SDS: Assessment Task 2

Process Assessment

Tutor:
Tutorial: Wed 02
Student:

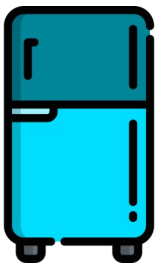


Table of contents

1 The Agile process

The Agile process is an iterative software development and project management methodology that breaks down software projects into short reflective sprints. This process enables teams to deliver quality working software faster while keeping the development project flexible and agile. The process highly values reflective thinking to further evolve and refine the process throughout the project so that it can remain highly adaptable to all situations. One of the most common agile methodologies used is called “Scrum” which my team adopted for this project.

2 Scrum workflow and process

Scrum is a framework for project management that works in iterative cycles called sprints to achieve a well defined goal. There are designated roles that formulate the structure of the scrum team: The Product Owner who communicates between the developers and the customer, The Scrum master who ensures scrum best practices are being followed, and The Development Team who engages in scrum meetings to deliver the final product.

2.1 Artifacts:

Product Backlog:

The product backlog is a prioritised list of tasks (see Appendix A) created by the Product Owner that describes the features that need to be completed by the development team. The list of tasks are derived from user stories and the product requirements gathered from the client.

We chose to use JIRA to organise and categorise the backlog as it made the process of creating, assigning, prioritising each task much easier. The product owner role for our project was a joint effort between the team and co-team leads. We utilised Confluence to create our initial user stories, leveraging its collaborative ability to work on creating user stories as a group. This allowed us to have much more quality user stories resulting in a quality product backlog.

Sprint Backlog:

The sprint backlog is a small selective list of tasks that is to be completed over the sprint time period (see Appendix B). The tasks are chosen from the product backlog during the “Sprint Planning” ceremony (see section 2.4).

We once again utilised JIRA for this as it is perfect for handling this use case. We first created sprint categories in an incremental order which we could then distribute relevant tasks to. We then assigned these tasks to relevant members of the group. We had no particular system when assigning tasks, it was mainly based on personal interest.

User Stories:

User stories describe the functionality of a product backlog item or feature by understanding it from the end-users perspective. This allows the team to understand how a specific feature will provide value to the end-user.

The team chose to write the initial user stories together. It meant that we had a far wider and more quality selection of user stories to use. Our Business Analyst (BA) collated these and formulated a more finalised version (see Appendix H). Each one of these user stories has associated Acceptance Criteria (see Appendix H second photo) to compare against to ensure the software produced was meeting the correct expectations.

2.2 Extended Artifacts:

Meeting minutes:

Meeting minutes are a record of a meeting’s time and the discussion points raised within the meeting. Usually an outline is created detailing these discussion points.

We choose to transcribe each meeting to have a more detailed record (see Appendix C).

Defects:

An issue or bug within the software that is broken. This may result in parts of the development becoming blocked as the issue needs to be resolved first.

We ran into an issue regarding the access to real data from the IOT device as the firewall disallowed any outside connections to the websocket server. It meant that we couldn’t use any real data. This was an issue that had to be resolved by our client and UTS IT. To overcome this issue I set up a meeting with the researchers (client) to communicate the problem and try to find a compromise (see Appendix J). Within that meeting we set up a secure websocket connection which still ran into the problem of the firewall however in the off chance the firewall got lifted we’d be able to seamlessly transition into real data. This ended up being the case and once the firewall was lifted we already had all the connections set up and ready to go.

Burndown Chart:

A sprint burndown chart shows when tasks are created and completed. It is mainly used to identify if the sprint is progressing well. This is highlighted by the ratio between the amount of completed tasks vs the amount created.

Jira automatically creates this chart for us for each sprint. The chart for our initial sprint trended upwards (see Appendix D), highlighting our poor sprint planning and how we needed to better plan future sprints.

UI Mockups:

Ui mockups are a visual representation of the user interface for the application and help guide developers on how the application should look and feel.

We chose to utilise Figma for our UI mockups (see Appendix I). We had a UI team of 3 people create the design for each page of the application as well as mobile views. As a developer it really helped fast track development of frontend pages in the application as the design, layout and colour were already established.

2.3 Workflow:

Sprint:

Sprints are the main process of the “Scrum” methodology. They’re time boxed periods (2 weeks) whereby work from the relevant sprint backlog is to be completed within an established start and end date.

The Software Development Studio (SDS) subject outlined a sprint guide with five overall sprints, one being an initial sprint, three being development sprints and the fourth being a finalisation sprint. Our team followed this sprint outline however we needed to incorporate more development time within the final sprint. Within our project we utilised JIRA to track the progress of each sprint as it was easy to have all sprint related information centralised.

Testing:

Testing can be done in many forms, the most common are manual testing, unit test, and e2e tests.

In terms of development we conducted extensive manual testing as outlined in our merge requests (see Appendix G) where each dev QA tested their own feature. We also evaluated each feature against relevant user story acceptance criteria to ensure the software feature was meeting expectations.

Documentation:

In a development/software context documentation is the process of creating an informational record of a feature that usually describes how it works.

We documented and versioned our software code through the use of version control software called git, this is graphically displayed within gitlab. Gitlab allowed us to leverage the merge request/review feature. This allows for clear chronological documentation of each feature being implemented and merged within the main development branch (see Appendix G). Our BA also created a user manual detailing how each part of the web application works (see Appendix K).

2.4 Ceremonies:

Sprint planning:

Sprint planning is a “Scrum” ceremony performed at the beginning of the sprint whereby the team identifies what work can be achieved within the sprint time period (see Appendix E). After the meeting every team member should have a clear understanding of what work needs to be completed and how it will be delivered.

Our team conducted a sprint planning meeting for each new sprint cycle. We verbally discussed what needed to be completed and then documented this information within JIRA by creating specific detailed tasks for the sprint backlog.

Standup meetings:

Standup meetings also known as “Daily Scrum” are small daily meetings designed to inform the team of work that is or has been completed and any blockers members of the team may be facing.

Our team participated in standup every time we met for the SDS class, which was once a week.The development team did have some extra meetings whereby standup was once again performed on more of a technical level to ensure each developer was aware of which tasks were in progress. Standup meetings have a general set of questions that we followed these being: What did I work on yesterday? What am I working on today? What issues are blocking me?

Sprint Review:

Sprint reviews are meetings conducted to showcase and review work that has been completed throughout the sprint. Usually this involves meeting with stakeholders to gather feedback.

The development team showcased the work completed throughout the sprint every week in class to all the other team members. We also showcased the progress to the client through an MS Teams call to get their feedback.

Sprint Retrospective:

Sprint retrospectives are the final ceremony conducted at the end of each sprint. The purpose of this meeting is to reflect on the sprint as a whole, what worked and what didn't. The ceremony provides insight into what needs to be done to improve the overall effectiveness of each sprint.

Our team was pretty consistent in conducting a sprint retrospective in the final stages of each sprint. We utilised a different tool called Mural (see Appendix F) which is a web based platform designed to make sprint retrospectives easier. We had four questions: What went well? What went poorly? What ideas do you have? How should we take action?

3 Personal Reflection and Learnings

3.1 Agile Process Learnings:

I've learnt a lot about the agile process, and more specifically the day-to-day use of "Scrum" and the tools related to it such as Jira and Confluence. I learnt that sprint planning meetings are quite challenging, especially as the lead developer, I have to not only give time estimates for the development of the project, but also try to divide work evenly among four developers (including myself). What I struggled with the most throughout the project was distributing and assigning tasks to others. I found it difficult to align tasks to the interests of each developer while also making it autonomous enough where each task was not blocked by another. I also had to factor in the complexity/difficulty of each task and try to find the correct balance. If I didn't, it meant that the developer would contact me for support which would take away time for me to work on my own development tasks. Getting that balance right with each task was really difficult, it's something that I can definitely improve upon for the future.

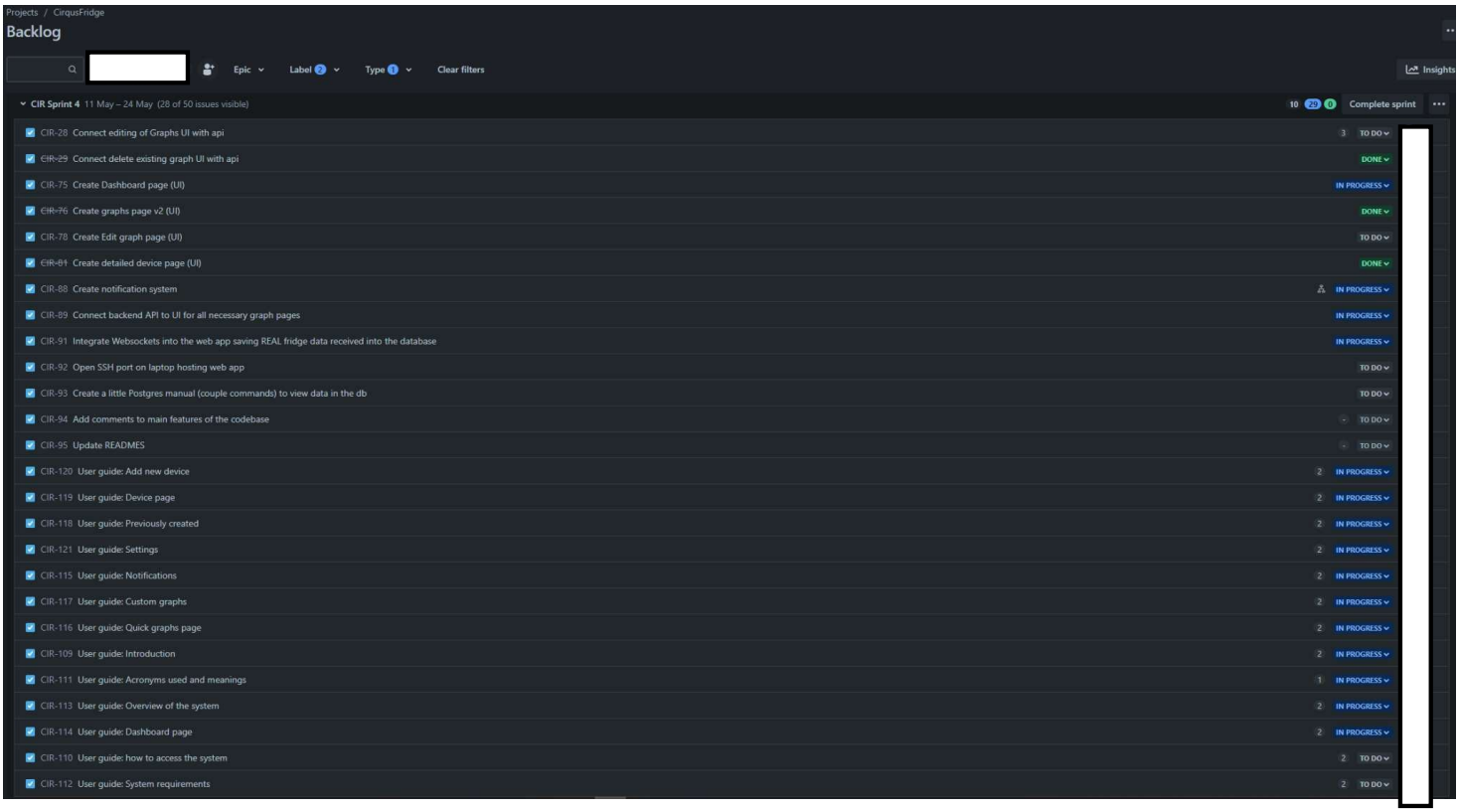
4 Appendix

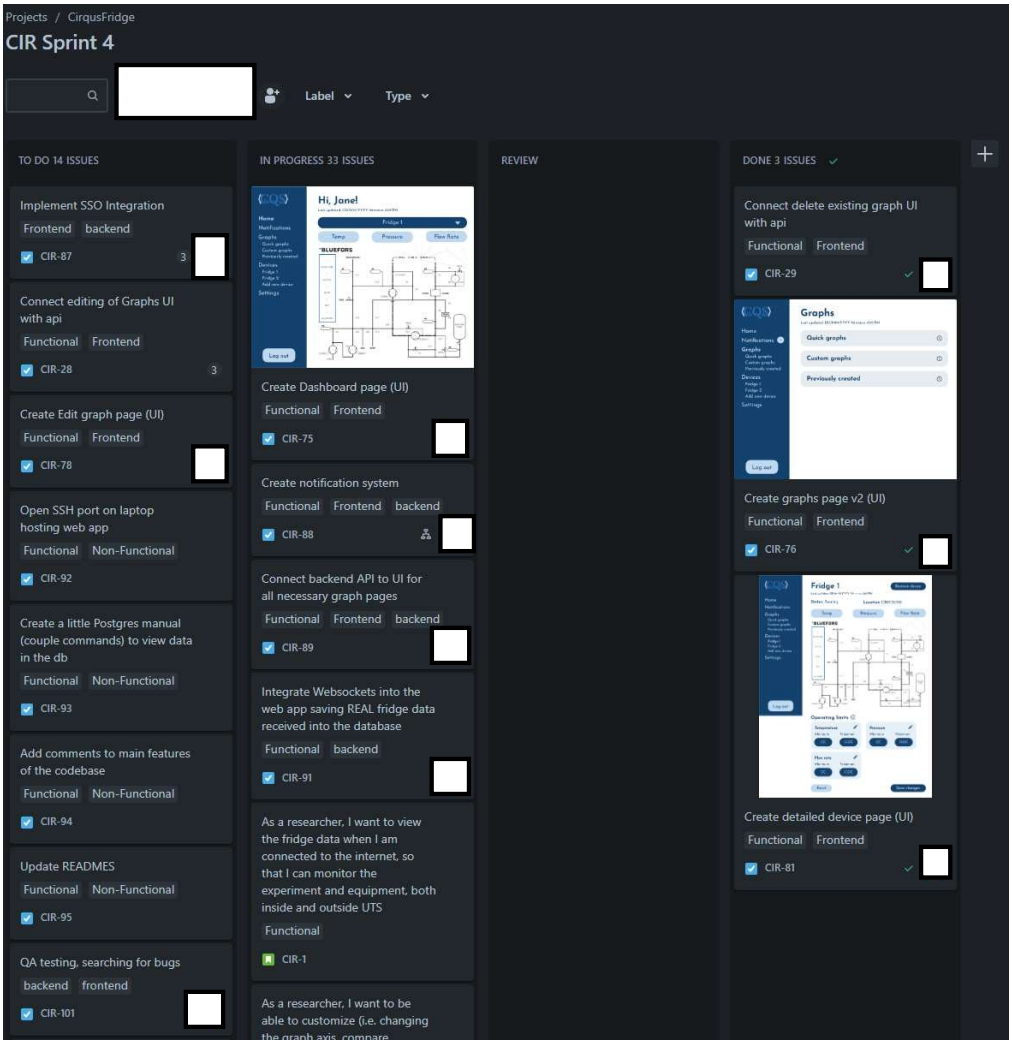
Appendix A: Initial product backlog created at sprint 1 (You may have to zoom in to see the tasks)

Exported backlog as CSV: Link here

Date	Event	Issue
Wed, Mar 22 2023, 2:00pm	Sprint started	
Wed, Mar 22 2023, 4:50pm	Added to sprint	CIR-13 Create Basic Web Application
Wed, Mar 22 2023, 4:50pm	Added to sprint	CIR-14 Create Basic Database
Thu, Mar 23 2023, 5:53pm	Added to sprint	CIR-15 Version 1 of UI/UX
Thu, Mar 23 2023, 5:56pm	Added to sprint	CIR-16 Write acceptance criteria for user stories
Thu, Mar 23 2023, 6:04pm	Estimate updated	CIR-15 Version 1 of UI/UX
Thu, Mar 23 2023, 6:10pm	Added to sprint	CIR-17 Development set up
Thu, Mar 23 2023, 6:12pm	Added to sprint	CIR-18 Define branching structure and GitLab practices
Thu, Mar 23 2023, 6:13pm	Added to sprint	CIR-19 Project scope sign off
Thu, Mar 23 2023, 6:14pm	Added to sprint	CIR-20 Follow up on SSO login for the Fridge Monitoring web application
Thu, Mar 23 2023, 6:15pm	Estimate updated	CIR-14 Create Basic Database
Thu, Mar 23 2023, 6:15pm	Estimate updated	CIR-13 Create Basic Web Application
Thu, Mar 23 2023, 6:18pm	Added to sprint	CIR-5 As a researcher, I want to know how the data (such as pressure, temperature, etc.) fluctuates over time, so that I can use this data to determine future trends
Thu, Mar 23 2023, 6:18pm	Added to sprint	CIR-4 As a researcher, I want to be able to view archived data as graphs so that the data is easy to read and monitor
Thu, Mar 23 2023, 6:18pm	Added to sprint	CIR-9 As a researcher, I want to be able to use a range of devices (laptops, mobile phones, tablets) to view the data, so that I can view the status of the equipment however is most convenient for me at the time
Sun, Mar 26 2023, 9:51am	Added to sprint	CIR-21 Read data from database
Sun, Mar 26 2023, 9:51am	Added to sprint	CIR-22 Create "Device/Fridge" class
Sun, Mar 26 2023, 9:51am	Added to sprint	CIR-23 Create "chart/graph" class
Sun, Mar 26 2023, 9:51am	Added to sprint	CIR-24 Store created classes to disk
Sun, Mar 26 2023, 9:52am	Added to sprint	CIR-25 Create a new table in the database on creation of a new "Device/Fridge" object
Sun, Mar 26 2023, 9:53am	Added to sprint	CIR-26 Generate "charts/graphs" using library and display them
Sun, Mar 26 2023, 9:53am	Added to sprint	CIR-27 Add dynamic scaling of "charts/graphs" based off a user's scroll wheel
Sun, Mar 26 2023, 9:53am	Added to sprint	CIR-28 Connect editing of Graphs UI with api
Sun, Mar 26 2023, 9:53am	Added to sprint	CIR-29 Connect delete existing graph UI with api
Sun, Mar 26 2023, 3:32pm	Issue completed	CIR-14 Create Basic Database
Sun, Mar 26 2023, 3:33pm	Issue completed	CIR-13 Create Basic Web Application
Sun, Mar 26 2023, 9:07pm	Added to sprint	CIR-30 Create application logic flowchart/user flow diagram
Sun, Mar 26 2023, 9:08pm	Added to sprint	CIR-31 System architecture diagram
Wed, Mar 29 2023, 2:54pm	Added to sprint	CIR-32 Implement React router and integrate with the Sidebar
Sat, Apr 01 2023, 12:16am	Issue completed	CIR-17 Development set up
Mon, Apr 03 2023, 1:37pm	Added to sprint	CIR-42 Research Recharts (JS Library) for "charts/graphs"
Mon, Apr 03 2023, 3:54pm	Removed from sprint	CIR-27 Add dynamic scaling of "charts/graphs" based off a user's scroll wheel
Mon, Apr 03 2023, 9:15pm	Issue completed	CIR-26 Generate "charts/graphs" using library and display them
Tue, Apr 04 2023, 3:53pm	Added to sprint	CIR-45 Use Case Diagram
Tue, Apr 04 2023, 3:54pm	Added to sprint	CIR-46 Data Dictionary Diagram
Wed, Apr 05 2023, 2:40pm	Issue completed	CIR-16 Write acceptance criteria for user stories
Wed, Apr 05 2023, 2:41pm	Issue completed	CIR-18 Define branching structure and GitLab practices

Appendix B: Current **s**print 4 backlog (You may have to zoom in to see the tasks)

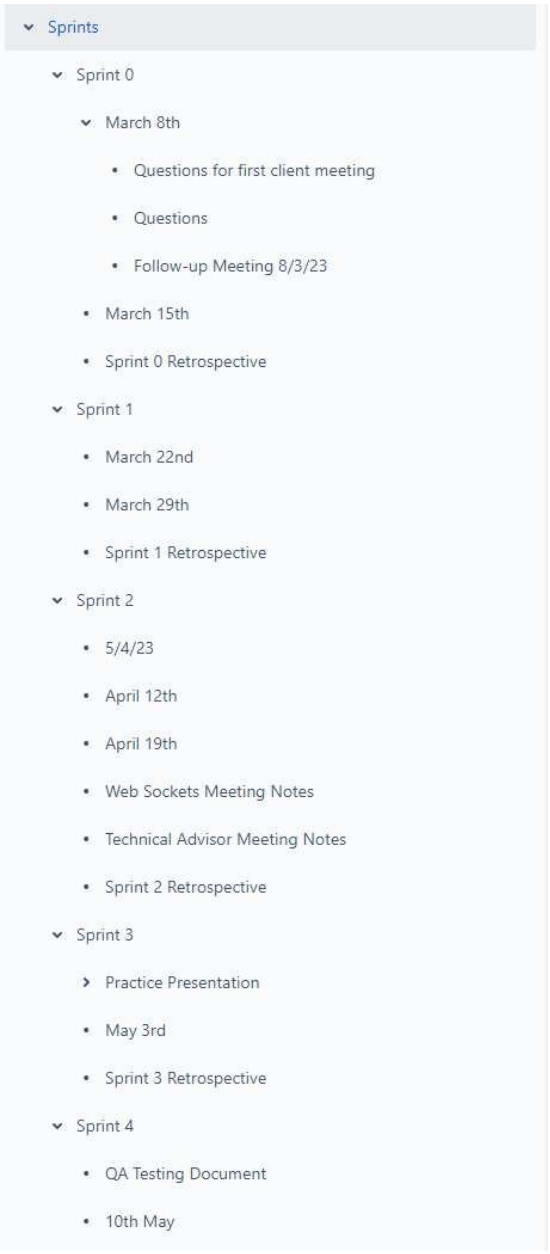




Appendix C: Meeting minutes example. Our team decided to have the BA person transcribe each meeting we had. A snippet of one meeting is shown on the left. Evidence of all our meeting minutes are shown on the right.

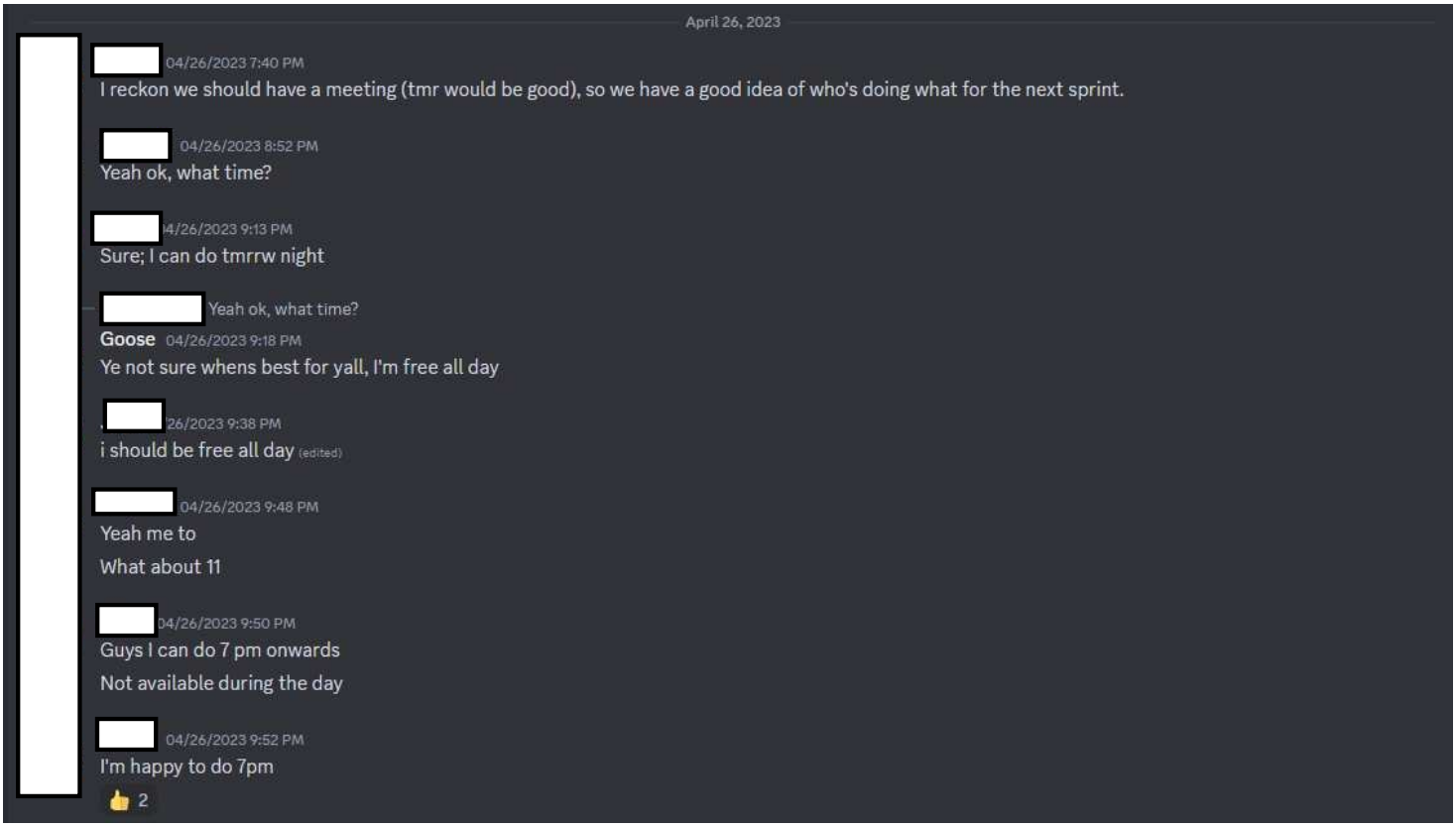
May 3rd

- There is a firewall that is blocking deployment of the solution. So we currently plan to write out documentation to future proof the solution in the event that we can't get IT to support us in deploying it. Does that sound good?
  - Adrien – Sounds good but can be designed it as if it were to connect?
  - Braeden – Yep that's the plan.
  - Adrien – Yep I assume that it isn't all that different in implementation.
  - Braeden – Yep so the implementation and product wouldn't change but we just wanted to make sure that its understood that we will be documenting how to connect to the fridge as if there was no firewall issue
- Jeyden – For the email notifications, if it exceeds a certain number then there is a payment associated with it.
  - Adrien – I didn't know that, what service is that?
  - Jeyden – EmailJS
  - Adrien – Do you know any work arounds Giorge?
  - Giorge – Nup
  - Adrien – What does EmailJS provide specifically?
  - Jeyden – It just sends the email securely.
  - Adrien – Are there any ways to send notifications without paying for a service? Like using MS Teams
  - Jeyden – It should be free yes
  - Adrien – What are you basing that on? Between now and next week, could you find a service that is free and understand how it works? We will also let Nathan know and see what he thinks
  - Liam – this is just something we wanted to raise with you.
  - Braeden – Just to clarify the service is technically free if you send less than 200 per month.
  - Adrien – We hope that the machines don't break but we would exceed it over 200 per month.
  - Giorge – Do we know how much it costs?
  - Braeden - $9 for 2000 per month. We will also share the link to it
  - Liam – We also will investigate teams notifications



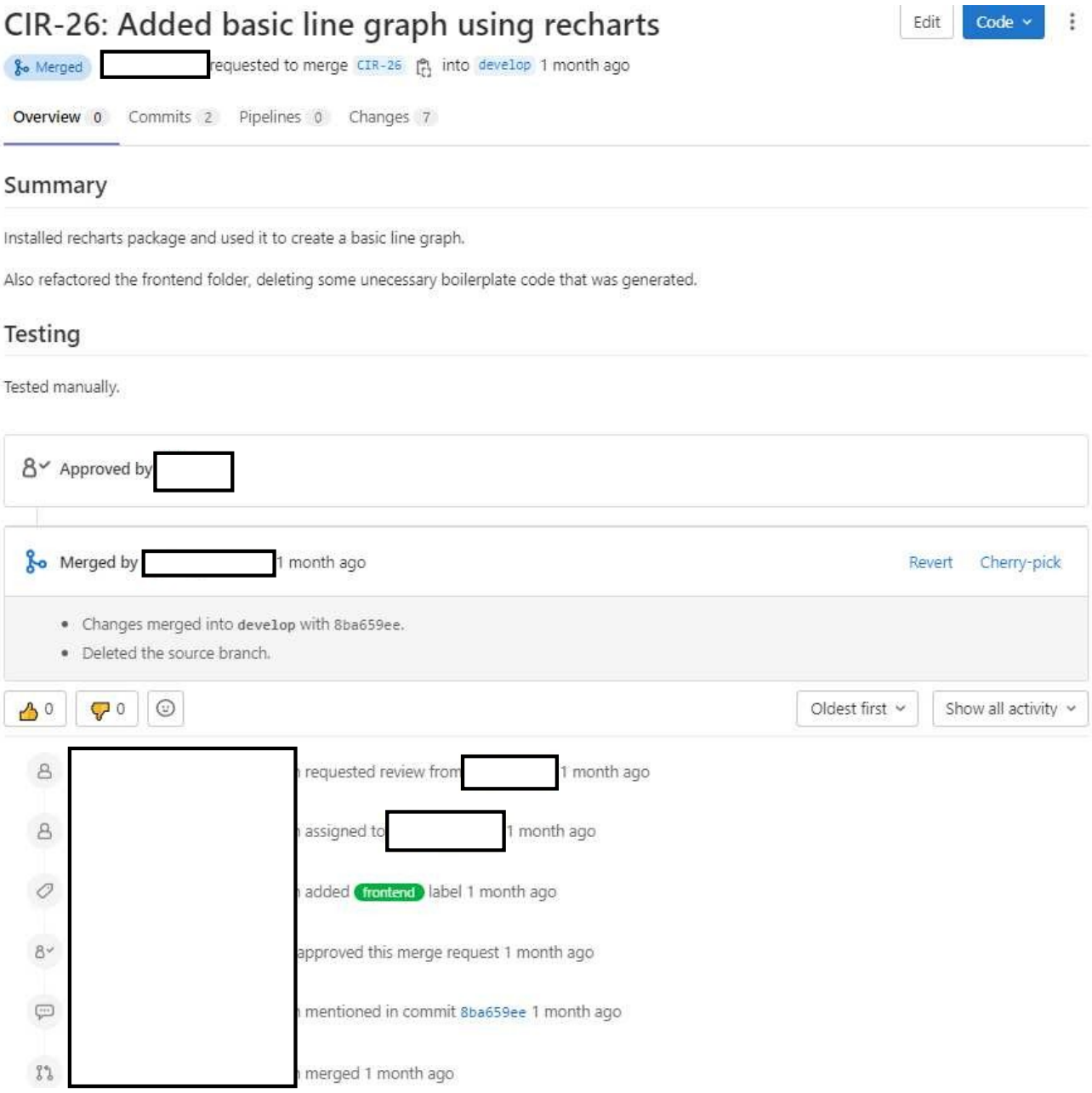
Appendix D: Sprint 1 Burndown Chart. This first chart is trending upwards highlighting our poor sprint planning. The second chart shows our improvement for sprint 2.

Appendix E: Sprint Planning Meeting for developers organised on Discord.

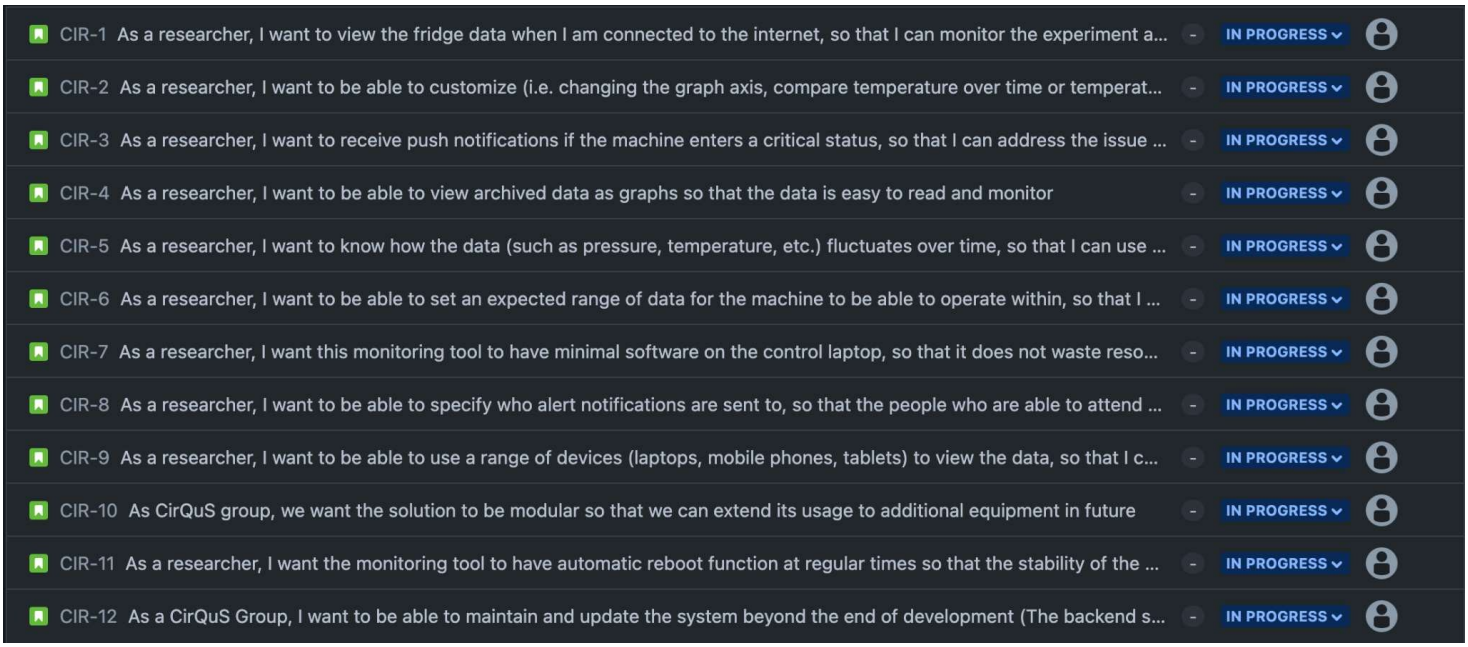


Appendix F: Example of Sprint 3's retrospective. We used a website called Mural.

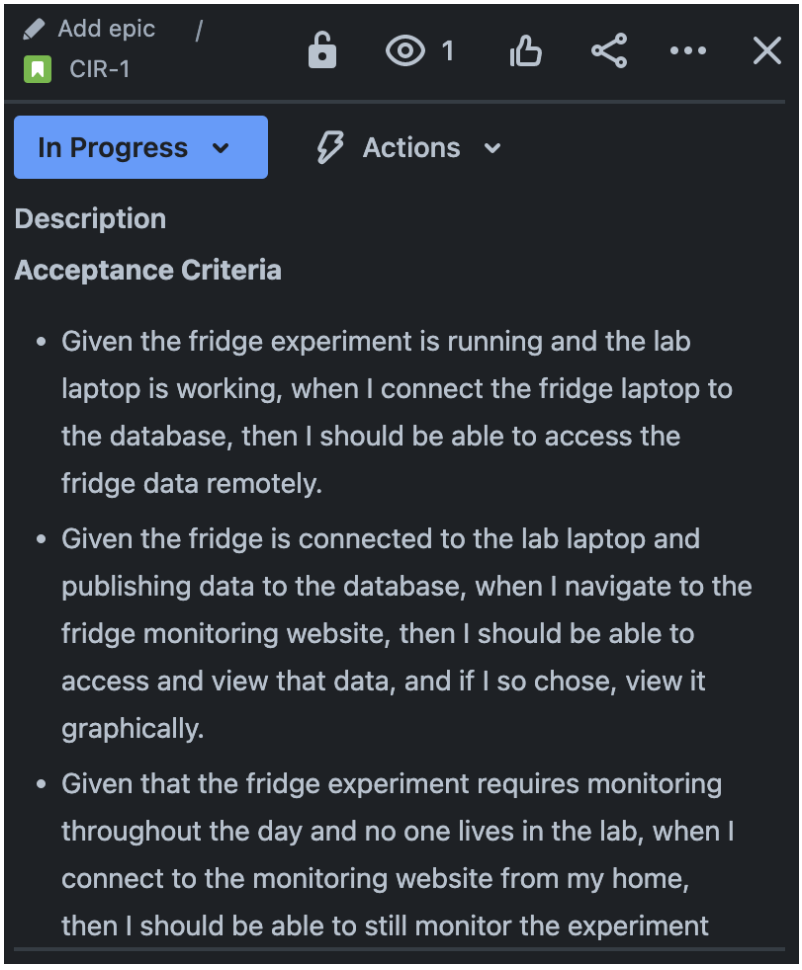
Appendix G: Gitlab merge request.



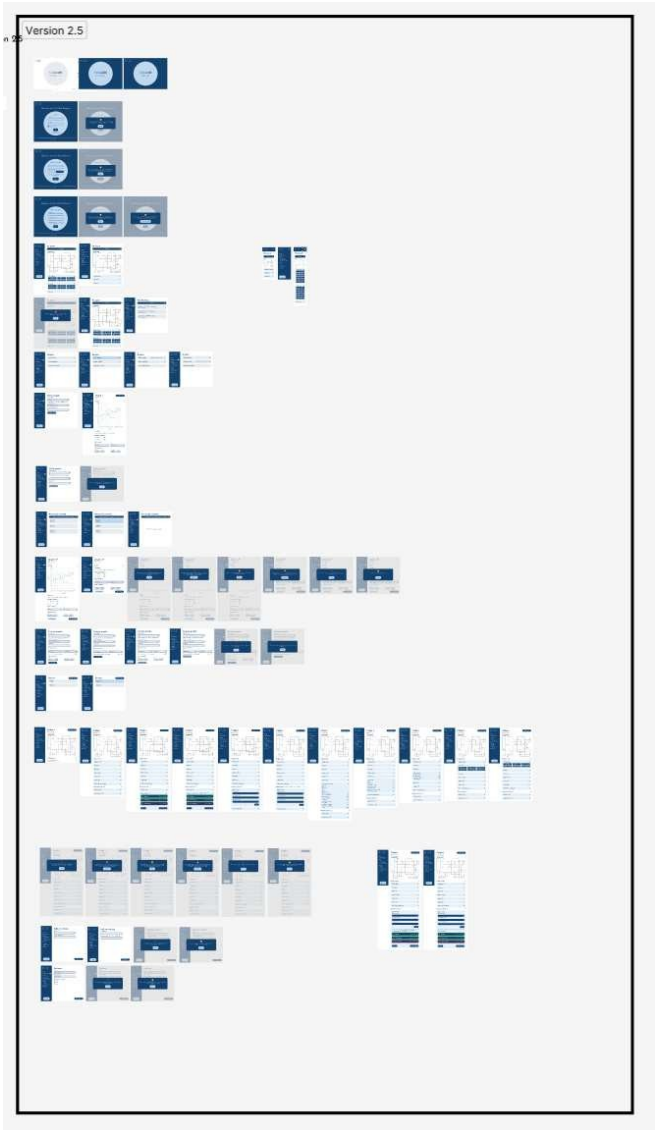
Appendix H: User stories.



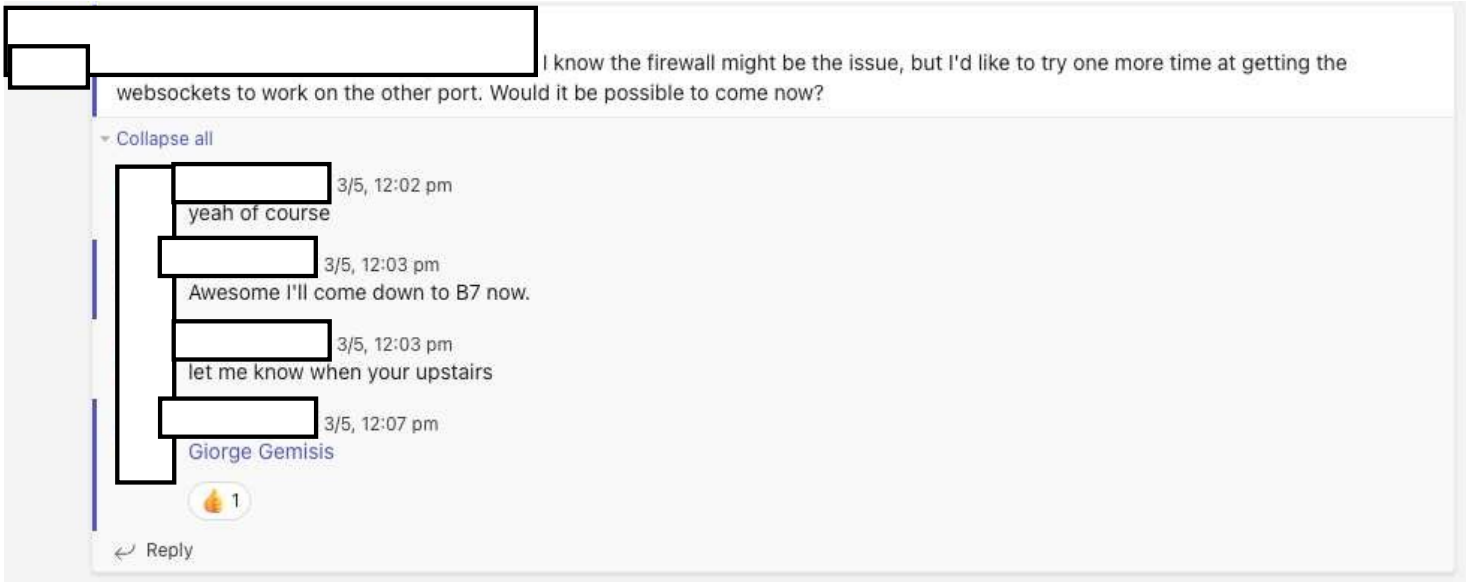
Associated Acceptance Criteria for the first user story (This was one of the main user stories assigned to me).



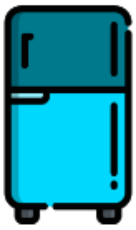
Appendix I: Web application design and UI mockups created using Figma.



Appendix J: Trying to resolve the websocket issue.



Appendix K: User manual. Detailed documentation of our application.



Argus User Guide

Structured walkthrough of the Dilution Fridge Monitoring Tool developed for CirQuS Group

Table of Contents

10. Dashboard

Upon successful login the Dashboard page is the first page a user is presented with. It provides "at a glance" information about the connected dilution fridges including the fridge front panel (indicating the current configuration), the temperature channels, the pressure channels and the flow rate.

The dashboard page also acts as the application home page and can be accessed from the side menu bar at any time by selecting the "home" option.

You can freely select the page you want to go to on the left side of the main page for page hopping. At the same time, you can click on the drop down to select a fridge to view the current configuration for a connected fridge.