Project Handover Document

**Company ----**

Project -----

*Trimester 2, 2022*

Table of Contents

[1. Project Information 3](#_Toc113485150)

[1.1. Company Acting Director 3](#_Toc113485151)

[1.2. Project Team 3](#_Toc113485152)

[2. Project Overview 3](#_Toc113485153)

[3. User Manual 3](#_Toc113485154)

[4. Completed Deliverables 4](#_Toc113485155)

[5. Roadmap 4](#_Toc113485156)

[6. Open Issues 4](#_Toc113485157)

[7. Lessons Learned 5](#_Toc113485158)

[8. Product Development Life Cycle 5](#_Toc113485159)

[8.1. New Tasks 5](#_Toc113485160)

[8.2. Definition of Done 5](#_Toc113485161)

[8.3. Task Review 6](#_Toc113485162)

[8.4. Testing 6](#_Toc113485163)

[8.5. Branching Strategy 6](#_Toc113485164)

[9. Product Architecture 6](#_Toc113485165)

[9.1. UML Diagram 6](#_Toc113485166)

[9.2. Tech Stack 6](#_Toc113485167)

[10. Source Code 7](#_Toc113485168)

[11. Login Credentials 7](#_Toc113485169)

[12. Other Relevant Information 7](#_Toc113485170)

[13. Appendices 7](#_Toc113485171)

# Project Information

## Company Acting Director

**Name**

*Title*

Email

List multiple if applicable.

## Project Team

**Project Name**

Company

# Project Overview

Give an overview or ‘executive summary’ of the project, including any necessary high-level information for someone reading about your work for the first time. The following questions should help guide your thinking.

What is the project about?

What problem is the project solve?

What are the aims of the project?

What are the deliverables?

# User Manual

Give instructions for how someone should use your product or navigate around your development environment. Include images, diagrams, or anything that would help a first-time user to use your product correctly.

Better yet, you could create short instructional videos using software like [Loom](https://www.loom.com/) and include the video links in this section. You may find this option is considerably easier than trying to communicate your instructions through text! (Note: this is just a suggestion, it’s not mandatory.) Here are some ideas of what to cover:

If your product currently requires a complex set of steps to activate, include that.

If your product has a hardware component, explain how to activate and sync the hardware with the software.

If your team has a user experience journey that they’ve mapped out for when a user navigates your product, run through a demo of that.

# Completed Deliverables

Provide a list of product features and/or deliverables, including a brief description, that have been completed this trimester. Please relate these deliverables to their corresponding Trello cards if this is possible.

Only include features and/or deliverables that are fully complete – incomplete work is to be listed in section 4. Roadmap.

Make sure to explicitly highlight which features and/or deliverables where completed this Trimester and which team member(s) were primarily responsible for their completion.

Also, please indicate where each of the completed deliverables can be found (E.g., MS Teams, GitHub repository) and make sure to include a URL link to the resource.

# Roadmap

Provide a list of features and/or deliverables that are planned to be completed in the project’s future (E.g., next trimester or two trimesters in the future).

Please also include features and/or deliverables that are in progress but not yet complete. The state of each incomplete work item should be briefly described.

This section should pair up perfectly with your Roadmap on Trello. Make sure both this section and your Trello Roadmap are updated upon handing over the project.

# Open Issues

List all of the issues and challenges that the team is still facing, and any progress that has been made so far to address them.

The purpose of this section is to flag things that may interfere with the future teams’s ability to work on the project, and to give advice as to how these issues could be fixed in future.

Here are some examples of Open Issues:

* Software compatibility issues that arise when members of the team use different version of software.
* An unclear process for reviewing completed tasks on Trello, leading to a backlog of work that is sitting somewhere between unfinished and finished.
* An essential team member had to leave the team with no notice, and there is currently a skill void in their place.

# Lessons Learned

List key lessons learned from the project this Trimester and what you recommend future teams should do differently. You must also explain why you believe this to be the case.

In particular, try to think about processes or technology that you would recommend be changed in the future; things that an uniformed team may mistake for a good idea at first, but later learn to be ineffective.

For example, maybe your team had challenges communicating their progress during panel presentations, but towards the end of the Trimester, you developed an effective method for conveying progress accurately. This would be a great thing to talk about.

# Product Development Life Cycle

This section should explain how your team undertakes work. It is an attempt to codify your processes so that they can be understood and followed by new members.

As a team, you may not have clearly defined your Product Development Life Cycle, and that’s okay! This is an excellent opportunity to explain the work methods, processes and habits that your team has been developing intuitively over the course of the Trimester.

## New Tasks

How are new tasks created?

How does your team form new ideas about work that needs to be done and turn those ideas into distinct, actionable tasks?

For example, maybe your team meets at the start of each week, reviews your progress in your current sprint, makes a big long list of everything to be done, and then converts that list into a series of cards on Trello. This process would be something you talk about in this section.

## Definition of Done

How does the team know when a task is done?

What are criteria for a successfully completed task?

This may seem obvious, but it in a software development project having a definition of done can ensure a certain standard of work that holds all team members accountable. For example, messy, clunky code that “just works” is very different to clean, well-commented code that works AND is easy to understand. Which would you prefer to be your team’s definition of done?

## Task Review

Who reviews a task once it’s been marked as done?

How does the team ensure that all work is looked over before it’s contributed to the main repository or working prototype?

If you don’t currently have a system for reviewing tasks, make sure to flag this for next trimester’s team to work on as soon as they begin.

## Testing

How do you test your product to see if it does what it was originally planned to do?

If your product isn’t heavily comprised of software, how can you build in testing to your team’s product development life cycle to ensure that “stuff works as it should”?

## Branching Strategy

How does your team currently use GitHub repository?

What rules for commits and pull-requests have been put in place so far?

How should new members use GitHub repository in a way that doesn’t result in all commits being dumped in a messy Master branch?

Again, if your team hasn’t formally discussed a branching strategy, this a great opportunity to describe what your current system is and how it could be improved going forward.

For example, if you currently have all members of the team commit directly to the Master branch, can you recommend any tutorials for the future team to review that might lead to a cleaner, more organised and more efficient repository?

# Product Architecture

## UML Diagram

Provide a high-level map of the project showing all of its components and how they relate to each other.

An example of this is a UML diagram. Don’t feel that you need to follow any particular UML paradigm, so long as your diagram is informative and easy to read.

Resources like [Lucidchart](https://www.lucidchart.com/) and [Draw.io](https://app.diagrams.net/) are incredibly useful for this.

## Tech Stack

List all of the software and hardware utilised in this project. For each tool, give a short description and explain why it was chosen.

# Source Code

All source code should be found on your team’s GitHub repository, unless your project has unique constraints that require you to store your code elsewhere. This includes any resources (e.g., wireframes, designs) that need to be transferred over to the new team as well.

Please provide all of the necessary instructions to accessing your source code. This includes URLs of online hosted repositories, links to any software dependencies, database components, or external libraries.

If your code is hosted on a server external to Deakin, make sure to also transfer digital copies of your code over to your client and the next team as a backup.

# Login Credentials

Please provide all credentials (usernames and passwords) for any of the resources, websites, or platforms being utilised for this project. Please make sure that none of these credentials share passwords or usernames with any of your team’s private credentials.

# Other Relevant Information

This section is an invitation to add any additional information that you think will help to onboard new members. If you choose not to add any extra sections to this document, this section should be deleted.

Please edit this entire document as you see fit. If you think adding 5 extra sections that aren’t listed here will help to communicate the nuances of your project to future members, go ahead! We want you to take full ownership of your handover and this document.

# Appendices

Include all relevant artefacts delivered during the course of the project. Anything that will paint a clearer picture of your team’s progress this trimester, the things that informed decisions, and the evolution of your product.

Please also include a link to your team’s showcase video.