

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | Asian Movie App | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | Shemul Hossain – X00192450 |  | | | |
|  | | | | **Patricia Torres – X00188069**—Enterprise App Developments 2—CA2 |  | | | |
|  | | |  | | |  | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  |  | | |  |
|  | INTRODUCTION | | | | | | |  |
|  |  | | |  |  | | |  |
|  |  | |  | | |  | |  |
|  |  |  | This mobile application was designed to showcase and manage a catalog of Asian films. Allowing users to browse, filter, and discover movies, as well as contribute their own content through reviews and ratings.  This is achieved by connecting the application to a RESTful web service hosted on Azure, allowing real-time interaction with a cloud-based database. | | |  |  |  |
| Decorative | | | | |
|  |  |  |  |
|  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Decorative | |  |  | | |  |  | |
|  | | System Architecture | | | | |  | |
|  | |  |  | | |  |  | |
|  | This application consists of two main components: The Android Mobile Application and A RESTful Web Service. The mobile application was built using Kotlin and Jetpack Compose. The mobile app provides the user with an interface for browsing, adding, editing, and reviewing movies. It uses Jetpack Compose for a fully declarative UI. It makes use Retrofit for network communication between the app and the backend service.  The backend was built using ASP.Net Core Web API with a code-first Entity Framework approach. This allows for easy management and migration of the database schema through C#. The backend provides APIs to allow CRUD operations for both movies and reviews, while following RESTful principles. The service also provides OpenAPI metadata allowing the use of Swagger for easy testing and exploration of endpoints.  The Backend service is hosted on Azure App Service, and the database is hosted using Azure SQL Database, which ensures high availability, automated backups, and secure storage. | | | | | | |  |
|  | Database Schema | | |  |  | | |  |
|  | Movie Table   |  |  | | --- | --- | | Id | Int(Primary Key) | | Title | String | | Genre | String | | Description | String | | ImageUrl | String | | Rating | Float | | ReleaseDate | DateTime | | | |  | Review Table   |  |  | | --- | --- | | Id | Int(Primary Key) | | ReviewerName | String | | Rating | Int(1-10) | | ReviewText | String | | MovieId | Foreign Key | | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Asian Movie App | PAGE 3 |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Service URI Scheme | | | | |  | |
|  | |  |  | | |  |  | |
|  | Movies  Http Method Endpoint Description   |  |  |  | | --- | --- | --- | | Get | /api/movies | Get paginated, filterable, and sortable movie list. Query params: title, genre, sort, order, page, pageSize | | Get | /api/movies/{id} | Get details for a specific movie by ID. Includes reviews. | | Get | /api/movies/search?title=&genre= | Search movies by title and/or genre. | | Get | /api/movies/{id}/average-rating | Get average rating of a movie. | | Get | /api/movies/top?count=5 | Get top-rated movies. | | Get | /api/movies/latest?count=5 | Get latest released movies. | | Get | /api/movies/random | Get a random movie. | | Post | /api/movies | Add a new movie. | | Put | /api/movies/{id} | Update a movie. | | Delete | /api/movies/{id} | Delete a movie. | |  |  |  | | Reviews | (Nested under Movies) |  | | Post | /api/movies/{movieId}/reviews | Add a new review to a movie. | | Put | /api/movies/{movieId}/reviews/{reviewId} | Update a review for a movie. | | Delete | /api/movies/{movieId}/reviews/{reviewId} | Delete a review from a movie. | | | | | | | |  |
|  |  | | |  |  | | |  |
|  |  | | |  |  | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Asian Movie App | PAGE 4 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Shemul Hossain’s Contributions | | |  | |
|  | |  |  |  |  | |
|  | My contributions to this project are as follows I set up the GitHub repository for this project. I then went on to set up the backend. This includes making the models for the database. I then went about setting up the services, this meant I had to make all the APIs in the controller cs file that would be called by the frontend. I also wrote the Unit Tests for those services to ensure they worked as intended and were reliable. After that I went about hosting the database on Azure SQL Database and deploying the services to Azure App Service, this was done through Visual Studios publish feature.  Backend Testing Report   |  | | --- | |  |   The unit tests I wrote can be found in the tests project and run for all the services, this ensures that they work as intended, and as shown in the image all 17-unit tests have passed. Showing that the services are indeed working as intended. | | | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Asian Movie App | PAGE 5 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Shemul Hossain’s Contributions | | |  | |
|  | |  |  |  |  | |
|  | While I was mainly in charge of setting up the backend I also did work on the front end. I created the android project through android studio. I then made some of the features of the app. This includes making the movie list screen, which displays a scrollable list of movies fetched from the RESTful backend. I added the movie details screen, which is the screen you go to when you click on a movie. This screen provides more details on the selected movie. I added the add movie functionality, this can be accessed through a FAB button and allows the user to add a movie, this movie then gets saved to the database.  Inside the movie details screen I added the ability to edit/delete the chosen movie. In that same details screen I also added the ability to view/add/edit/delete reviews. In the home screen I then added the functionality to filter the list of movies by the genre, sort the movie by ratings and release date. I also added the Top Rated and Latest buttons which return the top 8 rated movies and the 8 most recent movies. I added the random movies button as a surprise me type feature allowing users to get a random movie from the list. I also implemented a search feature to allow users to search for the movie they want. All the features are built off the RESTful APIs det up in the backend. I also handled the Internationalization of the app into French. I also worked on the end-to-end testing of the front end, using espresso. I wrote 7 tests, to test the functionality Lastly, I added some data, so the app had movies and reviews to look at/explore.  Frontend Test Report   |  | | --- | |  | |  | |  | |  | |  | |  | |  | | | | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Asian Movie App | PAGE 6 |  |

### Patricia Torres’s Contributions

During the development of the mobile application, I primarily focused on the frontend to refine the overall layout and design using Kotlin in Android Studio. My goal was to ensure that the user interface was intuitive, visually appealing, and functionally seamless for users searching and reviewing movies. I worked on various UI elements, improving responsiveness and usability, while maintaining consistency across different screen sizes.

One of the key features I implemented was the average rating system, which allows users to see an aggregated score based on all submitted ratings for each movie. This functionality was designed to enhance user engagement by providing a clear indication of how well a particular movie was received by the community. To achieve this, I built a system that dynamically collects ratings from users submitting reviews, sums them up, and calculates the average rating, which is then displayed beneath the movie description. This feature was developed using the “Average-Rating” function located in the Controller file, ensuring proper integration with the app's backend logic while maintaining smooth data flow between the user interface and stored ratings.

Additionally, I took charge of the internationalization process, adapting the app to support German localization. This involved translating all UI text elements to German, ensuring that all menus, buttons, and labels were correctly displayed for German-speaking users. I also worked on formatting adjustments to accommodate linguistic differences while maintaining a consistent design structure across all supported languages.

Beyond UI refinements, I also contributed to testing and validation by writing a test for the search functionality of the application. This test ensures that the search feature correctly retrieves movies based on user input, handles edge cases, and maintains efficiency across various queries. By implementing structured test cases, I was able to verify the reliability and accuracy of the search mechanism, helping to improve overall app performance.

Through these efforts, I contributed significantly to both the user experience and core functionality of the application, ensuring that it is visually polished, feature-rich, and well-tested for optimal usability.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | App Flow Screenshots | | |  | |
|  | |  |  |  |  | |
|  | |  | | --- | | Home Screen | |  | | | | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Asian Movie App | PAGE 7 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | App Flow Screenshots | | |  | |
|  | |  |  |  |  | |
|  | |  | | --- | | Add Movie Screen | |  | | | | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Asian Movie App | PAGE 8 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | App Flow Screenshots | | |  | |
|  | |  |  |  |  | |
|  | |  | | --- | | Movie Details Screen | |  | | | | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Asian Movie App | PAGE 9 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | App Flow Screenshots | | |  | |
|  | |  |  |  |  | |
|  | |  | | --- | | French Internationalization | |  | |  | | | | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Asian Movie App | PAGE 10 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | App Flow Screenshots | | |  | |
|  | |  |  |  |  | |
|  | |  | | --- | | French Internationalization | |  | |  | | | | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Asian Movie App | PAGE 11 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | App Flow Screenshots | | |  | |
|  | |  |  |  |  | |
|  | |  | | --- | | German Internationalization | |  | |  | | | | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Asian Movie App |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Appendix | | |  | |
|  | |  |  |  |  | |
|  | GitHub Repository: <https://github.com/ShemulHossain/asian-movie-app>  Azure App Service Domain: <https://asianmovieapi20250423025652.azurewebsites.net/>  Live API Swagger: <https://asianmovieapi20250423025652.azurewebsites.net/swagger/index.html>  Things to Note:  When launching the application, you might have to close it and open it again (maybe a few times). This is due to the backend being put to sleep after a period of inactivity.  The location of the APK: asian-movie-app\apk\app-debug.apk | | | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Asian Movie App | PAGE 12 |  |