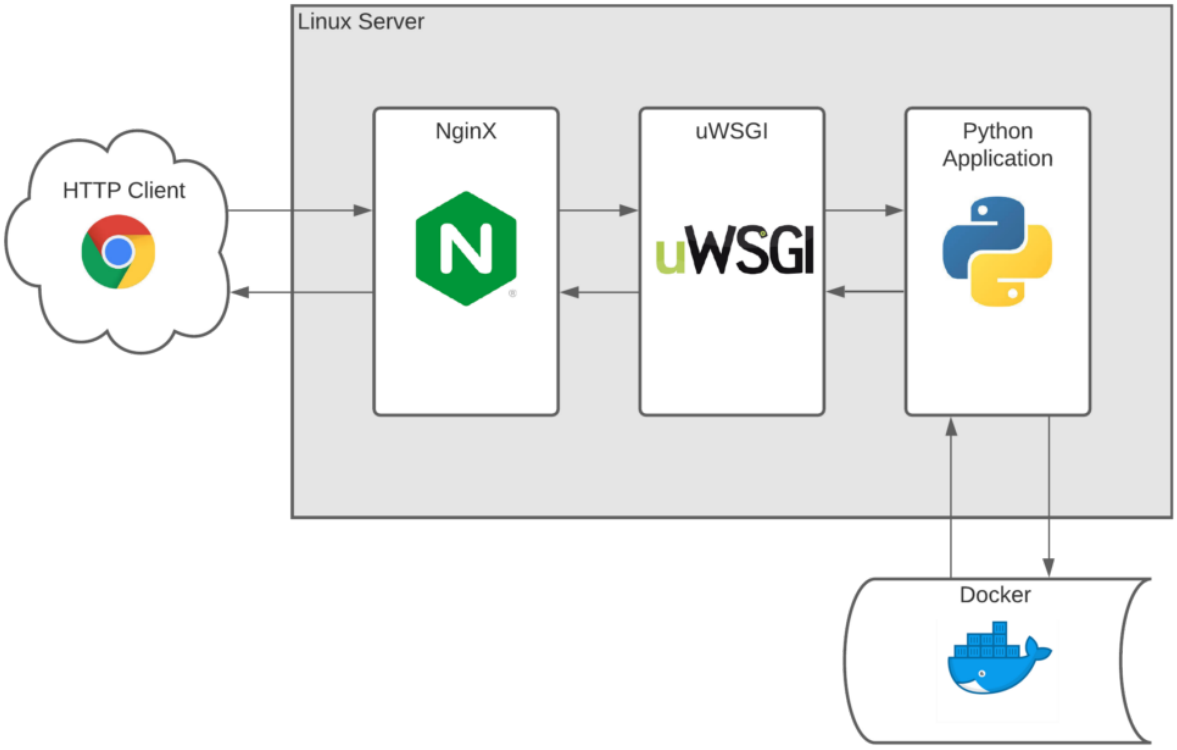
Group B - Kattis Problem Practice Tool - Design Document

Brendan Carr, Brooks Langley, Will Flanagan, Casey Stamper

Design Document

1. Architecture



2. Code Repository

Our project can be found on Github at this link: https://github.com/willflanagan1/523-kattis

3. Detailed Data Definitions

The /kattis/problems directory hosts the files that are wrapped up and mounted in a volume for Docker. The files within this directory must follow the Kattis Problem Package format. More details regarding this format can be found here.

For the problems that are being displayed to the users at the /problems endpoint, the database these problems are being read from contains the fields name, description, inputDesc, outputDesc, sampleIn, sampleOut. Each field represents one problem in the table of problems. There are not any restrictions on the fields as they can contain letters, numbers, and other characters. All fields are not necessary to have a problem, however, and problems without specified fields will be empty.

4. Design Rationale: Design Decisions

Our team entered an existing system infrastructure for our project. The location for where our project would live was already in place before we began development. A template of files and modules were also already in place. Because of this, our team did not need to spend significant amounts of time discussing and implementing a system design. We did have control over smaller aspects of our application. For example, our application was to connect to a Docker container. We decided as a team it would make sense to us a connection that integrated with Python so we could export its functionality across other areas of the project. The Docker SDK for Python became the clear design choice for this endeavor. Another example was how to manage the Kattis Problem Package Format for student file submissions. We decided it unnecessary to keep storage of all of the user's files and instead track a score of how their submission file performed. We accomplished this by temporarily storing the submission file as it was needed and deleting it when it had no further use. As a result, the file system that the project used would stay lightweight and not get crowded with outdated data.