Team Introduction

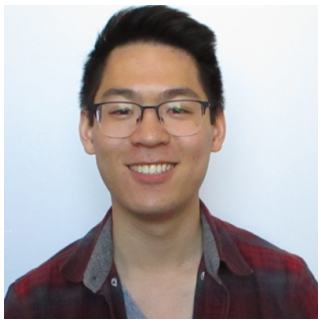
Team name: Team UTSGSC

Picture of team:

Team's goals in the project: The team's ultimate goal is to learn the best practices and principles of engineering large software systems and the processes associated with contributing to the open-source community. To define more concretely, the team would ideally like to fully implement a feature or fix an issue that would be accepted into the master branch of an open-source project. Although, this may be a little challenging, the team is ready to take this on by applying the technical skills learned throughout internships and would love to walk away with the experience and motivation to further contribute to open-source projects beyond this course.

Team's strengths: We are a diverse group of Computer Science students with a passion for Software Engineering. This passion is reinforced by our diverse strengths in mobile and full stack development gained through our combined internship experience at over 8 companies including Red Hat, Shopify, Salesforce and Google in various technology stacks. In addition to our technical expertise, we also have the ability to effectively work together as we have proven teamwork and communication skills to provide a transparent, efficient and supportive work environment.

Team Member Introductions

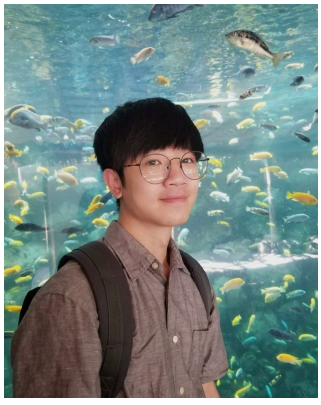


Ka Kit Jason Cheung

My name is Jason, I'm currently a fourth year Computer Science student specializing in Software Engineering at the University of Toronto. I have experience as a full stack developer working with technologies and languages such as Angular, Ruby on Rails, Flask, PostgreSQL, Python and Ruby. I've also accumulated one year of internship experience at companies such as Index Exchange, Intersect and Google where I wore multiple hats like Scrum Master, Software Engineer and Technical Solutions Consultant, and was able to provide meaningful impact through my work. My technical expertise is mainly in back-end development and cloud, especially Google Cloud Platform but am passionate about continuous learning in all domains of technology and software development. My non-technical interests include hiking, binge watching Netflix, trying out different types of cuisines, restaurants and being an all around foodie.

Arvinth Vijayanathan

I am a fourth year Computer Science Specialist student from the University of Toronto St. George. My coursework along with internship experience brings my strengths in software engineering. Completing an year of internships has given me experience in software testing and software development working at Indigo, Intersect, and Salesforce. Through both professional and personal experiences I have acquired knowledge to be comfortable to quickly adapt to various technologies, languages, and framework. My expertise however would be in Java, Python, and Flutter but I seek to explore and learn more technologies. As a near to graduate computer science student I hope to join a software development team that would help me further expand my skill set. Eventually hoping to apply my skills into a start-up of my own. Outside of the technical world I often seek to get out of my comfort zone, by travelling to different countries, trying new activities, and exploring different cultures.



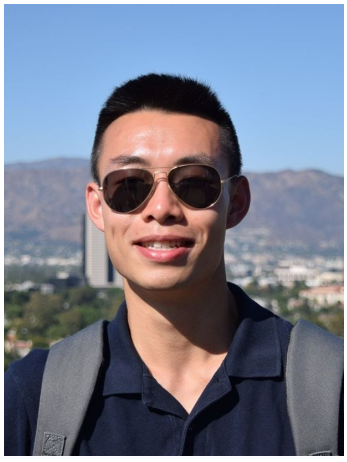
Peter Xiao

I am currently a fourth year student studying computer science at the University of Toronto. My focus of study within the program is software engineering. Across the many courses that I have taken within my program, I gained many skills that I was able to apply to the internships that I have done. After trying out the many different ways I can apply my knowledge in the field, I found a liking to android development. My speciality is in android development, mainly utilizing

Kotlin/Java to develop android applications. I am also proficient in Python. With Google constantly developing new and different ways to create and maintain android applications, I am always excited to be learning the new tools that could be used. Since I am near completion of my program, I hope to find a company that would suit my personality, as well as a team of people that I would look forward to be working with every day. Outside of school, I love playing the piano, as well as playing volleyball and video games in my leisure time. I am constantly looking to improve my skills, as well as learn new things for my leisure activities.

Mathusan Selvarajah

I am a final year computer science student at the University of Toronto. Throughout my post secondary life I have always tried to jump into and learn about different fields in computer science, which I was able to achieve through my internships and personal projects. Lately while working at Red Hat, I was on the Kubernetes team where I was a regular contributor to the storage aspects of the Kubernetes open source project. Through my previous internships at Connected and Gemalto, I was also able to work in the IOT and Cryptography fields. Aside from school and internships I like to learn and work on projects on my own. Through these projects I have been able to learn some native IOS development and cross platform (IOS and Android) development using the Flutter framework. In terms of technical strengths, I have had experiences in using many different languages and frameworks but would consider Python, Java, JavaScript, and GO as my strengths. Above all I enjoy working together with new people to solve problems. Aside from school and my technology interests, I really enjoy watching and playing basketball, travelling to new countries, and last but not least the occasional gaming.



Richard Wei

I'm a fourth year coop computer science student at the university of Toronto. I'm immensely passionate about Android development, with experiences at Shopify developing both Android apps as an intern on the Point of Sale app team and Android libraries on the Mobile App Bridge team. I have a deep interest in the internal workings of both application development and the systems they run on which I am exploring through database management systems and compiler optimization courses. I'm proficient in Java, Kotlin, Android development, Android Architecture, as well as Python, Typescript and C. I'm learning C++ at the moment as well! Most of my free time is divided between developing mobile applications and learning about how something works (like React Native or Flutter) but when I'm not focused on these, I enjoy working out to clear my head, reading sci-fi and fantasy, and digital painting

Team Agreement

Methods of communication and response times

- Primary long distance communication will be done through a Facebook Messenger group chat that contains all group members. This group chat will be used for all project-related communication including technical project conversations such as blockers, progress updates and non-technical project conversations like coordinating in person meeting times to discuss, demo, and work together.
    - **Expected response times**: Within 6 hours for every message unless explicitly stated.
    - Video should also be used in the group chat wherever the team decides it may be easier to communicate verbally instead of through text.
- Secondary long distance communication will be done through text message. Team members will share their phone numbers with each other. Text messages will be used as a last resort when team members are not responsive and are required immediately. Examples include when a team meeting is about to start but a member has not showed up and has not previously communicated that they will be absent.
    - **Expected response times**: Within 30 minutes.

Meetings

- Weekly Meetings:
    - Once the project begins, there will be weekly meetings where each team member will demonstrate the progress, completed work, and any other details to update the team. This should mainly focus on work done within the given week, unless there is work overflowing from previous weeks.
    - **Mandatory?** Yes.
    - **Attendance:** Attendance by each member is mandatory so that everyone has clear visibility of progress made by each team member. If a team member cannot attend, they must notify every member of the team at least 12 hours prior to the start of the meeting. If a member is not able to attend, they must send their update in the Facebook Messenger group chat by the end of the day of the meeting.
    - **When:** Friday 3pm-4pm. Can be shorter depending on how much content there is in the meeting.
    - **Where:** UTSC, HW table unless otherwise specified.
    - **Face-to-face or Online:** Face-to-face
    - **Who takes minutes and where they can be found:** Minutes can be found under the meetings/weekly/ directory in the team_10 GitHub repository.
    - **Preparation:** Preparation required to demonstrate progress, finished work, etc. This can include demos as well.
- Stand up/Progress meeting:

- Once the project begins, there will be short stand up meetings around 15 - 20 minutes long, 3 times a week. These meetings will be structured so that each member will take turns talking about:
- What was accomplished.
- What are the blockers.
- What will be accomplished by the next progress meeting.
- **Mandatory?** Depends on how much work and progress is made between meetings.
- **Attendance:** Attendance by each member is mandatory so that everyone has clear visibility of progress made by each team member and potentially assist with others' blockers. If a team member cannot attend, they must notify every member of the team at least 6 hours prior to the start of the meeting. If a member is not able to attend, they must send their update in the Facebook Messenger group chat by the end of the day of the meeting.
- **When:** Every Monday, Wednesday, Friday from 6pm-6:15pm. Can be shorter/longer depending on how much content there is in the meeting.
- **Where:** Video call on Facebook Messenger.
- **Face-to-face or Online:** Online
- **Who takes minutes and where they can be found:** Minutes can be found under the meetings/standup/ directory in the team_10 GitHub repository.
- **Preparation:** Preparation is optional. Can be used to demonstrate things involved in the meeting like blockers, etc. This can include demos as well.

Version Control

Note that we the commit and style guidelines of the project (if it has any) will always take precedence over what we've decided below.

- **What to commit:**
  - All team members are expected to fully proofread and test any code before committing it into the repository.
  - Members should not be commiting code that is not up to common coding standards.
    - Unused code should not be committed
    - Unnecessary print statements should not be committed
    - Code must be readable to other team member
    - Proper variable names should be used
    - Naming conventions used should align with the project chosen
- **Commit message standards:**
  - Team members should use commit messages the provide valuable information about what has been done
  - Commit messages should be in proper english
  - Team members should commit code every time an ample amount of work has been completed.

Division of Work

- First look through and list out the issues/features that would be worked on as a group
- After which an estimate on the hours required to complete each task would be required. The sum of these hours gives the estimated total hours of work.
    - Dividing this total by 5 will tell approximately how much work (in terms of hours) each team member is required to contribute. With this in mind the set of tasks (including features and issues) would be delegated to each group member and would be expected to be completed by the deadlines that are formed from a group consensus.
    - If work was incomplete by the deadline due to blockers or any other factors, the team will discuss and come up with a plan to help the team member move his/her task to completion.

Submitting Deliverables

- **When to submit:** All code and deliverables must be submitted by the deadline indicated on the deliverable instruction sheet, course website or by the instructor.
- **Who will submit:** Code will be submitted by the author of the code, whereas non-code deliverables will be submitted by alternating team members. The submitters will be in the order of Arvinth, Jason, Peter, Mathusan, Richard and repeat until the last deliverable of the course.
- **Who will review the submission:** Prior to the final submission of deliverables, they must be reviewed by all members of the team and approved. Code must have at least one approval before it's merged into the codebase. If pull requests have unavoidable complexity, there must be at least 2 approval before merging.

Contingency Plans

- **What if a team member drops out?**
    - In the case a team member drops out, the work assigned to the member would be equally split amongst the remaining members.
    - If for some reason the task can not be completed by any remaining member on the team, the team will get in contact with the instructor to figure out what to do.
- **What if a team member consistently misses meetings?**
    - Contact the team member and find out the reason for this. Ask if another time would work better for them.
    - If we cannot come to a resolution, the team will get in contact with the instructor regarding the issue.
- **What if a team member is academically dishonest?**
    - Inform the instructor about this

We accept these guidelines and intend to fulfill them (sign below):

S. Madhusan

Peter Xiao

Review the guidelines with your TA. Make a copy for the team and submit the agreement to the TA. In the event of team disagreements, you may be asked to show this form to your TA or instructor.