University of Westminster
Department of Computer Science

5COSC011C Coursework 2 (Semester 2)

Module leader	Torin Wirasingha
Unit	Coursework 2
Weighting:	50%
Qualifying mark	30%
Description	
Learning Outcomes Covered in this Assignment:	LO1, LO2, LO3, LO4, LO5
Handed Out:	15/3/2021
Due Date	13/4/2021 13:00
Expected deliverables	Source code/XML files/Resources (images, etc)
Method of Submission:	Online via Blackboard
Type of Feedback and Due Date:	Individual feedback via Blackboard within 3 weeks of submission **All marks will remain provisional until formally agreed by an Assessment Board.**

Assessment regulations

Refer to section 4 of the "How you study" guide for undergraduate students for a clarification of how you are assessed, penalties and late submissions, what constitutes plagiarism etc.

Penalty for Late Submission

If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 40 – 49%, in which case the mark will be capped at the pass mark (40%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid.

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website :http://www.westminster.ac.uk/study/current-students/resources/academic-regulations

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero

shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website:**http://www.westminster.ac.uk/study/current-students/resources/academic-regulations**

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website:**http://www.westminster.ac.uk/study/current-students/resources/academic-regulations**

5COSC011C MOBILE APPLICATION DEVELOPMENT - Assignment 2
Deadline 13/4/2021, 13:00

Torin Wirasingha
Email: torin.w@iit.ac.lk

Description

You are required to implement an Android application described by the specifications below.

You are not allowed to use third-party libraries. The only libraries you can use are the standard Android API libraries found in the following URL:

https://developer.android.com/reference/

It is important to follow exactly the specifications and your implementation must conform to these:

The application developed will help the user to keep track of the movies he/she has watched, organise them and look up information about them from the IMDB database.

1. When the application starts, it presents the user with 6 buttons labelled *Register Movie*, *Display Movies*, *Favourites*, *Edit Movies*, *Search* and *Ratings*.
   (3 marks)

2. Clicking on the *Register Movie* button, it should display to the user a page where the user can register (create) a movie that he/she has watched. The page should contain 6 textboxes which can be used to enter the title of the movie, the year, the director, a list of actors/actresses separated by commas, rating and review. For example, the user can enter the following information corresponding to the 6 textboxes "Godzilla vs. Kong", "2021", "Adam Wingard", "Alexander Skarsgård, Millie Bobby Brown, Rebecca Hall, Brian Tyree Henry, Shun Oguri, Eiza González, Julian Dennison", "6", "A fast-paced action movie keeping you in suspense for 90 minutes. Excellent performance by Shun Oguri.".

   The year textbox should accept only numeric values which are greater than 1895. The rating textbox should accept only numeric values corresponding to the range of $1-10$ (the user gives between 1 and 10 stars as his rating)

   The page should also contain a *Save* button, which when pressed saves all the details entered in an SQLite database local to the mobile device.
   (11 marks)

3. Clicking on the *Display Movies* button it displays to the user a list of movies sorted in alphabetical order (one per line using the title of the movie entered in the previous question). A tickbox next to each movie is also displayed (the tickbox should initially be blank, i.e. not ticked). The displayed movies should come from the database and they should correspond to movies which the user created using the *Register Movie* functionality in this instance of running the application or a previous instance that the application was run (i.e. the application should NOT delete any saved movies when it exits and all data are persisted in the database).

If the list of movies is more than one page long, it should provide scrollbars that the user can use to see all the movies.

The page should also provide a button *Add to Favourites*. The user can select which movies he/she wishes to add as favourites by using the tickboxes and subsequently pressing the *Add to Favourites* button the application will add the selected movies to the database as favourites (for this part, simply add an extra column in the database table containing the list of all all movies).

(11 marks)

4. Pressing the *Favourites* button the application should display to the user a list of all his/her favourite movies (one per line), i.e. all the movies that the user added to his favourites list using the *Add to Favourites* button in the *Display Movies* page in the previous question. Next to each movie there is a checked tickbox (ticked it means it is a favourite and as only favourite movies are shown in this page, all tickboxes should be initially checked).

A button titled *Save* is also present in the page. The user can deselect any favourite movie by unchecking the tickbox for any movie(s) and subsequently pressing *Save* will update the list of favourite movies in the database, i.e. the status of any unselected movie will be updated in the database and become not favourite.

(12 marks)

5. Clicking on the *Edit Movies* button the list of all watched movies in the database will appear (i.e. all movies watched whether they are favourites or not favourites)

After the list of movies appear, the user can click on the name of any movie and a new page will display all the details of that movie from the database, i.e. title, year, director, actors, rating, review and its favourite status indicated as "Favourite" or "Not Favourite". The rating should appear by displaying the appropriate number of stars in yellow colour (e.g. if 6 is given then 6 stars should appear in yellow, followed by 4 stars in white colour).

The user should be able to edit the information displayed, in order to change any of the details of the movie including whether it is a favourite or not, and then pressing an *Update* button which is also present on the page will update the details of the movie in the database. The user should be able to edit the rating by clicking in any of stars which will becoming yellow (active) or white (inactive). E.g. a rating of 6 should be indicated by 6 yellow starts followed by 4 white stars (6 out of 10).

(10 marks)

6. Pressing the *Search* button, the user is presented with a single textbox and a `Lookup` button. The user can type in the textbox any string and subsequently press the `Lookup` button to display all movies in the database which contain the typed string in either the *Title* or the *Director* or *Actors* fields of the movie in the database.

2

The search should be case insensitive and a match does not have to be a complete match but partial. I.e. if the user types the string "rUISE" the displayed movie(s) could be, a movie which contains "Tom Cruise" in the list of actors or containing in its Title "The Cruise to London".

(10 marks)

7. (a) Pressing the *Ratings* button the user is presented with the list of movies available in his database and a *Find movie in IMDB* button.

The user can select which movie he/she would like to lookup its IMDB rating, by selecting a movie title (for simplicity you can use radio buttons next to each movie) .

After the selection of a movie the user can click on the *Find movie in IMDB* button to display the list of titles and their ratings (ONLY the titles of the movies) which matches with the title of the selected movie.

The list of movies should come from the following web service:

`https://imdb-api.com/`

You will need to create a free account on that web service which gives you a unique key which you can use to make queries.

You will need to parse the returned JSON formatted response after you send a similar request with the following (you need to replace the YOUR_API_KEY with your own key):

`https://imdb-api.com/en/API/SearchTitle/YOUR_API_KEY/Inception 2010`

The above retrieves from the web service all titles which contain both the title "Inception" and 2010 as the year.

For example, it returns:

```json
{
    "searchType":"Title",
    "expression":"Inception 2010",
    "results":[
      {
        "id":"tt1886283",
        "resultType":"Title",
        "image":"https://imdb-api.com/images/original/nopicture.jpg",
        "title":"The Distressed Watcher",
        "description":"(2010{2011)"
      },
      {
        "id":"tt2078432",
        "resultType":"Title",
        "image":"https://imdb-api.com/images/original/nopicture.jpg",
        "title":"Travel Companions",
        "description":"(2010{2012)"
      },
      {
        "id":"tt2135194",
        "resultType":"Title",
        "image":"https://imdb-api.com/images/original/nopicture.jpg",
        "title":"Hollywood on Set",
```

```
                      "description":"(2003{ )"
                },
                {
                      "id":"tt2135220",
                      "resultType":"Title",
                      "image":"https://imdb-api.com/images/original/nopicture.jpg",
                      "title":"Hollywood on Set",
                      "description":"(2003{ )"
                },
                {
                      "id":"tt2160855",
                      "resultType":"Title",
                      "image":"https://imdb-api.com/images/original/MV5BODdiMzIzMjEtMj
                                      E1MC00ZjY0LWE1NTctOGYwMWYwMTQzNGE0XkEyXkFqcGdeQX
                                      VyMjg2MTMyNTM@._V1_Ratio1.7600_AL_.jpg",
                      "title":"How It Should Have Ended",
                      "description":"(2005{ )"
                }
          ],
          "errorMessage":""
    }
```

The returned **id** (starting with **tt**, the first title above returns an id of **tt1375666**) which then can be used to extract the rating of that title:

`https://imdb-api.com/en/API/UserRatings/YOUR_API_KEY/tt1375666`

This request returns a JSON format reply containing the rating (8.8) of the title with ID tt1375666. The rating can be found in the `totalRating` field.

(15 marks)

(b) **This subquestion should only be attempted if you completed the previous "Ratings" subquestion. Hardcoded solutions, i.e. ratings not coming from the web service will not receive any marks**.

Expand the "Ratings" functionality so that when the user clicks on the title of a movie, the image of the movie is displayed in an `ImageView`. For this you will need to extract the URL corresponding to the movie image from the JSON response, i.e. the "image" field which follows the "title" field.

`(8 marks)`

Marking Scheme: The marks achieved for each part of the program are indicated in the description of the task above. In addition to these the following will be taken into account:

- *Code readability* (structure, comments, variable naming, etc,): 5%

- *Implementation* (e.g. quality, efficiency, look and feel of the application, based on fonts, colours, etc.): 15%

The maximum for work which does not compile (or XML files with syntax errors causing the Java code not to compile) is 30%.

Submission of assignments using a different method other than Blackboard will not be accepted and zero (0) marks will be awarded in such cases.

Deadline: **Tuesday 13th of April 2021, 13:00.**

Submission Instructions

Files to submit: All of the files of the Android Studio project of your application in a zip file. To simplify testing your submitted project should include 10 films with 4 marked as favourites in the application database.

Referencing code: Any code taken from other resources (i.e. a textbook or internet) should be referenced in comments within your code (full textbook details or full web URL), identifying the exact code that you used it as part of your application and the exact portions of the original source code that you reused.

You should submit via BlackBoard's Assignment functionality (do NOT use email, as email submissions will be ignored.), all the files described above. A single zip file with the name `wNNNNNNNN` (where `wNNNNNNN` is your university ID login name) containing all the above files could be submitted alternatively. You can create such a file by using the main menu in Android Studio and choose `File->Manage IDE Settings->Export to Zip File...`

Note that Blackboard will allow to make a submission multiple times. Make sure before submitting (i.e. before pressing the `Submit` button), that all the files you want to submit are contained there (or in the zip file you submit).

In the case of more than one submissions, only your last submission before the deadline given to you will be marked, so make sure that all the files are included in the last submission attempt and the last attempt is before the coursework deadline.

Request to mark submissions which are earlier than the last submission before the given deadline will be ignored as it is your responsibility to make sure everything is included in your last submission.

The following describes how to submit your work via BlackBoard:

1. Access `https://learning.westminster.ac.uk` and login using your username and password (if either of those is not known to you, contact the Service Desk, tel: +44 (0) 207 915 5488 or log a call via `https://servicedesk.westminster.ac.uk`.).

2. Click on the module's name, MODULE: 5COSC011C.2 Mobile Application Development (IIT Sri Lanka) found under `My Modules & Courses`.

3. Click on the `Assessment->Submit Coursework 2->Coursework`.

4. Click on `View Assignment`.

5. Attach your zip file containing all the Java source code files, by using the `Browse` button.

6. Create a Word or PDF file with the following information:

- *Comments*: Type your full name and your registration number, followed by:

  "I confirm that I understand what plagiarism is and have read and understood the section on Assessment Offences in the Essential Information for Students. The work that I have submitted is entirely my own. Any work from other authors is duly referenced and acknowledged."

7. Attach the file with the statement above.

8. Check that you have attached both the zip and the statement file.

9. Click the `Submit` button.

If Blackboard is unavailable before the deadline you must email the Registry at `com_submission@iit.ac.lk` with cc: to myself and your personal tutor before the deadline with a copy of the assignment, following the naming, title and comments conventions as given above and stating the time that you tried to access Blackboard. You are still expected to submit your assignment via Blackboard. Please keep checking Blackboard's availability at regular intervals up to and after the deadline for submission. You must submit your coursework through Blackboard as soon as you can after Blackboard becomes available again even if you have also emailed the coursework to the above recipients.