Computer Science 4474B: Team Profile and Topic Proposal

Group 02

January 20, 2020

1 Team Profile

1.1 Jiarui (Jordan) Yang

I am a fourth-year Specialization in Computer Science (minor in math) student. I have studied languages in Java, Python, C, and C++. In the past, I took nearly all the required CS courses, CS1026/1027, CS2211/2209/2214/2208/2212, CS3305/3307/3331/3342. My favorite classes were any courses that involved with team projects, such as CS2212 and 3307. Additionally, I had one year of internship experience in the science internship program. My team and I were building a banking fraud detection system using Hadoop, Elasticsearch, and other big data tools.

1.2 Dhrumil (Drew) Barot

I am a third-year Honors Specialization in Computer Science student. I have experience working with Python, Java, C/C++, and JavaScript+HTML/CSS. I also have experience working with High Performance/Distributed Computing (Hadoop), Deep Learning (TensorFlow, Caffe), and Computer Vision (OpenCV). In the past, I have taken several Computer Science courses, including project-based ones such as CS2212 and CS3307. Currently, I am taking CS33040/3342/3350. Over the past two summers, I have applied my Computer Science knowledge through internships in the financial markets, using languages like C++ and Python to conduct trading. I have also competed in several hackathons, applying my skills in a shortened time frame.

1.3 Ben Tilden

I am a third-year Honors Specialization Computer Science Student. So far, I have completed a numerous amount of CS courses at Western, including but not limited to CS1026/1027, CS2210/2214/2212, and CS3305/3307/3357. Languages that I have gained experience from these courses include Python, Java, C, C++, and SQL. During past summers, at internships, I became comfortable with the software development process working with C# and databases. Furthermore, I have pursued online education to learn about front end developer, languages such as HTML, CSS, and JavaScript and frameworks like Bootstrap and React.

1.4 Seif Elmolla

I am a fourth-year Computer science and business student. I have experience in numerous programming languages such as C, C++, Java, and Python. Additionally, I have experience working with applications used in CS 2210/2211/2212/2214, and CS3305/3350/3319. This includes the use of databases such as SQL and computer architecture. Throughout my internships, I was able to apply all the concepts taught in school to further my understanding of the technology sector. Combining my business experience in start-ups, I was able to understand user-centered designs better to give users a better overall experience.

1.5 Ryley Reid

I am a fourth-year Computer Science student. I have experience in numerous programming languages such as Python, Java, and C++. I have taken several CS courses in the past, including CS2212/2210/3307, amongst others. I also have experience with hardware projects and image processing. I have competed in various hackathons as well, applying my skills in a shortened time frame.

1.6 Wayne (Jack) Maysuik

I am a fourth-year Honours Specialization Computer Science student. I have experience in numerous programming languages such as Python, Javascript, and Java. Past courses such as CS 2212, CS 3307, and CS 4480 have familiarized me with project management and hitting progress-driven deadlines. I have also taken all of the required CS courses, providing me with a strong foundation in the fundamentals of CS. At work, I am regularly responsible for creating or modifying components in our web app using React and TypeScript. At past jobs, I gained experience using .NET and C# to develop internal web apps for employees to manage their data.

2 Application Proposal - Improved Chat System

2.1 Technology Stack

Our group was comfortable using Python (v3.7) due to its wide array of open-source libraries and ease of use. For our front-end, various members are satisfied with JavaScript (Node/React) to implement a GUI. For a database solution, SQL seemed to be a comfortable and viable option.

2.2 Project Scope

Current Internet chat systems have not evolved far from their early peers. Systems still require proprietary logins and are inherently complex to use due to dependencies and social graphs. We aim to build a simplified messaging service that focuses on ease of use for all users and places emphasis on security.

Our basis for this redesigned application is Facebook Messenger, which was released in 2011 and had approximately 1.3bn monthly active users. The chat system requires a Facebook account and connection with "Friends" on Facebook for many features. While this is adequate for 1-on-1 chats, the process of creating and managing group chats is relatively cumbersome.

In our modified system, we would allow users to create chat rooms through a straightforward interface, and share URLs or QR codes to allow other members to join. Chat rooms may have an admin or may not. To access the chats, a simple chat-specific login could be created, which would be stored locally within the chat object, allowing for complete independence. Useful features like spam filters, user customizability, file upload, and chatbots would enhance the user's experience with the system. Many principles of Human-Computer Interaction may be considered and implemented as needed to improve the overall experience as well.

2.3 Core Features

2.3.1 Disposable Sign-Up

The ability to enter the chat as a guest or to create a chat-specific ID. Facebook requires users to have a Facebook account before being granted access to use their messenger application.

2.3.2 Chat-local User Accounts

Allow users to register for a chat with email and a password, not requiring any proprietary information. No permanent accounts need to be made. To retain login access, a user can optionally create an account with a username and password.

2.3.3 Unique Chat Rooms

A chat room can be made simply by entering a name and configuring some basic settings. Instead of creating a group then inviting specific members (only members that are your Facebook friend can be invited), a user can create a chat room with a unique code that will act as the key for anyone to join the group.

2.3.4 Chat Connect Identifier

Adding on to the idea from above, the group creator can either invite users with a unique key, URL, or QR code.

2.3.5 Chat Merge

Instead of having to create a new group to merge two groups as you do on Facebook, our application will have a master merge option.

2.3.6 Auto-Spam Filter

Use machine-learning techniques to classify messages as spam, and block if necessary. Unlike Facebook, our application will limit data usage, security threats, and our server space by including an auto-spam filter for chats.

2.3.7 End-to-end Encryption

Implement encryption protocols so messages can be sent securely. This feature will be similar to Facebook's Messenger app to ensure the security of messages being sent.

2.3.8 End-to-end Encryption

Implement encryption protocols so messages can be sent securely. This feature will be similar to Facebook's Messenger app to ensure the security of messages being sent.

2.3.9 Simple Callable Chatbots

Using the @ symbol, users can direct meeting times, locations, and the essential information to bots in the chat room server, such as event reminder bots.

2.3.10 Current Active Users

This feature will be similar to Facebook's Messenger app in which users will be able to see the status of the activity of other users in the chat room.

2.3.11 Support for Multimedia

This feature will be similar to Messenger's ability to send files under a set file size to all chat users (this file will be held in the chat itself).

2.3.12 Allow Configurable Chats

Allow various administrator and privacy options to conversations, similar to Messenger. Allow admin-controlled, uncontrolled, private, closed, and open chats.

2.3.13 User Customizability

Allow unique usernames, profile photos, and support rich text. Similar to Facebook's Messenger, we will incorporate customizability in usernames and profile photos for users with and without accounts.