MANAGEMENT INFORMATION

Deliverable 1

Team QQ
SENG3011

1 CONTENTS

2 PROJECT PLAN

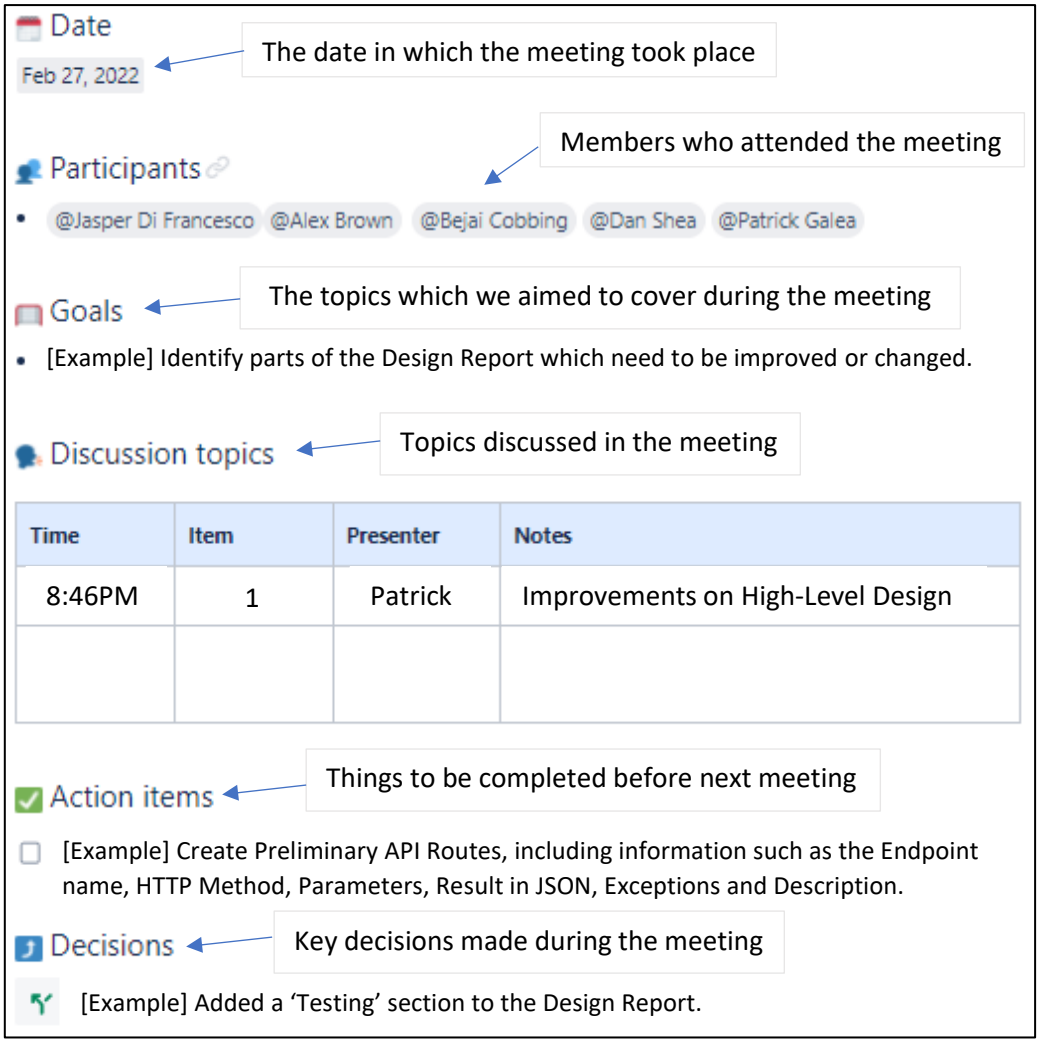
2.1 DEVELOPMENT AND COMMUNICATION STRATEGY

2.1.1 Team Meetings

Our communication strategy involves conducting at least two team meetings each week (usually one at the start of the week, and towards the end of the week) where all team members will say what they've completed since the previous team meeting, and what they aim to complete before the next meeting commences. The purpose of this is so team members are consistently up to date with the progress of the project. In addition, we will raise any challenges faced or problems encountered to ensure we can be flexible with work assigned and be able to deal with changing circumstances in a suitable manner. These team meetings will take place online using Microsoft Teams as it allows for the most flexibility between members, however we have also considered conducting in-person meetings towards the end of the term when we move onto the more practical side of development. In-person team meetings are convenient for all team members since we all live close to campus, and we believe they will be beneficial in allowing for more effective communication.

2.1.2 Meeting Minutes

During each team meeting, a designated team member will be noting down meeting minutes. The meeting minutes will be kept track of using Confluence (see section 1.3 for more detail). The Confluence meeting minutes template can be seen below:



The benefits of recording meeting minutes include:

- Members who missed a meeting can keep up to date with what was discussed.
- A record of who completed what can serve as evidence for how well work was shared between group members, and roughly how much each team member completed.
- The process of recording what work has been done, what work needs to be completed and the timeframe to do so can serve as a beneficial reminder and motivation boost to get that work completed.

2.1.3 Development Strategy

Our development strategy focuses on a prototyping approach, as we will start by developing a limited working model to gain an understanding of how our ideas will translate to implementation. From there, we will continue to add features and continuously test our product.

The strategy our team will follow for the implementation of new features is to use Jira. This will allow us to develop while using the Agile Methodology. Jira will allow us to use the 'Board' tool to create 'issues' (which are essentially jobs) which need to be completed for each epic. For more details, please see section 2.2.3.

2.2 Team Member Responsibilities and Project plan

2.2.1 Project Management Responsibilities

2.2.1.1 Product owner

Bejai Cobbing is taking on the role of the product owner within the Agile team. His responsibilities will include defining user stories and prioritising the team backlog so that product progress is maximised, and all developers are working efficiently. Within the stories responsibility, he will be ensuring that the stories are meeting the user's needs and encompassing all the desired functionality of the system.

Bejai is skilled in writing accurate user stories and making sure they always link back to the objectives of the project. Additionally, Bejai is a good leader and can make well thought out decisions regarding the direction of the project, so he is best suited to this role.

2.2.1.2 Scrum master

Alex Brown will be taking on the role of scrum master within the Agile team. His responsibilities will include facilitating the development of the team and ensuring they follow the agreed-upon processes. By way of facilitation, Alex will be working on removing obstacles that will be impeding the team's achievement of goals.

Alex has experience building applications at scale and also is on the architecture team so hence has the broadest knowledge of the system. He will be effective in removing obstacles and ensuring all team members are on track. Hence, due to his experience and knowledge of the system Alex is best suited to this role.

2.2.1.3 Retrospective

Jasper Di Francesco will be taking on the role of retrospective facilitator in the Agile team. His responsibilities are to facilitate the sprint retrospectives, which are designed to ensure that the team is building the habit of continuous improvement. By doing a retrospective at the end of a sprint, the team can discuss what to focus on more for the next sprint and can always be improving how we work.

Jasper has experience leading retrospectives in different teams and helping people to find methods of improvement. Given his experience, Jasper is most suited to this role.

2.2.1.4 Head of Testing

Dan Shea will be taking the head of testing role in the Agile team. The role of the head of testing is to ensure that all the testing done on the system is thorough and we have as high branch coverage as possible. Another responsibility is to enforce the process of writing tests as you write a function to ensure that as much of the function is covered as possible. By having someone assigned to lead the charge by way of testing, we can ensure the system is as bug-free as possible and will function as expected in as many cases as we can anticipate.

Dan has extensive experience writing rigorous tests for software and is skilled in thinking of edge cases for functions. Given Dan's experience testing and knowledge of different testing frameworks, Dan is best suited to this role.

2.2.1.5 UX Designer

Patrick Galea will be taking the role of UX designer in the team. UX designers are integral to the success of a website and the customer experience. They are specifically concerned with; the

creative design of a website, the ease of searching for information and the links between pages. This is particularly important for an API as maintaining an intuitive design of the website while having to mix and combine such a large amount of data and present it in a way that is logical is a significant challenge.

Patrick has lots of experience in customer service and design and can effectively apply these skills in a development environment to ensure our system caters to the user in the most effective way possible. Given Patrick's experience, he is best suited to this role.

2.2.2 Development Responsibilities

2.2.2.1 Team descriptions

Within the development team, we will be splitting up into multiple different teams. These are as follows; API server team, testing team, web scraping team, frontend team, and architecture team. The first stage of development consists mainly of building the API server and the web scraper. We anticipate the API web server to be easier to create than the web scraper, so once the API team has finished building the API and it has been sufficiently tested, all members will switch to the web scraping team to ensure we are always maximising our output.

2.2.2.2 API server team

The API server team is concerned with managing the routes for the API, which involves collecting the data from the database and organising it into a JSON structure. This team will also be managing connections to the database and the schema of the database, to ensure all the data is consistent, correct and only authorised parties can access the database. The API server team is only concerned with reading data from the database and not writing to the database, the main difficulty will be writing efficient SQL queries which can give the user back as much information as possible while keeping it relevant to the search query.

Later in the project, the team is planning to implement far more advanced searches than simply finding a disease with a location and time period. This will involve techniques such as fuzzy searching, geolocation queries that will allow a user to search an area, the ability to predict potential outbreaks, and the impact of an outbreak on residents in a particular area.

At this later stage in the project, this is where the API team will also implement more advanced routes which combine the APIs from other SENG3011 teams.

2.2.2.3 Testing Team

The testing team is perhaps the most important team, whose job it is to ensure the software is as bug-free as possible. This will include writing black-box tests for each function to make sure we are getting the correct output based on input. This also includes writing integration tests to make sure the system is working correctly as a whole. Each person in the team will write some tests for each function, however, the people in the testing team will be checking over those tests and writing more tests to ensure the software is thoroughly and rigorously tested. A big focus of the testing team will be making sure that the system handles invalid/incorrect input appropriately, with descriptive and informative error messages being sent back to the user after an invalid request.

2.2.2.4 Web Scraping Team

The web scraping team has the task of collecting articles from the CIDRAP website, and sifting through them to find keywords like locations, diseases, and symptoms. They will then be writing that data to the database, creating the article object and any report objects that exist within that article. Given that no one on the team has any experience with natural language processing

(NLP), this will be a very challenging task. The most difficult will be ensuring that each symptom or location found within the main body of the text gets matched up with the correct disease. This can be done through testing the proximity of each word to the disease, however, the difference between how articles are written poses a significant challenge.

2.2.2.4.1 Frontend Team

The frontend team is tasked with building the GUI for the API and integrating it with our API server. As stated in the design details, the frontend server will be written in JavaScript using the React framework and the Next.js framework. A background in JavaScript is required for the fast and efficient development of the frontend. The frontend team is also tasked with ensuring a seamless user experience. User experience and ensuring the user interface is as user-friendly and intuitive as possible is such an important part of producing a platform, so the entire development team will be discussing this at a later stage in the project to ensure we have the best product possible.

2.2.2.5 Team Structure

Each person in the development group will be in multiple teams to ensure our skills are being spread out, and each team has been assigned based on the skills of each member and their eagerness to learn more about a particular topic.

Team	**Members**
API Server team	• Jasper Di Francesco • Patrick Galea • Dan Shea
Web scraping team	• Alex Brown • Bejai Cobbing
Testing team	• Bejai Cobbing • Patrick Galea • Dan Shea
Frontend team	• Alex Brown • Jasper Di Francesco
Architecture team	• Alex Brown

3 PROJECT MANAGEMENT TOOLS

3.1 OVERVIEW OF MANAGEMENT TOOLS

The project management tools that our team will use throughout the project are Confluence and Jira, which are both Atlassian products.

3.1.1 Confluence

Confluence is a wiki-styled team workspace which allows people and organisations to create spaces and pages. Confluence spaces help structure teams so that team members have access to the correct information. Team members within a certain space have access to the spaces' pages. When creating a new Confluence page, users have access to more than 70 templates which helps to make life easier.

Our team has chosen to use Confluence since it is a simple and effective way to organise and plan our work. It allows us to create our own pages which can act as 'folders' for other pages, and

we can add whatever information we like to our pages. We have found the page templates to be extremely useful, for example we are utilising the meeting minutes template to create separate pages for each of our meetings, which saves us time when taking our meeting minutes (see section 1.1.2).

We have decided to use Confluence as the central location for our documents throughout the project. We were originally planning on using Microsoft Teams, but once we started using Confluence, we realised that it will be much more useful for us. Confluence is far more flexible than teams, making it much easier to organise our documents. Confluence also gives us access to templates which Teams does not give. Finally, although Confluence has a steeper learning curve, it is a tool that is widely used within industry, so it is a worthwhile investment of our time to learn it now since we will likely need to use it once we graduate.

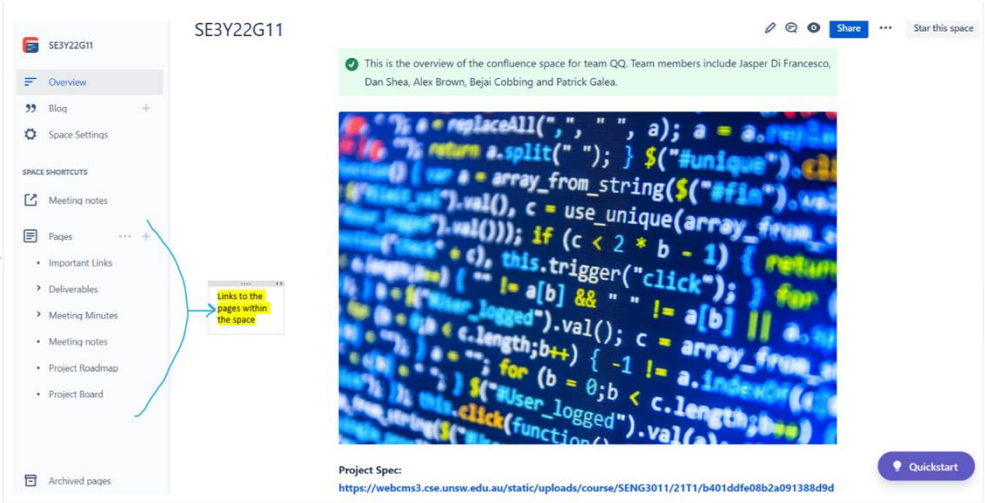
3.1.2 Jira

Jira is an issue-tracking software specially made for both bug tracking and managing project tasks. Our team will be making use of the Roadmap and Board sections, where we can create epics and issues to keep track of future deadlines and work that needs to be done. Epics are major milestones or bodies of work, which can then be broken down into issues which are smaller tasks that need to be completed. For example, an epic might be 'Develop Web Scraper' and an issue might be 'Write Unit Tests for Scraper'. The Roadmap section shows a timeline of all epics that need to be completed within the project and their due dates, and the Board section can show us the status of all the issues that need to be completed within a certain epic.

Our team has chosen to utilise Jira since it is a very effective way to organise future tasks, it gives us visual tools to show us what needs to be done and when. Jira will help us to plan and track how far into each aspect of the project we are. We plan on utilising the Agile Methodology while developing our web application, and Jira is the perfect tool to help us to do so, since the epics and issues setup easily translates to sprints in the Agile Methodology.

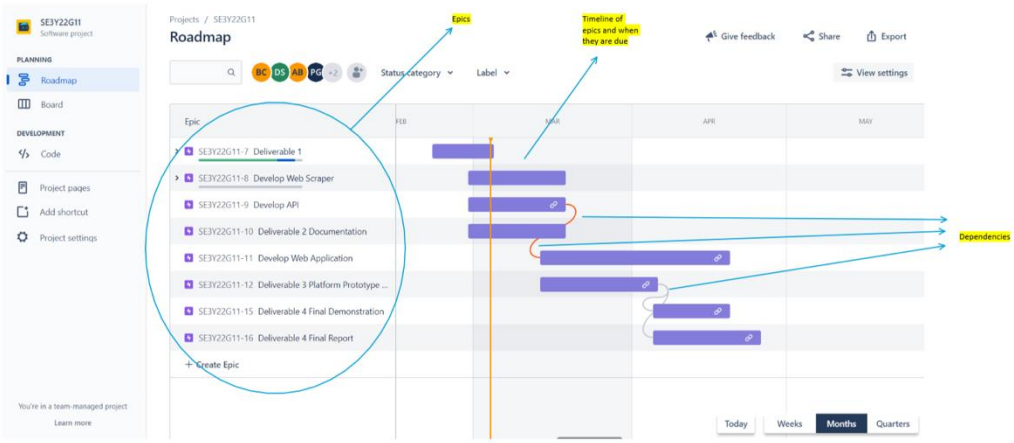
3.2 Annotated Diagrams and walkthrough of Management Tools

3.2.1 Confluence Space Overview



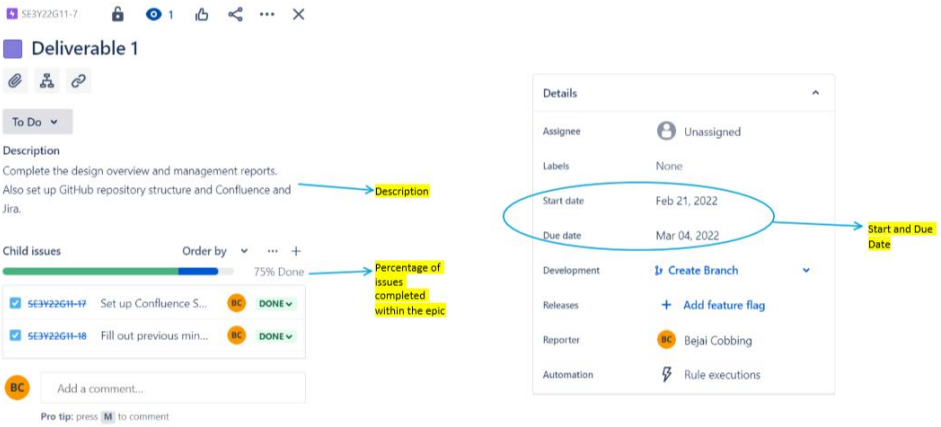
Above is the overview section of the "QQ" space on Confluence. The QQ space has each of our team members in it, so we all have access to each of the pages in the space. These pages can be seen on the toolbar to the left.

3.2.2 Project Roadmap



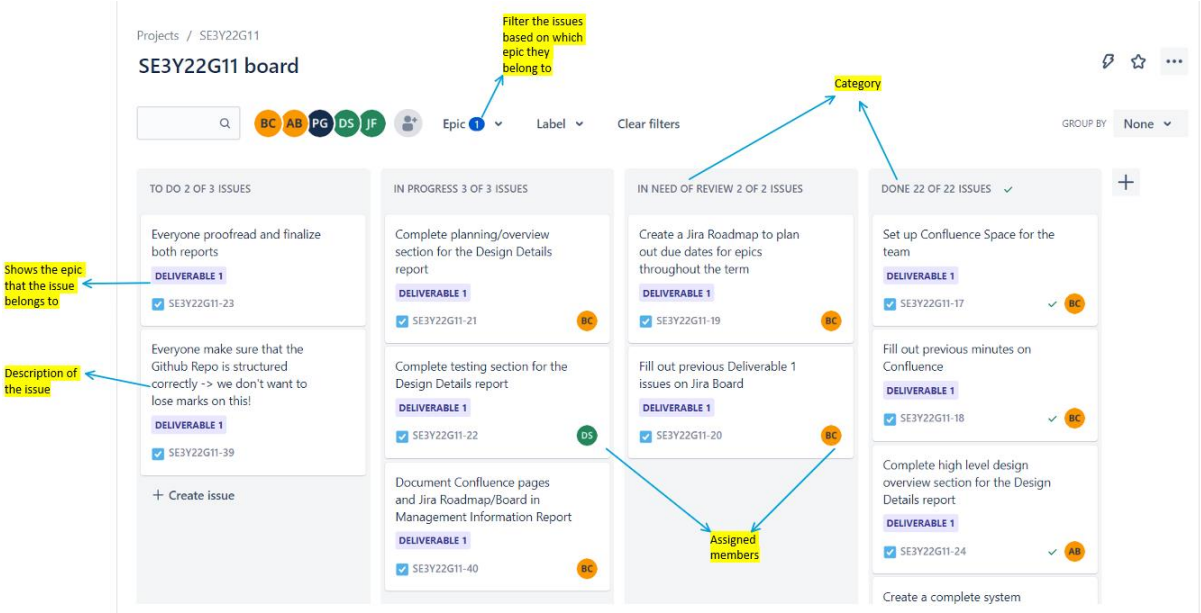
Above is the Project Roadmap section on Jira. The Roadmap is a timeline of all the epics within the project. Epics are the major milestones or bodies of work. This page can be reached by clicking on the “Project Roadmap” page on the Confluence space.

On the left is a list of all the epics within our project. Then to the right there is a timeline of each of the epics and when they are due. If the individual epics are clicked on, Jira will display more information about the epic such as a description, the percentage of issues completed within the epic, and the start and due date of the issue:



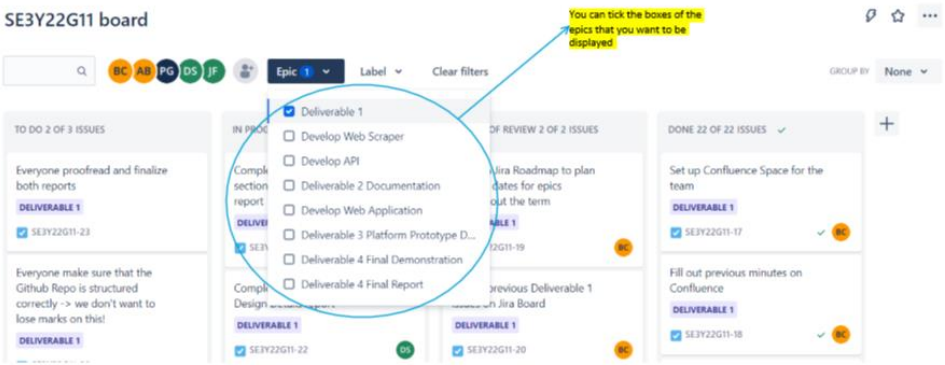
The lines in between some of the epics (in the first picture) are called dependencies. A dependency is when one epic depends upon another in order to be completed. Epics that are depended on should be prioritised to be completed first so as to avoid ‘roadblocks’ which is when a task cannot be completed since it is waiting on the completion of another task. For example, “Develop Web Application” is dependent upon “Develop API” since the web application cannot be completed without the API being completed first.

3.2.3 Project Board



This is Project Board section on Jira. It can be reached by clicking on the" Project Board" page from the Confluence space.

The Project Board section shows each of the individual issues. Issues in Jira are the smaller building blocks that make up epics, they are the individual tasks. Each issue belongs to an individual epic. By clicking on the "Epic" button up the top, you can filter the issues based which epics you want to see:



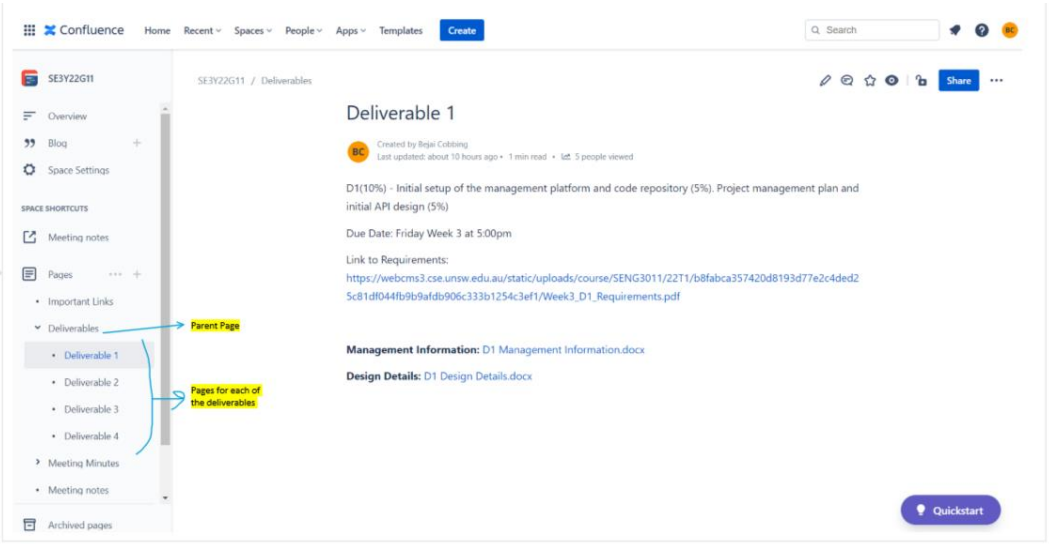
Looking back at the first picture, there are four categories which an issue can be categorised under:

- To do – Issues which need to be done.
- In progress – Issues which have currently being completed.
- In need of review – Issues which have been resolved (or features which have been implemented) and need to be reviewed before being pushed into the deployed version of the product. If the issue not approved, it would get moved back into the 'in progress' category where the team member who was assigned to that issue will need to make suitable changes.
- Done – Issues which have been resolved and reviewed.

Using a task board and creating issues which can be categorised is an effective tool for organising the delegation of tasks and understanding how far into development each aspect of the

project is. For each feature our team plans to implement, we will create a corresponding issue and move it between categories when appropriate. We will use the assignment feature to ensure each team member knows which issues they are working on, which issues other team members are working on. We will also use labels so that issues of a specific nature (such as bugs) can be easily identifiable in the task board. When a team member believes they have implemented a feature, they will ask for it to be reviewed by another team member. If approved, they can then move that issue into the 'done' category and continue working on other issues!

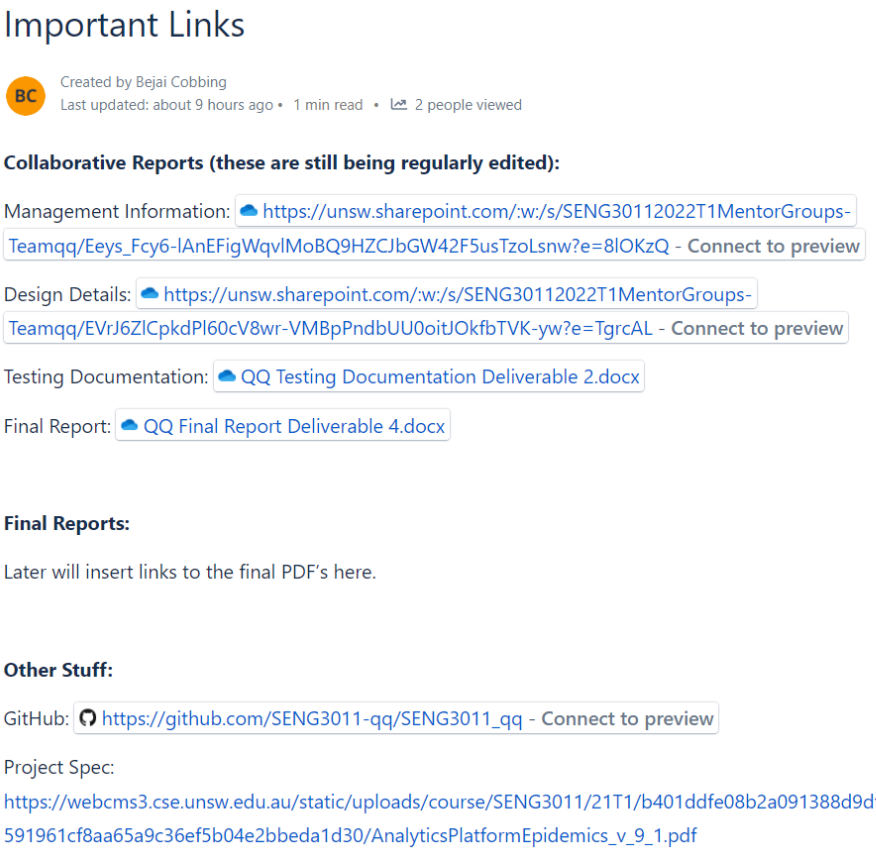
3.2.4 Deliverables Pages



The above picture shows the Confluence page for Deliverable 1. Our team has created separate pages for information on each of the deliverables. There is some admin information, a link to the spec for this deliverable 1, and links to the collaborative documents for this deliverable. The pages for the other deliverables are very similar.

To the left, it can be seen that there is a "Deliverables" parent page, and then each of the Deliverables are child pages of this page.

3.2.5 Important Links Page



Important Links

Created by Bejai Cobbing
Last updated: about 9 hours ago • 1 min read • 2 people viewed

Collaborative Reports (these are still being regularly edited):

Management Information: https://unsw.sharepoint.com/:w:/s/SENG30112022T1MentorGroups-Teamqq/Eeys_Fcy6-lAnEFigWqvlMoBQ9HZCJbGW42F5usTzoLsnw?e=8lOKzQ - Connect to preview

Design Details: https://unsw.sharepoint.com/:w:/s/SENG30112022T1MentorGroups-Teamqq/EVrJ6ZlCpkdPl60cV8wr-VMBpPndbUU0oitJOkfbTVK-yw?e=TgrcAL - Connect to preview

Testing Documentation: QQ Testing Documentation Deliverable 2.docx

Final Report: QQ Final Report Deliverable 4.docx

Final Reports:

Later will insert links to the final PDF's here.

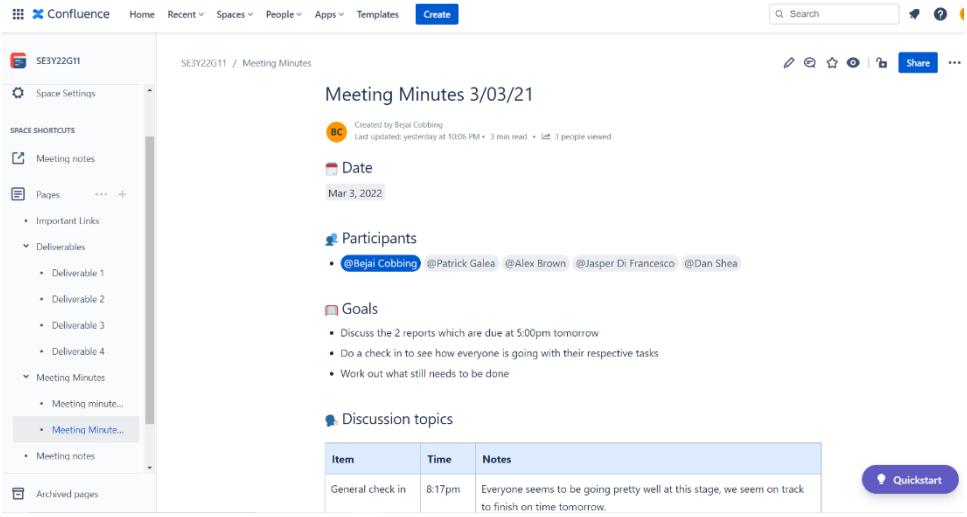
Other Stuff:

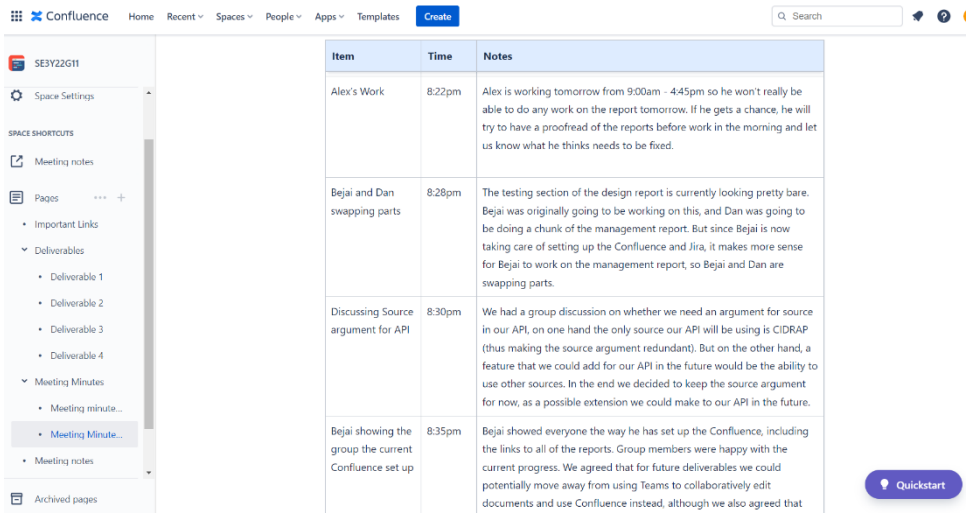
GitHub: https://github.com/SENG3011-qq/SENG3011_qq - Connect to preview

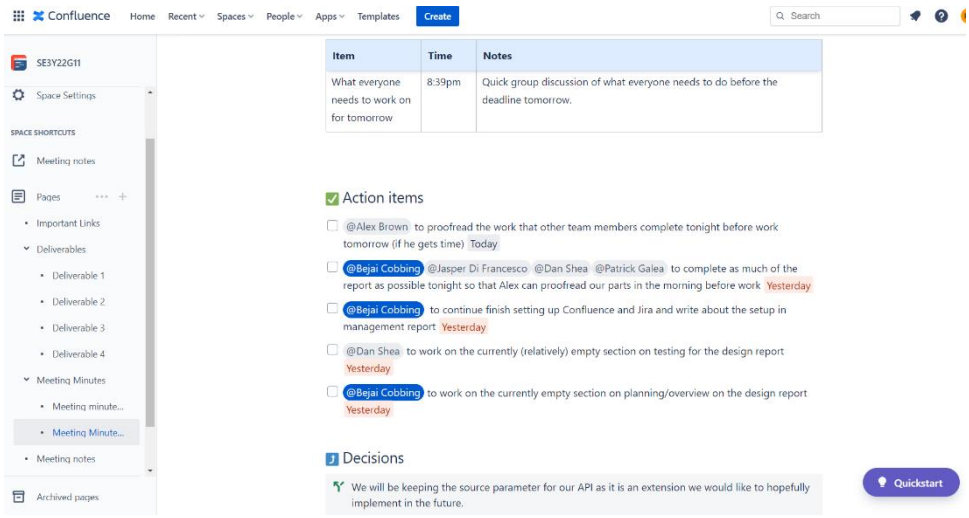
Project Spec:
https://webcms3.cse.unsw.edu.au/static/uploads/course/SENG3011/21T1/b401ddfe08b2a091388d9df9fe7bd591961cf8aa65a9c36ef5b04e2bbeda1d30/AnalyticsPlatformEpidemics_v_9_1.pdf

The above picture shows our Important Links Confluence page. There are links to all the collaborative documents that we will be using throughout the project, a placeholder space for the final PDF's which were inserted after this screenshot was taken, and some links to other things such as the team's GitHub repository.

3.2.6 Meeting Minutes Pages

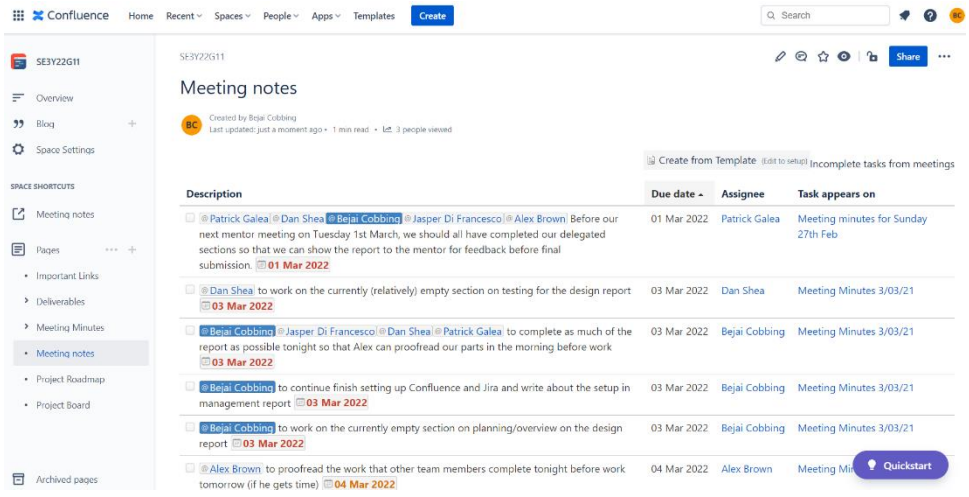


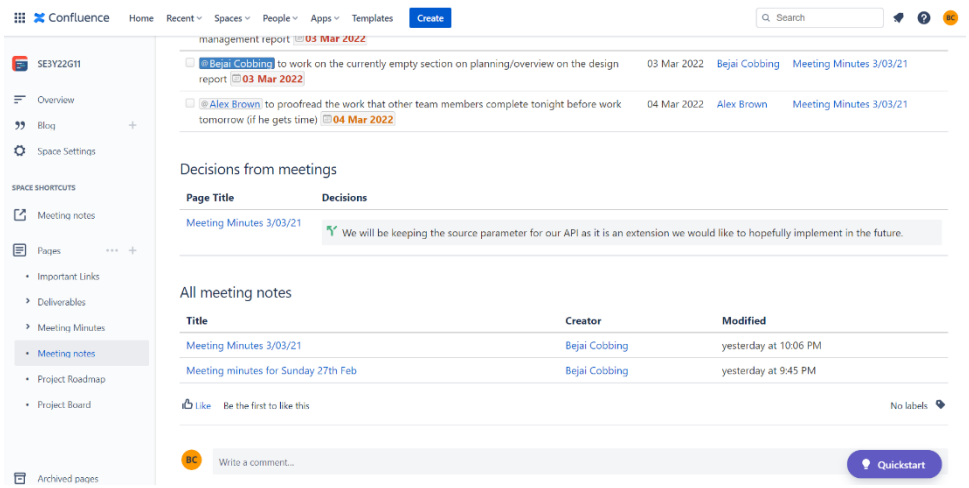




The above pictures show an example of one of our Meeting Minutes Confluence pages. It was created using the meeting minutes Confluence template, which made our lives a lot easier. (Also see section 1.1.2 for more about meeting minutes).

3.2.7 Meeting Notes Page





The above pictures show our team's Meeting Notes Confluence page.