

# **CSCI 4100U - Mobile Devices**

## **Final Project - The Entertainment Tracker**

Group Name:

course-project-csci-4100-final-project

git repo:

<https://github.com/CSCI4100U/course-project-csci-4100-final-project>

Group Members:

1. Alexander Giannoulis (Alex-Giannoulis)
2. Sejal Shingal (Sejal1109)
3. Bence Takacs (BenceTakacs15)
4. Ebubechukwu Okeke (ebube52)
5. David Dickson (David-A-DavidD)

## **Overview of the project**

Our application “The Entertainment Tracker” lets the user track and review different forms of media. For now, the users have the ability to track and view two types of entertainment media: Movies and Books. As a user of the application, upon registering you will be able to see Daily Trending movies and books and in depth information about each movie or book. From there you can add the movie or book to your watch or read list and also you can post a review for it if you’ve already watched/read it which is also visible to other users. To make it easier for you, you can find the nearest Cinemas and book stores based on your location to freely indulge in the entertainment world! We hope you have a fun time tracking and sharing your opinions about the latest movies and books!

## **Group members and contributions**

### **Alexander Giannoulis (100748198)**

#### **Contributions:**

- Cloud storage used in final iteration for movies
- Set up local storage and model for books
- Internationalization supporting English, French and Spanish for the entire app
- Initial implementation of snack bars
- Helped work on UI and functionality for Books
- Account page UI and Functionality
- Implemented DateRangePickers using calendars used for filtering in movie view(With Sejal)
- Implemented extra functionality to allow users to switch between dark mode and light mode
- Implemented a youtube player to show trailers for movies on the movie\_detail page but was unable to (Deprecated due to lag)

#### **Files Worked on:**

- Every viewable page for internationalization
- Account\_view.dart
- Book.dart, book\_model
- DB\_Utils.dart,
- Assets/i18n/en.json, es.json, fr.json
- themeProvider.dart, theme\_button.dart
- Movie\_model.dart, movie.dart
- Movie\_view.dart
- main.dart

### **Bence Takacs (100743604)**

#### **Contributions:**

- Initial UI layout
- Local storage for books
- Cloud storage for movies
- Error message upon failing to load data from Firestore DB
- Refactored book system and helped set up its HTTP requests (with Sejal)
- Forms to add movies and books before being deprecated
- Review system for movies including its cloud storage
- Search bar functionality and API calls for books and movies

- Caching data retrieved from web APIs
- Login and register pages and functionality
- Chart for trending movie ratings (with Sejal)
- Licenses Form

#### **Files worked on:**

- All models/\*\_model.dart files
- All components/\*\_tile.dart files
- login\_page.dart and register\_page.dart in views/auth/
- book.dart, movie.dart, trending.dart, review.dart, book\_author.dart in classes/
- movie\_search\_delegate.dart and book\_search\_delegate.dart in models/
- models/fetch\_data.dart and models/db\_utils.dart
- movie\_view.dart, movie\_details.dart, books\_view.dart, book\_details.dart, review\_list.dart, home\_page.dart, chart\_page.dart in views/

#### **Sejal Shingal (100738147)**

##### **Contributions:**

- Initial Setup of Cloud Storage and Movie Collection using Firebase
- Research and finding the APIs used: The Movie DB (<https://www.themoviedb.org/documentation/api>), OpenLibrary (<https://openlibrary.org/developers/api>), TomTom Api for Geocoding (<https://developer.tomtom.com/products/places-api>).
- Implemented HTTPS Requests for Movies and Trending Movies and the UI
- HTTPS Requests for GeoCoding And the UI using PageBuilder
- Navigation Drawer to make navigation to all pages cleaner
- Maps and Geolocation
- Datepickers and sort the My Movie List based of the selected date range
- Added Custom Font for the application
- Helped on Book API, local storage and UI (Along with Bence)
- Overall application UI and UI design for basically all pages
- Edit Account page which uses Firebase Authentication data and update it
- Snack Bars for trending pages and list pages that show up and use delay before navigation to detail screens
- Attempted to implement a youtube player to show trailers for movies on the movie\_detail page but was unable to (deprecated)

##### **Files worked on:**

- All view pages for UI design
- Classes: book.dart, movie.dart, trending.dart and trending\_book.dart with Bence
- Classes: mapMarker.dart, movie\_cast.dart

- Components: drawer.dart
- Components: book\_tile.dart and movie\_tile.dart with Bence
- Models: db\_utils and movie\_model with Alex
- Views: books\_view.dart, books\_details.dart, chart\_page.dart, movie\_details, movie\_view, trending\_books\_view with Bence
- Views: edit\_profile, map\_bookstore, map\_view,, trending\_movie\_view

### **David Dickson (100784459)**

#### **Contributions:**

- Initial TextFormFields for books and movies while laying out app foundation
- Initial notification button before project check in
- Removed class fetch fetch\_data\_books (Did not run properly)
- Attempted to set up and show book API for trending on phone screen (Unable to Implement)
- Attempt to find working book API(Unable to Implement)
- Attempt to have a time notification(Unable to Implement)
- Notifications setup
- Notifications when the movie or book is added to a list

#### **Files worked on:**

- Drawer.dart
- fetch\_data\_books.dart(deleted)
- Book.dart
- Book\_view.dart
- Notifications\_manager.dart
- Book\_details.dart
- Movie\_details.dart
- Movie\_view.dart(Precheckin)
- Book\_view.dart(Precheckin)

### **Ebubechukwu Okeke (100784002)**

#### **Contributions:**

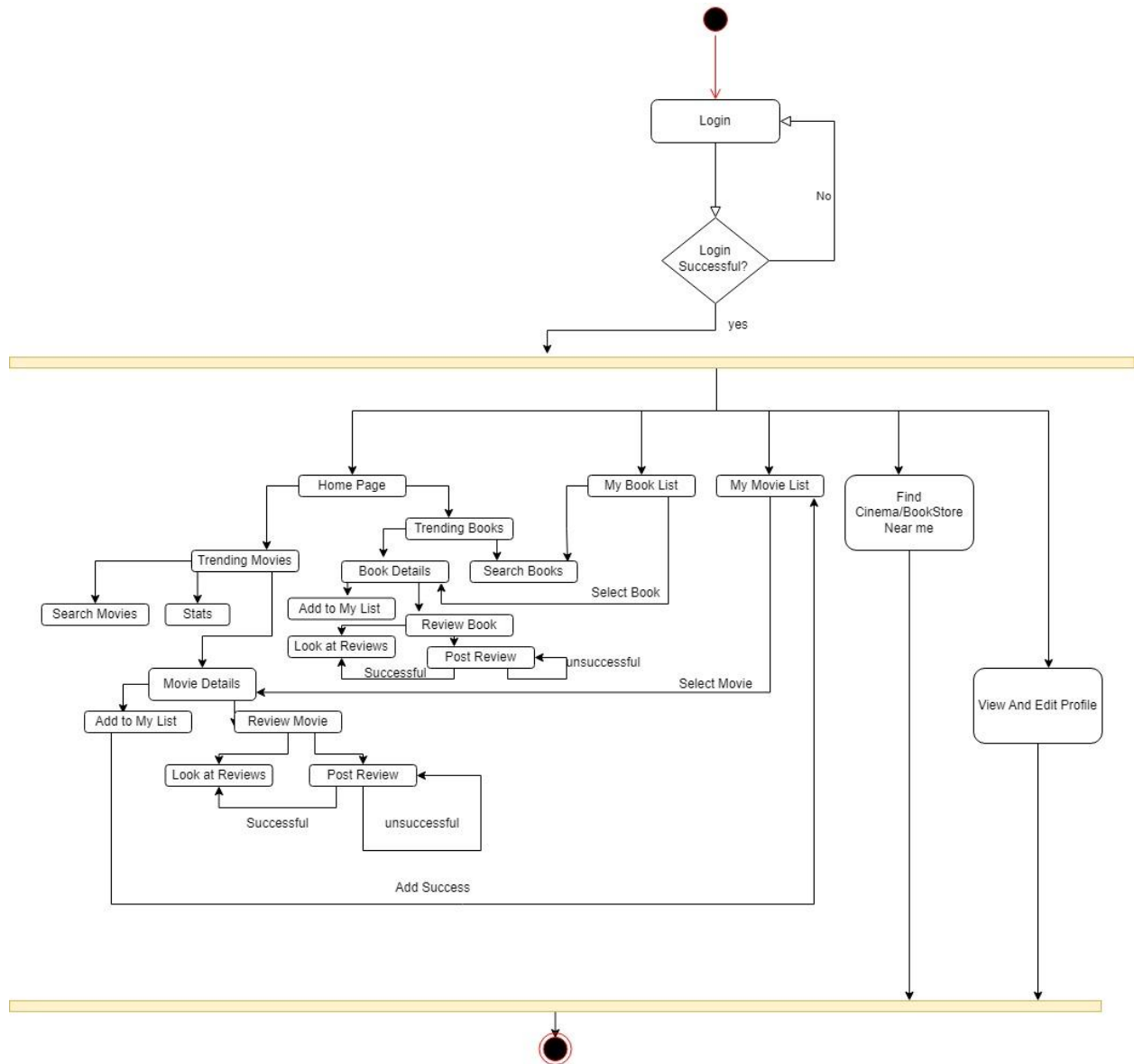
- Helped on the UI and functionality of the login\_page.dart
- Attempted to add maps and geolocation for the nearest location of movie theaters and bookstores (Unable to implement and No code pushed for it)
- Initial UI of date\_picker.dart

#### **Files worked on:**

- Date\_picker.dart
- login\_page.dart

## Code design

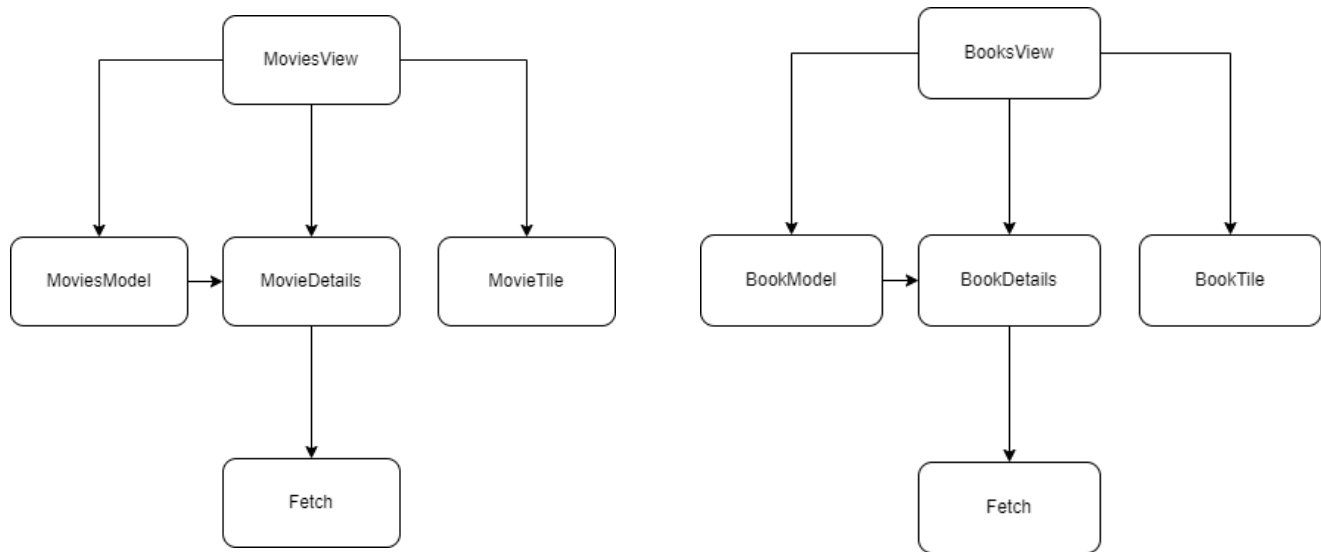
### High Level Activity Diagram



Login and Register uses Firebase Authentication to authenticate and store user credentials which are required to access full functionality of application

All web requests are executed within the various functions in the static Fetch class in models/fetch\_data.dart. Anything that needs to access data from a web API, for example to get movie data from The Movie DB, will use these functions.

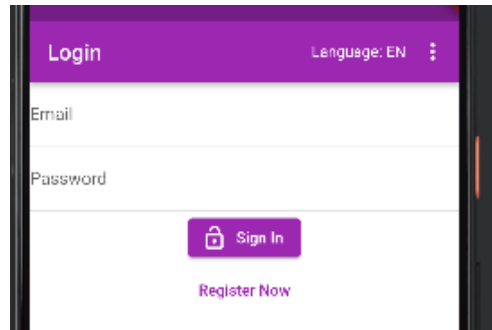
The MoviesModel and BookModel classes handle interaction between the visual components of the app and the cloud/local storage. Movies are stored in a Firestore database, whereas books are stored in a local database with sqflite. There is also a ReviewModel class which handles review data stored in the Firestore database as well.



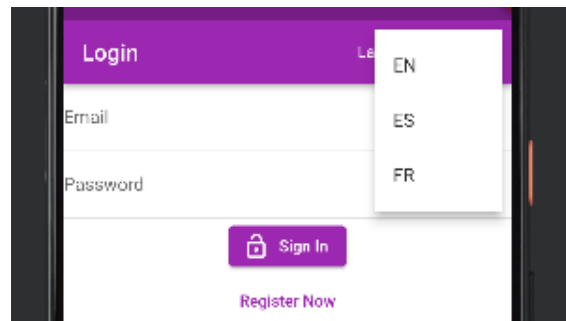
Maps are implemented using Mapbox and geolocation and geocoding is used to get users location, we also use an API to find cinemas/bookstores near the users current location to be displayed on the map.

## Users Guide

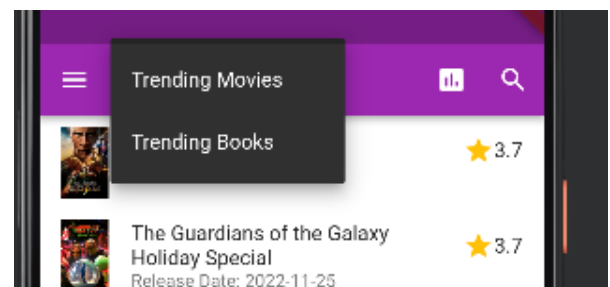
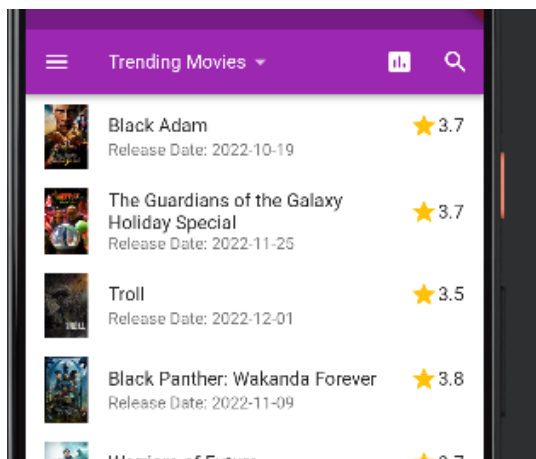
1. The first thing you'll be prompted to do upon launching our application is to login (if you're an existing user) or Register to access the full functionality of our application.



2. You can also set a preferred language at this point before logging in. We support 3 languages at the moment: English, French and Spanish. You can also switch between these languages under the Account Settings page!

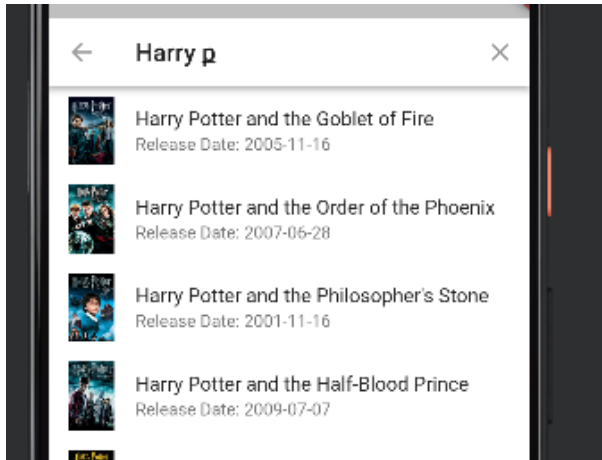


3. Once you're successfully logged in you'll be able to see our home page with the trending movies by default. You can toggle between trending movies and books based on your preference.

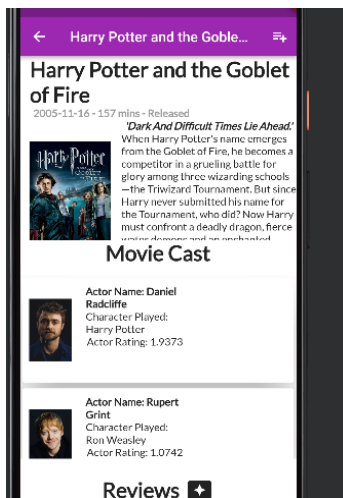




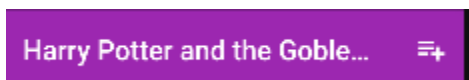
4. If you're looking for a specific movie or book you can use our search button to search for it!



5. Clicking on any movie/book on the trending or the search page takes you to a new page with detailed information about the selected movie/book

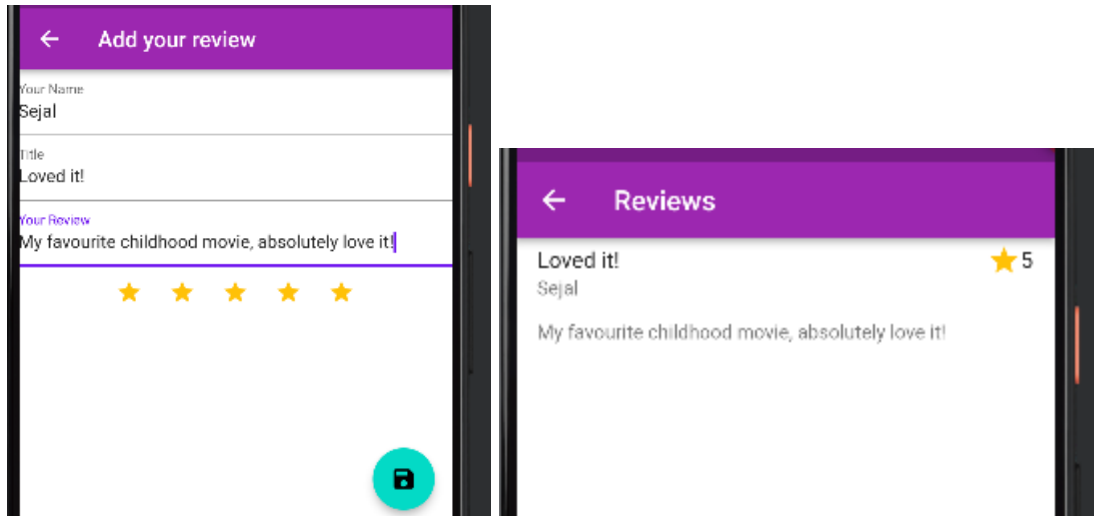


6. From here you can either add it your watch/read list using the add to list button

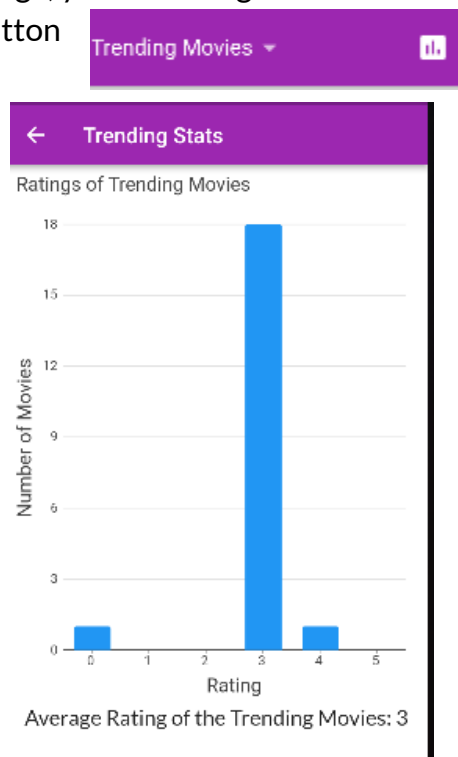


or look at the available reviews by clicking on the review button and even post

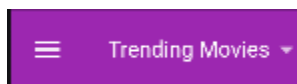
your own! **Reviews**

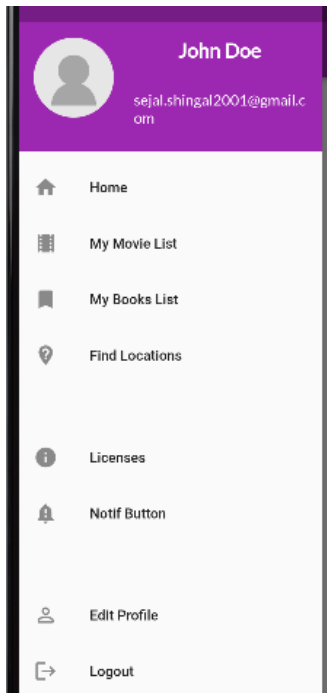


7. Going back to the home page, you can also get some stats about the trending movies using our chart button

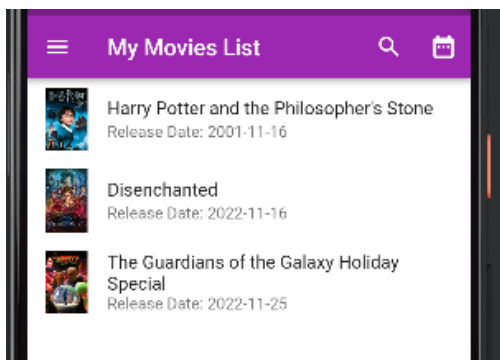


8. You can use our Navigation Drawer to navigate to other pages!

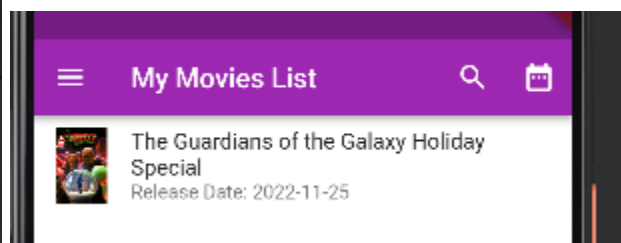
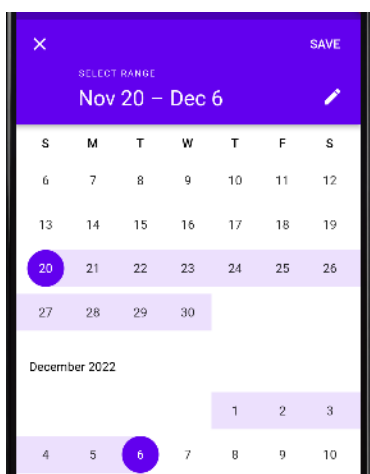




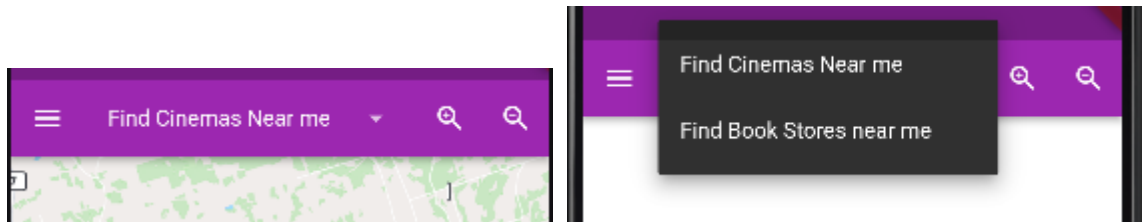
9. Your Movie and Book list are where your watch and read lists are stored.



10. You can also filter for movies in your list within a specific date range or even look up for more movies/books to add to it from here as well:

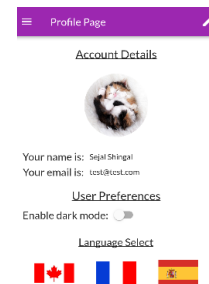


11. You can use our Find Locations tab to find Cinemas and bookstores near you! You can toggle between them as follows:

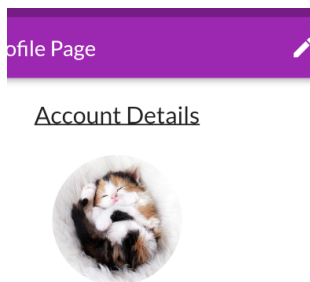


12. You can also zoom in and out of the map using the Icons on the header.

13. The last drawer option is the my profile page.

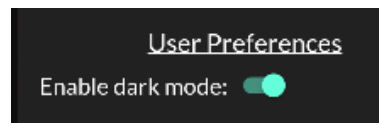


14. You can edit your profile by clicking the pencil in the top right corner. This brings you to a new page in which you can add your Full name, and profile picture link.

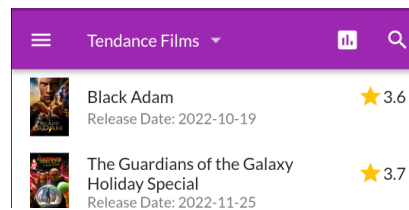
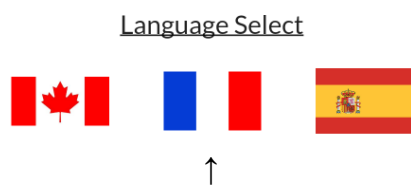


15. Additionally, one can access their user preferences on this page too. To enable dark mode click the slider, and for language selection click the flag of the language you want to select and you will be redirected to the home page with the language selected.

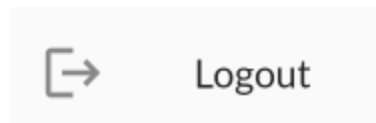
Dark mode:



## Language Selection:



16. Lastly, the user can log out of the application by clicking the logout button at the bottom of the navigation drawer. This logs the user out and redirects them to the home page.



## List of Functionalities and Usage:

- Multiple Screens and Navigation - Uses Drawer for navigation and multiple screens implemented throughout the application.
- Dialogs and Pickers - Implemented in Movie\_view to filter movies in myList based on selected date range (Filters by Release date)
- Snackbars and Notifications - Snackbars in trending\_movie\_view, trending\_books\_view, movie\_view and books\_view called when clicked on a movie or book tile to go to the detail page says "Getting Movie/Book information for \$Title"
- local storage - Local Storage is used to store users' book/read list locally.  
Files: book\_model, db\_utils
- cloud storage - Cloud Storage used to store User Specific Movie/Watch List, Reviews added by all users for each movie/book, User credentials and information  
Files: Movie\_model, review\_model, movie, review
- data tables and charts - Used to display stats for trending\_movie\_view
- Maps - Used Mapbox to display a map in map\_view and map\_bookstore
- Geolocation - Used in map\_view and map\_bookstore to get users current location
- Geocoding - same as geolocation
- Internationalization - All View pages support internationalization and assets/i18n has all three language json
- HTTPS Requests: Used in most view pages to get data, all requests being made in fetch\_data file and uses different class to build list used in view