

IT Project Group 10 Handover Documentation

Customisable AI Use Scales Across Assessment Formats

Group Members:

- Maddy Wagner (Scrum Master) madelinewagner@me.com
- Winston Zhang (Product Owner) winston.zhang8@gmail.com
- Violet Zheng (Back-End Lead) violet.ywz@gmail.com
- Jason Jiang (Front-End Lead) jsunjiang@gmail.com
- Bonnie Hou (Testing Lead)

Table of Contents

<u>Project Requirements</u>	2
<u>User Stories</u>	2
<u>Motivational Model</u>	3
<u>Who/Do/Be/Feel Model</u>	4
<u>Design and Architecture</u>	5
<u>Frontend Design Model</u>	5
<u>Architecture Goals and Constraints</u>	9
<u>Domain Model</u>	11
<u>System Architecture Diagram</u>	11
<u>Use Case Diagram</u>	12
<u>System Sequence Diagram</u>	13
<u>User Manual</u>	16
<u>Staff User Guide</u>	16
<u>System Admins User Guide</u>	21
<u>Key Classes and Application Layers</u>	23
<u>Licenses</u>	24
<u>Third-party Dependencies and Licenses</u>	24
<u>Backend (Python/Django)</u>	24
<u>Frontend (Node/React)</u>	25
<u>Database Access</u>	26
<u>Database Structure</u>	27
<u>Data Documentation</u>	34
<u>Deployment Guidelines</u>	35
<u>Project Documentation</u>	45
<u>Sprint Releases</u>	45
<u>Sprint 1 Release (01/09/2025)</u>	45
<u>Sprint 2 Release (08/09/2025)</u>	45
<u>Sprint 3 Release (22/09/2025)</u>	46
<u>Sprint 4 Release (06/10/2025)</u>	46
<u>Sprint 5 Release (13/10/2025)</u>	46
<u>Sprint 6 Release (20/10/2025)</u>	47
<u>Handover Release (05/11/2025)</u>	47
<u>Meeting and Stand Up Notes</u>	48

Project Requirements

This first section outlines the requirements of the Customisable AI Use Scale Platform, as determined during our first client meeting on the 11th of August 2025.

User Stories

The following user stories have been implemented in the current iteration of the codebase:

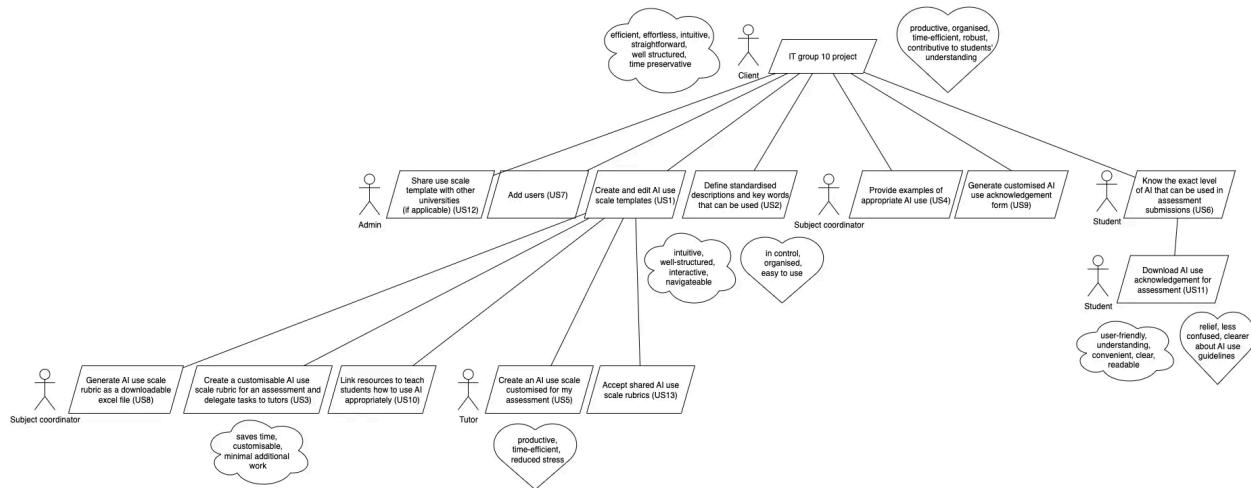
- US1 (must): As a **system admin**, I want to be able to create and edit AI using scale templates so that other users have access to readily available and accurate templates.
- US2 (must): As a **system admin**, I want to be able to define standardised descriptions and key words that subject coordinators can use so that the system is easy to understand by all potential users from different faculties.
- US3 (must): As a **subject coordinator**, I want to be able to create a customisable AI use scale rubric for my assessment and also delegate such tasks to tutors so that I set a clear standard of AI use for my students and be able to judge the amount of acceptable amount of AI use when marking students' submissions.
- US4 (must): As a **subject coordinator**, I want to provide examples of appropriate AI use (e.g. prompt examples) so that students have a clearer understanding of the appropriate use cases of AI.
- US5 (must): As a **tutor**, I want to be able to easily create an AI use scale rubric customised for my assessment so that I have a clearer idea of the acceptable amount of AI use when I'm marking students' submissions.
- US6 (must): As a **student**, I want to know the exact level of AI I am able to use in my assessment submissions in the form of an easily readable rubric so that I can ensure I am complying with academic integrity guidelines.
- US7 (should): As a **system admin**, I want to be able to add users so that they can create their own AI use scales that are applicable to their own subjects.
- US8 (should): As a **subject coordinator**, I want to be able to generate an AI use scale rubric as a downloadable excel file so that students can view it.
- US9 (should): As a **subject coordinator**, I want to be able to generate a customised AI use acknowledgement form as a file so that students can download and edit it.
- US10 (should): As a **subject coordinator**, I want to be able to link resources to teach students how to use AI appropriately for each defined level within the scale so that students have a clearer understanding of the appropriate level of AI.
- US11 (should): As a **student**, I want to be able to download an AI use acknowledgement for my assessment so that I can fill in the form with areas for which I've used AI.

The following user stories have not been implemented:

- US12 (could): As a **system admin**, I want to be able to share the AI use scale template with other universities (if applicable).
 - This has not been implemented at client's request. The current system will be stored on the University of Melbourne servers, therefore, it does not make sense for it in its current state to be available to other universities
- US13 (could): As a **tutor**, I want to be able to accept AI use scale rubrics that have been shared with me so that I can help contribute to subjects that I teach.
 - While this is a feature that could potentially be useful for collaboration in the future, in the current iteration of our software, we prioritised allowing all staff members (both subject coordinators and tutors) being able to create templates so this is a feature that has not been implemented as of yet

Motivational Model

This diagram helps show how all the user stories are connected together, while also considering aspects from the who/do/be/feel model found on page 4, most importantly, how users should feel when interacting with the system and how the system should behave in response to the user.



Who/Do/Be/Feel Model

Subject coordinators	<p>Create assessment specific templates for their subjects</p> <p>Customise templates from community templates repository for their assessment</p> <p>Provide examples for the AI use cases.</p> <p>Inform students of AI usage guidelines and requirements in table format. Easily create and customise AI use scale tables.</p> <p>Keep track of prior versions of their AI use scale.</p> <p>Share their created AI use scale with other staff.</p> <p>Edit a shared template with other staff.</p> <p>Invite other tutors for the subject to the system.</p>	Efficient User friendly Interactive Easy to use Flexible Customisable Intuitive Quick to learn Convenient Straightforward High quality Structured Practical	Trusting Organised Productive Time-efficient Sufficiently communicated Robust Contribute to the understanding and integrity of their students Secure
Tutors	Same as subject coordinator minus the inviting other tutors to the system.	Same as subject coordinator	Same as subject coordinator
Admin	Manage user roles & credentials. Monitor user statistics and AI use scale statistics. Create AI use scale base templates to share with other users. Invite users to the system. Edit/remove all templates in the system.	Organised Easy to understand Intuitive Easy to make changes Practical Effortless Secure	In control Flexible Assured At ease Comfortable Feel assured and clear on what to do when other users have issues regarding their templates/credentials.
Students	Easily understand the level of appropriate AI use in their assessment in table format. Acknowledge their usage of AI within their assessment split by tasks in a form.	Ease of use Well-structured Easy to understand Intuitive	Assured Clear Comfortable with level of AI use for assessment Less confused

Design and Architecture

This next section outlines our initial design models for the Customisable AI Use Scales platform

Frontend Design Model

This design model serves to outline the front-end design of the AI Use Scales Builder project. It provides a reference for developers and stakeholders to ensure consistency across the UI.

Primary tools and frameworks used:

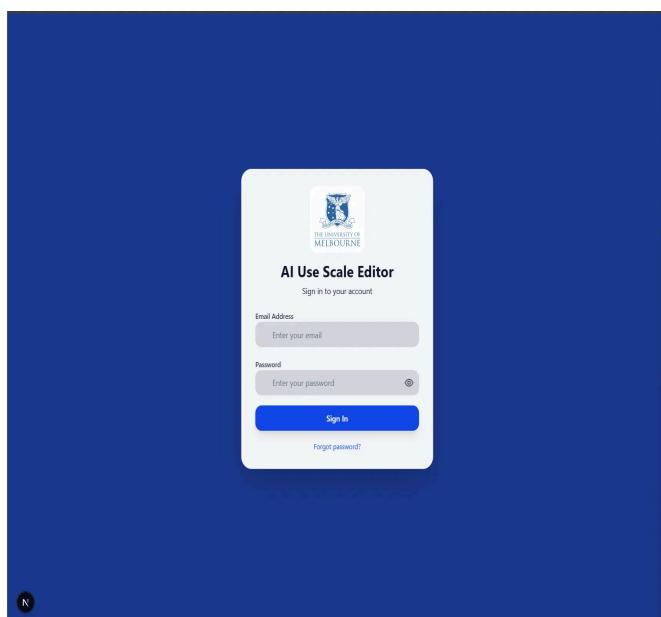
- React
- Next.js
- Tailwind CSS
- Figma

Colour Scheme:

- Blue 900 and white/95 are primarily used throughout in line with the University of Melbourne's theme.

Log-in Page

Purpose: Allows users to authenticate with their email and password before accessing the dashboard. Provides error feedback for failed logins and handles password visibility toggling.

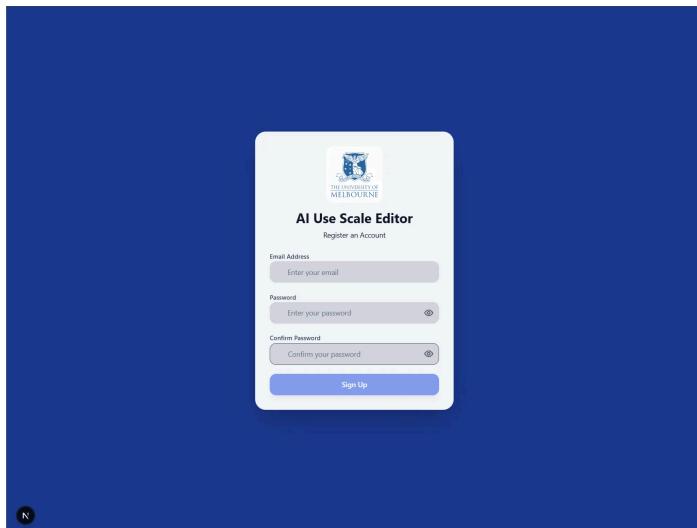


Key Components:

- Login Card: White, semi-transparent, rounded corners, shadow, and slight hover scaling effect.
- Error Message: If the user incorrectly inputs their password, a red error box with fade-in animation will be shown.
- Email Address: Placeholder text “Enter your email”.
- Password: Placeholder text: “Enter your password”
- Sign-in Button: Blue button with white text, rounded edges, shadow.
- “Forgot password?” link in blue with underline hover effect.

Registration Page

Purpose: Allows new users to create an account by providing required details, enabling them to log in and access the dashboard. Includes validation and feedback for successful or failed registration attempts.



Key components:

- Very much the same as the login page, but now including an additional "confirm password" component.
- The sign-up button replaces the sign-in button.
- Also an option to choose your role (admin, subject coordinator, tutor)

Home Page: Students/Tutors/Subject Coordinators/System Admin

Purpose: Serves as the main landing page after login, providing a personalised overview of the system based on the user's role. It allows users to quickly access relevant tools, notifications, and resources while giving a clear summary of their responsibilities and available actions.

- Design structure will be a single homepage component (with role-based branching)

A screenshot of the "AI Use Guidelines Builder" home page. At the top, it says "Welcome back, Dr. Sarah Johnson" and "Manage your AI guidelines and help students understand appropriate AI use." There are three navigation tabs: "My Templates" (12), "Active Subjects" (5), and "Assigned Templates" (8). Below these are "Quick Actions" with links to "Create New AI Guidelines", "View My Templates", "Browse All Templates", and "Manage Subjects". The "My Subjects" section shows three subjects: "Advanced Writing" (ENG401, 28 students, 3 templates), "Chemistry 101" (CHM101, 45 students, 2 templates), and "Business Ethics" (BUS301, 32 students, 3 templates). A "View All" link is also present.

Key Components:

- Personalised welcome message.
- Role-specific shortcuts or cards for quick navigation.
- Notifications or alerts area.
- Access to profile settings and logout (this will most likely be in the sidebar).

The granular details of the home page's features will depend on the type of user as outlined below.

- Tutors/Subject Coordinators: Overview of courses and assessments that they manage, connect to pages where they can create, edit, or duplicate AI using scale templates.
- System Admin: Manage users through backend, create and upload base templates to template repository, statistics of the template usages

Dashboard Page

Purpose: Displays a user-specific overview of AI Use Scale templates, allowing quick access to personal and all templates. Provides navigation to create new templates, view/edit existing ones, and manage the user profile.

Template Name	Subject Code	Semester	Year	Version	Creator Name	Type	Publishable?	Actions
Code Assignment AI Use Template	COMP10001	1	2024	v0	Ben Connor	Template	Yes	<button>Preview</button> <button>Duplicate</button>
Database Project AI Use Template	COMP20008	2	2024	v0	Ben Connor	Template	Yes	<button>Preview</button> <button>Duplicate</button>
Student Group Project AI Use Template	COMP30022	2	2025	v0	Ben Connor	Template	Yes	<button>Preview</button> <button>Duplicate</button>

Key Components:

- A sidebar on the left-hand side of the page, bg-blue-900, contains the University of Melbourne Logo.
- Subpages include: Homepage (not shown), My Templates (Personal Templates) and All Templates (Community Templates)

The template table contains a button option to create a new template.

Table displays user's accessible templates and relevant information, with options to preview and duplicate.

Template Builder Page

Purpose: This page lets instructors create and edit assessment guidelines around AI use. It captures metadata (title, subject, version, etc.), allows defining multiple AI use levels, and saves the final template to the backend.

The screenshot shows the 'AI Use Scales Builder' interface. At the top, there's a header with the title and a logo. Below it, a sidebar on the left lists 'AI Use Scale Repository' and 'Writing'. A main content area contains a table for defining AI use levels. The table has columns for 'Task', 'AI Assessment Use Levels', 'Instructions to Students', 'Examples', 'AI Generated Content', and 'AI'. There are four rows in the table, each corresponding to a different AI use level: 'No AI Use Permitted', 'AI for Research & Brainstorming Only', 'AI as Writing Assistant', and 'Collaborative AI Use Encouraged'. Each row includes detailed instructions for students and examples of AI-generated content. At the bottom, there are buttons for 'Export Guidelines' and 'Save Guidelines'.

Key Components:

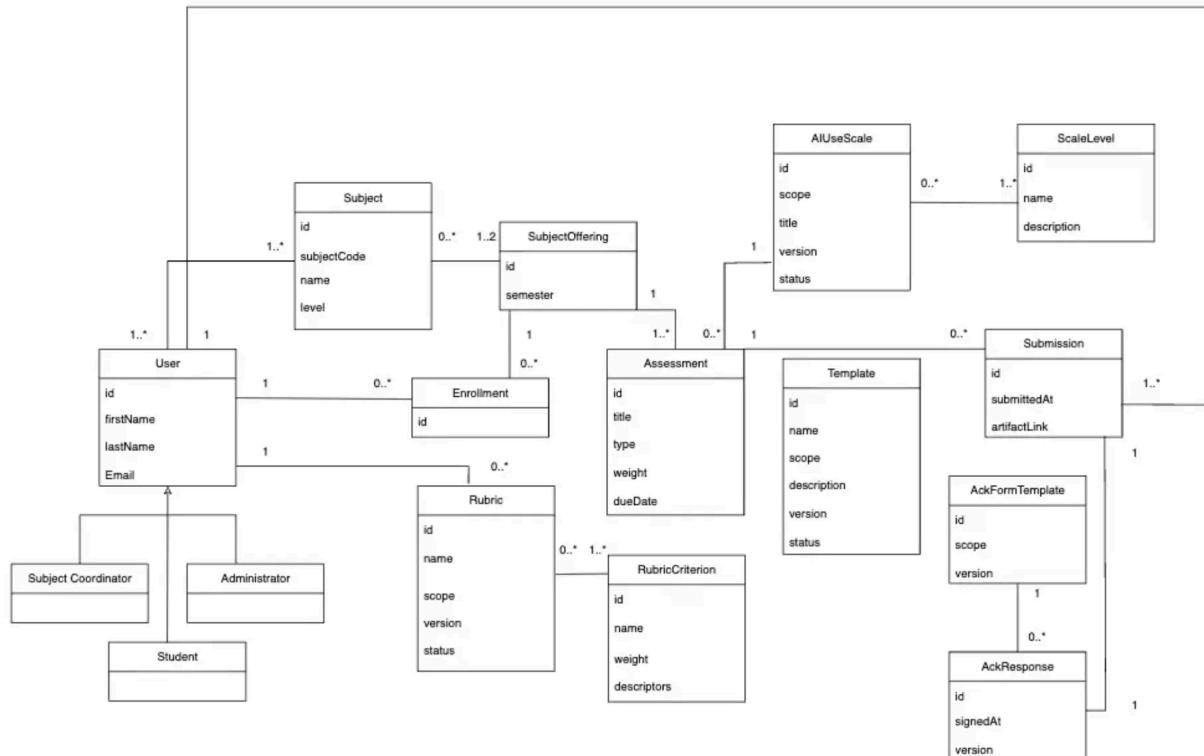
- Left-hand side template repository containing pre-made templates from the database
- Template metadata fields provided at the top.
- Option to manage and add use case levels.
- Template builder contains editable fields relating to different components of the AI guideline.
- Option to export and save guidelines is provided at the bottom.

Architecture Goals and Constraints

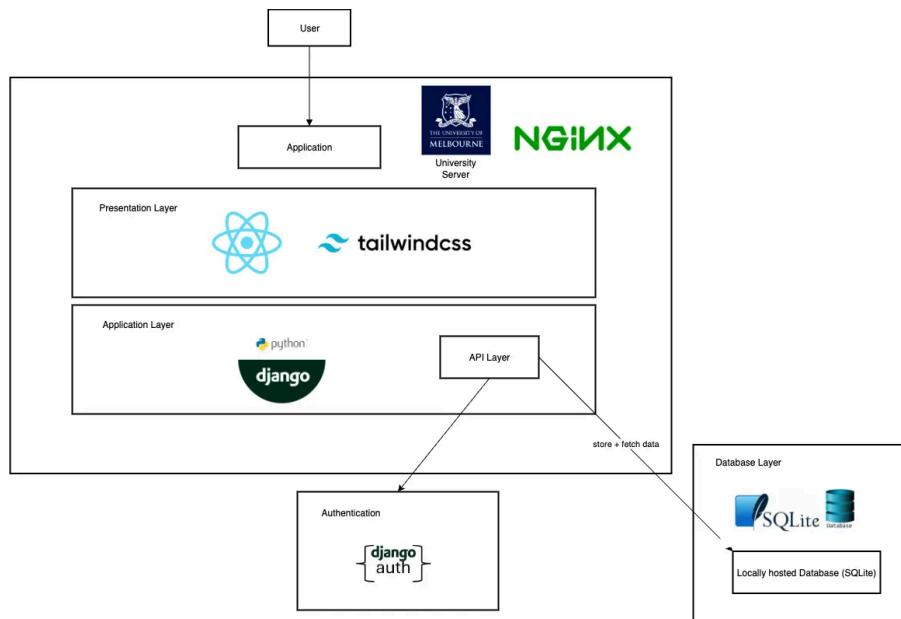
Requirement	What	Why	How
Authentication	The system should ensure that users access different features of the system depending on their role (e.g. student, professor) and verify who they say they are.	To protect unauthorised access to features and data of the system.	Use external authentication systems (e.g. Google, OAuth2, university SSO), requiring the user to login with correct login info. Restrict logins to only from university VPN.
Confidentiality	The system should ensure that each user's created AI scale tables are only accessible to their chosen audiences.	To ensure sensitive data is protected and only visible to trusted users.	Ensure that the data is only displayed to the user if it is visible to them (either by choice of another user or the user owns that data).
Scalability	The system should ensure that multiple concurrent users (up to 10) can handle high traffic during assignment release periods (up to 1000 visits per day) and don't cause system crashes and glitches.	To ensure smoothness of user experience and prevent data loss and inconsistency.	Use good cloud hosts and web servers to host the system (e.g. horizontal scaling and load balancing if needed in high load conditions)
Data persistence	System should ensure AI use scale data is saved and can be accessed later by the owner	To prevent data loss and contamination.	Use a database to store information
Data consistency	The system should ensure that multiple users editing the same template do not interfere with each other.	To prevent data contamination and unexpected behaviours with data being overridden at the same time.	Use a database & locking for concurrency critical areas.
Usability	System should allow smooth creation & editing experience of AI use scales	To improve user experience	A responsive and smooth UI.

Speed	The system should allow sharing of AI use scales to students within 2 minutes.	To improve user experience and accessibility.	Good API design to allow fast sharing of AI use scales, low latency database retrieval and writes.
Authorisation	Enforce fine-grained permissions by role (student/instructor/admin) and context (own class, own scale).	Least-privilege access and safer sharing.	Role/permission tables
Non-repudiation	Immutable event trail for scale edits, publications, acknowledgements, and submissions.	Academic integrity investigations & compliance.	Time stamps.
Privacy	User data is not shared to third parties.	Compliance and privacy of institutional data.	Use a locally hosted database (e.g. SQLite) and deploy the website on a Unimelb server and only allow university VPN users to access it.

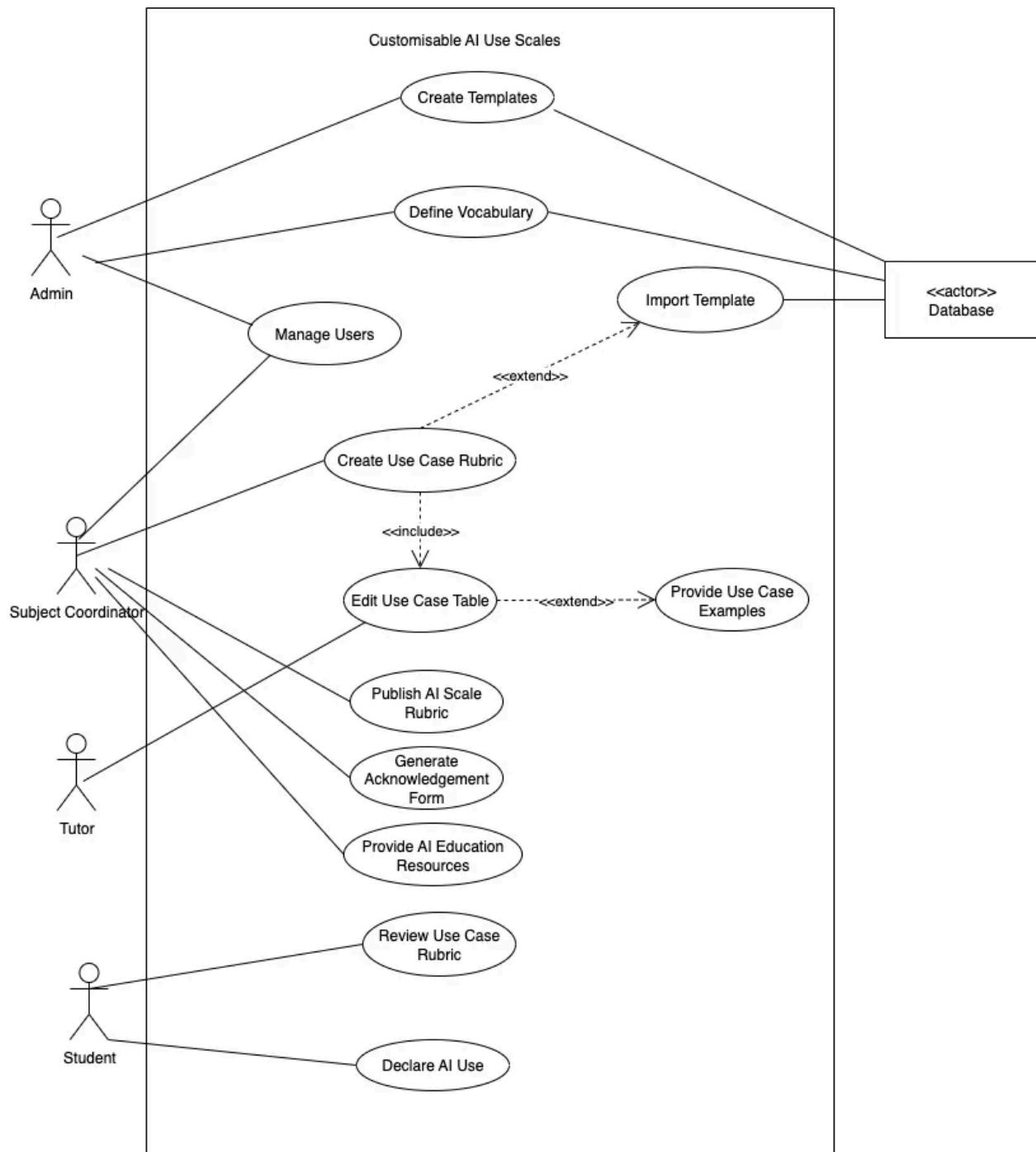
Domain Model



System Architecture Diagram

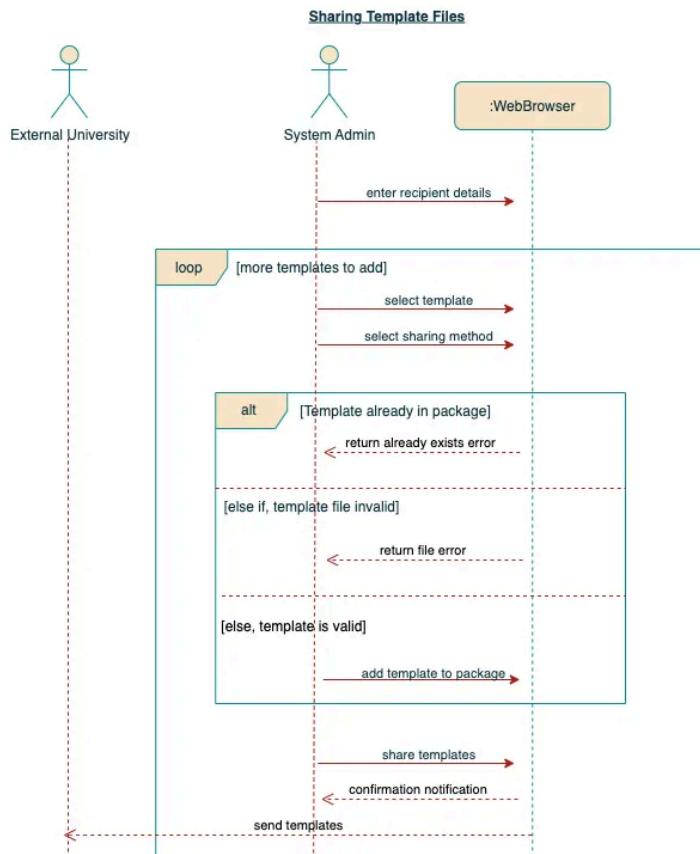
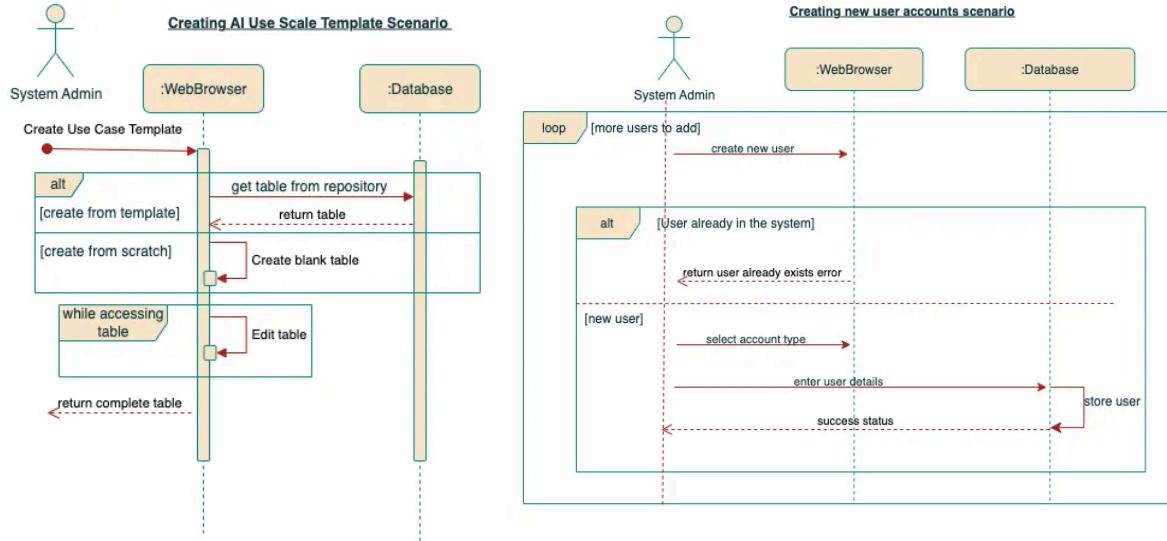


Use Case Diagram

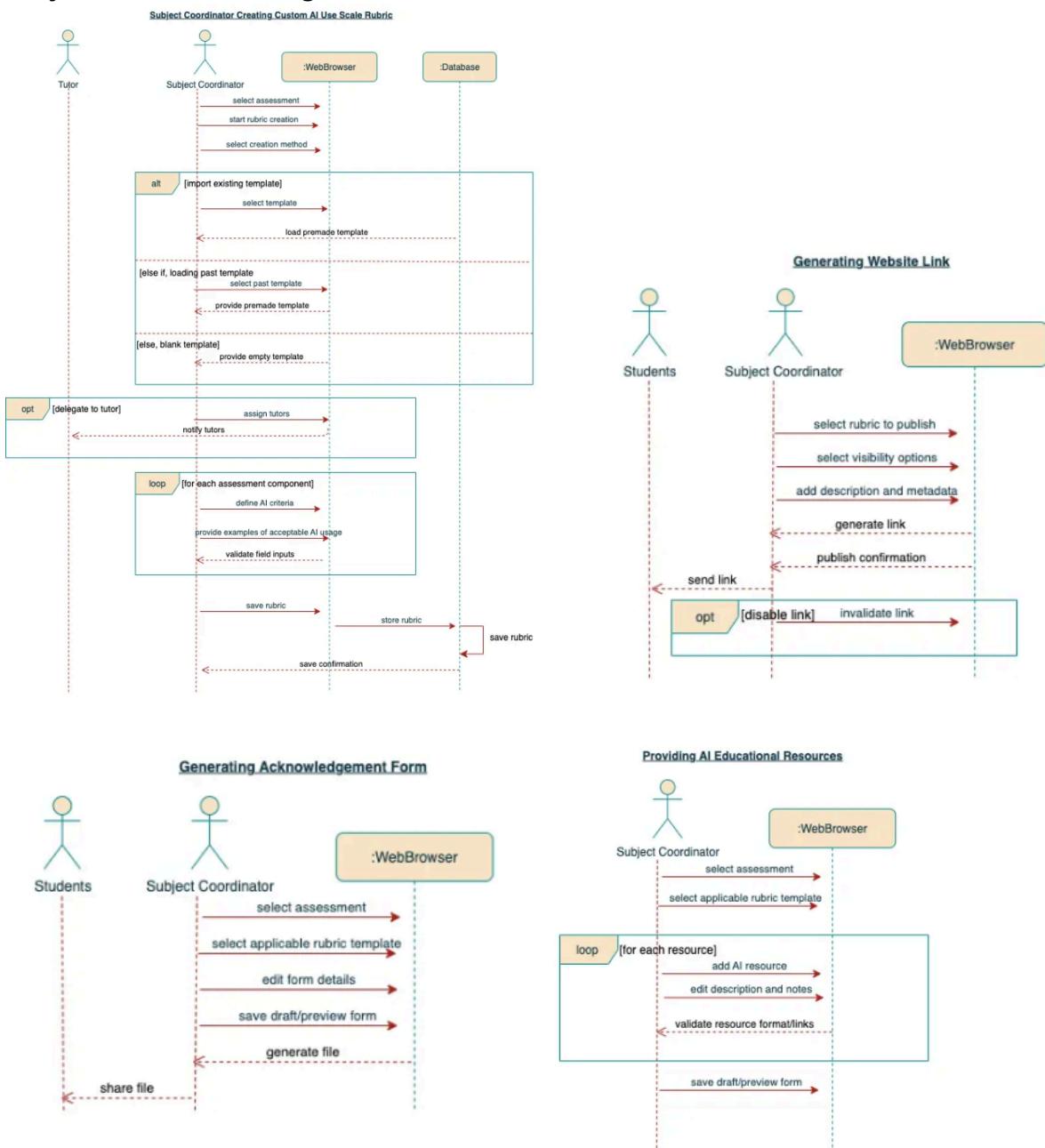


System Sequence Diagrams

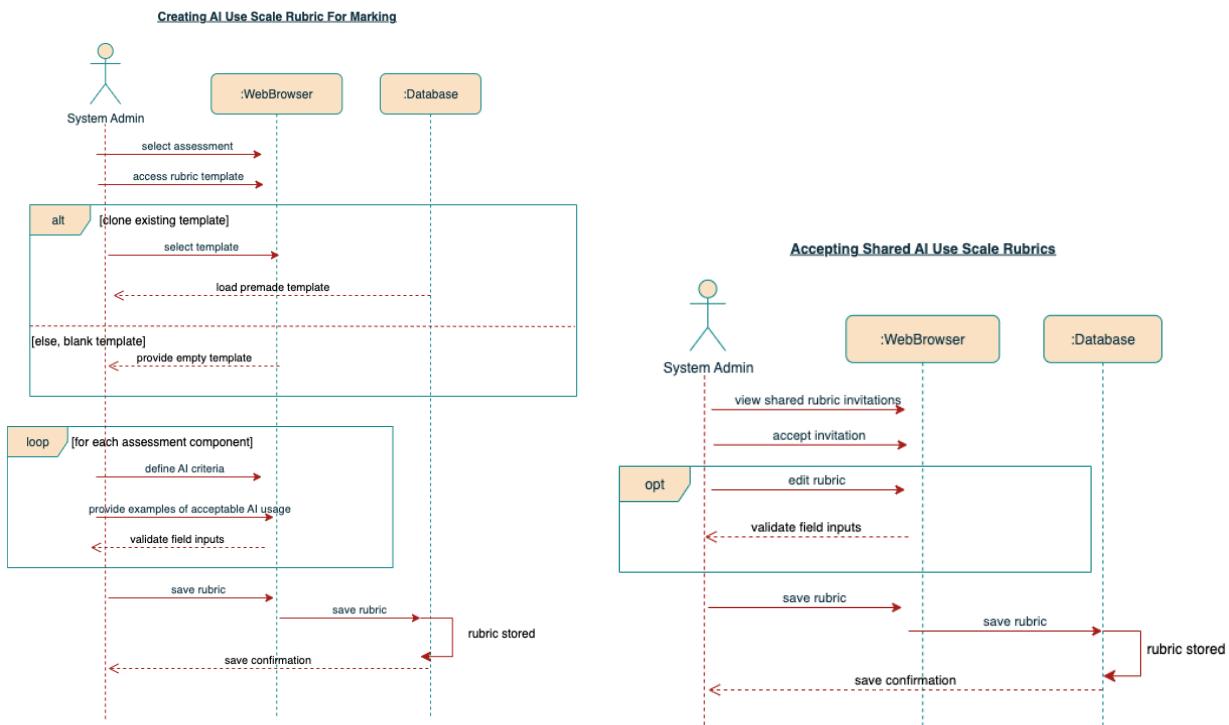
System Admin Diagrams:



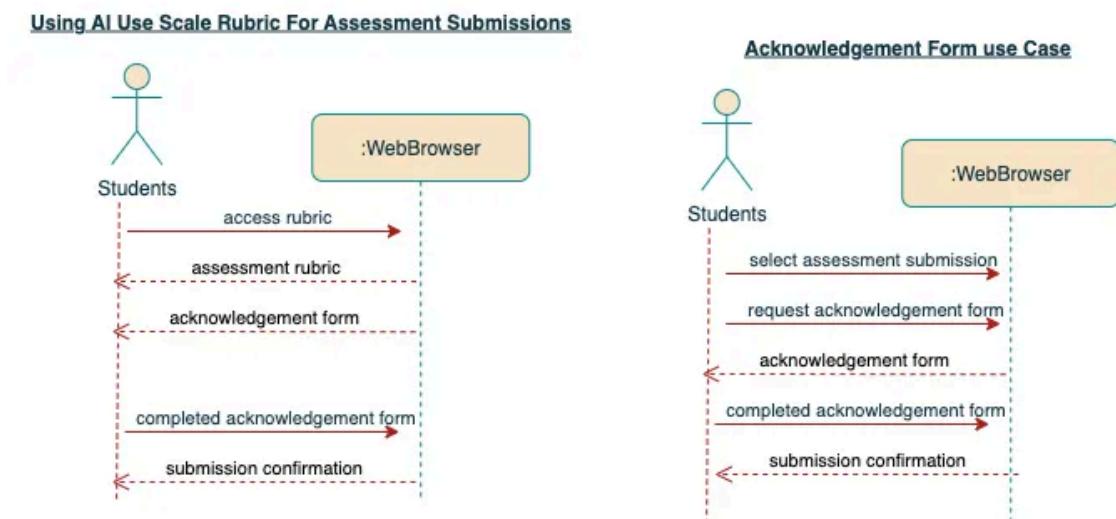
Subject Coordinator Diagrams:



Tutor Diagrams:



Student Diagrams:



User Manual

This section provides a step-by-step instruction to help guide users through this AI Use Scales platform efficiently. We hope that users of this application will be able to create and manage AI-related assessment guidelines, while ensuring consistent policy alignment across academic contexts.

Our project intends for two types of users:

- **Staff**
- **System Admins**

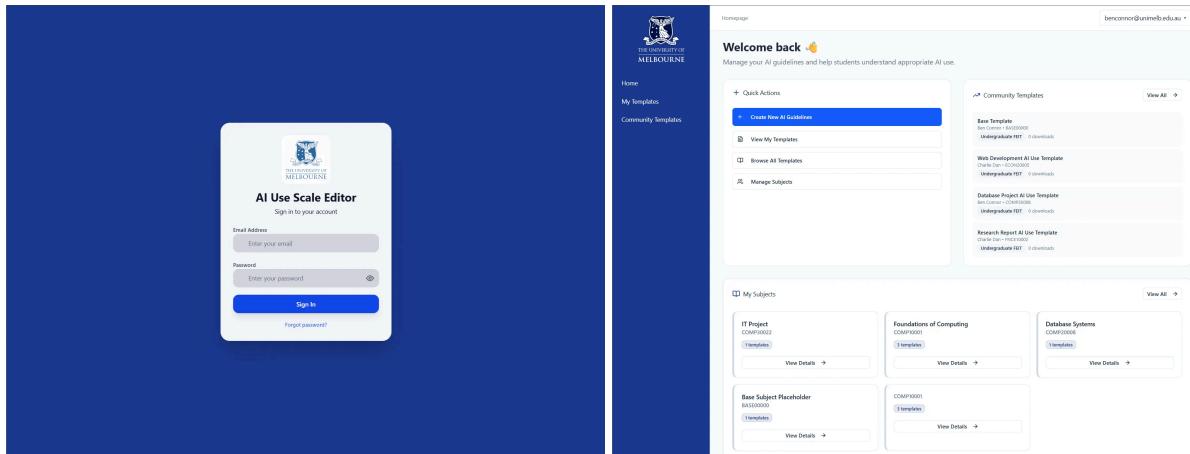
Staff User Guide

Below is a step-by-step guide for the staff users of this platform, including subject coordinators and tutors. The primary goals for these users is to:

- create and edit customisable AI use scale rubrics for assessments.
- duplicate/import community templates.
- generate an AI use scale rubric as a downloadable excel file so that students can view it.

Before users may perform any of these actions, they must first sign in through the log in page with their details. Refer to the README file to see how to add a user to the template.

Users will then be brought to the staff homepage, which contains many ease-of-access features for convenience.



Creating a New Template

- Click “+ Create New AI Use Scale” on the dashboard.
- Users will then be brought to the “templateBuilder” page where they can fill out the details relevant to the scope of the target assessment.

The screenshot shows the 'AI Use Scales Builder' interface. On the left, there's a sidebar for 'AI Use Scale Repository' with a 'Base Template' section containing 'Standard scale' and '4 levels'. A 'Use This Scale' button is present. The main area has tabs for 'Assessment Title' (set to 'AI Use Guidelines for Assessment'), 'Assessment Scope' (set to 'Assignment'), 'Version' (0), 'Subject Code' (COMP1001), 'Semester' (I), and 'Year' (2025). Below these are dropdowns for 'Select a version...' and 'Publishable?' (unchecked). A 'Manage AI Use Levels' button is located at the bottom right of this section. The central part of the screen displays a table with four rows. Each row contains a 'Task' column (with placeholder text 'Task title / description...'), an 'AI Assessment Use Levels' column (with placeholder text 'AI use level name...'), an 'Instructions to Students' column (with placeholder text 'Clear instructions for students about what AI use is permitted...'), an 'Examples' column (with placeholder text 'Specific examples of permitted AI use...'), and an 'AI Generated Content' column (with placeholder text 'Describe/attach AI-generated parts, constraints, or links...'). Each row has a checkbox in the first column and a 'Manage AI Use Levels' button in the last column. At the bottom of the table, it says '4 AI use levels defined + Assignment'. There are 'Export Format' buttons for Excel and PDF, and a 'Save Guidelines' button.

- Add AI use levels as necessary by clicking the “Manage AI Use Levels” button.

This screenshot is similar to the one above, but a modal window titled 'Manage AI Use Levels' is open over the main table. The modal contains four rows, each with a 'Level name' input field and a 'Manage AI Use Levels' button. The main table below is identical to the one in the first screenshot, showing four rows of AI use levels. The bottom of the table still says '4 AI use levels defined + Assignment'.

- Click the “Save Guidelines” button at the bottom of the page once finished. The saved template will now appear in the user’s “My Templates” page.

Editing a Template

- Navigate to the “MyTemplates” page.

The screenshot shows the 'My Templates' section of a university's digital platform. At the top, there's a search bar with placeholder text 'Search by name, subject code, or owner...'. Below it is a table listing five AI Use Scales:

Template Name	Subject Code	Semester	Year	Version	Creator Name	Publishable?	Actions
AI Use Guidelines for Assessment	COMP10001	1	2025	v1	Ben Connor	No	<button>Edit</button> <button>Delete</button>
Base Template	BASE00000	1	2024	v1	Ben Connor	Yes	<button>Edit</button> <button>Delete</button>
Code Assignment AI Use Template	COMP10001	1	2024	v0	Ben Connor	Yes	<button>Edit</button> <button>Delete</button>
Database Project AI Use Template	COMP20008	2	2024	v0	Ben Connor	Yes	<button>Edit</button> <button>Delete</button>
Student Group Project AI Use Template	COMP30022	2	2025	v0	Ben Connor	Yes	<button>Edit</button> <button>Delete</button>

- Identify the target template and press the “Edit” button.
- Change the version number of the template, indicating a new edit of the template.

The screenshot shows the 'AI Use Scales Builder' interface. On the left, there's a sidebar for 'AI Use Scale Repository' with a 'Writing' section containing a 'Base Template' (Standard scale, 4 levels) and a 'Use This Scale' button. The main area has tabs for 'Assessment Title' (set to 'AI Use Guidelines for Assessment') and 'Assessment Scope' (set to 'Assignment'). Below these are fields for 'Version' (1), 'Subject Code' (COMP10001), 'Semester' (1), and 'Year' (2025). A dropdown menu for 'AI Use Levels' shows '0' and '1' selected. To the right is a table for defining AI use levels for different tasks:

Task	AI Assessment Use Levels	Instructions to Students	Examples	AI Generated Content	AI
Task title / description...	AI for Research & Brainstorming Only	You may use AI tools for initial research, topic exploration, and brainstorming ideas. However, all analysis, asides/writing, and final work must be your own.	Using ChatGPT to understand complex topics, generating research questions, exploring different perspectives on a subject	Add details here...	<input checked="" type="checkbox"/>
Task title / description...	No AI Use Permitted	This assignment is completed entirely without AI assistance. This level requires that students rely solely on their knowledge, understanding, and skills. AI must not be used at any point during the assessment.	Traditional exams, mid-term essays, mathematical problem-solving without computational aids, original creative writing	Add details here...	<input checked="" type="checkbox"/>
Task title / description...	Collaborative AI Use Encouraged	All tools are encouraged as collaborative partners. You may use AI for research, drafting, analysis, and refinement while demonstrating critical evaluation of AI outputs.	Co-writing with AI, using AI for data analysis, AI-assisted coding projects, collaborative problem-solving with AI	Add details here...	<input checked="" type="checkbox"/>
Task title / description...	AI as Writing Assistant	AI tools may be used to assist with writing tasks such as grammar checking, style suggestions, and structural feedback. The core ideas and arguments must be your own.	Using Grammarly for editing, ChatGPT for feedback on draft structure, AI tools for citation formatting	Add details here...	<input checked="" type="checkbox"/>

At the bottom, there's a note '4 AI use levels defined + Assignment', export options ('Export Format: Excel PDF Export as Excel'), and a 'Save Guidelines' button.

- Click “Save Guidelines” once finished. The saved version will now update the template under the user’s “My Templates”.

Deleting a Template

- Navigate to the “My Templates” page.

- Identify the template that you want to delete and click “**Delete**”.

Exporting a Template

- At the bottom of the templateBuilder page, choose the desired export format.
- Click “**Export as Excel/PDF**”.



- A download prompt will begin shortly.

Importing a Community Template

- Navigate to the “Community Templates” page.

A screenshot of a web page titled "Community Templates". On the left is a sidebar with the University of Melbourne logo and links for "Home", "My Templates", and "Community Templates". The main area shows a table of templates with columns: Template Name, Subject Code, Semester, Year, Version, Creator Name, Publishable?, and Actions. Each row has a "Duplicate" button. Below the table, it says "Showing 6..." and "Or create your own! + Create New AI Use Scale".

Template Name	Subject Code	Semester	Year	Version	Creator Name	Publishable?	Actions
Base Template	BASE00000	1	2025	v1	Ben Connor	Yes	<button>Duplicate</button>
Web Development AI Use Template	ECON20005	1	2025	v1	Charlie Dan	Yes	<button>Duplicate</button>
Database Project AI Use Template	COMP20008	1	2025	v1	Ben Connor	Yes	<button>Duplicate</button>
Research Report AI Use Template	FNCE10002	1	2025	v1	Charlie Dan	Yes	<button>Duplicate</button>
Code Assignment AI Use Template	COMP10001	1	2025	v1	Ben Connor	Yes	<button>Duplicate</button>
Student Group Project AI Use Template	COMP30022	1	2025	v1	Ben Connor	Yes	<button>Duplicate</button>

- Identify the desired template to import and click “**Duplicate**”.
- An editable version of the community template will then appear in the “My Template” page.

Viewing Active Templates

- At the homepage, users will have a “My Subjects” section that shows the subjects they manage.

- Click “View Details” to see the active templates for that subject.

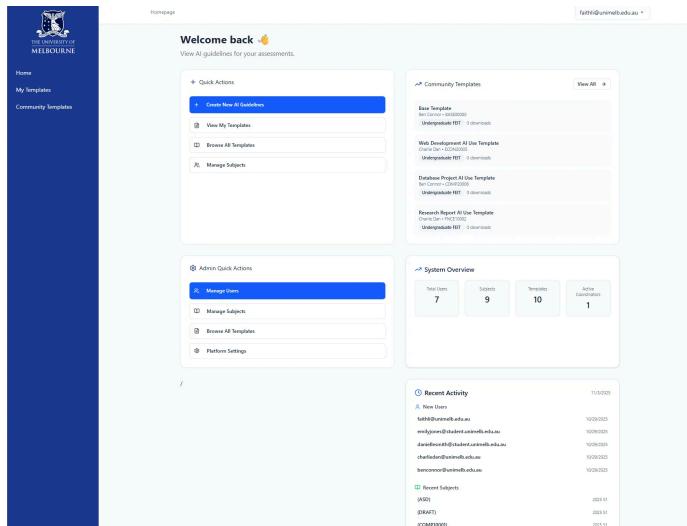
The screenshot shows the University of Melbourne AI Guidelines portal. At the top, there's a dark blue sidebar with the university logo and navigation links like 'Home', 'My Templates', and 'Community Templates'. The main area has a light gray background with a 'Welcome back' message and a 'Manage your AI guidelines and help students understand appropriate AI use.' note. On the left, there's a 'Quick Actions' panel with 'Create New AI Guidelines', 'View My Templates', 'Browse All Templates', and 'Manage Subjects'. Below that is a 'My Subjects' section listing 'IT Project' (COMP30022, 2 templates) and 'Base Subject Placeholder' (BASE30000, 1 template). A central modal window is open, titled 'Subject Information' for 'Subject Code: COMP30022'. It shows 'Templates: 2'. Under 'Assigned AI Guidelines Templates', there are two entries: 'Student Group Project AI Use Template (1)' (Active, Assigned: 1/1/2023, Due: 12/9/2023) and 'Student Group Project AI Use Template (1)' (Active, Assigned: 1/1/2023, Due: 12/9/2023). There are also sections for 'Community Templates' and 'View All' buttons.

System Admins User Guide

Below is a step-by-step guide for the admin users of this platform. The primary goals for these users is to:

- create customisable AI use scale community templates for other users to import.
- manage community templates.

Similar to other users, admins will first need to sign in with their details at the log in page before they are brought to the system admin homepage:



Creating a Community Template

- Click “+ Create New AI Use Scale” on the dashboard.
- Customise the template as outlined in “**Creating a New Template**” in the Staff User Guide.
- Ensure that “**Publishable?**” has been ticked.

A screenshot of the AI Use Scales Builder interface. It shows a form for creating a new assessment titled "demo". The form includes fields for Assessment Title, Version, Subject Code, Semester, Year, and a "Published" checkbox which is checked. Below the form is a table with four rows, each representing a task. The columns are Task, AI Assessment Use Levels, Instructions to Students, Examples, AI Generated Content, and AI. The tasks are: "Task title / description... AI use level: none..." with instructions about completing entire assignment; "Task title / description... AI use level: none..." with instructions about using ChatGPT to understand assignment requirements and answer questions; "Task title / description... AI use level: none..." with instructions about using AI for problem-solving; and "Task title / description... AI use level: none..." with instructions about using AI for research and writing. Each row has a "Describe/Link" button next to the AI Generated Content column.

- Save the template once finished. The template will now be viewable in “My Templates” and “Community Templates”.

My Templates

Template Name	Subject Code	Semester	Year	Version	Creator Name	Publishable?	Actions
demo	COMP10001	1	2025	v0	Faith Li	Yes	<button>Edit</button> <button>Delete</button>
New AI Use Scale	DRAFT	1	2025	v0	Faith Li	No	<button>Edit</button> <button>Delete</button>

Community Templates

Template Name	Subject Code	Semester	Year	Version	Creator Name	Publishable?	Actions
demo	COMP10001	1	2025	v1	Faith Li	Yes	<button>Duplicate</button>
Base Template	BASE00000	1	2025	v1	Ben Connor	Yes	<button>Duplicate</button>
Web Development AI Use Template	ECON20005	1	2025	v1	Charlie Dan	Yes	<button>Duplicate</button>
Database Project AI Use Template	COMP20008	1	2025	v1	Ben Connor	Yes	<button>Duplicate</button>
Research Report AI Use Template	FNCE10002	1	2025	v1	Charlie Dan	Yes	<button>Duplicate</button>
Code Assignment AI Use Template	COMP10001	1	2025	v1	Ben Connor	Yes	<button>Duplicate</button>
Student Group Project AI Use Template	COMP30022	1	2025	v1	Ben Connor	Yes	<button>Duplicate</button>

Editing a Community Template

The action of editing a community template is the same as the process outlined in “[Editing a Template](#)” in the Staff User Guide.

Deleting a Community Template

- Navigate to “My Templates” and locate the template that you intend to delete.
- Click the “Delete” button.
- The template will then disappear from the “Community Templates” page.

Key Classes and Application Layers

Frontend

All relevant frontend files are in `/src/app`

Authentication related api helper functions: `/src/app/authentication/auth.tsx`

Community templates page: `/src/app/communityTemplates/page.tsx`

Components: `/src/app/components`

- To increase code readability and reusability, we defined major components in separate files. E.g. the `AIGuidelinesBuilder.tsx` is the page for editing AI use scale templates, `topbar.tsx` defines the top navigation bar

Home Dashboard page: `/src/app/homePage/page.tsx`

Login page: `src/app/login/page.tsx`

My Templates page: `src/app/myTemplates/page.tsx`

Registration page: `src/app/registration/page.tsx`

Backend

All relevant backend files are in `/ai_scale_app`

Backend testing files are in `/ai_scale_app/tests`

`ai_scale_app/admin.py`

- Imports the models (schemas) defined in `models.py`

`ai_scale_app/models.py`

- Defines and generates schemas for our relational database. Whenever a change is made here, you need to use the `makemigrations` and `migrate` command as described in `README.md` to ensure the changes are synced to the actual database.

`ai_scale_app/urls.py`

- Defines the backend API endpoints and what API function defined in `views.py` they correspond to.

`ai_scale_app/views.py`

- Defines all backend API functions.

Licenses

All code developed for this project is **assigned to the client upon handover**. This means the client obtains full rights to use, modify, distribute and sublicense the code without restriction.

Third-party Dependencies and Licences

The application uses several open-source libraries on both the backend and the frontend.

These packages remain under their respective upstream licences; however, they allow redistribution in commercial products.

Backend (Python / Django)

Library	Purpose	Licence
Django	Web framework	BSD 3-Clause
Django REST Framework	API layer for Django	BSD 3-Clause
django-cors-headers	Cross-Origin Resource Sharing middleware	MIT
asgiref	ASGI utilities used by Django	BSD 3-Clause
sqlparse	SQL parsing and formatting library	BSD
pytest	Testing framework (development only)	MIT

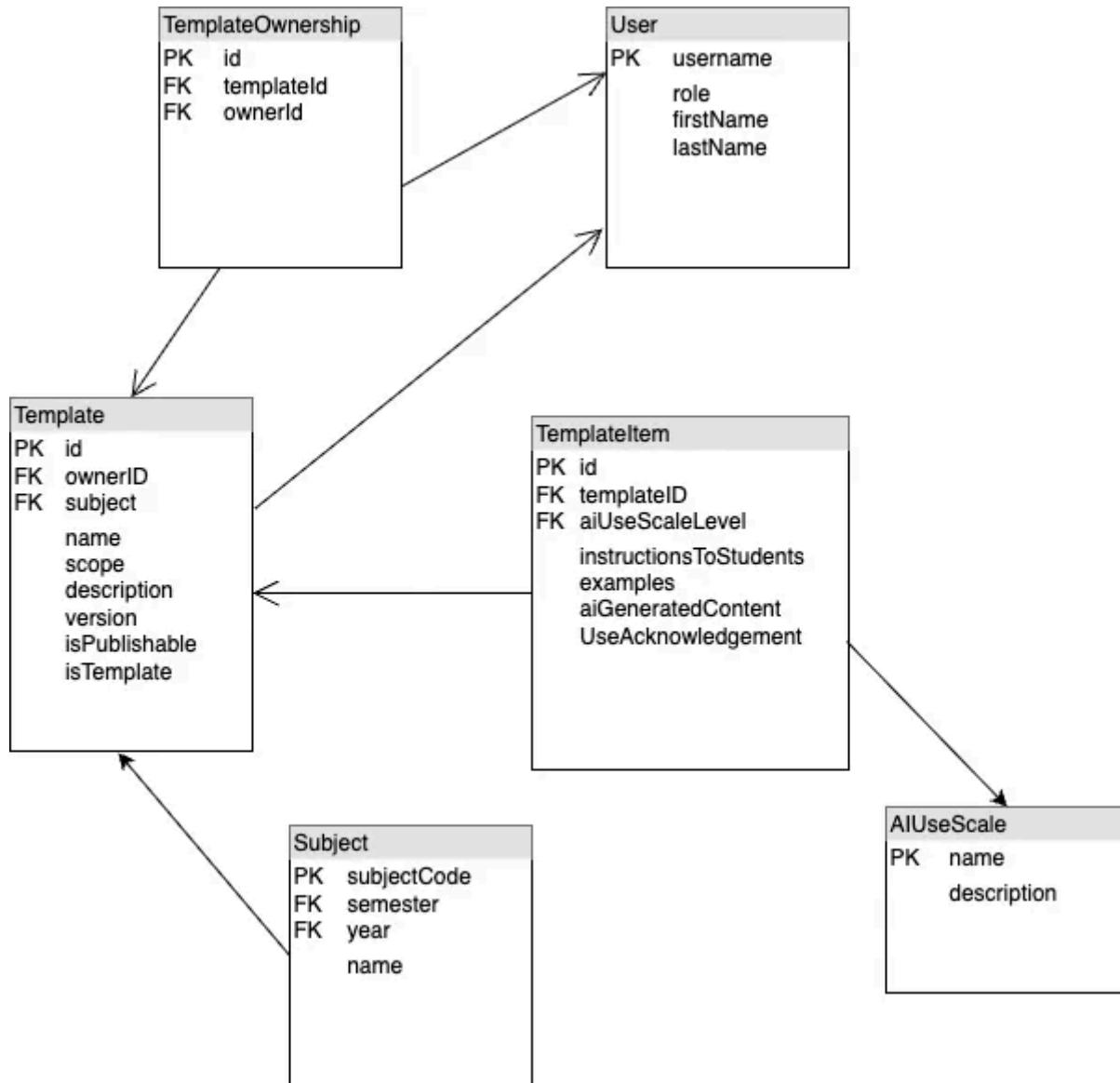
These Python libraries are permissively licensed (BSD or MIT) and require only attribution. They impose no copyleft restrictions, so they can be included in proprietary projects without making the rest of the code open source.

Frontend (Node / React)

Library	Purpose	Licence
<code>@emotion/react</code>	CSS-in-JS styling library for React	MIT
<code>@emotion/styled</code>	Styled components using Emotion	MIT
<code>@mui/material</code>	Material-Design UI component library	MIT
<code>@radix-ui/react-alert-dialog</code>	Accessible alert dialog component	MIT
<code>@radix-ui/react-dialog</code>	Accessible modal dialog component	MIT
<code>@radix-ui/react-label</code>	Accessible label primitive	MIT
<code>@radix-ui/react-scroll-area</code>	Custom scroll area component	MIT
<code>@radix-ui/react-slider</code>	Slider component	MIT
<code>@radix-ui/react-slot</code>	Slot primitive for headless components	MIT
<code>@szhsin/react-menu</code>	Dropdown/menu component	MIT
<code>bootstrap</code>	CSS framework	MIT
<code>class-variance-authority</code>	Utility for variant-based CSS class generation	Apache 2.0
<code>html2pdf.js</code>	Client-side HTML-to-PDF conversion library	MIT
<code>input-otp</code>	One-time-password input component	MIT
<code>lucide-react</code>	Icon set for React applications	ISC
<code>next</code>	React framework for SSR/SSG	MIT
<code>print-js</code>	Client-side printing helper	MIT
<code>radix-ui</code>	Core Radix UI primitives	MIT
<code>react</code>	React core library	MIT
<code>react-bootstrap</code>	Bootstrap components for React	MIT
<code>react-dom</code>	React DOM renderer	MIT
<code>tailwind-merge</code>	Utility for merging Tailwind CSS classes	MIT
<code>xlsx</code>	Spreadsheet library for reading/writing Excel files	Apache 2.0

Database Access

Overview of the data scheme can be viewed via the database schema diagram in the database diagram shown below:



The backend database can be viewed by logging into the backend server as a superuser. You can register yourself as a superuser through the commands described in the README.md

The superuser status allows you to view all information stored in the database (except passwords, these are hashed), and edit them as needed.

Database Structure

The database is a relational schema implemented in Django ORM, using SQLite for local development and compatible with PostgreSQL in production. It supports user management, subject enrolment, AI use templates, acknowledgement forms, and logging of user actions.

Overview

The database consists of 10 main tables, grouped into logical domains:

Domain	Models
User & Access Control	User
Academic Structure	Subject, Enrolment
Template System	Template, TemplateItem, TemplateOwnership, AIUseScale
Acknowledgement Forms	AcknowledgementForm, AcknowledgementFormItem
System Logging	AuditLog

Each entity is interconnected through ForeignKey relationships to maintain referential integrity.

User Model

Table: User

Extends: `AbstractUser` (Django built-in authentication)

Field	Type	Description
<code>id</code>	Integer (PK)	Unique identifier
<code>username, email, password</code>	Inherited from <code>AbstractUser</code>	
<code>role</code>	CharField (max_length=20)	User's role in the system — one of: <i>Student, Staff, Coordinator, Admin</i>

Purpose: Stores all users with a defined role type. Provides authentication and ownership references across the database.

Subject Model

Table: Subject

Field	Type	Description
<code>id</code>	Integer (PK)	Unique subject identifier
<code>subjectCode</code>	CharField (max_length=10)	Course or subject code
<code>semester</code>	PositiveSmallIntegerField	Semester number (1 or 2)
<code>year</code>	PositiveSmallIntegerField	Academic year
<code>name</code>	CharField (max_length=100, optional)	Descriptive name of the subject

Constraints:

- Unique combination of (`subjectCode, year, semester`)
- Indexed for fast lookup

Purpose:

Defines academic units in which templates and acknowledgements can be contextualized.

Enrolment Model

Table: Enrolment

Field	Type	Description
id	Integer (PK)	Unique enrolment ID
subjectId	ForeignKey → Subject	Subject enrolled in
studentId	ForeignKey → User	Student enrolled in the subject

Constraints:

- Each (subjectId, studentId) pair must be unique
- Indexed on subjectId and studentId

Purpose:

Links students to subjects. Supports role-specific access to templates or forms.

AI Use Scale Model

Table: AIUseScale

Field	Type	Description
id	Integer (PK)	Scale ID
name	TextField (unique)	AI usage level label (e.g., “Low Assistance”, “Substantial Generation”)

Purpose:

Defines levels or categories of AI usage to be referenced by templates and acknowledgement forms.

Template Model

Table: Template

Field	Type	Description
<code>id</code>	Integer (PK)	Template ID
<code>ownerId</code>	ForeignKey → <code>User</code>	Owner or creator of the template
<code>name</code>	CharField (max_length=120)	Template title
<code>scope</code>	CharField (optional)	Context or intended use
<code>description</code>	TextField (optional)	Explanation of purpose
<code>subject</code>	ForeignKey → <code>Subject</code> (nullable)	Associated subject, if any
<code>version</code>	PositiveSmallIntegerField (default=0)	Version control for template iterations
<code>isPublishable</code>	BooleanField	Whether template is visible to others
<code>isTemplate</code>	BooleanField	Distinguishes between base template and derived copy

Constraints:

- Unique (`ownerId`, `name`, `version`)
- Indexed for efficient ownership and version lookup

Purpose:

Central structure defining AI use templates shared between users, subjects, and forms.

TemplateOwnership Model

Table: TemplateOwnership

Field	Type	Description
id	Integer (PK)	Ownership record
templateId	ForeignKey → Template	Linked template
ownerId	ForeignKey → User	Co-owner or shared recipient

Constraints:

- Each (templateId, ownerId) pair must be unique
- Indexed on both foreign keys

Purpose:

Allows template sharing and co-ownership across users (e.g., between staff and coordinators).

TemplateItem Model

Table: TemplateItem

Field	Type	Description
id	Integer (PK)	Template item identifier
templateId	ForeignKey → Template	Parent template
task	TextField	Description of the student task
aiUseScaleLevel	ForeignKey → AIUseScale (nullable)	AI usage classification
instructionsToStudents	TextField (optional)	Instructions for students
examples	TextField (optional)	Example of acceptable work
aiGeneratedContent	TextField (optional)	Example of AI-produced content
useAcknowledgement	BooleanField	Whether acknowledgment is required

Purpose:

Defines the granular components of a template, linked to a specific AI use scale level.

AcknowledgementForm Model

Table: AcknowledgementForm

Field	Type	Description
<code>id</code>	Integer (PK)	Acknowledgement form ID
<code>templateId</code>	ForeignKey → Template	Template this form is derived from
<code>name</code>	CharField (max_length=120)	Form name
<code>subject</code>	ForeignKey → Subject (nullable)	Associated subject

Constraints:

- Unique (`templateId`, `name`) combination
- Indexed on (`templateId`, `name`)

Purpose:

Represents acknowledgment forms that students fill in based on a given template.

AcknowledgementFormItem Model

Table: AcknowledgementFormItem

Field	Type	Description
<code>id</code>	Integer (PK)	Acknowledgement item ID
<code>ackFormId</code>	ForeignKey → AcknowledgementForm	Linked form
<code>aiToolsUsed</code>	TextField (optional)	Tools used by the student
<code>purposeUsage</code>	TextField (optional)	Purpose of AI use
<code>keyPromptsUsed</code>	TextField (optional)	Prompts or inputs provided to AI

Purpose:

Captures granular responses within an acknowledgment form — each entry corresponds to one AI-assisted task.

AuditLog Model

Table: AuditLog

Field	Type	Description
<code>id</code>	Integer (PK)	Log ID
<code>user</code>	ForeignKey → User (nullable, SET_NULL)	Actor performing the action
<code>action</code>	CharField (choices: CREATE, UPDATE, DELETE)	Type of operation
<code>model_name</code>	CharField	Affected model
<code>object_id</code>	PositiveIntegerField	Primary key of the affected record
<code>details</code>	JSONField (optional)	Change details (before/after snapshots)
<code>timestamp</code>	DateTimeField	Time of the action

Purpose:

Tracks all CRUD actions performed within the system for auditing and debugging.

Data Documentation

All test data and dummy user data are stored in
`/ai_scale_app/table_data/dummy_data.json`

This json file loads information directly into the database. Information on how to load this file into the database are described in the README.md. During data loading, the dummy passwords are hashed when they are stored in the database.

All dummy user passwords are currently set to **12345678** for convenience of testing. Different dummy users with different user roles (admin, subject coordinator, staff) are provided.

Deployment Guidelines

Prerequisites

Create these accounts before starting:

1. GitHub account: <https://github.com/signup>
 2. Vercel account: <https://vercel.com/signup> (sign up with GitHub)
 3. Render account: <https://render.com/register> (sign up with GitHub)
 4. AWS account: <https://aws.amazon.com/free> (requires credit card)
-

Step 1: Fork the Repository

1. Go to the repository URL provided by your vendor
 2. Click the Fork button (top right)
 3. Select your GitHub account
 4. Click Create fork
 5. You now have your own copy at `github.com/YOUR-USERNAME/REPO-NAME`
-

Step 2: AWS S3 Setup

Create S3 Bucket

1. Log into AWS Console: <https://console.aws.amazon.com/>
2. Search for S3 in the top search bar
3. Click Create bucket

4. Configure:
 - Bucket name: your-company-ai-scale (must be globally unique)
 - Region: ap-northeast-1 (or closest to you)
 - Block Public Access: Keep all boxes checked
5. Click Create bucket
6. Save your bucket name for later

Create IAM User

1. Search for IAM in the AWS search bar
2. Click Users in left sidebar
3. Click Create user
4. User name: django-s3-access
5. Click Next

Create Access Policy

1. Select Attach policies directly
2. Click Create policy (opens new tab)
3. Click JSON tab and insert the following

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [
```

```

"s3:PutObject",
"s3:GetObject",
"s3:DeleteObject",
"s3>ListBucket"

],

"Resource": [
"arn:aws:s3:::YOUR-BUCKET-NAME",
"arn:aws:s3:::YOUR-BUCKET-NAME/*"

]
}

]
}

```

4. Replace YOUR-BUCKET-NAME with your actual bucket name
5. Click Next
6. Policy name: DjangoS3AccessPolicy
7. Click Create policy

Attach Policy to User

1. Return to user creation tab
2. Refresh the policy list
3. Search for DjangoS3AccessPolicy
4. Check the box next to it

5. Click Next
6. Click Create user

Create Access Keys

1. Click on django-s3-access username
2. Click Security credentials tab
3. Scroll to Access keys section
4. Click Create access key
5. Select Application running outside AWS
6. Check the confirmation box
7. Click Next
8. Description: Render Django Backend
9. Click Create access key
10. Click Download .csv file and save securely
11. Copy both values to a secure note:
 - Access key ID: starts with AKIA
 - Secret access key: long random string
12. Click Done

Step 3: Deploy Backend on Render

Create Web Service

1. Go to <https://dashboard.render.com/>

2. Click New +
3. Select Web Service
4. Find your forked repository and click Connect
5. If you don't see your repo, click Configure account and grant access

Configure Service

Fill in these settings:

Name: ai-scale-backend
Region: Oregon (US West)
Branch: main
Root Directory: backend
Runtime: Python 3
Build Command: pip install -r requirements.txt
Start Command: bash start.sh
Instance Type: Free

Add Environment Variables

Click Add Environment Variable for each:

ALLOWED_HOSTS=URL Of Render Deployment

AWS_ACCESS_KEY_ID=Your AWS Access Key

AWS_REGION=ap-northeast-1

AWS_SECRET_ACCESS_KEY=Your Secret Access Key

AWS_STORAGE_BUCKET_NAME=Your S3 Bucket Name

CORS_ALLOWED_ORIGINS=URL of Vercel Deployment

CSRF_TRUSTED_ORIGINS=URL of Vercel Deployment

DEBUG=False

MEDIA_ROOT=/data/media

PORt=10000

```
PYTHON_VERSION=3.12.2  
S3_BUCKET_NAME=Your S3 Bucket Name  
SECRET_KEY=Your Secret Key  
SQLITE_PATH=/data/db.sqlite3  
STATIC_ROOT=/data/static
```

Deploy

1. Click Create Web Service
 2. Wait 3-5 minutes for deployment
 3. Wait for status to change to Live (green dot)
 4. Copy your backend URL from the top of the page
Example: <https://ai-scale-backend.onrender.com>
 5. Test it by visiting: <https://YOUR-BACKEND-URL.onrender.com/health/>
You should see: {"status": "ok", "timestamp": "...", ...}
-

Step 4: Deploy Frontend on Vercel

Create Project

1. Go to <https://vercel.com/dashboard>
2. Click Add New
3. Select Project
4. Find your forked repository and click Import
5. If you don't see your repo, click Adjust GitHub App Permissions

Configure Project

Framework Preset: Next.js (auto-detected)
Root Directory: frontend (click Edit if needed)
Build Command: npm run build
Output Directory: .next
Install Command: npm install

Add Environment Variable

1. Expand Environment Variables section
2. Add variable:

NEXT_PUBLIC_API_URL: URL of Backend Deployment

3. Keep Production checked

Deploy

1. Click Deploy
 2. Wait 2-3 minutes for build to complete
 3. Click Continue to Dashboard
 4. Copy your frontend URL from the top
Example: <https://your-project.vercel.app>
-

Step 5: Connect Frontend and Backend

Add Frontend URL to Backend

1. Go back to Render dashboard
2. Click on your backend service
3. Click Environment in left sidebar

4. Click Add Environment Variable
 5. Key: FRONTEND_URL
Value: Your Vercel URL (no trailing slash)
Example: <https://your-project.vercel.app>
 6. Click Save Changes
 7. Wait for automatic redeployment (1-2 minutes)
-

Step 6: Create Admin User

Access Render Shell

1. In Render dashboard, click on your backend service
2. Click Shell tab at the top
3. Wait for shell to load (10-20 seconds)

Create Superuser

1. Type this command:

```
python manage.py createsuperuser
```

2. Fill in the prompts:
Username: admin
Email: your-email@example.com
Password: (enter a secure password)
Password (again): (re-enter password)
3. You should see: Superuser created successfully

Test Login

1. Go to your Vercel URL
 2. Navigate to login page
 3. Enter username: admin and your password
 4. Click Login
 5. You should be logged in successfully
-

Step 7: Verify Deployment

Run through this checklist:

Backend:

- Health check works: <https://YOUR-BACKEND.onrender.com/health/>
- No errors in Render Logs tab
- S3 bucket contains db.sqlite3 file

Frontend:

- Homepage loads: <https://your-project.vercel.app>
- Login works with admin credentials
- No errors in browser console (F12)

Database:

- Create test data in the application
- Refresh the page
- Data should persist

Shutdown Instructions

1. Shutdown Backend (Render)

Option A — Sleep Automatically (Free Tier default):

- No action needed. Render automatically suspends the backend after 15 minutes of inactivity.

Option B — Manual Stop:

1. Go to **Render dashboard** → **ai-scale-backend** → **Settings**
 2. Scroll down to “**Advanced**”
 3. Click “**Suspend Service**”
 4. Confirm suspension.
-

2. Shutdown Frontend (Vercel)

Option A — Temporarily Pause:

1. Go to **Vercel dashboard** → **Project** → **Settings** → **General**
2. Scroll to “**Danger Zone**”
3. Click “**Pause Deployments**” to stop new builds (site remains visible).

Option B — Full Stop:

1. Go to **Vercel dashboard** → **Project** → **Settings** → **Danger Zone**
2. Click “**Delete Project**”
3. Confirm deletion.

Project Documentation

Below we have listed more documentation from the development process of the Customisable AI Use Scales App that might be useful. This includes what exactly was developed during each sprint, as well as our meeting notes

Sprint Releases

Sprint 1 Release (01/09/2025)

In this sprint, we focused on setting up the front-end and the back-end frameworks.

Tasks Completed:

- Set up React app
- Designed pages on Figma (as shown in the frontend design model above)
- Set up Django/python + SQLite database frameworks
- Started implementing database table structure
- Started writing database queries

Sprint 2 Release (08/09/2025)

In this sprint, we finished coding the login page and template creation page, integrating APIs to the login page

Tasks Completed

- Finished creating log in page and template page
- Designed and Implemented APIs for
 - Editing use scale tables
 - Logging in; and
 - Displaying account information
- We also need to make sure that pull requests are proactively being reviewed by a second person before merging
- And code needs to be pushed to git frequently — especially front-end
- Pull requests should be merged to main by another user in the team. This way we know that they are being actively reviewed
- Whenever a change is made to a component, this should be pushed immediately to git
 - No need to pull request for all major changes (this can be done for feature implementations) but by doing this, we can see the progress that is being made
- Work on improving acceptance criteria for the Jira. This will help us understand the requirements of each ticket clearer

Sprint 3 Release (22/09/2025)

In this sprint, we completed user stories 1-5

Tasks Completed

- Created a dashboard page for all templates assigned to a user
- Created a registration page
- Designed different home pages for different users
 - Admin
 - Subject coordinators
 - Tutors
- Started testing backend
 - Wrote Unit Tests
 - Ran Unit Tests
- Integrated front-end with back-end API
- Finalised backend use scale APIs by cross checking with Figma

Sprint 4 Release (06/10/2025)

In this sprint, we implemented user stories 6 and 8 and started working on user story 11. We also began implementing versioning and performed general bug fixes on features implemented in the previous sprint

Tasks Completed

- Finalised registration page with roles
 - Removed students in registration
- Created a back button on the table creation page
- Fixed sidebar
 - Created separate pages for community/personal templates
- Fixed log out bug (make user data unload when logging out)
- Fixed edit template functionality so it does not create a duplicate
- Implemented duplicate button
- Created user sessions
- Determined how to upload default repositories as System Admin

Sprint 5 Release (13/10/2025)

In this sprint we completed user stories 11 and 9. We also performed bug fixes and fixed any major UI flaws

Tasks Completed:

- Built system admin home page
- Replaced mock data with actual data from database
- Fixed access for different roles
- Fixed various UI components including
 - Save prompt appearing only when changes are made
 - Editing buttons so they have hover feedback

- Checking inconsistent colour themes
 - Changing the top banner to show type of page and role of user
 - Implemented preview button for both home page and dashboard pages
- Fixed any outstanding VS code errors
- Fixed routing to redirect automatically to log in
- Added a delete button
- Fixed the create new AI use scale button
- Added a way to read all templates in the database into the web browser so it can be displayed in the “all templates” page
- Made search bar in front end work
- Updated use acknowledgement field in guidelines

Sprint 6 Release (20/10/2025)

In this sprint, we focused on fixing the product so we had a polished prototype to demo. We also completed all must and should user stories (that is, user stories 1-11)

Tasks Completed

- Created the foundations for deployment
- Worked on code formatting (commenting)
- Renamed “all templates” section to “community templates”
- Fixed log in bug (so cookies are passed along correctly on the first login attempt)
- When a template is duplicated, it now displays the current user’s name instead of the original creator’s name
- Implemented delete button so templates are now deleted
- Statistics surrounding the application are now shown to system admins on their home page
- A community templates widget is now shown on the home page
- “Create new use scale button” is fixed again, resulting in faster table creation times

Handover Release (05/11/2025)

Prior to handover, we were focused on fixing any last minute bugs as well as ensuring that versioning access was more intuitive to a user of the system

Tasks Completed:

- Changed versioning to a drop down menu within the template builder to allow for ease of navigation
- Finalise deployment of the product
- Fixed the “save guidelines” button on the templatebuilder page so pressing “back” + “save and continue” saved the current changes on the template
- Made registration page accessible from starting login page

Meeting and Stand Up Notes

This section includes in-depth notes about what was discussed with the client at meetings, as well as our notes from when we met together as a group, performed stand ups and what was decided upon when.

11/08/2025

- Based on an existing **template**, the client is interested in a buildable customised scale
 - This should collate old templates from the past in a repository
 - This is **assessment specific**, with different scales for different assessments
 - However, one scale might be used for many subjects
 - We'll need to make a decision on how exactly to display this
 - Eg. Programming assessments will have different AI Assessment scales and different task components
 - Efficient for stakeholders to import these templates to save time.
- The scale should explicitly outline how students should acknowledge their use of GenAI tools in their submissions
- There should be **standardised descriptions and key words (vocabulary)**, everything else should be customisable
 - Since its standardised, it should not be able to be changed
- The **subject coordinator** should be able to provide examples (e.g. prompts to ChatGPT) of what is allowed/not allowed
 - They are users of the system
 - Could potentially take additional ideas from students, for discussion together and modify to fit their ideas
- There should be guidelines in place on how students should acknowledge their use of AI too.
- There will be additional fields to fill in post submission of the assessment
 - Like a form that will be handed in alongside submission
- Should link to resources to teach students how to use AI appropriately for each defined level within the scale
- Current layout is split into individual assessment tasks with AI use for each one written out
 - Includes examples, the AI use acknowledgement and student declaration
 - The **student** needs to fill out the required cells
- The assessor should clearly specify what they want students to do (for communication purposes) so they can check the student's use of AI is appropriate
 - **No focus on detection, we're just collecting the information**
- The templates must be publishable so they can be accessed by students
 - e.g. providing an HTML link? It just needs to be viewable at this stage
 - There should be a downloadable acknowledgement form (e.g. an Excel document)
- Re-use is not top priority
- The fields and templates should evolve, making the platform **extendable**
 - If we update an original, we should be able to automatically update new templates based off of it
- No connectivity to LMS required

- A **system admin** should be allowed since it is **necessary** to add user/system management
 - They assign rights to create a subject use scale and potentially create spaces for the subject
 - There is the potential extensibility to different universities, which would result in different AI acknowledgement requirements
 - Architecture should be easily extendable to other universities
- Users should be able to choose whether to share their repositories with others, what repositories they own and those that they have been shared into
 - By default, coordinators create a template for their own subject. But, they should have a possibility to share with others if they wish to.
 - Need to allow/disallow live collaboration editing of a template if so
 - Once updated, other coordinators should be notified that there is a new version of the template, and can update their own versions to date.
 - Stakeholders should be able to work on the same template to help build scales for their assessment
 - This should be similar to a google doc where updates are live. But perhaps, there should be some way to prevent two people from editing the same thing at the same time (deadlock); Eg. Powerpoint only allows one user to access and edit a slide at a time.
- We should preserve version history because some people may not want to update
 - Keep record of time-stamp, date, etc.
 - Coordinators should have an option to either stick with the original or update to the new version.
 - There should be some sort of connecting link, rather than just a one dimensional copy and paste of words.
- This is a **web-based software** so assume laptop use for now—no installation required
- Can use any programming tools at this stage but it should be hosted on a virtual cloud platform. We can come up with the best way to design and present this to the users
- Should be able to delegate the subject coordinator role to other people (e.g. **tutors**)
- For now, it is okay to purely focus on subject coordinators (and potentially the admin since this would still potentially be important)
- Data should be tracked regarding how many people are using specific levels/templates
 - They should be able to see those using their templates
- The **client wants to see the user stories**, so make sure we share these with them in the next week!

18/08/2025

- Discussed moving forward with the design stage
 - Assigned group members roles for what they should be focussing on this week
- Discussions about what we should use for architecture. Currently considering:
 - Back-End: DynamoDB (database), OAuth2 (university SSO for authentication), Python/Django (framework)
 - Front-End: React
 - Web-Server: Nginx
 - Deployment: GitHub Pages, AWS Cloudfront (for scalability)
 - APIs: ...
- Product Owner (Winston) emailed client with the user stories and models that we developed in last week's

- Further discussions about sending another email to clarify the bare bones components that our client wants us to implement, this will be drafted and sent before our next meeting

21/08/2025

Stand Up

Since last week we have:

- Maddy: Started front-end design. Put preliminary images in Notion but offered to send to group if required
- Violet: Finished database model. Will try to start setting up the database
- Jason: In progress with system sequence diagrams. Will help with front-end design once finished

25/08/2025

Stand Up

Since last week we have:

- Maddy: Started front-end design. Only modelled the start of a log-in page and a potential table layout at this stage
- Jason: Completed system sequence diagrams. Started helping with front-end design and fleshed out log in and registration screen ideas
- Winston: Completed physical and system architecture diagrams
- Violet: Completed database diagram. Also started setting up database using DynamoDB
- Bonnie: Completed domain diagram

Notes from Client Discussion

- Presented Moscow table to client to get clarification
 - This is meeting their requirements for now
 - User stories are all also okay
- Client specified that admin users create the default use levels in each template
 - Admin can create use scale templates too, especially the default ones
 - This is related to the vocabulary
 - So the subject coordinator can modify the default templates created by the admin to create their own version
- Asked about whether we need to create a Google Form for students to acknowledge their AI use with
 - Just an excel file is fine for now — like with other assessments
- Clarified level of authentication they are after with the system
 - This is a high priority to them
 - They are assuming the software is hosted within the university network so outsiders shouldn't be able to access it anyway
 - This provides a built-in protection through the university
- Ran our current plans surrounding web development by them (AWS)
 - Since this is university data, they would prefer the system to be **self-hosted** (so it is low-risk). We can scale the program up in the future
 - This means we should make the IP within the network so that only people within the university can use it, making it easy and safe

- Attempted to show the system sequence diagrams, though the clients trust us with design and aren't wishing to see it. We can clarify requirements with them though
- **The final product will be hosted on a Linux cloud research server within UniMelb**
- They have specified the most difficult part of this project will be handling the differentiation of use scales (that is, keeping track of all the versions in the system)
 - Each use scale should be linked to a subject

Other Meeting Items

- Also discussed how we want to format code so it is consistent between front and back-ends. This is outlined in the coding standard

01/09/2025

Stand Up

Since last week, we have:

- Jason: Has created a table and a section in the front-end. Will push to Git tonight.
- Winston: Half a log in page that currently isn't connected to anything. Can't use third parties though.
- Violet and Bonnie: Have created all tables, written sample SQL queries, and started routing

Some issues we encountered include:

- Some uncertainty with authentication in the log in page. Product owner (Winston) will follow this up with the client
- Front-end people are confused with how to integrate back-end with front-end, so we need to figure out how exactly this works

Other Meeting Notes

- Discussions about who will be performing testing
 - Bonnie as the testing lead is doing testing. Maddy has agreed to help
 - Decided that if we start pushing to main, we should start testing
- Should have a meeting with the client next week to provide a progress update
 - We need to be proactive in scheduling this!

08/09/2025

Stand Up:

Since last week, we have:

- Violet: Set up dummy entries in tables, written essential APIs, wrote database related documentation, confirmed login/registration APIs, and integrated the login API with frontend
- Bonnie: Added more dummy data to the tables (at least 5 for each table) and created use scale table and started working on API for interacting with editing AI use scale templates
- Jason: Worked more on the table creation page

- Winston: Finished log in page

Notes from Client Discussion

- **Login:** Good - keep as is.
- **Within scope (for now):** create an admin-only tool to build and manage templates and scales.
- **Out of scope:** subject enrolments, subject views, and student data. Admins create and manage everything.
- **UI requirements:** every template/scale must clearly show subject code and name (e.g. COMP10002 - Foundations of Computing), year and term/semester the scale applies to. The UI should also make it obvious which scales belong to which subject.
- **Admin experience:** Backend must be intuitive and straight-forward. Provide an admin dashboard with a list of templates/scales, search and filters, create/edit/delete, duplicate, and assign to subject/year/term. There should also be a default template, allowing creating a baseline template that admins can use.
- **Versioning:** Track versions and allow for observing changed history
- Focus on core functionality first

Other Meeting Notes

- Violet merged Log In page to the main branch so that back-end could be read in
- Sprint time frame was discussed to be okay for now, but we have increased the time frame to 2 weeks for the next one because we have more items that we would like to get done

15/09/2025

Stand Up

Since last week, we have:

- Maddy: Coded up dashboard pages
 - Suggestion from team to make tables match between front-end pages (the dashboard vs the AI use scale table creation)
- Violet: Finished up all the api's for dashboard page, editing & creating templates. Ready to integrate with frontend (dashboard, basic template editing) this upcoming week.
 - Cleaned up and finalised database schemas, also wrote up API documentation
- Bonnie: Created unit tests to ensure functionality of the tables and login API
 - Created unit test for user table
 - Create 4 unit tests to check the login API (setting up user, testing login with correct credentials, testing login with incorrect credentials, testing login with missing fields)
- Jason: Created the template page to view and edit an AI use scale template. Is aiming to work on different home pages for different levels of users (admin, coordinator, staff) this upcoming week.
 - Fixed and addressed the errors within template builder page,
 - Added a section for subject coordinators to provide resources on the template
 - About to start on the home page coding
- Winston: Created registration page

Other Meeting Notes:

- Would be good to have the UniMelb logo on all front-end pages
- Cross-referencing old client meeting notes, we need to have semester and year of study in the table
 - This needs to be updated in both the dashboard page (and also reflected in the back-end), as well as potentially in the template page as well
- Discussions about a duplication API — need it to display all templates owned by user **and** also the admin-created base templates
- We need to do some basic routing. All the pages are good individually but they need to be connected together
- Also need to perform testing with Django Flow
- Need the front-end design to be clearer to what the actual product is currently looking like
- We should try and get feedback for what our current front-end design looks like since we've made substantial progress since we last saw the client
 - This should be done through Ed Discussion

22/09/2025

Stand Up:

Since last week, we have:

- Maddy: Fixed formatting for dashboard, linked it back to login page, reformatted stand ups and the retrospectives, brought back the team meeting page, started working on next progress report
- Bonnie: Wrote up sprint cadence and ceremonies, created the testing plan, finished up unit testing and integration testing. This involved testing of the:
 - Authorisation API
 - Template API, and the
 - Views and Models API
- Jason: Finished front-end diagrams, fixed figma template, started base template of home page for users
- Winston: Wrote deployment and decision making documentation, created registration page. Emailed images of current front-end design to client

Some issues we encountered included:

- Search bar removed from dashboard page?
 - Okay to add back (was removed by accident)
- Confusion over how deployment process works

Notes from Client Discussion

- Team performed a working demo of how the web application looks so far
 - This involved logging in, navigating the dashboard, duplicating templates (editing, etc.)
 - Client is happy with how the app runs and looks at the current moment
- Asked about potential things they might want us to include going into our next sprint

- Mentioned it'd be good to have statistics that track the AI Use Scales (specifically the levels)
- Would also potentially like a more detailed versioning history than what is currently in place (so you can easily see all the different versions)

So some changes we should implement include:

- Admin statistics (user info, template info) on Admin dashboard
- Drop down view for navigating different versions of the template

Productive Outcomes

- Got additional feedback from client, and delegated tasks to deal with the feedback items

29/09/2025

Stand Up

Since last week we:

- Maddy: Separated dashboard page into a personal templates page and a community templates page. Still need to work on filtering table so it only shows templates made by the current user
- Bonnie: Created a new API endpoint for duplicating templates in the dashboard. So now the duplication button is implemented.
- Violet: Implemented user session capabilities, including session cookies so now users are automatically re-routed to login if trying to visit other pages without being logged in. Also added CI for Django and node.js on GitHub Workflow, fixing bugs associated with this
- Jason: Implemented back button for the template builder page which routes users back to the previous page they were on. Started design of the home page and updated the front-end design model to reflect this
- Winston: Figured out how to output data to an Excel file. Working on code for this, will push when done since it needs to be formatted

Some issues we encountered included:

- Template editing broken - can change details but button doesn't revert back to save
 - Was fixed on another branch so pushing to main solved this problem
- Current known bug with versioning where editing an older version results in newer versions being overridden
 - Violet will work on correcting the API for this in the next week

06/10/2025

Stand Up

Since last week we:

- Maddy: Created new branch and swapped the tables on each page since it already filters for templates created by the current user

- Bonnie: Put base template in dummy data. Planned out structure of presentation
- Jason: Mainly focussed on the home page (since there's 2 different versions). This is done for staff (both tutors and subject coordinators) at this stage.
- Winston: Finished registration page, made ai use scale table convertible into a pdf
- Violet: Debugged issues related to versioning and saving new templates. Implemented Create New AI Use Scale button and integrated registration API

Some issues we encountered included:

- How to read all templates that are in the database?
- Some of the interactive UI components (e.g. buttons) do not function as anticipated
 - Should work on fixing this in the next sprint
- None of the other team members can get the create new AI use scale button to work?

Other Meeting Items:

- Have some questions we need to ask the client regarding the UI. It was recommended to do this sooner rather than later since we don't have a lot of time left so product owner (Winston) has contacted them via Ed to ask

Productive Outcomes

- Discussed what features need to have additional bug fixes for next week
- Delegated bug fixing tasks to team members
- Confirmed that student login view is not necessary

13/10/2025

Stand-Up

Since last week, we:

- Maddy: Added the delete button, have not implemented connections to the back end yet. Did not implement user functionality, want to clarify requirements of students + system with client
- Winston: Exported to excel so now you can export as either a pdf or an excel. This download now happens automatically
- Violet: home page, all templates has connections to back end now
- Bonnie: powerpoint, search bar now works
- Jason: still working on home page for the system admin but this is mostly done, fixed the coding errors

Some issues we encountered included:

- After logging in, it automatically redirects back to log in, even with correct details
 - Maybe some cookies aren't being passed along properly?
 - Back end is aware of this and will aim to fix in the next sprint
- Adding an extra button to the personal templates page gives a next.js error
 - This appears to only be happening on Maddy's system, so is not an issue
- Formatting in Excel is weird
 - A non-issue, just need it in the format

- Lots of miscommunication surrounding tasks this week
 - Need to work on acceptance criteria so this remains clear

Other Meeting Items

- Discussed about making contact with client this week to confirm if student view is okay as they don't expect students to be interacting directly with the system
 - Product owner Winston to email them about this early this week
 - Also since we did not meet with them today, we should send them a video of our current prototype so they can give feedback before end of wk 12

16/10/2025

- Discussed about sending a demo to the client before we meet with them tomorrow
 - Filmed going through different functionalities of the current system (logging in, editing templates + versioning, duplicating templates, deleting templates, home page, etc.)
 - Winston emailed to client

Productive Outcomes

- Found areas for bug fixing/polishing
- Sent off screen recording of current product to clients for feedback

17/10/2025

Notes from Client Discussion

- Did a walk through of the system with both Winn and Stella present since the demo video we sent them yesterday had no audio
- They seem pretty happy with it but would be interested in the following changes
 - Consolidate versioning
 - Get rid of the edit button in community templates (users shouldn't be able to edit other people's templates)
 - work on formatting excel
 - finalise difference between template and publishable
 - maybe rename them?
- Will organise handover with them at a later date
- Multiple people editing the same template
 - Though this was already identified as a "could"

So some changes we should implement include:

- Get rid of the isPublishable status of each template and remove them from all display
- Remove the 'edit' button in community templates