

## ANTI-SNAKES SECURITY/ SNAKE STALKER (ASS/SS)

# PROJECT PLAN

### Project's Vision

For students and markers

who need to track the contributions of group members in a software project,

the ANTI-SNAKES SECURITY

is a time and task tracker

that can effectively track the time spent on the project and the amount of work done by each group member in a team project

Unlike grading superficially based on the final result,

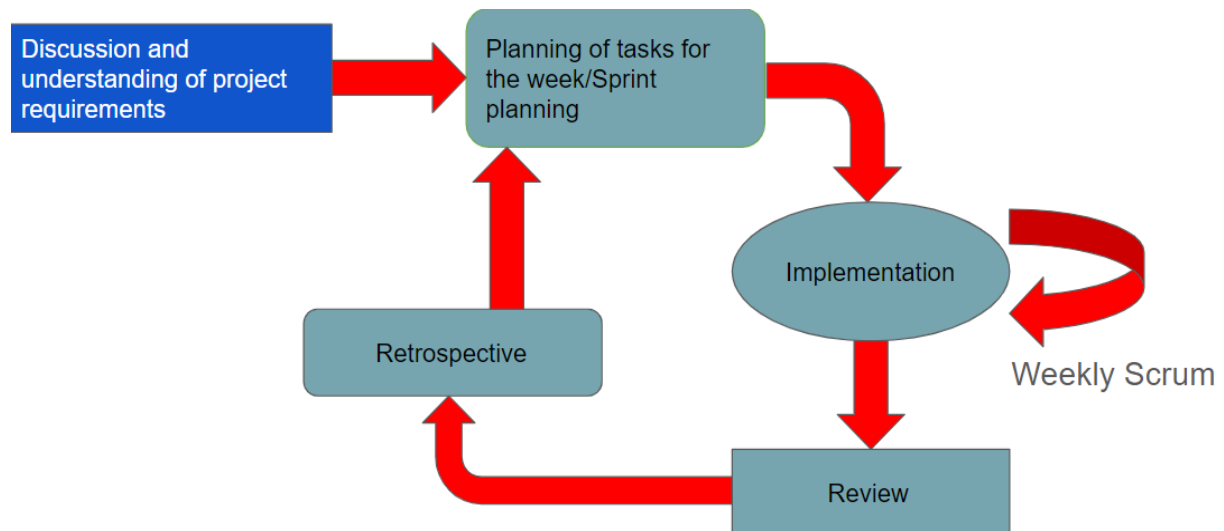
our product would ensure fairness prevails and is accessible anytime and anywhere.

### Team Members

NAME	Role and Responsibility	Contact
Julian Yong Hao	Role: Developer / Scrum Master  Responsibility: Plan and breakdown the team's task among team members and ensure team members move forward with their task while understanding the intended outcomes. Ensure the program is coded as intended.	<a href="mailto:jyon0017@student.monash.edu">jyon0017@student.monash.edu</a>
Low Chern Wey Joshua	Role: Developer / Quality assurance  Responsibility: Ensuring the program runs as intended and informs the team of possible missteps in the project.	<a href="mailto:clow0007@student.monash.edu">clow0007@student.monash.edu</a>
Teh Jian Xiang	Role: Developer / Product owner  Responsibility: Ensure effective communication between the team and key stakeholders, confirm every step taken by the team is on track.	<a href="mailto:jteh0009@student.monash.edu">jteh0009@student.monash.edu</a>

Marcus Lim Tau Whang	Role: Developer  Responsibility: Focuses on front-end development (UX & UI), ensuring ease and smoothness of interaction between users and the web app.	<a href="mailto:mlim0032@student.monash.edu">mlim0032@student.monash.edu</a>
Tan Song Shun	Role: Developer  Responsibility: Ensure the back end of the application is implemented correctly.	<a href="mailto:stan0101@student.monash.edu">stan0101@student.monash.edu</a>

### **Process Model**



Based on the process model above, our team will start on discussing and understanding the project requirements, set up our team organization and make fundamental decisions. Following on, we will start our development works in sprints. In every iteration of the sprint, we will have our sprint planning, implementation, review and retrospective and instead of having daily scrum meetings, we will have weekly scrum meetings during our lab times. We made this decision as we considered that everyone's time could match with the daily scrum meetings.

### **Definition of Done**

The work is done when the Software is/has:

1. Reviewed and tested by the whole team.
2. Met all the requirements of the clients and performed as per the requirements.
3. Reviewed and approved by the client.

## **Scrum Ceremonies**

- Sprint planning
  - Run before we start each sprint.
  - Decide to implement how much task is to be set on the product backlog.
  - Set a goal and how we achieve it.
  - Distribution of tasks and identify roles of each team member.
- Weekly Scrum
  - Scrum meetings will be run weekly during our lab times.
  - Review the team's progress and possible adjustments to the current plan.
- Sprint review
  - Run at the conclusion of each sprint.
  - Review the sprint outcome with product owner and stakeholders.
- Retrospective
  - Run at the conclusion of each sprint.
  - Review product development and identify areas for improvement with the team.

## **How we allocate tasks**

Steps:

1. Have a discussion through a meeting with the team.
2. Discuss the details and requirements of the tasks.
3. At the end of discussion, each team member would be able to understand the requirements of each task and the approach of the team in general.
4. Discuss with each team member and ask which part of the task they would prefer.
5. Delegate the task according to their ability, preferences and compatibility with the specific task.

## **How we track our progress**

Steps:

1. Report during our weekly scrum meeting.
2. Daily progress will be reported in Whatsapp group.
3. Working together at the same time and hitting the goal that we set before we start our work.
4. Check Github commits and issues.
5. Have a dedicated member be responsible for ensuring the progress is advancing smoothly.

## **How we manage our backlogs**

Steps:

1. Constantly reviewing the backlog.
2. Updating the backlog after each completed task/when a new task is allocated.

## **How we track time spent on a task**

Steps:

1. Constantly reviewing the backlog.
2. Team members constantly update their progress.
3. Check github commits.

# ANALYSIS OF ALTERNATIVES

## Programming languages:

### Javascript ✓

#### Pros

- Very fast as it runs on Google App Engine.
- Can process multiple requests at once which may be important for the student tracker.

#### Cons

- There is a slight learning curve to code in Javascript especially for the backend, and all team members have limited to no Javascript experience.
- Lack of documentation may hamper development speed.

### Python

#### Pros

- Beginner friendly language especially for web development and most team members have proficiency in Python.
- Extensive documentation on various libraries such as Flask and Django for web development are available which will speed up the development significantly.

#### Cons

- Slow when compared to Javascript.
- Not very scalable, requires multithreading for complex web applications which the team has no experience in.
- Python is not suitable for real time web applications, which means it might not be suitable for the student tracker.

### PHP

#### Pros

- Build for web development which makes it suitable for rapid web development.
- Extensive database support which may be important for student tracker because we are definitely going to store quite a bit of data.
- Not a very complex language which reduces the time required to learn it.
- Decent speed.

#### Cons

- Not suitable for complex web applications on its own and requires external web frameworks for it, which means that team members have to dedicate extra time and effort to learn the frameworks before even starting to develop the application.
- Have some security flaws.

(\*Additionally use HTML and CSS to make the web application more user friendly and attractive. )

## **Platform (Web/Desktop/Mobile application):**

### **Web application ✓**

#### **Pros**

- Can be accessed anywhere with almost any device.
- User does not need to update the application because the application is updated in the backend server.

#### **Cons**

- Harder to set up and deploy, requires decent knowledge on web application architecture.
- Might be slower compared to computer/mobile applications.

### **Desktop application**

#### **Pros**

- Runs faster than both web and mobile applications.
- Easiest to develop out of the three platforms.

#### **Cons**

- Can only be accessed through a laptop/desktop which limits user mobility.
- The application must be installed on the PC which users might not like.
- Users cannot update applications easily as they have to download patches from the internet.

### **Mobile application**

#### **Pros**

- Users can easily access the application through a mobile phone and receive notifications at any time.
- Easy to push updates to the Google Play Store, and users will be able to update automatically.

#### **Cons**

- Developing mobile applications requires knowledge on Android Studio which is not the easiest to learn.
- Careful optimisation is required as mobile applications are prone to crashing if it is optimised poorly.

### **Spike:**

### **Programming Language**

#### **Javascript:**

```
console.log("Hello World");
```

Hello World

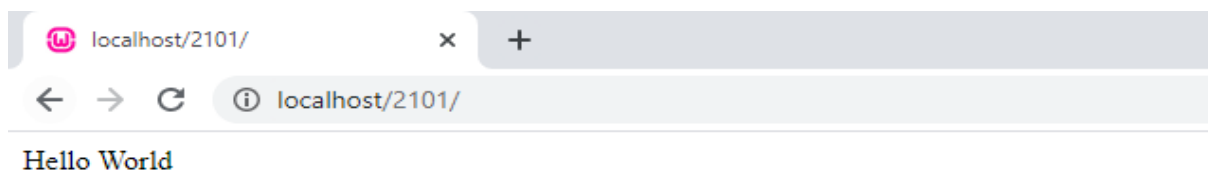
helloworld.js:3

#### **Python:**

```
>>> print("Hello World")  
Hello World
```

#### **PHP:**

```
1 <?php  
2 echo "Hello World";  
3 ?>  
4
```



### **Reasons of choice of programming language:**

- Javascript has the ability to support both backend and frontend development, and there's no need for change of language/environment.
- By using javascript in both the frontend and backend, bridging is not required unlike when using python for backend.
- Javascript is good at handling applications with high amounts of input/output, which the student tracker will definitely experience.

### **Reasons of choice of platform:**

- The client wants to access the application at anywhere and anytime with any device, hence a web app is more suitable, as it can be accessed with just an internet connection.
- Web applications have a better user experience than any other application. As every student and marker are accessing the same version of web app, they will always be accessing the most-up-to-date version of software

### **Ways of communicating and collaborating with each other:**

- Tools we plan to use are Zoom, Whatsapp and Email:  
Factors that are taken into consideration are preference and accessibility of members.

Meetings and face to face discussions will be held on **Zoom** Sessions. Unlike Microsoft Teams, Zoom's screen sharing for coding is easier to perform, as members could easily annotate the screen for any queries/comments.

**Whatsapp** is used by every member of the team hence it will be easy and convenient to communicate among team members even when everyone's schedule is different. It can also be used to track progress very effectively.

**Email** is used to access our created Google drive and Google Docs to complete documentation tasks by all members simultaneously. Emails sent by the client can be easily forwarded to every team member to avoid passing wrong messages and information.

- Strategies to communicate effectively:
  - Meeting schedule and agendas are fixed beforehand.
  - Ask appropriate and direct questions to the right person.
  - Constantly update with each other on each's progress.
  - Ensure each member is kept up to date with the task at hand.

# RISK REGISTER

No.	RISK	IMPACT (0-5)	LIKELIHOOD (0-5)	MONITORING STRATEGY	MITIGATION
1	Failure to deliver project deliverables due to not being able to work on the project due to personal issues	4	3	Team members should report their own progress and if someone cannot work on the project the tasks need to be reassigned	Ensure a good amount of work is done daily
2	Technical difficulties/ function cannot be implemented which causes deliverables to be delayed	3	5	Find resources online or get external help. Reflect to the client if the function required is unable to implement.	Make sure a requirement is possible to implement with the team's technical level before even accepting the requirement
3	Data security breach which causes user data to be stolen/exposed	4	2	Make sure proper authentication is used and security guidelines are followed	Use intrusion detection mechanisms to get hold of data breaches before it causes damage.  Use Encryption
4	Conflicts in the team due to differences in opinion and ideas for the project causes deliverables to be delayed	4	1	Always ask each team member what do they think about the progress of the project and if they are satisfied with the running of the team	Constantly have discussion to ensure each member understands the flow of the team's direction and any underlying concerns that some members may have should be answered as soon as possible to ease them.



**Impact:**

TERM	DEFINITION
<b>EXTREME (5)</b>	May derail or destroy progress of project
<b>HIGH (4)</b>	Major effect on the project
<b>MODERATE (3)</b>	May affect some key parts of the project
<b>LOW (2)</b>	A small error that can be easily fixed
<b>INSIGNIFICANT (1)</b>	Negligible effect on project
<b>NONE (0)</b>	No need for concern

**Likelihood:**

TERM	DEFINITION
<b>VERY LIKELY (5)</b>	Will occur frequently throughout the project
<b>LIKELY (4)</b>	Likely to occur frequently
<b>POSSIBLE (3)</b>	May occur at least once or twice
<b>UNLIKELY (2)</b>	Not generally expected to occur
<b>VERY UNLIKELY (1)</b>	Has the possibility of occurring although it would be very rare
<b>IMPOSSIBLE (0)</b>	Never occurs