Objectives:

In this assignment, you are to make use of what you have learnt in this module to design and develop a Movie viewer application on the Android platform using JetPack compose.

Scenario:

PopCornMovie is a company that allow their users to view movies details. They are looking for a mobile application that allows users to view movie details. They have decided to hire you to develop this application on the Android platform.

Deadline

- 2nd Feb 2025, 23:59

This assignment holds a total of 30% of your final ICA. This paper has a total of 100 marks.

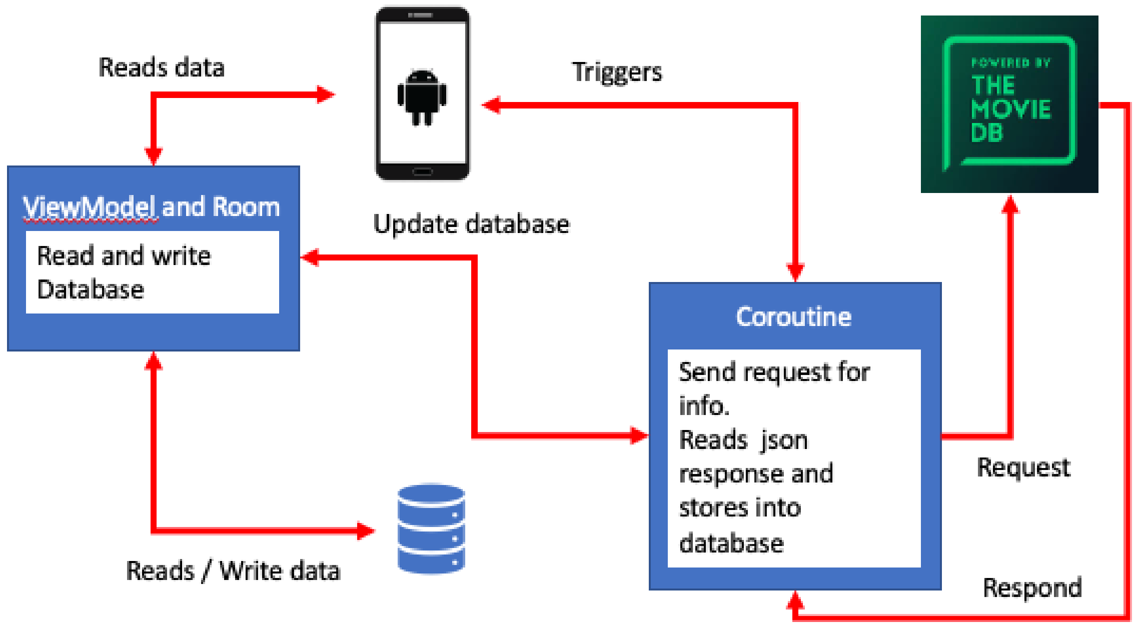
Listed below are the deliverables that is expected from you.
Deliverables:

- Zipped file with
  - source codes that fulfils the assignment requirements. (21212A.zip)
  - powerpoint file. (21212A.pptx)
  - APK file. (21212A.apk)

Instructions:

- This is an INDIVIDUAL assignment. Students are to submit their assignment to BrightSpace.
- Zip up the above deliverables and name it after your admin no.
- Marks will be deducted for the following conditions.
  - Applications that cannot run upon first time installation
  - Late submissions
  - Not submitting work based on instructions given in the assignment
- Students caught plagiarizing from other source (Internet, friends, etc..) will cause their submission to be voided.
- All projects must be error free and is able to build and run upon open.
- Project details
  - Minimal SDK: API 34
  - Application Name: [Admin Number] Movie Viewer
  - Package: com.it2161.[Admin number].movieviewer
  - Emulator: Pixel 8 Pro

Overview



The application that you will be creating will make use of “Retrofit” to request from “TheMovieDB” the information that it requires. “TheMovieDB” will response with the data in JSON format. Your app should then make use of a coroutine to retrieve and store the data into the database using the view models and room.

Implementation

The below states the features required for this mobile application. The URL that describes the request and response body of the API to be used for the implantation of the feature. In the case where a URL is not provided, the feature does not require any API call to “TheMovieDB”. Your job is to go through the feature specification stated below, design and develop an Android application.

Basic

1. Login and registration

   User must be able to register and login to access the application.

   Details should include at least:

   - User id
   - Password
   - Preferred name

   [5 marks]

2. View Profile

   User must be able to view the details mentioned in #1.

   [5 marks]

3. Movie List

   At the landing screen, the user must be able to toggle between the lists below.

   - Popular [Default]

     URL: https://developer.themoviedb.org/reference/movie-popular-list
   - Top Rated

     URL: https://developer.themoviedb.org/reference/movie-top-rated-list
   - Now playing

     URL: https://developer.themoviedb.org/reference/movie-now-playing-list
   - Upcoming

     URL: https://developer.themoviedb.org/reference/movie-upcoming-list

   Each movie detail comes with an image, make use of the instructions in the following link to retrieve the image: https://developer.themoviedb.org/docs/image-basics

   [4 x 5 marks]

4. Movie detail

   The user must be able view the movie details after selecting it from a list of movies. Display the following details provided by "TheMovieDB".

adult	genres	Original language	Title
Release date	Run time	Vote count	overview
Vote average	revenue		

   URL: https://developer.themoviedb.org/reference/movie-details

   [5 marks]

5. Reviews

   Display the reviews written by other users regarding a movie.

   URL: https://developer.themoviedb.org/reference/movie-reviews

   [5 marks]

Advanced

1. Persistent user data
   - Registered user information will be stored in the device until the data is cleared or after the application is removed from the device.

   [5 marks]

2. Favorite movie list
   - The list will be stored locally in the device. The list will be a new list for a new registered user in the device.
   - Tapping on an item in the list will navigate the user to a similar detail list as #3 above.

   [10 marks]

3. Offline support
   - All movie list information to be stored in database via Room.

   [5 marks]

   - All movie list and detail must be available when phone goes offline.

   [3 marks]

   - All other features will displayed as “unavailable” to the user.

   [2 marks]

4. Search

   Allow user to search for a movie(s) based on a keyword provided by the user. Display to the user the results provided by “TheMovieDB”.

   [5 marks]

   URL: https://developer.themoviedb.org/reference/search-keyword

5. Similar movies

   Recommend to the users, similar movies based on another movie.

   URL: https://developer.themoviedb.org/reference/movie-similar

   [5 marks]

UI/UX

The mobile application will also be accessed based on the following:

- Visual design
  - Balance:
    - Create visually balanced screens with a thoughtful distribution of elements (text, buttons, images).
    - Avoid clutter by leveraging white space effectively.
  - Alignment:
    - Align elements to a grid or structure for neatness and a professional appearance.
  - Hierarchy:
    - Use size, color, and placement to prioritize content and guide the user's attention.
    - For example, make the primary action button more prominent than secondary actions.
- Consistency
  - Consistent use of fonts, colors, iconography, and spacing throughout the app.
  - Ensure similar UI elements (e.g., buttons, cards) look and feel the same across screens
  - Similar actions should produce similar results across the app.
    - Example: A "Save" button should always be in the same location and behave the same way.
  - Provide consistent feedback (e.g., all errors use the same style of message box, success notifications follow the same format).
- Ease of use
  - Intuitive Navigation
    - Clear Structure
      - Navigation menus (e.g., bottom navigation bars, hamburger menus) should be simple and accessible.
      - Include a logical hierarchy that makes it easy for users to locate features and content.
    - Obvious Call-to-Action (CTA)
      - Buttons, links, and other actionable elements should be visually distinct and labeled with clear, concise text.
      - Use recognizable icons with text labels when necessary.

- Minimal Cognitive Load
  - Simplicity:
    - Minimize the number of steps required to complete tasks (e.g., searching for a movie or adding it to a watchlist).
    - Avoid overwhelming users with too many options or excessive information on a single screen.

- Feedback and error handling
  - State Indicators:
    - Clearly differentiate between active, inactive, and hovered/selected states of UI elements (e.g., buttons, tabs).
  - Loading Indicators:
    - Use spinners, progress bars, or skeleton screens to indicate loading and improve perceived performance.
  - Animations and Transitions:
    - Add subtle animations to enhance interactivity (e.g., button presses, screen transitions), but avoid overusing them.

[10 marks]

<u>Rubrics for UI/UX</u>

	Advance	Proficient	Functional	Developing	Incomplete
Visual design [2 marks]	• Fully adheres to material design • Layout is visually appealing, professional and polished • Smooth animations and transitions enhance the experience without overwhelming the user	• Adheres to Material Design 3 principles with minor inconsistencies. • Overall design is clean and aesthetically pleasing, but lacks polish or advanced features.	• Layout is functional but lacks attention to detail or refinement	UI appears cluttered, outdated, or unprofessional.	UI is incomplete or lacks basic usability.
Consistency [2 marks]	• UI elements are consistently styled and behave predictably across all screens.	UI elements are mostly consistent, with occasional minor deviations.	Inconsistent styling across screens (e.g., varying fonts, button styles).	Significant inconsistencies in UI styling and behaviour.	No consistency in styling or behaviour
Ease of use [2 marks]	• Intuitive navigation with clear call-to-actions and no unnecessary complexity. • Users can complete core	Navigation is straightforward, but some tasks may require additional effort or clarification.	Navigation is somewhat intuitive, but some features are difficult to find or use.	Navigation is confusing, with unclear or missing call-to-actions.	Navigation is broken or extremely confusing.

	tasks (e.g., searching for movies, viewing details, managing watchlist) easily and quickly				
Feedback and error handling [4 marks]	• Users receive clear feedback on interactions (e.g., loading indicators, success/error messages). • Graceful error handling for API failures, empty states, or offline scenarios.	• Basic feedback is provided, but some actions lack visual or textual cues. • Error handling is present but not comprehensive.	Limited feedback; users may be unsure if actions are successful or if errors occur	No feedback for user actions, and errors are not handled gracefully.	No feedback for user actions; app crashes or fails without explanation.

Documentations

Making use of PowerPoint slides, provide the following information:

- Screen flow using screenshots of each screen done.
  - Each screen to indicate the UI/UX improvement made and the reasons for it.

[5 marks]

- Flow chart to explain how your codes perform the following:
  - Data request from “TheMovieDB” till the movie list is displayed to the user

    [5 marks]
  - Saving of Favorite movie list into database

    [2 marks]
  - Retrieval of Favorite movie list from database till it is displayed to the user

    [3 marks]

The flow chart should include the methods and class used. Marks are given for completeness and accuracy.