KING'S College LONDON

Faculty of Natural, Mathematical, and Engineering Sciences

Department of Informatics

Software Engineering Group Project

Insert project title here

Team Name Here
Author One
Author Two
Author Three
Author Four
Author Five
Author Six
Author Seven
Author Eight

February 4, 2025

Contents

Chapter 1

Introduction

Instructions!

Please remove all expectations and assessment environments from this page before submitting, including this instruction. This is placeholder content to help you write the report. It must not be included in the final report.

Expectations

A good introduction gives the reader a good idea of what the project is about in the shortest amount of text. A reader should be able to get through this chapter very quickly and decide whether this project is relevant to their interests.

- A summary of the overriding business objective of the application. Explain what the client, the end-users of your application, or the organisation deploying it would gain from the application. Explain this value from the perspective of the client's or users' world, not from your perspective as a developer. Do not list features, requirements, or specifications here. The software is a means to an end, not the objective in itself.
- A statement of the type of system you built, including the platforms on which it can be used, and the technology stack used to build it.

This chapter does not require any sections of subsections. In many projects, two short paragraphs will be enough. However, you can use more if the objective cannot be explained in a single paragraph. Note, though, that you will be developing the objectives further in the next chapter.

Length requirements

A single page should be sufficient for this chapter. More than two pages is too long.

Chapter 2

Objectives and stakeholders

Instructions!

Please remove all expectations and assessment environments from this page before submitting, including this instruction. This is placeholder content to help you write the report. It must not be included in the final report.

Length requirements

There are no strict length requirements for this chapter. However, it is important that the writing is clear and concise. Avoid repeating yourself. It is expected that a typical project needs 1–3 pages, but this can vary considerably from project to project.

Expectations

This is a short chapter that identifies the objectives and stakeholders of the project. A good report also discusses how the different types of stakeholder are affected by the project or how they could be affected by the project.
As you explain the project objectives, focus exclusively on what value the system adds to the stakeholders. Your objectives should justify *why* you are building the system. The software requirements are *not* objectives. The software requirements explain *what* your system can do. In other words, they are a way to achieve objectives. Software requirements do *not* belong in this chapter. Your project should have at least one objective. Otherwise, the project is a pointless exercise. Many objectives do not necessarily make for a better project.
Some stakeholders of a project will be obvious: e.g. the people who the beneficiaries of the value your project adds. However, it is important to be aware of all stakeholders. The stakeholder D.A.N.C.E. tool to identify all stakeholders and it is explained in a video on project initiation. Consider carefully how each stakeholder is affected by your project and who the key stakeholders are.
The section headers below provide a suggested structure. You should feel free to change it or extend it.

2.1 Project objectives

2.2 Stakeholder

2.2.1 Stakeholder analysis

2.2.2 Key stakeholders

Chapter 3

Specifications

Instructions!

Please remove all expectations and assessment environments from this page before submitting, including this instruction. This is placeholder content to help you write the report. It must not be included in the final report.

Length requirements

There are no strict length requirements for this chapter. However, it is important that the writing is clear and concise. Avoid repeating yourself. It is expected that a typical project needs 1–5 pages, but this can vary considerably from project to project.

Expectations

This chapter should provide the specifications of the software that you have implemented. Good specifications are clear, concise, consistent (with one another and with the project objectives), estimable (the effort required of functional requirements should be estimable), and testable (you should be able to create automated or manual tests for these specifications).
A significant part of this chapter covers functional requirements/specifications. These refer to the product features your team has developed. Present a clear overview of all functional requirements/specifications that can be found in the deployed system. For each feature, explain how it contributes to the project objectives, and explain how/where it can be found. Do *not* mix the specifications the team managed to implement with those they did not.
If necessary or helpful for clarity, explain what is out of scope of the project in a distinct section. You may discuss ideas for functional specifications the team did not manage to implement, provided this discussion is clearly distinct from the project's achieved functional specifications.
The project objectives will also entail certain non-functional specifications. These may include, for example, reliability, usability, portability, scalability, performance, compatibility, security, compliance, etc. Identify which non-functional specifications are important to this project and define them. Make sure to present them in order of importance, and relate each one to the project objectives.
The section headers below provide a suggested structure. You should feel free to change it or extend it.

3.1 Functional specifications

3.2 Non-functional specifications

Chapter 4

Project management

Instructions!

Please remove all expectations and assessment environments from this page before submitting, including this instruction. This is placeholder content to help you write the report. It must not be included in the final report.

Length requirements

There are no strict length requirements for this chapter. However, it is important that the writing is clear and concise. Avoid repeating yourself. Make sure to clearly signpost evidence for claims. It should be possible for a reader to understand this chapter without consulting other sources. It is expected that a typical project needs 3–6 pages, but this can vary considerably from project to project.

Expectations

In this chapter, the team must present, reflect on, and evaluate the team's approaches to project management. This discussion should present the main approaches to project management the team decided to use. Include the team's rationale for these decisions, and the team's observations on the effectiveness of each approach.

During the project, the team may have made significant changes to project management approaches, or made new decisions where initial arrangements were insufficient. A typical team will also have encountered difficulties, such as risks that materialised and impacted the team. Discuss the most critical decisions the team made, why you made them, and what the results were. If certain approaches were not entirely successful, discuss what you would do differently in a future project.

A good report shows a systematic approach to project management, justifications for key decisions, and a critical reflection on the effectiveness of different approaches. Ideally, the justifications and reflections in this chapter are based on project management theory.

Note that "a systematic approach to project management", "critical reflection", and use of "project management theory" do not necessary imply that your team's project management approach is a success story. Thoughful, considered project management improves your chances of success, but projects do not always go according to plan and team's make mistakes. The markers are looking for

- Good justifications for approaching a project in a particular way.
- Critical reflection on the results, including recognition of problems.
- The team taking ownership of the problems they encountered.

There is no one best way of organising this content. One approach is to cover different phases of the project management lifecycle (from initialisation to project close). Another approach is to cover different project management tasks (e.g. team management, scheduling, planning, risk, etc.). Yet another approach is to cover the most impactful events. Each project is different and you should take the approach that works best for your team.

There is also no minimum or maximum number of issues to cover. Cover what is more important in the depth necessary to do the topic justice.

Chapter 5

Design and implementation

labelchap:design-and-implementation

Instructions!

Please remove all expectations and assessment environments from this page before submitting, including this instruction. This is placeholder content to help you write the report. It must not be included in the final report.

Length requirements

There are no strict length requirements for this chapter. However, it is important that the writing is clear and concise. Avoid repeating yourself. Make sure to clearly signpost evidence for claims. It should be possible for a reader to understand this chapter without consulting other sources. It is expected that a typical project needs 3–6 pages, but this can vary considerably from project to project.

Expectations

In this chapter, the team should discuss the key design and implementation decisions made during the project, and how these are reflected in the end product. Design refers to the way the software system is organised into smaller components and the way in which they are related. We are *not* looking for a discussion of a series of screenshots of your application.

Normally, all teams should present the overall architecture of the software. The remainder of the chapter will depend on the type of system that is built. To decide what to cover, consider the following to aspects of the project:

- What design and implementation decisions did you make to improve software quality?
- What design and implementation decisions did you make to achieve challenging functional and non-functional specifications

Depending on your answers to these questions, you may cover the design an implementation specific components, interfaces, the overall class structure, the database design, algorithms, business processes, etc.

A good report considers alternative options for key decisions and includes sound justifications for the decisions made. If applicable, you may reflect on changes made during the project. Ideally, key decisions are rooted in Software Engineering theory.

5.1 Architecture

Expectations

Add other sections and sub-sections as necessary.

Chapter 6

Testing

Instructions!

Please remove all expectations and assessment environments from this page before submitting, including this instruction. This is placeholder content to help you write the report. It must not be included in the final report.

Length requirements

There are no strict length requirements for this chapter. However, it is important that the writing is clear and concise. Avoid repeating yourself. Make sure to clearly signpost evidence for claims. It is expected that a typical project needs 1–5 pages, but this can vary considerably from project to project.

Expectations

This chapter discusses how the software has been tested.
Start by identifying the different ways the team has used to test the software, including the tools used in each approach. Below, a section title "Approach and tools" is provided a suggestion for a header under which to cover this. The purpose of this section is to help the reader understand what testing is included in the software. For each distinct testing approach, clearly identify its purpose, and the extent to which you relied on it. Also, identify where in the source code or the report the reader can find evidence of each type of testing.
The team may have used certain processes to ensure the quality of testing. If so, explain what these were and how effective these approaches were. If you decide to discuss this, it is important to identify evidence of the processes you employed. For example, you could identify minuted meeting decisions that clearly demonstrate your team's application of a particular appraoch. Unsupported claims tend not to be so convincing.
A good report includes a critical evaluation covering both strengths and weaknesses. Claims made in this evaluation should also be justified with evidence. If your team has code coverage results, make sure to present summary results in the report, but identify the location of the source report (typically HTML pages) in the source code. Again, unsupported claims remain unconvincing.
If your team used manual testing as well as automated testing, it is important to include the following:

- What did your rely on manual testing for? In other words, what was the purpose of manual testing.
- To what extent did you rely on manual testing for aspects of software testing that would ideally be automated.
- What are the respective limits of automated and manual testing.
- An appendix listing each manual test. For each manual test, describe precisely the actions the tester must undertake and what they are expected to observe. Also add when the test was last performed and whether it was successful.

6.1 Approach and tools

6.2 Quality assurance processes

6.3 Evaluation of testing

Chapter 7

Machine learning

Instructions!

Please remove all expectations and assessment environments from this page before submitting, including this instruction. This is placeholder content to help you write the report. It must not be included in the final report.

Instructions!

By default, this chapter is not included in the report because most teams do *not* need this chapter. If your project requires this chapter, uncomment the `input` command that includes this chapter.

Expectations

If your project includes training a machine learning model, and you made an effort to evaluate alternative approaches, teams should include this chapter discussing the machine learning work. It is recommended to include a short summary description for each considered approach, a clear description and rationale of the approach taken to evaluate and compare the different algorithms, a table summarising the results obtained in evaluation, and your team's decision on what algorithm was chosen for incorporation in the system. Also explain how the team's results can be replicated. This chapter is not required if an existing machine learning model or tool was used, or if the team did not evaluate the machine learning component.

Appendix A

Manual tests

Instructions!

Please remove all expectations and assessment environments from this page before submitting, including this instruction. This is placeholder content to help you write the report. It must not be included in the final report.

Instructions!

By default, this chapter is not included in the report because many teams do *not* need this chapter. If your project requires this chapter, uncomment the `input` command that includes this chapter.

Expectations

If your team relied on manual testing, and the manual testing was carefully planned such that the team has a detailed written specification of its tests, then this chapter should be used to provide a list of all the team's tests. A template table is provided to help format the tests.

#	Test script	Expected outcome	Outcome/date
1	Specify the actions the tester must undertake to run the test.	Specify what the tester must observe.	