**ENGINEERING JOURNAL TEMPLATE**

# Date

* every page MUST have a date
* should always be a the top corner.

# Tasks

* list of tasks for the day
* this serves as a way for you to collect your thought and set goals for the day/meeting

# Reflection

* should reflect what you have done for today
* may include more if you happen to achieve a few more goals than listed in the “Tasks” section
* should highlight interesting findings, especially those unexpected

# Issues:

*Hardware:*

1. outstanding issues on hardware development
2. numbered entries.
3. say “none” if absolutely no issue.
4. content MUST be informative, and contain substance. Information like “chassis does not work” does not provide any useful information. Instead, information content will be something like: “Chassis lacks of bracing… too much friction on the axles so the chassis is sacking in the middle…”

*Software:*

* outstanding issues on software development
* numbered entries.
* again, content MUST be informative… do not say things like “navigation is not working”.. that is not useful!

# Solutions

*Hardware:*

* if this is a solution which something happened today, it can be recorded in the reflection.
* If this is to resolve an issue happened before today, you should refer it by the date followed by the number, e.g.

e.g. issue 1 recorded on June 1st.

solution : 6/1-1) solution is to …. etc.

*Software:*

* Same as the hardware section.

**ENGINEERING JOURNAL TEMPLATE**

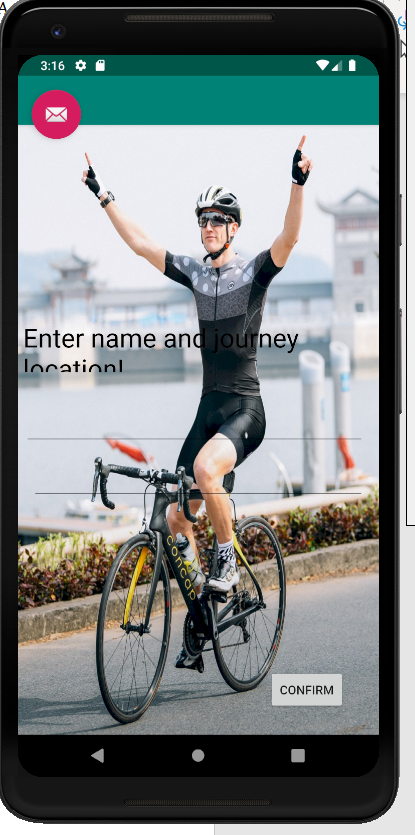
# Date

14th – 18th October

# Tasks

* Download Android Studio and begin examples to become familiar with the software.
* Ask technician for update on parts

# Reflection

* Found there is a lot that can be done on Android Studio
* Need to set some time if I want to build the user-friendly app that I have envisaged
* 

# Issues:

*Hardware:*

1. Still waiting on parts to arrive
2. Once arrived, I plan to set up raspberry pi for first time, then set up the hall effect sensor to detect a magnet.
3. Once I am happy this code is smooth, my aim is to get it reading an accurate speed.

*Software:*

1. Spending time on Android Studio while my parts are arriving
2. Researching information on the Rpi such as pinouts, linux commands etc.
   1. Some references <https://circuitdigest.com/microcontroller-projects/interfacing-hall-sensor-with-raspberry-pi>
   2. <https://pinout.xyz/pinout/>

# Solutions

*Hardware:*

* Tech says hardware will be here next week.

*Software:*

* Currently having problems with button location on Android Studio(15th Oct 2019). My plan is to check out tutorials on constraints that will stabilize my buttons to my desired location
* Also adding a png picture that did not stretch out when uploaded into my app. After some time I realized adding a picture of the correct size was mandatory followed by

**android:background="@drawable/cycling"**

**ENGINEERING JOURNAL TEMPLATE**

# Date

**21th – 25th October**

# Tasks

* Parts have arrived!
* Boot up raspberry pi and begin basic tests on gpio

# Reflection

* Configuring Rpi for first time and got basic gpio commands was not a problem.
* Testing hall effect sensor did not work first time around, I plan to re-review the pin outs, I am nearly sure I have configured them incorrectly

# Issues:

*Hardware:*

1. Had issue with Raspberry Pi, no hdmi input was recognized on monitor, troubleshooting via google I decided to delete the pre-installed NOOBS(New Out Of the Box Software) and download Raspbian OS. After imaging this on the SD card it worked.
2. Did the basic setup commands that are suggested when using a Rpi for the first time.
   1. sudo apt-get update
   2. sudo apt-get upgrade
3. Wasted some time wiring my hall effect sensor incorrectly but have got the correct pinout and tested some code successfully

*Software:*

* After completing the blink test mentioned below I wired up my hall effect sensor.
* Ran basic code to detect when a magnet gets close to it.

# Solutions

*Software:*

1. Activated the GPIO pins on the Rpi <https://www.raspberrypi.org/forums/viewtopic.php?t=223598>
2. Began basic blink test with a LED <https://raspberrypihq.com/making-a-led-blink-using-the-raspberry-pi-and-python/>
3. Ran basic code to detect when a magnet gets close to it on python and calling it in the command window.