

MANAGEMENT INFORMATION

SENG3011 REPORT

TELEUBBIES

Yin Huey Tan z5211373
Amanda (Xiaorui) Li z5206613
Sarah Oakman z5206178
Lavanya Sood z5208121
Yiyun Yang z5187469

1.0 TEAM RESPONSIBILITIES

1.1 DELIVERABLE 1

Our team will be working together to develop the API with each person taking lead in different parts of the development. The responsibilities below include current and future deliverables, while making sure that everyone gets involved in every development stage.

Yin Huey Tan : Backend API Development Lead (Design Report 1.3,1. 4)

Xiaorui Li : Web Scraper Lead (Design Report 1.1)

Sarah Oakman : Database Management and Web Service Lead (Design report 1.2,1.5, 2.1)

Lavanya Sood: API Testing and Frontend Lead (Design report 3.0)

Yiyun Yang : API Documentation Lead (Design Report 2.2)

1.2 DELIVERABLE 2

For deliverable 2 our team was working together on different parts to build a successful api

Yin Huey Tan : Backend API Development, GET and POST Request, Postman tests, Pytests

Xiaorui Li : Database Management, Web Scraping

Sarah Oakman : Database Management , Web Scraping, herokuapp development, logfiles

Lavanya Sood: API Documentation, DELETE Request, Pytests

Yiyun Yang : API Documentation, PUT Request, Timezone, Pytests

2.0 TEAM COORDINATION

2.1 PROJECT MANAGEMENT

For this project, we will be using JIRA software for planning, managing and tracking our tasks in order to collaborate and handle our workflow efficiently. To keep track of our progress and ensure timely product delivery, we will be adapting the scrum methodology to manage our project in an agile manner. User stories will be created and sprints will be created in JIRA with a timeframe of about a week. We will plan our tasks before each sprint and delegate our tasks according to our respective responsibilities.

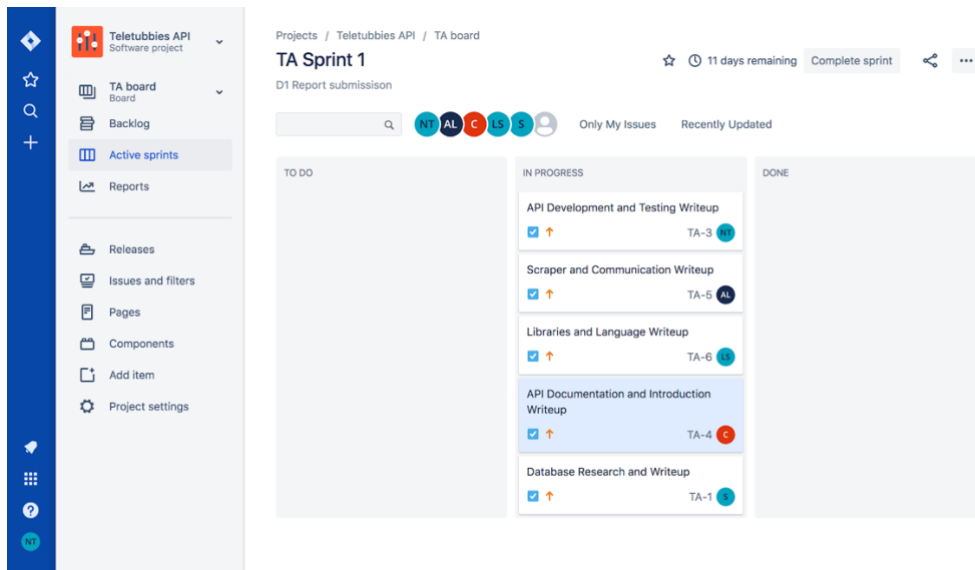


Fig 1.0 – The JIRA Sprint for Deliverable 1

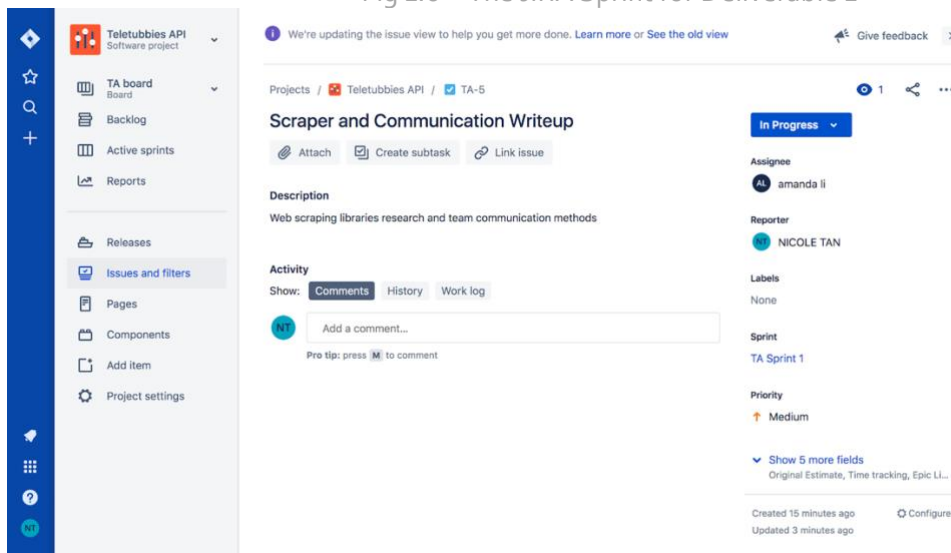


Fig 2.0 – Example of a task on the sprint

2.2 PROJECT TIMELINE

In order to visualise and plan our project holistically, we created a Gantt Chart timeline to mark important milestone deadlines, iterations and our own hard and soft deadlines (outlined in the figure below by gradually darker colours). This helps us all be on the same page, as well as notifies us of when we should begin tasks in order to avoid last-minute rushes. As a result, Gantt Charts aid in time management, tracking project status and dependencies (relying on other sections to be completed), and resource allocation. Below is a general overview Gantt Chart and smaller sub-tasks will be added as the project progresses. For example, if getting data depends on web scraping, we can make sure web scraping is completed as early as possible and allocate people accordingly, as well as make sure a person is tasked with too much responsibility at one time.

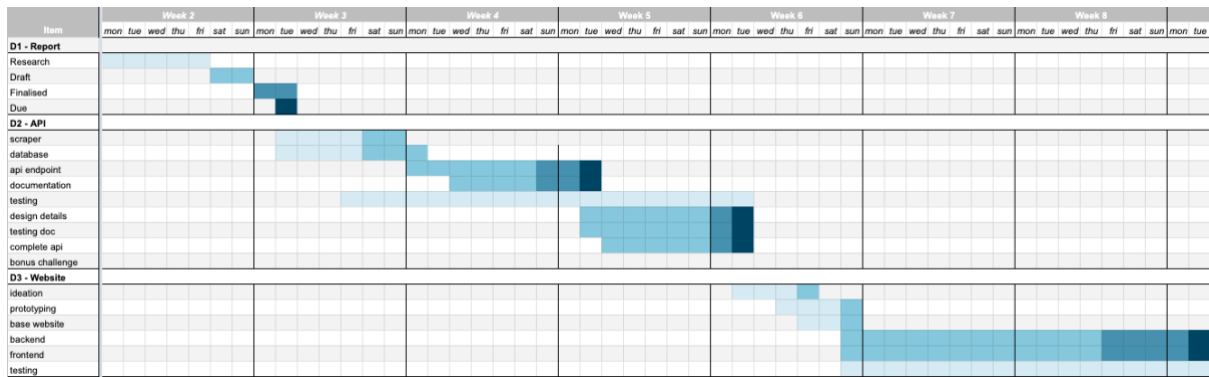


Fig 3.0 – Gantt Chart

2.3 TEAM COMMUNICATION

Another important aspect of team projects is clear and consistent communication, in order to address issues quickly. We will be communicating mainly through weekly meetings and Facebook Messenger chat.

We already had a group chat set up in Messenger, and it is a social media website which we all use and check frequently, allowing efficient communication of issues. Additionally, if members are unable to make meetings, we can video and call them to keep them updated. If members need to be notified immediately of changes, they can be tagged/mentioned directly as well.

2.4 TEAM COLLABORATION

In order to complete tasks efficiently, we need to be able to work simultaneously.

For document sharing, Google Drive allows our files to be automatically backed up, ability to restore previous versions through editing history, work on our own sections concurrently and view changes live, as well as allow others to comment concerns and suggest improvements. It's also a tool we all use regularly.

For code sharing, we'll be using Github to work simultaneously on different branches and merging later. Github helps us with version control (ability to restore old versions as well as commit statements that outline changes) and documentation (readme files and contribution analysis). We've also used Github for multiple projects before, meaning less time will be spent on learning how to use a new management tool.

A past problem we've encountered is branching and merging conflicts with Git such as code being deleted or replaced. To overcome these issues, we'll be using SourceTree which is a free Git desktop Graphical User Interface for Windows and Mac. SourceTree displays a visual representation of the repository, enabling branch switching easily and executing Git commands in one click. It also provides support for stashing local changes, tells users how far ahead or behind they are and alerts them to push or pull.