plans
- Section V includes any references that were used for this document.

Document Revision History

Rev 1.0 2020-11-02 Initial Version (Iteration 1)

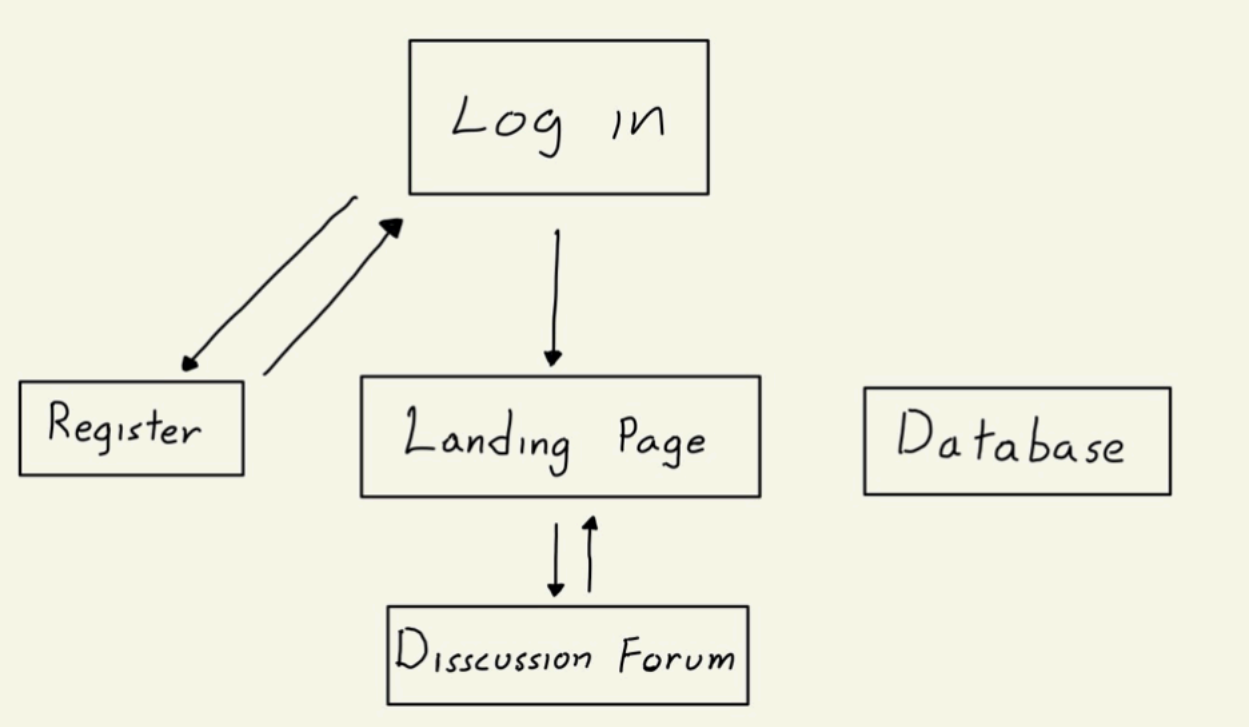
II. Architecture Design

II.1. Overview

Our program is made up of website pages and a database which contains users and their movies (movie ratings, watched movies, etc.). The structure will be very similar to the smile project that we have completed earlier this semester. We will have a login and register page, and once you login, you can look at your movies. Similar to the smile project, you can also see other people's reviews, like smiles functionality to look at other people's posts.

The rationale is the same that the smile project used. We want the program to flow in a logical manner, where you must be logged in to see content and to make posts. You must go to the page that you want (i.e. movie reviews, watch lists, etc.), and then the program will take you there.

Everything will be coupled to the users account (in the database). The users can look up how other people reviewed their movies, and then write their own. Other users can then comment on the reviews. These reviews and comments will all be tied to the posts, which will then be tied to the users account.



The subsystems are fairly self explanatory. There will be the login page, where you can login if you already are a user, or be redirected to a register page, where you can become a user. If you are already a user and you successfully login, then you are redirected to the landing page. If you successfully register as a new user, then you are redirected back to the login page. The landing page will have most of the main functionality of the program. This is where you can add movies to your watchlist, get movie recommendations, and search for new movies. There will also be a button to go to a discussion board page from here. This is where you can see other user's posts and reviews. This will all be stored in a database.

III. Design Details

III.1. Subsystem Design

Methods	URL	Description
GET, POST	/login	Login to account from this page
GET, POST	/register	Creating account for users who don't have an account yet
GET, POST	/logout	Logging out from the account
GET, POST	/landing	Where the movies will be displayed
GET, POST	/discussion	Where the discussion forum will be
POST	/addMovie	Adds movies to the database of movies
POST	/comment	Posts a comment to a discussion or movie

III.1.1. Login Page

This will be the first page that the users using this application will be greeted with. It will display the name of the app along with text boxes for users to input their username/email and password. This will

interact with the database as that will be the way to tell if the user and password matchup. There will also be a remember me checkbox, which will bypass this screen if checked next time the user comes to this app. This will also have to be remembered in the database. Finally, there will be a hyperlink for new users that don't have an account yet, and it will send you to the register page.

III.1.2. Register Page

There will be four text boxes for users to enter information into. They will be as follows: (1) username (2) email (3) password (4) re-enter password. The app will check to make sure that the username and email have not been used before by consulting the database. The app will also make sure that the password and re-entered password are matching. If both of these tests are passed, then the user is created, entered into the database, and then redirected to the login page.

III.1.3. Landing Page

This is the main section of the entire app. There will be a top navigation bar, where you can log out quickly, or search your own database for movies. There will be a button for you to enter new movies. This is something we need to consider further, but this will probably need to redirect the user to a new page that lets them enter the movie details. The movies that have been entered into the watched list will be displayed. The user will be able to scroll through the movies they have seen. There will also be a randomly generated recommendation list, set up in a similar fashion. We want all the movies that are displayed here to be clickable, and they will send you to that movies discussion forum page. There will also be a watch list that contains movies the user wants to watch next.

III.1.4. Discussion Forum Page

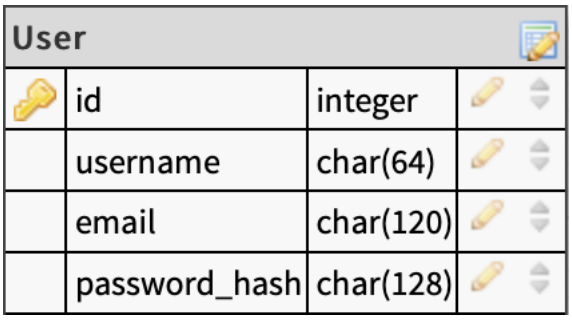
Finally, there will be the discussion forum page. This page will display a specific movie, and display that movie's details. There will be textboxes to make comments, and these comments will be displayed. There will also be buttons to add this to the users watch list. Tags, genres, etc. will be displayed here too. This will all be stored in a database.

III.2. Data design

So far, for iteration 1, we have one table in our database. To create an account and log in to the application, the database must store information about the user's credentials. This table is the User model, where the information about the user is stored. The information includes:

- Id – this field contains an auto generated id which acts as an identifier for the user
- Username – this field contains the user's username which they will use to log-in to the application
- Email – this field contains the user's email, which will be used to register an account with the application
- Password_hash – this field will contain the user's password, which will allow the user to log-in to their account, and is also required to register an account.

The database will record this information which will then be used to connect the users' credentials with later information such as posts and updates.



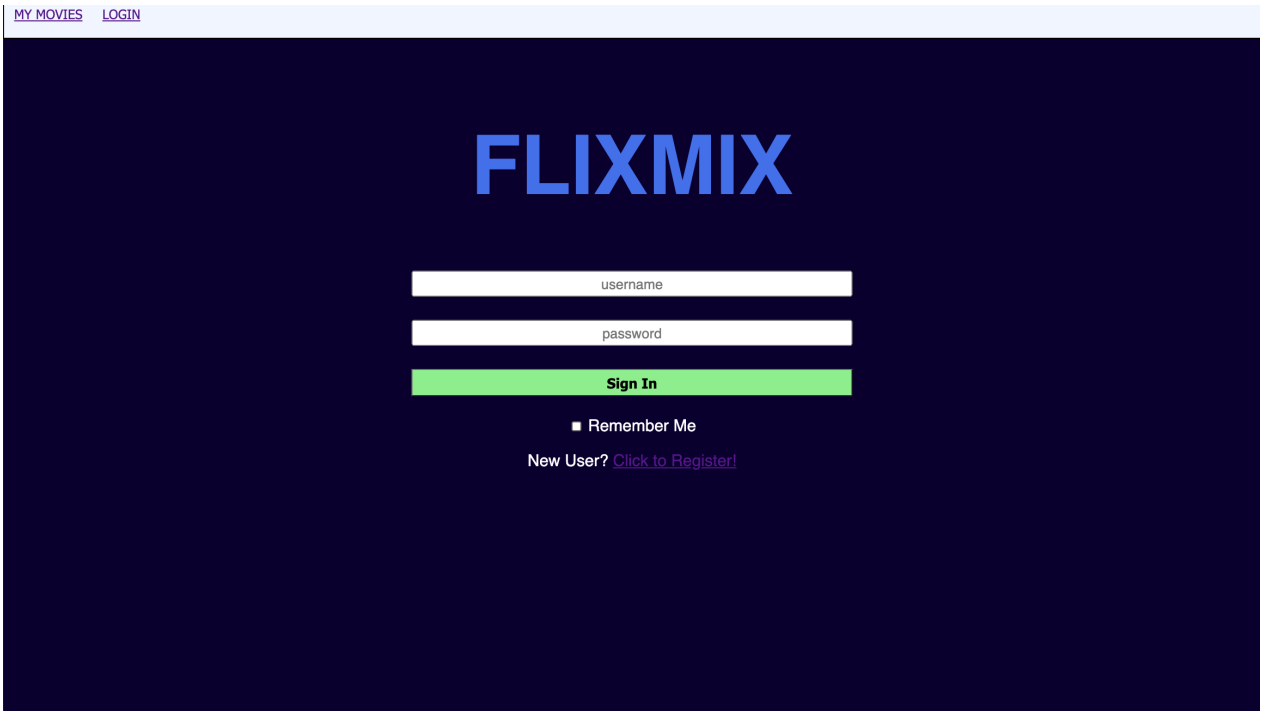
III.3. User Interface Design

For iteration 1, our team created a login, and a register (sign-up) page. This user interface implements use-case number seven from the Project Requirements Document. The user can successfully register an account with a username, email and password, which in turn gets registered into the database. After a valid account is registered into the database, the user is then able to log-in to their account and view their feed, and the forum.

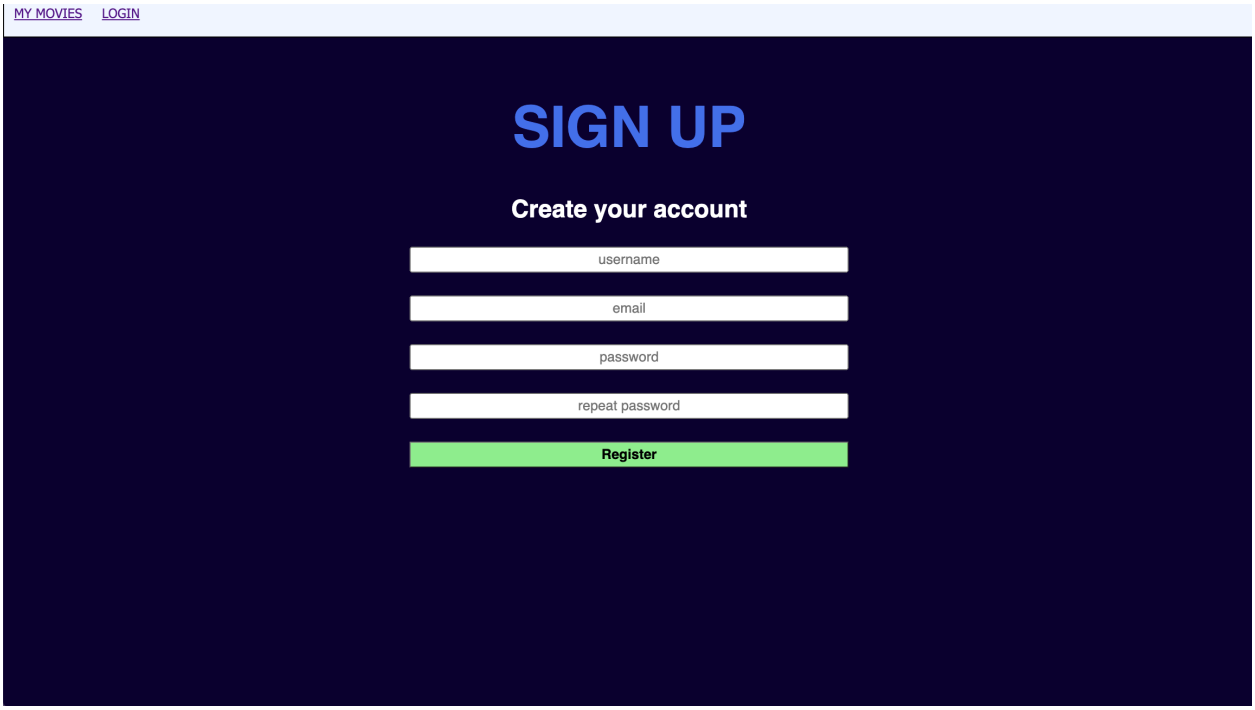
The username and password fields in the log-in page are checked for validity, and are cross checked to ensure that it exists in the database as valid log-in credentials. Similarly, the sign-up page, where the user creates an account will be checked for field validity. All input fields must be completed, and a valid email must be used. Once the user is logged in, the 'logout' button will appear in the navigation bar, where the user will be able to log out of the account.

Additionally, the 'remember me' field in the log-in page allows for the system to remember the user's username and password so that they can log-in easily the next time around.

Visually, we decided that the color scheme for this application should reflect our logo—various shades of blues. Our color scheme reflects this. Additionally, we have chosen a very simple, user friendly system for the input boxes. The minimalistic, yet concise layout should make the website less busy and overwhelming. For future iterations, we are planning on including slight changes to the visual aspect of this interface. Such as a proper background, and a more design focused navigation bar. Additionally, we are going to implement an 'about us' section which explains to the user the purpose of this application.



Log-in page



Registration page (sign-up)

IV. Progress Report

The main difficulty we faced during this iteration was the process of getting used to collaborating on a version control—gitlab. There were issues that we came across as a team, but found solutions to address these issues and eventually got through them. We did not face any considerable technical issues with code as it was straightforward. We faced a few errors that we had not encountered before, however we got through those quickly. For iteration 1, we have created a log-in and sign-up page, which allows the user to register and log-in to an account. We have discussed our plans and possible adjustments for the future which can be made to the login and registration pages. Such plans include design changes, and possible layout changes which could make the onboarding experience much easier for the users. To allow for the best user-experience, we will work to ensure that the design and information architecture is up to date with current design principles. Our goal is to create the best possible application experience for our users. We were planning on implementing the movies page for this iteration as well, however we were having a hard time deciding on how our database would store the information. We are considering the possibility of using an outside database for movie recommendations. This, we still must research and consider these possibilities, thus we were unable to include it in this iteration. To test our functionality for this iteration, we used basic UI testing, where we confirmed by checking the database whether the user information was stored correctly or not. We also did functionality tests which insured that the registration and login are working as intended.

V. Testing Plan

N/A for iteration 1

VI. References

N/A