Want a quick solution to the daunting task of picking a movie for you and your friends to watch? We have the solution for you!

Superstars

Project Portfolio

8 February, 2022

Introduction

When you and your friends want to watch a movie, the last thing you want to do is spend more time debating what to watch than actually watching something. Each friend is bound to have differing opinions; one may want to watch an action movie, while another wants to watch a horror movie. Our website will let you make a profile to store all of your watched movies and ratings, then start a group watch to combine you and your friends' favorite movies that will generate something you will all like.

The technologies we plan to use to accomplish this project will be JavaScript, GitHub, Enterprise Architect, IMDb, React, and SQL.

Core Features:

- Profile
    - User creates an account/profile.
    - User completes a "preference quiz"
        - They will rate a few movies they liked and a few movies they did not like from Top 100 movies.
    - User Database
        - Stores both negative and positive ratings of movies the user has watched.
        - If a user has seen a movie before it will not appear in recommendations.
- Movie Compromise & Social Aspect
    - "Friends List" panel feature on homepage to see other user accounts that you follow.
    - Users can invite friends from the "Friends List" panel to create a "Watch Group."
        - User who initiates the group is the leader of the group
        - Combines ratings from each user's profile to generate one movie to satisfy all users
        - Regeneration option on leader of group's profile to generate a new movie recommendation

Viable Features:

- Recommendations will be accompanied by a list of streaming services they are available on.
- Once movies are recommended there will be an option of "I've watched this."

Stretch Features:

- Import data on movie watches and reviews from Letterboxd to implement users' Letterboxd data for personal or more detailed recommendations.
- Certain social-media-like features such as friends' reviews or suggestions of movies with options to share on Twitter, iMessage, etc.
- Signing in by using profiles from other applications as a shortcut(Google, Facebook, etc).

- Every month there will be a popup of movies released in that month and the user can add any of these movies to their watch list.
- Movie Compromise & Social Aspect
    - Filter System
        - “In the Moment” Filter
            - Overrides the automatic movie generation based on user profiles and lets each member of a “Watch Group” input a movie based on what they want to watch at the moment
                - i.e. if a user’s profile is strongly weighted toward a specific genre and they want to watch something else
            - The program takes movie tags from a database such as genre, release year, and budget.
            - The program compares the user-inserted films’ tags with database tags to find a new recommendation that would best satisfy the inputs of the group.
        - Other Filters (Genre, Year, Budget, etc.)

The Superstars Team

Superstars is a team of six:

Riley Bell (Leader, Front-end)	Finn Phayer (Collaborator, Back-end)
Cameron Crochet (Collaborator, Front-end & Back-end)	Teddie Swize (Collaborator, Back-end)
Kirk McBrayer (Collaborator, Front-end & Back-end)	Emily Vu (Collaborator, Front-end)

GitHub: https://github.com/rbells/super-stars

System Requirements

Requirements

- Users can sign up on website and store their movie ratings
- Website will generate movie recommendations based on personal ratings and other users’ ratings
- Receives data from the IMDB movie database by calling its API
- Implements existing API from the web to select movies users are most likely to enjoy

Epics

Epic #1

As a person in a friend group, I want a movie to be picked for me and my friends, so we do not have to waste time debating what to watch.

User Stories

User Story #1

As a host, I want to use the movies my guests suggest to find a movie we all can enjoy to avoid arguments and enjoy our time together.

User Story #2

As a subscriber of streaming services, I want my movie suggestion to include exactly which streaming services it is available on so that it is easily accessible.

Project Management

Continuity of Operations Plan (COOP)

Communication/Coordination

- Members will fill out a weekly form detailing the days and times they are available for that week. From this form, weekly meetings will be tentatively set to maintain a level of flexibility for members' schedules.
- Members will post issues, questions, solutions, etc. in the group Discord, which will serve as the primary method of communication.

Contingency Plan

- If a member or members are not able to meet in-person, they will meet via web conference. If a member or members is not able to meet at a specific time, the leader and the team members that are working on functions related to the absent team member's assignments will decide a different time to meet. This meeting will entail an update of the project and any changes to the absent member's assignment.
- If a member suddenly becomes temporarily, indefinitely, or permanently unavailable to continue working on their respective assignment for whatever reason, members working on other assignments will divide the absent member's work amongst each other. If the absent member returns, they will be updated on the project's status with their next assignment to be determined as a group.
- All members agree to have an open discussion or impromptu meeting about workloads to ensure the work is equally distributed.

Project Plan

System Architecture Design and Development

#	Activity	Pre #	Estimated Effort	Actual Effort	Estimated Start Date	Estimated Finish Date	Actual Start Date	Actual Finish Date
1.0	System Requirements							
1.1	Epics & User stories		30%	30%	2/2	2/8	2/3	2/7
1.2	Core, Viable and Stretch features	1.1	50%	60%	2/2	2/8	2/3	2/7
1.3	Continuity of Operations Plan		[20%]	[10%]	2/2	2/8	2/3	2/6
1.3.1	Communications Plan		10%	5%	2/2	2/8	2/3	2/6
1.3.2	Contingency Plan		10%	5%	2/2	2/8	2/3	2/6
2.0	System Architecture**							
2.1	Components Diagram (Enterprise Architect)	1.0	.5	1.6	2/10	2/22	2/15	2/22
2.1.1	User I/O	2.1	.8	.6	2/10	2/22	2/15	2/17
2.1.2	External Data Sources	2.1.1	.4	.1	2/10	2/22	2/15	2/17
2.1.3	Subcomponents & connections	2.1.1-2	.6	.2	2/10	2/22	2/15	2/17
2.1.3.1	Interfaces	2.1.3	.2	.1	2/10	2/22	2/15	2/17
2.2	Data Flow Diagram (Enterprise Architect)	2.1	.5	.4	2/10	2/22	2/17	2/22
2.3	Prototype Source Code		1	1.5	2/18	2/22	2/21	2/21

System Implementation <Milestone 3: System Implementation>

Milestone 3 (System Implementation): The WBS activity chart for the milestone should be updated to include actual level of effort and start and completion dates.]

#	Activity	Pre #	Estimated Effort	Actual Effort	Estimated Start Date	Estimated Finish Date	Actual Start Date	Actual Finish Date
3.0	System Components & Patterns							
3.1	Component 1-5		2.3		2/24	3/9		
3.2	Design Patterns		1.7		2/24	3/9		

Project Postmortem <Postmortem>

Project Wins

[Provide a bulleted list of at least 3 positive aspects of the project.]

Root Cause Analysis

[Provide a bulleted list of at least 3 negative aspects of the project. For each negative, provide the answer to the three successive “Why” questions.]

Lessons Learned

[For each negative aspect identified in the Root Cause Analysis, provide a mitigation strategy (i.e., what process should be introduced) to ensure that the problem is not repeated in subsequent projects.]

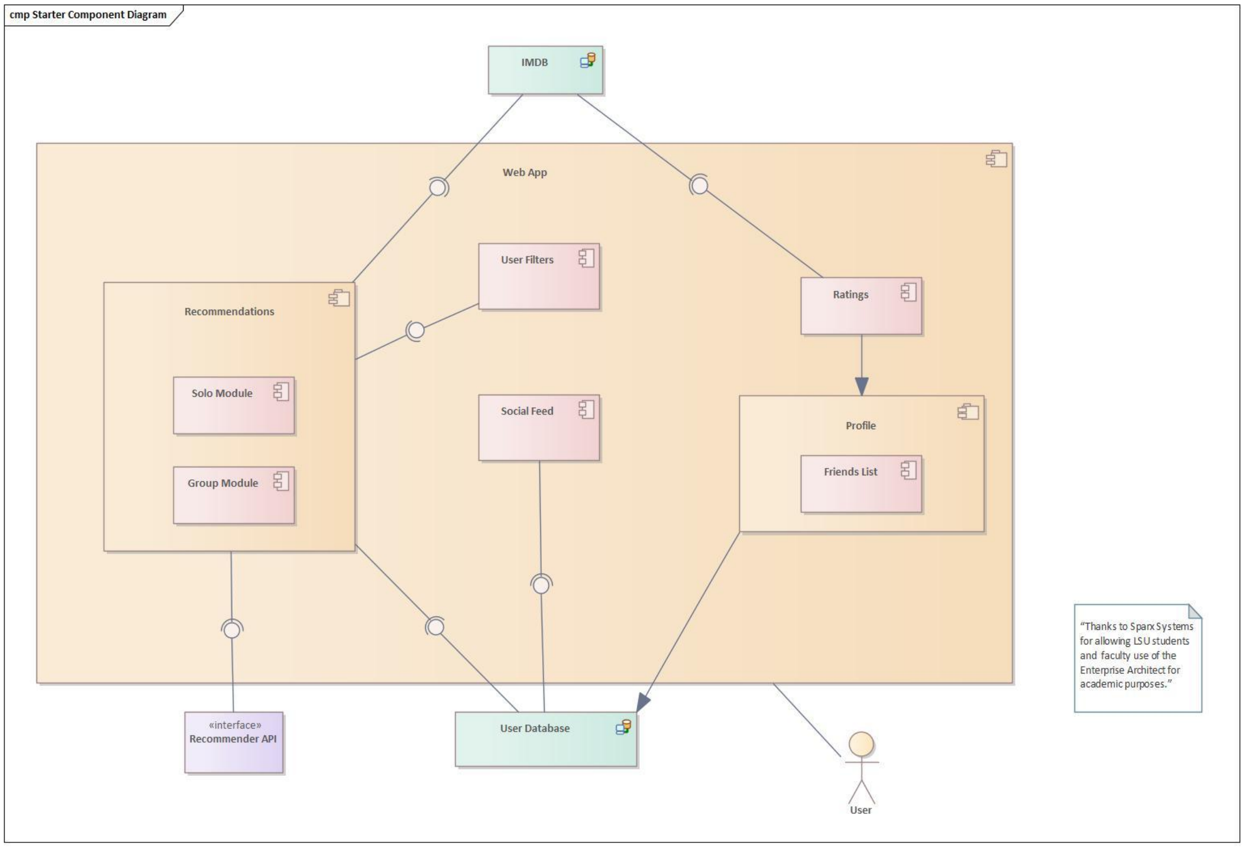
System Design

The system is primarily a data centric design since every part of the application relies on data given by users to be stored in databases. The recommender and social feed components of the system are completely dependent on the data.

System Architecture

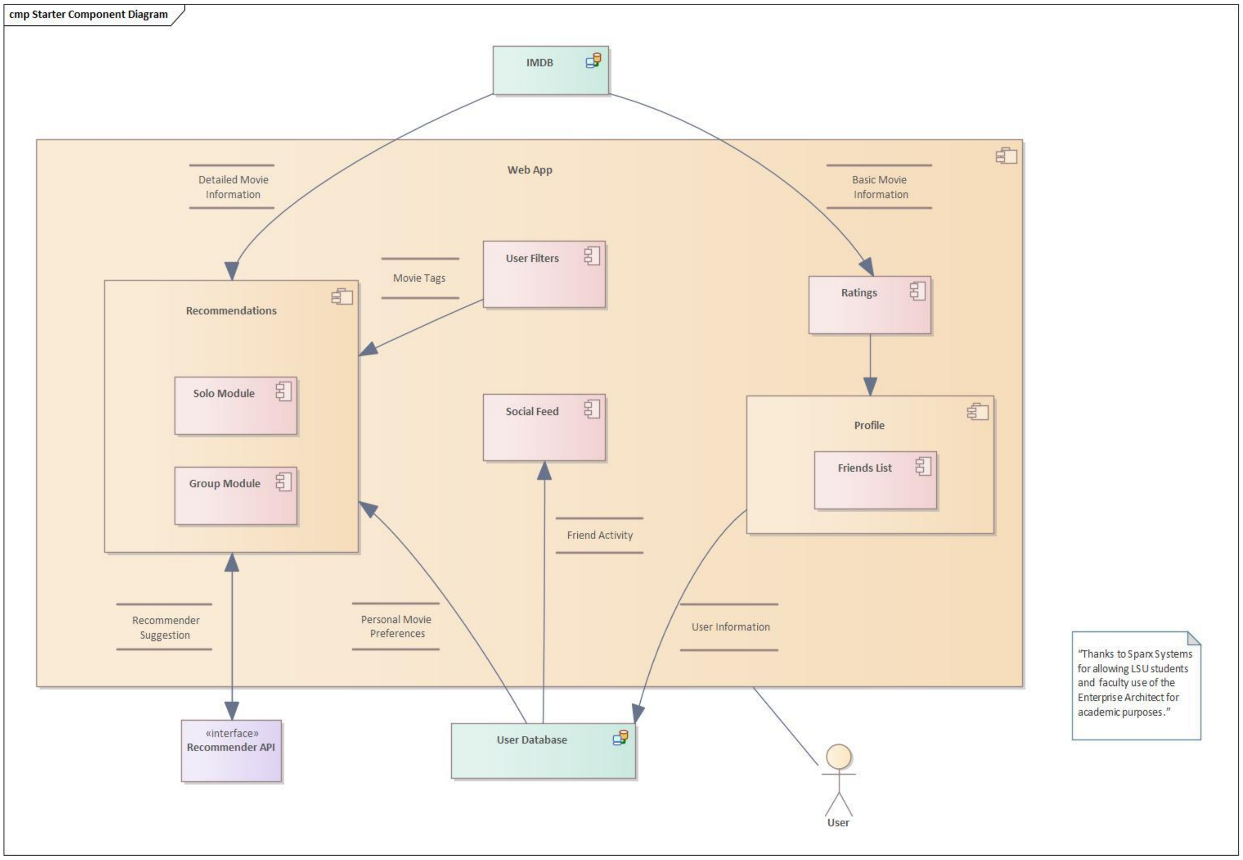
The architecture is data centric that produces movie recs for an individual and a group as well as creates a social feed of users’ friends’ movie ratings and reviews.

Component Design


cmp Starter Component Diagram
IMDB
Web App
User Filters
Recommendations
Ratings
Solo Module
Social Feed
Profile
Group Module
Friends List
«interface»
Recommender API
User Database
User
"Thanks to Sparx Systems for allowing LSU students and faculty use of the Enterprise Architect for academic purposes."

Recommendations from the Recommender API are received and then filtered in the recommendations module for an individual or group of users, after using data from the IMDB database. User ratings and preferences are stored in the User Database, which influences the Recommendations that are produced. The Social Feed displays ratings and reviews from the users and their friends from the User Database.

Data Flow


cmp Starter Component Diagram
IMDB
Web App
Detailed Movie Information
Basic Movie Information
User Filters
Movie Tags
Ratings
Recommendations
Solo Module
Group Module
Social Feed
Profile
Friends List
Friend Activity
Recommender Suggestion
Personal Movie Preferences
User Information
«interface» Recommender API
User Database
User
"Thanks to Sparx Systems for allowing LSU students and faculty use of the Enterprise Architect for academic purposes."

Movie classification data will be received from the IMDb database. Names and lists of movies will be searchable via the user's profile page. Ratings and reviews from the users made on their profile page will be stored in the user database. The user database will provide the friends list capability on users' profile pages, reviews and ratings from the users' friends to the social feed, and users' ratings to the recommender module. The recommender module will receive users' ratings and any appropriate filters the user wishes to apply to their recommendation. The recommender API will receive the users' ratings and filters and return movie recommendations.

System Components <Milestone 3: System Implementation>

[*Include a component sub-section for each component in the architecture diagram. Each component subsection will include a class diagram*]

Component [Component Name 1]

[*A short description of the component.*]

[*An EA class diagram of the component that includes method parameters.*]

Component [Component Name 2]

[*A short description of the component.*]

[*An EA class diagram of the component that includes method parameters.*]

Component [Component Name n]

[*A short description of the component*.]

[*An EA class diagram of the component that includes method parameters.*]

Design Pattern <Milestone 3: System Implementation>

[*Class diagram of design pattern incorporated into the project. Pattern must be specific to the project and not a general design pattern class diagram. The project must include at least 2 design patterns covered in class.*]

Design Pattern <Milestone 3: System Implementation>

[*Class diagram of design pattern incorporated into the project. Pattern must be specific to the project and not a general design pattern class diagram. The project must include at least 2 design patterns covered in class.*]