

Team Information

The Team Dream
CL_Tuesday04pm_Team012



Team Membership

Name	Contact	Technical Strengths	Professional Strengths	Fun Fact
Aryan Chordia (Team Lead Sprint 1)	acho0098@student.mona sh.edu	<ul style="list-style-type: none"> • Java • Python • SQL 	<ul style="list-style-type: none"> • Time Management • AGILE • Communication 	I like swimming.
Naailah Taqui Hasan	nhas0021@student.mona sh.edu	<ul style="list-style-type: none"> • Java • Python • Unit Testing 	<ul style="list-style-type: none"> • Time Management • Project Coordination • Agile Project management 	I know Morse Code and I am learning AUSLAN
Sunny Cho	scho0103@student.mona sh.edu	<ul style="list-style-type: none"> • Python • Java • Game Development 	<ul style="list-style-type: none"> • Teaching Experience • Time Management • Open Mindedness 	I play the electric guitar (badly).

Team Schedule

Total Weekly Commitment: 6–8 hours per member, members can complete their hours at any time during the week based on their availability and task requirements. This includes collaborative hours spent working with others.

Weekly Meeting Schedule

- **Thursday at 9:00 PM** – Regular Meeting
- **Friday at 5:00 PM** – If Required
- **Tuesday (Before or After 4:00 PM Workshop)**

Workload Management

- The workload will be distributed evenly among the three team members, with each taking on tasks they feel most confident in.
- If a team member is unable to complete their work or requires assistance, they must promptly notify the others via Discord, our chosen communication platform. The team will then collaborate to find a solution and ensure the project's progress remains on track.
- If a team member is unable to complete a task due to personal reasons, which must be respected and not questioned by others, they must immediately inform the rest of the team. The remaining members will then collaborate to find a solution and ensure the project's progress remains uninterrupted.

Technology Stack and Justification

Selected Tech Stack

Python with Tkinter

Tkinter is Python's standard library for creating graphical user interfaces (GUIs), making it a versatile and widely used tool for developing a range of applications, including utility tools, games, and other interactive graphical software. Its inclusion in Python's standard library ensures accessibility and ease of integration without requiring additional installations. Tkinter provides developers with a suite of customizable widgets such as buttons, labels, text fields, and canvases, enabling the creation of user-friendly and visually appealing applications.

As a Python-based library, Tkinter is inherently cross-platform, allowing applications to run seamlessly across Windows, macOS, and Linux with minimal modifications. Its simplicity and readability make it an excellent choice for developers of all experience levels, from beginners exploring GUI development to professionals seeking a quick prototyping tool. Moreover, Python's flexible and dynamic nature supports agile development and rapid iteration.

However, the dynamic typing system in Python lacks the static analysability provided by languages like Java, which rely on stricter type systems. To mitigate this, Python offers type hints, which, when used in strict mode, can provide a degree of static analysis and improve code clarity. This partial enforcement of type safety combines with Python's adaptability to uphold a balance between flexibility and robustness. Nevertheless, this advantage comes with a trade-off, as developers must maintain coding discipline. When working in collaborative environments, it's critical to ensure that the problem space is clearly and consistently expressed in code to avoid ambiguity and potential bugs.

Tkinter is ideal for projects that prioritize ease of use and cross-platform compatibility, while also requiring the flexibility to adapt and evolve as the application grows. It bridges the gap between simplicity and functionality, making it a reliable option for diverse software development needs.

Although the team has more experience with Java-based game development due to prior coursework, most members are familiar with Tkinter and find it more intuitive for UI development. Additionally, Tkinter is more beginner-friendly compared to Java Swing, making it an accessible choice for all contributors. Given these advantages: ease of use, cross-platform compatibility, and flexibility, Python with Tkinter has been selected as the preferred tech stack.

Excluded Tech Stack

Java with Java Swing

While the team has experience with Java from previous coursework, Swing requires a more complex setup for GUI development compared to Tkinter. This makes it less beginner-friendly and more time-consuming to implement. The team also possesses substantial experience in Python.

Java Swing often requires a significant amount of boilerplate code to create even simple UI elements, making development slower and harder to maintain. In contrast, Tkinter allows for more concise and readable code. While Java is cross-platform, Swing applications may not always look or behave consistently across different operating systems.

Java Swing does provide a richer set of pre-built UI components compared to Tkinter, offering greater customization for complex applications. It is also platform-independent, ensuring consistent performance across various operating systems. However, the advantages of Tkinter ultimately outweigh those of Java Swing. Tkinter better aligns with the team's experience, project requirements, and development priorities, leading to its selection over Java Swing.