University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 1
Date: October 6th, 2023
Group Number: 68

Name	Student Number	CS ID	Email
Liam Buchan	75087510	s0d7t	liamjamesbuchan@gmail.com
Eric Lee	44415206	g4g9q	elee1820@gmail.com
Sammi Pau	56010564	q7d9h	sammi.pau@live.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Project Description

Domain

Data archiving and indexing.

What aspects are modeled by the database?

The project aims to help users index and archive data through the bittorrent protocol. To achieve this, the service allows users to upload bittorrent files which point to the data itself along with metadata and tagging that allows other users to easily find the files. For example, a user could upload an Arch Linux ISO torrent, tag it with the “Arch” tag and place it within the “Linux ISO” category.

Database Specifications

Functionality

The database will store the paths of files and their appropriate metadata, user information, roles, comments, requests. People will be using the database to login, find torrents via category and tags, comment and vote on torrents, view their snatch history (what files they’ve downloaded), moderate torrents, and make requests for new torrents to be added.

Application Platform

Database

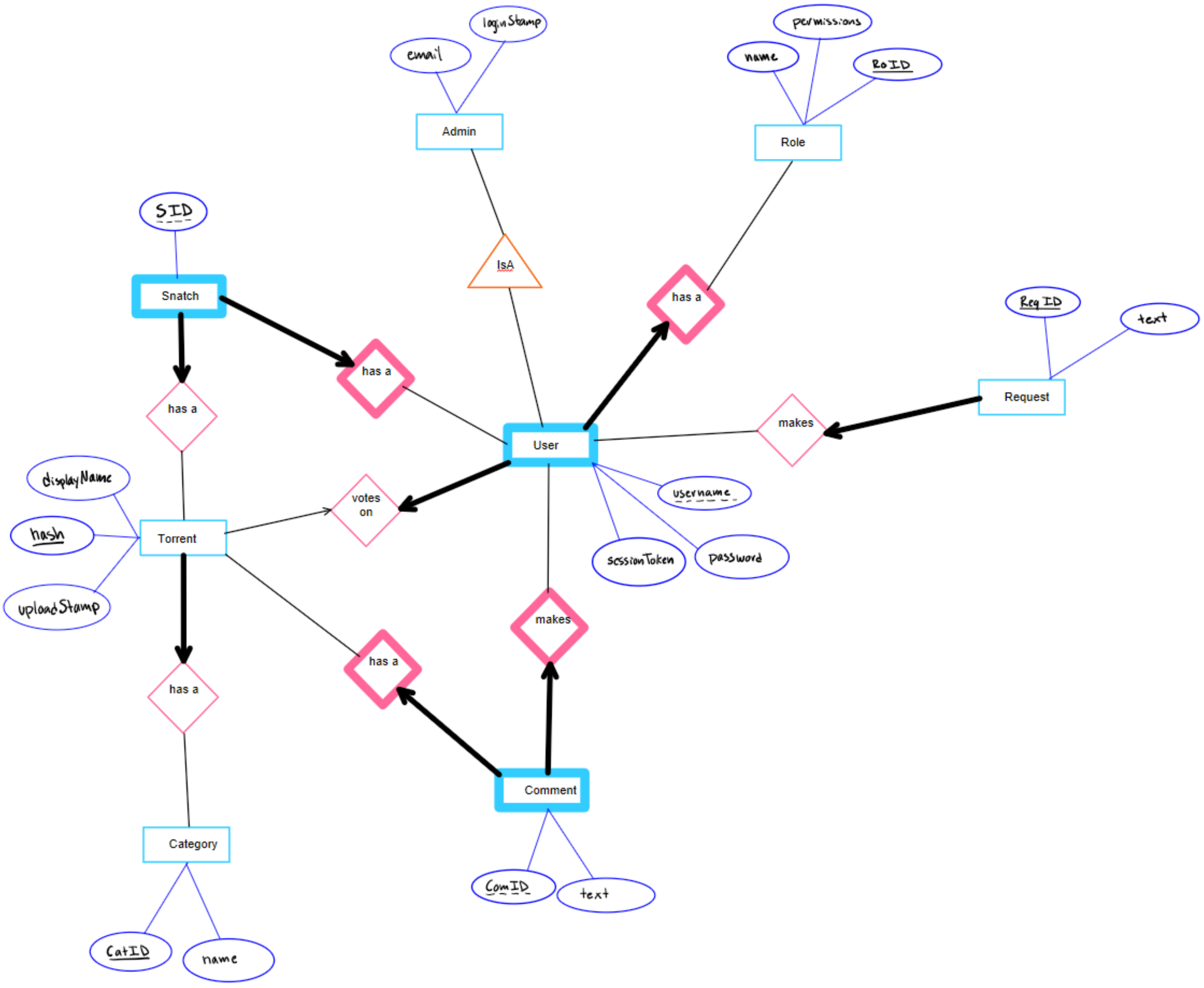
We will be using a PostgreSQL DBMS hosted on a personal machine. We will create a docker-compose file to make it easy to spin the service on any environment.

Expected tech stack

The backend will be a REST API written in Go. We expect to primarily use the standard library for the HTTP server and SQL interface, however we might use mux and sqlx which are third party libraries that minorly improve on the standard libraries functionality.

Our frontend will be written in TypeScript using the React framework.

ER Diagram


loginStamp
email
Admin
permissions
name
RoID
Role
SID
IsA
has a
Snatch
has a
has a
ReqID
text
Request
makes
User
displayName
votes on
username
hash
Torrent
sessionToken
password
uploadStamp
makes
has a
has a
Comment
Category
ComID
text
CatID
name