Assignment 3
Spiking and Analysis of Alternatives

Group assignment - worth 10% of your final mark
Due Monday 17th April, 23:00

In this assignment, you'll be going through project inception: doing some initial investigation into what will be needed, and deciding on an approach that's right for your team.

The project:

"Morse Code Decoder: The server reads form the motion sensor and sends the data to clients that should interpret the received data into alphabets".

Note: for A3, you are not required to implement the above project.

Task 1. Spikes (10 + 10 = 20 marks)

There are several technologies you can use for this project. We've implemented most of them in previous labs.

1. Client-Server real-time bi-directional communication
    1.1. Socket.io for real-time bi-directional client server communication.
    1.2. Firebase real-time database as a communication platform.

2. Interact with IoT
    2.1. "Johnny-five" library
    2.2. "Serialport library". (will be described in FIT3140 Website)

There may be other toolchains that can be used to implement IoT projects- if you find a suitable one, you can evaluate it for this project (with the proviso that there must, at a minimum, be a free educational version available).

For Python developer, you can implement points 1.1, 1.2, and 2.2 in Python as shown in the following references:
  a) Socket.io:
     a.1. https://pypi.python.org/pypi/python-socketio,
     a.2. https://flask-socketio.readthedocs.io/en/latest/
  b) Firebase:
     b.1. https://pypi.python.org/pypi/python-firebase/1.2
     b.2. http://ozgur.github.io/python-firebase/
     b.3. https://github.com/thisbejim/Pyrebase
  c) Serialport:
     c.1. http://playground.arduino.cc/Interfacing/Python
     c.2. https://www.amazon.com/Python-Programming-Arduino-Pratik-Desai/dp/1783285931

Before you can decide on a toolchain, you'll need to investigate the options available to you.
There is a certain amount of technical risk associated with this decision: you need to select an option that you either know or can learn to use quickly, is not too hard to install and maintain, and can support all the required functionality.

To implement this project, you have to propose a platform that manages two messaging services: client-server (options 1.1, 1.2, a, or b) and IoT-server (options 2.1, 2.2, or c).

In order to investigate these options, we want you to implement some spikes. **A spike is a tiny program** – the smallest amount of code that demonstrates the feature or approach you're looking at. You can think of it as "throwaway" code, because it's usually not integrated into your final project, but it shouldn't be thrown away completely. That's because, when it's finished, it is a working example of your new toolkit that you can use for reference.

For each platform you want to investigate, you must write short programs that as a minimum:
- The server reads data (HIGH or LOW) from the motion sensor and prints it with the timestamp (current time) on the console.
- Send the data to the client.
- The client receives the data and calculates the response time. The response time is defined as the total time between sending the motion data from the server and the receiving by the client. In other words, the server has to send with the motion data, its timestamp that is going to be compared at the client side.

You should spike (i.e. code up) each of these features for each platform you are evaluating.

You must evaluate at least two platforms. You may combine multiple features in a single program, but I suggest that you make your programs as small and simple as possible. There's no point in writing a lot of glue code here. It would take a lot of time and not contribute to the risk reduction.

Spike code is not usually integrated into the final version of a project, so your code does not have to be fully robust or systematically documented in the same way production code would. You should comment it, though – you and your partner will be referring to it when you implement similar features in your project, so good inline comments on the mechanics of the code will save you a lot of time and effort later on.

You need to plan your spikes. You can use the template provided at the bottom of this document (on Moodle as well). Normally, you would also write an outcome report, but in this case the outcome reporting will be done in the analysis of alternatives.

Task 2. Analysis of alternatives (10 marks)

Once you've done some hands-on exploration, you need to write the whole thing up systematically in a report. Your report should cover:

- the criteria you're basing your choice on, both technical and nontechnical
- the options you are considering, with some basic background on each.
- for each option, how well it meets each of your criteria - including your evidence from conducting the spikes.
- which option you recommend

This is the kind of report you would usually write when your coworkers or managers have asked for your input on a technical decision. As such, you are writing for a technically-literate audience that values clear and concise writing. You may refer to external documents, websites, etc. in your report, provided you include them in your bibliography.

The exact criteria you choose to examine are up to you, but they should encompass things that you will need to efficiently develop and potentially support and maintain the app. This can include your existing expertise with specific technologies, but that is not a sufficient basis to make a decision. A report whose analysis reads something like "We know JavaScript. Johnny-five uses JavaScript. Therefore we will use Johnny-five." will receive a low mark!

The length of the report will depend on how many alternatives you consider and which criteria you choose to address.
Be very careful with your choice. You will be expected to stick with it for Assignments 4 and 5. This means that you'll need to consider your decision in the light of both partners' programming language background and skillsets

What to submit

Your group should submit:
- spike source code
- spike plan
- an analysis of alternatives detailing the toolchains you've investigated and clearly stating your preferred option

All required files are to be submitted to GitHub Assignment 3 repository and be archived in a ZIP archiving format and submitted to Moodle.

Marking criteria

When marking your submission, your demonstrator will take into account:
- the quality of your writing
- the clarity of your analysis report
- the thoroughness of your analysis and the feasibility of your recommendation
- the readability and presentation quality of your documents
- the functionality and readability of your spike code

You will receive a mark out of 30 for the assignment, but it contributes 10% to your final mark; we will divide the mark by 3 to calculate its contribution to your final grade.

Penalties

Late submissions attract a late penalty of 10% (**not 5%**) per working day, unless an extension has been arranged beforehand with the lecturer.

Although all students in a team will usually receive the same mark on group tasks, if there is evidence that the distribution of workload in the team is unfair, then we may penalize students who do not appear to be pulling their weight. Penalties may also be applied for breaches of the academic honesty policy.

Spike Plan

Name: *spikes should have a descriptive name*

Context:
In this section you should explain the general context in which the spike is being performed.

Gap:

In this section you should explain what risks the spike is being conducted to mitigate. That is, what do you hope to learn out of this spike to avoid during the rest of the project?

Goals/Deliverables:

This should be a reasonably detailed description of what you want to come out of the spike, including artefacts that will be delivered, to mitigate the risk.

Planned start date: eg 31/March/2012
Deadline: 6/April/2012

Planning notes:
Put in a description of your plan for achieving these. In what order are you to tackle these problems? Given that you are both working on this, how will you divide up responsibility and coordinate work?