**Software Implementation and Testing Document**

**For**

**Group <21>**

Version 3.0

**Authors**:

Irfan Y

Gabriel R

Leonardo N

# Programming Languages (5 points)

Golang - we chose this language because of its inherent concurrency and for its highly performant compiled nature. We use it on the backend for building out our HTTP server infrastructure.

NextJS - we use this for the front end, and for our all our UI. We personally chose NextJS due to its ease of use, quick development time, and also server-side rendering capabilities. Our UI is not going to have many dynamic components since we believe it is important to take as much compute off the client as possible.

TypeScript - we went with TypeScript over plain JavaScript for type safety and a more clean scalable codebase. It also allows us to have types and speciic data types later on when we build out our ile upload and WebRTC ile transfer functionality.

HTML + CSS - these are foundational and had to be used. It is an easy markup language for building and styling a UI.

# Platforms, APIs, Databases, and other technologies used (5 points)

NextJS as the framework/platform for the front-end.

We are using the Digital Ocean VPS for hosting our Go HTTP server, PostgreSQL instance, object storage (DO Spaces) and frontend.

# Execution-based Functional Testing (10 points)

We had each member of our team go through the functional requirements on the RD and test that they were working as intended.

# Execution-based Non-Functional Testing (10 points)

We did test for non-functional requirements by timing how quick pages would load, and how quick responses could be sent back. We also made sure all of our endpoints and hosted databases were quick in their response and not experiencing error.

# Non-Execution-based Testing (10 points)

We made sure all of our code was syntactically correct and easy to understand/read. After a group review of our code base, we even went ahead and refactored the entire skeleton of the backend to make it easier to develop on and scale overtime.