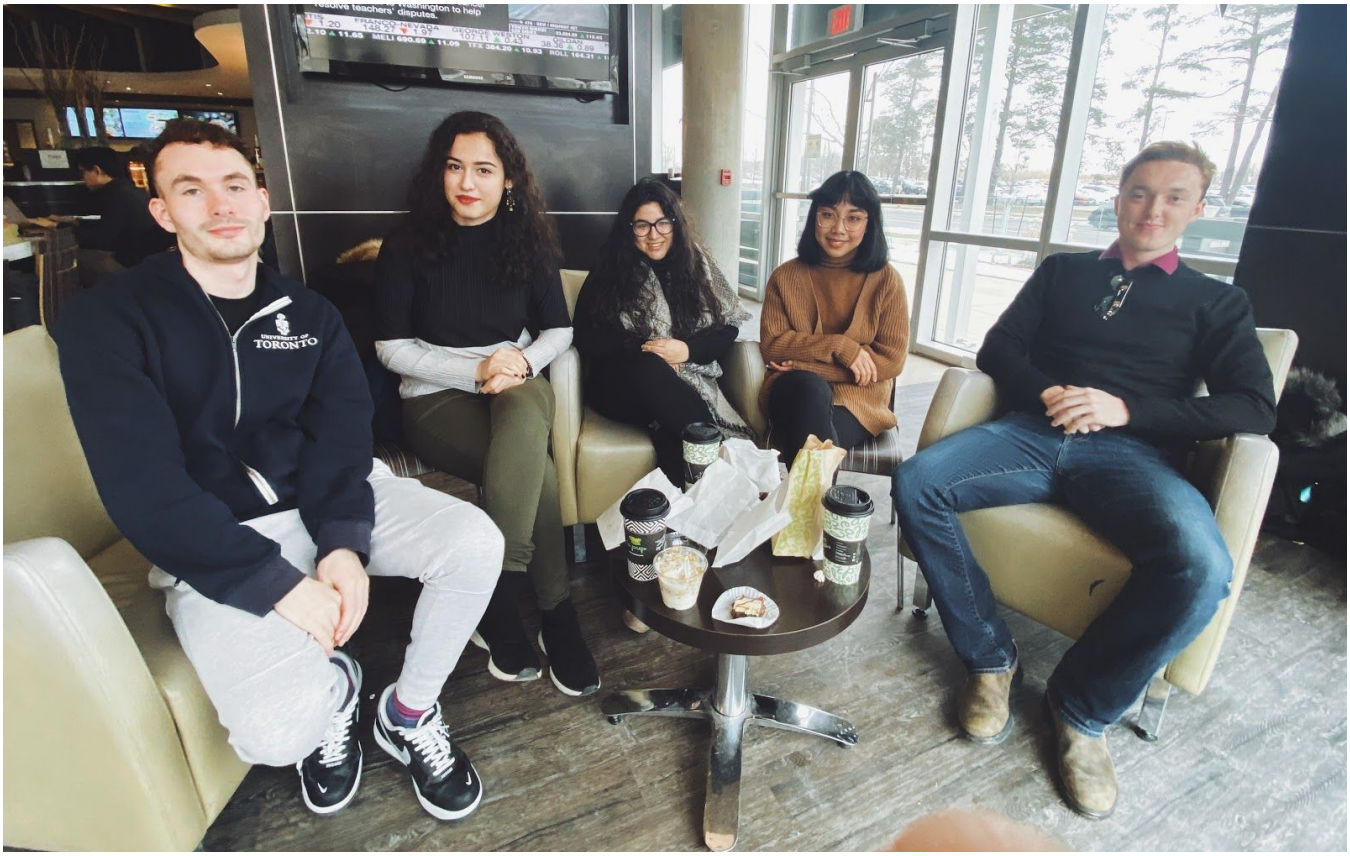
🧀 Mozzarella 🧀

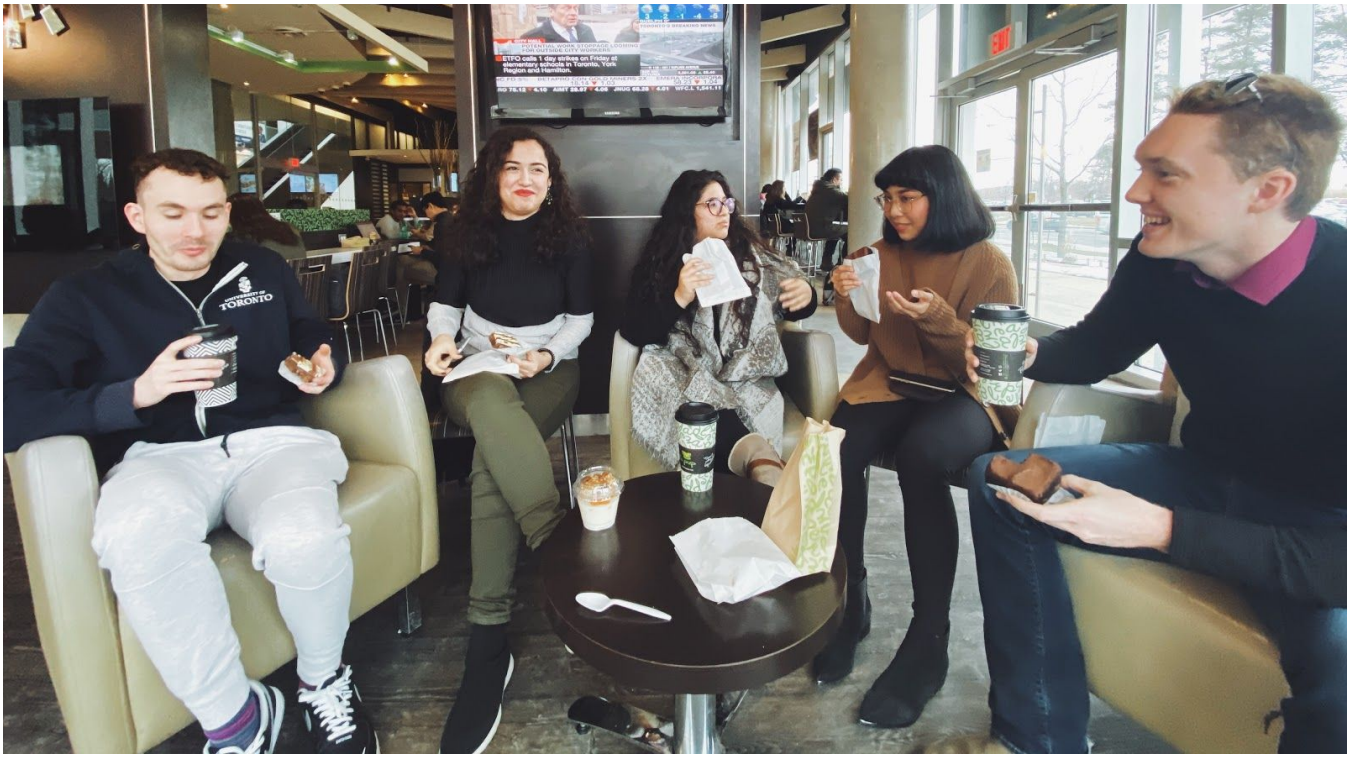
Introducing our Team

We are team Mozzarella! We are five Computer Science students who love constantly learning — and our goal this semester is to better ourselves as software engineers. Contributing to open-source software is a completely new undertaking for us. Most of us are only familiar with creating our own side projects, or our experiences in larger codebases during co-op terms. But we realize that some of the most important software that we use is open-source!



About Us

Our team has many strengths. We come from a diverse background of interests and expertise (some outside of tech). Some of us are experienced in web development, while all of us have dabbled with small projects of our own. As for coursework, we have a wide range of interests, from algorithms, pedagogy (most of us are TAs), machine learning, and artificial intelligence. We hope to bring out the best in each of our team members. Furthermore, most of us have had prior work experience in software development through internships. We've all seen terrible codebases, and definitely wrote some bad code before. We are prepared to apply what we know, but to also learn what we haven't yet mastered. We are hopeful about this project and are determined to make a meaningful contribution to our project of interest!



Introducing our Team Members

Joe Armitage

Joe is a Senior Year student in the Computer Science program, pursuing a Specialist in Software Engineering. When not writing code, learning computer science or teaching it, he likes to play tennis (as much as possible) and occasionally play a story-based RPG or FPS video game. His work experience is primarily as a teaching assistant for computer science and math, where he has taught tutorials in Computer Organization, Linear Algebra, Introduction to Programming, Software Engineering, and Functional Programming.

In previous personal and school projects, Joe has developed a taste and aptitude for developing single-page web applications preferring a Sequelize -> Express -> React stack for a RESTful API. His preferred programming language for simple tasks is Python, but he also deeply enjoys coding in Haskell as well as C, for the problems it makes sense to solve with these languages. His favourite topic in Computer Science is hardware organization, not just because of how much fun it is to teach, but also because it's actually physics at a low enough level, and that's cool!

After graduation, Joe plans to make a plan. Having so far worked solely in academia (in this field), Joe hopes to shop around and find the kind of development or education career he finds most fulfilling, with no rush. He's looking forward to seeing if open-source development tickles his fancy and if it may suit him as a career option; he's done a little before (a 2 line commit to small repo) and has always been a huge proponent and user of open software.

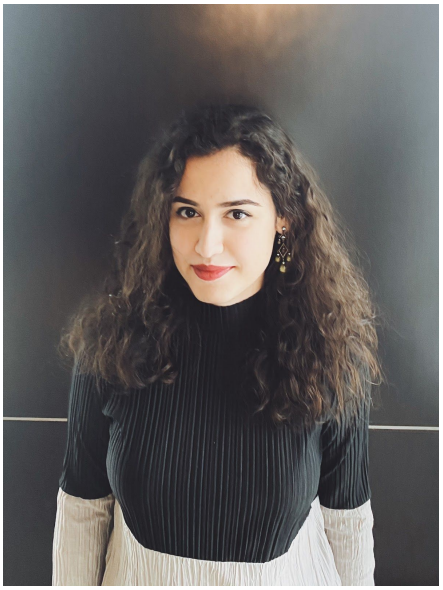
Jacob Chamberlain

Jacob is a Senior Year student in the Computer Science program, pursuing a Specialist in Software Engineering. He has completed two internships as a Junior Developer — the first being at the engineering company MCW, where his role was primarily to write unit tests. Jacob then completed another internship at Ultimate Software. He was initially a QA Engineer, and eventually transitioned into a software development role over the course of 8 months (His co-workers there forgot he was an intern and were surprised that he had to leave to finish school!) Jacob has a keen interest in Machine Learning, and Artificial Intelligence, as he considers them the more interesting topics in Computer Science.

Jacob has experience writing a variety of code in Java, Python, C, and C#. He's created RESTful web applications and has done Machine Learning projects in Python. After graduating, Jacob plans

(if everything falls into place) to become rich enough to start his company without worrying that he will bottom out. He looks forward to the experience of working on an open-source program, as he's never worked on one before!

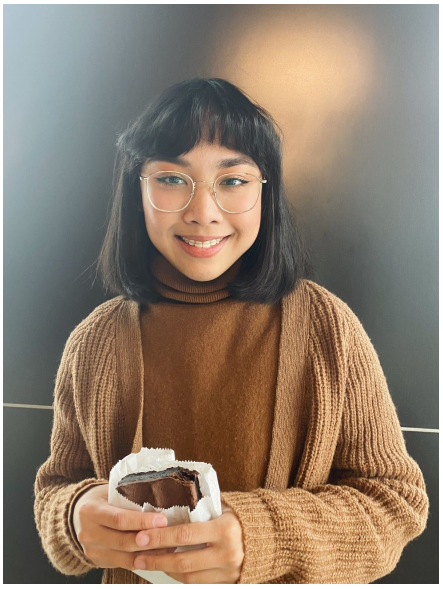
Saba Kiaei



Saba is a fourth-year student who is passionate about cybersecurity and ethics of computer science. Her most recent professional experience was an internship with AirMatrix where she worked on designing a new routing engine and full-stack development of a web application. She is also familiar with C, C++, JavaScript, Python, Java, Bash, Haskell, Node.js, Express.js, GraphQL, React, HTML, CSS, SQL, Android, Swift, Docker, Git, and Amazon Web Service.

In addition, Saba has been a teaching assistant for Introduction to Software Engineering and Numerical Analysis. However, Saba's favourite computer science topic is security, specifically, web security. During her free time, Saba mostly goes rock climbing, plays the piano, and skates. Saba is looking forward to learning how to contribute to an open-source project for the first time and continuing to do so in future opportunities. She is currently looking for a summer internship in security, and secure a job after graduation.

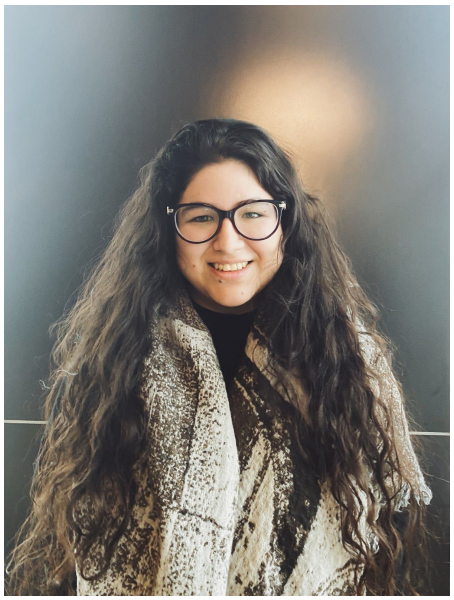
Fides Linga



Fides is a Senior Year student in the Computer Science program, pursuing a Specialist in Software Engineering. She has primarily worked in full-stack web development, most recently during an internship with TickTrade Systems. There, Fides helped with the existing FX Trading platform and has picked up a lot of knowledge about FX Trading. Fides has a keen interest in UX/UI Design and likes to think about how to improve all the tools she's using. In her spare time, Fides likes to make playful illustrations. She sometimes volunteers as a graphic designer for student organizations such as U of T Hacks.

Fides has used major web development frameworks such as Angular and React. She also has experience with Python, Java, and C, but mostly in academic settings. Her favourite Computer Science topics are anything related to algorithms and data structures. She finds that there's a lot to explore in these topics! After graduating, Fides plans on working as a software developer in the "tech" industry, preferably in consumer-facing products. She looks forward to learning more about the process of developing in an open-source project!

Angela Zavaleta



Angela is a Senior Year student pursuing a Specialist in Software Engineering Co-op and a Major in Mathematics. She did a 16-month internship as a Business Analyst for the Ontario Public Service, where she worked on various projects for economic ministries — such as the Ontario Business Programs Guide, and the Accessibility Compliance Report tracking system. Angela has also been a Teaching Assistant for various CMS courses (such as Computer Organization, and Calculus) since Fall 2016. Last summer, she took an interest in Computer Science education (pedagogical studies) and has been working on a couple of related research projects since then! This upcoming March, she will be presenting part of her research at a Computer Science education international conference (SIGCSE).

In her spare time, Angela likes to participate in Reddit Gifts, which are random gift exchange events held on Reddit. She is also a self-proclaimed Twitter addict and tennis fan. Angela has experience in Python, Java, and C, and she's currently learning some web development. After graduation, she hopes to get into graduate school to further research in Computer Science education. She has never contributed to an open-source project and she is excited that she gets the chance to learn more about how to do it.

Team Agreement

1. Methods of Communication

Most communications will be done on Slack. There will be a different channel for each topic of conversation (such as for different deliverables, meeting minutes, etc). There will also be a messenger group chat in case a person is unresponsive on Slack.

2. Communication Response Times

2.1 Messenger Communications

Within two days before the deadline, team members should respond within 4 hours. Otherwise, team members should check the messenger group chat at least once a day.

2.2 Slack Communications

Team members should check the Slack workspace at least once a day.

3. Meeting Attendance

3.1 Mandatory Meetings

Team members will have a weekly meeting (sometimes with the TA) every Tuesday 12 - 2 PM. The meetings may not take the entire 2 hours, but team members must be available throughout the 12 -2 PM time slot.

3.2 Non-mandatory Meetings

Team members will participate in non-mandatory meetings (meaning; attendance may be excused) on Fridays 10 AM - 11 AM, and Sundays 4 PM - 5 PM (to discuss finalizing the deliverable)

4. Running Meetings

4.1 Time and Location of Meetings:

Meetings (mandatory and otherwise) will occur on Tuesday 12 - 2 PM (IC402), Friday 10 - 11 AM and Sunday 4 PM (Phone call).

4.2 Meeting Minutes

Each team member will rotate being responsible for taking meeting minutes. The same team member is responsible for adding action items on the group's Trello board, which assigns every member's tasks for the deliverable.

5. Meeting Preparation

5.1 Meeting Agenda

Prior to all meetings, an agenda will be created which will outline different topics to discuss during the meeting. Notes will be taken with respect to this agenda. This agenda will outline who will be the leader of the meeting.

5.2 Meeting Leader

Each team member will rotate being the meeting leader. A meeting leader will ensure that everything in the meeting agenda has been discussed and addressed.

5.3 Responsibilities of Members

All team members are responsible for being able to discuss the following points in order during the meeting:

- Are there any blocking issues — something that you need that is getting in the way of getting your work done.
- What work you have completed between the last meeting, up to today's meeting.
- What work you are planning to do after this meeting.

6. Version Control

6.1 Repository

All version control of deliverables will be done with the git repository located at:
https://github.com/CSCD01/team_04-project

6.2 What to Commit

Any file that is required for a deliverable should be committed to the repository. Miscellaneous files such as binary files, meeting minutes, etc, should not be committed. If the deliverable contains code, then others should be able to compile it. If the deliverable contains a written document, then source files for the word processor (e.g. Latex) can be committed, as well as generated PDF versions of the document.

6.3 Content of Log Messages

Commit messages will contain a short description of the changes made to the repository.

6.4 Code Reviews

When one team member is finished with their task, then two other team members must review and approve the code changes before their commits can be merged. If commits are reviewed but not approved, then the team member who was responsible for the task will make the necessary changes as required for the task.

6.5 Branching Strategy

Team members will adhere to the following rules:

- Do not commit directly to the master branch.
- To begin working on the next deliverable, a new branch will be made for that deliverable. All work done for deliverables will be committed to their respective branches.
- After a successful code review, deliverable branches will be merged to master. This process begins with rebasing the branch to master, to keep the git history on the master branch clean.

7. Division of Work

7.1 How to Divide Work

The work will be broken down into tasks in a meeting. The tasks will be prioritized, assigned a certain amount of time, and will be assigned to a person in the meeting. Each task needs to be completed by the assigned team member.

7.2 Designating Work to Members

Each team member will express their preferences for available tasks. These preferences should be based on the skill set and expertise of team members.

8. Submitting Deliverables

8.1 When to Submit

Team members will strive to submit the completed deliverable 1-2 days prior to the due date (the earlier, the better). The process of submitting a deliverable involves merging the deliverable branch to the master branch.

8.2 Who will Submit

Team members will designate one person who will merge the deliverable branch to the master branch. This designation will be decided before or while working on the deliverable.

8.3 Reviewing the Submission

All team members should review the submission (but may be excused if they are not available), but one team member is specifically assigned to thoroughly review the deliverable. This assignment will be on a rotation basis.

9. Contingency Planning

9.1 If a Team Member Drops Out

As soon as possible after learning about this situation, one team member will reach out to the instructor, on behalf of the group, to notify them of the situation. There will be a meeting to discuss how to redistribute the work allocated to the missing team member to the rest of the members. Ideally, this meeting will be held at most one or two days after learning that a team member has dropped the course.

9.2 If a Team Member Consistently Misses Meetings

It is the responsibility of team members to know when and where meetings occur. If a team member suddenly cannot attend a meeting, it is their responsibility to notify the team as early as possible, so that the meeting time/location can be adjusted to better accommodate everyone (if possible). In some circumstances, a team member can be excused from attending a meeting.

If a team member misses meetings without letting the group know, then the team will try to address the situation by reaching out to that team member. The team will let that team member know what is expected of them, and that team member is responsible for notifying the team of what to expect in similar future situations.

If a team member consistently misses meetings without letting the group know, then one team member will reach out to the instructor, on behalf of the group, to notify them of the situation.

9.3 If a Team Member Cannot Finish Their Work

It is the responsibility of team members to notify the group as early as possible if they are not confident they can finish their tasks. Once the team knows that a team member cannot finish their work, then that work will be delegated to other members.

If a team member does not finish their work on time, then the team will try to address the situation by reaching out to that team member. Similarly to section 9.2, expectations will be made within the team. Should the team member consistently not be able to finish their work, then another team member will reach out to the instructor, on behalf of the group, to notify them of the situation.

9.4 If a Team Member is Academically Dishonest

One team member, on behalf of the group, will make an appointment with the instructor to discuss next steps, up to and including kicking the offending team member off the team.

Contract Agreement

Team members of Mozzarella hereby agree to the terms and requirements outlined above.

Name	**Signatures**	**Date**
Joe Armitage		2020/02/04
Jacob Chamberlain		2020/02/04
Saba Kiaei		Feb 4, 2020
Fides Linga		Feb 4, 2020
Angela Zavaleta		Feb 4th, 2020