3/30/2023

Jadcup Digital Twin: Factory Automation

PROJECT PROPOSAL V4

PRESENTED BY: YERAN EDMONDS, MYLES HOSKEN, JANE JUNG, JOSHUA LADOWSKY, HARSHIL PATEL

CLIENT: DANIEL LI

MENTOR: MATTHEW KUO

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Scope: The project scope focuses on integrating Internet of Things (IoT) technology, cloud computing, and the development of a factory simulation for Jadcup's manufacturing process. These key elements will be used to connect machines, optimize production, and provide efficient data storage and management. The factory simulation will be done via a digital twin to replicate the manufacturing process, allowing the team to test potential enhancements before implementation. This project will deliver a functional prototype of a user-friendly Jadcup monitoring application that centralizes PLC data and promotes horizontal scaling in the manufacturing process. The goal is to modernize Jadcup's operations and usher in Industry 4.0 capabilities, elevating the company's position within the sustainable cup manufacturing sector.

Method: The chosen methodology for this project is Agile, specifically Kanban. Kanban is a well-suited methodology as we currently have unclear requirements that will become evident as development goes on. *For Further details, refer to Section 6: Project Management Methodology.*

Risks:

- Unclear Requirements
- Incorrect work performed.
- Scope Creep
- Mismanaged feature prioritization.
- Poor product design.
- Poor communication with the client.
- Poor time management.
- Workplace injury (Health and Safety).
- Poor communication with other members of the team.
- Team member unavailability.
- Lockdown.
- Loss of data.
- Out of scope requirements.

Cost: $13,241.41

SECTION 1: TERMS OF REFERENCE

Jadcup is a major manufacturer of sustainable and compostable disposable cups in New Zealand.
Daniel Li, the founder of Jadcup, has identified that the current manufacturing process has room for improvement. In the existing process of cup manufacturing, raw materials such as cardboard are fed into the machines, which then produce the finished cups through a series of automated steps, such as moulding, trimming, and printing. While the process of cup manufacturing is currently automated, the monitoring and adjusting of the process can only be done using a built-in monitor connected to the machines. This limits the ability to monitor and optimize the process in real time, which can lead to inefficiencies, production downtime, and potential quality issues. This production process is known as Industry 3.0. In Industry 3.0, the process is automated using Programmable Logic Controllers (PLCs) and information technology. This process can often operate largely without human interference, but a human aspect is still required to repeat manual labour and change parameters through the machines interface. In response to the need for a more efficient and modernized approach to the cup manufacturing process, Daniel has reached out to our team for assistance in addressing the gap for improvement and exploring the possibilities of incorporating advanced monitoring and control systems, such as Industry 4.0 solutions.
Our team consists of:

- Joshua Ladowsky (Service request manager/developer)
- Myles Hosken (Service delivery manager/developer)
- Jane Jung (Developer)
- Yeran Edmonds (Developer)
- Harshil Patel (Developer)

We are supported by the guidance and mentorship of:

- Matthew Kuo (Mentor)
- Tony Clear (Moderator)
- Professor Jacqui Whalley (Course leader)
- Dr Ramesh Lal (Course leader)

Together, we aim to develop and implement a remote monitoring system for the manufacturing process that will facilitate real-time monitoring, data analysis, and optimization. This system will enhance productivity, reduce downtime, and uphold high-quality standards in the production of sustainable and compostable disposable cups. By enabling Jadcup to promptly adjust parameters and detect potential issues, the remote monitoring system will help prevent major problems. The implementation of this system will elevate Jadcup's standing to Industry 4.0, also known as the fourth industrial revolution (Schwab, 2016). In Industry 4.0, processes are automated based on data rather than human input. To achieve this advancement, Jadcup will incorporate various technologies in conjunction with the remote monitoring system.

- **Internet of things (IoT):**
  The Internet of things has been recognized as a key component in smart factories. Each machine can be installed with a low-cost sensor that houses an IP address. This address allows the machines to connect to internet-enabled devices. This connectivity allows large amounts of data and analytics to be collected and analysed. (IBM, n.d.).
- **Cloud Computing:**
  Cloud computing is a key component for an Industry 4.0 strategy. Cloud computing makes it possible to store large amounts of data to be processed more efficiently and cost-effectively. To unlock the full capabilities that smart manufacturing has to offer, it is necessary to connect aspects of the business such as, engineering, supply chain, production, sales, and distribution. The use of cloud computing can facilitate this integration. Jadcup will use cloud computing to host a server which will allow machines to talk to one another within the network, along with having a database to store login credentials.
- **Digital Twin:**
  In this project we will implement a digital twin of Jadcup's manufacturing process. A digital twin is created by simulating a factory floor and pulling data from sensors, devices, and the current PLCs. Using factory simulation software such as Factory I/O, we can simulate the production process to create workflows and test changes to improve capacity and minimize downtime.

Within the project scope, the integration of Internet of Things (IoT) technology, cloud computing, and the development of a digital twin for Jadcup's manufacturing process will play key roles. IoT will be used to connect machines, enabling the collection and analysis of extensive data, which will help identify areas for improvement and optimize the production process. Cloud computing will be employed to provide efficient and economical data storage and management, allowing for seamless data access and processing across different aspects of the project. The digital twin, constructed using 3D modelling software, will be utilized to simulate the production process. This simulation will enable the team to test potential enhancements, such as increasing capacity and reducing downtime, before implementing them in the real-world manufacturing environment.

The project will not involve a complete overhaul of the manufacturing process or the replacement of existing machinery. Instead, the focus will be on leveraging these key technologies to augment the current system, improving monitoring and control mechanisms such as running speed, inching speed, actual speed, and temperatures. Enhancing control over these parameters will guide Jadcup's transition toward Industry 4.0. *See appendix 1.1 for the current PLC display at Jadcup.*

SECTION 2: PROJECT RATIONALE

By investing in this project, Daniel Li and Jadcup are demonstrating their commitment to continuously improving their product quality, enhancing customer satisfaction, and maintaining their position as a leader in sustainable and compostable disposable cups in New Zealand. The transition from Industry 3.0 to Industry 4.0 is a significant advancement for Jadcup and will provide a several benefits for the business:

- **Efficiency and productivity:**
  Data-driven automation through the Internet of Things will allow Jadcup to streamline its operations, automate repetitive tasks, and fully optimize its processes.
- **Improved Quality:**
  Industry 4.0 technologies can detect and resolve quality issues in real-time, resulting in higher production quality by detecting issues before they become a problem.
- **Cost saving:**
  Automating processes under Industry 4.0 can allow Jadcup to reduce labour costs, minimize production downtime, and reduce waste all leading to cost savings. With data driving the production process, there will be less need for workers to be in the factory.
- **Enhanced Flexibility:**
  Having access to Industry 4.0 technologies will ensure that Jadcup can respond quickly to changing market demands and customer needs by having the capability to change input and output production streams along with reconfiguration and customization faster.
- **Increased Safety:**
  A higher stand of automation can help reduce the risk of workplace accidents ensuring the safety of employees.
- **Enhanced Data Collection:**
  Jadcup under Industry 4.0 will have access to real-time data on their operations, giving them control and information to analyse performance and make informed decisions to improve their process.
- **Scalability:**
  Once the remote monitoring system works for one machine, with the use of the Digital Twin and Industry 4.0, when the time comes for Jadcup to expand, the system will be modular and decoupled. Allowing Jadcup to expand effectively without many blockers.
- **Custom Manufacturing:**
  A big part of the transition from Industry 3.0 to Industry 4.0 is the removal of "minimum batch requirements". Industry 3.0 is all about mass production, utilizing the Digital Twin, Jadcup can easily create small batches of specialized orders for customers. Reducing waste and meeting customer needs in a modern way.

The key contribution of this project is to enhance Jadcup's dedication to continuous improvement, customer satisfaction, and leadership in sustainable and disposable cup production within New Zealand. The introduction of the remote monitoring system and Industry 4.0 technologies will play key roles in realizing these advantages. The transition to Industry 4.0 will enable Jadcup to efficiently monitor the manufacturing process and respond to any issues or faults in real-time. This will drive faster production and minimize material wastage by reducing downtime and manufacturing errors. In addition to real-time monitoring, the incorporation of Industry 4.0 technologies offers other significant benefits. Cloud computing will facilitate the

monitoring of the production line in any locale, and the modular decoupled design of the system guarantees scalability, meaning Jadcup can grow effectively without hitting significant roadblocks. To further enhance Jadcup's manufacturing capabilities, the Industry 4.0 Digital Twin will enable Jadcup to test and safely prototype along with optimizing key manufacturing processes before implementing them in the factory. Furthermore, the shift to Industry 4.0 will equip Jadcup with the tools to respond to the changing needs of their customers and solidify their place as a leader in the field.

SECTION 3: PROJECT OBJECTIVE AND SCOPE

Project Objective & Purpose

The objective of this project is to develop a prototype for a remote monitoring system that allows Jadcup to connect and interact with programmable logic controllers (PLCs).

The purpose of this prototype is to determine the viability of a monitoring interface to manage PLCs on the Jadcup factory floor.

Project Scope

The Jadcup monitoring application is a limited, functional prototype to read and control PLC data through a user-friendly application. Local cup manufacturer Jadcup seeks to modernize their business by integrating their PLCs into a single, secure interface for easy access to their factory floor. By bringing industrial IoT to the company, they hope to bring horizontal production scaling to their cup manufacturing process and launch their business into Industry 4. The proposed dashboard will take the form of a web application to allow viewing on several different devices.

Functional Requirements

- The application must allow users to connect & control PLCs.
- The application must show data managed by PLCs.
- The application must be modular enough to support multiple, different PLCs.
- The application must provide secure user authentication and authorization.
- The application must have a simple interface designed with large buttons and easy-to-read data.
- The application must be paired with a working factory simulation that demonstrates the monitoring application's functionality.

Non-functional requirements

- The application must be responsive and have a fast load time, with a maximum response time of 100 milliseconds.
- The application must be functional and user-friendly when run on a PC.
- The application must be accessible on a machine within the office intranet, with a connection to the internet to interface with cloud services necessary to run the application.
- The application must be secure and follow the three information security pillars of CIA (confidentiality, integrity, availability).

Further reading: See 3.1 Scope Statement document in the appendix.

Key stakeholders

- The primary stakeholder for this project is Jadcup who will use the monitoring application to connect and interact with PLCs.

- The project team, team mentor, team mediator and paper leads are also stakeholders in this project.

Further reading: See 3.2 Stakeholder Management Strategy in the appendix.

Key risks and mitigation strategies

- Technical upskilling redundancy: To mitigate the risk of team members unable to participate or unable to progress, each skill required for the prototype is being upskilled by two developers.
- Managed time efficiency: To mitigate time scheduling constraints, the team have concise meeting agendas and action items with set deadlines. Communication is also strong within the team in the case that an item cannot be completed.
- Minimum-viable product: To mitigate risks with unclear feature requirements & 'scope creep', the team have collaborated with the client and the mentor to identify minimum specifications to deliver an accurate prototype.

Further reading: See Appendix 9.1 Risk Register, 9.2 Issue Log and 9.3, Quality Assurance Plan

SECTION 4: PROJECT TOOLS AND TECHNOLOGIES

Project Technology Architecture

The baseline software infrastructure for the monitoring application prototype will need to be supported by PCs and have the capability to connect to PLCs.

Options Considered:

- **FERN** Stack (Firebase, ExpressJS, ReactJS, NodeJS)
  A modern tech stack that uses industry technologies like React to deploy a single-page web application. Provides support for authentication but would require upskilling in technologies less familiar to the team and implementing a server and web API.
- **Java** Desktop Application
  A simple application that would leverage the team's prior knowledge in Java and Swing libraries. Limited in support for web applications, interfacing with PLCs and would need to be compiled and run on individual machines – not very accessible.
- **LAMP** Stack (Linux, Apache, MySQL, PHP)
  The classic technology stack for building dynamic websites. Proven, widely used technology that would suit an always-online application that would require upskilling in Linux hosting and server cybersecurity.
- **MEAN** stack (MongoDB, ExpressJS, AngularJS, NodeJS)
  Popular modern tech stack like FERN that uses a full-stack JavaScript framework. Would require similar upskilling to FERN but Firebase has better support for authentication and Angular is becoming phased out.

Further reading: See 3.3 Identified Technical Architecture in the Appendix.

Project Technology – Selection & Rationale

Given the requirements, technical capability, and team interest, we have chosen to use the **FERN** stack for our application. FERN provides a database layer, server-side layer, front-end layer, and scripting stack that work together to provide an efficient development environment for a local web application.

- **Single-language tech stack**
  With Javascript as the sole language for the application, developers can more easily switch between the different application layers for rapid implementation. This also provides redundancy should one developer be unable to participate or want a second opinion and reduces our upskilling time.
- **Flexibility**
  ReactJS is a flexible framework with multiple components and libraries allows for a highly customizable interface to suit the client's needs. In addition, there are Javascript libraries like NodeRED that exist to support PLC configuration.
- **Community Support**
  Javascript and React have large active developer communities that provide a wealth of resources for upskilling. This also supports app flexibility in the case that libraries or components we require don't meet our expected needs.
- **Cost-Effective**
  FERN stack is open-source and free, making it cost-effect for a university project. The scripting and server layers are completely free, and our database layer has free read/write access up to a specific number of records, which will be more than enough for this prototype.

Limitations & Constraints:

It is currently beyond-scope for our MVP to directly interface with Jadcup's PLCs while hooked into their cup-manufacturing machines. While this will allow for rapid iteration, it is possible that the prototype will need post-project support to work with Jadcup devices.

Technology provided by AUT:

- FactoryIO service keys
- Schneider Modicon M221

SECTION 5: SKILLS ANALYSIS

The development of the project deliverables requires the combination of several components, each of which involves knowledge and skills from different fields of software development.

The table below identifies the skills required for the project, as well as the skills missing that members of the team need to upskill before the development stage of the project can begin.
For details of individual skill levels see Appendix 5.1: Skill Matrix.

Required Skills	Missing Skills

• Application Server
  ○ Online Servers
  ○ Server Administration
• Component Technology
  ○ PLC
  ○ Factory I/O
• Database
  ○ NoSQL
• Language/Development tools
  ○ C#
  ○ Git
  ○ HTML/CSS
  ○ Java
  ○ NodeRED
	• JavaScript/React
	• PLC
	• Factory I/O
	• Web API (Factory I/O)
	• NodeRED/SCADA
	• Databases

• Multimedia
  ○ UI Design
  ○ UX Design
• Project Management
  ○ Agile Methodology
  ○ Client Management
• Project Management tools
  ○ Github
  ○ Teams
  ○ Trello
• QA Expertise
  ○ QA Tools
  ○ Unit Testing
• Web Technology
  ○ Javascript
  ○ NodeJS
  ○ PHP
  ○ React
  ○ Visual Studio
  ○ VS Code
  ○ Web APIs	

The upskilling phase for missing skills will commence immediately after the presentation of the proposal (at latest 7th of April) and will be no longer than 2 weeks. During this time, everyone in the team will be researching and familiarizing themselves with the relevant skills and tools. Once the upskilling phase is finished, team members are expected to have gained the relevant skills and abilities which can be applied to efficiently develop high-quality outcomes for the project.

For details on the Training Plan see Appendix 5.3
For details on Scheduled Upskilling see Appendix 5.3

SECTION 6: PROJECT MANAGEMENT METHODOLOGY

PROJECT METHODOLOGY

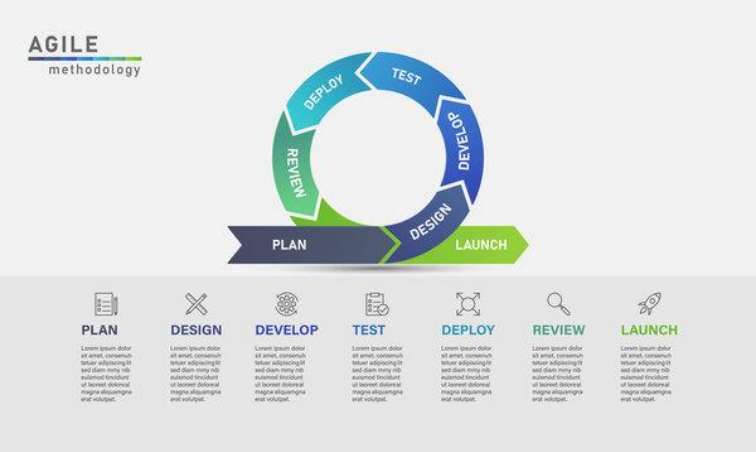


Figure 1. (Ztaro, 2020)

We have decided to adopt the Agile methodology when we develop our project. Agile is an iterative approach to project development that involves breaking down tasks into subtasks and completing them in short, rapid phases (atlassian, n.d.). This framework is highly flexible and enables us to make changes to the project as needed without causing major disruptions to the overall schedule.

We chose Agile over more traditional methods such as Waterfall as waterfall relies on a sequential process for completing tasks, where project phases can only begin after the previous phase ends (Communications Adobe Team, 2018). The phases are also identified and pre-planned, which requires the knowledge of the project scope and team skills to set reasonable deadlines for the phases.

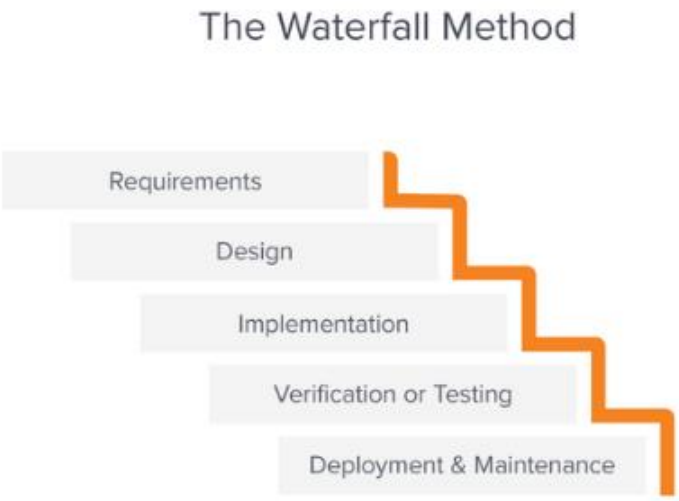


Figure 2. (Communications Adobe Team, 2022)

Advantageous characteristics of agile:

- **Flexible Development**
  Our team and our client are continuing to discover the design intention of the Digital Twin prototype as we develop. While we have functional requirements, we're still unclear on technical design and lack team knowledge both in design and technology stack. Because of this, we need rapid iteration through numerous meetings and highly visible iterations. For this reason, Agile's focus on user stories as well as its prioritization of task progress identification over task deadlines will be invaluable for development.
- **Visual Approach**
  Agile's iterative approach allows us to easily identify the current stage of progression a task is on. Compared to a procedural approach of waterfall, this approach allows us to observe the progression stages of multiple tasks at once and is much easier to distinguish which task needs work, who oversees the task and any additional notes made on the task. For our project, this is much more favourable as there are a lot of different tasks that need to be completed together.
- **Non-hierarchal team structure**
  Waterfall requires a leader to decide the project hierarchy and approve tasks before they can be completed. This strategy increases the cohesiveness of the overall deliverables, but none of the team members possess the information required of the leader, nor do we have time to constantly be checking up with someone before acting. Therefore, to accommodate our project characteristics and to better align with our team's philosophy of shared responsibility and minimize any stress related to hierarchy, we have decided that the roles provided by Kanban are more suitable for us.

Agile's iterative approach allows us to work collaboratively, visually plan and prioritize tasks, and deliver outcomes quickly. By working in short iterations where we can develop and test deliverables together, we can get feedback from stakeholders and adjust our approach as needed, which helps us deliver better outcomes that meet the changing needs of the project. Overall, Agile provides us with a flexible and adaptable framework that is ideal for our project's evolving requirements and constraints.

Within Agile there are a lot of frameworks that adopt the key concepts and make modifications to suit different teams. An example of this would be SCRUM, which is primarily known for its use of "sprints", which is used by teams to regularly deliver results when developing outcomes. Scrum also establishes key roles, such as the product owner, scrum master and development team to assign different responsibilities to members and establish a leadership within the team.

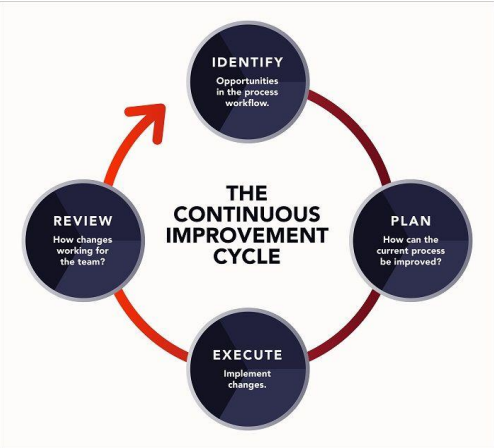


Figure 3 (Lynn, n.d.)

Lean is another agile methodology that prioritizes identifying the value of the outcome that to be delivered to the customer, and establishing a process that brings the highest value to the customer (Kanbanize, n.d.). During this process, activities that are decided to not bring value are discarded to focus team effort onto more important tasks. Tasks are completed by using a pull method, which means they are only completed if there is demand for their value.

Another example is Kanban, which uses a visual approach when planning a project. Kanban converts tasks into independent cards that can be placed in different columns to represent its status of completion, as shown below. Rather than using sprints to set goals of progression for outcomes, Kanban focuses more on the continuous flow of progress in tasks using cycles (Atlassian, n.d.). There are roles such as service request manager and service delivery manager, but there are no set roles that establish leaderships within a team.

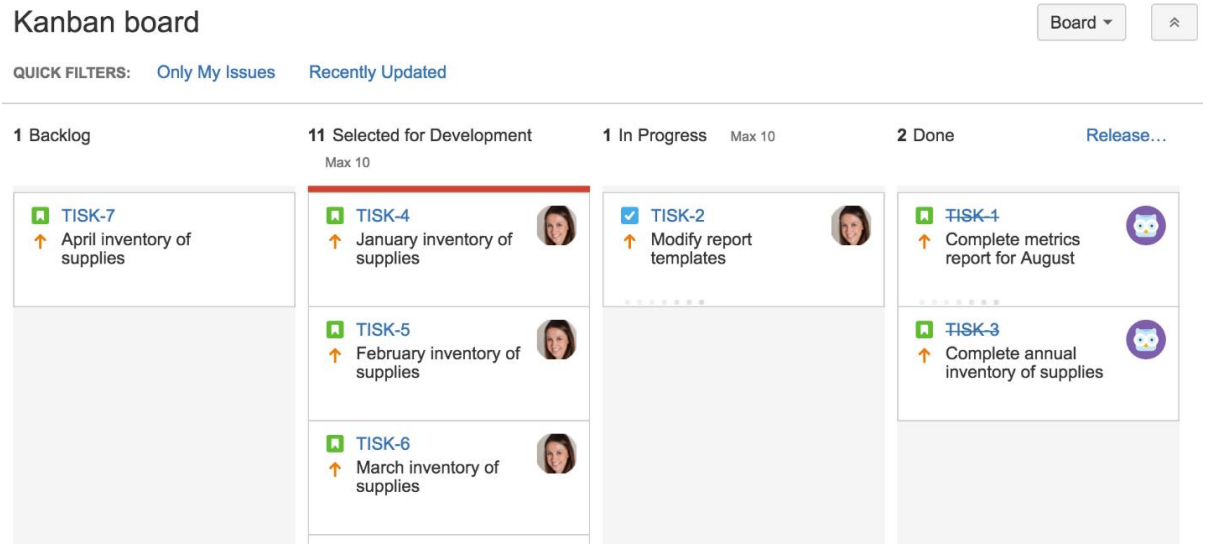


Figure 4. (DX Heros, 2019)

Our team decided to use Kanban for the following reasons:

- **Change Philosophy**
  Our project is exploratory/ research orientated, which means that there are some aspects of our project that we do not yet have a clear solution for. Kanban's flexible structure allows for changes in the workflow to occur without derailing the entire project schedule, which is much more advantageous to our team than using scrum or lean.
- **Flow Method**
  Kanban focuses on the flow of work for the project, rather than establishing goals to finish a task within a sprint (like scrum) or reducing waste and efficiency (like lean). As mentioned before, currently our project has a lot of outcomes where we cannot give an accurate estimation of the workload involved or the time required. Due to this, SCRUM sprints would be difficult to plan for in advance, and we cannot use lean methods and establish activities that can be discarded.
- **Visual Management**

Out of all 3 methods, Kanban's visual management structure employs the use of visual structures (such as labels) in workflow, which can make it easier to identify a task's status or the member assigned to the task easier. Our project is predicted to have a lot of tasks developed at once, and with kanban the progress of each task is clearly visible to everyone and is more advantageous for our project than SCRUM's sprint review or lean's value assignments.

- **Flow Limit**
  Kanban's flow limit ensures that the work done by the team is at an optimal pace without exceeding the work capacity by ensuring some tasks can start once current ones are completed (Kanbanize, n.d.). Although other methods can also limit work, Kanban's visual approach makes identifying and preventing potential bottlenecks easier. For example, our project has a flow limit of 5, which means that there can be only up to 5 tasks in a column at any time excluding certain exception cases (kanbanize, n.d.).

WORK BREAKDOWN STRUCTURE

The work breakdown structure (WBS) is used to break down the key tasks required for our project's success into smaller and more manageable subtasks for our team to complete. By identifying all the tasks required and organizing them into a hierarchal structure it improves visibility of our project scope and the progress made.

- The key tasks in the WBS were established by identifying the deliverables requested by the client as well as the assessment requirements established by AUT.
- Phases were established by modifying the basic agile structure (projectmanager.com, n.d.) to better accommodate the project characteristics. The WBS uses the phases to group together relevant key tasks.
    - The initiation phase of this project was undertaken prior to our team being tasked with the project. As such it has not been included in the structure
    - An Additional Phase was added for upskilling, this phase was added into the project under advisement from the AUT teaching staff to assist in ensuring our team has the necessary skills to achieve our outlined outcomes.

Phase 1- Planning

During the planning phase, we lay out the foundations for our project development and start gathering information on the project's requirements, the tech stack, and the methods we will use to develop a healthy and productive working strategy. This involves:

- Meeting and becoming acquainted with team members through initial team meeting.
- Initial team meeting with mentor to familiarize ourselves with the project and client background.
- Discovering and researching requirements for project.
- Meeting with client to establish details of project, identify project scope and gain insight on factory layout.
- Breaking down project into tasks and establishing user stories.
- Completing appropriate project plans.
- Completing and presenting project proposal.

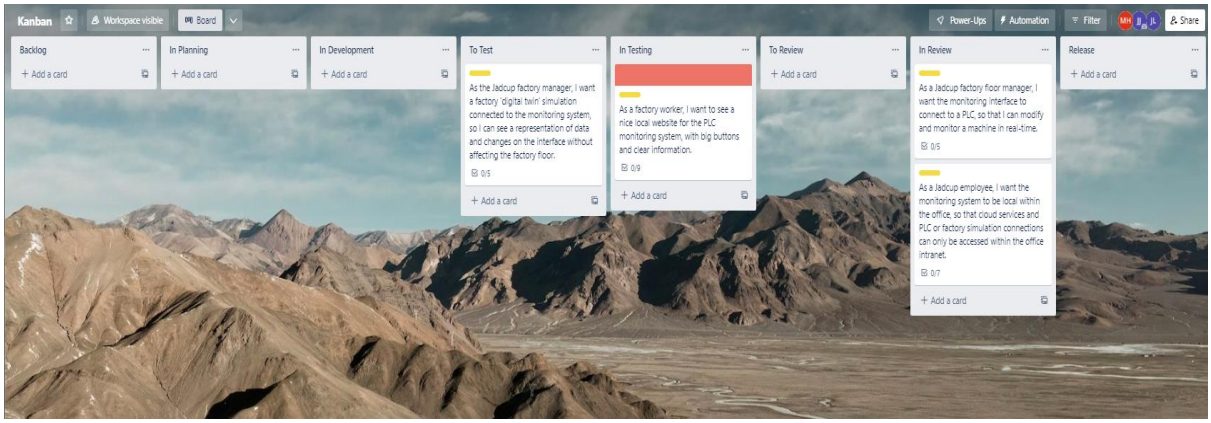
Phase 2- Upskilling

Due to the wide range of skills required for the project, we require some time to analyse further project requirements and ensure that we have the skills and knowledge necessary to efficiently provide the stakeholders with a high-quality outcome. During this phase, we will be following the upskilling schedule and

will be individually upskilling on different fields depending on the tasks we are responsible for, as well as fields that require the proficiency for all team members (e.g., GitHub, Quality assurance strategies, Kanban).

For details on the Upskilling Plan see Appendix 5.2- Upskilling schedule.

Phase 3- Development/Monitoring



Because the requirements for this project involve creating a network of several independent components (e.g., Web Application, Factory IO simulation) each with many subtasks to be completed, we decided that it would be beneficial to develop them simultaneously. Due to this, we combined the development and monitoring phases into one so that we can monitor the component's functionality alone and with other components as we develop them. Additionally, as we have not started the development process, it is highly likely that there will be more subtasks discovered as we develop, at which time we can leverage kanban's focus over continuous work over achievement of pre-established task goals.

As part of the monitoring and development phase, we used Trello, which provides a structure that accommodates for Kanban's visual structure to lay out our tasks. The board as shown above has the following phases:

- Backlog
- In Planning
- In Development
- To Test
- In Testing
- To Review
- In Review
- Release

These phases on our Kanban board allow our team members to see exactly what work is needed for each story as well as what work needs to be done. The subtasks are split into each of these phases and are represented as checklists on the user story card. All subtasks in each phase must be completed before the story is ready to move to the next phase. Utilising Trello's tools we can add comments, documents, checklists, and much more to better visualise and unify our document spread during the development phase.

For further details on our implementation of kanban with Trello and the Release phase, refer to Appendix 6.2 Implementation of Kanban.

Note that because Kanban allows developers to add and modify user story subtasks as they go, we expect to update our subtask checklists for each phase before moving forward, so that all work is clear and understandable before progressing to the next phase.

For an outline of our Quality Assurance procedure refer to Section 9 – Risk and Issue Management Subsection 4: Quality Assurance.

Phase 4- Closing

This phase is used to prepare for the completion and handover of the project deliverables. During this time the team will be completing the final release of the project deliverables and the handover to client and start reviewing the project documentation to prepare for the review of the project.

Phase 5- Project Review

The final phase of the project outlines the assessments for the team project post-handover, and marks estimated deadlines for all requirements of the project. During this time the team will be finalizing the project portfolio, as well as the individual reflective reports.

We decided to add a separate phase for upskilling, as our team had a lot of required skills and we wanted to ensure that every member of the team had the adequate skill levels to be able to complete their assigned tasks. The key tasks would be the tasks on the 3rd column.

For a detailed diagram of the Work Breakdown Structure see appendix 6.1 Work Breakdown diagram.

SECTION 7: TEAM ROLES AND RESPONSIBILITIES

Service Request Manager/ Developer: Josh Ladowsky

- Managing the team
- Organizing team meetings
- Managing team schedules
- Delegating tasks between team members
- Communicating with mentor
- Resolving conflicts
- Delivering project requirements
- Completing delegated tasks
- Attending and actively contributing to meetings
- Communicating any issues/problems
- Documenting project processes

Service Delivery Manager/ Developer: Myles Hosken

- Representative of the team when communicating with client (e.g., email)
- Organizing client meetings
- Delivering team opinions to client during meetings or via online communication
- Answering client's questions
- Delivering project requirements
- Completing delegated tasks
- Attending and actively contributing to meetings
- Communicating any issues/problems
- Documenting project processes

Developer: Harshil Patel

- Delivering project requirements
- Completing delegated tasks
- Attending and actively contributing to meetings
- Communicating any issues/problems
- Documenting project processes

Developer: Jane Jung

- Delivering project requirements
- Completing delegated tasks
- Attending and actively contributing to meetings
- Communicating any issues/problems
- Documenting project processes

Developer: Yeran Edmonds

- Delivering project requirements
- Completing delegated tasks
- Attending and actively contributing to meetings
- Communicating any issues/problems
- Documenting project processes

Every individual in the team is responsible ensuring that they follow the rules of the team contract to maintain a healthy working environment for the duration of this project.
For details on the Team Contract see Appendix 7.1 Team Contract

SECTION 8: PROJECT SCHEDULE

MILESTONE REPORT

For the milestone report we used assessment and breaks dates as key achievements and accomplishments that have been used up for the proposal presentation. We are using these milestones to help prioritize tasks and ensure that we are working effectively and efficiently through our project. Furthermore, we are using milestones to set realistic goals and objectives for the project.

For further information on milestones, see Appendix 8.1 Milestone Report

PROJECT SCHEDULE

The project schedule follows the WBS and uses the project milestones to establish the estimated start and end of each phase. Note that due to the nature of the kanban framework, excluding the deadlines written in the milestone report, the durations of all items can be modified dynamically.

Our project schedule is split into the following:

	Start	End
Phase 1- Planning	06/03/23	07/04/23
Phase 2- Upskilling	10/04/23	23/04/23
Phase 3- Development/Monitoring	24/04/23	24/09/23
Phase 4- Closing	25/09/23	06/10/23
Phase 5- Project Review	09/10/23	27/10/23

Because we had a large chunk of time dedicated to developing/monitoring, we decided to further plan out iterations so that it would be easier for us to identify the tasks we needed to complete during each week. As shown in the project schedule, each iteration has a planning phase, where we have time allocated to get together as a group and analyse the user stories we need to complete and assign members/discuss details. At the end of each iteration, there is also a review/update session, in which the team will gather and update each other on the work done that iteration, as well as take feedback from each other to further develop the assigned tasks until the start of the next iteration.

For further details of project timeline as well as more details of each iteration phase see: Appendix 8.2 Project Schedule

TEAM SCHEDULE FOR PART 1

Part 1 involves the planning and upskilling phases as well as the start of the development phase. Below are the details for weekly scheduling as well as the established meeting times agreed upon by all participants.

Semester 1 availability

	Mon	Tue	Wed	Thu	Fri
6:00					
7:00					
8:00					
9:00					
10:00					
11:00					
12:00					
13:00					
14:00					
15:00					
16:00					
17:00					
18:00					
19:00					
20:00					
21:00					

Harshil unavailable

Jane unavailable

Joshua unavailable

Myles unavailable

Yeran unavailable

offsite

workshop

Meeting schedule

	Frequency	Date/Time	Location
Offsite hours	Weekly	Tuesday 8am-2pm	
Team meeting	Weekly	Tuesday 12pm	WZ701
Makeup meeting		Saturday 9am	Online
Mentor meeting	Weekly	Tuesday 10am	WZ1101
Client meeting	One-off	16th March 1pm	Jadcup headquarters

Due to the conflicting schedules between team members, we decided that we would hold the team meetings during offsite hours, and when required we would have a makeup meeting in the weekends.

For further information on meetings see 8.3 Meeting Minutes

SECTION 9: RISK/ISSUE MANAGEMENT

RISKS

Potential project risks that could pose a threat during the project development were identified and evaluated, as well as the potential responses that could be made by the team to prevent the risks from happening or reduce the damage as much as possible.

The threat levels of the risks were based on the probability of the risk occurring, as well as the impact of the risk if it were to occur. The team members assigned on risk documentation will be regularly updating the risk register during team stand-up meetings.

We have added a Risk Register to our appendix that specifies our risks, but a particularly strong risk worth mentioning is the communication and review plans with our client, Jadcup. As AUT approached Jadcup with this project, the client has less buy-in and vested interest in the project and is less beholden to communication and reviewing of the product. Additionally, because they didn't conceive the idea, they have less clear requirements or ideas of what they want. Our current risk mitigation strategy is to keep the client informed, and bake any feedback into our change management strategy.

For details see Appendix 8.1: Risk Register

ISSUE LOG

The issue log shows records of unexpected problems the team has faced so far during the project, and the potential solutions to mitigate the issues, as well as the measures taken to prevent reoccurrence of similar issues in the future. The issues are prioritized between low/medium/high threat levels, which corresponds to how soon we attempt to find a solution.

For details see Appendix 8.2: Issue Log

CHANGE MANAGEMENT

According to (Ramesh Lal, 2021), Kanban frameworks prioritize value and flow over-time-based milestones, this allows for project changes to be easily integrated into the project's lifecycle. We have adapted the same approach by breaking down the project schedule into 3 sections:
High-level: High-level planning is the process of the team setting overall goals and objectives according to the project's requirements. A work breakdown structure has been created by identifying key milestones and deliverables that needed to be achieved.

Design phase: The design phase breaks down the high-level plan into more specific features, user stories and requirements that needed to be implemented to the achieve the goals and objectives set in the high-level plan.
Iteration planning: Iteration planning is utilized to break down the design plan into smaller and more manageable tasks that can be completed in a set period. Iteration planning assigns tasks to individual team members and focuses on short term goals and priorities, and allows the team to use the milestones established earlier as a baseline to structure their work for the week as well as giving freedom for teams to dynamically change the tasks depending on the current task progress and the value/priority of the task at the current stage.

The advantage of using Kanban is that changes are managed dynamically during Iteration planning. Changes can be identified in conjunction are identified continuously during the planning and the review/update phases of the current iteration.

The impact of the change is measured based on its impact to the flow of the following iteration and project. The team will calculate the changes and how much work needs to be done to effectively make the change. After review, should implementation have a minor effect on the flow of the project then the change will be implemented during the update phase. If the change disrupts the flow, further identification of the changes will be divided into several cards/tasks and implemented onto the Kanban board as necessary. During the planning phase of the following iteration, the changes will be moved into the priority swim lane on the Kanban Board.

If a change is requested directly by the client, the same process will be followed. The impact and value of the change will be considered and worked into the planning and review/update phase of the iteration. If a change is identified to be too large or out of scope, the team will discuss a compromise and come to a solution. If the change can be broken down and worked into the current iteration without disrupting the flow, new cards will be created and placed onto the Kanban board. If the current amount of in development cards will be exceeded, then achievable changes will be worked into the backlog and reviewed during the next review/update phase.

The change management strategy is in place to ensure that current tasks and development are kept relevant. Before every iteration and at each stand-up (Saturday meeting) and mentor meeting (Tuesday), a change review may take place to ensure that the team is on track and understands their current tasks and requirements. If a change is required, the Product Delivery Manager will be responsible for updating the change and implementing the update within the Kanban board. If the client requires documentation of the change, it will be produced and sent.

QUALITY ASSURANCE

Kanban's methodology divides the progression of a task into phases of planning, analysis, development, review, and release. During the review phase, we will be applying the outlined procedures in the Quality Assurance plan to ensure that our outcomes are robust.

Using our Quality Assurance Plan, we will be able to minimise our outlined risks and produce a project of higher quality that will have a reduced risk of damaging the client's equipment. This is done by ensuring any all code developed is tested.

To ensure the quality of our tests we will utilise a variety of metrics measuring our overall testing and bug distribution. As well as designing systems for reporting false QA test outcomes and non-compliance of test outcomes.

Should non-compliance or false QA testing be discovered then the Audit and review procedures are outlined to ensure any testing done is to a professional standard.

For Details about Q.A. procedures see Appendix 9.3: Quality Assurance Plan

SECTION 10: COST

The following outlines the estimated costs of the project from now till the end of the project due to further research is required to find pricing of Database and Dashboard hosting as many services such as firebase (Google Developers, n.d.) and AWS (Amazon Web Servies, n.d.) use systems to pay as you use so precise estimations can only be made when the system is in further development as such an estimate has been made based on a low price system averaged from both of these.

Furthermore, currently it is expected that Jadcup's existing hardware is adequate for the systems to be developed and no additional modifications or modules will need to be added into the environment.

Senior Management costs are based on AUT's estimated hourly pay for one academic hour, while the Development Team's wage is based on the lower end of a junior developer's hourly wage. Trainee cost is assumed from the average warehouse worker wage in NZ.

WBS items	# Units/Hrs	Cost/Unit/Hr	Subtotals	WBS Level 1 Totals $ of Total	% of Total
1. Project Management				$9,699.00	73.25%
1.1 Senior Management	30	$163.30	$4,899.00		
1.2. Development Team	150	$32.00	$4,800.00		
3. Software				$1,052.28	7.95%
3.1 Factory I/O	7,333.33	$0.08	$550.00		
3.2 DB/ Dashboard hosting (Estimated cost)	3,666.67	$0.14	$502.28		
4. Testing (10% of software)				$105.23	0.79%
5. Training				$178.00	1.34%
5.1 Trainee cost	2	$25.00	$50.00		
5.3 Project team members	4	$32.00	$128.00		

6. Reserves (20% of total estimates)				$2,206.90	16.67%
Total project cost estimate				$13,241.41	

REFERENCES

Amazon Web Servies. (n.d.). *AWS Pricing Calculator*. Retrieved March 30, 2023, from AWS: https://calculator.aws/#/?nc2=pr

Atlassian. (n.d.). *Kanban - A brief introduction*. Retrieved March 30, 2023, from Atlassian: https://www.atlassian.com/agile/kanban

atlassian. (n.d.). *What is Agile?* Retrieved March 30, 2023, from atlassian: https://www.atlassian.com/agile

Communications Adobe Team. (2022, March 18). *Waterfall Methodology: A Complete Guide.* Retrieved March 29, 2023, from Adobe Buisness: https://business.adobe.com/blog/basics/waterfall

DX Heros. (2019, November 12). *Kanban | Developer Experiance Knowledge Base*. Retrieved March 29, 2023, from develop erexperience: https://developerexperience.io/articles/kanban

Google Developers. (n.d.). *Firenbase Pricing*. Retrieved March 30, 2023, from Firebase google: https://firebase.google.com/pricing

IBM. (n.d.). *What is Industry 4.0?* Retrieved March 29, 2023, from IBM: https://www.ibm.com/topics/industry-4-0

kanbanize. (n.d.). *Prioritizing Tasks with Kanban*. Retrieved March 25, 2023, from kanbanize: https://kanbanize.com/kanban-resources/getting-started/prioritizing-tasks-with-kanban

Kanbanize. (n.d.). *What Is a WIP Limit in Kanban, and Why Is It Important?* Retrieved March 30, 2023, from Kanbanize: https://kanbanize.com/kanban-resources/getting-started/what-is-wip#:~:text=Kanban%20WIP%20limits%20help%20your,you%20finish%20throughout%20the%20organization

Kanbanize. (n.d.). *What Is Lean Management? Definition and Benefits*. Retrieved March 30, 2023, from Kanbanize: https://kanbanize.com/lean-management/what-is-lean-management

Lynn, R. (n.d.). *Lean Methodology*. Retrieved March 30, 2023, from Planview: https://www.planview.com/resources/articles/lean-methodology/

projectmanager.com. (n.d.). *What Is a Work Breakdown Structure (WBS) in Project management*. Retrieved March 29, 2023, from projectmanager.com: https://www.projectmanager.com/guides/work-breakdown-structure

Ramesh Lal, T. c. (2021). Three Levels of Agile Planning in a Software Vendor Environment. *ACIS 2021*, pp. 1-12. Retrieved from https://aisel.aisnet.org/acis2021/48/

Schwab, K. (2016, January 14). *The Fourth Industrial Revolution: what it means, how to respond*. Retrieved March 29, 2009, from https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/: https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/

Ztaro. (2020). Agile development process infographic stock illustration. *Agile development process infographic stock illustration.* IStockphoto.

DISCLAIMER

Auckland University of Technology

Bachelor of Computer & Information Sciences

Research & Development Project

Disclaimer:

Clients should note the general basis upon which the Auckland University of Technology undertakes its student projects on behalf of external sponsors:

While all due care and diligence will be expected to be taken by the students, (acting in software development, research, or other IT professional capacities), and the Auckland University of Technology, and student efforts will be supervised by experienced AUT lecturers, it must be recognised that these projects are undertaken in the course of student instruction. There is therefore no guarantee that students will succeed in their efforts.

This inherently means that the client assumes a degree of risk. This is part of an arrangement, which is intended to be of mutual benefit. On completion of the project it is hoped that the client will receive a professionally documented and soundly constructed working software application, some part thereof, or other appropriate set of IT artefacts, while the students are exposed to live external environments and problems, in a realistic project and customer context.

In consequence of the above, the students, acting in their assigned professional capacities and the Auckland University of Technology, disclaim responsibility and offer no warranty in respect of the "technology solution" or services delivered, (e.g. a "software application" and its associated documentation), both in relation to their use and results from their use.

PROJECT CHARTER

Project Title: Jadcup Digital Twin Factory Management

Project Objectives:

Provide Jadcup with a prototype for a user-friendly interface to connect and control PLCs remotely.

Main Project Success Criteria:

- The application must be delivered by the end of October 2023.
- The application must be fully functional and meet all the functional and non-functional requirements.
- The application must have a high level of user satisfaction, as measured by feedback received from the client.

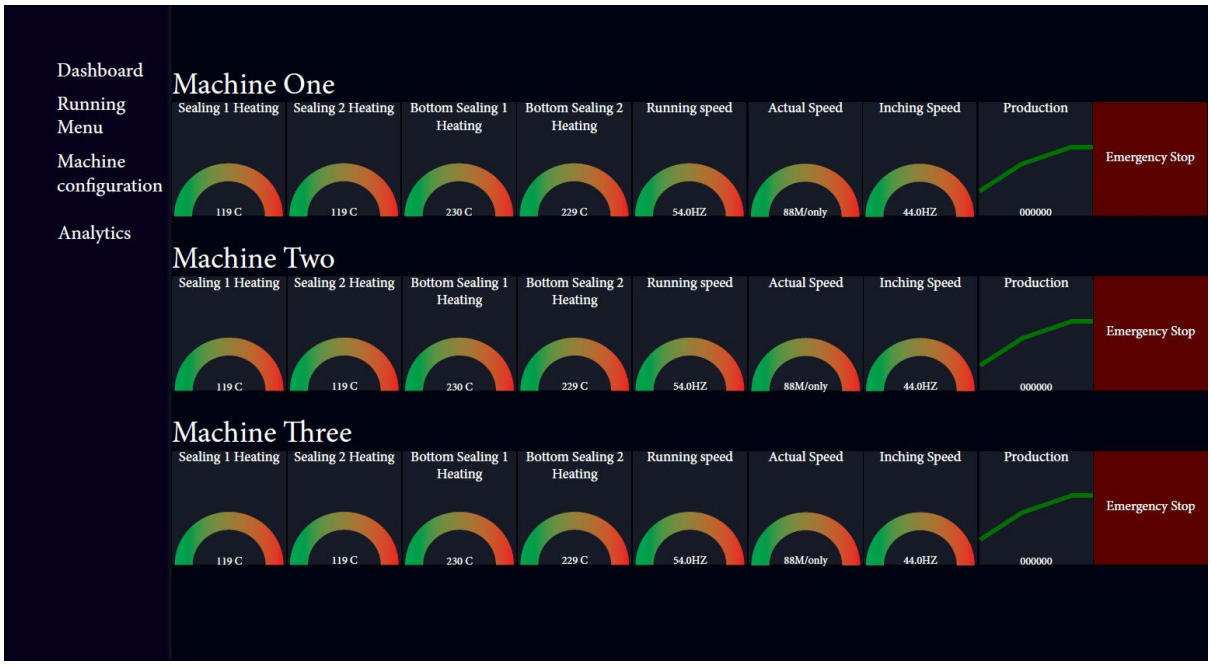
Approach:

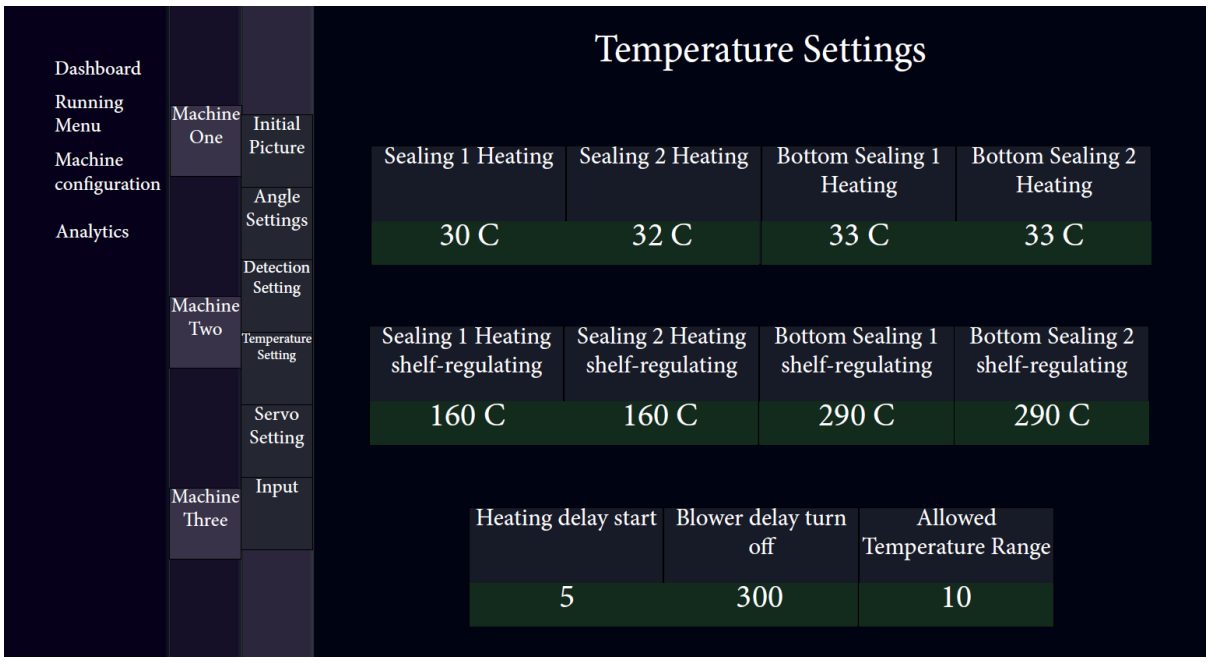
Develop a web-based FERN application that can connect and interact with programmable logic controllers (PLC) and a simulation that can demonstrate communication via remote control.

Roles and Responsibilities

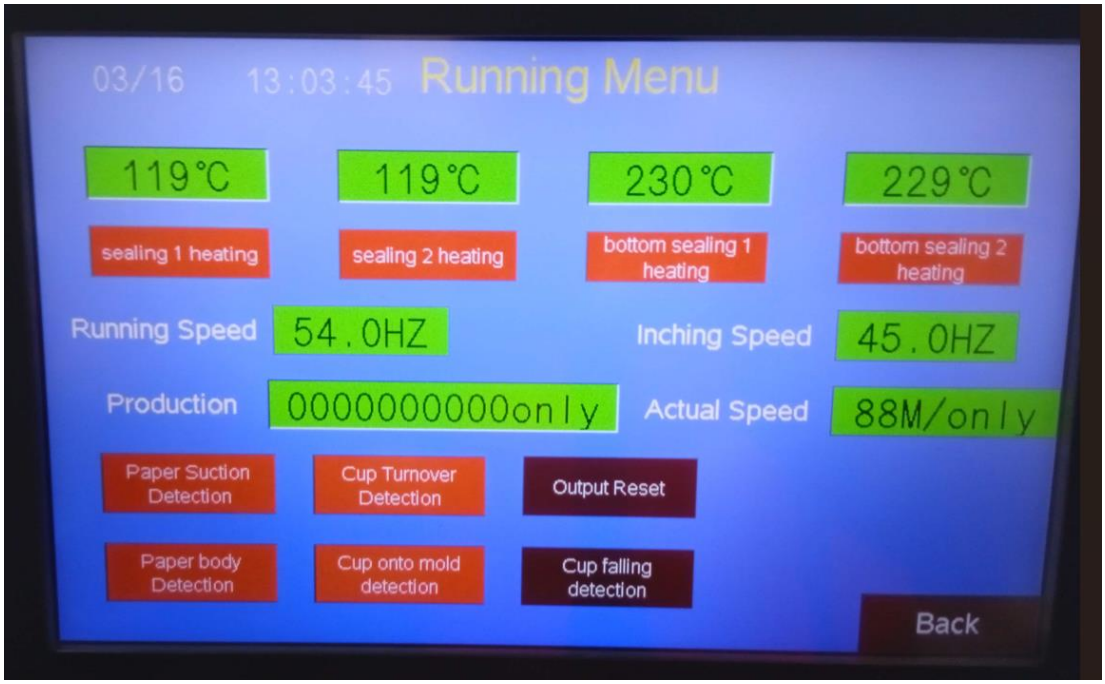
Role	**Name**	**Organization/ Position**	**Contact Information**
Team Member	Harshil Patel	Development Team	dtm3080@autuni.ac.nz
Team Member	Jane Jung	Development Team	ssr1891@autuni.ac.nz
Team Member	Joshua Ladowsky	Team Leader/ Development Team	trf1640@autuni.ac.nz
Team Member	Myles Hosken	Client Liaison/ Development Team	jky9144@autuni.ac.nz
Team Member	Yeran Edmonds	Development Team	rfh8366@autuni.ac.nz
Team Mentor	Matthew Kuo	AUT	matthew.kuo@autuni.ac.nz
Client	Daniel Li	Jadcup Founder	daniel@jadcup.co.nz
Paper Leader	Ramesh	AUT	ramesh.lal@aut.ac.nz
Paper Leader	Jacqui	AUT	jacqueline.whalley@aut.ac.nz
Moderator	Tony	AUT	tony.clear@aut.ac.nz
External Client Engagement	Weihua	AUT	weihua.li@aut.ac.nz

UI DESIGN DRAFTS





1.1 EXISTING PROGRAMMABLE LOGIC CONTROLLER INTERFACE



3.1 SCOPE STATEMENT

Project Title: Jadcup Digital Twin

Date: 08/03/2023

Project Justification:

Jadcup wants to modernize their factory processes and pull their production into Industry 4.

This project is a prototype to determine the viability of an interface to manage PLCs on the Jadcup factory floor.

Purpose and Objectives:

The purpose of this project is to develop a web-based FERN (Firebase, Express.js, React.js, Node.js) application that allows Jadcup to connect and interact with programmable logic controllers (PLCs).

The objective of this project is to provide Jadcup with a prototype for a user-friendly interface to connect and control PLCs remotely.

Product Characteristics and Requirements:

1. Web interface
2. Factory I/O Simulation (strictly for the purposes of this prototype)
3. Read/Modify PLC Data

Functional Requirements:

- The application must allow users to connect & control PLCs.
- The application must show data managed by PLCs.
- The application must be modular enough to support multiple, different PLCs.

- The application must provide secure user authentication and authorization.

Non-Functional Requirements:

- The application must be responsive and have a fast load time.
- The application must be functional and user-friendly when run on a PC.

Summary of Project Deliverables

1. A functioning FERN web application prototype for interfacing with PLCs
2. A design document for the application
3. A presentation for the web application

Assumptions & Constraints:

- The application will be developed using Kanban methodology.
- The development team will consist of five developers, one of which is a service delivery manager, and one of which is a service request manager.
- The team will be managed & report to a Team Mentor and a Client
- The application will be hosted on a cloud-based platform.
- The development will be completed by the end of October, 2023.
- The application will be a prototype.

Stakeholders:

- The primary stakeholder for this project is Jadcup who will use the FERN application to connect and interact with PLCs.
- The project team, team mentor, team mediator and paper leads are also stakeholders in this project.

Project Success Criteria:

- The application must be delivered by the end of October, 2023.
- The application must be fully functional and meet all the functional and non-functional requirements.
- The application must have a high level of user satisfaction, as measured by feedback received from the client.

3.2 STAKEHOLDER MANAGEMENT STRATEGY REGISTER

Name	Position	Project Role	Level of Interest	Level of Influence	Potential Management Strategies
Matthew Kuo	AUT Lecturer	Team Mentor	Medium	Medium	Keep informed, do his clear requirements. He likes seeing work done and his team keeping ahead and on-schedule. Has a preference for strong technical knowledge.
Yeran Edmonds	AUT Student	Team Member	High	High	Prefers clear tasks. Can be bribed with snacks.
Harshil Patel	AUT Student	Team Member	High	High	Keep informed, get opinion on development. Prefers to be directly given tasks.
Jane Jung	AUT Student	Team Member	High	High	Keep informed. Likes to keep on top of work and stay ahead.
Myles Hosken	AUT Student	Team Member	High	High	Prefers to get work done asap to not work weekends. Likes to stay ahead as much as possible.
Joshua Ladowsky	AUT Student	Team Member	High	High	Prioritizes Factory IO development and managing the PLCs. Strong technical background.
Tony Clear	AUT Lecturer	Team Mediator	Medium	Medium	Likes Agile practices and clear, readable documentation.
Ramesh	AUT Lecturer	Paper Leader	Low	Medium	Very low communication/influence but prefers Agile and nicely presented information (coloured documents).
Jacqui	AUT Lecturer	Paper Leader	Low	Medium	Very low communication/influence.
Client	Jadcup Project Manager	Team Client	Medium	High	Very busy. Prefers to keep informed via email

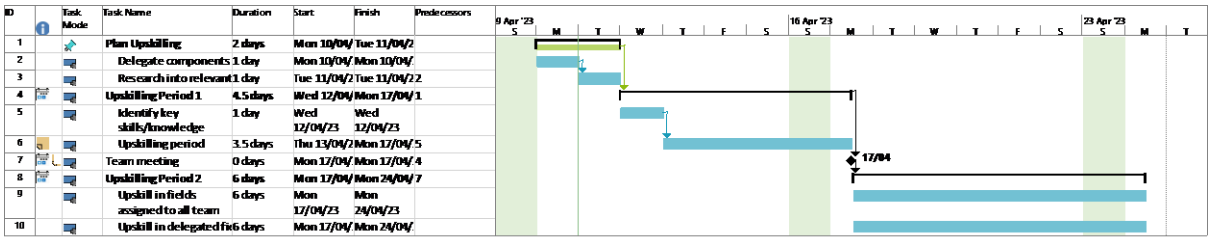
4.3 IDENTIFIED TECHNICAL ARCHITECTURE

Technology	Purpose
Git	Industry standard version control and code management tool for collaboration. We will use Git in combination with Fork, a Github extension for visualizing branches and merging, to keep track of changes and version history.
JavaScript / React	Dynamic and responsive user interface capability for the web dashboard that can be customized according to the client's needs. Third-party libraries can facilitate communication between the factory's Programmable Logic Controllers (PLCs) and the dashboard.
Programmable Logic Controllers (PLCs)	Industry standard for machine programming and automation, used to control the factory machinery.
FactoryIO	Simulation software to replicate the factory environment. It will allow us to create a prototype factory and begin reading data from the simulated PLC.
Web API's	Application Programming Interface (API) for real-time communication between the factory and dashboard.
NodeRED	Open-source flow-based programming tool built on top of Node.js. Used to create a workflow for the PLC and facilitate direct communication with the dashboard within the same tech stack.
Online Database	Online database to store information securely and reliably, such as client login credentials. Firebase has been considered as it offers 1GB of data transfer per month, which should be sufficient for the MVP and the client's needs.
Online Server	Hosts the web dashboard. Selection of hosting provider, operating system, and necessary software is required. Reliable free services may not be available, and the cost of the server may need to be factored into the project budget.
IDE	An Integrated Development Environment (IDE) for JavaScript development, such as Visual Studio Code or WebStorm, would be suitable for developing the web dashboard using JavaScript/React. Additionally, an IDE or development environment for programming the PLCs may be required depending on the specific hardware and software used.

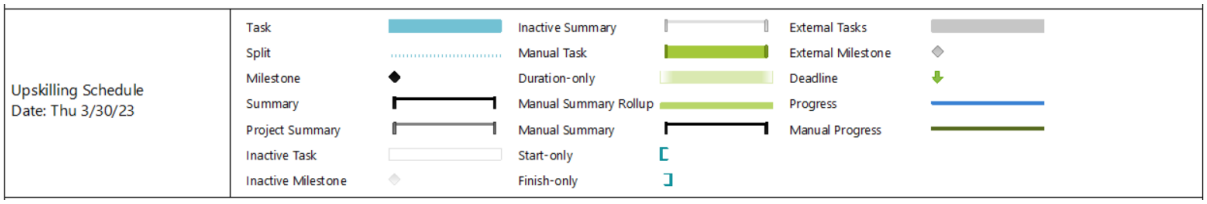
5.1 SKILLS MATRIX

Group	Skill	Harshil ISTP-A					Jane-ESTP-A					Josh - INTP-T					Myles - ISFJ-T					Yeran - ISFJ-A					Total
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	
Application Server	Online Servers	1							3				2				1							3			10
	Server Administration	1						2				1					1							3			8
Component Technology	PLC		2				1						2				1					1					7
	Factory I/O	1					1							3			1					1					7
Database	NoSQL		2					2				1						2							4		11
	SQL			3					3				2					2						3			13
	Online Database		2						3			1						2						3			11
Language/Development Tools	C#				4						5					5			3					3			20
	Git			3						4					4				3					3			17
	HTML/CSS			3							5			3					3				2				16
	IDEs			3						4				3						4					4		18
	Java			3							5		2							4				3			17
	NodeRed/SCADA	1					1					1					1					1					5
	Visual Studio					5					5				4					4				3			21
	Vs Code		2								5					5		2						3			17
Multimedia	Adobe Illustrator		2						3					3			1					1					10
	UI Design			3						4		1						2				1					11
	UX Design		2							4		1						2				1					10
Project Management	Agile Methodology Knowledge			3					3					3					3					3			15
	Client Management		2						3				2							4				3			14
Project Management Tools	Github				4					4				3					3						4		18
	Teams					5					5			3					3				2				18
	Trello					5					5				4			2						3			19
QA Expertise	QA Tools	1					1								4			2					2				10
Web Technology	JavaScript			3						4			2						3						4		16
	Node.Js		2				1						2					2						3			10
	PHP	1						2					2					2				1					8
	React	1						2				1							3				2				9
	Web APIs		2						3				2				1								4		12

5.2 UPSKILLING SCHEDULE



ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		Plan Upskilling	2 days	Mon 10/04/2	Tue 11/04/2	
2		Delegate components	1 day	Mon 10/04/	Mon 10/04/	
3		Research into relevant	1 day	Tue 11/04/2	Tue 11/04/22	
4		Upskilling Period 1	4.5 days	Wed 12/04/	Mon 17/04/1	
5		Identify key skills/knowledge	1 day	Wed 12/04/23	Wed 12/04/23	
6		Upskilling period	3.5 days	Thu 13/04/2	Mon 17/04/	5
7		Team meeting	0 days	Mon 17/04/	Mon 17/04/	4
8		Upskilling Period 2	6 days	Mon 17/04/	Mon 24/04/	7
9		Upskill in fields assigned to all team	6 days	Mon 17/04/23	Mon 24/04/23	
10		Upskill in delegated fie	6 days	Mon 17/04/	Mon 24/04/	



Note: Upskilling phase 1 is individual commenced by all team members independently, Upskilling phase 2 will involve team members collaborating to unify skill application strategies for different platforms.

5.3 TRAINING PLAN

The below is a training plan targeted towards industrial automation in conjunction with a web dashboard. The identified skills are generalized as some members skills are higher in some areas than others.

1. **Kanban Methodology**. Kanban is the chosen methodology for the project members will need to be familiar and up to date with:
    - The principles of Kanban. Workflow, visualization, learning how to prioritize and limit work in progress (WIP).
    - Understanding of the Kanban board.
    - Introducing Kanban into the project.
    - Utilizing Kanban alongside a visual board such as Jira or Trello.

2. **JavaScript / React**. As the main deliverable is a web dashboard that is responsible for reading and controlling Programmable Logic Controllers, we have identified that JavaScript / React will be a suitable language for this.
    - Transferring existing programming knowledge into JavaScript. Basics and advanced programming techniques.
    - If react is needed, then understanding react architecture, react components, events, built-in hooks, and using in-built libraries.
    - Implementing a simple React application.

3. **Factory I/O.** As professionally suggested, FactoryIO has been decided as the 3D simulation tool to replicate a factory.
    - Familiarity with FactoryIO and what it is capable of.
    - Factory simulation and creation as best as can be replicated to the client's factory.
    - Connecting FactoryIO to a Programmable Logic Controller (PLC).
    - Simulating a production line according to the client's requirements.

4. **Programmable Logic Controllers (PLCs).** PLCs are a new area of upskilling that the team will need to be familiar and capable of working with.
    - Being familiar with common industrial protocols such as Ethernet/IP, OPC UA and Modbus.
    - Configuring PLCs to communicate and transfer data with other devices.
    - Familiarity with common debugging and troubleshooting techniques and issues.

- Coding PLCs to read and write data using a protocol.

5. **Factory I/O Web APIs.** Connecting FactoryIO with a web dashboard is going to require communication over an API to read and write data.
   - Understanding how to connect FactoryIO with a web socket and web API.
   - Creating and sending HTTP requests which may read and write data to either Factory I/O and the dashboard.

6. **NodeRED and SCADA.** NodeRED and SCADA are common tools used in industrial settings. To simulate a factory alongside PLCs, knowledge for NodeRED and SCADA may need to be investigated. SCADA systems consist of hardware and software which analyse and collect data from sensors. This data is processed and sent to control devices.
   - NodeRED and SCADA familiarity
   - Understanding, creating, and managing flows.
   - Integration with SCADA and NodeRED.

https://www.allaboutcircuits.com/technical-articles/an-introduction-to-scada-systems/

7. **Databases.** A database may be needed to hold and store machine and user login information.
   - Recap on Database syntax.
   - Understanding database schemas.
   - Connecting the database to the web dashboard and PLC/computer.
   - Hosting a database either locally or cloud based.

This training plan encompasses a diverse range of topics that are pertinent to a research and development project in industrial automation. It offers a thorough comprehension of crucial areas including Kanban methodology, Javascript / React, PLC protocols, Factory I/O, Factory I/O Web API's, NodeRED and SCADA, and databases. Upon finishing the training, participants will possess the requisite abilities and expertise to build a virtual factory, manipulate it using various software tools, and manage and archive data associated with the factory.

6.1 WORK BREAKDOWN STRUCTURE

1. **Planning**
   a. Discover Requirements
      i. Functional
      ii. Non-functional
   b. Research Tech Stack
   c. Complete Proposal
      i. Terms of Reference
         1. Project Charter
      ii. Project Rationale
         1. Project Scope
      iii. Objective/Scope
         1. Scope Statement
         2. Stakeholder management
         3. Identified Techincal Infrastructure
      iv. Skill Analysis
         1. Skill Matrix
         2. Training Plan
      v. Team Roles
         1. Team Contract
      vi. Team Schedule for part 1
         1. Meeting minutes

vii. Project Management Methodology
    1. WBS
viii. Risk and Issues Management
    1. Risk Register
    2. Issue Log
    3. Change Management
    4. Quality Assurance form
ix. Project Schedule
    1. Milestone Report
    2. Gantt Chart
x. Cost Estimate

2. Upskilling

a. Analyse Project Requirements
b. Delegate Tasks

3. Development and Monitoring

a. Iteration 1
    i. milestones
        1. Server Host Online
        2. Website hosted.
    ii. Planning
        1. Develop user stories.
        2. Plan Factory IO
    iii. Implementation
        1. Create GIT repo and branching strategy.
        2. Create Server Host
        3. Connect Website to Server
    iv. Review/Update
b. Iteration 2
    i. milestones
        1. Factory IO prototype working.
        2. UI Design mock-up is designed and approved.
    ii. Planning
        1. Analyse User Stories.
        2. Plan firebase connection.
        3. Plan website design.
    iii. Implementation
        1. Develop Factory IO prototype.
        2. Develop backend server.
    iv. Review/Update
c. Iteration 3
    i. milestones
        1. Prepare for Stakeholder Demo One.
        2. Mid-term Review.
        3. Web API connects to database.
    ii. Planning
        1. Analyse User Stories.
        2. Plan Factory IO Backend Connection.
    iii. Implementation
        1. Develop Factory IO prototype features
        2. Develop Website prototype
        3. Develop Firebase Connection
    iv. Review/Update
d. Iteration 4
    i. milestones

1. Web API connects to database

ii. Planning

1. Analyse User Stories
2. Plan Website UI design
3. Research PLC Connections

iii. Implementation

1. Develop Factory IO prototype features.
2. Develop Firebase Connection

iv. Review/Update

e. Iteration 5

i. Milestones

1. UI Design implemented onto web host.

ii. Planning

1. Analyse User Stories

iii. Implementation

1. Develop Factory IO prototype features.
2. Develop Website UI

iv. Review/Update

f. Iteration 6

i. Milestones

1. Web API Connects to Factory IO/ PLC

g. Iteration 7

i. Milestones

1. Prepare for Stakeholder Demo Two
2. Database Connects to Website

h. Iteration 8

i. Milestones

1. Website can modify Factory IO/ PLC

i. Iteration 9

i. Milestones

1. Website meets speed requirements.

j. Iteration 10

i. Milestones

1. Prepare for Stakeholder Demo Three

k. Iteration 11

i. Milestones

1. Optimizing & Testing

4. Closing

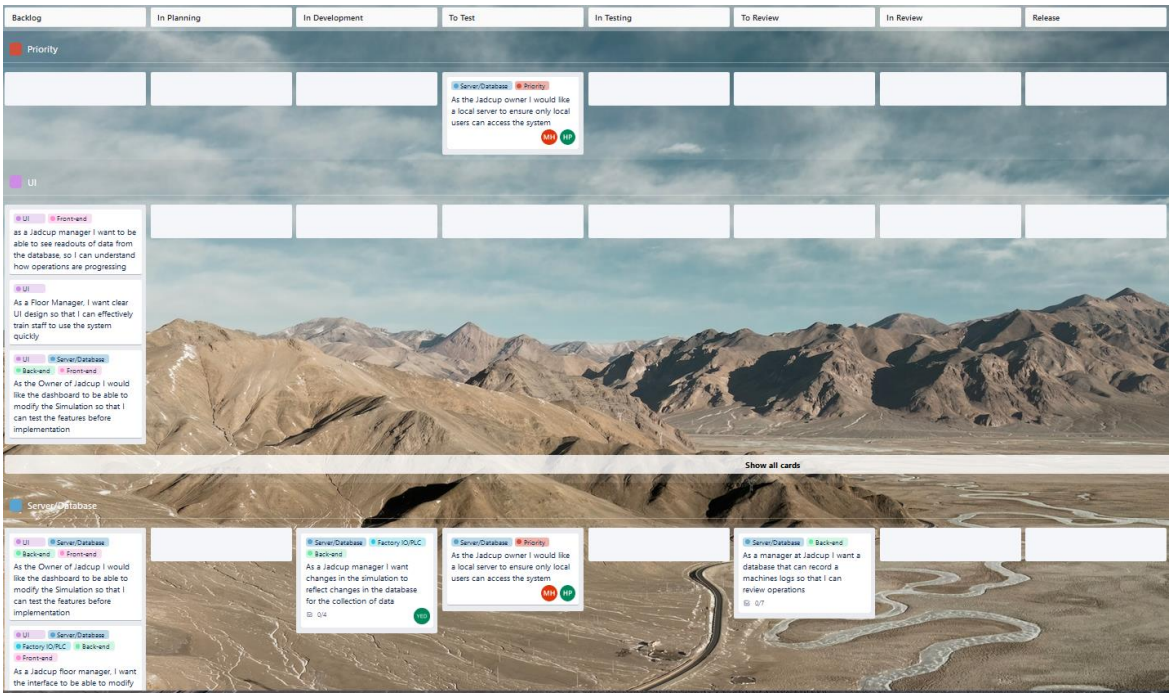
a. Final Release
b. Project Deliverables Handover

5. Project Review

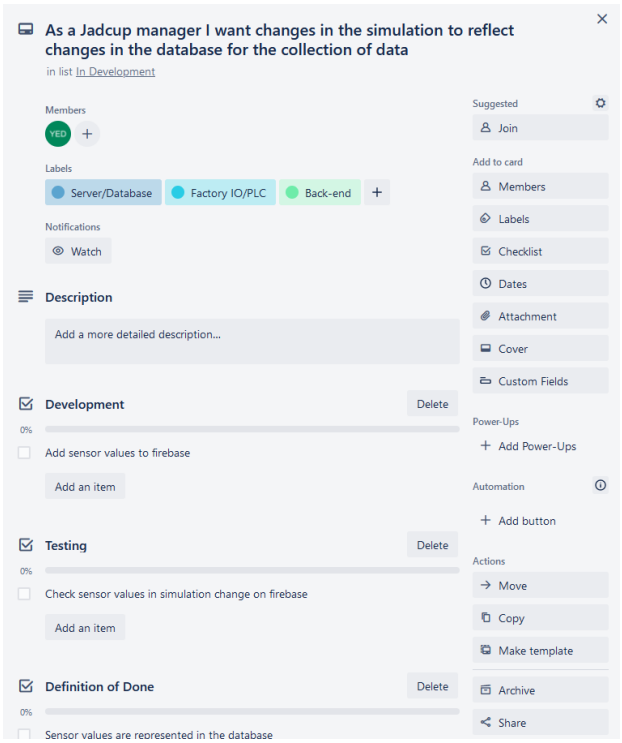
a. Client Feedback
b. Mentor Feedback
c. Team Poster
d. Final Project Presentation
e. Team Portfolio
f. Reflective Report

6.2 IMPLEMENTATION OF KANBAN

- The kanban board is represented digitally via a Trello board. Using a swimlanes powerup we can use our tags to generate swimlanes for our cards

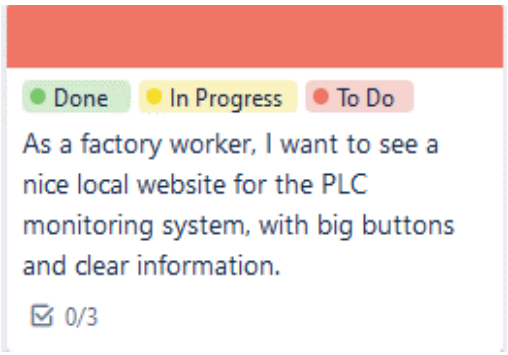


- The key tasks are identified and formatted into user stories. Each user story the contains a list of acceptance tests (can also be identified as subtasks to complete) that the task must pass before it can be marked as completed).



- User stories (tasks) are placed in the product backlog column as cards and can be moved between columns depending on the level of progression.

- If a card becomes expedited, it will be tagged with priority.



Occasionally during a project there becomes a card that blocks the flow of production, when this occurs this can ripple into other cards and cause holding patterns on other cards until this one is completed. A card may need to be expedited due to failures in testing/ reviewing or because it is holding other cards that need to be done. To reduce this risk and limit these consequences we can expedite a card, this means that it becomes the priority in being tested, reviewed, or otherwise worked on. Cards that are expedited ignore the flow limit of a given column.

7.1 TEAM CONTRACT

1. **Working online**
   The team has recognized due to schedule timings we will not have much time outside of Monday-Friday to work collaboratively as a team. To combat this the team has agreed to meet on Saturdays 9:30am-12pm online via Discord.

2. **Commitments**

   As a team we will:

   - Agree to attempt work that we have the ability for.
   - Be realistic when planning and making schedules.
   - Highlight any potential problems and work to fix them before they happen.
   - Keep other team members informed and communicate effectively.
   - Keep information regarding the client confidential.
   - Focus on the project.
   - Proceed to see the project through to its completion.

3. **Team meetings**

   As a team we will:

   - Meet at the agreed scheduled time.
   - Notify team members if you will be absent with adequate notice.
   - Give every member the opportunity to contribute.
   - Be open and listen to all ideas that people present.
   - Not place blame on any single member of the team.
   - Keep meeting conversation to the topic at hand, further clarification can be taken offline.

4. **Problem Solving**

   As a team we will:

   - Encourage all team member's input.
   - Use each other's ideas and build off them.

5. Conflicts

As a team we will:

- View any conflicts as part of working in a team and an opportunity to grow as a team.
- Seek to understand each other's opinions.
- Clarify with each other what we understand and communicate if it is correct.
- Acknowledge valid points that the other person has made.
- Points of view will be stated in a non-judgemental and non-attacking manner.
- Seek to find common ground.
- If a resolution cannot be found, third party support can step in.

6. Meeting Guidelines

- The allocated offsite hours on **Tuesday 8am-2pm** will stay and be reserved for the project meetings and work the team needs to complete.
- Team members will attend meetings prepared.
- Meeting discussions will stay on topic.
- A brief for the next week's meeting will be discussed or planned throughout the week before the meeting.

7. Documentation

- Use OneDrive to host final submission.
- Keep working documents on Teams for mentor visibility.
- Use Trello to organize work required and work results.

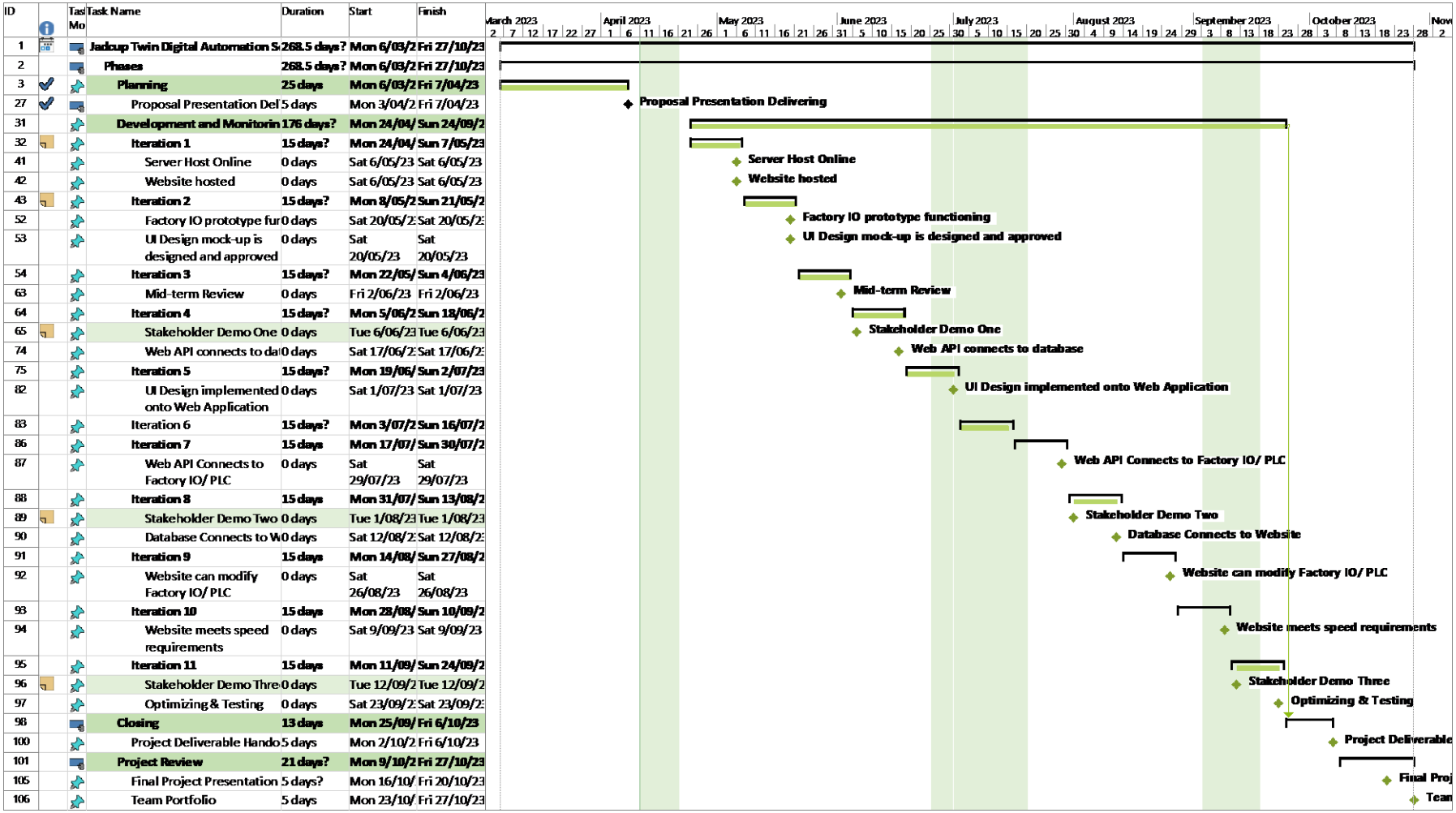
8.1 MILESTONE REPORT

Milestone	Iteration	Date	Status	Issues/Comments
Project Proposal Deadline		31/03/2023	Completed	
Proposal Presentation		07/04/2023	Completed	
Upskilling completed		23/04/2023	Running	
Server Host Online	1	06/05/2023	Running	
Website hosted	1	06/05/2023	Running	
UI Design mock-up is designed and approved	2	20/05/2023	Running	
Factory IO prototype working	2	20/05/2023	Running	
Stakeholder Demo one		After Iteration 3		Specific date to be scheduled with stakeholders.
Mid-term Review	3	02/06/2023	Running	
Web API connects to database	4	17/06/2023	Running	
UI Design implemented onto web host	5	1/07/2023	Running	
Web API Connects to Factory IO/ PLC	7	29/07/2023	Running	
Stakeholder Demo Two		After Iteration 7		Specific date to be scheduled with stakeholders.
Database Connects to Website	8	12/08/2023	Running	
Website can modify Factory IO/ PLC	9	26/08/2023	Running	
Website meets speed requirements	10	9/09/2023	Running	

Stakeholder Demo Three		After Iteration 10		Specific date to be scheduled with stakeholders.
Optimization & Testing	11	23/09/2023	Running	
Project Deliverable Handover		S2 Week 11	Running	Exact deadline unspecified
Client Feedback		S2 Week 12	Running	Exact deadline unspecified
Mentor Feedback		S2 Week 13	Running	Exact deadline unspecified
Team Poster		S2 Week 13	Running	Exact deadline unspecified
Team Portfolio		S2 Week 14	Running	Exact deadline unspecified
Reflective Report		S2 Week 14	Running	Exact deadline unspecified

8.2 PROJECT SCHEDULE

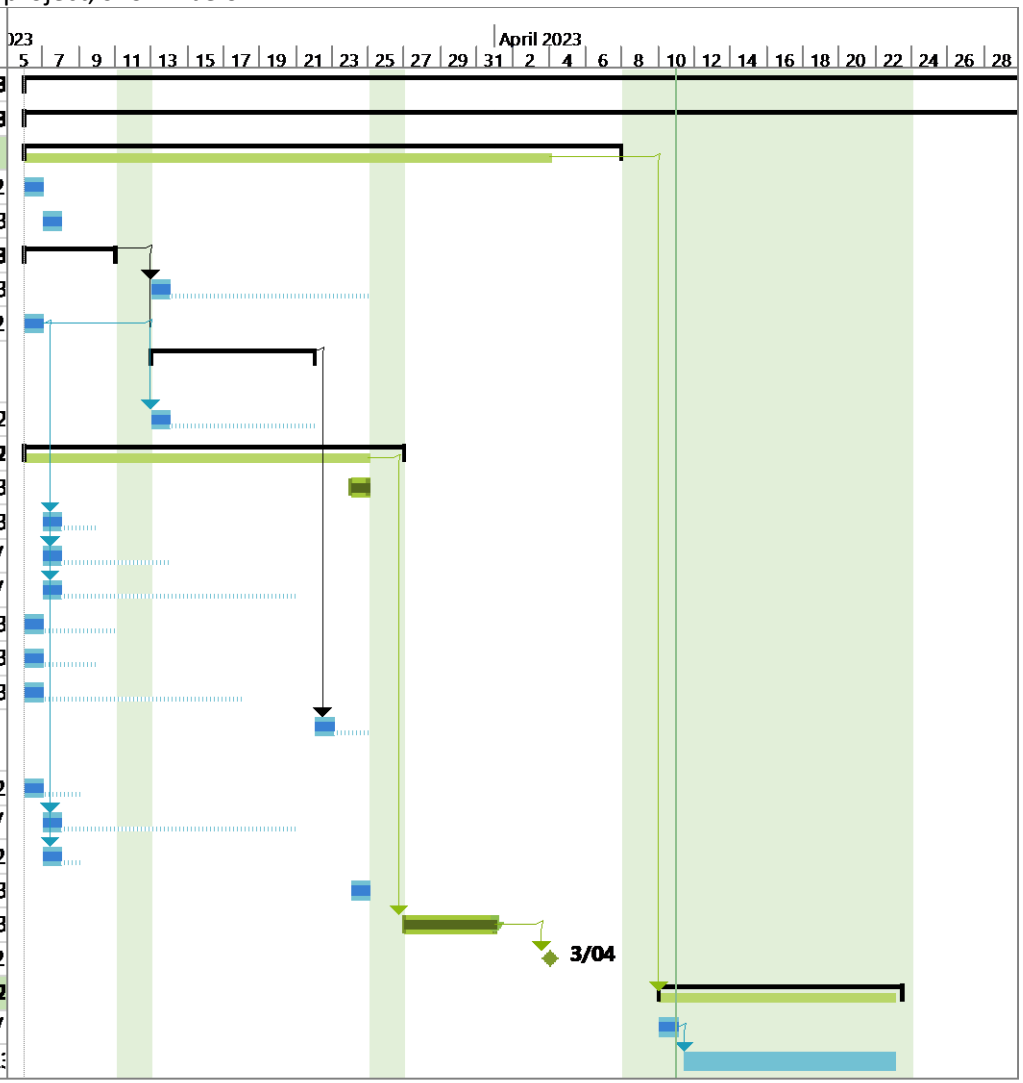
Sections highlighted in green mark the mid-term break for semester 1, semester break, and the mid-term break for section 2. Below is a summary of the entire year, with milestones to establish key stages of progress.

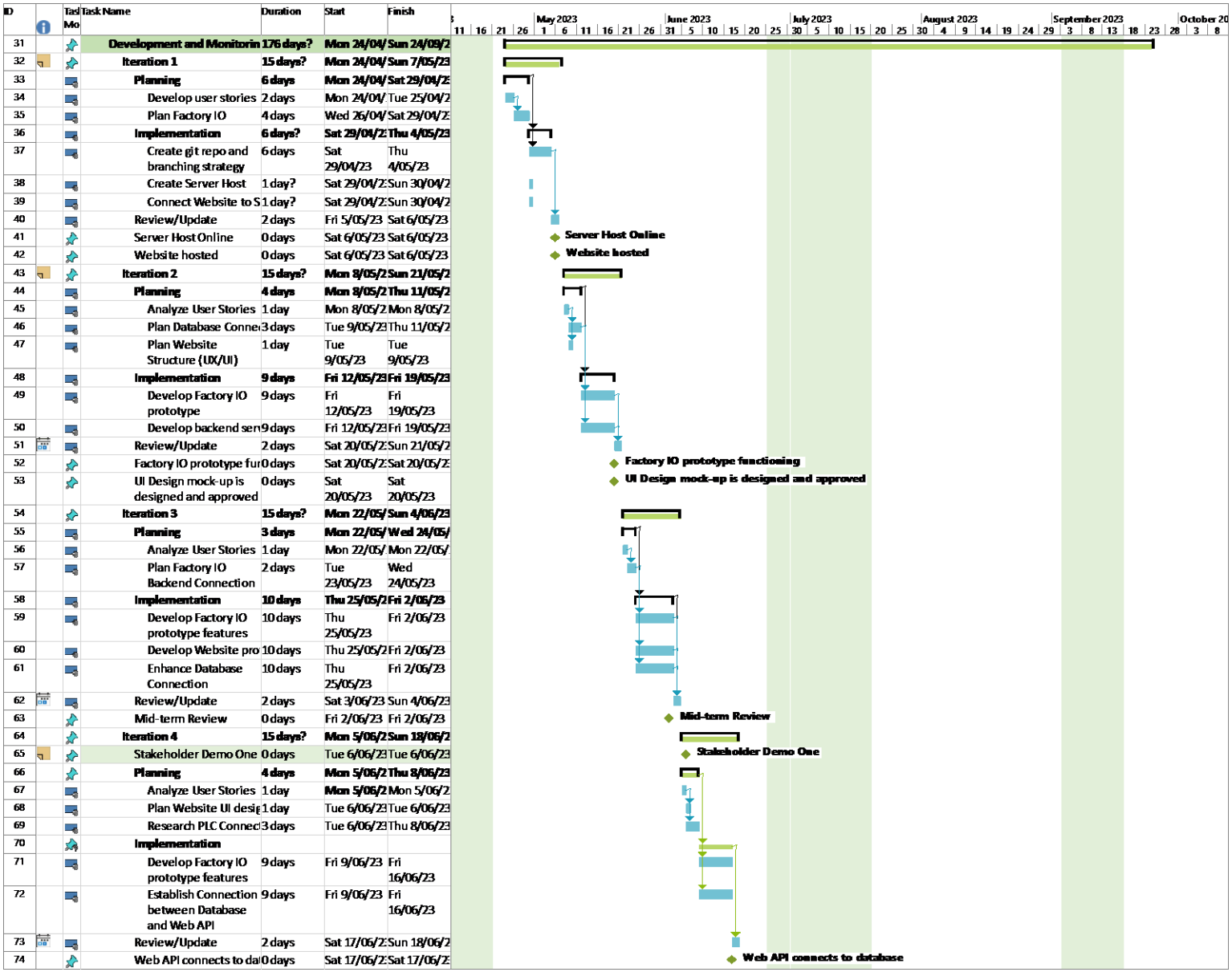


ID	Task Mo	Task Name	Duration	Start	Finish
1		**Jadcup Twin Digital Automation S**	**268.5 days?**	**Mon 6/03/2**	**Fri 27/10/23**
2		**Phases**	**268.5 days?**	**Mon 6/03/2**	**Fri 27/10/23**
3		**Planning**	**25 days**	**Mon 6/03/2**	**Fri 7/04/23**
27		Proposal Presentation Del	5 days	Mon 3/04/2	Fri 7/04/23
31		**Development and Monitorin**	**176 days?**	**Mon 24/04/**	**Sun 24/09/2**
32		**Iteration 1**	**15 days?**	**Mon 24/04/**	**Sun 7/05/23**
41		Server Host Online	0 days	Sat 6/05/23	Sat 6/05/23
42		Website hosted	0 days	Sat 6/05/23	Sat 6/05/23
43		**Iteration 2**	**15 days?**	**Mon 8/05/2**	**Sun 21/05/2**
52		Factory IO prototype fur	0 days	Sat 20/05/2	Sat 20/05/2
53		UI Design mock-up is designed and approved	0 days	Sat 20/05/23	Sat 20/05/23
54		**Iteration 3**	**15 days?**	**Mon 22/05/**	**Sun 4/06/23**
63		Mid-term Review	0 days	Fri 2/06/23	Fri 2/06/23
64		**Iteration 4**	**15 days?**	**Mon 5/06/2**	**Sun 18/06/2**
65		Stakeholder Demo One	0 days	Tue 6/06/23	Tue 6/06/23
74		Web API connects to da	0 days	Sat 17/06/2	Sat 17/06/2
75		**Iteration 5**	**15 days?**	**Mon 19/06/**	**Sun 2/07/23**
82		UI Design implemented onto Web Application	0 days	Sat 1/07/23	Sat 1/07/23
83		Iteration 6	15 days?	Mon 3/07/2	Sun 16/07/2
86		**Iteration 7**	**15 days**	**Mon 17/07/**	**Sun 30/07/2**
87		Web API Connects to Factory IO/ PLC	0 days	Sat 29/07/23	Sat 29/07/23
88		**Iteration 8**	**15 days**	**Mon 31/07/**	**Sun 13/08/2**
89		Stakeholder Demo Two	0 days	Tue 1/08/23	Tue 1/08/23
90		Database Connects to W	0 days	Sat 12/08/2	Sat 12/08/2
91		**Iteration 9**	**15 days**	**Mon 14/08/**	**Sun 27/08/2**
92		Website can modify Factory IO/ PLC	0 days	Sat 26/08/23	Sat 26/08/23
93		**Iteration 10**	**15 days**	**Mon 28/08/**	**Sun 10/09/2**
94		Website meets speed requirements	0 days	Sat 9/09/23	Sat 9/09/23
95		**Iteration 11**	**15 days**	**Mon 11/09/**	**Sun 24/09/2**
96		Stakeholder Demo Thre	0 days	Tue 12/09/2	Tue 12/09/2
97		Optimizing & Testing	0 days	Sat 23/09/2	Sat 23/09/2
98		**Closing**	**13 days**	**Mon 25/09/**	**Fri 6/10/23**
100		Project Deliverable Hando	5 days	Mon 2/10/2	Fri 6/10/23
101		**Project Review**	**21 days?**	**Mon 9/10/2**	**Fri 27/10/23**
105		Final Project Presentation	5 days?	Mon 16/10/	Fri 20/10/23
106		Team Portfolio	5 days	Mon 23/10/	Fri 27/10/23

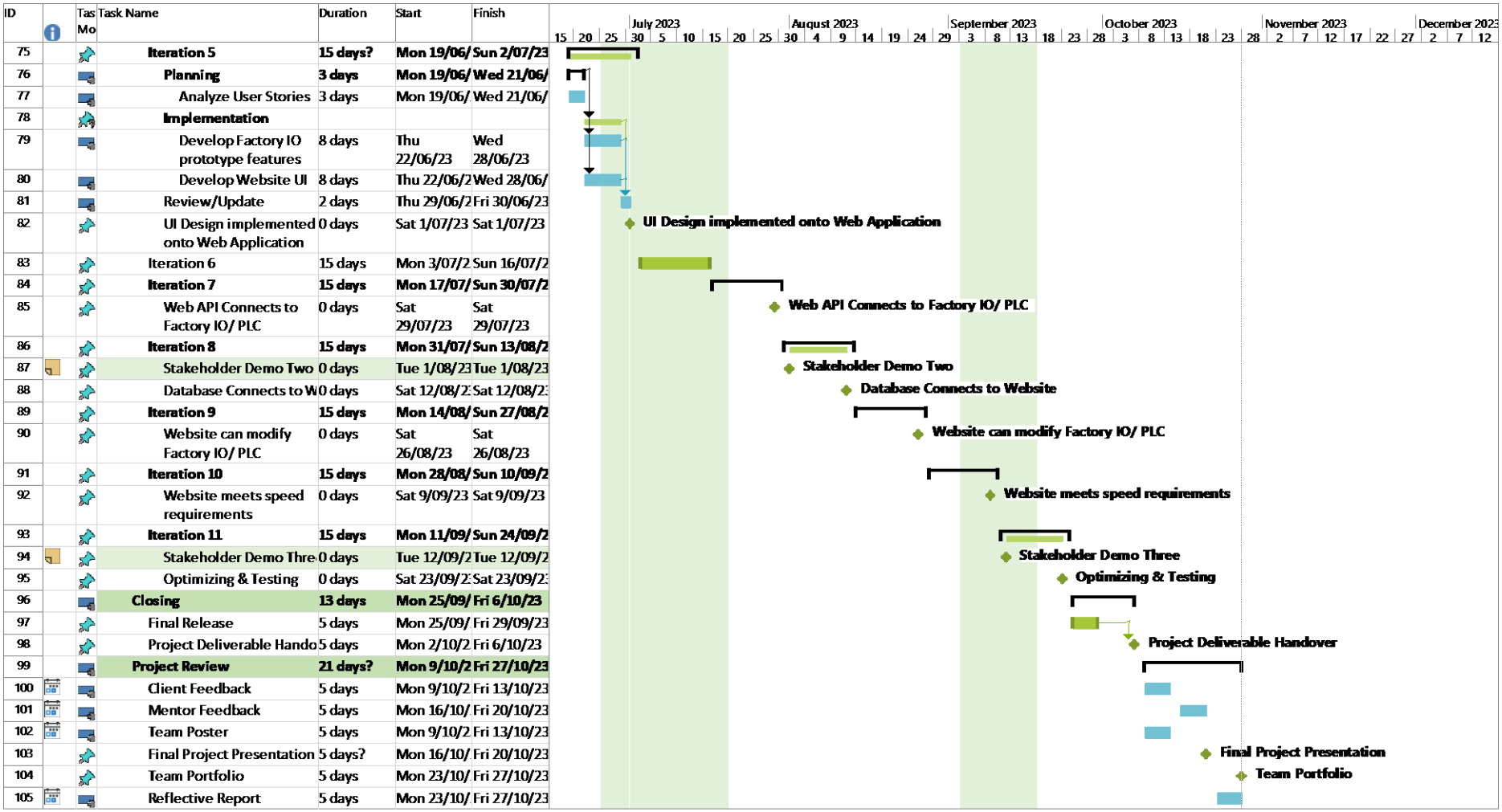
The two pages below go more in-depth into the different stages of the project, shown below.

ID	Tas Mo	Task Name	Duration	Start	Finish
1		**Jadcup Twin Digital Automation S**	**268.5 days?**	**Mon 6/03/2**	**Fri 27/10/23**
2		**Phases**	**268.5 days?**	**Mon 6/03/2**	**Fri 27/10/23**
3		**Planning**	**25 days**	**Mon 6/03/2**	**Fri 7/04/23**
4		Team Meeting	1 day	Mon 6/03/2	Mon 6/03/2
5		Mentor Meeting	1 day	Tue 7/03/23	Tue 7/03/23
6		**Discover Requirements**	**5 days**	**Mon 6/03/2**	**Fri 10/03/23**
9		Research Tech Stack	13 days	Mon 13/03/	Fri 24/03/23
10		Client Meeting	1 day	Mon 6/03/2	Mon 6/03/2
11		**Identify Project Management**	**10 days**	**Mon 13/03/23**	**Tue 21/03/23**
12		Establish User Stories	10 days	Mon 13/03/	Tue 21/03/2
13		**Complete Project Proposa**	**16 days**	**Mon 6/03/2**	**Sun 26/03/2**
14		Executive Summary	1 day	Fri 24/03/23	Fri 24/03/23
15		Terms of Reference	3 days	Tue 7/03/23	Thu 9/03/23
16		Project Rationale	8 days	Tue 7/03/23	Mon 13/03/
17		Objective and Scope	16 days	Tue 7/03/23	Mon 20/03/
18		Skills Analysis	5 days	Mon 6/03/2	Fri 10/03/23
19		Team Roles	4 days	Mon 6/03/2	Thu 9/03/23
20		Team Schedule for Part	13 days	Mon 6/03/2	Fri 17/03/23
21		Project Management Methodology	3 days	Wed 22/03/23	Fri 24/03/23
22		Risk and Issues Manage	3 days	Mon 6/03/2	Wed 8/03/2
23		Project Schedule	16 days	Tue 7/03/23	Mon 20/03/
24		Cost Estimate	2 days	Tue 7/03/23	Wed 8/03/2
25		Disclaimer	1 day	Fri 24/03/23	Fri 24/03/23
26		Proposal Review and Subn	5 days	Mon 27/03/	Fri 31/03/23
27		Proposal Presentation Del	1 day	Mon 3/04/2	Mon 3/04/2
28		**Upskilling**	**15 days**	**Mon 10/04/**	**Sun 23/04/2**
29		Project Analysis and Deleg	1 day	Mon 10/04/	Mon 10/04/
30		Upskilling	13 days	Tue 11/04/2	Sat 22/04/2

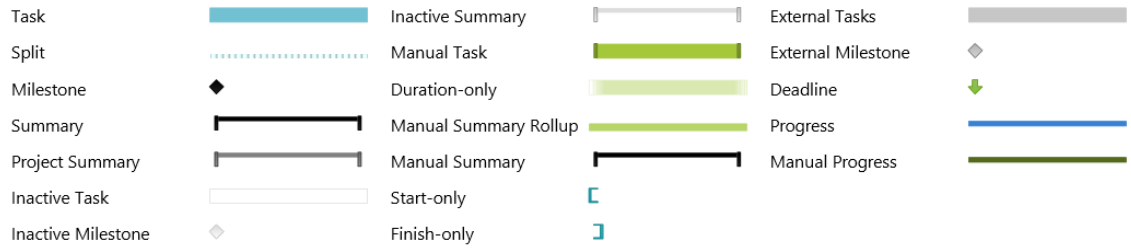




ID	Task Mo	Task Name	Duration	Start	Finish
31		**Development and Monitorin**	**176 days?**	**Mon 24/04/**	**Sun 24/09/2**
32		**Iteration 1**	**15 days?**	**Mon 24/04/**	**Sun 7/05/23**
33		**Planning**	**6 days**	**Mon 24/04/**	**Sat 29/04/2**
34		Develop user stories	2 days	Mon 24/04/	Tue 25/04/2
35		Plan Factory IO	4 days	Wed 26/04/	Sat 29/04/2
36		**Implementation**	**6 days?**	**Sat 29/04/2**	**Thu 4/05/23**
37		Create git repo and branching strategy	6 days	Sat 29/04/23	Thu 4/05/23
38		Create Server Host	1 day?	Sat 29/04/2	Sun 30/04/2
39		Connect Website to S	1 day?	Sat 29/04/2	Sun 30/04/2
40		Review/Update	2 days	Fri 5/05/23	Sat 6/05/23
41		Server Host Online	0 days	Sat 6/05/23	Sat 6/05/23
42		Website hosted	0 days	Sat 6/05/23	Sat 6/05/23
43		**Iteration 2**	**15 days?**	**Mon 8/05/2**	**Sun 21/05/2**
44		**Planning**	**4 days**	**Mon 8/05/2**	**Thu 11/05/2**
45		Analyze User Stories	1 day	Mon 8/05/2	Mon 8/05/2
46		Plan Database Conne	3 days	Tue 9/05/23	Thu 11/05/2
47		Plan Website Structure (UX/UI)	1 day	Tue 9/05/23	Tue 9/05/23
48		**Implementation**	**9 days**	**Fri 12/05/23**	**Fri 19/05/23**
49		Develop Factory IO prototype	9 days	Fri 12/05/23	Fri 19/05/23
50		Develop backend ser	9 days	Fri 12/05/23	Fri 19/05/23
51		Review/Update	2 days	Sat 20/05/2	Sun 21/05/2
52		Factory IO prototype fur	0 days	Sat 20/05/2	Sat 20/05/2
53		UI Design mock-up is designed and approved	0 days	Sat 20/05/23	Sat 20/05/23
54		**Iteration 3**	**15 days?**	**Mon 22/05/**	**Sun 4/06/23**
55		**Planning**	**3 days**	**Mon 22/05/**	**Wed 24/05/**
56		Analyze User Stories	1 day	Mon 22/05/	Mon 22/05/
57		Plan Factory IO Backend Connection	2 days	Tue 23/05/23	Wed 24/05/23
58		**Implementation**	**10 days**	**Thu 25/05/2**	**Fri 2/06/23**
59		Develop Factory IO prototype features	10 days	Thu 25/05/23	Fri 2/06/23
60		Develop Website pro	10 days	Thu 25/05/2	Fri 2/06/23
61		Enhance Database Connection	10 days	Thu 25/05/23	Fri 2/06/23
62		Review/Update	2 days	Sat 3/06/23	Sun 4/06/23
63		Mid-term Review	0 days	Fri 2/06/23	Fri 2/06/23
64		**Iteration 4**	**15 days?**	**Mon 5/06/2**	**Sun 18/06/2**
65		Stakeholder Demo One	0 days	Tue 6/06/23	Tue 6/06/23
66		**Planning**	**4 days**	**Mon 5/06/2**	**Thu 8/06/23**
67		Analyze User Stories	1 day	Mon 5/06/2	Mon 5/06/2
68		Plan Website UI desi	1 day	Tue 6/06/23	Tue 6/06/23
69		Research PLC Connec	3 days	Tue 6/06/23	Thu 8/06/23
70		**Implementation**			
71		Develop Factory IO prototype features	9 days	Fri 9/06/23	Fri 16/06/23
72		Establish Connection between Database and Web API	9 days	Fri 9/06/23	Fri 16/06/23
73		Review/Update	2 days	Sat 17/06/2	Sun 18/06/2
74		Web API connects to da	0 days	Sat 17/06/2	Sat 17/06/2



ID	Task Mo	Task Name	Duration	Start	Finish
75		**Iteration 5**	**15 days?**	**Mon 19/06/**	**Sun 2/07/23**
76		**Planning**	**3 days**	**Mon 19/06/**	**Wed 21/06/**
77		Analyze User Stories	3 days	Mon 19/06/	Wed 21/06/
78		**Implementation**			
79		Develop Factory IO prototype features	8 days	Thu 22/06/23	Wed 28/06/23
80		Develop Website UI	8 days	Thu 22/06/2	Wed 28/06/
81		Review/Update	2 days	Thu 29/06/2	Fri 30/06/23
82		UI Design implemented onto Web Application	0 days	Sat 1/07/23	Sat 1/07/23
83		Iteration 6	15 days	Mon 3/07/2	Sun 16/07/2
84		**Iteration 7**	**15 days**	**Mon 17/07/**	**Sun 30/07/2**
85		Web API Connects to Factory IO/ PLC	0 days	Sat 29/07/23	Sat 29/07/23
86		**Iteration 8**	**15 days**	**Mon 31/07/**	**Sun 13/08/2**
87		Stakeholder Demo Two	0 days	Tue 1/08/23	Tue 1/08/23
88		Database Connects to W	0 days	Sat 12/08/2	Sat 12/08/2
89		**Iteration 9**	**15 days**	**Mon 14/08/**	**Sun 27/08/2**
90		Website can modify Factory IO/ PLC	0 days	Sat 26/08/23	Sat 26/08/23
91		**Iteration 10**	**15 days**	**Mon 28/08/**	**Sun 10/09/2**
92		Website meets speed requirements	0 days	Sat 9/09/23	Sat 9/09/23
93		**Iteration 11**	**15 days**	**Mon 11/09/**	**Sun 24/09/2**
94		Stakeholder Demo Thre	0 days	Tue 12/09/2	Tue 12/09/2
95		Optimizing & Testing	0 days	Sat 23/09/2	Sat 23/09/2
96		**Closing**	**13 days**	**Mon 25/09/**	**Fri 6/10/23**
97		Final Release	5 days	Mon 25/09/	Fri 29/09/23
98		Project Deliverable Hando	5 days	Mon 2/10/2	Fri 6/10/23
99		**Project Review**	**21 days?**	**Mon 9/10/2**	**Fri 27/10/23**
100		Client Feedback	5 days	Mon 9/10/2	Fri 13/10/23
101		Mentor Feedback	5 days	Mon 16/10/	Fri 20/10/23
102		Team Poster	5 days	Mon 9/10/2	Fri 13/10/23
103		Final Project Presentation	5 days?	Mon 16/10/	Fri 20/10/23
104		Team Portfolio	5 days	Mon 23/10/	Fri 27/10/23
105		Reflective Report	5 days	Mon 23/10/	Fri 27/10/23



Note: each iteration contains phases for planning, implementation, review and updating, and can be assigned dynamically every week as we progress. The iterations do not set restrictions or deadlines and focuses on creating a reliable structure the team can plan with to complete milestones. Additionally, the tasks identified so far are hypothetical suggestions based on the milestones, and are flexible for change during the development of the project.

Iteration planning phase:

Discuss standup meetings, and how they impact upcoming iteration
Discuss how we pull cards out of backlog based from decisions made from the standup meetings

Implementation phase:

Complete the cards we pulled out from standup meetings, or start working

Review phase:

We confirm with the client or mentor that they are happy with the progress made, and check if there are any changes required before marking card as complete/continuing development

Update phase:

If applicable, the changes requested in the review phase are implemented during this short phase before continuing to next iteration. If the change scope is out of reach to complete within this phase, the change is converted into a new card which is set to be the highest priority to complete in the next iteration.

8.3 MEETING MINUTES

Meeting type	**Date**	**Time**	**Location**
Team meeting	Monday 6th March	6pm	WZ701
Mentor meeting	Tuesday 7th March	11am	WZ1101
Mentor meeting	Tuesday 14th March	10am	WZ1101
Team meeting	Tuesday 14th March	11am	WZ601
Client meeting	Thursday 16th March	1pm	Jadcup headquarters
Team meeting	Saturday 18th March	9am	Online
Team meeting	Tuesday 21st March	9.30am	WZ701
Mentor meeting	Tuesday 21st March	10am	WZ1101
Team meeting	Tuesday 21st March	11am	WZ1101
Team meeting	Saturday 25th March	9am	Online
Mentor meeting	Tuesday 27th March	10am	WZ1101

Team meeting	Tuesday 27th March	11am	WZ701

Risk Register for Jadcup Digital Factory Automation

Date: 13-Mar

No.	Threat	Risk	Description	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact	Status
1	Medium	Unclear Requirements	The design and necessary features of the project may be and remain unclear throughout the project	The project is as new for the client as it is for the team		Keep open and clear communication with the client's requirements	Client Liaison	Medium	High	Watching
2	Low-Medium	Incorrect work performed	With uncertainty of outcomes or requirements, work may be wasted in unnecessary areas	Unclear, undocumented team direction		Clarify work with the team, mentor, and client regularly	Team Manager	Low	Medium	Unresolved
3	Medium	Scope Creep	The work required for the project may gradually increase and overwhelm the team	Designing or discussing features that don't align with the prototype MVP		Check each feature aligns with the MVP	Whole Team	High	High	Unresolved
4	Low-Medium	Mismanaged feature prioritization	Features incorrectly prioritized may result in delays	Poorly managed and discussed features. Improper		Weekly meetings to discuss	Whole Team	Medium	Low	Unresolved

			if one system requires on another, undelivered system (AKA, the web frontend relying on the backend & database)	discussion on slow-to-implement systems		prioritization and delays				
5	Low-Medium	Poor product design	The tech stack or user design may be irrelevant to the prototype's end goals	Research and discussion not properly performed between the team and client		Be as thorough as the team can be with tech stack research and keep discussion open with the team.	Whole Team	Low	Medium	Watching
6	High	Poor communication with the client	Poor communication with clients can impact on the final project as well as the team, which can lead to problems like dissatisfied clients and pressure for the team.	The client's naive language might not be English, which can lead to difficulties in understanding what the clients wants and does not want as well as the team not able to give an answer or question related to the project. Additionally, client	If the client gets back to us with a slew of changes, we know this risk has become an issue.	Send demos and reviews to the client for them to get back to us. Any feedback from the client can be baked into our change management through iteration or design planning. In the meantime, get assistance from our mentor	Whole Team	High	High	Unresolved

				communication is minimal and client meetings are very infrequent, lacking feedback or discussion about the project.		regarding requirements and feedback as part of our change management.				
7	Medium	poor time-management	The standard and quality of the prototype could be impacted by improper time management for the entire project.	Not finishing task on schedule can lead to more time catching up		Finish your work and task on schedule. Stay focus.	Whole Team	High	Medium	Watching
8	Low-Medium	Workplace injury (Health and Safety)	Not knowing the health and safety hazards of the environment	The project in a factory environment is a first for most of the team		Researching and learning the basic health and safety rule of the workshop	Whole Team	Low	High	Watching
9	High	Legal Issue	The clients need to tell the team their legal or regulation requirements, so that we don't break the client requirements	By not knowing the client's legal or regulation requirements		Ask the client if there is any legal or regulation requirements	Team Manager	Low	High	Watching

10	high	Poor communication with other members of the team	Not letting your team members know about problems or struggles that you are having.	Trying to solve issue/problem yourself, or struggling in finishing up your task		Letting the team know your issue or problems that you may been having.	Whole Team	Medium	Medium	Watching
11	high	team member unavailable to participate	Someone might have had an issue come up or some sort of personal reason that is preventing them from coming.	Overwhelmed with everyday problem/obstacl e		Let the team and mentor know about the situation.	Team Members	Medium	Low	Watching
12	Medium-high	lockdown happening again	Another global pandemic issue according	Not caring about yourself or others during pandemic		Staying home when you are not feeling well	Whole Team	Low	High	Watching
13	high	data loss	Not being able to save or back up the data from PCLs	PCs we store information break, we lose power, etc.		Test and trial before implementing it into the prototype	Whole Team	Medium	Medium	Watching
14	high	Requirement beyond scope	Trying to implement a key feature which will take more time to build than what the team	client negotiation, work delegation, or change management plans.		Breaking down the feature to thorough analysis the difficulty of the project	Whole team	high	high	watching

15	Medium	Data availability	Docs that we requested from the client in risk register	Might be due to the client policy			Whole team	Medium	Medium	watching

9.2 ISSUE LOG

#	Issue Description	Impact on Project	Report date	Reported by	Assigned To	Priority (M/H/L)	Resolve date	Status	Comments
1	Late team formation	Reduced amount of time for the project by a week	27/3/23	Jane	Jane	High		Active	Will be working during weekends if necessary
2	Scheduling issue	Everyone's timetables during the week clash which prevents the team from having meetings	27/3/23	Jane	Everyone	High	13/3/23	Closed	Everyone agreed to meet during the weekend (Saturday)
3	Communication with client	Client prefers Mandarin over English which none of us can speak	28/3/23	Matthew	Matthew	Medium	16/3/23	Closed	Matthew acted as the translator during the meeting, and the client agreed to give us his email if we needed further clarifications
4	Behind schedule	Lack of information of the project delayed the start of the proposal	14/3/23	Matthew	Jane	High	15/3/23	Active	Wrote out the proposal's template, will be updating regularly with completed plans
5	Lots of requirements	The project requires lots of different components that all require different skillsets	16/3/23	Jane	Everyone	High	20/4/23	Active	Will designate tasks between members based on their skills, as we

									may start upskilling before submitting proposal. Will designate tasks between members based on their skills, as we will start upskilling before submitting proposal.
6	Insufficient amount of work being completed	Saturday meetings aren't enough for us to be productive as meeting online during the weekend reduces motivation	29/4/23	Myles	Everyone	High	29/4/23	Resolved	Additional meeting on Tuesday
7	Yeran got into accident	Yeran got into an accident on the motorway this meeting, and he had the prototype we were supposed to show Matthew	9/5/23	Yeran	Yeran	Medium-High	9/5/23	Resolved	He's fine
8	Daniel isn't responding	We wanted to get the source code of the machines but Daniel hasn't replied even after we sent him several emails, which caused us to start worrying whether we would be able to receive proper feedback from him at all	25/5/23	Myles	Daniel & Team	High		Active	He's still not responding
9	Incorrect FactoryIO licenses	We were initially provided with starter edition FactoryIO licenses which only provided drivers for control i/o, proprietary software for interfacing with the factory scenes. It	25/5/23	Yeran & Joshua	Team & Matthew	High	11/5/23	Resolved	We have been provided a Factory I/O Ultimate Edition license by AUT, which means we can play around with the factory scenes using whatever library and unblocks future work.

		meant we didn't have a way to interface with factory scenes with javascript modbus library. We used trial accounts but these are locked after one month, and locked to hardware.							
10	Tonys ghosting as well								

9.3 QUALITY ASSURANCE PLAN

1. Abstract

1.1. Introduction

While developing a dashboard for integration in industrial settings, it is vital that we ensure that there are no issues that could cause damage to the product, environment, or equipment involved. To prevent this, we must implement a consistent process of testing and review that we'll ensure we detect potential issues before they are introduced into a production environment.

1.2. Purpose

The follow is a plan to establish guidelines to Assure the quality of any code developed for this project. Defining the expectations for the testing of any developed feature or code will assist in ensuring that any problems that are found will be consistent and efficient.

1.3. Scope

The Quality Assurance Plan focuses on the testing of features as they are developed along with the documentation of those tests along with the process and documentation of code reviews and Q.A. Evaluation.

This plan is necessary to ensure that no feature we develop will cause damage to the warehouse environment that the system is to be deployed into and the machinery that it is implemented with.

1.4. Policy Statement

We will strive to produce an effective and quality product that will modernize Jadcup's production line.

In our endeavor to achieve this we will:

- Nurture a team culture that strives for quality.
- Ensure internal testing standards.
- Create an environment where feedback is encouraged.

- Record testing results for internal evaluation of current Q.A. procedures.
- Review of QA Metric results to identify common issues in development.
- Ensure that any requirements proved by the client are fulfilled.

2. Management

2.1. Organizational Structure

Members of the Quality Assurance and Technical Team report to Senior Management who further reports to the client. However, it is the responsibility of the Q.A./ Technical Team to develop, plan, and execute Testing and Reviews as they occur.

Because we are utilizing a Kanban methodology we will not be utilizing a QA Manager who would focus on developing acceptance tests and test code. Instead a member of the QA team will take on that role for a specific card on our Kanban board.

2.2. Roles and Responsibilities

2.2.1. Senior Management

Matthew Kuo will act as Senior Management for Quality Assurance, however in this role his main responsibilities will be acting as a third party for any issues that could arise in the process of Q.A. disputes.

2.2.2. Task Lead

Any task listed on the Kanban board will be assigned a task leader who is responsible for delivering the item for testing and review, this member will not be able to act as a member of the QA Team for that specific item.

2.2.3. Quality Assurance/ Technical Team

The Q.A. and Technical Team will be a single team made up of the Development team.

2.2.3.1. Quality Assurance Team

Quality Assurance Team members are responsible for the testing of cards and recording the results.

2.2.3.2. Technical Team

Technical Team members undertake the process of code walkthroughs and review the tests that have been done.

3. Q.A. procedure

The Key to Ensuring our Q.A. Procedures are moving smoothly is using our Kanban Board on Trello to visualize the process and make it easy to detect at a glance what should be prioritized. Utilizing the Trello environment, we are able to assign developers and add reports and other documents for ease of communication as well as allowing us to work remotely.

While typically the Q.A. process is underdone when members of the respective teams are available, should a test need to be expedited then the card can be colored red so it is visually marked as urgent additionally members of the team will be alerted by the Task Lead that this card must be done

quickly. A card may require an expedited timeline due to its importance to the stability of the project or because it is holding up further development of other cards. Any card that is marked to be expedited will not be counted against the flow rate of the systems.

3.1. Testing

Once a developer has finished working on a given card the developer utilizes the Kanban Board to move the card from the *In Development* column to the *Waiting for Testing* column, adding a link to the corresponding push request through Github. Once the card has been moved to the new column a member of the Q.A. Team can be added to create unit tests, integration tests, and/or, Functional tests depending on the requirements of a given card. Any test that is used should be run several times to ensure consistency.

Should the tests fail, the card will be moved back into the In Development Column, the push request will be declined, and the Task Lead will be notified. Should the Testing succeed, the card will be moved into the *In Review* column.

Once testing has concluded then a Testing Results matrix will be attached to the card as outlined in *4.1 Testing Matrices.* And records any applicable data into the Metric Record as outlined in *4.3. Q.A. Metric Record.*

3.2. Code Walkthrough

The Code walkthrough is key in ensuring that testing and coding standards are being upheld but also can act as a teaching tool where members of the development team they are less knowledgeable about a topic can learn.

When a Card has been tested and has moved into the *In Review* Column then a review of both the card and code is undertaken, the review process is underdone by members of the technical team that did not work on developing the card, this includes the member of the Q.A. team that produced the tests.

In the case where there are no eligible members of the technical team remaining, the walkthrough will be done with a selection of the Developer team that feels they did not work on enough amount of the feature to be biased in any way.

During the Walkthrough process the Technical Team will review both the code of the feature and testing that has been done. Should the team find that more tests are required then the card will return to the *In Testing* Column. If the team determines that the code overreaches the scope of a given card, then they will reach out to the Task Lead for justification. If this Justification is not sufficient then the card will be moved back to the *In Development* column. Otherwise once review is completed the Technical Team will move the card to the *Release* column.

Once the code walkthrough has concluded for any of the above reasons a Code Review document will be attached to the given card as outlined in *4.2. Code Review Reports*.

3.3. Q.A. Evaluations

Evaluations are the processes we utilize to ensure the products overall health, through the review of the Q.A. Metric Record as an overview of testing results and any key or repeating issues can be found and a discussion on possible changes can be undertaken.

Evaluation reviews of the Metric Record, along with any key issues that are detected and collated into an Evaluation Report such that it can be used as evidence and or justification in a change request.

4. Required Documentation

4.1. Testing Matrices

Testing Matrices are utilized to record and display the testing that has occurred on a given card. These matrices are designed so that anyone who views them can quickly review the tests that have occurred, test description, the expected outcomes, and the actual outcome. Along with dates and who developed the tests.

Test type	Test #	Test Description	Expected outcome	Actual Outcome	Test Pass
Unit Test		Testing for correct read of machine output			Test fail
	1		159	28	Fail
	2		159	28	Fail
	3		159	28	Fail

4.2. Code Review Reports

Code Review Documents are used to identify issues and determine any weak points in our testing and development cycles. The document records any notes that the Technical Team has regarding any aspect of either tests used or code written. The document is key in detailing any reason that a card has failed the walkthrough and may be shorter if there are no such issues.

4.3. Q.A. Metric Record.

The Metric Record is used to collate and record results from Testing matrices and Review Reports. The document acts as a living document that allows us to at a glance determine any issues that occur. The metrics recorded and the reasons for them can be found in *6.0 Quality Assurance Metrics.*

4.4. Evaluation Reports

Evaluation Reports are Documents that record evaluation sessions and document what metrics and reoccurring issues have been detected, detailing specific instances if required. The document is detailed just that it can be used as evidence in a Change Request Form and should be comprehensive enough to not require someone to review individual Review Reports and Test Matrices.

5. Problem Reporting Procedures

5.1. Noncompliance Reporting procedure

Should a member of the Q.A. Team or Technical Team find that cards are being repeatedly submitted without the required changes having been resolved then that member should bring this to attention in the following meeting and a discussion must be made regarding why the changes are not being made.

Should this become a recurring issue the problem will be escalated to Senior Management members for assistance in finding an amicable resolution.

6. Quality Assurance Metrics

The following Section outlines what Metrics we will use to gage the effectiveness of our QA process and assist in identifying repeating issues.

6.1. Test Coverage

6.1.1. Test Execution

Test Execution measures the number of tests we run and their results. Results are classified as pass or fail. Can be used alongside other metrics or alone.

6.1.2. Tests Per Card

TPC Measures how many tests we are running on a given card, this metric allows us to identify if we are under/over testing our cards. Can be used with other metrics or alone.

6.1.3. Bug per card

This metric allows us to find how effective our methods are at detecting bugs in a system. It is important to know that this metric does not need to be high but should act as a tool alongside other metrics such as test Execution and TPC to measure quality in our testing.

6.2. Bug distribution

Bug Distribution Allows us to identify common sources of issues such as integration, performance, security, unit level, eta. By measuring this distribution, we can identify weak points in our skills and determine where we should be putting extra focus during our testing.

7. Quality Standards

7.1. Definition of Ready (DoR)

The Definition of Ready defines how we decide if a user story is viable to start developing on. By using this metric, we can guarantee that we are not starting work that we cannot finish. For example, if a user story requires a database to be running and the story that develops the database has not been finished, then that user story should not start.

7.2. Definition of Done (DoD)

The Definition of Done is a metric used to ensure that all developers understand what an outcome should look like when deciding if their systems are ready to be considered finalized. By using this metric, we can standardize the both the testing and review stages to ensure that there are no discrepancies between user stories as well as ensure that user stories are not entering testing and review stages before they should.

When considering if a user story meets the DoD, then we must consider the following for the following stages:

- Development:
    - The code only affects necessary systems.
    - Developers have reasonable expectations that the code will pass testing and review.
    - Code achieves user story.
- Testing:
    - The code passes unit tests.
    - The code passes integration tests.
    - Test matrix filled and attached to card.
    - Matrix added to metric record.
- Review:
    - Code fulfills programming standards.
    - The code is well documented.
    - Testing done is relevant and effective.
    - Evaluation report written and attached to card.