|  |  |  |
| --- | --- | --- |
| **Course:** COMP1661  Application Development for Mobile Devices | **Coursework Number:**  1 of 2 | **Contribution:** 80% of course |
| **Greenwich Coordinator:**  Mohammed Hassouna | **TERM 2 2014/15** | **Deadline:**  **TBC** |
| **This coursework should take an average student who is up-to-date with tutorial work approximately 40 hours** | | |
| **Learning Outcomes:** A, B and C | | |

|  |
| --- |
| **Plagiarism** is presenting somebody else’s work as your own. It includes: copying information directly from the Web or books without referencing the material; submitting joint coursework as an individual effort; copying another student’s coursework; stealing or buying coursework from someone else and submitting it as your own work.  Suspected plagiarism will be investigated and if found to have occurred will be dealt with according to the procedures set down by the University.   **All material copied or amended from any source (e.g. internet, books) must be referenced correctly according to the reference style you are using.  Your work will be submitted for electronic plagiarism checking.  Any attempt to bypass our plagiarism detection systems will be treated as a severe Assessment Offence.** |

**Coursework Submission Requirements**

* An electronic copy of your work for this coursework should be fully uploaded by midnight (local time) on the Deadline Date.
* The last version you upload will be the one that is marked.
* For this coursework you must submit a single Acrobat PDF document. In general, any text in the document must not be an image (ie must not be scanned) and would normally be generated from other documents (eg MS Office using "Save As .. PDF").
* Check whether you also need to upload a single ZIP file containing supporting evidence.
* There are limits on the file size.
* Make sure that any files you upload are virus-free and not protected by a password or corrupted otherwise they will be treated as null submissions.
* Comments on your work will be available from the Coursework page on the Intranet. The grade will be made available in the portal.
* You must NOT submit a paper copy of this coursework.

**Coursework Regulations**

1. Coursework submitted late without an Extenuating Circumstances claim will receive a ZERO grade.   
   If you have extenuating circumstances you may submit your coursework up to two weeks after the published deadline without penalty but this is subject to acceptance of your claim by the School Extenuating Circumstances Panel. If your claim is rejected then you will receive a zero grade for your work.
2. Coursework submitted more than two weeks late will be given feedback but a grade of non-submission will be awarded regardless of any extenuating circumstances. However, if your Extenuating Circumstances claim is accepted then the Extenuating Circumstances Panel will recommend to the Progression and Award Board that you be permitted to retake a different item of assessment at a future assessment point.

All coursework must be submitted as above.

**Detailed Specification**

**Please read the entire coursework specification before starting work.**

Many mainland and underground train stations have several public facilities available, such as toilets, wifi and disabled access. You are to create **a mobile app** that allows customers to record their assessment of the facilities in these public places. It is to be called **FacilitiesReviewApp** and the first part of the app is simply to allow the customer to identify and publish information about which stations have which facilities (your sample must include no less than 10 stations but no more than 20). The second part of the app is to be used to record details of facilities that the customer has visited and reviewed, and allow them publish their reviews instantly to their preferred public forum (social media, blog, website, etc). The FacilitiesReviewApp is to be implemented using Android.

**FacilitiesReviewApp**

**The application**

The features are listed in the order of importance with the most important first and you should implement them in this order.

1. **Enter details of Station being reviewed**

The minimum details that the user needs to able to enter are given below. Note that the reviewer must be able to enter all of these fields. "required field" means that the reviewer must enter something in this field otherwise they will get an error message.

* Station id (e.g. "zz123") – this is a required field
* Name of the Station being reviewed (e.g. “London Bridge”) – required field
* Type of Station (such as “Underground”, “National Rail” or “DLR”) – required field
* Type of Facilities available (e.g. “m/f toilets”, “wifi”, “lifts/ramp/disabled access to platform”) – required field
* Location of the establishment – this is an optional field (i.e. the system must allow the reviewer to enter this data if they wish but they can choose not to)
* One or more other fields of your own invention

If the reviewer doesn’t enter anything in one of the required fields or enters an invalid value in a field, the program should display an error message.

Once the details have been accepted by the program (e.g. no required fields were missing and all data is valid) it should display the details for confirmation and allow the reviewer to go back and change any details that they wish.

1. **Store and list the details**

Details entered by the reviewer should be stored on the device in an SQLite database.

The reviewer should be able to list **all** the details of **all** stations that have been entered into the app.

1. **Search and delete**

The user should be able to search for stations in the database by name. At its simplest this could mean entering a name and displaying the first station that matches, though ideally they should be able to enter the first few letters of the name and display all matching stations.

The user should be able to delete specific stations from the application.

1. **Add reviews**

Reviewers will be able to enter details of the review. They should be able to select a station and then add the following details:

* Date of the review - this is a required field
* Type of facility reviewed (e.g. “toilet”, “wifi”, “lift”) – required field
* Overall rating (make up your own rating scheme e.g. stars or letters) – required field
* Publish review to xxxxx [destination] – optional field
* Additional comments – optional field

The reviews should be stored in an SQLite database. It should be possible to display all reviews for a particular facility across several stations.

1. **Adding additional features to the Android App**

The above are the core requirements for the app. If you have implemented these and want to add some additional features then you may. **Any enhancements should be implemented in addition to NOT instead of the core requirements**. You can think of your own enhancements. Here are some possible examples:

* Allow photos taken by the camera to be added to the data stored
* Pick up the location automatically from the reviewer's location
* Make use of an external web service other than the one provided for the coursework.
  1. **Report**

Write a report consisting of **all** the following sections:

**Section 1.** A **concise table** containing a checklist of the features you have been able to implement. Please refer to the features list given above in the specification. For example, you might write:

|  |  |
| --- | --- |
| **Feature** | **Implementation** |
| a) | * Fully implemented |
| b) | * Fully implemented |
| c) | * Implemented but throws an exception if no matching pet is found |
| d) | * I've tried some coding but it's not working ☹ |
| e) | * No additional features implemented |

**Section 2.** A concise list of any bugs and/or weaknesses in your app (if you don't think there are any bugs or weaknesses then say so). Bugs that are declared in this list will lose you fewer marks than ones that you don't declare!

**Section 3.** A brief (less than half a page) description of any special strengths of your app that you think should be taken into account in awarding a mark. Please be very specific and realistic in this: vague statements such as "it is easy to use" or "it is well designed" will not gain you marks.

**Section 4.** Screen shots demonstrating each of the features that you have implemented. Give captions or annotations to explain which features are being demonstrated.

**Section 5.** An evaluation of your app. Write between 600 and 1000 words evaluating the app that you have produced**.** Try to be specific and justify any statements you make. Just saying things like ‘my app is well designed" without justifying the statement will not gain you any marks. Also explain how your app could be improved. Again, you need to try to be specific e.g. saying something like "It needs to be made more secure by adding security features" will not gain many marks. Your evaluation should include, but need not be limited to, the following aspects of your app:

* + 1. Human computer interaction (you will have a lecture on this)
    2. Security
    3. Maintainability

This sort of discussion will form an important part of your final project report so use this opportunity as a way of practicing your skills in writing an evaluation.

**Deliverables**

1. A zip file containing all the files required to run your Android app. Please try to structure your work so that it is easy for the person marking your work to compile and run your program(s) if they need to. Any compilation, installation or running instructions should be included in a “readme” file.

If you have **borrowed code** **or ideas** from anywhere other than the lecture notes and tutorial examples (e.g. from a book, somewhere on the web or a fellow student) then include a reference showing where the code or ideas came from and label your code carefully to show which bits are yours and which bits are borrowed. This will protect you against accusations of plagiarism. Be aware that **the marker will look for similarities between your code and that submitted by other students** so please do not share your code with any other students as this is considered to be plagiarism.

1. A demonstration (approximately 10 to 15 minutes) of what you have achieved. You will be asked to demonstrate your app running on a machine in the labs at Greenwich. You will be asked questions about your app and be expected to show an understanding of the code that you have written and the design decisions that you have made. If you are unable to answer questions about your code you may be investigated for plagiarism. If you develop your program at home it is **your responsibility** to make sure that it runs in the labs. If it will not run on the labs machine you will be allowed to demonstrate it on your laptop **BUT you may lose marks**. Please allow yourself plenty of time to get your app working on the lab machines before your demonstration time.

1. A report consisting of **all** the sections described in section 1.2 of the detailed specification. This is the final deliverable.

**Grading Criteria**

**For a very high First (85% and over)**

* A standard android app fully implementing at least features a), b), c) and d) with one or more good enhancements
* An outstanding report

**For a First** **(70% to 84%)** the following is required

* A standard android application fully implementing at least features a), b), c) and d)
* A very good report

**For a 2:1 mark in the range 60 to 69%** the following are required:

* A standard android application fully implementing at least features a), b), and a good attempt at c)
* A good report

**For a 2:2 mark in the range 50 to 59%** the following are required:

* A standard android application fully implementing at least feature a) and a good attempt at b)
* An acceptable report

**For a 3rd mark in the range 40 to 49%** the following are required:

* A standard android application with a good attempt at features a) and b)
* An acceptable report

For an answer that does not fit into any of these categories (e.g. features a to e are implemented but the report is only “acceptable”) the grade will be determined by taking the lower mark applicable (e.g. range 50 to 59% in this example) and making some upward adjustment for the other aspects of the answer. Please be aware to pass you must submit a working program **and** an acceptable report. For instance even if all features a) to e) are implemented but the report is very, very weak or has several sections missing then you may fail the coursework.

**NOTE: Failure to do your demonstration will normally result in you being awarded 0% coursework. Even if your implementation and report are excellent and would be awarded a mark of 80% but you don’t do a demonstration then you may score 0% for the coursework**

**Assessment Criteria**

Your program will be assessed on the following criteria.

* **Features implemented.**  The number of features that you have successfully implemented will have a big effect on your overall mark.
* **The quality of the program code you produce**. Credit will be given for inclusion of meaningful comments in the code, use of the sensible naming standards (e.g. for packages, classes, variables, and methods), code layout (e.g. indentation to make the structure of "if" statements and loops clear), avoidance of unnecessary duplicate code and proper use of exceptions. Java coding conventions (covering naming and indentation etc) can be found at <http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>
* **The user interface.**  Credit will be given for making your application as pleasant an experience as possible for the user. Examples of good practice are: allowing the user to choose options rather than their having to type in input, sensible default values, validation of input, and meaningful messages. Credit will be given for showing the use of a range of appropriate features from the Android GUI API.
* **Quality of the report.** Are all the required sections included and completed properly? Is the report clear and easy read? Does it have a logical structure? Is the evaluation (section 5) realistic and does it show that you have really thought about your app and what enhancements could be made to it?