FIT2101 Assignment 1 - Project Inception

Project vision

The elevator pitch template is used as a basis to help create a short but concise project vision statement.

For	1) Students 2) Markers
Who	1) Struggle with keeping track of their contributions and time spent on the group work 2) Are having troubles marking the students' group work
The...is a...	The program is a web-based group work management system
That	Allows students to easily input their time spent and contributions on group work so that it can be reviewed by the marker
Unlike	The current system
Our product	Provides a simple interface that reduces the hassle involved when students enter their tasks, which can be accurately reported to the marker

Vision

Our vision is to make students' and markers' lives easier when managing group work by creating a software that allows students to easily enter their time spent and contributions so that they can be reviewed by the marker.

Team

Team member	Contact information	Roles and responsibilities
Morad Abou Shadi (29799260)	**Email:** mabo0003@student.monash.edu **Contact no:** 018-3240299	- Scrum Master - Programmer
Ng Zhe Ren (29799538)	**Email:** zngg0018@student.monash.edu **Contact no:** 011-33383726	- Tester (Quality assurance) - Programmer

Teoh Ze Loong (29800692)	**Email:** zteo0009@student.monash.edu **Contact no:** 013-8982492	- Programmer - Minutes taker (Makes sure tasks are done on time)
Teo Wei Sheng (29800668)	**Email:** wteo0011@student.monash.edu **Contact no:** 012-9983599	- Programmer - Secretary (Handles the documentation)

Team Process Model + Scrum Ceremonies

Scrum will be used as the process model that we build our project on. In Scrum, teams will work on the project in an incremental and iterative manner through short cycles of development called sprints. With this systematic manner of work, we can focus on delivering high business values to the client in a short amount of time.

Scrum feature	Description / How our team plans on conducting the feature
Project inception	Scrum usually begins with a project inception, where members decide on the vision, technology stack, roles, and risks in the project. This will be conducted and addressed accordingly using this document. We will also work with the client to create an appropriate product backlog, as well as establish our project's definition of done.
Sprint Planning Meeting	After working everything out, we will conduct a sprint planning meeting (possibly during the lab) to come up with a sprint backlog before starting to work on our first sprint. Though the meeting is usually divided into two distinct sub-meetings, we will likely combine them and conduct them together due to certain constraints. During the meeting, we will work with the client to go over the highest-priority items in the product backlog and their feasibility before deciding whether we want to include them in our following sprint backlog.
Sprint, Daily Scrum	During the sprint, we are expected to conduct a Daily Scrum at the beginning of each day to keep each other updated on progress and any difficulties faced. However, this is not practical for our project. As an alternative, we will update each other everyday through either text or email and an emergency meeting will only

	be held if we encounter a major issue that has to be discussed thoroughly. Otherwise, we will typically only hold a meeting once per week to go over any important updates or issues over the past week.
Product review, Retrospective meeting, Backlog refinement	In a Scrum project, the team usually holds a product review and retrospective meeting after a sprint to go over the current status of the product as well as what went well and what went badly during the sprint. In our context, since sprints are likely to be conducted back-to-back, we will only have a short product review, backlog refinement, and retrospective meeting, while also coming up with the backlog for the next sprint, all in one meeting. With that, the entire process will be repeated until we eventually achieve our project's definition of done and is completed.

Task allocation and management

Through the initial scrum meeting in the inception phase, the team will identify the strengths and weaknesses of each member and come to an agreement regarding who is responsible for which task and the appropriate timeline for everything; keeping in mind that tasks to be worked on are not limited to those allocated to it, and are only held responsible.

Tracking time spent will be done through a timesheet that members fill in when they start and end the working session. This will allow the team to view how much of an input each member is making, which will be discussed during the usual scrum ceremonies. It is necessary to track time spent and contribution to ensure everything is done and to allocate more members to a given task if it requires extra effort.

Storing and keeping track of the backlogs and files will be accomplished given the fact that each member will have the software saved locally and through the use of Gitlab/Gitkraken which are project management git repositories. They also help the team track progress to ensure that the timeline is being followed, the timeline followed will be a dynamic structure that is set initially but open to change based on the situation; all the details will be discussed and decided during the scrum ceremonies.

Our definition of 'DONE'

➔ The feature or functionality that was implemented is working properly without any bugs or issues.
➔ Have complete documentation and rationale for all functionality and decisions.
➔ Revisit project scope and compare requirements with implemented functionality.

➔ Carry out testing and quality assurance strategies, ensuring software works as expected, is well covered, and cannot be manipulated.

Analysis of Alternatives

Summary

The analysis of alternatives essentially serves as a discussion hub for the major decisions that are to be made in our project, for instance, the programming language, communication tools and so on. In this section, we will compare and contrast the available options for each decision that is to be made so that an optimal solution for our project can be decided and agreed upon. To achieve that, we will factor in the pros and cons of each option in every decision against their alternatives, while taking into consideration what criteria we want to prioritize over the others in our project.

Terms of reference

Factor	Importance (Rate out of 10)
Cost	7
Functionality	10
Ease of use	7
Accessibility	8
Compatibility	10

Table 1.1

Table 1.1 displays the importance of each factor taken into consideration when choosing a certain option for different decisions throughout the project. The criterias are given a rating from 1 to 10, 1 being the least important while 10 being the most important.

Decision	**Option**	**Criteria**	**Ratings**
What IDE are we using	Atom	Cost	10 (cheapest)
		Functionality	8
		Ease of use	10
		Accessibility	10
		Compatibility	9
	JetBrains	Cost	10 (cheapest)
		Functionality	10

		Ease of use	8
		Accessibility	10
		Compatibility	10
	VisualStudio	Cost	10 (cheapest)
		Functionality	10
		Ease of use	5
		Accessibility	10
		Compatibility	10
What programing languages are we using	Javascript	Cost	10
		Functionality	10
		Ease of use	7
		Accessibility	7
		Compatibility	7
	HTML	Cost	10
		Functionality	6
		Ease of use	6
		Accessibility	8
		Compatibility	7
	CSS	Cost	10
		Functionality	8
		Ease of use	8
		Accessibility	8
		Compatibility	10
	C	Cost	10
		Functionality	7
		Ease of use	3
		Accessibility	8
		Compatibility	7

	Python	Cost	10
		Functionality	7
		Ease of use	10
		Accessibility	10
		Compatibility	7
	Java	Cost	10
		Functionality	9
		Ease of use	6
		Accessibility	7
		Compatibility	9
What communication tools are we using	Whatsapp	Cost	10
		Functionality	8
		Ease of use	10
		Accessibility	10
		Compatibility	10
	Discord	Cost	10
		Functionality	10
		Ease of use	10
		Accessibility	10
		Compatibility	10
	Zoom	Cost	6
		Functionality	8
		Ease of use	8
		Accessibility	8
		Compatibility	10
	Microsoft Team	Cost	6
		Functionality	8
		Ease of use	7

		Accessibility	9
		Compatibility	8
	Slack	Cost	7
		Functionality	9
		Ease of use	8
		Accessibility	7
		Compatibility	8
What platform are we using	Desktop app	Cost	9
		Functionality	10
		Ease of use	9
		Accessibility	8
		Compatibility	7
	Mobile app	Cost	7
		Functionality	10
		Ease of use	10
		Accessibility	9
		Compatibility	6
	Web app	Cost	10
		Functionality	8
		Ease of use	8
		Accessibility	10
		Compatibility	10

Body

Platform for application

When deciding on the platform to use for the application, the criteria that stands out the most is the accessibility and compatibility, which dictates how easy it is to use, navigate, and gain access to the application. Out of the 3 possible options, the web app seems to come out on top for these criterias due to its cross-platform nature, which allows for the user to access the application from various devices easily. On the other hand, desktop apps and mobile apps are more platform dependent, meaning that they are more targeted towards a specific demographic of devices or operating systems. For them to work across multiple platforms, we would have to have a unique set of code for each platform, which would prove to be slightly troublesome and time consuming.

On the other hand, in terms of the functionalities provided by using each platform, the web app falls short of both desktop app and mobile app. This is because desktop and mobile apps have greater access to their respective device's features and capabilities, while web apps are restricted to only the capabilities of the browser, which can prove to be very limited. This also causes web apps to run slightly slower compared to the other two.

Lastly, in terms of cost, web apps are generally cheaper to make compared to mobile and desktop apps. As mentioned earlier, since desktop and mobile apps are platform dependent, we would need to make separate versions of apps to cater for each platform that we want our app to target. Hence, that would also equate to a rise in cost of development. Not only that, mobile apps also require regular maintenance updates to check for bugs and security issues, which would also lead to an increase of cost. On the other hand, building a web app usually takes a shorter amount of time, which makes the cost relatively cheaper. It is also cross-platform, meaning that the one application we make can work across all platforms, without needing to add any additional changes, which also reduces the cost of development.

Programming language

While there are a myriad of programming languages that can be used to write an application, they all have different pros and cons which makes some better than the other in certain situations or circumstances. Hence, it is important to analyze each of them so that we can choose the most appropriate one for our project.

When deciding on a programming language, the cost typically does not matter as they are all usually free. It ultimately boils down to which of them is easier to use and each team member's proficiency and familiarity with each language. In terms of the ease of use criteria, Python is arguably one of the easiest programming languages to use and learn as it is very versatile and beginner-friendly. On the flip side of things, Javascript and Java will be a little harder, while C is easily the hardest language to use out of the ones that we have listed due to it being a lower level language. HTML and CSS will not be included in the comparison as they are more catered towards front-end development, and are essential if we ever decide to make a web app.

Next, the functionality of the programming language chosen is also an important aspect when deciding the programming language to use in our project. This criteria is harder to gauge due to the flexibility of programming languages, which allows them to perform a wide variety of functions. This means that a function created using a particular programming language can also be accomplished the same way using another. However, there are still differences as some programming languages generally perform something better or are more suited towards accomplishing something compared to the rest. For example, since

JavaScript can be used to code for both the front-end and back-end while also providing rich interfaces for webpages, it is very commonly used for web development. On the other hand, a programming language like Java is better for multi-threading, and might be better suited for programming mobile applications, given its object-oriented nature.

IDE

For our choice of IDE, the cost once again, doesn't really matter as most IDEs in the market will have a community version that is already well-suited and sufficient enough for us to develop our project.

On the other hand, a criteria such as functionality would definitely play a more prominent role in our decision-making process as it can dictate the speed and efficiency of our coding process. Comparing the IDEs listed, Visual Studio and Jetbrains indubitably offer a wider variety of functionalities to benefit the user compared to Atom, such as the ability to add breakpoints in the code, options to refactor and search for pieces of code, and so on. The functionalities that are offered by Atom are really basic and are nowhere as numerous as its counterparts. However, the huge number of functionalities provided also comes with a drawback, which is that it makes the IDE harder to use. Since the Visual Studio and Jetbrains offer an abundance of functionalities, it also means that they need to make space in their interface to fit them in. At some point, this large amount of features ends up cluttering the interface, making it harder for people that are unfamiliar with the IDE to navigate it. This is also why these two IDEs are often said to come with a slight learning curve. Contrarily, due to Atom only having basic functionalities, it has a more simple interface, which makes using the IDE a lot more easier.

Lastly, all the IDEs listed above are all stellar in terms of accessibility and compatibility as they can be installed easily from the browser on any computer. Hence, we do not have to take them into account when making our final decision.

Communication medium

Finally, a good communication tool that is easy to use, highly compatible, and is equipped with the appropriate functionalities is imperative towards building a successful project. Hence, the available communication tools have to be examined based on the specified criterias so that we can choose the appropriate one for our project.

Right off the bat, the cost of the communication tool plays an important role as some of them might incur extra charges for extra storage, voice calls with more people, longer voice calls, and so on. Referring to the options that we have listed, WhatsApp and Discord come out ahead in this criteria as they are completely free to use without having any restrictions just because they are free. On the other hand, the free version of Zoom only allows for a maximum meeting time of 40 mins, which is not that ideal for doing projects, Microsoft Team requires purchasing, and Slack also has some features that are only available to its premium paid version.

Next, in our opinion, having used all the communication tools listed, they are all almost at an equal footing in terms of compatibility, as they are all able to be accessed from both mobile devices and computers. However, in terms of ease of access, WhatsApp and Discord once again remain superior as their simple interface makes it easy to access and communicate with each other. The other tools are slightly harder to set up, which would take up more time and is more troublesome.

Recommendations

The platform our team has decided to use is **web application** as it is less costly compared to mobile app and desktop application. The reason it is less costly is because mobile applications require different implementations for android phones and iPhones. Meanwhile only one implementation of the application is needed if we implement a web application. Not only that, web applications have higher portability as mobile phones, desktop, and laptops can access web applications.

Given the fact that we have prior knowledge of using **JavaScript, HTML and CSS** to program web pages, we will select them as our choice of programming languages, while using Atom IDE to perform our coding as they are suitable for building a web application platform. JavaScript, HTML, and CSS are also one of the best options to be used when developing a web application.

For communication between team members, **Discord** and **WhatsApp** are used because of how easy it is to use them, as both platforms are compatible with both mobile and desktop so it is easier to gain access to them. They also contain a wide variety of functionalities that can greatly benefit us in our project, on top of being absolutely free to use.

Risk register / Risk management plan

Risks are uncertain events that may occur throughout the project which can have detrimental effects towards its outcome. Hence, to prevent the risks from happening, or to simply mitigate its effects, we ought to come up with a risk management plan that evaluates the types of risks that might occur, their likelihood, and how to mitigate them.

Risk 1:

Type of risk	Likelihood (1-10)	Impact	Monitoring strategy (if any)
Poor management and work delegation	7	- Project will not be completed in scheduled time - Some team members might be overworked while others are underworked - If the project is not showing much progress towards the end due to poor work delegation, the team might try to rush the project as a final resort, causing the software produced to be of poor quality	- Have each team member record down their progress and time spent on tasks on a timesheet

Mitigation plan

Hold a discussion at the start of the project to discuss and decide each team member's roles and responsibilities. The roles can be recorded in a document so that each team member can be held accountable if they do not fulfil and perform their duties.

Risk 2:

Type of risk	Likelihood (1-10)	Impact	Monitoring strategy (if any)
Server crash or any technical difficulties	5	- Project will not be completed in scheduled time - Data might be lost, delaying the completion of the project	- None

Mitigation plan

Have an offline copy of the work with up to date functionalities as the server

Risk 3:

Type of risk	Likelihood (1-10)	Impact	Monitoring strategy(if any)
Team member experiencing health issues	4	- Role or responsibility being taken by that team member cannot be carried out - Product cannot be finished by deadline	- Assign someone to check on the condition of each and every team member to make sure every team member's health is ensured. - Make sure working hours are at acceptable durations to avoid overworking team members.

Mitigation plan

Hold online conferences when a team member contracts an infectious disease. Reduce the workload of someone that is getting sick.

Risk 4 :

Type of risk	Likelihood (1-10)	Impact	Monitoring strategy(if any)
Technical risk such as team members not knowing how to code using a needed programming language	5	- Project will not be completed on time - Some team members might have to work extra to cover for the mistake of the team member	- Make sure the team members have a grasp of the ability of others' coding abilities.

Mitigation plan

Make sure every team member has the sufficient skills required for the project. Have them familiarize themselves by writing “Hello, World” in the required language.

Risk 5:

Type of risk	Likelihood (1-10)	Impact	Monitoring strategy(if any)
Data Loss	6	- Project will not be completed on time - Might lose recent implemented functionalities - Increase additional workload on everyone	- none

Mitigation plan

Have a backup of the code/project at a different server or an offline copy. Update the back up data frequently.

Risk 6:

Type of risk	Likelihood (1-10)	Impact	Monitoring strategy(if any)
Team members facing resource issues (such as laptop hardware issues).	3	- Project will not be completed on time - Some team members might have to work extra to cover for the mistake of the team member	- Make sure resources used by team members are always on good condition

Mitigation plan

Have a backup resource for team members to use if any issue occurs to the resources being used. For instance, have a backup laptop for team members to use if their laptop breaks down.