The solution to this exercise *must* be submitted as a link (and credentials if needed) to a git repository, using an online service of your choice (for example, github). The source should be buildable using XCode or Android Studio with minimal documentation.

The exercise is split into the following sections:

* **Primary Goals**
* **Mandatory Concerns**
* **Bonus Goals**
* **Optional Goals**

**Primary Goals:**

* These goals are listed in the order in which they should be achieved.
* Only when one goal is achieved, should you move on to the next one.
* If any goals are not achieved when the solution is submitted, please include a written suggestion explaining how those goals would be achieved given more time.

**Mandatory Concerns:**

* These are concerns that MUST be addressed for any goal that is achieved.
* Failure to keep these concerns in mind when solving for a particular goal is potentially more costly than not attempting a goal at all.
* Failure to achieve all goals but success at addressing mandatory concerns for all goals achieved is considered better than the alternative.

**Bonus Goals:**

* These are goals that can be attempted when all primary goals are clear.
* Unless primary goals are attempted, bonus goals will not factor in determining success. Otherwise, they will act as a bonus.

**Optional Goals:**

* These are NOT expected to be attempted by most candidates. However, you should definitely think about them and be prepared to discuss potential solutions when called in.

**Exercise:**

**Mandatory concerns**

* All functionality should be implemented in a way that strives to separate the UI from background code. Adherence to the MVC pattern is the preferred method of achieving this
* The UI must not be laggy, buggy, unstable, or poor. Neatness and clarity are absolute requirements. Actual Artwork or graphics are NOT considered important.
* Keep best practices in mind.
* Document extensively.

**Primary Goals:**

* Build an app that can display the user’s current location on a map
* Update the map in real time as the user moves.
* In response to a “tracking on/off” UI switch, record or do not record the user’s movements.
* Display a path over the map as the user moves.
* 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.
* Allow the user to see all their journeys in a list.
* Allow the user to see the start and end times of their journeys when they select them from the list.
* The app should record the user’s location in the background if the user selects “tracking on” and backgrounds the app.
* The app should not use the battery if the user selects “tracking off” and backgrounds the app.
* If the app is resumed from the background during tracking, it should correctly display a path representing the journey that is currently being recorded.

**Bonus Goals:**

* Allow the user to see each journey’s path plotted on a map, when selected from the list.
* Show any other interesting data you can think of relating to a journey, when selected from the list.
* Secure the data that’s stored on the device.

**Optional**

* Retain the user’s data if the app is deleted and re-installed.
* Detect the user’s motion, and automatically determine when to turn tracking “on” or “off”, in a way that conserves battery power.

**Time Allocated:**

* The maximum time allowed will be included on the accompanying email.