Security in IOT Devices

Final report

Submitted for the BSc in

Computer Science

April 2021

by

Kai Christopher Michael Tindall

Word count: XXXXX

This document is a template for your guidance. You don’t have to stick to it precisely. It may not suit your particular project. Modify it if you need to – but please discuss with your supervisor before making significant changes to the organisation and content.

Paragraphs in red, like this one, are instructions and extra information. You must delete them before submitting your report.

This template document has a number of paragraph styles predefined. If you use them (‘Heading 1’, ‘Heading 2’ and ‘Heading 3’) and don’t alter them, then your report will automatically have properly numbered paragraphs and your table of contents will be automatically generated with the right page numbers. Use ‘Normal’ as the style for general text paragraphs in your document.

The document is divided into sections by ‘section breaks’ in Word. These are needed to keep the page numbering correct, so only delete them if you know what you are doing.

On this page (and I hope this is obvious) you must replace the words “The Title Of Your Project”, “BSc/MEng”, “Your degree title here” and “Your Full Name Here” with the correct information. You’d be surprised how often people forget to do this.

You must also replace “XXXXX” with the actual word count (excluding acknowledgements, abstract, table of contents, references and appendices) of your document. Note penalties for exceeding the word count will be applied, according to university regulations.

Abstract

The abstract is a short, self-contained statement describing the whole of your work. It should be less than a page (typically half a page) and should summarise the scope, purpose, results and content of the work. The abstract might be thought of as a summary which you would read quickly to decide if the rest of the document is worth taking the time to read in detail. In scientific publishing, abstracts are often used as sources of keywords and concepts for searching, so it’s important to ensure that the main ideas and conclusions of your work are present.

When someone has read your abstract, they should know what your project was about, how you did it and what the end result was. It doesn’t need to contain references or literature reviews. For example, here’s the abstract from Bell and Brooks (2019)’ s paper “*What makes students satisfied? A discussion and analysis of the UK's national student survey*” (which, incidentally, is very interesting):

“*This paper analyses data from the National Students Survey, determining which groups of students expressed the greatest levels of satisfaction. We find students registered on clinical degrees and those studying humanities to be the most satisfied, with those in general engineering and media studies the least. We also find contentment to be higher among part-time students, and significantly higher among Russell group and post-1992 universities. We further investigate the sub-areas that drive overall student satisfaction, finding teaching and course organisation to be the most important aspects, with resources and assessment and feedback far less relevant. We then develop a multi-attribute measure of satisfaction which we argue produces a more accurate and more stable reflection of overall student satisfaction than that based on a single question.*”

Note that it’s very specific. There is no waffle, just details of what they did, how they did it, and what their conclusions were.

Delete the red paragraphs and replace this one with your content (use the “Normal” paragraph style).

This project creates a secure environment for IoT devices to transmit their data to a central control server. This server then publishes that information to an API that the end use can access. I did this using the open-source Crypto++ library for the cryptological functionality. TALK ABOUT RESULTS WHEN AVAILABLE. IF USED HTTP AS API THEN TALK ABOUT LIBS USED FOR THAT.

Acknowledgements

*I would like to acknowledge the support of the Crypto++ Library contributors for their useful online resources and the free and open nature of their project. Without them this project would have taken significantly longer.*

Contents

[Abstract i](#_Toc31119449)

[Acknowledgements ii](#_Toc31119450)

[1 Introduction 3](#_Toc31119451)

[1.1 Background to the project 3](#_Toc31119452)

[1.2 Aims and objectives 3](#_Toc31119453)

[1.3 Research question 3](#_Toc31119454)

[2 Literature review 4](#_Toc31119455)

[3 Requirements 5](#_Toc31119456)

[3.1 Product requirements 5](#_Toc31119457)

[3.2 Functional requirements 5](#_Toc31119458)

[3.2.1 Interfaces 5](#_Toc31119459)

[3.2.2 Functional Capabilities 5](#_Toc31119460)

[3.2.3 Performance Levels 5](#_Toc31119461)

[3.2.4 Data Structures/Elements 5](#_Toc31119462)

[3.2.5 Safety 5](#_Toc31119463)

[3.2.6 Reliability 5](#_Toc31119464)

[3.2.7 Security/Privacy 5](#_Toc31119465)

[3.2.8 Quality 5](#_Toc31119466)

[3.2.9 Constraints and Limitations 5](#_Toc31119467)

[3.2.10 Performance requirements 5](#_Toc31119468)

[3.3 Design constraints 5](#_Toc31119469)

[4 Design 6](#_Toc31119470)

[4.1 Software design 6](#_Toc31119471)

[4.2 Experimental design 6](#_Toc31119472)

[5 Implementation and testing 7](#_Toc31119473)

[5.1 Implementation 7](#_Toc31119474)

[5.2 Testing 7](#_Toc31119475)

[6 Evaluation 8](#_Toc31119476)

[7 Conclusion 9](#_Toc31119477)

[References 10](#_Toc31119478)

[Appendix A – Interesting but not vital material 11](#_Toc31119479)

[Appendix B – Other things which may be useful 12](#_Toc31119480)

# Introduction

In this section you will describe the project’s purpose, aims and objectives. You will introduce the project’s stakeholders and the reason for doing it. You will provide a brief overview of the report’s organisation – which may of course be different to this template.

There may be some overlap with the content of the PID in this section, but it should not simply be a repeat. The introduction in this report will be informed by the activities you have undertaken and their results, whereas the PID was concerned with forward planning.

Note that the sub-headings below are suggestions only; you may organise this section differently as appropriate to your project.

Delete the red paragraphs and replace this one with your content (use the “Normal” paragraph style).

## Background to the project

## Aims and objectives

## Research question

# Literature review

The literature review is a survey of the history and state of the art in the domain of your project. It will summarize the work that has already been done in the field; this may be scientific literature, known techniques, and even previous student projects. It will provide a historical perspective on how the subject area has arrived at its current state by looking at important developments over time. If appropriate, it may examine existing software in the domain, especially in terms of the technology used and the features offered. The focus of the literature review is to summarise the existing arguments and ideas of others, identifying which are important.

A good literature review could be a project on its own, and form a very useful guide to anyone new to the particular field. It would identify the important work, authors and publications which would be a good place to begin research activity. Open questions and areas where new work is required would be discussed. Really good reviews are often published in scientific journals. Your literature review is not expected to be quite so substantial, but should still provide a comprehensive summary which will allow the reader to understand the field.

Images can be very useful here. Remember to attribute them properly to avoid accusations of plagiarism. Your literature review will naturally refer to lots of existing work, which must all be properly cited and referenced – see the ‘References’ section towards the end of this document.

Delete the red paragraphs and replace this one with your content (use the “Normal” paragraph style).

# Requirements

If your project is primarily concerned with developing software or hardware, then you will be expected to include a section describing the requirements. Some of this might well come from primary research. If so, document it here. If yours is a more theoretically based research project, this section might be ‘Theoretical development’ instead. These requirements will have a basis in the PID’s objectives and deliverables, but they may have changed. If so, explain why.

The suggestions below are not definitive.

Delete the red paragraphs and replace the ones below with your content. Modify them to suit your project.

## Product requirements

What will your software or hardware do? Who requires it? You might want to refer back to your aims and objectives to inform this section, and perhaps consider if they are still appropriate. UML use case diagrams are very helpful here (even for hardware).

Delete the red paragraph and replace this one with your content (use the “Normal” paragraph style).

## Functional requirements

The exact content here will vary (especially if your project is hardware-based), but there are some standard items which you should consider including:

### Interfaces

### Functional Capabilities

### Performance Levels

### Data Structures/Elements

### Safety

### Reliability

### Security/Privacy

### Quality

### Constraints and Limitations

### Performance requirements

## Design constraints

You might include this in the next section if you prefer. Consider the limitations on how you are able to conduct your project. Relate the bounds (time and resources are obvious ones) which have an impact.

Delete the red paragraph and replace this one with your content (use the “Normal” paragraph style).

# Design

If your project involves the development of software and/or hardware, then you will need to include a section in which you describe its design in detail. If you conduct any experiments (either in a research-oriented project or simply doing user evaluation) then you should describe their design and methodology here.

Delete the red paragraph and replace this one with your content (use the “Normal” paragraph style).

## Software design

Typical content will be detailed software design, from architecture to implementation level. As well as your text, you should include UML diagrams, including class structures, activity and sequence diagrams as appropriate. Don’t just drop diagrams in willy-nilly, though. Use them strategically to illustrate points in your text. Remember that ‘a picture is worth a thousand words’ (we don’t apply this rule literally) but pictures on their own don’t explain everything.

If your project requires user interface design, don’t forget to include that. Screenshots, wireframes and other diagrams are welcome.

Delete the red paragraph and replace this one with your content (use the “Normal” paragraph style).

## Hardware design

If your project involves building hardware, give full details about the process here. Include diagrams as appropriate Use them strategically to illustrate points in your text. Remember that ‘a picture is worth a thousand words’ (we don’t apply this rule literally) but pictures on their own don’t explain everything.

If your project requires electronics and/or mechanical design, don’t forget to include that. Photos, CAD drawings, electronic schematics and other diagrams are welcome.

Delete the red paragraph and replace this one with your content (use the “Normal” paragraph style).

## Experimental design

If you are going to evaluate your software or hardware by means of any tests or surveys, then explain their design here. If you are doing other experiments (for example measuring the performance of algorithms, extracting data from environmental monitoring systems or evaluating the performance of mechanisms) then you should explain how you have designed the experiments, how they must be conducted and what you expect to learn from them. This is especially important for research-oriented projects.

Delete the red paragraph and replace this one with your content (use the “Normal” paragraph style).

# Implementation and testing

Delete the red paragraphs and replace this one with your content (use the “Normal” paragraph style).

## Implementation

In this section you will describe what you did, and why you made the important decisions affecting your actions. It’s not a diary – don’t write a blow-by-blow account of every little thing that happened. Be selective and report those choices and techniques which made a difference. Make sure you discuss what options you considered. Explain how the criteria and methodology you used to select amongst different options (which tools are most appropriate, for example).

It may help to imagine that you are reading this project in the future, trying to replicate the work without making the same mistakes along the way. What would you need to know to make your job easier, and what is unimportant or obvious? Explain how you implemented the design in the previous chapter.

This is the place in which you would explain any novel or especially complex algorithms, data structures or systems you have used.

Make it clear what you have done, and what is pre-existing. For example, if you are using third party software libraries, describe how you have used them, and how they have benefited your project rather than simply what they do. If you have built on a framework, make it clear how you have developed new functionality.

Delete the red paragraphs and replace this one with your content (use the “Normal” paragraph style).

## Testing

If you are developing software or hardware, you must test it. This section should explain how your work will be (or has been) tested.

You should have a test plan at the very least (full details of it and its results if required can go in an appendix). Ideally, you will have automated tests for any software you build. You will also define user acceptance tests, or something similar which can be used to determine whether your output meets the requirements stated earlier. Explain how and when the tests should be conducted.

Delete the red paragraphs and replace this one with your content (use the “Normal” paragraph style).

# Evaluation

This section evaluates the *software (or other artefact)* you have developed. You should compare it with the original specification and see how well it satisfies the requirements. You may wish to refer back to your aims and objectives at this point. You should report the results of user testing and a summary of feedback if that has been collected.

If you have done experiments, then the results of these should be reported and discussed here.

If you have involved people in doing user evaluations, that information should be include here.

Delete the red paragraphs and replace this one with your content (use the “Normal” paragraph style).

# Conclusion

In this section you should evaluate the *project* as a whole, and draw conclusions from the work you have done. Ask yourself what the project has achieved – what is its contribution? Has it met its initial aims and objectives? If not, why? How does the work you have done enhance the field in general? What has been learned from the project? If you have a well defined research question, has it been answered? What do the results mean?

You should also use this section to reflect on the *process* by which you undertook the project. Was your methodology appropriate (and did you stick to it)? Was your time planning good? Did you complete the primary and secondary objectives, and if not then why? What have you learned from the process? What would you do better/differently if you had more time?

Sometimes, it’s appropriate to include a subsection on ‘Further work’, making suggestions of how to proceed and what could be done to enhance the project in future.

Delete the red paragraphs and replace this one with your content (use the “Normal” paragraph style).

References

References must be formatted in the correct manner. For this assignment you must use the University of Hull’ approved variant of the Harvard referencing style (Fallin 2019), fully described at https://libguides.hull.ac.uk/referencing/harvard. Note that the details of the expected format vary depending on the type of document being referenced. Make sure you are familiar with them. If you use reference management software such as Zotero, EndNote or RefWorks, then you should be able to export a table of references in the correct format, which will save you work.

Every reference should have at least one citation in the text. Most will probably be in the ‘Background’ or ‘Literature review’ sections.

Remember that there is a difference between references and a bibliography. You will certainly need references, but a bibliography is optional.

There is much more information and guidance about referencing on the library’s website at https://libguides.hull.ac.uk/referencing/home

Some examples, illustrating different types of source:

Bahraini, M.S., Bozorg, M., Rad, A.B., (2018). SLAM in dynamic environments via ML-RANSAC. *Mechatronics* 49, 105–118.

Fallin, L., (2019)*. LibGuides: Referencing your work: Harvard Hull.* Available online: http://libguides.hull.ac.uk/referencing/harvard (accessed 10/10/2019).

Janis, I., (1972). *Victims of Groupthink: A psychological study of foreign-policy decisions and fiascoes.* Houghton Mifflin, Boston.

Office For Students (2018) *. Securing student success: Regulation framework for higher education in England*. Available online: https://www.officeforstudents.org.uk/media/1406/ofs2018\_01.pdf (accessed 10/10/2019)

Schmuck, P., Chli, M., (2019). CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams. *Journal of Field Robotics* 36, 763–781.

Delete the red paragraphs and replace this one with your content (use the “Normal” paragraph style).

Appendix A – Interesting but not vital material

Appendices are used to include information which may be of interest but is not necessary for the reader. You do not have to include appendices if there is no need for them.

You might, for example, want to include some details of a particular piece of software (an API, perhaps) or hardware which your project uses. This might be something that a reader might wish to consult, but you wouldn’t want to include in the main body of the report. You could also put raw data from experiments in an appendix, or perhaps survey results. It should still be information of relevance, but nothing that everyone would be expected to read.

If you wish to refer to elements of your PID, you could include them in appendices.

Delete the red paragraphs and replace this one with your content (use the “Normal” paragraph style).

Appendix B – Other things which may be useful

You can have more than one appendix, or none at all. Give them meaningful names and titles (not the ones given here), so that you can refer to them in the text, and so that they appear in the table of contents.

Delete the red paragraphs and replace this one with your content (use the “Normal” paragraph style).