Assignment Cover Page

Programme	Course Code and Title
Bachelor of Computer Science (Hons)	CSE3033/N Software Engineering
Student’s name / student’s id	Lecturer’s name
0204677 Lim Zhe Yuan	Tan Phit Huan

Date issued	Submission Deadline	Indicative Weighting
Week 2 – 6 Feb 2023	Week 6 – 10 March 2023	30%
Assignment title	Assignment 1	

This assessment assesses the following course learning outcomes

# as in Course Guide	UOWM KDU Penang University College Learning Outcome
CLO2	Analyse the empirical nature of software engineering and the application of empirical methods in software engineering development
CLO3	Evaluate advanced software engineering techniques and processes in the development of a software artefact.

# as in Course Guide	University of Lincoln Learning Outcome
LO2	Analyse the empirical nature of software engineering and the application of empirical methods in software engineering development
LO4	Critique current software engineering processes in safety critical system

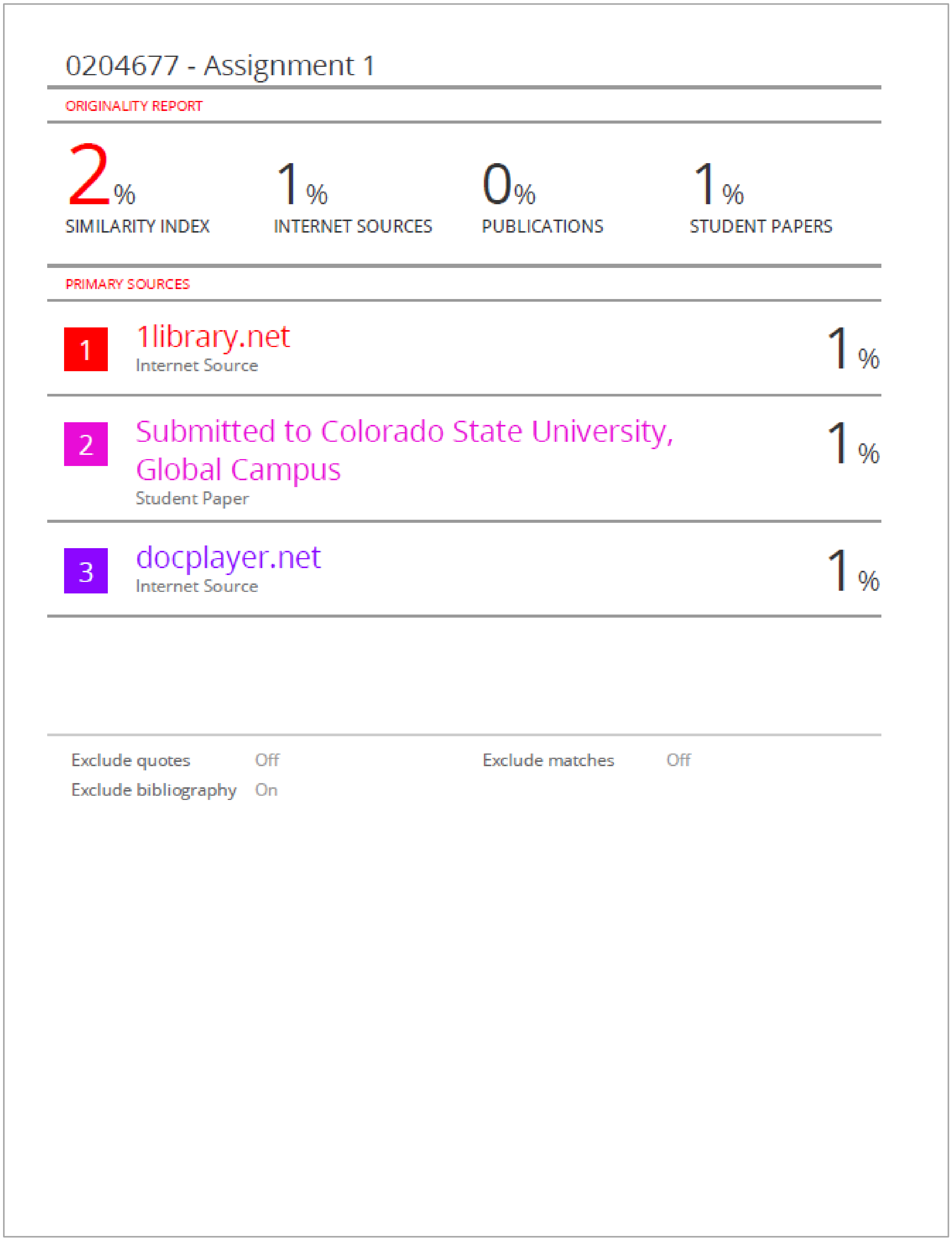
Student’s declaration

I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

Student’s signature: Zhe Yuan

Date: **10th March 2023**

TurnItIn Similarity Report



0204677 - Assignment 1

ORIGINALITY REPORT

2%	1%	0%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	1library.net Internet Source	1%
2	Submitted to Colorado State University, Global Campus Student Paper	1%
3	docplayer.net Internet Source	1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

Table of Contents

Main Report

- **Tealive**

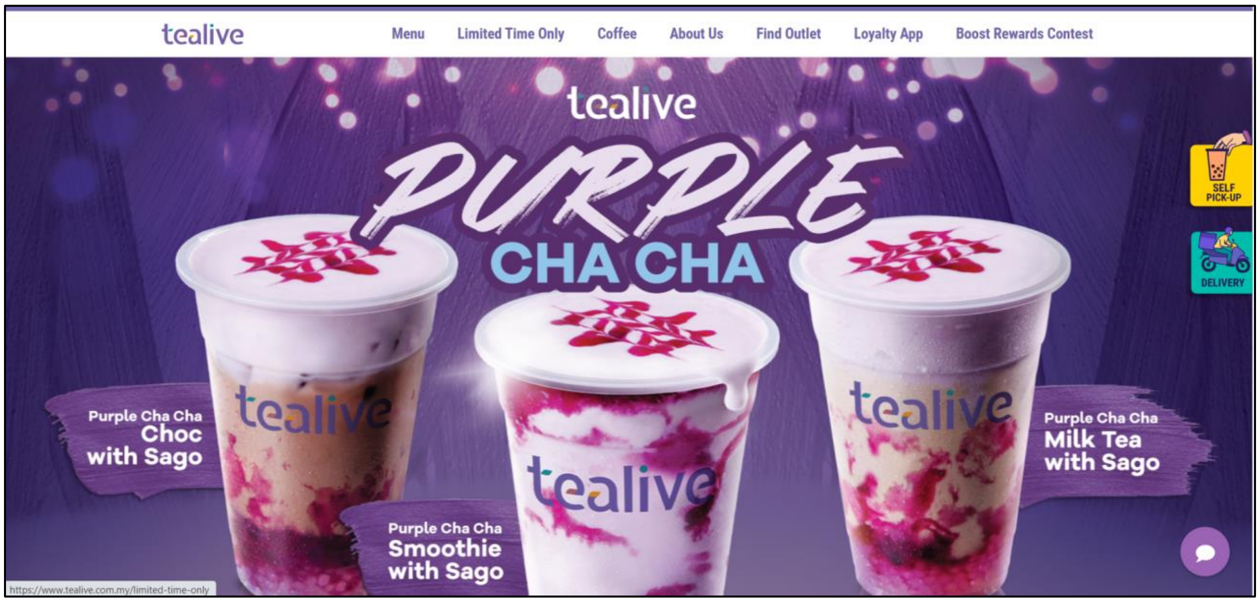

tealive
Menu
Limited Time Only
Coffee
About Us
Find Outlet
Loyalty App
Boost Rewards Contest
tealive
PURPLE
CHA CHA
Purple Cha Cha Choc with Sago
Purple Cha Cha Smoothie with Sago
Purple Cha Cha Milk Tea with Sago
SELF PICK-UP
DELIVERY
https://www.tealive.com.my/limited-time-only

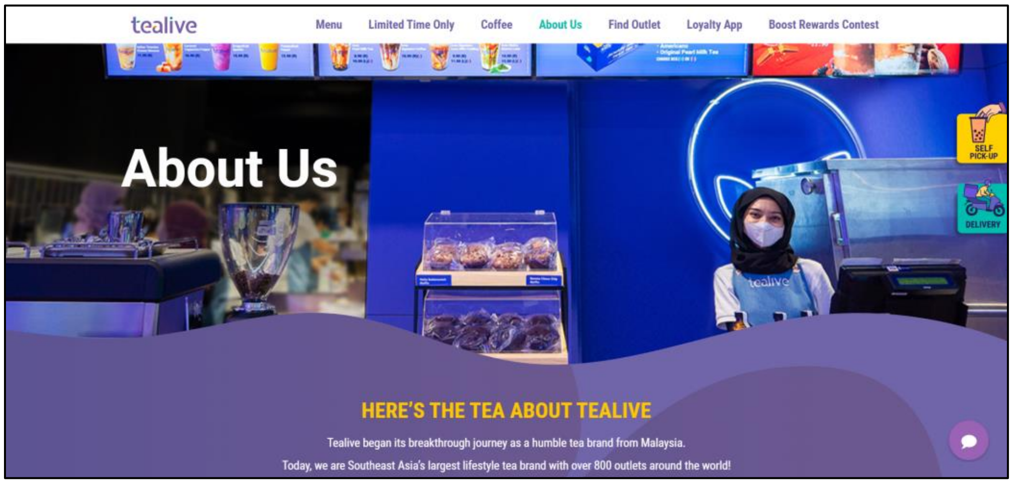
Figure 1: *Tealive official website*

1. Software Development Methodology

One software development methodology that the development team may have used to develop Tealive is the **waterfall development methodology**. According to Coltuneac (2021), waterfall methodology is suitable if the requirements and scope of the project are clearly defined and stays the same as it progresses. It is also used if the project is predictable from the beginning and clients know exactly what they want in their final product. Tealive fits this description as their application's primary features are stable and does not require the development team to constantly introduce more requirements into the website based on customer feedback. The current state of the website is capable enough to fulfill most of their customer's tasks as it is. More importantly, the development team would be able to ensure that the main functionalities of the website will be able to operate within expectations without any foreseeable failures by planning and designing them out before any actual development.

Considering the adoption of the waterfall approach by the development team, the drawbacks of the methodology include the lack of requirement clarity and visibility. Clients usually have a difficult time detailing their product and their lack of understanding of their intended product. This may halt development progress for a period of time or result in extensive changes after production and inaccuracy overall of the deliverables. Making changes to the product is also a costly and risky endeavour as the development team has to consider all previous waterfall stages before making new decisions (Coltuneac, 2021).

2. Design Review


tealive
Menu
Limited Time Only
Coffee
About Us
Find Outlet
Loyalty App
Boost Rewards Contest
About Us
SELF PICK-UP
DELIVERY
HERE'S THE TEA ABOUT TEALIVE
Tealive began its breakthrough journey as a humble tea brand from Malaysia.
Today, we are Southeast Asia's largest lifestyle tea brand with over 800 outlets around the world!

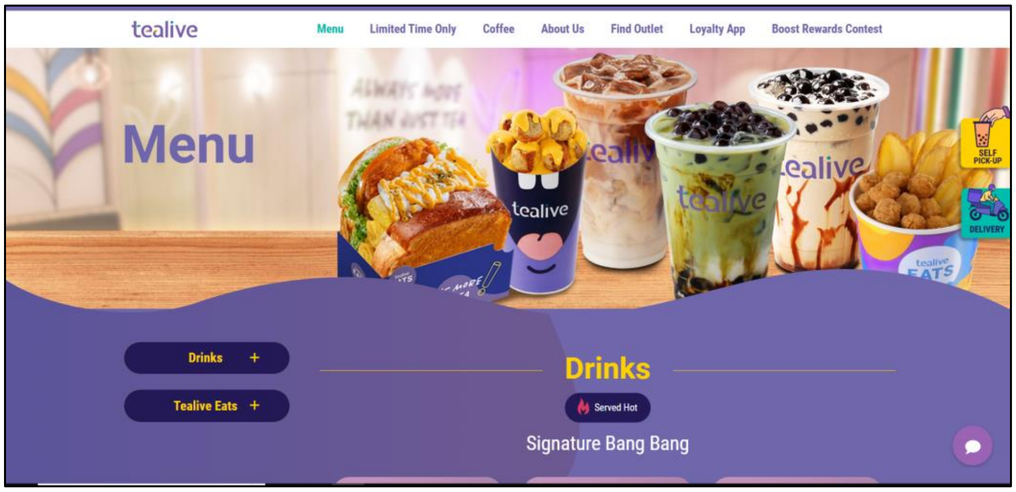


Figure 2: *Tealive consistency example – Similar page structures and styles*

Tealive applications maintains **consistency** with their graphical elements. Themes for font families and colors have been made for the application and is used sparingly throughout the entire application to make the applications visually appealing. Consistent spacing are also used to increase breathable spaces within the applications, making it feel less congested. A sense of consistency will also fulfill user expectations of the application and make them feel less skeptical about having unpredictable designs within the application, increasing their trust and willingness to use the application (UX World, 2021).

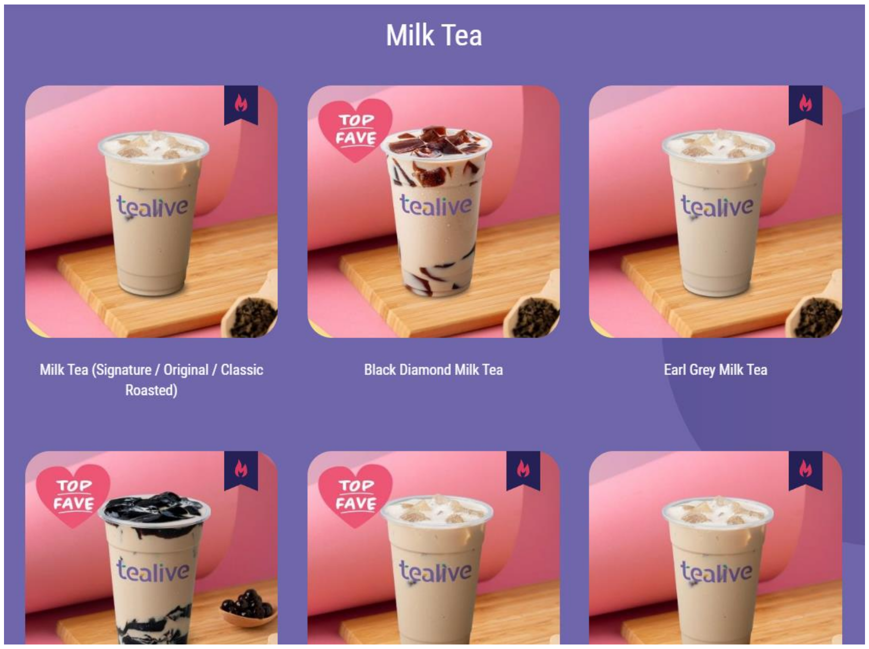


Figure 3: *Tealive predictability example – Predictable menu items*

As mentioned earlier, Tealive also incorporated the principle of **predictability** into its design which works closely with consistency. Users would be able to anticipate the display style of the menu items without considering other changes to their layouts. This reduces the user's mental load when they are attempting to process Tealive's menu visually and reduces chances of user being confused when they are using the menu. In a broader sense, users do not need to allocate a portion of their memory for remembering the interfaces of the application because they will always display similarly to one another.

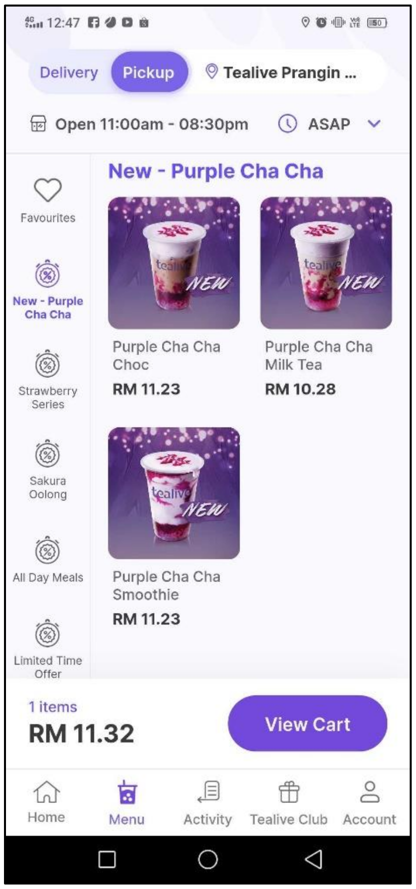


Figure 4: *Tealive efficiency example – Accessible shopping cart in the menu*

Last but not least, Tealive's mobile application UI design also considers the **efficiency** of the application to the users. The shopping cart is always accessible when users are browsing the menu. It allows the user to freely navigate between the two pages to check the orders that they have added without the need to hop in-between any unnecessary pages. Contact methods for each outlet are also provided with the outlet locator to allow users to get in contact with them easily using Waze or WhatsApp. These minor details reduce the time and effort that users use to complete their tasks, allowing them to focus on other tasks that matters.

3. Software Ethics

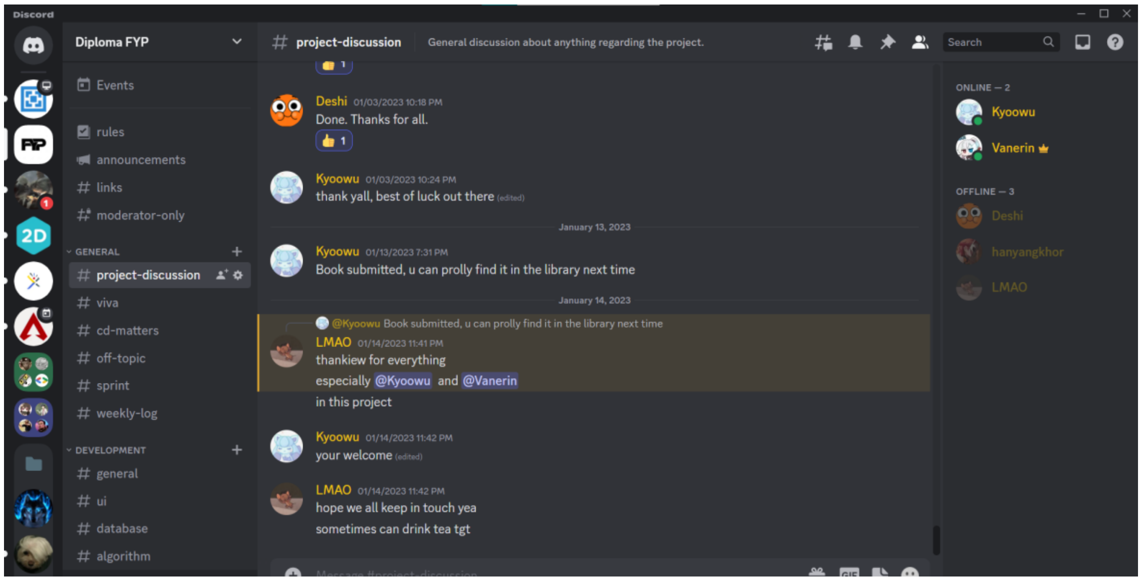
According to the principles listed in the Code of Ethics developed by the IEEE-CS/ACM joint task force (1999), software engineers should be ethical in their profession in terms of the public, client-employer relationship, product, judgement, management, profession, colleagues, and their selves. While most of these principles apply to various other methodologies, the principle that resonates the most with the waterfall development methodology can be the Product principle. The Product principle advocates that software engineers should ensure products and related modifications are developed with professionalism and meet the highest level of standards possible. In accordance with this requirement, developers should also perform cost analysis, conduct adequate system debugging, and ensure that all forms of system development activities do not violate any legal and ethical rules (IEEE-CS/ACM Joint Task Force, 1999). As waterfall development allow development teams to put in their utmost effort in each phase of a project, there are no excuses for development teams to reduce and compromise product quality, and instead strive for developing a complete product that is exactly as described by the client. It increases the reliance of clients towards the development team because clients would begin to understand that the development team is capable enough to deliver requirements accurately in time as they were instructed to do.

4. Software Sustainability & Improvement

Application-wise, Tealive successfully developed all main features that are stable and fulfils user expectations, easing the process of performing their tasks. However, the assumption of the use of waterfall methodology does not make Tealive sustainable for the long run as new requirements would be discovered along with changing society trends. Therefore, it is important to choose a methodology that would allow the developed software to adapt to changing requirements without breaking the finalized product and needing the development team to allocate additional resources excessively which may be limited.

A logical suggestion that could replace the waterfall methodology used by the Tealive development team can be the throwaway prototyping methodology. Unlike the waterfall methodology, software that were developed using prototyping models can be flexible and undergo multiple versioning to enable seamless integration of new features with the software on production. Due to the use of throwaway prototypes, new concepts can be researched on with minimal use of resources, but it still depends on the degree of redundancy that development teams want to achieve. Users would still be able to use an older version of the product too while a new system version is being developed, reducing application downtimes and increasing transparency of the work done behind the scenes (Volchko, 2017; McQuillan, 2022).

- **Discord**



project-discussion General discussion about anything regarding the project.

Deshi 01/03/2023 10:18 PM
Done. Thanks for all.

Kyoowu 01/03/2023 10:24 PM
thank yall, best of luck out there (edited)

January 13, 2023

Kyoowu 01/13/2023 7:31 PM
Book submitted, u can prolly find it in the library next time

January 14, 2023

@Kyoowu Book submitted, u can prolly find it in the library next time
LMAO 01/14/2023 11:41 PM
thankiew for everything
especially @Kyoowu and @Vanerin
in this project

Kyoowu 01/14/2023 11:42 PM
your welcome (edited)

LMAO 01/14/2023 11:42 PM
hope we all keep in touch yea
sometimes can drink tea tgt

Figure 5: *Discord desktop application*

1. Software Development Methodology

One software development methodology that the development team may have used to develop Discord is the **Scrum development methodology**. According to Nelly (2020) in a Discord blog post, Discord's development team always listens to user feedbacks to improve their application's user experience and specifically developed user features such as the "Activity Feed", "Library", and "Channel Following" features in multiple version releases because users demanded for it. It relates to Thompson's **(n.d.)**, description of Scrum development methodology where the approach requires short, incremental product changes and a continuous, cyclic process. It also relates to the fact mentioned by Thompson **(n.d.)** that Scrum development adapts to constantly changing requirements and is highly responsive to customer requests, so actual development steps and estimations are not always initially accurate. Therefore, it is important for Discord to adopt this methodology because an online chat application requires many development iterations to revise and squash bugs that will compromise communication quality.

By adopting Scrum methodology, the development team may have faced issues such as scope creeps, implementation failures, and the inability to measure the project's exact progress at a given point in time because of frequent product distributions and a never-ending development process. These issues cause developers to have limited time to work on existing issues that have been identified due to the need to handle new user requirements (Upstack, 2021). Developers may also have difficulties reworking on existing features as no proper documentation has been made for the product.

2. Design Review

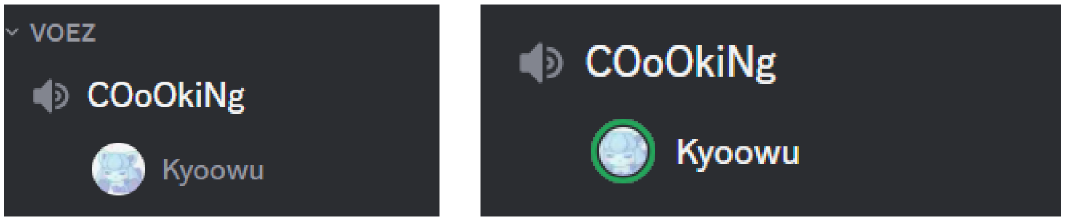
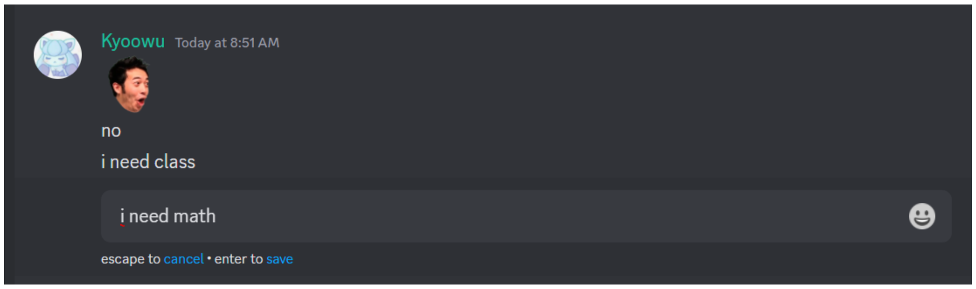


Figure 6: *Discord responsiveness example – Immediate audio visualizer*

Among the UI design principles that Discord have adopted, Discord achieved high **responsiveness** for most stateful functionalities that exist within the application. For instance, when user is speaking into their microphones in a voice channel during a call, their profile icons in the channel will light up in accordance with their speech. Text messages would also become transparent while they are being sent to show that the message have not been broadcasted to other users. These changes serve as acknowledgement for the user about the ongoing transmission of communication data and enables them to detect communication failures based on visual feedback.



Kyoowu Today at 8:51 AM

no

i need class

i need math

escape to cancel • enter to save

Figure 7: *Discord recovery example – Editable message*

Discord also puts a lot of effort in **error recovery**. It allows users to correct mistakes that they have done seamlessly without affecting any ongoing process within the application. For instance, users can still edit their messages even if they have already been sent out or delete an unintentional message. When a message could not be sent successfully. This ensures that user messages are delivered correctly and avoid any miscommunication between different individuals. Whenever users make changes to application settings, Discord also prevent users from any accidental navigation away from the settings page and provides user the option to revert the changes back to their original values in case of an unintended user change.

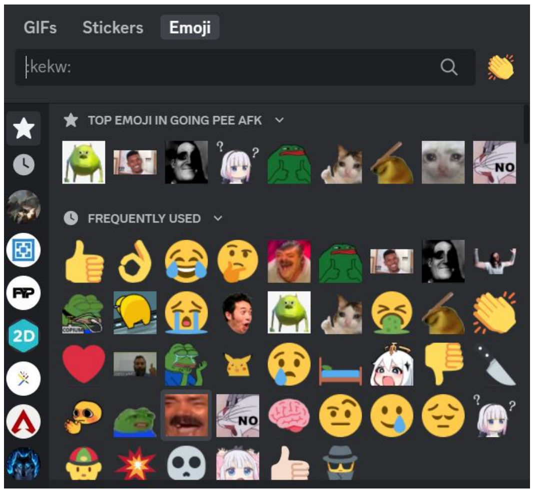


Figure 8: *Discord user memory reduction example – Ranked emojis*

In terms of a golden rule of design, Discord also helped **reduce user memory loads** when using the application. A search box is provided for users to search for keywords that they want to find. This feature is helpful for the user as it enables the user to search for messages or mentions that were sent long ago with ease. Discord's emoji selector is also designed to help users find their desired emojis easily. Frequently used or favourited emojis are displayed at a more visible area to make them accessible to the user. It removes the need for users to remember how to search for the emojis that they want.

3. Software Ethics

According to West (2016), creators of Scrum have introduced and added 5 Scrum values in their guide to a successful Scrum team – Courage, Focus, Commitment, Respect, and Openness. Based on Verheyen's (2013) description of these 5 values, Scrum members are urged to dedicate themselves in their actions and efforts, but not too much in results because of Scrum's incremental development nature. When working on a project, members must focus on important tasks and be courageous in handling uncertainties or rework. They must also be transparent about work progress to enable better collaboration and respect diversity when communicating among stakeholders (Verheyen, 2013; West, 2016). These values act as a rule of thumb for development teams to determine their preference of development behaviour. It builds trust between Scrum members as the knowledge of Scrum's key values governs them in making decisions that do not threaten another parties' best interests. In terms of development, development teams would also be less stressful about having to meet expectations of delivering fully equipped features and can break features apart into smaller, manageable chunks to develop sequentially instead. By making development feasible for an average developer, more effort can be used in ensuring code integrity and prolong its usefulness for a feature.

4. Software Sustainability & Improvement

As Discord is built upon various user feedbacks, it is expected that Discord will be able to sustain for a long time as it constantly evolves around current user needs. Though, some users on free accounts might beg to differ because Discord locks some of the most basic features behind Nitro, a premium account subscription plan. According to Baxter's (2022) article, some basic features that were locked behind Nitro Basic include animated emojis and increased file upload limits. Although the Basic plan is offered at an affordable price, basic features as mentioned earlier should not be locked behind a paywall as the application should ease the process of communication as much as possible. Users may feel the need to switch to another application to improve their communication quality with the removal of basic features, making Discord a less viable option as a communication platform.

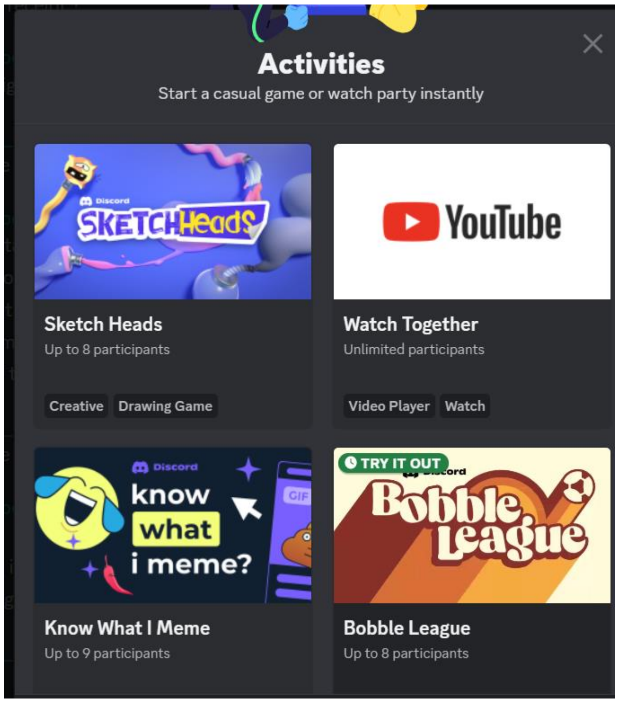


Figure 9: *Games and Youtube available to use on Discord if user subscribed to Nitro plan*

A possible solution to this issue is to simply make the features mentioned above accessible to the average user. There is no harm to make a few communication features free to use as it increases the user's dependence on the application to perform their daily tasks. In retortion to Discord's methods of securing profit, Discord already monetized other system features such as user flairs, profile customizations, and increased video or voice quality in voice channels. Discord also collaborates with various vendors and game creators, allowing them to place their services in the application. Other methods notwithstanding, these methods are sufficient to generate income that allow Discord to continue operating normally.

References

Baxter, D. (2022) *Discord is rolling out 'Nitro Basic' – a cheaper plan with four useful features*. TechRadar. Available at https://www.techradar.com/news/discord-is-rolling-out-nitro-basic-today-a-cheaper-plan-with-some-great-features [Accessed: 3 March 2023].

Coltuneac, A. (2021) *The Waterfall Method: Definition, Pros & Cons and When to Use It*. Allegra Blog. Available at https://www.trackplus.com/blog/en/the-waterfall-method-definition-pros-cons-and-when-to-use-it/#When_should_you_use_the_waterfall_methodology [Accessed: 24 February 2023].

IEEE-CS/ACM Joint Task Force (1999) *Code of Ethics*. IEEE Computer Society. Available at https://www.computer.org/education/code-of-ethics [Accessed: 3 March 2023].

McGreal, D. (2016) *A Code of Ethics for Software*. Scrum.org. Available at https://www.scrum.org/resources/blog/code-ethics-software [Accessed: 3 March 2023].

McQuillan, R. (2022) *What is Throwaway Prototyping? In-Depth Guide*. Budibase. Available at https://budibase.com/blog/inside-it/throwaway-prototyping/ [Accessed: 3 March 2023].

Nelly (2020) *PRODUCT & FEATURES: TIDYING UP DISCORD*. Discord. Available at https://discord.com/blog/tidying-up-discord [Accessed: 27 February 2023].

West, D. (2016) *Updates to the Scrum Guide: The 5 Scrum values take center stage*. Available at https://www.scrum.org/resources/blog/5-scrum-values-take-center-stage [Accessed: 3 March 2023].

Talab, Z. (2021) *What Is the Waterfall Software Development Methodology?* Codeguru. Available at https://www.codeguru.com/tools/waterfall-development-methodology/ [Accessed: 24 February 2023].

Thompson, K. (n.d.) *When to Use Scrum?* Cprime. Available at https://www.cprime.com/resources/blog/when-to-use-scrum/ [Accessed: 24 February 2023].

Upstack (2021) *Scrum Software: Pros And Cons To Consider*. Medium. Available at https://medium.com/geekculture/scrum-software-pros-and-cons-to-consider-b8d008061e04 [Accessed: 24 February 2023].

UX World (2021) *Why Predictability is Important for UX Design?* UX Design World. Available at https://uxdworld.com/2021/07/12/why-predictability-is-important-for-ux-design/ [Accessed: 27 February 2023].

Verheyen, G. (2013) *There's value in the Scrum Values*. Available at https://guntherverheyen.com/2013/05/03/theres-value-in-the-scrum-values/ [Accessed: 3 March 2023].

Volchko, J. (2017) *Prototyping Methodology: Steps on How to Use It Correctly*. Lumitex. Available at https://www.lumitex.com/blog/prototyping-methodology [Accessed: 3 March 2023].

Marking Rubric – Assignment 1

Section	Failed (0-49)	Third Class (50-59)	Second Class Lower (60-69)	Second Class Upper (70-79)	First Class (80-100)	Mark
Section 1: Software Development Methodology (30%) [CLO3]	No discussion or totally wrong description of software development methodologies for the 2 software.	1 of 2 software development methodology description is incorrect or both software development methodologies are described with very brief justification.	The software development methodologies are described with a few areas of justifications are unclear or incorrect.	The description of the software development methodologies for the 2 software are correct with 1-2 areas of justifications are unclear or incorrect.	The description of the software development methodologies for the 2 software are correct and the justifications are detailed and clear.	Raw mark /100 Section mark /30
Section 2: Design Review (30%) [CLO3]	No review of the scope and design aspects of the 2 software or all review areas are incorrect.	The review of the scope and design aspects of the 2 software are largely irrelevant with very limited evidence and explanations.	The review of the scope and design aspects of the 2 software are given but a few areas of scopes or design metrics are unclear or incorrect.	The review of the scope and design aspects of the 2 software are given with 1-2 areas of explanations are unclear or incorrect.	The review of the scope and design aspects of the 2 software are given and the explanations are detailed and clear with full evidence.	Raw mark /100 Section mark /30
Section 3: Software Ethics (20%) [CLO2]	No discussion on software ethics or good practices is given or all discussion are incorrect.	Only 1 of 2 software's software ethics or good practices is discussed, or both software's software ethics or good practices are discussed but largely irrelevant with very limited evidence and explanations.	The software ethics or good practices of 2 software are discussed but a few areas are unclear or incorrect, wrongly justified.	The software ethics or good practices of 2 software are discussed with 1-2 areas are unclear or incorrect, wrongly justified.	The software ethics or good practices of 2 software are discussed and the explanations are detailed and clear with full evidence.	Raw mark /100 Section mark /20

Section 4: Software Sustainability & Improvement (20%) [CLO2]	No discussion on software sustainability & improvement or all discussion are incorrect.	Only 1 of 2 software's sustainability & improvement is discussed, or both software's sustainability & improvement are discussed but irrelevant with very limited evidence and explanations.	The software's sustainability & improvement area of the 2 software are discussed but a few areas are unclear or incorrect, wrongly justified.	The software's sustainability & improvement area of the 2 software are discussed with 1-2 areas are unclear or incorrect, wrongly justified.	The software's sustainability & improvement area of the 2 software are discussed and the explanations are detailed and clear with full evidence.	Raw mark /100 Section mark /20
					Total Score:	/100

Written comments area: