

SWE6140 GROUP WORK IN LARGE SCALE SOFTWARE DEVELOPMENT	Contribution: 60% of course
Coursework – GROUP A	PDF file required - ZIP file also required
Lecturer: Dr Lawrence Nderu	Due date: 18 th November 2016
This coursework should take an average student approximately 40 hours	
Learning Outcomes: A, B, C and D.	

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.

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").
- 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.

Detailed Specification

Please read the entire coursework specification before starting work.

You are to create a mobile app for use by people attending various types of public events (i.e., festivals, community events, etc.). The app will allow the user to enter, store and upload reports of the event as it is happening. The app will be called madDiscovery. The features the app should support are given below

- Features **a) to e)** are to be implemented as a **hybrid app coded using PhoneGap**
- 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 these fields.

- Event name (e.g. festivals name) – required field
- location of the event - optional field
- date of the event – required field
- start time of the event – optional field
- names of event organizer – required fields

b) Implement forms validation and check for duplicate events

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

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

The user should be able to list **all** the details for all events entered into the app.

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

Once the basic details of an event have been entered it should **not** be possible for the user to make another entry for the event. For example, if they have already entered "USIU Campus Art festival", 10/10/2016, "USIU", "ART" then it shouldn't be possible for them to make another entry with the same details.

d) Search

The user should be able to **search** for events. At its simplest this could mean entering or selecting an event type and displaying the details of all the events for that event. Ideally the user should be able to enter more details and search for events that match. For instance they could enter the organizer, location and date.

e) Add a report input screen

The user may select one of the events that they have entered and use this screen to enter information about something that happens during the event (e.g. prizes been given). It is up to you how complex or simple you want to make this. At its most simple the user can just enter a textual description of what happens.

The user must be able to enter multiple reports for a single event.

When the event is over the user should be able to indicate that event has ended. Once the event is over the user shouldn't be able to enter any more reports for it.

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

It should be possible for a user to select an event and display all the reports for that event.

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 event data stored
- Send social media messages (e.g. twitter) each time a report is added
- Show the locations of the events on a map

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)	<ul style="list-style-type: none"> Implemented but the app can only do a simple search by event name
e)	<ul style="list-style-type: none"> I have created the user interface for report entry but the data is not being stored ☹️
e)	<ul style="list-style-type: none"> I have create a PhoneGap prototype of feature a) but not feature b)
f)	<ul style="list-style-type: none"> 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!
- **Section 3.** A brief (less than half a page) description of any special strengths of your app(s) 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(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:
 - Human computer interaction (you will have a lecture on this)
 - Security
 - Maintainability
 - Changes that would need to be made in order 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.

Deliverables

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 **I 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.

2. A demonstration (approximately 15 minutes) of what you have achieved. 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. For instance I may ask you to take me through the code that implements a particular feature. If you are unable to answer questions about your code you may be investigated for plagiarism. If you develop your app(s) at home (in a desktop at home for example) it is **your responsibility** to make sure that it runs in the labs at Karen.
3. 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.

1st Class (90% to 100%)

- A **hybrid app coded using PhoneGap** application fully implementing at least features a), b), c), d) and e). Very few minor bugs or weaknesses. Good 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 fully implementing at least features a), b), c), d) and e). Very few minor bugs or weaknesses. Good 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 fully implementing at least features a), b), c) and d). Very few minor bugs or weaknesses. Good 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 fully implementing at least features a), b), c) and a good attempt at d). Few minor bugs or weaknesses.
- 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 fully implementing at least features a) and b), c) with an attempt at d), possibly with some bugs and weaknesses
- 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 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.

NOTE: Failure to do your demonstration will normally result in you being awarded 0% coursework and a failure in this course. 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 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 to 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?