Functionality

Primary Goals:	Status
- Build an app that can display the user's current location on a map	Implemented
- Update the map in real time as the user moves.	Implemented
- In response to a "tracking on/off" UI switch, record or do not record the user's movements.	Implemented
- Display a path over the map as the user moves, that represents the user's movements that are currently being recorded.	Implemented
- If a journey is defined as a set of recorded locations between a tracking on and a tracking off switch, retain the user's journeys.	Implemented
- Allow the user to see all their journeys in a list.	Implemented
- Allow the user to see the start and end times of their journeys when they select them from the list.	Implemented
- The app should record the user's location in the background if the user selects "tracking on" and closes the UI.	Implemented
- The app should not use the battery if the user selects "tracking off" and backgrounds the app.	Implemented
- If the app is resumed from the background during tracking, it should correctly display a path representing the journey that is currently being recorded.	Implemented
Bonus Goals:	
- Allow the user to see each journey's path plotted on a map, when selected from the list.	Implemented
- Show any other interesting data you can think of relating to a journey, when selected from the list.	Implemented
- Secure the data that's stored on the device.	Partially Implemented
Optional	
- Retain the user's data if the app is deleted and re-installed.	Partially Implemented
- Detect the user's motion, and automatically determine when to turn tracking "on" or "off", in a way that conserves battery power.	Not Implemented

- Securing the data on the device is pretty simple with Realm. A key would be generated, then stored on the device secure keychain. This can then be used to decrypt the Realm DB file when needed. Realm encryption is bank level secure so would be suitable for sensitive data (https://academy.realm.io/posts/tim-oliver-realm-cocoa-tutorial-on-encryption-with-realm/)

- Retaining user data when app is quit is already implemented, to be able to restore on re-install I think I would need to use some form of cloud storage. Maybe Firebase or AWS?
- Detecting when a user is moving would use CoreMotion to detect that the device is moving, then detect the type of movement. I can then start the location tracking if needed.

3rd Party Libraries

I have used 3 3rd party libraries in this project, only as alternatives or extensions of current functionality in swift. I've included my reasoning for using them and how I could have implemented the same functionality using the 'stock' Swift features.

- SwifterSwift implements a load of very useful extensions to most swift data types which makes certain tasks/functions easier to call and more concise. I could have added these extensions manually if needed, or not used them at all.
- ISHPullUp implements a pull up ViewController similar to Apple Maps, which I thought felt nice in the project. I could have used the standard navigation controller to present the ViewControllers in a more traditional way.
- Realm is an alternative to CoreData that I find is a lot easier to work with in terms of creating models, adding objects, accessing them and just general debugging. It performs the same functionality as CoreData, and usually better performance. The model structure is very simple for this application, so this could have been implemented in CoreData in a very similar way.

All 3 are covered under either MIT or Apache 2.0 licence so are free to use without any restrictions.

Future Functionality

- Delete and rearrange journeys in the view journeys
- Tagging/adding comments to journeys to be able to search
- Geolocate the start and end locations to provide more context to user
- Broadcast location to a different device using the same app