



## IT6037 Project Review Report

Student (Ākonga) Name:	Gray Hawes
Student ID:	20220228
Team Name:	Data Access Geeks
Date Submitted:	18/09/23

## Project Overview

Provide a *brief* summary of the project, its objectives and its success in meeting those objectives.

Our team had the goal of developing a student-focused database containing articles spanning three main subjects: Arts, Mathematics, and Technology. Following the creation of this database, we designed and developed a web application that allows users to access and engage with these articles.

## Project Objectives and Metrics

Review the key project metrics and identify any deviation from the baseline plans.

Metric	Planned value	Actual	Reason for variation
Time	Iteration 1: Documentation (21/08/23 – 27/08/23)	Iteration 1: Documentation (21/08/23 – 27/08/23)	No variation
	Iteration 2: Developing the Database (28/08/23 – 03/09/23)	Iteration 2: Developing the Database (28/08/23 – 03/09/23)	
	Iteration 3: Developing Database Queries (04/09/23 – 10/09/23)	Iteration 3: Developing Database Queries (04/09/23 – 10/09/23)	
	Iteration 4: Implementing Database Using Web Application (11/09/23 – 15/09/23)	Iteration 4: Implementing Database Using Web Application (11/09/23 – 15/09/23)	

<b>Agile software development</b>	<p>We agreed to use a Kanban Board to keep track of tasks and categorise them into which needed to be completed, which were currently being completed and which had already been completed. We implemented this on Microsoft Teams under the 'Tasks' tab and also in our team's GitHub organisation, using the Issues tab/function.</p> <p>Our team also used the Scrum methodology, complementary to using Agile. Every week after the first week of the project, we had two Scrum Masters. One for backend development and one for frontend development.</p>	<p>We used the Microsoft Teams Kanban Board and the Issues tab on GitHub.</p> <p>We also stayed consistent to our roles documented in the Planning Document and had two Scrum Masters most weeks for the project's duration.</p>	No variation.
<b>Risks</b>	<p>It wasn't officially identified as a risk during the project but the unavailability of team members posed a risk to the project's completion, or at least the quality of. I myself probably had the most unavailability due to me being away for 4/5 days and then falling ill immediately after my return.</p>	<p>Not a risk to the project, but a personal risk to myself. I didn't get to contribute to the programming side of development as much as I had planned to and wanted to.</p>	<p>I was away for a few days for leisure reasons and then I got sick right after coming back.</p>
<b>Other</b>			

## Project Delivery

Review the key aspects of project delivery in relation to the development process and the outputs delivered during the various iterations. Identify key learnings for each item, where applicable.

### Team Organisation

Refer back to the Team Contract you created at the start of the project when you identified the team's shared values and behaviours, communication and collaboration methods and issue resolution approach. Consider your collaboration tools and stand-up meetings too.

Review Item	Met/Unmet	Comments/Learning
<i>List a specific deliverable or process component and identify what you were trying to achieve with it, e.g. principles/concepts behind it.</i>	<i>Assess the effectiveness of the deliverable/process. If they were not fully effective, then explain why and whether there was an impact.</i>	<i>Identify key learnings here.</i>
Stand-up Meetings	Stand-up meetings were completed by the team members inconsistently. However, this could be due to being unfamiliar with the format of the stand-up meetings, because the members most inconsistent with updating were Pyper, Ciar and Jethro. And all three of them were in a different team from the rest of us for the last team project, where our team used this format.	Maybe we could have reminded the team more to complete the stand-up meetings.
Group Discussion Board (Microsoft Teams)	The team used the group discussion board on Microsoft Teams for posting updates.	

Email	I didn't have anyone email me but that doesn't mean that other team members emailed each other.	
GitHub	We used GitHub for developing the web application collaboratively.	
Microsoft Teams Direct Messages	I think the direct messages on Microsoft Teams was used the most between team members. I had a few conversations with different team members using them.	
Sprint Planning Meeting	We would hold a planning meeting each sprint at the start of the week and we would work around people's availability so everyone or most of the team could join.	
Sprint Review Session	We would hold a meeting at the end of every week to go over what we accomplished during the sprint and what we will probably work on for the next sprint, to be confirmed in the meeting for that week.	

## Requirements Analysis & Planning

Refer back to the client's requirements, the Planning document and your Kanban board. Think about how you investigated your solution and what software development practices you applied during this stage.

Review Item	Met/Unmet	Comments/Learning
<i>List a specific deliverable or process component and identify what you were trying to achieve with it, e.g. principles/concepts behind it.</i>	<i>Assess the effectiveness of the deliverable/process. If they were not fully effective, then explain why and whether there was an impact.</i>	<i>Identify key learnings here.</i>
Database Type: selecting an appropriate database to use for developing the web application	Met. A non-relational database was thought to be most effective for the type of data that the web application deals with.	
Type of application (web application or mobile application)	Met. A web application allows for a wider audience rather than a mobile-only application.	
Ability to search by key word.	Met.	
Ability to filter by category.	Met.	

## Solution Design

Refer back to the System Design document and the Planning document. Think about how you developed your database, including database queries, how you managed digital assets within your database, how you applied HCI and UX principles to the development of your user interface, how you managed security requirements in regard to user authentication and authorisation, and what software development practices you applied during this phase.

Review Item	Met/Unmet	Comments/Learning
<i>List a specific deliverable or process component and identify what you were trying to achieve with it, e.g. principles/concepts behind it.</i>	<i>Assess the effectiveness of the deliverable/process. If they were not fully effective, then explain why and whether there was an impact.</i>	<i>Identify key learnings here.</i>
Database Queries: covering SCRUM functions.	Database Queries were written for each SCRUM function needed for the web application.	
HCI and UX principles fulfilled	Mostly met. Colours are used to show different areas of content and to indicate functionality (such as buttons and interactivity).	Personally, I think the colour scheme of the web application could be improved. But this was decided upon via a group decision.
Authorisation of users	Not met. This was due to there not being enough time for us, mainly Jethro, to be familiar with implementing Auth0.	

## Implementation & Testing

Refer back to the final database application code in GitHub, the Testing document and the Planning document. Think about how you implemented and tested your solution and what software development practices you applied during this phase.

Review Item	Met/Unmet	Comments/Learning
<i>List a specific deliverable or process component and identify what you were trying to achieve with it, e.g. principles/concepts behind it.</i>	<i>Assess the effectiveness of the deliverable/process. If they were not fully effective, then explain why and whether there was an impact.</i>	<i>Identify key learnings here.</i>
SCRUM testing	Each SCRUM functionality was tested in the forms of carrying out all the relevant user stories. This was done by following a standard test plan that was then catered to the type of functionality being tested.	
Auth0 Testing	Auth0 was tested during development by Jethro as he was working on it, but it was also tested by Ciar at key points.	

## Collaboration Using Version Control

Refer back to how you used GitHub as a version control tool.

Review Item	Met/Unmet	Comments/Learning
-------------	-----------	-------------------



<i>Identify what governance system you had in place to apply version control.</i>	<i>Assess the effectiveness of version control in GitHub. If not fully effective, then explain why and whether there was an impact.</i>	<i>Identify key learnings here.</i>
Use of the Issues tab in GitHub. Using Issues allowed everyone to see what everyone was working on so that there wouldn't be overlaps, but to also avoid clashes of code if possible.	Met. The Issues were used.	
GitHub managers. Two team members were set to be our official GitHub managers. This means that they would review and hopefully approve pull requests and check progress.	Met. Ciar also became a GitHub manager.	

## Lessons Learnt

### What worked well

Provide a list of any aspects of the project that went better than expected or worked well and what could potentially be improved next time.

<b>Description</b>	<b>Recommendation</b>
Identify what deliverable/area worked well.	Capture any lesson(s) learnt for future projects.
Communication.	
Completing work.	
Wireframes.	Maybe a different colour scheme.

## What didn't work as expected

Provide a list of any aspects of the project that didn't go as well as expected, including issues and how they were resolved, and a recommendation for what could be improved next time.

Description	Recommendation
Identify what deliverable/area didn't work well.	Capture any lesson(s) learnt for future projects.
Authorisation. Auth0 wasn't fully implemented.	Choose a different method of authorisation that is either more familiar, or seems easier to comprehend if not. Time it well.

## Team performance

Review and identify what worked well when it came to the overall performance of the project team and highlight areas for improvement. Provide a list of aspects of the project that did and didn't go as well as expected.

Description	Recommendation
Identify what area of team performance didn't work well.	Capture any lesson(s) learnt for future projects.
Stand-ups. Not all team members participated in the stand-up meetings consistently.	More reminders. Give an example of how the stand-ups are going to be conducted so everyone is familiar before beginning, which might be more encouraging for them to participate.

## Related Documents

- Planning Document
- Requirements Document
- Testing Document
- Scrum Master Documents