# Ride comfort mobile app

## Background:

The idea of this ride comfort app is to measure customer's ride quality when they make a journey in a vehicle. It is particularly interesting when it comes to quantify the ride quality of driverless vehicle. The existing measurement of the ride quality of a journey is using an expensive sensors, see picture below (figure 1 &2). It is essentially an accelerometer and gyroscope, but with much higher accuracy. However, nowadays the smartphone is well equipped with a good accelerometer and gyroscope; it could potentially achieve the required accuracy for the ride quality measurement.

- Sensor setup

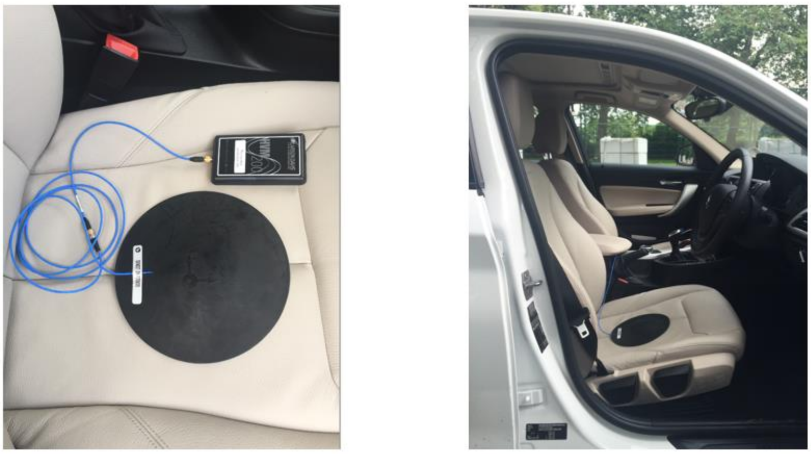


Figure 1: vibration sensor in the car

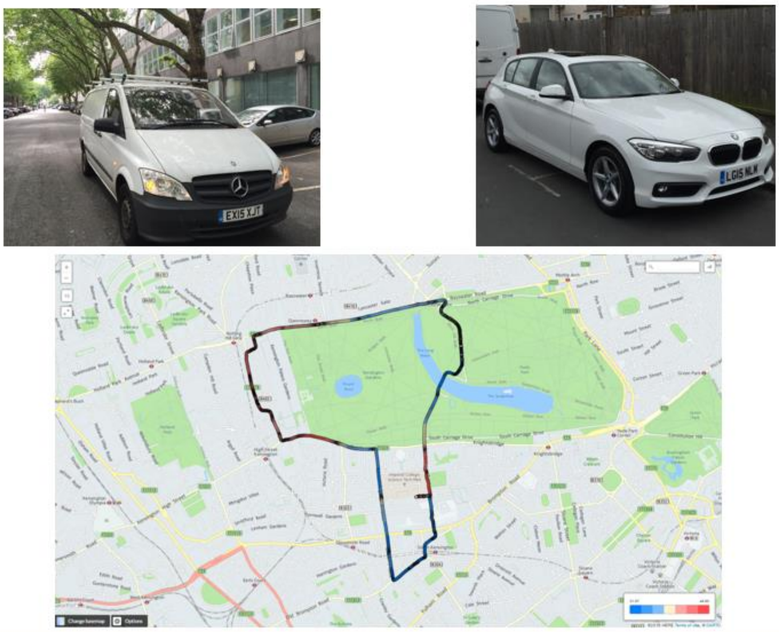


Figure 2: comparison of ride quality between car and van

## App Specs:

Functionality:

1. Get sensor raw readings from the accelerometer and gyroscope;
2. Calculate running RMS, minimum, maximum RMS for each axis;
3. Be able to use the device to collect data, view data;
4. Be able to adjust sample frequency;
5. Upload data to backend and user can request to send data via email.

Design:

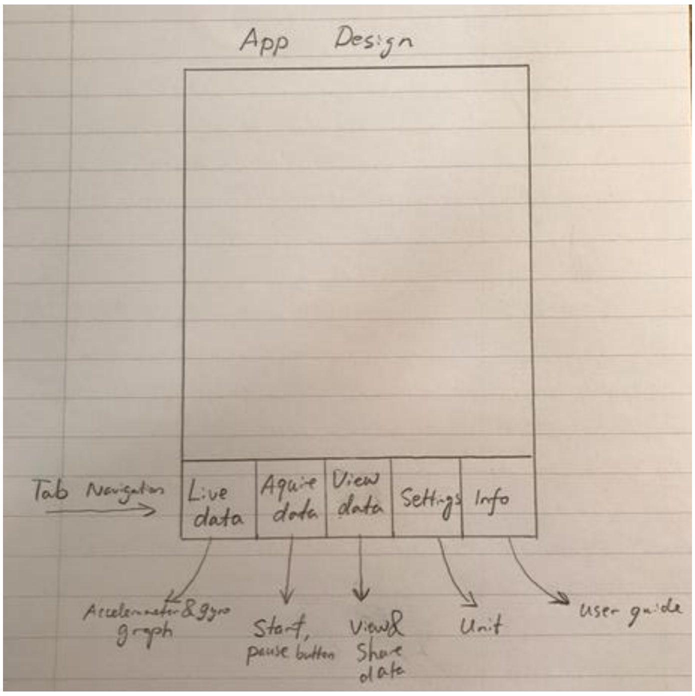
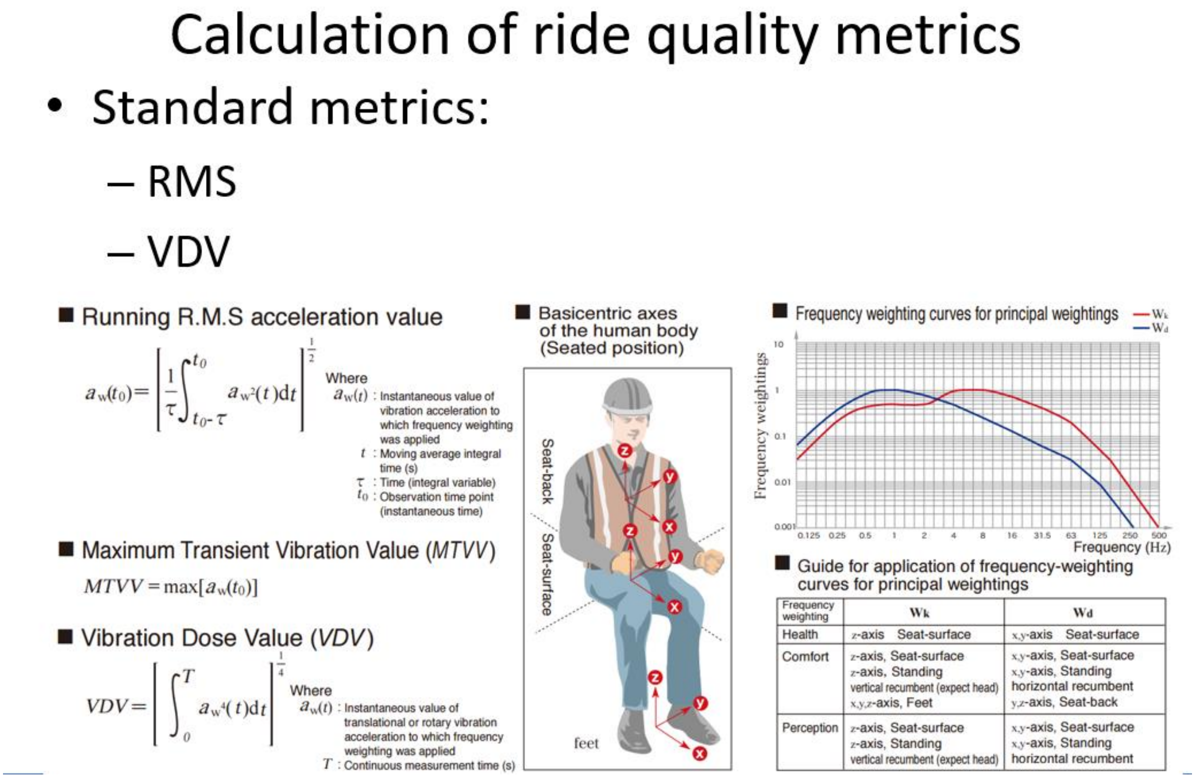



Figure 3: Brief overview of app layout

App name:

ICL-RideComfort

## Calculating ride quality metrics:




**Values needs to be calculated:**

X: running RMS, min RMS, max RMS

Y: running RMS, min RMS, max RMS

Z : running RMS, min RMS, max RMS

Overall RMS

**Example calculation:**

| | Accelerometer reading | | |
|---|---|---|---|
| Time | X-axis | Y-axis | Z-axis |
| 0 | 0.17158 | 0.205341 | 0.98495 |
| 0.1 | 0.097244 | 0.190006 | 0.998675 |
| 0.2 | 0.064279 | 0.151361 | 1.048559 |
| 0.3 | 0.065042 | 0.106226 | 1.036592 |
| 0.4 | 0.062876 | 0.054812 | 0.979403 |
| 0.5 | 0.121148 | 0.084047 | 1.010871 |

Running RMS:

$$RMS = \sqrt{\frac{a_t^2}{t - t_{t-1}}}$$

| | Accelerometer reading | | | Running RMS for each axis (m/s2) | | |
|---|---|---|---|---|---|---|
| Time | X-axis | Y-axis | Z-axis | RMS-X | RMS-Y | RMS-Z |
| 0 | 0.17158 | 0.205341 | 0.98495 | | | |
| 0.1 | 0.097244 | 0.190006 | *0.998675* | *0.30751253* | *0.600852* | *3.158088* |
| 0.2 | 0.064279 | 0.151361 | 1.048559 | *0.20326805* | *0.478646* | *3.315835* |
| 0.3 | 0.065042 | 0.106226 | 1.036592 | *0.20568086* | *0.335916* | *3.277992* |
| 0.4 | 0.062876 | 0.054812 | 0.979403 | *0.19883137* | *0.173331* | *3.097144* |
| 0.5 | 0.121148 | 0.084047 | 1.010871 | *0.38310361* | *0.26578* | *3.196655* |