

Rachel Chen

CS 601 – Fall 2020

Project Requirements & Timeline

Rent a Minecraft Server

The project aims to make renting a Minecraft server easier and to let people connect in the world of COVID through the power of community via gaming. It should be affordable by billing users based on usage, and easy to use.

Core Features (MVP):

- Integrate with Stripe to charge user for spec of the Minecraft server (e.g. 6 players);
- Store only the user email, and password-less login by sending email with token to access control panel;
- Allow the user to start, shutdown, and restart the Minecraft server;
- Keep track of the usage time of the Minecraft server;
- Provision a Minecraft server upon successful payment;
- Health-check Minecraft server and restart if the server is down unexpectedly.

Optional Features (vNext):

- Only charge the user based on the actual time the server was up (usage-based billing);
- Value-added package to allow users to run mods and upload custom worlds;
- Allow user to whitelist players and only allow them to connect to their server;
- Allow user to download their world in progress on the server;
- Dynamically purchase more server capacity (e.g. on DigitalOcean) when Minecraft server capacity available for rent is running out.

Languages & Tools:

- Project should be built with Golang, and data should be persisted to PostgreSQL;
- For MVP, the project should be running on the server that also runs the Minecraft servers;
- In the future, the project itself should communicate with server capacity via gRPC, and a local daemon should be running on each server, instead of running the project alongside with the project itself.

Timeline & Milestone

Milestone #1 (9/28): Setup database schema, and finish integration with Stripe to store user information

Milestone #2 (10/5): Provisioning Minecraft server and record usage time

Milestone #3 (10/12): Allow user control the status of the Minecraft server and health-check

Milestone #4 (10/19)-: (Optional features)