TEAM DATA

04.02.2020

—

DHARMIK SHAH
YUN JIE (JEFFREY) LI
JEFFREY SO
ALVIN TANG
JIMMY PANG

TABLE OF CONTENTS

INTRODUCTION

In the CSCD01 Winter class of 2020, Team Data will be working to contribute to the open source project **VSCode** (hopefully)**.**

The goals of the team are to work on a big existing feature, or create a new one that will improve the lives of developers. As we use VSCode on a daily basis, we would love to have this merged into the master branch, and that would make us really proud of our work.

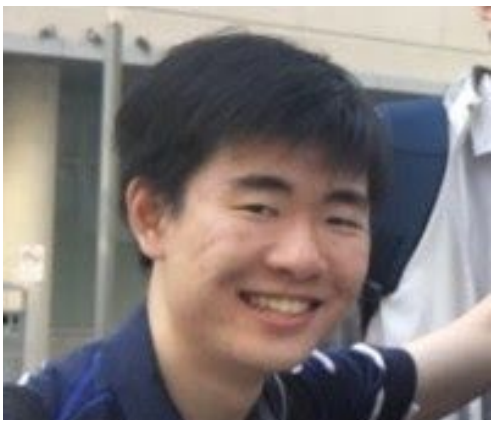
The team's strengths are mainly centered around Web Development, and since VSCode runs mainly on Typescript, it fits right under our toolbelt. Other team strengths include that we have all worked in the real world, and have practical experience with version control systems, writing proper code, and working in teams. Moreover, we all have practise in agile development, and have worked with APIs and large amounts of data in the past. Thus, understanding the code base should not be a very difficult task.

We will use these strengths to do our best in CSCD01 and contribute something meaningful to the VSCode repo.

MEMBERS

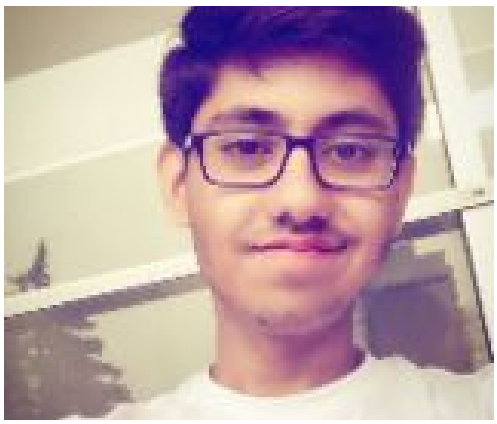
ALVIN TANG
Computer Science - Software Engineering Stream
alvin.tang@mail.utoronto.ca / 647-785-8122

Hello! I am a fourth year Computer Science student specializing in the software engineering stream. Before starting university, I had a keen interest in web development which has propelled me to gain the experiences necessary to place me where I am today. These experiences include my co-op work terms at Public Services and Procurement Canada as a Web Developer, and OpenText as a Software Developer, respectively. I have also participated in clubs at the university, most notably as a Vice President of Information Technology at LIVE Competition where I led a team of eight developers. With the combined experience, I have worked with over 10+ frontend and backend frameworks, with experience in CI/CD, containerization (e.g. Kubernetes), and cloud services. I look forward to gaining more experience with software development team processes and contributing to the open source community that I usually rely on.



JEFFREY LI
Computer Science - Software Engineering Stream
yunjie.li@mail.utoronto.ca / 647-210-9608

I am a fourth year undergraduate in software engineering. I have always had an interest in computer science since high school, having developed Windows applications and games in Visual Basic, participated in coding competitions and hackathons. I am proficient in Python, Java, and C#. From my previous DevOps experience at CaseWare, I have gained some basic understanding of build automation and CI/CD pipelines. As a DevOps Developer, I took part in the development of an automated build pipeline for CaseWare Cloud's servers using Jenkins and Ansible. I took part in scrum meetings and pair coding sessions. I really enjoyed the idea of developing a system to help put forward new versions of a widely-used online platform.



DHARMIK SHAH
Computer Science - Software Engineering Stream
dharmik.shah@mail.utoronto.ca / 647-517-2457

I'm currently in my 4th year studying Computer Science at UTSC, and I have always had the self-motivation to learn new technologies on the web. I was able to learn the web suite and develop my own website, and develop numerous other self projects. I was employed as a Software Development Intern at Index Exchange, where I worked on tasks using Golang, Javascript, and really around the full stack. We worked in a small team to build a product used by many Ad Exchanges in the world. We followed an agile methodology when developing software, so I'm sure I can take all I have learned over the years, and real world experience, to help in the development of the task we are about to do.

JEFFREY SO
Computer Science - Software Engineering Stream
jeffsh.so@mail.utoronto.ca / 647-927-3637

I'm in my 4th year undergrad studying Software Engineering at UTSC and have always had a passion for computers and motivation to learn how to build my own apps. I have developed my foundation in automation through bash scripting and Ansible from my employment at CI Financial and IFDS, along with Web Development, with environmental exposure to Full Stack Development at Fidelity. Utilizing agile methodology and additional work experiences that I have learned under my belt, I'm confident in my ability to work with my team in an efficient manner to produce a quality product.

JIMMY PANG
Computer Science - Software Engineering Stream
jimmy.pang@mail.utoronto.ca / 416-825-1623

I am a 4th year student at UTSC studying Computer Science and specializing in software engineering. I am very passionate about backend design and frontend integration. I have a strong foundation in front-end and web development from my experience at Caseware International and SCI Marketview. I have experience in Java, Python, C, as well as frameworks such as Angular and React, as well as agile methodology and teamwork skills. I am very confident in working together with the team to efficiently create a good product.

OVERALL TEAM STRENGTHS

1. Real world developmental experience and exposure to agile/DevOps practices, as well as a drive to learn new technologies to perfect the product
2. Strong in Python, front-end web development, and UI/UX design
3. Able to work with API's effectively to gather data and reformat as appropriate
4. Able to work with large sections of code efficiently and under time constraints

SHARING A MEAL

TEAM AGREEMENT

The following is a list of guidelines that each member in the team should abide to. It has been agreed upon by each individual and as such, should be used as a cheat sheet if one ever forgets their role in the project. Here is a list of the tasks/engagements we deemed necessary to include here, and their suggested solutions.

Methods of Communication

Phone calls

- Major emergencies

Messenger

- Minor emergencies
- Reminding of group meeting / important contact

Discord

- Regular team communication
- GitHub push updates
- Meeting planning
- Stand-ups

Communication Response Times

Phone calls

Almost immediately. If a person misses a call, they are expected to call back or respond on messenger immediately.

Messenger

Within an hour.

Discord

Within 24 hours. During pull request reviews, the reviewer and the author should respond within 6 hours.

Meetings

Stand-up Meetings

Stand-up meetings through discord allow us to discuss our progress and where we're stuck. Certain meetings can be missed if a member notifies the team at least 12 hours in advance but for the most part it is mandatory. A team member can miss up to 5 meetings during the whole project. Meeting minutes are not necessary for these types of meetings and nothing needs to be prepared. The following times are planned but can easily be adjusted on a week-to-week basis:

- Sunday from 6pm to 6:15pm
- Wednesday from 11am to 11:15am

Weekly Meetings

These in-person meetings at the fourth floor lab in IC will allow us to discuss plans for the next sprint cycle. These meetings are mandatory and a member must notify the team 30 minutes in advance if they plan to come late. If a member cannot attend, they must notify the team at least 24 hours in advance and read the meeting minutes by the following day. Meeting minutes will be written by one member each meeting, with another member within that role the next meeting. In regards to the preparation of these meetings, each member must write down what to discuss in the meeting and provide it to the team meeting leader, Dharmik, so he can compile an agenda a day before the meeting. The following time is planned and cannot be adjusted unless the adjustment is agreed upon by every individual moving forward:

- Friday from 3pm to 3:30pm

Teaching Assistant Meetings

To be determined.

Version Control

- For each new feature or for even minor bugs, create your own branch from the master branch
    - Should be named <your_initials>/<task_#>
- Commit messages should use this template (not strict):
    - <your_initals>/<task_#>: <Added/Removed/Modified> <Body>
- The master branch should always contain working code. If it isn't, the member with the commit that broke the code must revert their work.
- Work should not be directly merged to master, instead the member should utilize the pull request feature and assign the appropriate individuals to review the work
- For anything that should not be committed (e.g. IDE specific files), should be added to a .gitignore file

Division of Work

Get your task done on time. We will discuss as a team to separate the tasks but for the most part, new tasks are assigned based on each member's current workload.

Submitting Work

- Commit to your feature branch everytime a new chunk of logic is added, removed, or modified
- All working code should be merged to master at least two days in advance of a deadline, and the master codebase should be tested and reviewed by Alvin
- Everyone should collectively merge their work to master as time goes by, with each merge reviewed properly by at least one member of a team

Contingency Planning

- If a team member drops out, his workload is distributed with the remaining members
- If a team member is sick for a significant period of time, they will be assigned a small portion of the workload (only if and only if they are able to accomplish it) and distribute the rest of the work with the other members unless someone volunteers to take over the rest of the work
- If a team member is unable to complete some work, they must let everyone know so that all members can plan around it
- If a team member consistently misses meetings or does not participate to try to complete their fair share of the work, the team will talk to them as a group. If it can not be resolved after two warnings, then the TA will be notified
- If a team member is academically dishonest, we will talk to the TA and professor to see how the team can proceed