**Backlog**

**Features:**

|  |  |  |
| --- | --- | --- |
|  | **Relative Weight** | **Priority** |
| 1.Server - Motion data entered through a motion sensor | 1 | Medium |
| 2.Server - Read all the detected motion | 1 | Medium |
| 3.Server - Classify the motions into long and short | 2 | High |
| 4.Server - Upload the motion data into Firebase in real-time in order to store them | 2 | Medium |
| 5.Client - Find matches for motion signal by using Morse code coding table | 3 | High |
| 6.Client - Decode the motion sensor messages to be decoded into English message | 1 | low |
| 1. Client - Print out the decoded messages | 2 | Medium |
| 9.Client - Application can be started and stoped by users | 1 | low |

**Server:**

1.Motion data entered through a motion sensor:

The input data for the Morse code application are motion. A motion sensor will be used for the purpose of collecting motion data.

2. Read all the detected motion:

The server side of Morse code application will be able to read all the motion detected by the sensor.

3.Classify the motions into long and short

Server will determine the motion data into long and short motion. A motion longer that 5 second will be classified as longer motion, which will be represented as a string ‘1’. Whose short motions that are short than 5 second will be represented as ‘0’. Moreover the server will also determine different gaps between motions: Normal gap: about 1 second.

Gap ‘new\_letter’ : longer than 3 second and shorter than 7 second.

Gap ‘new\_word’: longer than 7 second

4.Upload the motion data into Firebase in real-time in order to store them.

After classifying each motion or gap, server will upload it to Firebase. It will be stored in Firebase In real - time

**Client:**

5&6. Find matches for motion signal by using Morse code coding table/Decode the motion sensor messages to be decoded into English letters

The web client can collect motions from Firebase. Then It will generate a motion sequence for a letter using gap ‘new\_letter’. After that, client compares sequence with the Morse code table given on the Moodle to get a letter

7.Print out the decoded messages

On the client side, It will print letters collected from step 5 and 6 and append them into an array called ‘Message’.It will print and append a ‘space’ when it read gap ‘new\_word’. The printing progress will be processed in real-time

1. Application can be started and stoped by users

User can start or terminate the application by click the checkbox ‘Accept signals’.

**User stories:**

1. As an app user, I want to enter data through a motion sensor.
2. As an app user, I want the server to listen to all motion sensor’s signals.
3. As an app user, I want to read the detected motion signals from the server console.
4. As an app user, I want the server can classify the motion into long and short.
5. As an app user, I wish that the server can upload the motion data into Firebase in real-time in order to store them.
6. As an app user, I want to the client can find matches for motion signal by using Morse code coding table.
7. As an app user, I want the client side to print out the decoded messages.
8. As a user, I want support for numeric values.
9. As a user, I want support for punctuation.
10. As a user, I want the client side to be able to start the transmission process.

11.As an app user, I want to decode the motion sensor messages to be decoded into English letters.