COMP1424 (2022/23)	**Mobile Application Development**	**Contribution: 100% of course**
Course Leader: **Dr Markus A. Wolf**	**Practical Coursework**	**Deadline Date:** **Tuesday 06/12/2022**

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 of Tuesday 06/12/2022, 23:30 using the link on the coursework Moodle page for COMP1424.**
- **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 (see the relevant course Moodle page).**
- 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 https://www.gre.ac.uk/policies/undergraduate-and-postgraduate-taught

Detailed Specification

Please read the entire coursework specification before starting work.

This assignment consists of three parts:

- **Part A (implementation) will be completed in a group**
- **Part B (group report) will be completed in a group**
- **Part C (individual report) must be completed individually**

M-Hike - Hiker App

Many people enjoy going for hikes in nature in their free time. A community of hikers has approached you to create a mobile app which would allow them to record details of hikes they are planning and record observations during the hike.

You are to create a mobile app for use by hikers that they can use to record details of their planned hiked and then upload them to a server where they can later be shared with others. The app is to be called **M-Hike** and will allow hikers to record planned hikes and observations during a hike and to carry out searches. The features the app should support are given below:

- Features **a)** to **e)** are to be implemented as a **native Android app coded in Java**.
- Features **f)** and **g)** are to be implemented as a **hybrid app coded using Cordova or other hybrid technology**.
- Feature **g)** can be implemented as **either or both** additions to the native Android app or Cordova app.

Your final app is the culmination of all your hard work on this course, which should become a strong addition to your programming portfolio. You should produce an app that is well-designed, robust and useful. The GUI design should be clean, simple to navigate, and operate smoothly without sluggishness or crashes. The app should not require instructions or a manual to use.

Part A – Implementation (80%)

Organise yourselves into groups consisting of 3 or 4 students. Exceptionally, you can work individually, but please approach the tutor if you are thinking of doing this. If you cannot find a group to join, your tutors will allocate you to a group. **All groups are final on the 25th of October 2022** and no changes will be made to groups after this date.

Description of the application

a) Enter details of hikes (10%):

Note that users must be able to enter all of the following fields. ***Required*** fields mean that the user must enter something in this field; otherwise they will be prompted with an error message. ***Optional*** fields mean that the user can enter something if they wish, but they will not get an error message if nothing is entered.

The user should be able to enter:

- Name of hike (e.g. “Snowdon”, “Trosley Country Park”, etc.) – *Required field*
- Location - *Required field*
- Date of the hike - *Required field*
- Parking available (i.e. “Yes” or “No”) - *Required field*
- Length the hike - *Required field*
- Level of difficulty - *Required field*
- Description – *Optional field*
- Two or more other fields of your own invention – be creative!

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.

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.

b) Store, view and delete details or reset the database (10%)

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

The user should be able to **list all** the **details of all hikes** that have been entered into the app, edit or delete individual hikes, and delete all the details from the database.

c) Add observations to a hike (10%)

Hikers will be able to enter observations during a hike. Observations could include sightings of animals, types of vegetation encountered during the hike, weather conditions, conditions of the trails, etc. They should be able to select a hike and then add the following details:

- Observation - *Required field*
- Time of the observation - *Required field (ideally this should default to the current date and time)*
- Additional comments - *Optional field*

The user must be able to enter multiple observations for a single hike.

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

It should be possible for a user to select a hike and display all observations.

d) Search (10%)

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

Advanced search options will allow searching for all hikes with the following criteria: name. location, length and/or date. It should be possible for a user to select an item from the resulting search list and to display its full details.

e) Upload details to a cloud-based web service (10%)

The user should be able to upload all the details of hikes stored on the mobile device to a cloud-based web service. It should be possible for all hikes to be uploaded at once. YOU DO NOT NEED TO WRITE the web application as it will be made available to you. See the Moodle site for this course in for the URL of the web service and further information about how to use it (https://stuiis.cms.gre.ac.uk/COMP1424CoreWS/comp1424cw).

The web service will accept the data and reply with a message which your app should display. Ideally your app should format the message so that is easy for the user to read.

PLEASE NOTE: This access to this URL is restricted to the University's network, so it will only work when you are in the University labs.

The format in which the data must be uploaded

The data you upload to the web service should be sent in an HTTP POST message. The HTTP POST message will contain one field called **jsonpayload**. The value of **jsonpayload** is a string in JSON format.

You have some flexibility over how you design the JSON however the following are essential:

- The JSON must be valid. See http://www.w3schools.com/js/js_json_syntax.asp for details on JSON syntax.
- You must include a field with the name "userId" which has a value of your university user id e.g. "userId":"wm123"
- The hike details must be represented as an array with the name "detailList"
- Each element in the array "detailList" represents a hike about which details are being uploaded.
- Each hike in "detailList" must at a minimum have a field called "name" whose value is the hike name or the brief description of the hike
- Below is an incomplete example of what the JSON might look like
  {"userId":"wm123",
  "detailList":[
  {"name":"Snowdon", *other fields*},
  {"name":"Trosley Country Park", *other fields*}]}

The format of the response

The web service will respond with an HTTP response containing JSON which represents a reply to your upload. The JSON reply will contain the following fields:

field name	value
uploadResponseCode	A code indicating whether or not your upload was successful. Values are either SUCCESS or FAILURE. If the code is FAILURE the message field will contain more information about what went wrong.
userid	The same userId that was included in the upload message
number	The number of records uploaded
names	The names of the records uploaded
message	A message giving more information about what happened. Particularly important if the uploadResponseCode is not SUCCESS

An example response is given below. Note that the order of the fields may vary

```
{"uploadResponseCode":"SUCCESS",
"userid":"wm123",
"number":2,
"names":"Snowdon, Trosley Country Park",
"message":"successful upload – all done!"}
```

Your app should display the data sent in the response. You could just display the raw JSON message, but it would be better to format the output to make it more readable for the user.

The JSON string which is prepared and sent to the web service should also be printed to the LogCat, so one can see that the correct JSON is prepared and sent.

f) Create a cross-platform prototype of the app using Cordova or other hybrid technology (10%)

Implement as much as you can of features a) using Cordova or other hybrid technology of your choice.

g) Implement persistence using Cordova or other hybrid technology (10%)

Implement as much as you can of features b) using Cordova or other hybrid technology of your choice.

h) Add additional features to either or both the Android and hybrid version of the app (10%)

Features a) to g) 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 data stored
- Pick up the location automatically from the user's location
- Show the location of a hike on a map
- Make use of an external web service other than the one provided for the coursework.
- Anything you can think of – again be creative!

NOTE: Please note that each individual student's grade within the group will be adjusted based on the contribution made to the implementation and group report

Part B – Group Report (7%)

Completed by the group.

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

- **Section 1. (2%)** 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)	I have created the user interface for data entry but the data is not being stored ☹
d)	Implemented but the app throws an exception if no matching result is found.
e)	Fully implemented
f)	I have create a Cordova prototype of feature a)
g)	Features b) were not implemented in the Cordova prototype
h)	No additional features implemented

- **Section 2. (3%)** Screen shots demonstrating each of the features that you have implemented. Give captions or annotations to explain which features are being demonstrated.

- **Section 3. (2%)** Code listing of any code files you have written. You do not need to include generated code. Please clearly label the code, so it indicates the source file and programming language.

  Please note that you must include the actual code, not just file names.

- **Section 4. (0%)** Include a completed CW Contributions Form (found at the end of this specification).

NOTE: Please note that each individual student's grade within the group will be adjusted based on the contribution made to the implementation and group report

Part C – Individual Report (13%)

The individual report is to be completed **individually.**

Write an individual report consisting of **all** the following sections:

- **Section 1. (3%)** Write a reflection on how it was to work in a group and of your role within the team. Discuss lessons learnt, what you think went well and what you think could have been improved and how.

- **Section 2. (8%)** An evaluation of your app(s). Write between 700 to 1000 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:

  i. Human computer interaction (you will have a lecture about this)
  ii. Security (you will need to research this yourself)
  iii. Ability of the app to run on a range of screen sizes and how this could be improved (we will touch on this in the course but you will need to do additional research)
  iv. 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 MSc project report so use this opportunity as a way of practicing your skills in writing an evaluation.

- **English Proficiency (2%)** – marks will be allocated to the level of language used in the individual report, including correct spelling and use of grammar

Demonstration (0% but see below)

The demonstration is carried out by the group.

You are required to prepare a brief video showing your implementation (one video per group). The duration should be approximately 10 minutes. There will be a dedicated Panopto coursework submission area – more details on this will be available on Moodle.

Your group must also attend a brief Q&A session with your tutor where you will be asked questions about your implementation and the code.

Failure to demonstrate will result in a failed assessment. You will be allocated a time slot closer to the submission deadline.

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 (eg. 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 MANDATORY**. It must be **uploaded by the deadline** along with your report (deliverable 3) **or you will lose marks and are likely to fail the coursework**.

   Please include a file called "Group List.txt", which lists all the members of the group, including the name and id of each member.

2. A Video Demonstration of your app and what you have achieved, showing the app in operation and displaying all functionalities you have implemented. This should be approximately 10 minutes in duration and must be uploaded via the module's Panopto assessment submission link. There should be only one video submission per group.

   The video you upload on Panopto should follow the follow naming convention – "Surname1,Surname2, etc."

   You should include the surnames of all group members in the video name, so we can easily identify the group.

3. A brief Q&A session where your group will be asked questions about your app(s) and be expected to show an understanding of the code that you have written and the design decisions that you have made. For instance, the tutor may ask you to talk them through the code that implements a particular feature. If you are unable to answer questions about your code you may be investigated for plagiarism. The demonstrations will **take place on 13/12/22**, and the exact time and details of your Q&A meeting will be announced on the Moodle site for the course.

4. A PDF combining the individual and group report consisting of **all** the sections described in the specification. It must be **uploaded by 06/12/22.**

Deliverable	**Date**	**Type**	**Group/Individual**
Report PDF (combining the individual and group report)	06/12/2022	Submission on Moodle	Individual
Code	06/12/2022	Submission on Moodle	Group
Code Q & A (Live Demonstrations)	13/12/2022	MS Team	Group
Video Demonstration	13/12/2022	Submission on Panopto	Group

Formative Assessment

Groups are encouraged to show the progress made with the implementation in the labs on a weekly basis to get feedback on how the work progresses and, on the design, and implementation decisions taken at every stage of the assignment.

There are also weekly lab exercises, which train you in the knowledge required to complete the assessment. Completed exercises do not need to be uploaded but shown to the tutor for feedback.

Rubric

A rubric for the assignment is available as a separate document on the module's Moodle page.

Grading Criteria

This coursework will not be marked anonymously.

Each of the 8 features are worth 10% of the overall assessment. This doesn't mean that you will automatically obtain the full 10% for a feature for implementing it, but it will be graded based on the assessment criteria below. Section 3 of the report is also worth 10% and each remaining section of the report is graded out of a maximum of 2.5%.

For a very high distinction (90% to 100%)

- A native Android application fully implementing all features. No bugs and negligible weaknesses. Exemplary quality code.
- Two or more creative additional features (h) to 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 and **compare** with possible alternative implementations.
- Reports complete, accurate and easy to read. The evaluation section within specified word count, logically structured and making some insightful points about **all four** of the issues specified.

For a high distinction (80% to 89%)

- A native Android application fully implementing all features. No bugs and very minor weaknesses. Outstanding quality code.
- A working hybrid prototype and one or more creative additional features (h) to 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.
- Reports complete, accurate and easy to read. The evaluation section within specified word count, logically structured and making some insightful points about **all four** of the issues specified.

For a distinction (70% to 79%) the following is required

- A native Android application fully implementing at least six features. Very few minor bugs or weaknesses. Excellent quality code.
- A working hybrid prototype and one or more very good additional features (g) 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.
- Reports complete, accurate and easy to read. The evaluation section within specified word count, logically structured and making some insightful points about at least three of the four issues specified.

For a merit mark in the range 60 to 69% the following are required:

- A native Android application fully implementing at least features 5. Few minor bugs or some weaknesses. Very good quality code.
- A working hybrid prototype.
- Demonstration: able to show good knowledge of code implemented.
- Reports complete, accurate and easy to read. The evaluation section within specified word count logically structured and making sensible points about at least two of the four issues specified.

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

- At least six features have been completed across the native Android application and the hybrid prototype, possibly with some bugs and weaknesses. Good quality code.
- Demonstration: able to show good knowledge of code implemented.
- Reports complete including some attempt at the evaluation section.

For a marginal fail mark in the range 40 to 49% the following are required:

- At least five features have been completed across the native Android application and the hybrid prototype, possibly with some bugs and weaknesses. Good quality code.
- Demonstration: able to show reasonable knowledge of the attempted features.
- Reports mostly complete.

For a fail mark in the range below 40%:

- A native or hybrid android application attempting four of less feature but buggy.
- Demonstration: unable to show reasonable knowledge or understanding of the attempted features.
- Reports missing or mostly incomplete.

For a coursework that does not fit into any of these categories (e.g. features a) to d) are implemented but section 3 of the report is less than 700 words and weak) the grade will be determined by taking the lower mark applicable and making some upward adjustment for the other aspects of the coursework. Please be aware to pass you must submit a working app **and** an acceptable report.

NOTE: Failure to do your Demonstration will normally result in you being awarded 0% for the 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 app(s) will be assessed on the following criteria.

- **Features implemented.** The number of features (listed as a) to h) 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 (eg. 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, for the Android app use of appropriate Java language features and Android APIs. Java coding conventions (covering naming and indentation etc) can be found at http://www.oracle.com/technetwork/java/codeconvtoc-136057.html – (please note that this Oracle page is no longer maintained as the java conventions are common standard, but the page continues to be used as a sources of reference).

- **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 reports 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 realistic and does it show that you have really thought about your app(s) and the specified issues as well as how they may be enhanced 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?

NOTE: Please note that each individual student's grade within the group will be adjusted based on the contribution made to the implementation and group report

Summative Feedback

Feedback on the final submission will be provided in written format and made available on Moodle within 15 working days of submission.

1. Coursework Contribution Form (completed as a group)

In percentage, please indicate the work contribution of each member. This should be agreed by all group members**. The total of all members work must add up to 100%**

Team member name	**Student ID**	**individual overall work contribution (%)**	**Note**
Student:			
Student:			
Student:			
Student:			
		Total 100%	