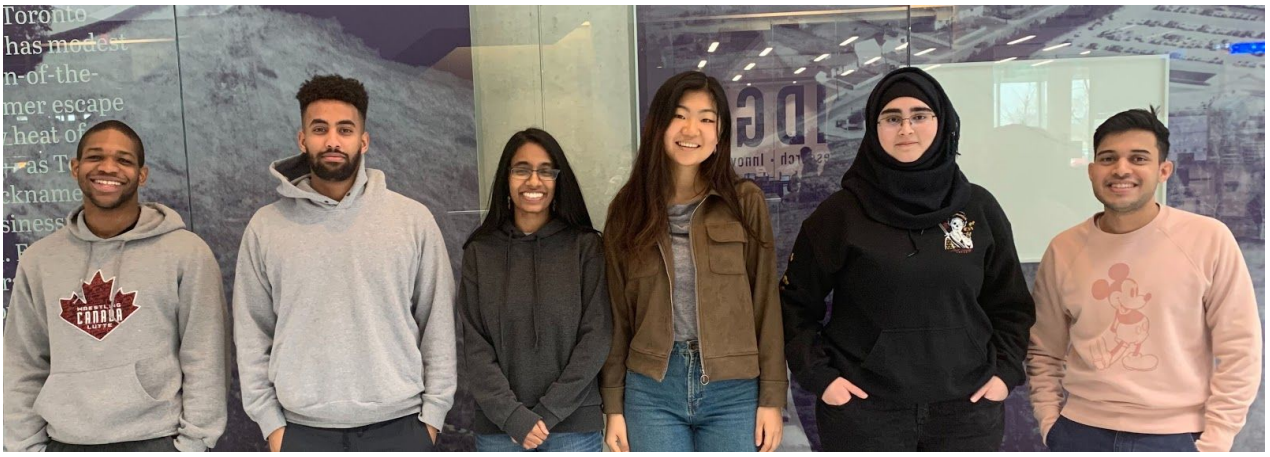
Deliverable 0

Team_01 (GitHub):
Abel Debalkew
Carl Alvares
Jayatharani Thirugnanasampanthan
Kemar Harris
Lan Yao
Seemin Syed

Table of Contents

Team Introduction: FOSSkateers



(L-R) Kemar Harris, Abel Debalkew, Jayatharani Thirugnanasampanthan, Lan Yao, Seemin Syed, Carl Alvares

Goals

FOSSkateers is looking to support the open-source community as contributors; the team does not simply want to consume the software they use, but play an active role in developing and maintaining such software. As such, the team is interested in joining a useful project welcoming to first-time contributors.

One of FOSSkateers' goals is to understand an open-source codebase with the help of documentation and full-time project members. The team's more ambiguous goal is to make substantial contributions to the codebase in a short period of time. Specifically, the team is looking to have a few significant bug fixes and features accepted to the codebase. All bug fixes and features will be properly documented, implemented, and tested according to the project's guidelines.

Strengths

Technical Strengths

All members of FOSSkateers have prior experience in designing, developing, testing, and maintaining real-world software; these experiences will enable FOSSkateers to contribute quality code to the open-source project. Seemin Syed has prior experience working in quality assurance (QA). Carl Alvares and Jayatharani Thirugnanasampanthan have extensive back-end web development experience. Lan Yao has prior experience with front-end web development. Abel Debalkew has experience in full stack web development and data analysis. Kemar Harris also has experience in working with data analysis and has worked with large software systems.

In addition, all members have prior experience with Python and Git through academic work and internships, which can be considered a strength when working on the open-source project[1].

Non-Technical Strengths

All members of FOSSkateers are comfortable with writing, which will be an asset when documenting our work. All members also reported different experiences working in teams. Sharing these experiences in the initial meeting allowed FOSSkateers to discuss the team expectations agreement thoroughly. Having worked in teams previously helped some members develop strategies to handle conflicts.

(L-R) Abel Debalkew, Lan Yao, Carl Alvares, Seemin Syed, Jayatharani Thirugnanasampanthan, Kemar Harris

[1] The team's top choices for open-source projects are written in Python.

Team Member Biographies

Abel Debalkew

Abel Debalkew is a fifth year Computer Science and Neuroscience double major at the University of Toronto. He has interests in pursuing Software Engineering and exploring the applications of technology in other fields.

He has completed two previous internships. The first was as a Bioinformatics Intern at the Hospital for Sick Children where his main tasks involved classifying medulloblastoma tumors and data visualization.The second was as a Software Developer at the University Health Network where he built web applications to store patient information from clinics.

Abel has also completed numerous projects at school and outside of school. These projects helped him become familiar with numerous technologies. Abel is eager to contribute as a Software Engineer and learn more about the discipline. Outside of work and school, he has many interests such as music, exercise, shows, reading, and spending time with family and friends.

Carl Alvares

Carl Alvares is a third year Computer Science student at the University of Toronto. He has a background in Finance. His interests are in software engineering, with a focus on back-end web development.

He has most recently done a co-op position as a Software Developer at CaseWare, an accounting software company. At CaseWare, Carl worked on developing APIs and crafting queries to support new features for AnalyticsAI, a web application which automatically analyses financial statements and flags suspicious entries using a variety of accounting tests.

Carl had the opportunity to attend HackNYU 2019, where he and his team achieved 3rd place in the Financial Empowerment track for their project, SubTrakt, a Chrome extension which automatically tracks your subscription services and reminds you when bill payments are due.

Outside of software, Carl enjoys travelling, Nintendo games, and discussing current events.

Jayatharani Thirugnanasampanthan

Jayatharani (Jaya) Thirugnanasampanthan is a fifth year Computer Science student at the University of Toronto. Her interests lie in Software Engineering, as well as the intersection of computing and the humanities (Digital Humanities).

Jaya completed her Professional Experience Year (PEY) in 2018-2019 working as a developer in the public sector. In this role, she assisted in the development and maintenance of several web applications. Jaya's strong involvement in these applications eventually led her to participate actively in the discussions between clients and business analysts to clarify business requirements.

Jaya's PEY experience has sparked her interest in being part of a client-facing team. After graduation, she will assist clients across various industries in the design, functional implementation, and support of Salesforce customer relationship management (CRM) applications.

Outside of her professional life, Jaya is keen on understanding urban issues affecting Toronto and is always looking to get involved through volunteering.

Kemar Harris

Kemar Harris is a third year Computer Science student specializing in Software Engineering at the University of Toronto. Kemar loves the architectural design of software, and appreciates systems design.

Kemar did a twelve-month internship as a Software Developer at the transportation company Thales, where he worked on the 4LM project, a train system in the United Kingdom. While at Thales, Kemar worked on tools such as a color-coded data visualizer for the train system logs, a virtual guideway that can be used for verifying train data, a testing environment for the train data, and several other automation tools.

Outside of school, Kemar has a deep love for game design. This evolved into him developing his own game which further honed his software engineering skills. Because of this, his strengths now include designing, writing, and maintaining software systems. He also learned the importance of writing code that is simple, yet easy to maintain. This project has inspired him to look into a career in the game design industry that allows him to combine his love for software design and architecture with his love for gaming.

Lan Yao

Lan Yao is a fifth year Computer Science student specializing in Software Engineering at the University of Toronto.

She worked as a front end developer at Caseware, a fintech company specializing in accounting software. She self-taught Angular and developed components for an internal library used by six cross functional internal squads and more than a hundred companies. She implemented features for their external cloud product using AngularJs framework that was released in October 2019. Other than development, she also assisted with facilitating user research tailored for an Agile environment and collected feedback from end users and distributors.

Outside academics, Lan is also busy with extracurriculars. She is the Vice President of Communication at UofTHacks VII, which is an annual hackathon hosted by the University of Toronto. She is also a developer in the web development team for a business case competition called LIVE. She and her team developed a platform called Blueboard used by participants to submit their work, executives to manage LIVE, and judges to submit their scorings of each team's performance for the five rounds.

In her spare time, Lan enjoys running and has participated in Scotiabank's half-marathon in 2018. She also likes yoga and traveling.

Seemin Syed

Seemin Syed is a third year Computer Science student specializing in Software Engineering at the University of Toronto.

She has previously worked as a QA Analyst and Database DevOps Analyst for multiple projects at a full-stack mobile application development company, on applications ranging from mental health to sports to vacation planning.

She has contributed to a Play Store Android application for an online novel translation community and volunteers as an editor for a translation group, working on weekly releases between reading, reviewing, and discussing novels translated by others.

Beyond literature, Seemin is also interested in TV shows that involve philosophy and technology such as *Black Mirror*, *The Good Place,* and *Haunting of Hill House*.
To her, the best conversations are the ones that are ripe for discussions about technology, humanity and how the two have evolved and will evolve in the future– especially since she intends to participate in creating that future!

Team Expectations Agreement

We accept the guidelines shared below and intend to fulfill them.

Digital Signatures:	Abel Debalkew Carl Alvares Jayatharani Thirugnanasampanthan Kemar Harris Lan Yao Seemin Syed

---

Communication

Methods of Communication

Slack will be the primary communication tool as the platform can be customized for our team's needs. Slack allows us to have all discussions in a central place, but still have them organized through the use of Slack channels. In addition, Slack enables us to integrate external apps which will be helpful for our project, such as Google Docs. Furthermore, Slack offers useful features that will help the team manage information, such as threads and pinning.

Google Hangouts will be used for discussions where text may not be sufficient.

It is recommended that members do not reach each other via email, phone, text, or messenger as the team would like most (if not all) discussions to be accessible to all members.

Communication Response Times

Members of FOSSkateers are expected to check Slack on a daily basis (including weekends and holidays). Members will read and respond to messages on Slack within 5 hours. In addition, members should be checking Slack between 10:00 am and 8:00 pm.

If a member is aware they will be unable to read or respond to messages in a timely manner due to special circumstances (e.g., test, vacation, illness), it is expected that they notify the team at least two days in advance (and four days before a deliverable is due). Having this notice will allow the the team to move the project forward in the team member's absence and/or make appropriate arrangements to accommodate the team member.

Meetings

Meeting Attendance

All in-person and online meetings are mandatory, though exceptions will be made for members who have a legitimate reason for not being able to attend. It is expected that in

such cases, members give advance notice so the team can attempt to reschedule the meeting at a time that works for everyone. Meetings will be held once a week either in-person on campus or online (details will be confirmed on a week-by-week basis).

Running Meetings

Meetings will be held once a week either in-person on campus or online (details will be confirmed on a week-by-week basis). Meetings will not take longer than an hour; having a set time limit for meetings will ensure the team is productive. In-person meetings will be scheduled between 3:30 and 4:30 pm on Fridays at IC. Online meetings will take place during the weekend (times to be determined).

All members will take turns recording meeting minutes using the Slack post feature and posting them in the “Meeting Notes” Slack channel by the end of the day. Attendance will be recorded in these meeting minutes.

Meeting Preparation

All members are required to discuss on Slack at least 24 hours before the scheduled meeting what topics to cover during the meeting to ensure the team is able to prepare themselves prior to the meeting (i.e., all required tasks must be completed) and is productive during this meeting. As team meetings will be capped at an hour, members are not expected to be coding during these meetings.

Version Control

Git (specifically, GitHub) will be used as it facilitates collaboration and the open-source project will be using this version control.

Committing Documentation

Documents which are part of a deliverable should be committed to the documents subfolder within that deliverable folder (e.g,. Deliverable 0/docs). All documents committed should be in PDF format.

Committing to Codebase

Code for a deliverable should be committed to the project subfolder within that deliverable folder (e.g,. code written for the second deliverable should be committed to Deliverable 2/project).

For code expected to be merged to the open-source project’s codebase, developers must adhere to the project’s guidelines (usually is linked on the project’s README.md). Consequently, developers should read guidelines carefully before starting their work and refer back when needed.

Unless specified by the open-source project’s guidelines, for the team’s codebase, a commit should only include changes to one specific feature. Developers should verify their code before and after every commit using test suites; verification must adhere to the standards of

the open-source project. Frequent committing is preferred so commits are relatively small and merging is manageable. A commit message should:

- be prefixed with the bug/feature issue number, if available (e.g., BUG #123)
- be no longer than 50 characters (excluding bug/feature issue number prefix)
- use proper punctuation and grammar
    - first character of message should be uppercase and message should be written in present tense for consistency
- describe the problem that the commit addresses

Sample commit message: BUG #123: Fix nullPointerException thrown in getMessages.

Generated artifacts should not be committed to the project; a gitignore file should be used to ensure generated artifacts are not committed. Developers should commit dependencies so developers do not need to install additional software to run the code.

Developers should avoid committing directly to the master branch to prevent broken code from being included in this branch. It is expected that new branches and pull-requests are made so that changes can be verified by another developer before being merged into the master branch. Branch names should be meaningful, preferably prefixed with the bug/feature issue number (i.e., names should give developers a sense of what kind of changes were made).

Division of Work

All work will be divided evenly. Members are expected to delegate tasks to themselves as soon as possible, preferably in a meeting setting so everyone is in attendance. Work may be completed in pairs if more than one member is interested in a specific task or would like to pair program.

The division of work will likely either be recorded on a task board (JIRA) or captured in the meeting minutes post[2]. It is everyone's responsibility to ensure their assigned tasks are recorded so that members are able to complete peer evaluations accurately.

Members of FOSSkateers agree to perform the following roles:

- Abel Debalkew: review Carl Alvares' pull-requests and record meeting minutes
- Carl Alvares: review Abel Debalkew's pull-requests and record meeting minutes
- Jayatharani Thirugnanasampanthan: review Kemar Harris' pull-requests, record meeting minutes, and upload single document deliverables
- Kemar Harris: review Jayatharani Thirugnanasampanthan's pull-requests and record meeting minutes
- Lan Yao: review Seemin Syed's pull-requests and record meeting minutes
- Seemin Syed: review Lan Yao's pull-requests and record meeting minutes

Specific roles will be discussed later once the development process has been chosen.

[2] Specific tools depend on the development process chosen by the team.

Submitting Deliverables

All deliverables should be committed to GitHub at least three hours prior to the deadline to ensure that work has been submitted and is accessible to everyone for reviewing. Details on where to submit deliverables on GitHub can be found under the “Version Control” section of this agreement. Members are responsible for committing their delegated tasks.

In regards to documentation, if a single document is only required to be uploaded for a deliverable, Jayatharani Thirugnanasampanthan will be held responsible for uploading this document by the team’s submission deadline.

A team member’s code will undergo review by another team member using pull-requests. Reviewers should verify code using test suites and adhere to the project’s requirements. As discussed in the “Version Control” section of this agreement, once a pull-request has been verified, the reviewer will merge the changes to the master branch.

Contingency Planning

If a team member drops out of the course, it is expected that they notify the team, teaching assistant, and instructor via email as soon as possible. The member’s work will be distributed evenly among the remaining team members.

Team members agree to attend all meetings, though exceptions can be made if the member communicates to the team well in advance. When a member misses a meeting (due to sickness, other commitments, etc), it is expected that they catch up by reading the meetings minutes and follow-up with others. If a team member appears to be taking advantage of this policy or has missed at least two meetings (without providing legitimate reasons), it is understood by the team member that their lack of attendance will be reflected in the peer evaluations. Furthermore, the teaching assistant and instructor will be notified as soon as possible via email with the entire group copied.

It is expected that code borrowed online is properly cited. If a member fails to follow this guideline, they are committing academic dishonesty and so the team will escalate the matter to the teaching staff.

Lastly, all members must meet expectations as outlined by the open-source project and this agreement. In the event that a member does not adhere to such expectations, they will receive a warning by the rest of the team. If the member routinely does not meet the open-source project’s (or team’s) expectations, the matter will be brought to the attention of the teaching assistant and instructor for further discussion.

Conflict Resolution Planning

In case of disagreement or conflict within the team, it is expected that other members will mediate and ensure that discussion remains civil and respectful. Time during the weekly meetings will be dedicated to discuss the execution of the past week, and bring up any sore points or disagreements. It is imperative that the team factors in the opinion and perspective

of every team member so as to reach an appropriate compromise and have all voices heard. This way, the team will foster an environment where conflicts can be easily avoided. If a more serious situation arises, one that the team cannot handle themselves, the team will involve the teaching assistant and possibly even the instructor.