# 2020-02-28 Biweekly Report

## Tasks completed:

### Front-end:

1. Design of Meeting environment has been improved.
    a. Chosen avatars for representing users.
    b. Meeting environment has been redesigned with more appropriate models.
2. Predetermined user gestures.
    a. Both presenter and attendees are able to select from a list of predetermined gestures performed by their avatars.
        i. Waving
        ii. Thumbs up
        iii. Standing
        iv. Sitting
        v. Thumbs down
3. Established connection between the headset to back-end server
    a. Through the use of `UnityWebRequest` API, users wearing the headset are now able to connect with the back-end server, which will be used to handle relevant meeting sessions.
4. Website for managing meeting session.
    a. The website has some basic features including:
        i. Create an account.
        ii. Log in and log out.
        iii. Start a meeting session (in progress).

### Back-end:

1. Migrate language to Rust:
    a. Compile-time strongly-typed systems programming language.
        i. More suitable for concurrency, filesystem-based operations, and better type-checking.
    b. Uses `actix-web` as the web server framework.
    c. Uses `serde` for JSON serialization/de-serialization.
    d. Implemented *configuration* component:
        i. Allows configuring the server via `config.toml` located at project root.
    e. Implemented *logging*:
        i. Various logging levels, can be changed via configuration file or environment variable.
2. Begin API documentation:
    a. Deployed to GitHub pages, viewable as a webpage live at https://jieyouxu.github.io/VRME-Server-Rust/.
3. Design interactions between the various subsystems:
    a. (1) Account and authentication, (2) meeting session management, (3) view state signalling

## Problems and difficulties faced:

1. In order for the avatars to perform predetermined gestures, movement animations must be created. This could be achieved by importing third party animations. However, many of the existing animations are limited, often incompatible with our avatar model. Therefore it is more appropriate to create the required animations from scratch. Unfortunately, unity is very limited in this area, so we used another 3D graphic

processing software called Blender which supports animation development. Incompatibility issues is resolved by importing our avatar model into the Blender and building the movement animations from there.

2. Voice chat integration (Photon voice)

## Tasks planned for the next two weeks:

### Front-end:

So far all developments had been done in presenter's perspective.

1. Cameras and UIs must be replicated and fitted onto the avatars of each attendee.
2. Implement PDF preview for all participants.
3. Website remains to be implemented with other features including:
    a. Handling meeting session.
    b. Upload/download presentation slides to the website.

### Back-end:

1. Account management and authentication (back-end)
    a. Account creation, access, update, delete
    b. Authentication via JSON Web Tokens (`JWT`)
2. Meeting session management
    a. Create, update (invite), terminate a meeting session
    b. Upload / update / download presentation slides for a meeting session (persistent file store)
3. Improve unit + integration test coverage
    a. Look into test coverage generation tool