**Air Quality Mapping**

Final Report for CS39440 Major Project

*Author*: Robert Mouncer ([rdm10@aber.ac.uk](mailto:rdm10@aber.ac.uk))

*Supervisor*: Dr. Neal Snooke (nns@aber.ac.uk)

24th April 2018

Version 1.0 (Draft)

This report is submitted as partial fulfilment of a Meng degree in  
Software Engineering (G601)

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Wales, UK

**Declaration of originality**

I confirm that:

* This submission is my own work, except where clearly indicated.
* I understand that there are severe penalties for Unacceptable Academic Practice, which can lead to loss of marks or even the withholding of a degree.
* I have read the regulations on Unacceptable Academic Practice from the University’s Academic Quality and Records Office (AQRO) and the relevant sections of the current Student Handbook of the Department of Computer Science.
* In submitting this work I understand and agree to abide by the University’s regulations governing these issues.

Name Robert Mouncer

Date 27/04/2018

**Acknowledgements**

I am grateful to Dr Neal Snooke, my project supervisor, for overseeing the project and guiding me in the right direction when I’ve gone off course.

I’d like to thank Riversimple, my industrial placement of 2016 to 2017, for the idea of the air quality mapping project and allowing me to pursue it.

**Abstract**

Air pollution has had a large impact on the world from hazardous gases effecting the atmosphere to the death of millions of people each year. This has been an increasingly concerning problem in recent years as the after effects have been researched and linked back to pollution being one of the main problems. 40,000 deaths within the UK each year have been linked back to air quality levels with a large portion of deaths happening in major cities such as London. Janurary-March 2017 it was estimated that nearly 40 million vehicles are on Great British roads [1]. This has a large involvement on the air pollution levels within the UK, but these vehicles may help provide a solution to this problem.

The Royal College of Physicians released 14 steps needed to improve pollution levels, one of these was to “monitor air pollution effectively” and to “educate the public” [2]. This projects purpose is to provide a means of educating the public of the air quality on public roads. The project will contain two systems, a monitoring system that will be used within vehicles to collect a range of data while travelling and a visualisation system to be used to educate the public in a proactive way. The monitoring system has been designed with the idea of the system being implemented on a small percentage on public roads to help build a dataset.

This report will explain the procedure taken to complete these two systems and the process leading up to their success.

Overall the project was a success with two working systems, though the overall functionality is limited the main goals of this project have been completed. This project provides a proof of concept to the idea originally from Riversimple.

**Contents**

1. Background, Analysis & Process 3

1.1. Background 3

1.2. Analysis 3

1.3. Process 3

2. Design 3

2.1. Overall Architecture 3

2.2. Detailed Design 3

2.2.1. Even More Detail 3

2.3. User Interface Design 3

2.4. Other Relevant Sections 3

3. Implementation 3

4. Testing 3

4.1. Overall Approach to Testing 3

4.2. Automated Testing 3

4.2.1. Unit Tests 3

4.2.2. User Interface Testing 3

4.2.3. Stress Testing 3

4.2.4. Other Types of Testing 3

4.3. Integration Testing 3

4.4. User Testing 3

5. Critical Evaluation 3

6. Appendices 3

A. Third-Party Code and Libraries 3

B. Ethics Submission 3

C. Code Samples 3

Annotated Bibliography 3

# Background, Analysis & Process

This section should discuss your preparation for the project, including background reading, your analysis of the problem and the process or method you have followed to help structure your work. It is likely that you will reuse part of your outline project specification, but at this point in the project you should have more to talk about.

**Notes:**

* All of the sections and text in this example are for illustration purposes. The main Chapters are a good starting point, but the content and actual sections that you include are likely to be different.
* Look at the document on the MMP: Final Report and Technical Work [7] for additional guidance.

## Background

What was your background preparation for the project? What similar systems did you assess? What was your motivation and interest in this project?

During the industrial placement at Riversimple from September 2016 to September 2017 a telemetry unit was being built to collect data from a hydrogen vehicle to monitor how customers were driving and report any fault with the vehicle. An employee at Riversimple mentioned an idea of being able to collect air quality data with the telemetry unit and use that data to show the public a comparison of air qualities around the country. With permission from Riversimple to pursue this project, the aim was to create a method of monitoring and visualisation air quality data inspired by the telemetry unit.

During the industrial placement it became clear of the state that the air pollution in the country was in due to personal and public transport. The motivation of this project stems from this and has developed over time. With little experience of such systems, the challenge was part of the motivation to complete this project and ensure its success.

**Background Preparation**

There was little background preparation for this project until the project had been reviewed and accepted as a major project. It began between the project being accepted and the start date of the major project.

The telemetry unit was built using a Raspberry Pi Model 3B+ (RPI). With little experience of using a RPI, preparation was carried out to ensure general knowledge of the microcontroller was known before work was carried out.

The operating system (OS) for the RPI were a vital part of the preparation, it needed to be decided what OS was to be used to suit the task at hand. Most RPI operating systems run using a variation of Linux. The time leading up to the project hand out, comparisons were made between the OS’s.

Supported protocols for the RPI were investigated, it turned out that most protocols could be used but had to be connected to the dedicated pin on the RPI board. If a protocol was not supported locally, a RPI HAT could be used to allow for support. A RPI HAT is an add-on board for the RPI, it stands for hardware attached on-top.

Hardware components needed to be investigated to ensure that it was possible to achieve what had been set out. Using the microcontroller, location and air quality data needed to be collected. Knowledge was known about GPS’s and being able to get the current location using them, but little was known about air quality sensors. Air quality sensors were found online but most of them needed additional components or were not suitable for the project. The accurate sensors and multi-gas sensors were very expensive and due to the extent of the project were not suitable. Several sensors were appropriate for the project. From this I knew I could create the hardware needed for the project, and if they were not very accurate or reliable, it would still prove as a proof of concept.

Having previous used a Google Maps API for commercial purposes, it was known the license agreement was not very permissive, research on other online map providers was conducted. OpenStreetMap was found to be very permissive and only required recognition on the webpage it was being used for.

Research was conducted on similar products to understand what solutions had been created and what to avoid for copyright purposes.

A blog was created to record what had been learnt and to keep track of any work that had been completed. This blog was hosted on the Aberystwyth University public\_html directory. My supervisor was given access to this to keep up to date with what had been worked on.

A GitHub account had already been created for personal use and this was used for the major project, though the repository was set to private.

**Similar Systems**

When researching the monitoring hardware and implementation it was discovered that a team measuring air pollution within London using pigeons [3], the pigeons would wear small backpacks with air quality sensors and a GPS. The type of hardware is the same, but the deployment is different.



Figure 1 - Pigeon Air Patrol

Pigeon Air Patrol also has an interactive map which shows the air pollution across London, no values are shown, rather an indicator is used whether the area has “fresh air” or a certain level of pollution such as “moderate” or “high”.

Plume labs, the same company that deployed Pigeon Air Patrol also are working on a device to measure air quality and location. It is a smart air quality tracker designed to be attached to a user’s possession, such as a bag or bicycle. This has not yet been released so information on it is limited.

The Department for Environment, Food and Rural affairs (DEFRA) have a pollution mapping website that is forecasted by the met office. The information states that the data is collected in various regions from monitoring sites and generated from current air quality issues.

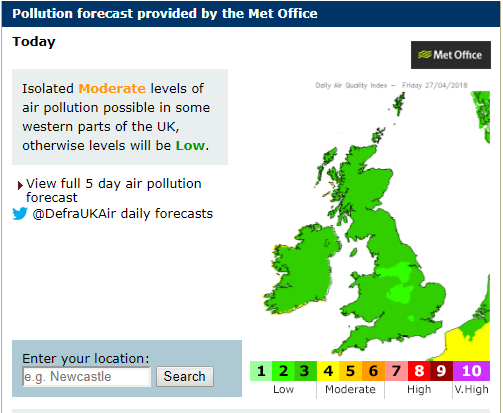


Figure 2 - UK-air DEFRA screenshot

Figure 1 shows a forecast of the pollution over the UK, with a non-interactive map, the only interaction is a search functionality. Once a search has been complete, the map becomes interactive using google maps. This then shows values of pollution at different points, rather than the whole area, as can be seen in figure 2.

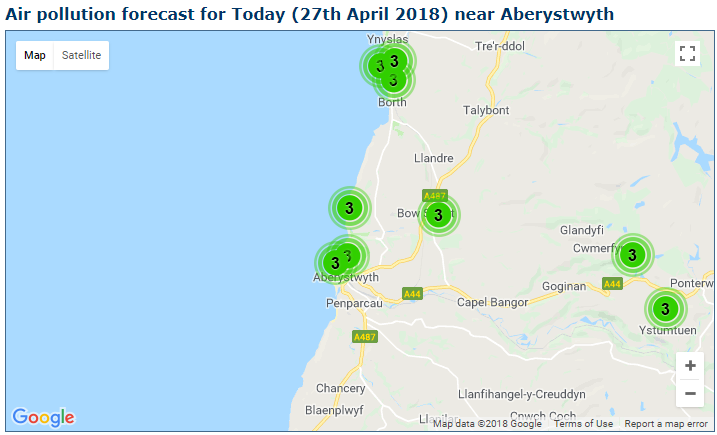


Figure 3 - interactive map on UK-air DEFRA

The difference between this and the major project is that fixed monitoring stations weren’t used. The idea was to use vehicles as monitoring stations as even for local use, can still collect a lot of information regarding the air quality. The visualisation of the map was designed as a heatmap/contour map.

The DEFRA website also has links to Wales, Scotland and Northern Ireland air quality sites. The Welsh site lacks even more functionality than the UK site. The Welsh site uses less monitoring stations than Wales has on the UK site. The Scottish site has the most interactive welcoming screen.

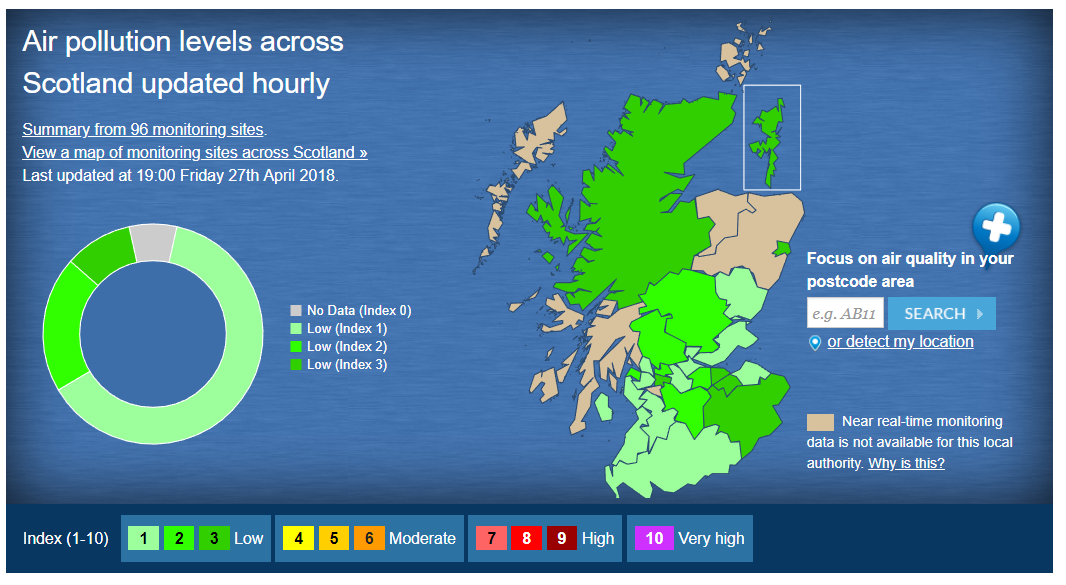


Figure 4 - Interactive welcome screen Scottish Air Quality

This allows users to select their location by clicking on their province. The design for the visualisation site is different to this and resembles no correlation. The interactive map is the same as the welsh and UK site, this is not what the major project aimed to achieve.

Riversimple contacted me before the major project started to make me aware of what they thought was a similar website. They had been in contact with the company at an event and were made aware of the website before I was.

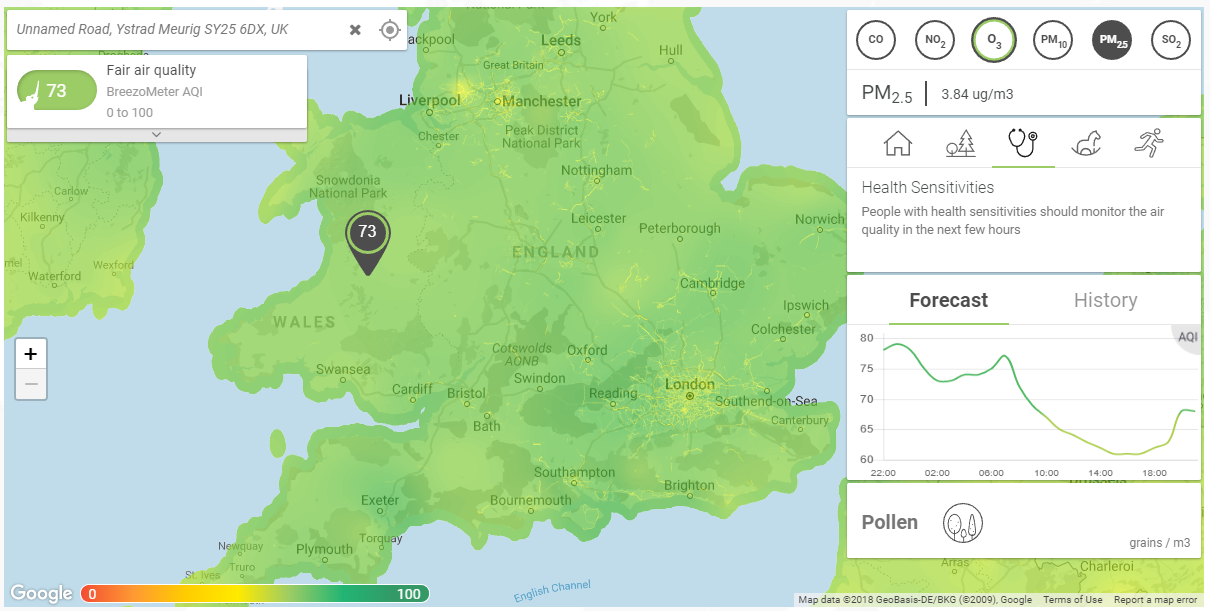


Figure 5 - Breezometer interactive map

Breezometer uses weather/air pollution stations and machine learning algorithms to predict the values. The monitoring system designed for this project is to use real world values rather than from monitoring stations and predicting the values. The interaction on Breezometer is very good and offers a lot of information. It shows different pollutants as one value on the map, but on the information menu, you can find individual information on the pollutants. Information is also given on the health issues and sensitives, this is a similar idea that was not fully implemented on the final version of the visualisation tool.

## Analysis

After studying the background research, it was drawn to a conclusion that this project would be possible within the timeframe. Various assumptions were made during the start of the project. The project would be split into two major components:

1. Monitoring system
2. Visualisation System

The monitoring system would collect the data and would need to be designed to be suitable within automobiles. The output of the monitoring system would be sets of data from journeys that would need to be uploaded to Aberystwyth University MySQL server.

The visualisation system would need to use the data collected from monitoring system that would be stored on the MySQL server. This would then be displayed to the user in a proactive way.

It was decided to use the Aberystwyth University MySQL server to store the data and public hosting for the website as access to it was very easy. It was already set up to suit the needs of this project.

Implementation of the project would require several different skills ranging from the use of hardware components in the creation of the monitoring system to web development.

The required skills needed and those to be developed are:

* Hardware selection
* Hardware creation
* Linux installation, command line and configurations
* Network Administration
* Python skills
* MySQL skills (Database Administration)
* Web development skills

The hardware would consist of a Raspberry Pi Model 3B+, air quality sensor and a GPS. The Raspberry Pi Model 3B+ was used because this resembles the microcontroller that Riversimple use for the telemetry unit that is currently being developed. This would be controlled by a Linux distribution called Raspbian Lite, meaning there is no GUI (Graphical User Interface) and would require an SSH (Secure Shell) connection or serial connection to be communicated with. Raspbian Lite was selected due to the RPI being a lightweight microcontroller, GUI is resource intensive and when the monitoring system is working a GUI is not required. Using the Lite version of Raspbian would require Linux command line skills, it was also chosen to improve these skills for future projects.

The RPI supports many different protocols for components such as UART (asynchronous receiver-transmitter), I2C (Inter-integrated Circuit), SPI (Serial Peripheral Interface) and provides GPIO (General-Purpose Input/Output). This was another reason that the RPI was chosen. A Raspberry Pi HAT can be used if a hardware component was not supported by default such as CAN (Controller Area Network) which is most often used in the automotive sector.

As it wouldn’t be viable to create the software running on the RPI through the SSH, a samba server needed to be set up to allow a standard client-server interaction. The samba server would host the files while it was possible to connect to the server from another computer and edit the files using an IDE or advanced text editor.

It became clear that the software on the RPI would not need to be that advanced but would rely on the Linux distro to be correctly configured to allow the scripts to run on start-up and upload data collected. It was decided that only two files would need to be created to collect and upload the data.

The use of the Aberystwyth University resources would need to be considered. Hosting the webpage on the public\_html directory would be needed as it allows the webpage to be accessed from anywhere, this would be useful when giving demonstrations or presentations on the current state of the project. The hosting server also allows the use of php scripts, this would be used to get the data from the server and not allow the user access to the database, it will be handled server side.

It was decided that the visualisation application would use OpenStreetMap (OSM) as the map provider. The online community for developing OSM applications is quite large due to the data being geological data being open to anyone to use with only recognition needed to be given. The data provided by Google Maps is copyrighted by many organisations and it wasn’t clear whether I would be limited by Googles API or Terms.

The visualisation application would need to show the data in a proactive way for easy educational purposes. It was proposed that either a contour or heatmap would be used to display the data for easy comparisons of different areas. OSM supports a variety of plugins, including ones for creating heatmaps, this was another reason to choose OSM over Google Maps.

Security was an issue with this project at this point as connection to a personal database is needed on both systems being developed. At this point security solutions had not been considered.

The functionality of each system had been defined by this point, how the functionality would be implemented was not.

The objectives of work at this point in the project were:

* Select the appropriate hardware to work with the model 3B+ RPI
* Install Raspbian Lite onto an SD for the RPI
* Set up a Samba Server on the RPI for easy development
* Hardware components design and assembly
* Design of Monitoring system
* Development of the monitoring system
* Design of Visualisation system
* Development of the Visualisation system
* Testing of both systems

Taking into account the problem and what you learned from the background work, what was your analysis of the problem? How did your analysis help to decompose the problem into the main tasks that you would undertake? Were there alternative approaches? Why did you choose one approach compared to the alternatives?

There should be a clear statement of the objectives of the work, which you will evaluate at the end of the work.

In most cases, the agreed objectives or requirements will be the result of a compromise between what would ideally have been produced and what was felt to be possible in the time available. A discussion of the process of arriving at the final list is usually appropriate.

As mentioned in the lectures, think about possible security issues for the project topic. Whilst these might not be relevant for all projects, do consider if there are relevant for your project. Where there are relevant security issues, discuss how they will this affect the work that you are doing. Carry forward this discussion into relevant areas for design, implementation and testing.

## Process

You need to describe briefly the life cycle model or research method that you used. You do not need to write about all of the different process models that you are aware of. Focus on the process model that you have used. It is possible that you needed to adapt an existing process model to suit your project; clearly identify what you used and how you adapted it for your needs.

# Design

You should concentrate on the more important aspects of the design. It is essential that an overview is presented before going into detail. As well as describing the design adopted it must also explain what other designs were considered and why they were rejected.

The design should describe what you expected to do, and might also explain areas that you had to revise after some investigation.

Typically, for an object-oriented design, the discussion will focus on the choice of objects and classes and the allocation of methods to classes. The use made of reusable components should be described and their source referenced. Particularly important decisions concerning data structures usually affect the architecture of a system and so should be described here.

How much material you include on detailed design and implementation will depend very much on the nature of the project. It should not be padded out. Think about the significant aspects of your system. For example, describe the design of the user interface if it is a critical aspect of your system, or provide detail about methods and data structures that are not trivial. Do not spend time on long lists of trivial items and repetitive descriptions. If in doubt about what is appropriate, speak to your supervisor.

You should also identify any support tools that you used. You should discuss your choice of implementation tools - programming language, compilers, database management system, program development environment, etc.

Some example sub-sections may be as follows, but the specific sections are for you to define.

## Overall Architecture

## Detailed Design

### Even More Detail

## User Interface Design

## Other Relevant Sections

# Implementation

The implementation should look at any issues you encountered as you tried to implement your design. During the work, you might have found that elements of your design were unnecessary or overly complex; perhaps third party libraries were available that simplified some of the functions that you intended to implement. If things were easier in some areas, then how did you adapt your project to take account of your findings?

It is more likely that things were more complex than you first thought. In particular, were there any problems or difficulties that you found during implementation that you had to address? Did such problems simply delay you or were they more significant?

You can conclude this section by reviewing the end of the implementation stage against the planned requirements.

# Testing

Detailed descriptions of every test case are definitely not what is required here. What is important is to show that you adopted a sensible strategy that was, in principle, capable of testing the system adequately even if you did not have the time to test the system fully.

Provide information in the body of your report and the appendix to explain the testing that has been performed. How does this testing address the requirements and design for the project?

How comprehensive is the testing within the constraints of the project? Are you testing the normal working behaviour? Are you testing the exceptional behaviour, e.g. error conditions? Are you testing security issues if they are relevant for your project?

Have you tested your system on “real users”? For example, if your system is supposed to solve a problem for a business, then it would be appropriate to present your approach to involve the users in the testing process and to record the results that you obtained. Depending on the level of detail, it is likely that you would put any detailed results in an appendix.

The following sections indicate some areas you might include. Other sections may be more appropriate to your project.

## Overall Approach to Testing

## Automated Testing

### Unit Tests

### User Interface Testing

### Stress Testing

### Other Types of Testing

## Integration Testing

## User Testing

# Critical Evaluation

Examiners expect to find in your dissertation a section addressing such questions as:

* Were the requirements correctly identified?
* Were the design decisions correct?
* Could a more suitable set of tools have been chosen?
* How well did the software meet the needs of those who were expecting to use it?
* How well were any other project aims achieved?
* If you were starting again, what would you do differently?

Other questions can be addressed as appropriate for a project.

Such material is regarded as an important part of the dissertation; it should demonstrate that you are capable not only of carrying out a piece of work but also of thinking critically about how you did it and how you might have done it better. This is seen as an important part of an honours degree.

There will be good things and room for improvement with any project. As you write this section, identify and discuss the parts of the work that went well and also consider ways in which the work could be improved.

In the latter stages of the module, we will discuss the evaluation. That will probably be around week 9, although that differs each year.

# Appendices

The appendices are for additional content that is useful to support the discussion in the report. It is material that is not necessarily needed in the body of the report, but its inclusion in the appendices makes it easy to access.

For example, if you have developed a Design Specification document as part of a plan-driven approach for the project, then it would be appropriate to include that document as an appendix. In the body of your report you would highlight the most interesting aspects of the design, referring your reader to the full specification for further detail.

If you have taken an agile approach to developing the project, then you may be less likely to have developed a full requirements specification. Perhaps you use stories to keep track of the functionality and the ’future conversations’. It might not be relevant to include all of those in the body of your report. Instead, you might include those in an appendix.

There is a balance to be struck between what is relevant to include in the body of your report and whether additional supporting evidence is appropriate in the appendices. Speak to your supervisor or the module coordinator if you have questions about this.

* 1. Third-Party Code and Libraries

If you have made use of any third party code or software libraries, i.e. any code that you have not designed and written yourself, then you must include this appendix.

As has been said in lectures, it is acceptable and likely that you will make use of third-party code and software libraries. If third party code or libraries are used, your work will build on that to produce notable new work. The key requirement is that we understand what is your original work and what work is based on that of other people.

Therefore, you need to clearly state what you have used and where the original material can be found. Also, if you have made any changes to the original versions, you must explain what you have changed.

As an example, you might include a definition such as:

**Apache POI library** – The project has been used to read and write Microsoft Excel files (XLS) as part of the interaction with the client’s existing system for processing data. Version 3.10-FINAL was used. The library is open source and it is available from the Apache Software Foundation [5]. The library is released using the Apache License [6]. This library was used without modification.

* 1. Ethics Submission

This appendix includes a copy of the ethics submission for the project. After you have completed your Ethics submission, you will receive a PDF with a summary of the comments. That document should be embedded in this report, either as images, an embedded PDF or as copied text. The content should also include the Ethics Application Number that you receive.

* 1. Code Samples

This is an example appendix. Include as many appendices as you need. The appendices do not count towards the overall word count for the report.

For some projects, it might be relevant to include some code extracts in an appendix. You are not expected to put all of your code here - the correct place for all of your code is in the technical submission that is made in addition to the Final Report. However, if there are some notable aspects of the code that you discuss, including that in an appendix might be useful to make it easier for your readers to access.

As a general guide, if you are discussing short extracts of code then you are advised to include such code in the body of the report. If there is a longer extract that is relevant, then you might include it as shown in the following section.

Only include code in the appendix if that code is discussed and referred to in the body of the report.

Random Number Generator

The Bayes Durham Shuffle ensures that the pseudo random numbers used in the simulation are further shuffled, ensuring minimal correlation between subsequent random outputs.

// Some example code here…

# Annotated Bibliography