Version 1.0 – RELEASE  
12th February 2018

Oliver Earl – [ole4@aber.ac.uk](mailto:ole4@aber.ac.uk)

G4R2: Computer Science with German

Supervised by: Dave Price - dap@aber.ac.uk

The Collation, Graphical Presentation and Distribution of Data from Magnetometers, Radio-Based Instruments, and the Internet To Track Solar Flare Activity Using a Web-Based Application

CS39440 – Major Project

# Project Description

This project is to develop a web-based application that collates data from numerous instruments, including magnetometers, radio-based equipment, and data from the Internet, such as that provided from the *European Space Agency* and allowing it to be accessed and viewed in graphical form by means of a web interface, as well as processing it and making it available it in a form easier used by physicists and computer scientists, such as JSON.

At present, the *Institute for Mathematics, Physics and Computer Science* at *Aberystwyth University* have a magnetometer, currently residing atop the *Physical Sciences* building, as well as numerous radio transmitters and receivers whose purpose it is to measure the Earth’s magnetic field and how it changes in response to solar activity such as solar flares. It is well known that solar activity has an sizable impact on the Earth’s magnetosphere and these events can be measured using the aforementioned instruments. However, working with the data produced by these instruments is a challenge as they are in differing formats, the majority of which are plain text files. An example of this is that current Physics undergraduates are required to graph the data manually on paper in their initial work using one of the institute’s magnetometers.

The project aims to streamline this process. The data is periodically (or on request if possible) retrieved from the specified source, such as one of the radio instruments. This data is converted to a format understood by the software and stored in a SQL database. Once stored in the application, it can be displayed via a front-end web interface using a JavaScript graph library, or it can be accessed in a computable form, such as JSON and downloaded. This is planned to operate in the form of a RESTful service, so that data can be retrieved programmatically. Both methods of data access will use password and/or token authentication to prevent unauthorised access.

The project uses a form of Scrum agile methodology for managing workflow and ensuring completion of all required tasks in a continuous manner. Due to the heightened importance of comprehensive documentation, additional time will be set aside during sprints to ensure that this is up-to-date. Required tasks and work are picked from a backlog as necessary until all required work is completed.

# Proposed Tasks

* **Research into appropriate web frameworks, languages and tools.** Many languages and frameworks are well suited for the task at hand, including but not limited to PHP and Laravel, Python and Django, Node.js and Ruby on Rails. It is would be highly beneficial that the language used supported RESTful implementations, would work seamlessly with an SQL database, and with a front-end that uses a JavaScript graphing library, such as Chart.js. HTML front-end libraries such as Foundation or Bootstrap are worth careful consideration in addition. Any tools or libraries used to maximise workflow efficiency, setting up automated end-to-end and unit level testing, and for deployment would also require research. Finally, Personal familiarity with any languages and frameworks would also be important.
* **Software Development**. Divided into the following primary subsections:
  + **1 Working Environments and Version Control**. Ensure any necessary software components are installed in the working environment. This includes webserver, backend programming language runtimes, debugging modules and anything else required. A similar environment should be set up for where the software is intended to be ran. For version control, Git will be used in conjunction with a private Git repository hosted on *BitBucket*.
  + **2 Communicating with Data Sources.** The web application needs to be able to retrieve information from necessary components, convert them to its own unified internal format and store them in a database. This will be namely a magnetometer, at least one radio-based instrument, and the ability to import data from online, potentially from a RESTful service or text file hosted elsewhere.
  + **3 Front-End Logic.** Once data is stored within the system, it must be retrievable and displayable using a front-end JavaScript graphing library so that users can compare and view data in a variety of graph formats. The web application itself must be easy to navigate and contain any necessary information as well as be completely responsive – working on all devices and display sizes. It should be protected by an authentication system.
  + **4 Computable Distribution.** Data in the system should be accessible programmatically, such as via the use of a RESTful service, and in a form that is useful to scientists, such as JSON or XML.
  + **5 Software Tests.** Unit and end-to-end test should be produced for all major functionality of the program, both in the backend language and for the frontend.
  + **6 REST CLI Client.** A small command line client should be developed for demonstrating and testing the programmatic accessing of data resources.
* **Project Meetings and Diary.** The project involves weekly group meetings with the supervisor and other candidates. One-to-one meetings will occur whenever they are necessary or requested by the supervisor. A diary is kept once a week, or more frequently should there be large breakthroughs in progress to track project progress in an informal way outside of the Scrum process. It is hosted on university-allocated web space using WordPress.
* **Demonstration Preparation.** Two demonstrations of the work are planned: one prior to the Easter break and one following the submission of the completed project. The program must be ready to be demonstrated, complete with sample data, at both stages regardless of progress. At the Final Demonstration, it should be possible to pull data from all configured sources, plot graphs of said data, as well as provide JSON (or similar) representations of data.
* **Report Finalisation.** This is any additional time outside of the development process to specifically work on the final report itself.

# Project Deliverables

* **Web Application**. The completed web application, including all necessary deployment scripts and libraries. A version is submitted, and is also available via Git.
* **REST CLI Client**. A small client will be included to demonstrate accessing resources from the web application in a RESTful way.
* **Software Tests.** Included with the software is a complete set of unit and end-to-end tests. The types of tests included will vary depending on technologies used, but may include tools such as Cucumber or PHPUnit.
* **Scrum Documentation**. All documentation outlying the agile process throughout development will be made available – information including but not limited to stories, sprint duration and backlog contents. This will be appended to the Final Report.
* **Mid Project Demonstration Report**. Summary of the mid-project demonstration, how it went, what was discussed, what was shown, and all feedback provided to be included as an appendix of the Final Report.
* **Code Documentation**. Generated documentation, such as JavaDoc or PHPDoc, will be appended to the Final Report.
* **Final Report**. Complete write-up for the project and will include any aforementioned appendices. Acknowledgements for 3rd party licenses, frameworks, libraries and tools are in this document.
* **Final Demonstration.** Not a physically tangible deliverable, but a final demonstration of the work done, and will be assessed.

# Initial Annotated Bibliography

[1] Langstaff, D. (2014). Worksheet [PDF]. Aberystwyth: Aberystwyth University.

PH25520 Experimental Physics – Measuring the Earth’s Magnetic Field

**Second Year Physics worksheet outlining usage of FGM-3 magnetometer tools to date, how to access the data and how to plot graphs manually.**

[2] Speake & Co Llanfapley (2001). Technical Document [PDF]. Abergavenny: Speake & Co Llanfapley – [billspeake@btconnect.com](mailto:billspeake@btconnect.com). FGM-Series Magnetic Field Sensors

**Technical documentation and instructions for the FGM-3 magnetometer.**

[3] Atlassian Corporation Plc (8th February 2018). The Git solution for professional teams. Retrieved February 11, 2018, from <https://www.bitbucket.com/>

**Provider of private Git repositories as well as tools for agile workflow, namely JIRA.**

[4] Leech, C. (2016, November 13). Ruby On Rails or Laravel, which do you use? Retrieved February 11, 2018, from <https://medium.com/@connorleech/php-laravel-ruby-on-rails-and-web-frameworks-32c1e50cea2d>

**Interesting article that goes in-depth into the differences between Ruby on Rails and the Laravel framework for PHP.**

[5] Bergmann, S. (2006). PHPUnit pocket guide: test-driven development in PHP. Beijing: OReilly.

**Helpful little pocket guide for PHPUnit – useful for beginners and those looking to carry out test-driven development in the PHP programming language.**