|  |  |  |  |
| --- | --- | --- | --- |
| **CSY2061: Mobile Application Development 1** | | | |
| Date of Issue |  | **Last Date for**  **Submission:** | **Sunday, 12th June**  **23:59:59** |
|  | | Module Tutor: | Amit Chaulagain (amit.chaulagain@nami.edu.np) |
|  | | | |

|  |
| --- |
|  |

**Guidelines – Please read carefully:**

1. The University of Northampton’s Policy on Academic Integrity and Misconduct will be strictly implemented.
2. This is not a group project and by submitting this assignment you are asserting that this submission is entirely your own individual work. You may discuss the assignment with other students, but any code written should be your own. **Sharing your work with another student or submitting code that was written by someone else may be deemed academic misconduct.**
3. If you have used any external code that you did not write, you must clearly reference it within your report.
4. You must submit all items of the assessment according to the submission procedure stated in this document. **Failure to follow the submission procedure may result in a penalty or capped grade.**

**Brief:**

Design, implement and test an application that allows users to register for an application that allows users to post comments. The system should enable users to register. Here are the specific requirements:

**Basic System Requirements:**

The application must allow the users to:

1. Register for the application (Full name, Email Address, Date Registered, Date Updated, Password etc)
2. Login into the application
3. Update user profile
   1. When updating the profile page, the Date Updated column should be updated to show the user when they last updated their profile details
4. All registered users should be able to access the General Timeline activity. Whilst on the General Timeline activity, users should be able to do the following:
5. Post a comment for all registered users in the application to see
6. When a user posts, the post should show the date of the post, the name of the user who posted.

**Enhancements (in order of importance – high to low):**

Additional features that you may include are as follows:

1. All registered users should be able to access My Timeline activity
   1. The My Timeline activity should show all the posts for the user who is currently logged into the application.
   2. All the posts must show the date of the post.
   3. The registered user should be able to delete any of their post
   4. The registered user should be able to update any of the post
2. An administrator should be able to log into the application
3. The administrator should be able to view all the registered users
4. The administrator should be able to delete users
5. When a registered user’s account is deleted, their posts should also be deleted.
6. The administrator should be able to export all users to a file.

These additional features are only suggestions and are not exhaustive. You may include any other useful features relevant to this application. You need to clarify/extract more detailed requirements from the tutor and are encouraged to use an agile methodology for software development.

**Deliverables:**

**All** requirements (A, B and C below) **MUST** be delivered to achieve a passing grade for this assignment.

1. Technical Report

The report should consist of the following sections (in the same order):

1. Username and password for all relevant accounts (if implemented)
2. A list of all the features implemented in a tabular format. For example:

|  |  |  |
| --- | --- | --- |
| **Feature** | **Implemented (Partial/Full)** | **Any comments** |
| Add a new post | Full | No error validation. |
| View all my post | Full |  |

1. Explanation of the main sections/fragments of the code. Provide information that would be useful for another developer (not an end user!) who may want to extend/maintain your system. You may want to refer to the class diagrams to explain code.
2. Screenshots of the system showing all key features
3. Evidence of Testing:
   1. Blackbox Testing: Test **logs** providing information of all the tests carried out (including any failed tests for functionality not implemented)
      1. Please provide screenshots and tables as part of the testing.
   2. List of any bugs and/or weaknesses in your system (if you do not think there are any, then say so). Bugs that are declared in this list will lose you fewer marks than ones that you do not declare.
4. References (use Harvard referencing):

If you have borrowed some code from elsewhere (e.g., from a book or some resource on the web you

**must** indicate clearly what they are and include references).

1. Source Code

The source code must be well documented with necessary comments. Consistent and clear indentation of the code is also important. Source code needs to be submitted in two forms:

1. As a single ZIP archive (.zip file consisting of all android files (java, xml)).
2. A commented full listing in a separate Word document named “Full Source Code Listing”.
3. Video Demonstration

In addition to the report, you **must** submit a video demo (URL) of your assignment. The demo should be about 10 minutes long (maximum:15 minutes) and should cover all your work in a logical way. You should explain the main phases of design and implementation covering the main fragments of code. Your voice needs to be clear for the marker to hear. It should also include a walkthrough of using the software and must demonstrate the key features. **The module tutor reserves the right to invite you for an online viva-voce. Poor demo/viva could negatively influence other sections in the marking criteria.**

**Submission Procedure:**

* E-Submission of documents through Turnitin on NILE as TWO separate WORD documents.

[Document 1 = **Report** & Document 2 = **FullSourceCodeListing**]

To do this, go to the NILE site for this module and use the link labelled ‘Submit your work’.

* E-Submission of a single ZIP archive that contains all the source code files (.java), (.xml), data files, ReadMe file and executable (jar). The archive must be named with your student ID, e.g. *12345678.zip* where 12345678 is your student ID. To do this, go to to the NILE site for this module and use the link labelled ‘Submit your work’. Clicking on the link (*SourceCodeEsubmission*), will take you into the submission form, where you can upload your ZIP archive using the ‘Attach File’ button (*Browse for Local File*). Finally, click the *Submit* button.
* When submitting your video demonstration, use of Kaltura (<https://video.northampton.ac.uk/>) is recommended. You must ensure that the video link is accessible to the marker (do not set it to private access).
* Failure to follow the above submission guidelines may result in a capped or fail grade.

**Marking Criteria:**

The grade for this assignment will form 100% of the overall assignment grade for the module. Marks are split according to the following scheme. In general, the following criteria will act as a guide to what you should expect:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | | F | G |
| Design (15%) | Excellent design of program and user interface. Adherence to object-oriented principles. Class diagrams are very well designed and presented. | Good quality design of program and user interface. Adequate adoption of object-oriented principles. Class diagrams are well designed and presented. | | Satisfactory design of program and user interface.  Class diagrams are satisfactory. | Faulty design of class diagrams. Very little discussion of the overall design. | No submission or no submission of merit |
| Functionality  (35%) | All criteria for (B) and many significant additional features. | All criteria for (C) and some significant additional features. | Most basic system requirements are met. | | Most basic system requirements are not met. | No submission or no submission of merit |
| Testing (15%) | Evidence of both white box and black box testing with extensive code coverage. | Evidence of both white box and black box testing with good code coverage. | Evidence of either white box or black box testing with satisfactory code coverage. | | No evidence of any white box or black box testing. | No submission or no submission of merit |
| Technical Report (10%) | Excellent description of main sections of the code. All sections covered in the right order. Any assumptions and bugs/weaknesses are clearly stated. | Good description of main sections of the code. All sections covered in the right order. | Satisfactory description of main sections of the code. Most sections are covered. | | No description of code sections. | No submission or no submission of merit |
| Code quality and Efficiency (15%) | Code is very well structured to enable reusability and debugging. Excellent work on error handling. | Code is well structured to enable reusability and debugging. Good work on error handling. | Some thought has been given on how the code is structured. Some work on error handling. | | Hardly any thought on how the code is structured. | No submission or no submission of merit |
| Demonstration  (10%) | Covers all implemented features in sufficient detail. Any known bugs are highlighted. Validation is tested (e.g., entering invalid values) | Covers all implemented features in sufficient detail | Satisfactory demo but needs more details. | | Poor explanation. Either not concise or difficult to understand. | No submission or no submission of merit |