| **COMP1786 (2021/22) Term 1** | **Mobile Application Design and Development** | **Contribution: 70% of course** |
| --- | --- | --- |
| **Course Leader: Keeran Jamil** | **Mobile Application Design and Development - CW** | **Deadline Date: 22nd November 2021** |
| This coursework should take an average student who is up-to-date with tutorial work approximately 40 hours  Feedback and grades are normally made available within 15 working days of the coursework deadline | | |
| **Learning Outcomes:** A. Have a critical awareness of and understand the technical challenges posed by current mobile devices and wireless communications; be able to critically evaluate and select appropriate solutions. B. Appreciate the need, and have the ability, to keep up with rapid changes and new developments; be able to identify, and predict changes in current trends in mobile communications technologies and systems. C. Select and critically evaluate suitable software tools and APIs for the development of a particular mobile application and understand their strengths, scope and limitations. D. Select and use appropriate application development tools to assist in the conception, design, writing and testing of various interactive programs for mobile devices. | | |

| 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 coursework from another student 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. Please see your student handbook for further details of what is / isn't plagiarism.   **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 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 must be fully uploaded on the Deadline Date .**

#### **For this coursework you must submit a single PDF document.  In general, any text in the document must not be an image (i.e. must not be scanned) and would normally be generated from other documents (e.g. MS Office using "Save As .. PDF"). An exception to this is hand written mathematical notation, but when scanning do ensure the file size is not excessive.**

#### **For this coursework you must also 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.**

#### **Your work will not be printed in colour. Please ensure that any pages with colour are acceptable when printed in Black and White.**

#### **You must NOT submit a paper copy of this coursework.**

#### **All courseworks must be submitted as above. Under no circumstances can they be accepted by academic staff**

The University website has details of the current Coursework Regulations, including details of penalties for late submission, procedures for Extenuating Circumstances, and penalties for Assessment Offences.  See <http://www2.gre.ac.uk/current-students/regs>

**Detailed Specification**

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

You are to create a mobile app to be used as **Rental Apartments Finder.** The app will allow the user to find, add and edit rental listings. The app will be called **RentalZ**. The features the app should support are given below

* Features **a)** to **e)** are to be implemented as a **hybrid app coded using PhoneGap or any other platform suitable.**
* Features **a) and b)** are to be implemented as a **native Android app coded in Java**.
* Feature **f)** can be implemented as **either or both additions to the** native android app or PhoneGap app.

**1.1 Description of the application**

**a) Design app screens - Basic details input screen**

Note that users must be able to enter all of the following fields:

* Property type (e.g. flat, house, bungalow) – required field
* Bedrooms (e.g. studio, one, two, etc.) - required field
* Date and time of adding the Property (when the property have been added) – required field
* Monthly rent price - required field
* Furniture types (e.g. Furnished, Unfurnished, Part Furnished) - optional field
* Notes – optional field
* Name of the reporter – required fields

**b) Implement forms validation**

**Required** field means that the user must enter something in this field otherwise they will get an error message. **Optional** field means that the user can enter something if they wish but they will not get an error message if they don't enter anything.

The app will check the input and if the user doesn’t enter anything in one of the required fields the app should display an error message to the user.

Once the details have been accepted by the app (e.g. no required fields were missing) it should display the details back to the user for confirmation and allow them to go back and change any details that they wish.

**c) Store, view and delete the basic details and check for duplicate events**

All the details entered by the user should be stored on the device in a Web database.

The user should be able to list **all** the details for all Properties entered in the app.

The user should be able to delete all the details from the database.

**d) Search**

The user should be able to **search** for a property. At its simplest this could mean entering or selecting a property type and displaying the details of all information about that type. Ideally the user should be able to enter more details and search for a property that matches.

**e) Add a note input screen**

The user may select one of the properties that they have entered and use this screen to enter information about something that is related to this property (e.g. condition, how close to schools, shops and public transportation). It is up to you how complex or simple you want to make this. At its most simple form the user can just enter a textual description of the property.

The app should store **all** details entered on the device in a Web database.

It should be possible for a user to select a property and display the report for that property.

**f)** Features **a) and b)** are to be implemented as a **native Android app coded in Java**

**g) Add additional features to either or both the Android or PhoneGap version of the app**

Features a) to e) 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. The idea is that these features stretch your skills so be prepared to do your own research and feel free to show off! You can think of your own enhancements. Here are some possible examples:

• Allow photos taken by the camera to be added to the property data stored

• Send social media messages (e.g. twitter) each time a report is added

• Save/Show the locations where the property located

**1.2 Report**

* **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 sometimes it gives an exception. I haven't been able to debug this |
| d) | * Implemented but the app can only do a simple search by property type |
| e) | * I have created the user interface for report entry but the data is not being stored ☹ |
| f) | * I have create a native prototype of feature a) but not feature b) |
| g) | * No additional features implemented |

* **Section 2.** A concise list of any bugs and/or weaknesses in your app(s) If you don't think there are any bugs or weaknesses then say so though this is quite unlikely. Bugs that are declared in this list will lose you fewer marks than ones that you don't declare!
  + Bugs: bug a. Your can provide screenshots or detailed explanations
  + Weaknesses: weakness 1, weakness 2…(Can be difficult to use, to learn, UI not good, validations, maintaining the code….)
* **Section 3.** A brief (less than half a page) description of any special strengths of your app(s) that you think should be considered when 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.
  + Strengths: easy to use, many validations, UI is good, extra functions, good coding standard.... You can supply the screenshots for evidence.
* **Section 4.** Screen shots demonstrating each of the features that you have implemented. Give captions or annotations to explain which features are being demonstrated.
  + Feature a)
    - brief summary of features
    - Screenshots for the feature
  + Feature b)
    - brief summary of features
    - Screenshots for the feature

….. …..

* **Section 5.** An evaluation of your app(s). Write between 300 and 500 words evaluating the app(s) that you have produced**.** 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 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
    4. Changes that would need to be made for the app(s) to be deployed for live use

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

1. Human computer interaction evaluation: evaluate your application based on some principles like(<https://www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces>)- use at least 4 of 5 principles of shneiderman
2. Security: have you implemented any security functions: having password, encrypted data. If not yet why?
3. Maintainability: how good coding structure, coding convention for latter maintenance
4. Changes that would need to be made for the app(s) to be deployed for live use: user interface, extra features….

**Deliverables**

1. A brief demonstration (approximately 10) of what you have achieved. You need to demonstrate your App running on a machine in the labs or on your own laptop.
   * You will be asked questions about your implementation and be expected to show an understanding of the functionality you have implemented and the design decisions that you have made.
   * The date, time and place of the demonstrations will be announced by the instructor.

**Student didn’t attend demonstration (maximum mark for work is 20%)**

1. A zip file containing all the files required to run your app(s). Please try to structure your work so that it is easy for the person marking your work to compile and run your app(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 another student) then include a reference showing where the code or ideas came from and comment your code very 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. Note that the upload of this zip file is a **mandatory interim submission**. It must be uploaded by the specified interim submission date **or you will lose marks and are likely to fail the coursework**.

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

**Grading Criteria**

Note that this coursework will not be marked anonymously. Student didn’t attend demonstration (maximum mark for work is 20%).

**1st Class (90% to 100%)**

* A **hybrid app coded using PhoneGap** application or any other platform suitablefully implementing at least features a), b), c), d) and e). Very few minor bugs or weaknesses. Exemplary quality code.
* A working naïve android application (a and b) and two or more good additional features (f) to one or both apps.
* Demonstration: able to show good knowledge of code implemented at the demonstration. Not only able to answer questions about **how** you did something but also **why** you chose to do it in a particular way and **compare** with possible alternative implementations.
* Report complete, accurate and easy to read. Section 5 within specified word count, logically structured and making some insightful points about all four of the issues specified.

**1st Class (80% to 89%)**

* A **hybrid app coded using PhoneGap** application or any other platform suitable fully implementing at least features a), b), c), d) and e). Very few minor bugs or weaknesses. Outstanding quality code.
* A working native Java android (a and b) and one or more good additional features (f) to one or both apps.
* Demonstration: able to show good knowledge of code implemented. Not only able to answer questions about **how** you did something but also **why** you chose to do it in a particular way.
* Report complete, accurate and easy to read. Section 5 within specified word count, logically structured and making some insightful points about all four of the issues specified.

**1st Class (70% to 79%)**

* A **hybrid app coded using PhoneGap** application or any other platform suitable fully implementing at least features a), b), c) and d). Very few minor bugs or weaknesses. Excellent quality code.
* A working native prototype (a and b).
* Demonstration: able to show good knowledge of code implemented. Not only able to answer questions about how you did something but also why you chose to do it in a particular way.
* Report complete, accurate and easy to read. Section 5 within specified word count, logically structured and making some insightful points about at least three of the four issues specified.

**Upper Second Class (60 to 69%)**

* A **hybrid app coded using PhoneGap** application or any other platform suitable fully implementing at least features a), b), c) and a good attempt at d). Few minor bugs or weaknesses. Very good quality code.
* A working native prototype (a and b).
* Demonstration: able to show good knowledge of code implemented.
* Report complete, accurate and easy to read. Section 5 within specified word count logically structured and making sensible points about at least two of the four issues specified.

**Lower Second Class (50 to 59%)**

* A **hybrid app coded using PhoneGap** application or any other platform suitable or any other platform suitable fully implementing at least features a) and b), c) with an attempt at d), possibly with some bugs and weaknesses. Good quality code.
* A working native prototype (a and b).
* Demonstration: able to show good knowledge of code implemented.
* Report complete including some attempt at section 5

**Third Class (40 to 49%)**

* A hybrid app coded using PhoneGap application or any other platform suitable with a good attempt at feature a),b) and some attempt at c).
* A working native prototype (a and b).
* Demonstration: able to show reasonable knowledge of the attempted features.
* Report mostly complete.

**Assessment Criteria**

Your app(s) will be assessed on the following criteria.

* **Features implemented.**  The number of features (listed as a to f in the specification above) that you have successfully implemented will have a big effect on your overall mark.
* **The quality of the application 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.
* **The user interface.** This is not a course about user interface design but 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.

Your report will be assessed on the following criteria.

* Are all the required sections included and completed properly?
* Does the report give an accurate reflection of what you have achieved?
* Is the report clear and easy read? Does it follow the structure specified?
* Is the evaluation (section 5) realistic and does it show that you have really thought about your app(s) and the specified issues and how they would need enhancing to be ready for live deployment. Do you show insight into the complexities of app development and the challenges of balancing the various constraints involved?