**G.O.A.T.**

Go On And Try: A Media Recommendation Service

**Team Members:**

* Peyton VanHook
* Peyton White
* Sean Hopkins

**Abstract:**

Goat will be a desktop-based media recommendation service that provides recommendations for new media to users based on user preferences. The service will aggregate listings of video games, movies, television programs, and books, and will distill media items’ summaries to a set of common tags. These tags will represent genre, tone, theme, content ratings, critical aggregate scores, and more. The key to Goat’s recommendation service will be our search algorithm, which, when provided with tags, will find media properties in it’s database that are most associated with the tags provided.

**Requirement list:**

***User interface***

* + Profile Login is always the start window.
  + From there the user has multiple options.
    - Enter a username and password that matches the database and be sent to Profile.
  + User profile utilizes Media Preferences, Save List, Seen It List, as well as display of profile photo, username and other optional descriptors.
    - Choose to “sign up” which sends them to Profile Sign Up.
  + Once required credentials are entered and saved, the user is prompted by the Media Preferences feature.
  + Once Media Preferences are finished the user is sent to their Profile.
* User Profile
  + Profile Login
    - The first window to open is the login window.
    - The login window prompts users for a username and password.
    - Once submitted, the username and password is sent to the server.
    - If the username is not in the user table an error is returned.
      * The user is then prompted to retry their login attempt until the submitted username is found.
    - If username is found but the password does not match the user table an error is returned.
      * The user is then prompted to retry their login attempt until the login info is correct.
    - Once successfully logged in, the user is sent to the Home screen.
    - The user can also click a link to the Profile Sign Up Screen to make a new account.
  + Profile Sign Up
    - A sign up window is opened for users who want to create a new profile.
    - The user is prompted to enter a username and password.
    - The username and password are required fields and are saved in the user table as parameters for the user login.
    - Will also ask for:
      * Birthday
      * Email
    - On submission, the registration data is sent to the server.
      * If the username is found in the user table, an error is returned.
      * User is prompted to use a different username, and make a new registration attempt.
      * If the email is found in the user table, an error is returned.
      * User is prompted to use a different email, and make a new registration attempt.
    - Once the registration attempt is successful, the user is created in the user table and the user is automatically logged in.
    - The user is then given the option to set Media Preferences which leads them to a separate window.
  + Profile Screen
    - A window displaying the users specific profile information:
      * Profile photo
      * Username
      * Email
      * Birthday
    - Media Preferences
      * Displays preferences set for each Media
      * May be edited at any time by user
        + “edit preferences” button
    - Rating List
      * Shows a list of media that the user has rated, and therefore seen.
  + Media Preferences
    - A window is opened for users to specify genre preferences.
    - This is used to narrow down specific user interests to provide more precise recommendations.
    - User is asked to select which of the three media types they’re interested in (can be multiple):
      * Video Games
      * Movies
      * Books
    - For each media type they selected, the user is presented with a list of genres appropriate to each type and is asked to select all genres they are interested in.
    - For each of the genres they select, the recommendation system will create a short list of top recommended items and return it to the user.
    - The user will be asked to rank each item, and the rankings will be saved to the user ranking list table.
  + Save Item For Later
    - A button on media pages that users may check if they are interested in a media item.
    - When checked, item information will be saved in their profiles Saved List.
    - If the item is unchecked the item will be removed from the Saved List.
  + Saved List
    - List of media that the user has saved for later.
    - List will be displayed on the home screen.
  + Rated it
    - Items the user has rated are saved to this list.
    - When an item is rated, the item is saved to the user rating table on the server.
  + Rated List
    - List of media the user has rated.
    - List will be displayed on the user profile, and the user will be given multiple options to display that list.
* Home Screen
  + Search bar
    - To find recommended items based on a title string.
  + Secondary features:
    - Saved List
    - Most Viewed Genres
      * The system will pull a list of genres based on the user’s rating table.
      * The system will use the keyword-based recommendation algorithm to populate a list of related items under that genre on the home screen.
* Media Item Screen
  + The media item screen will contain:
    - The media title
    - The genre list
    - The media description
    - A ranking dialogue for the user to submit item rankings
    - The media cover photo
    - The keyword list
    - A list of items recommended as being similar to the chosen media, using the recommendation system.

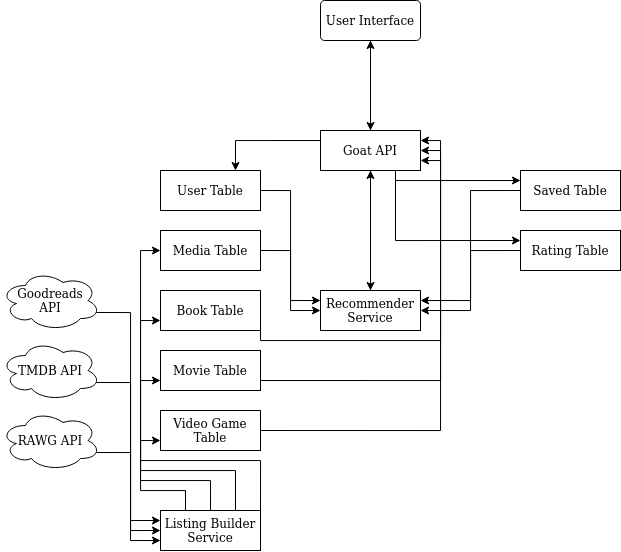
***Recommendation System***

* The recommendation system will have two input functions:
  + The ID-based recommendation system will take a list of media IDs and return a different list of media IDs with a similarity value.
  + The keyword-based recommendation system will take a list of keywords and will return a list of media IDs with a similarity value.
* Both configurations of the recommendation algorithm will also accept a limit on the number of results or a limit on the lowest similarity value returned.

***Database***

* User table
  + User\_Id(primary key)
  + Username
  + Password(Will be hashed using PHP)
  + Picture\_Path
  + Name
  + Email
  + Birthday
* Media Table
  + Id(Primary key)
  + Title\_Of\_Media
  + Date\_released
  + Media Type (Book, Movie, Video game), (1,2 or 3)
  + Picture\_Path
* Book Table
  + Id(Foreign key)
  + Author
  + Num\_Of\_Pages
* Movie Table
  + Id(Foreign key)
  + Duration
  + Producers
  + Actors
* Video Game Table
  + Id(Foreign key)
  + Console
* Genre Table
  + Id(Foreign key)
  + Genre(s)
* Saved List Table
  + Id(Foreign key)
  + User\_Id(Foreign key)
* Rating Table
  + Id(foreign key)
  + User\_Id(Foreign key)
  + Has\_Watched(0 or 1)
  + Rating
* Keyword Table
  + Id(foreign key)
  + Keywords
* Populating database
  + PHP activated when user clicks a media to be saved (saved list add)
  + PHP activated when user is logging in (checking accounts)
  + PHP activated when user is registering an account.
  + PHP activated when the user rates a media
  + PHP activated when ranking a media item or clicking on media (seen list add)
  + When application starts(login) it populated database with new information if new information has been added to the API we are pulling from
* Building the initial database
  + The system will populate the initial dataset from several downloaded datasets of media items.
  + An application will be written to unify disparate datasets into our data schema.
  + The application will write the unified data into the database.
* Updating the database
  + The system will source new releases from several online listing services, and pull data on those releases daily.
  + An intermediate function will be written for each listing service used to unify their data with our dataset.

**Block Diagram:**



**Timeline:**

|  |  |  |
| --- | --- | --- |
| Week | Team Member | Tasks |
| Sept 13-19 | Peyton V | * Create conversion programs for datasets to create initial media databases. * Decide on keywords and genres |
|  | Sean | * Start on Profile GUI   + Login   + Sign up |
|  | Peyton W | * Get webserver organized and tables created * Begin to think about keyword list and getting media * Research |
| Sept 20-26 | Peyton V | * Load converted dataset into database * Look at API calls and licensensing and see what data we can source live. |
|  | Sean | * Profile cont.   + Profile photo display   + Display of extra profile information     - e.g. birthday, email |
|  | Peyton W | * Populate database with media * Run queries to make sure getting information is efficient |
| Sept 27-Oct 3 | Peyton V | * Work with Peyton W to get timed API requests working * Research recommendation algorithm techniques and narrow down to one that will fit our parameters. * Determine where recommendation algorithm should run and how often it should run |
|  | Sean | * Start on profile lists   + Watch list   + Rating list |
|  | Peyton W | * Start working with the recommendation algorithm * Make sure database has what is needed for the RA |
| Oct 4-10 | Peyton V | * Construction recommendation algorithm interface * Implement keyword-based recommendation algorithm * Implement ID-based recommendation algorithm |
|  | Sean | * Finish User Profile   + Watch history   + Check for Profile bugs |
|  | Peyton W | * Start working with front end and database * Use PHP to get information from database for frontend |
| Oct 11-17 | Peyton V | * Construct cold-case profile for recommendation algorithm if necessary * Create API calls for UI to source media data not stored in database from internet sources. |
|  | Sean | * Start Media Item Screen   + Cover photo   + Media title |
|  | Peyton W | * Continue front end work with PHP and login efforts |
| Oct 18-24 | Peyton V | * Begin designing loose-data scrapper to source missing data on media from Wikipedia or another loose-data source. |
|  | Sean | * Media Item Screen   + Media Description   + Genre List |
|  | Peyton W | * Work with GUI to display information that we are getting from PHP |
| Oct 25-31 | Peyton V | * Begin working through loose-data with scrapper utility to create full media database. * Determine if any graphical design elements are needed for GUI presentation and, if so, begin designing those. |
|  | Sean | * Finish Media Item Screen   + Keyword List   + Recommended List |
|  | Peyton W | * Begin to code the token in PHP |
| Nov 1-7 | Peyton V | * Continue testing and work on scrapping media database. * Start implementing GUI image elements if necessary. |
|  | Sean | * Finish Media Item Screen   + Rating system   + Check for bugs * Start Home Screen GUI |
|  | Peyton W | * Have the token security done * Still work on the GUI front end |
| Nov 8-14 | Peyton V | * Continue testing and work on scrapping media database. * Continue implementing GUI image elements if necessary. |
|  | Sean | * Home Screen GUI   + Implement search bar |
|  | Peyton W | * Help in the front end of stuff needs to be finished in order to display information * Work on profile GUI |
| Nov 15-21 | Peyton V | * Aim to have full media database created and loaded into media database, replacing our initial working database. * Aim to have GUI elements finished and ready to be added into GUI by this point. |
|  | Sean | * Home Screen   + Implement saved list   + Most viewed genres |
|  | Peyton W | * Have GUI figured out with database and work to complete the full front end |
| Nov 22-28 | Peyton V | * Do final testing of recommendation algorithm and media dataset. |
|  | Sean | * Thorough recheck for bugs |
|  | Peyton W | * Finalize database efforts and GUI efforts and make sure new data is being populated into database |