School of Computing

Year 4 Project Proposal Form

SECTION A

Project Title: Source Code Analyser Engine

Student Name: Bernard O'Connor

Student ID: 14367821 Stream: CASE

Project Supervisor Name: Dr. David Sinclair

[Note: It is the student's responsibility to ensure that the Supervisor accepts your project and this is only recognised once the Supervisor assigns herself/himself via the project dashboard. Project proposals without an assigned Supervisor will not be accepted for presentation to the Approval Panel.]

SECTION B

Proposal Description – using the following headings:

General area covered by the project

My project will focus on the promotion of use of good coding conventions through the use of a source code analyzer engine which I will develop myself.

I with to provide an open source engine that is both available online through the use on a RESTful api and an offline version. My hope is that people may take my engine and use it in their online code teaching applications or to develop custom IDE plugins with it.

I will initially develop the code analyser engine to support Clojure, a functional language, but I wish to design the engine in a modular fashion so that other languages may be supported at a later date, possibly through crowd sourcing, which I will talk more about in the Achievements section.

Outline of the proposed project

Background - where the ideas came from

I have an interest in tutoring. In first year I was a programming lab tutor here in DCU. In the summer of Second Year I was considering approaching my old Secondary School to see if I could set up some sort of programming club / bootcamp as a summer job before I got my internship over the summer.

The idea of teaching coding stuck with me, and I felt that a tool that would give students feedback about the quality of their code would help make teaching a group of people more manageable.

Achievements - what functions it provides, who the users will be:
At the heart of the project, I wish the engine to be capable of taking some piece of source code, and to check if the code follows the conventional coding styles of the language in question. if the code does not do this, the engine will return a suggestion of how the code may be altered in order to improve the quality of the code.
An example of some of the suggestions that the engine could make would be the use of different functions to achieve the same thing in order to produce simpler code,

e.g in clojue instead of heavily nesting a group of operations in each other such as:
(prn (conj (reverse [1 2 3]) 4))
coders are instead encourage to use the thread-first marcro for readability:

```
(-> [1 2 3]
    reverse
    (conj 4)
    prn)
```

In the above example on each line the resulting value is fed in as the first argument to the function on the next line, whose result is fed in as the first argument of the next line, and so on.

I've mentioned before about trying to make language support modular, so that more languages can be supported at a later date and that the process of adding more language support can be crowdsourced. I hope to achieve this by separating the code to detect and suggest coding conventions into standardised "rulebooks". So initially I will create a Clojure "rulebook" that knows how to detect any poor code as outlined above, and what suggestions should be made.
So the workflow of the engine is that the initial part of the engine will read in the type of language being used and the source code. Then the engine will look up the rulebook for the the specific language being used, and then return to the user any suggestions that are returned from the rulebook. So I can crowd source the creation of the language specific rulebooks so long as they meet the standards that are expected by the engine.

Before I have mentioned that I wish the engine to be available for both online and offline use. To expand upon this I would want to host a version of the engine online myself, mainly for demonstration / quick prototype purposes. The functionality of the engine will be available through the use of a public API, which I will outline how to use on a small portal style website. The website will contain docs on how to use the api, how to contribute to the crowdsourced rulebooks, and links to download the engine, and then docs on how to use the engine locally, either on a webserver or how to use it for an IDE plugin. For the online hosted engine I will most probably have to put in place limits on the amount of processing that users may use, and if a user requires more processing power they must download and host the engine on their own webserver/machine.

For the Final Year Project Demonstration I intend to show an example of how a user may implement the engine both on their own website and if possible as part of an IDE plugin.

Justification - why/when/where/how it will be useful
With the ever increasing number of programming languages today I wish to enable the open source community to enhance their applications that they make to teach people how to use their language with my Source Code analyser Engine, so that not only will people be taught how to program in a language, but they will also be given feedback on how they may improve the quality of their code.

Programming language(s) - List the proposed language(s) to be used
I plan on using Clojure for the backend and ClojureScript for the frontend of my project

Programming tools / Tech stack – e.g. compiler, database, web server, etc.

- Leiningen (compiler & project manager)
- Clojail (Sandbox for Clojure code)
- Elasticsearch
- Figwheel / Jetty (Development and production web servers)
- Bootstrap (Web Elements Structure)
- Compojure (Routing library for clojurescript)
- CLJS-Ajax (ajax library for clojurescript)
- Reagent (interface between Clojurescript and React)
- Phantomjs (Clojure/Clojurescript testing suite)
- JSON reader and writers
- Intellij (IDE)

Learning Challenges - List the main new things (technologies, languages, tools, etc) that you will have to learn

- Designing an easy to use Public API
- Designing the project for use in multiple standards
- Identifying poor coding standards in source code
- Designing a non-language specific standard for the language rulebooks
- Creating an example code running/ teaching web application
- Creating an example IDE plugin

Hardware / software platform - State the hardware and software platform for development

- I will develop the project on a Linux laptop (Ubuntu). As Clojure and Clojurescript compile to java and javascript code respectively, the project should be runnable on both linux and windows devices, though I will focus my efforts on linux.

Special hardware / software requirements - Describe any special requirements.
No special requirements required.

Make use of figures / diagrams where appropriate.

Note: The final revision of your proposal form should be converted to a **PDF** in your GitLab repo from where it will be automatically collected.