Assignment 02: Building a GPS waypointing application

Barry Denby

October 7, 2025

1 Assignment Information

Course:	MSCC
Stage / Year:	1
Module:	Mobile Development
Semester:	1
Assignment:	2 of 2
Date of Issue:	2025-10-07
AI use:	**No AI use permitted for this assignment**
Minimum Gradle Version:	**8.10**
Assignment Deadline:	2025-12-21 @ 23:55 (End of week 12)
Assignment Submission:	Upload to Moodle
Assignment Weighting:	30% of Module

2 Introduction

NOTE: read the whole assignment brief first before implementing it contains very important information

In this assignment you will be tasked with building a GPS waypointing application using both the GPS sensor and a custom canvas.

The idea of an application like this is for hiking/orienteering where a laid out path that can be followed similar to Google Maps or other navigation apps does not exist. What a user will do is start GPS tracking at their starting point and will go hiking. Then at various points along the trail the user will stop and through a button click make a waypoint (i.e. a record of their current location).

If the user gets lost they should be able to select any of the waypoints and walk back to each waypoint in turn. If they follow all the waypoints back in reverse order they should get back to their original location.

1. How fully features, complete, and robust your code is. Along with how well your UI is thought out (80%)
2. How well documented your code is (20%)

NOTE: This is an individual assignment. It is not a group assignment. You can discuss ideas/algorithms but you cannot share code/documentation.

3 Submission and Penalties

You are required to submit two separate components to the Moodle

- An archive containing your complete Kotlin Android Studio Project. XML based layouts are not permitted. The accepted archive formats are: zip, rar, 7z, tar.gz, tar.bz2, tar.xz. The use of any other archive format will incur a 10% penalty before grading. Your archive should contain your .git directory which contains your git repository.
- A PDF containing documentation of your code. If you do not provide documentation you will automatcially lose 20% of your mark. Copying and pasting code (directly or by screenshot) into a PDF does not count as documentation.
- If your submission uses a version of gradle earlier than that listed above it will not be corrected due to incompatibility issues.

There are also a few penalties you should be aware of

- Code without a .git repository will be deducted 10%
- Remote repositories are not permitted. You are not permitted to use Github, Gitlab, BitBucket or any other remote repository. Your code will not be corrected if you use one of these.
- Code without documentation will be deducted 20%
- A git repository with less than 7 commits will be deducted 5%
- Code that fails to compile will be a failing assignment. At this stage you have zero excuse to produce non compiling code. I should be able to open your project and be able to compile and run without having to fix syntax errors.
- The use of libraries outside the SDK will incur a 20% penalty before grading. You have all you need in the standard SDK. I shouldn't have to figure out how to install and use an external library to get your app to work
- The standard late penalties will also apply

Very Important: Take note of the groups listed below. These are meant to be completed in order. Groups must be completed in full before the next group will be evaluated. Completed will mean that all tasks in the groups are visible and testable. If a single one is not visible and testable further groups will not be considered. e.g. if there are four tasks in Group 1 and task 3 is skipped or not visible or testable then Groups 2, 3, 4, and 5 will be ignored. Documentation will be treated separately irrespective of how many Groups you have completed.

You should also be aware that I will remove marks for the presence of bugs anywhere in the code and this will incur a deduction of between 1% and 10% depending on the severity. If you have enough of these bugs it is entirely possible that you may not score very many marks overall. I want robust bug free code that also validates all user input to make sure it is sensible in nature. Please be aware of the major bugs section. If any of these bugs are present in your application you will lose 10%.

4 Plagiarism

Be aware that we take plagiarism very seriously here. Plagiarism is where you take someone else's work and submit it as if it was your own work. There are many different ways plagiarism can happen. I will list a few here (this is not exhaustive):

- Finding something similar online (full implementation or tutorial) that does the same job and submit that.
- Finding something similar online (full implementation or tutorial) and transcribing (i.e. copying it out by hand)
- Working together on an individual assignment and sharing code together such that all implementation look the same.
- Getting a copy of someone else's code and submitting/transcribing that
- Paying someone to do your assignment. **NOTE: if you are caught participating in either side of such a transaction upto 5 years after you graduate you can be stripped of your degree.**
- Logging into someone elses Moodle account, downloading their assignment and uploading it to your own Moodle account.

I've had to deal with many cases of plagiarism over the last ten years so I can spot it and diagnose it easily, so don't do it. To prevent plagiarism include but not limited to the following:

- Do all your code by yourself
- Don't share your code with anyone, particularly if anyone looks for a copy of your code for reference.
- Don't post your code publicly online. Remember the use of GitHub, Gitlab, BitBucket etc is prohibited.
- If you need to find information online only query about very specific problems you have don't look for a full assignment or howto.
- Change the default password on your Moodle account. The default password can be determined if someone is connected to you through social media or they get one or two details from you.
- If you need to refer to anything online your only permitted source to reference is StackOverflow.
- Please note that AI tools such as ChatGPT will count as plagiarism. There use is strictly prohibited

Be aware that if you submit your assignment you accept that you understand what plagiarism is and that your assignment is not plagiarised in any way.

Also be aware that if you are caught for plagiarism you will not get another opportunity or a second chance to resubmit the assignment.

If you see the words **"pending review"** in your assignment feedback it is 99% likely that you will be called to a plagiarism meeting.

5 Coding Tasks (80%)

- Group 1 tasks (16%)

  1. Create two Kotlin files in your project. MainActivity.kt should contain the activity that your application starts with. SharedComposables.kt should contain all of your functions that are tagged with the “@Composable” annotation. If you need additional Kotlin files you have free choice of name for them, except these two.
  2. Create the custom composable for the waypoint display and force it to be square in size
  3. The custom composable should have the four major compass directions (north, south, east, and west) with north highlighted a different colour to the other compass directions
  4. A listener should be added to the rotation vector. The rotation vector should be used to angle the compass to point in the direction the user is facing. e.g. if the user is facing east then east on the compass should point to the top of the canvas

- Group 2 tasks (32%)

  5. Add a button to start and stop tracking
  6. When tracking is started a listener should be added to GPS that updates the user’s location every 5 seconds. The listener should be removed when tracking is stopped.
  7. While tracking a button should be added to the UI. When the user clicks this button it should save the current GPS position as a new waypoint
  8. The current list of waypoints should be stored in a list in a file created by the application. This should be immediately updated when a user adds a new waypoint.

- Group 3 tasks (48%)

  9. Add in a UI control to enable the user to clear the list of waypoints. This must trigger a dialog asking the user to confirm before deleting.
  10. Add in the UI controls for the user to select a waypoint.
  11. Once a waypoint has been selected its compass direction relative to the user should be set and displayed on the custom view. The waypoint direction should change if the user is moving relative to that waypoint.
  12. Add in a distance to waypoint display stating how many metres the user is from that waypoint. It should update as the user moves

- Group 4 tasks (64%)

  13. On the custom canvas add in coloured circles showing all waypoints within 500 metres of the user. The center of the compass is 0 metres while the edge of the compass is 500 metres away.

14. The currently selected waypoint should be highlighted on the custom canvas
15. The custom composable should enable touch input enabling the user to select waypoints by touching the coloured circle of that waypoint.
16. when a waypoint is touched the application should update to navigate to that waypoint.

- Group 5 tasks (80%)

17. Enable mulittouch input to adjust the scale of the compass using the pinch to zoom gesutre. it should be possible to adjust the scale from a minimum of 500 metres to a maximum of 2 kilometres.
18. The previous waypoint should be automatically selected when a user gets within 10 metres of the current waypoint
19. UI design: well thoughtout UI that is easy and intuitive to use.

6 Documentation Brackets (20%)

NOTE: Documentation should be around 1,500 words in length total

1. (20%): Document every method in your code from a high level perspective. i.e. give an overview of what the method does. Do not copy and paste code you will be penalised for this.