

# IMPERIAL

TITLE OF YOUR FYP/MSc REPORT

SUBTITLE IF NEEDED

Author

F.N. SURNAME

CID: 00000000

Supervised by

DR MARIOS KOGIAS

PROF. A. NOTHER

Advised by

KARL DER KRÜMELMONSTER (MSc)

A Thesis submitted in fulfillment of requirements for the degree of  
**Master of Engineering in Electronic and Information Engineering**

Department of Computing  
Imperial College London  
2025



# Abstract

The abstract is a very brief summary of the dissertation's contents. It should be about half a page long. Somebody unfamiliar with your project should have a good idea of what it's about having read the abstract alone and will know whether it will be of interest to them.



# Declaration of Originality

I hereby declare that the work presented in this thesis is my own unless otherwise stated. To the best of my knowledge the work is original and ideas developed in collaboration with others have been appropriately referenced.



# Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.





# Acknowledgements

Thank you to the kind souls at TUM who crafted this thesis template that I've modified for my own.



# Contents

<b>Abstract</b>	<b>i</b>
<b>Declaration of Originality</b>	<b>iii</b>
<b>Copyright Declaration</b>	<b>v</b>
<b>Acknowledgements</b>	<b>vii</b>
<b>List of Acronyms</b>	<b>xiii</b>
<b>List of Figures</b>	<b>xv</b>
<b>List of Tables</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem .....	1
1.2 Motivation .....	1
1.3 Objectives .....	2
1.4 Outline .....	2
<b>2 Background</b>	<b>3</b>
2.1 e.g. User Feedback .....	3
2.2 e.g. Representational State Transfer .....	3
2.3 e.g. Scrum .....	4
<b>3 Related Work</b>	<b>5</b>
<b>4 Requirements Analysis</b>	<b>7</b>
4.1 Overview .....	7
4.2 Current System .....	8
4.3 Proposed System .....	8

4.3.1 Functional Requirements .....	8
4.3.2 Nonfunctional Requirements .....	8
4.4 System Models .....	8
4.4.1 Scenarios .....	9
4.4.2 Use Case Model .....	9
4.4.3 Analysis Object Model .....	9
4.4.4 Dynamic Model .....	9
4.4.5 User Interface .....	10
<b>5 System Design</b>	<b>11</b>
5.1 Overview .....	11
5.2 Design Goals .....	12
5.3 Subsystem Decomposition .....	12
5.4 Hardware Software Mapping .....	12
5.5 Persistent Data Management .....	12
5.6 Access Control .....	12
5.7 Global Software Control .....	13
5.8 Boundry Conditions .....	13
<b>6 Case Study / Evaluation</b>	<b>15</b>
6.1 Design .....	15
6.2 Objectives .....	15
6.3 Results .....	16
6.4 Findings .....	16
6.5 Discussion .....	16
6.6 Limitations .....	16
<b>7 Summary</b>	<b>17</b>
7.1 Status .....	17
7.1.1 Realized Goals .....	17
7.1.2 Open Goals .....	18
7.2 Conclusion .....	18
7.3 Future Work .....	18

---

<b>Conclusions</b>	<b>19</b>
Future Directions .....	19
<b>A Supplementary Materials</b>	<b>21</b>
<b>B More Supplementary Materials</b>	<b>23</b>
<b>C Even More Supp Mats</b>	<b>25</b>
<b>Bibliography</b>	<b>27</b>



# List of Acronyms

<b>ACK</b>	Acknowledgement
<b>ASIC</b>	Application-Specific Integrated Circuit
<b>DMA</b>	Direct Memory Access
<b>FPGA</b>	Field-Programmable Gate Array
<b>NIC</b>	Network Interface Card
<b>SRPT</b>	Shortest Remaining Processing Time
<b>WAN</b>	Wide Area Network





# List of Figures



# List of Tables



# 1

## Introduction

### Contents

---

1.1 Problem .....	1
1.2 Motivation .....	1
1.3 Objectives .....	2
1.4 Outline .....	2

---

Introduce the topic of your thesis, e.g. with a little historical overview.

### 1.1 Problem

Describe the problem that you like to address in your thesis to show the importance of your work. Focus on the negative symptoms of the currently available solution.

### 1.2 Motivation

Motivate scientifically why solving this problem is necessary. What kind of benefits do we have by solving the problem?

### 1.3 Objectives

Describe the research goals and/or research questions and how you address them by summarizing what you want to achieve in your thesis, e.g. developing a system and then evaluating it.

### 1.4 Outline

Describe the outline of your thesis

# 2

## Background

### Contents

---

2.1 e.g. User Feedback .....	3
2.2 e.g. Representational State Transfer .....	3
2.3 e.g. Scrum .....	4

---

Describe each proven technology / concept shortly that is important to understand your thesis. Point out why it is interesting for your thesis. Make sure to incorporate references to important literature here.

### 2.1 e.g. User Feedback

This section would summarize the concept User Feedback using definitions, historical overviews and pointing out the most important aspects of User Feedback.

### 2.2 e.g. Representational State Transfer

This section would summarize the architectural style Representational State Transfer (REST) using definitions, historical overviews and pointing out the most important aspects of the architecture.

## 2.3 e.g. Scrum

2

This section would summarize the agile method Scrum using definitions, historical overviews and pointing out the most important aspects of Scrum.



# 3

## Related Work

Describe related work regarding your topic and emphasize your (scientific) contribution in contrast to existing approaches / concepts / workflows. Related work is usually current research by others and you defend yourself against the statement: “Why is your thesis relevant? The problem was already solved by XYZ.” If you have multiple related works, use subsections to separate them.



## 4

# Requirements Analysis

## Contents

---

<b>4.1 Overview</b>	<b>7</b>
<b>4.2 Current System</b>	<b>8</b>
<b>4.3 Proposed System</b>	<b>8</b>
4.3.1 Functional Requirements	8
4.3.2 Nonfunctional Requirements	8
<b>4.4 System Models</b>	<b>8</b>
4.4.1 Scenarios	9
4.4.2 Use Case Model	9
4.4.3 Analysis Object Model	9
4.4.4 Dynamic Model	9
4.4.5 User Interface	10

---

This chapter follows the Requirements Analysis Document Template in [1]. Important: Make sure that the whole chapter is independent of the chosen technology and development platform. The idea is that you illustrate concepts, taxonomies and relationships of the application domain independent of the solution domain! Cite [1] several times in this chapter.

## 4.1 Overview

Provide a short overview about the purpose, scope, objectives and success criteria of the system that you like to develop.

## 4.2 Current System

This section is only required if the proposed system (i.e. the system that you develop in the thesis) should replace an existing system.

## 4.3 Proposed System

If you leave out the section “Current system”, you can rename this section into “Requirements”.

### 4.3.1 Functional Requirements

List and describe all functional requirements of your system. Also mention requirements that you were not able to realize. The short title should be in the form “verb objective”

- FR1 Short Title: Short Description.
- FR2 Short Title: Short Description.
- FR3 Short Title: Short Description.

### 4.3.2 Nonfunctional Requirements

List and describe all nonfunctional requirements of your system. Also mention requirements that you were not able to realize. Categorize them using the FURPS+ model described in [1] without the category functionality that was already covered with the functional requirements.

- NFR1 Category: Short Description.
- NFR2 Category: Short Description.
- NFR3 Category: Short Description.

## 4.4 System Models

This section includes important system models for the requirements analysis.

#### 4.4.1 Scenarios

If you do not distinguish between visionary and demo scenarios, you can remove the two subsections below and list all scenarios here.

**Visionary Scenarios** Describe 1-2 visionary scenario here, i.e. a scenario that would perfectly solve your problem, even if it might not be realizable. Use free text description.

**Demo Scenarios** Describe 1-2 demo scenario here, i.e. a scenario that you can implement and demonstrate until the end of your thesis. Use free text description.

#### 4.4.2 Use Case Model

This subsection should contain a UML Use Case Diagram including roles and their use cases. You can use colors to indicate priorities. Think about splitting the diagram into multiple ones if you have more than 10 use cases. **Important:** Make sure to describe the most important use cases using the use case table template (`./tex/use-case-table.tex`). Also describe the rationale of the use case model, i.e. why you modeled it like you show it in the diagram.

#### 4.4.3 Analysis Object Model

This subsection should contain a UML Class Diagram showing the most important objects, attributes, methods and relations of your application domain including taxonomies using specification inheritance (see [1]). Do not insert objects, attributes or methods of the solution domain. **Important:** Make sure to describe the analysis object model thoroughly in the text so that readers are able to understand the diagram. Also write about the rationale how and why you modeled the concepts like this.

#### 4.4.4 Dynamic Model

This subsection should contain dynamic UML diagrams. These can be a UML state diagrams, UML communication diagrams or UML activity diagrams. **Important:** Make sure to describe the diagram and its rationale in the text. **Do not use UML sequence diagrams.**

#### 4.4.5 User Interface

# 4

Show mockups of the user interface of the software you develop and their connections / transitions. You can also create a storyboard. **Important:** Describe the mockups and their rationale in the text.

# 5

## System Design

### Contents

---

5.1 Overview .....	11
5.2 Design Goals .....	12
5.3 Subsystem Decomposition .....	12
5.4 Hardware Software Mapping .....	12
5.5 Persistent Data Management .....	12
5.6 Access Control .....	12
5.7 Global Software Control .....	13
5.8 Boundry Conditions .....	13

---

This chapter follows the System Design Document Template in [1]. You describe in this chapter how you map the concepts of the application domain to the solution domain. Some sections are optional, if they do not apply to your problem. Cite [1] several times in this chapter.

### 5.1 Overview

Provide a brief overview of the software architecture and references to other chapters (e.g. requirements analysis), references to existing systems, constraints impacting the software architecture..

## 5.2 Design Goals

Derive design goals from your nonfunctional requirements, prioritize them (as they might conflict with each other) and describe the rationale of your prioritization. Any trade-offs between design goals (e.g., build vs. buy, memory space vs. response time), and the rationale behind the specific solution should be described in this section

## 5.3 Subsystem Decomposition

Describe the architecture of your system by decomposing it into subsystems and the services provided by each subsystem. Use UML class diagrams including packages / components for each subsystem.

## 5.4 Hardware Software Mapping

This section describes how the subsystems are mapped onto existing hardware and software components. The description is accompanied by a UML deployment diagram. The existing components are often off-the-shelf components. If the components are distributed on different nodes, the network infrastructure and the protocols are also described.

## 5.5 Persistent Data Management

Optional section that describes how data is saved over the lifetime of the system and which data. Usually this is either done by saving data in structured files or in databases. If this is applicable for the thesis, describe the approach for persisting data here and show a UML class diagram how the entity objects are mapped to persistent storage. It contains a rationale of the selected storage scheme, file system or database, a description of the selected database and database administration issues.

## 5.6 Access Control

Optional section describing the access control and security issues based on the nonfunctional requirements in the requirements analysis. It also describes the implementation of the access matrix based on capabilities or access control lists, the selection of authentication mechanisms and the use of encryption algorithms.



## 5.7 Global Software Control

Optional section describing the control flow of the system, in particular, whether a monolithic, event-driven control flow or concurrent processes have been selected, how requests are initiated and specific synchronization issues

## 5.8 Boundry Conditions

Optional section describing the use cases how to start up the separate components of the system, how to shut them down, and what to do if a component or the system fails.



# 6

## Case Study / Evaluation

### Contents

---

6.1 Design .....	15
6.2 Objectives .....	15
6.3 Results .....	16
6.4 Findings .....	16
6.5 Discussion .....	16
6.6 Limitations .....	16

---

If you did an evaluation / case study, describe it here.

### 6.1 Design

Describe the design / methodology of the evaluation and why you did it like that. e.g. what kind of evaluation have you done (e.g. questionnaire, personal interviews, simulation, quantitative analysis of metrics), what kind of participants, what kind of questions, what was the procedure?

### 6.2 Objectives

Derive concrete objectives / hypotheses for this evaluation from the general ones in the introduction.

## 6

### 6.3 Results

Summarize the most interesting results of your evaluation (without interpretation). Additional results can be put into the appendix.

### 6.4 Findings

Interpret the results and conclude interesting findings

### 6.5 Discussion

Discuss the findings in more detail and also review possible disadvantages that you found

### 6.6 Limitations

Describe limitations and threats to validity of your evaluation, e.g. reliability, generalizability, selection bias, researcher bias

# 7

## Summary

### Contents

---

<b>7.1 Status .....</b>	<b>17</b>
7.1.1 Realized Goals .....	17
7.1.2 Open Goals .....	18
<b>7.2 Conclusion .....</b>	<b>18</b>
<b>7.3 Future Work .....</b>	<b>18</b>

---

This chapter includes the status of your thesis, a conclusion and an outlook about future work.

### 7.1 Status

Describe honestly the achieved goals (e.g. the well implemented and tested use cases) and the open goals here. if you only have achieved goals, you did something wrong in your analysis.

#### 7.1.1 Realized Goals

Summarize the achieved goals by repeating the realized requirements or use cases stating how you realized them.

## 7

### 7.1.2 Open Goals

Summarize the open goals by repeating the open requirements or use cases and explaining why you were not able to achieve them. Important: It might be suspicious, if you do not have open goals. This usually indicates that you did not thoroughly analyze your problems.

## 7.2 Conclusion

Recap shortly which problem you solved in your thesis and discuss your **contributions** here.

## 7.3 Future Work

Tell us the next steps (that you would do if you have more time). Be creative, visionary and open-minded here.

# Conclusions

All good projects conclude with an objective evaluation of the project's successes and failures and suggestions for future work which can take the project further. It is important to understand that there is no such thing as a perfect project.

Even the very best pieces of work have their limitations and you are expected to provide a proper critical appraisal of what you have done. Your assessors are bound to spot the limitations of your work and you are expected to be able to do the same.

## Future Directions

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua quaerat voluptatem. Ut enim aequaleam animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.







## Supplementary Materials

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequale doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

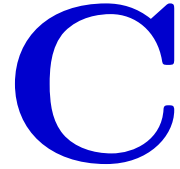
A

## B

## More Supplementary Materials

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequale doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

**B**



## Even More Supp Mats

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequale doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.



# Bibliography

- [1] B. Bruegge and A. H. Dutoit, *Object Oriented Software Engineering Using UML, Patterns, and Java*. Prentice Hall, 2009.