Database

Enterprise Wireless Connectivity

A certificate is created when Wi-Fi is activated that ensure that the device it wants to connect to is safe for use and will not get hacked. This is a similar process to connecting a device to a non-local Wi-Fi hotspot, such as at Humber. When the Wi-Fi system acknowledges that the device is safe for use, it allows the device Wi-Fi access and full access to all features and data of the Ping-Pong application. The user's interaction with the application not only allows full interaction with the machine but also connectivity to the application's database without any acknowledgement of the backend aspects. The firmware on machine is to be set up as the only service running on the Pi which only sends and receives that from the Firebase account so it make the account which prevents any harmful data being sent across. So it makes the machine safe to be connected to the any Home or Private Wi-Fi.

Database Configuration

The database uses email and password verification in both the login and signup verification screens to give the system additional information about the unique credentials for each user. Once a new user creates a new account in the signup page, the database stores the information so that the user can conveniently access the information when they use the application any sequential times when logging in. The database is hosted by FireBase, a real time database that allows the application's data to get transferred back and forth instantaneously. The group decided on using FireBase as it provided an easy and convenient interface that allows the users to change their user passwords and email accounts whenever necessary. FireBase is synchronized with the Ping-Pong Machine in real time to allow for the machine to change it's launch characteristics, such as launch angle, launch interval and difficulty settings instantly. This allows for a convenient play experience for the user as they get to experience the changes made in the application immediately after they make them. The group decided on implementing FireStore into our databases as well to allow for storing data that has to be used over time to show the user their growth over time. With FireStore, we can create a requirement both through the application and the machine that the database would only be able to store the user's latest 30 play sessions. This allows the app receive the whole play data at a quicker rate as it we can just get the whole

collection soring a user's data in one go. It also gives the user a more accurate impression of their progress in using the machine, since they only need to look through their most recent sessions instead of their full history. Cloud FireStore and Real Time Database allow for permanent access to the data instead of needing to create extensive backend code for checking and updating the database data, as they do not implement SQL. Values sent to the database through to Cloud Firestore include the launch angle, time interval, difficulty and the timestamp of when the play was used, consisting of both the time and date.

Unit and Production Testing

The production testing the machine involves many phases like testing the PCB the components connected to the PCB the Mechanical fitting of the all the components and enclosures. But one of the initial testing we can perform for the machine is the it's connectivity with the database as it takes instructions from the app. For that we don't even need to connect all the hardware to the Pi. We can just, instead of running the firmware for controlling all the components using the data we get from the app. We can just use the same Firebase connectivity certificate for the machine to run a pseudo code that can read data from the Real Time Database and just print it as an output. To show that accurate data is being received by the machine.

Security Considerations

Whenever a user creates a password when signing up to the application for the first time, their password gets hashed when it gets stored into the database, getting reformed into a random series of characters. This process is done by the Authentication module of Firebase and even the app creators cannot look up a user's passwords and the Authentication is marked with a system generated User ID that prevents user's from getting into other user's data. Also None of the user's can view any login credentials from FireBase aside from their own. All of these procedures are done in an effort to prevent user's from hacking into accounts they wouldn't otherwise be permitted to use. Security-wise, FireBase does all the authentication on its own instead of having to create backend code to enable security and user safety. Firebase Authentication allows user's to change their passwords and emails associated to the accounts, so if they feel that their information is compromised they can move their accounts to completely different emails but still keep their data about play settings. We can also monitor the amount data passing through any user's interaction with the Firabase which is really small as it only sends some Strings and integers across each time. So if huge chunk of

data is being transmitted then it means, the security of he data has been compromised and we can close all traffic or traffic from certain accounts to prevent losing any sensitive data.

Grading for this Milestone:

/1 Participated: Online session
/4 Online demo
Database description (fits into poster template's results column along with mobile application description):
/1 Addresses conection to enterprise wireless (v.s. home/open Wi-Fi/hotspot)
/1 Database configuration mentioned
/1 Security considered
May replace printing/enclosure block in poster template:
/2 Unit and production testing considerations