# Tutorial 1

## About this project

In this course, you will work in a group to create a phone app using the MIT App Inventor. The development of this app will simulate the activities you can expect in a real development environment. You will need to propose a design, gain approval for the project based on the design, implement your project, and report on the performance of your application.

## Setup your group

***//TODO: Any notes around how groups will be created/assigned/registered with the tutor. Eg: find 5 other people, and send email of student IDs and names, along with group name to tutor -don't worry if you cannot find 5 other people, they will be found for you ;)***

## Get to know your group

***//TODO: List of questions the group members should ask to introduce themselves and why they are here***

## Know your tools

Before anything else, it is important to become well acquainted with your tools, so that you can design the application.

### MIT App Inventor

When developing an application with the [MIT App Inventor platform](http://appinventor.mit.edu/explore/about-us.html), a google login account is required.

#### Develop

##### Build Environment

You will build your application in a browser [here](ai2.appinventor.mit.edu)

##### Test Environment

For those of you with an android device - use the [companion app](http://appinventor.mit.edu/explore/ai2/setup-device-wifi.html)

For anyone else - use the [emulator](http://appinventor.mit.edu/explore/ai2/setup-emulator.html)

***//TODO: Test that the emulator can run on a uni lab computer***

#### Deploy

##### Build

The app can be built/compiled for distribution and sharing. For sharing the file, it is easiest to use the second option "save to my computer" and downloaded as an **.apk** file.

To build the application: From the build environment, use the build menu option.

##### Source Code

The **.aia** file or source code for a project can be exported for the build environment and shared among others. When sharing this file, others will have the option to load the project, and make changes as required.

##### Sharing a built app or source code

More details about sharing the .aia and .apk files [here](http://appinventor.mit.edu/explore/ai2/share.html)

More details about side loading [here](http://www.techrepublic.com/blog/smartphones/how-to-side-load-apps-on-your-android-device/)

#### Publish

Details about publishing your application (including distribution at the playStore) can be found [here](http://appinventor.mit.edu/explore/ai2/google-play.html)

##### Version control

Note: Every app you publish must have a versionCode and a versionName. You can set these in the designer under the properties panel for the Screen1 component.

#### Training

##### Complete the HelloPurr Tutorial

Follow [this tutorial](http://appinventor.mit.edu/explore/ai2/hellopurr.html) to setup the Hellopurr App

Note: Do not feel compelled to use the provided kitty image and meow soundFile. Feel free to find your own or ask your tutor for an alternate image/soundFile

***//TODO: Still need to check/test check if a phone needs to be logged in with same google account as the build site, or if any phone with the AI2 app can scan the QR code to gain access***

##### Complete some other tutorials

Many more tutorials can be found [here](http://appinventor.mit.edu/explore/ai2/tutorials.html). The tutorials can be filtered based on the feature you wish to learn.

**Many hands make light work:** perhaps this is a good point to decide who in your group might complete a couple of tutorials over the next week? If you each complete a couple of tutorials on a different feature, your group will have better coverage of skills.

## What about other tools???

To work efficiently as a group, you will need to decide on a few other tools which will assist you. The main tools you should consider are:

1. Collaborative Documentation: How will your group manage the documentation for this project? Will you use a tool such as google drive with google docs? What about dropbox for file sharing? How about sharing a one note workbook? It will be up to you and your group members to discuss and decide on this component.
2. Communication: How will you and your group communicate with each other? Will you make use of your student email? What about other options?
   * Facebook group chat?
   * Facebook group page?
   * Google Hangouts?
   * Skype?
   * Discord? What are the pro's and cons of each? Will you want features such as screen sharing or is a simple chat client all you require?

## Work Flow and Policies

Once you have become familiar with the MIT App Inventor and picked your tools for collaboration and communication, it is a good idea to set the policies your group will operate with.

### What is a Policy?

A policy is simply a document which outlines a *course* or *principle* of action which has been adopted or proposed by an organization or individual. For this project, you can think of it as defining a workflow (set of activities) to make the App development easier.

### Why is this important?

In your working life as an engineer, you will need to abide by any number of policies.

### Why is this important for your group now?

Have you noticed any limitations with the tools you are using? Once clear limitation is that the MIT App Inventor tool is not collaborative in the same way google docs or another typically collaborative tool might be.

Your group will need to work out how you wish to use this tool, so that there is one master application being developed, without falling prey to syncing issues or missing features and lost work due to poor organisation.

### How will you choose to do this?

There is a number of ways you could choose to use this tool collaboratively.

* Assign a group member as the "developer" who will be responsible to update the application, and perhaps distribute the .apk or .aia files
  + What is good about this option: Keeps development centralised and there is less likely to be any chance of conflict
  + What is not so good about this option: Only one person is the "developer". Is there a way around this? Perhaps you can change who is in this role each week. How would you switch developers week to week? Will you simply pass the .aia file to the next person - or will you keep it somewhere centrally for the whole group (such as dropbox, or google drive)
* Create a google "admin" account that everyone can use to login and work on the app
  + What is good about this option: Only one account is necessary and there is no need for team members to load/redistribute projects
  + What is not so good about this option: Even though there might be one account, you cannot have 2 group members making changes to the same project in the MIT App Inventor build environment at the same time. If you do so, there can be syncing issues, and it is possible that your development project will become irreparably damaged. So what can you do instead? Perhaps you can setup some kind of communication protocol, such as "add a post to the Facebook page - when you wish to work on the application". That way, everyone should be able to know who is working on the application and when.
* Distribute the .aia files in a central repository so that group members can load a project to edit, then redistribute the .aia file once they have made changes
  + What is good about this option: Anyone in the group can load and distribute a project.
  + What is not so good about this option: How will you handle merge conflicts? Suppose two group members both load project\_a.aia and make changes. The first member uploads project\_b.aia while the second uploads project\_c.aia. How will your group choose to make a project\_d.aia which incorporates the changes from project\_b.aia and project\_c.aia? And what happens if there are features in each project with are in direct conflict with each other? How can you manage this? If your group decide to distribute the .aia files you will need to decide on a policy for this kind of situation
* Are there any other options you can think of?
* Remember, whatever system you choose to implement will likely be unique to your group. The key thing will be deciding and clearly communicating how you and your group expect to manage the workflow, so that you experience the least number of issues when using the tool collaboratively.

## Give it to me straight?

This week, there are a few tasks you should start on:

1. Familiarise yourself with MIT App Inventor: complete the HelloPurr tutorial plus a couple of extra tutorials
2. You and your group should decided on:
   * collaborative documentation tool
   * communication streams/tools
3. In your documentation area, work together to develop a policy regarding how your group will manage the collaboration of the application development

Please note: The above items are not assessable for this course. However, groups who complete these tasks will have a much better chance of achieving higher marks on their assessable items.