Posture Alert

1st Suhan Muppavram
ECE Department
University of Auckland
Auckland, New Zealand
smup473@aucklanduni.ac.nz

1st Nipoon Patel
ECE Department
University of Auckland
Auckland, New Zealand
npat759@aucklanduni.ac.nz

2nd Muhammad Nadeem
ECE Department
University of Auckland
Auckland, New Zealand
muhammad.nadeem@auckland.ac.nz

***Abstract*—Desk-based office workers typically spend a high amount of time sitting in the chair every day and often in prolonged unbroken bouts. They do not always sit properly as they are not aware of sitting instructions and their sitting posture. Excessive time spent in a wrong seating position is a major source of health problem and a leading cause of pathological degeneration of the vertebral disc. In this paper, we present a smart chair solution to remedy these problems by analyzing the sitting posture of the person and keeping him informed about his posture. The chair sends a real-time alert to the user whenever a wrong sitting posture is detected for a prolonged period of time, resulting in posture improvement and reducing the risk of repetitive stress injuries (RSI). We offer a solution by applying the Internet of Things techniques to create an intelligent decision-making environment. By analyzing the pressure on different positions of a chair, we recognized different sitting postures. Real-time feedback is provided through an accompanying smart-phone application alerting the users to correct their body balance. The system also generates summaries of postures and the activities over a specified period of time. Finally, We conducted experiments to observe the response of the users sitting posture to the alerting feedback. The experiments demonstrated a classification accuracy of around 95% and a significant reduction in the time spent in the wrong posture.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

In modern lifestyle, an average office worker spends up to 75% of his or her day sitting, and more than half of that comes in long episodes of nearly inert sedentary [1]. In fact, it is found that a typical worker with a desk-based job, spends approximately 9.95 hours of a day sitting down. In general, most people spend 8.07 hours in a day sitting down when not working [2]. Sedentary lifestyles are slowly becoming more of a concern in modern life. The World Health Organisation quotes that 60 to 85 percent of the worlds population does not engage in enough physical activity. Research also shows that long periods of physical inactivity raise a risk of heart disease, diabetes, cancer and obesity. Moreover, poor sitting postures such as leaning forward causes upper limb and neck pain [3]. Posture refers to the alignment of your spine with all its adjacent structures. A person with good posture maintains proper alignment through all sitting, standing and lying positions. Bad posture in the form of slouching, hunching or slumping creates misalignment along the spinal column that disrupts nearly all the main components of the musculoskeletal system. The spine is amongst the most important parts of the body. If the damage is caused to the spinal cord, one can lose the ability to move any part of their body Fig. 1. The difference between a healthy and unhealthy spine is shown in Fig. 1. A healthy spine is shaped like the letter **J** whereas the unhealthy spine is shaped like a letter **S**. A **S** shaped spine is the result of prolonged periods of slouching, hunching or slumping. A misaligned spine causes weight or stress to be redistributed throughout the body causing the joints to undergo a significant amount of stress-compensating for poor posture. This results in long-term pain due to degradation of the supportive connective tissue leading to Osteoarthritis. A condition arising due to the severe deterioration of the connective tissue between joints.

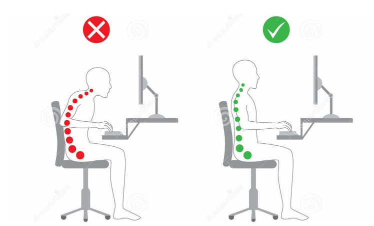


Fig. 1. Good vs poor posture [1]

These complications arising as a result of poor posture, such as RSI, not only affects personal life and productivity of an individual but also burdens the health system significantly. Globally, workers miss approximately 30 million days a year for these reasons and the US economy is said to lose approximately 28 billion dollars every year to his form of RSI [4]. Therefore, there is the pressing need for a system, apart from awareness, which could prevent the individuals from sitting in a wrong posture for a prolonged period of time. This will result in posture improvement and reducing the risk of repetitive stress injuries (RSI) that introduces the back pain. In this paper, we present a solution which use the Internet of Things techniques to solve this problem and create an intelligent decision-making environment. We address this issue by using a smart chair which is capable of recognizing different postures and alerts the user whenever a wrong posture is detected for a predefined period of time. The user receives an alert on the provided Android application which can be installed on a smartphone. The occupancy data, together with a time-stamp, is stored in the cloud which can be accessed by authorized users at any time and anywhere. The data can be used in various commercial and educational applications.

II. Related Work

A number of solutions have been proposed by the researchers to address the issue of posture recognition. The existing solutions are either not very effective or are too expensive to be affordable by an average user. The most important part of such a system is the sensing devices. In order to monitor the sitting posture accurately and reliably, sensors are needed around the users to get the sitting features. In the early phase of posture recognition research, visual information obtained from cameras was used to recognize the posture as it was a straightforward method demonstrated in the research presented by [5], [6]. The human skeleton is tracked using a depth camera, and sitting posture is recognized based on the relative positions among shoulder, hip, and knee. However, it always makes the users uncomfortable because it records other unnecessary information related to personal privacy. Also, the visual solution is easily compromised by the lighting level and would raise serious privacy concerns.

The expansion of this domain has lead to several mobile applications to monitor sitting and standing posture [7]. However, it is still difficult to realize continuous posture tracking and correction on the seat due to limited sensor capabilities on the smartphone. [8] uses the accelerometer readings from different spinal points by attaching smartphones to those points, and a web camera for detecting the upper body points location and distances to recognize the sitting posture. Not only was it highly intrusive in nature but also the size and the weight of the mobile devices made the user uncomfortable.

Several garment-based implementations have been proposed for posture measurement [9], [10], [12], [13]. This approach brings discomfort to users since the sensors are required to be attached to the body of the user. These solutions require a tight fit and a defined body-sensor alignment for accurate recordings bringing discomfort to the user. They also need expert-supervised evaluations who may disagree amongst them or could make errors when assessing training performance. Recently, wearable sensors have also been used to recognize the posture. [9] proposed a sensing garment to support posture coaching in children. It uses acceleration sensors embedded in the garment to measure the back-bending posture. The problem is that it does not alert the user and also requires human intervention. [10] implemented a small 4-inch long device that will be stuck to a user's lower back to help prevent slouching. Apart from being expensive, it is also intrusive and causes discomfort. Pressure sensing systems have demonstrated accuracy in the detection and tracking of posture in both blind and known studies. [14] developed a pressure sensor based system for detecting postures. It used a sensor array consisting of 64 sensors arranged on 8 by 8 cells. Even though the accuracy was as high as 93.9%, cost remained a major limitation. The approach used by [15] is very similar to one adopted by us in terms of sensor arrays used and techniques used to identify the sitting posture. The pressure sensor array is placed on the chair cushion to collect pressure data while the user is sitting. After collecting the data, a set of user-invariant features are modelled using the extracted data and then sitting posture is identified using machine learning method. A drawback of this approach is that it requires wired connection with the computer thus limiting its use to the people working on computers and no data is available for future analysis. The solution proposed by [16] places the cushion on the back, contrary to [15] which places the cushion on the bottom, and incorporates four motors to deliver the vibrotactile feedback to the user as negative reinforcement. A smartphone application is provided for calibrating the sensory cushion according to the users upright posture. The uses of the motors make this system power hungry and require permanent power. No details of the posture are provided for future analysis. Therefore, there is a need for an effective and smart solution with minimum cost, discomfort, and intrusiveness. We present an IoT based smart chair solution for posture monitoring which uses the minimum number of sensors together with the latest smart technologies such as IoT connectivity with the cloud, data storage, and mobile phone application for generating alerts and reports for the user as well as healthcare professionals.

III. System Architecture

The main components of the system shown in Fig. 2 are data acquisition, connectivity, data storage on cloud, data computations, alerting and reporting. The posture alert system is based on an array of connected sensors embedded on the physical chairs which are used to collect the information and this collected information is stored on a cloud server. This provides the flexibility of accessing the data from anywhere in a secured manner. Apart from two different types of sensor arrays, the data acquisition system also includes a microcontroller module to acquire the analog data, perform analog to digital conversion, and prepare this data for transmission using a wifi module. The data acquired is then JSON encoded with a timestamp, a posture value assigned and resultant JSON packet is sent to the cloud using RESTful APIs. The data is transmitted to Amazon Web Service(AWS) cloud server, which is running an instance of the webserver as a Node.JS application. Our server is located at the IP address shown in Fig.2. The API on the NodeJS application then stores these readings into a relational database. These values are then read into our machine learning algorithm, by invoking database queries to receive the most recent sensor readings.

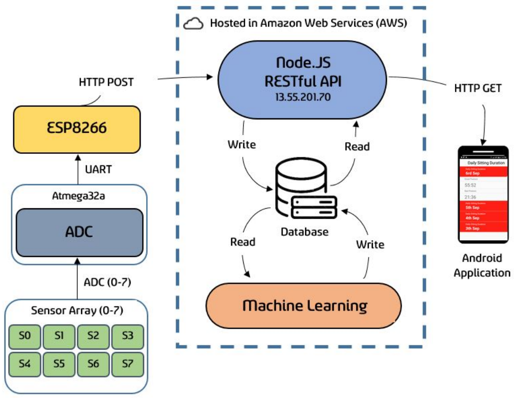


Fig. 2. Overall System Architecture

We use a machine learning technique to recognize the posture of an individual. The Nearest Neighbour machine learning algorithm, which is triggered every 10 seconds to process the received data and classify the posture. An interactive Android application is provided for issuing the alerts to the user and also generates reports in a way easily analyzable by the user. The retrieved data will be displayed on the Android application. Users can view their daily sitting posture, as a ratio in time of their good and bad posture rendered as a pie or a line chart as shown in Fig. 6.

TABLE I
PRESSURE SENSORS COMPARISON

Sensor	Max Force(Kg)	Area(mm^2)	Shape
FlexiForce A502	50	2580	Square
FlexiForce A201	50	73.9	Cirlce
FSR 406	12	1452	Square
FlexiForce A401	12	507	Circle
FlexiForce ESS301	0.5	73.9	Circle

IV. Design and Implementation

A. Hardware Implementation

When selecting the sensors, the two main criteria considered were surface area and force range. The pressure sensors with lower range tend to be less expensive compared to sensors with a higher maximum force. A comparison of the required features of the sensors considered during the design is provided in the Table I. Higher range sensor, Flexifore A502, were used for the high-pressure areas and medium-range sensor, FSR 406, for lower pressure area in order to minimize the cost. The system uses eight sensors in total, two higher range sensors and six medium range sensors. Eight critical points on a chair were discovered to detect the various seating positions as shown in Fig.3.

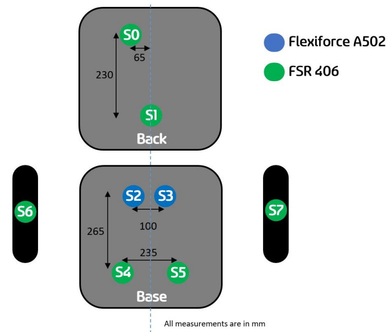


Fig. 3. Sensor placement on the chair

The arrangement shows six FSR 406 sensors placed at the positions on the chair highlighted in green, and two Flexiforce A502 sensors at position highlighted in blue, as this is the location on a chair where the most amount of force is applied. The final placement of sensors was a result of rigorous testing.

Although Arduino Yun was used during the prototyping stage for the data acquisition, the sheer size of the module was not suited for practical use. In order to reduce the footprint of the system, we opted to create a custom PCB design which comprised of the ATMega32A as the main micro-controller. This also encompassed the right amount of ADC channels required for the sensors as opposed to the Arduino Yun. The WiFi module, ESP8266, is mounted onto the PCB and communicates with micro-controller using the serial port. The resultant system was much more economical in terms of size and cost as shown in Figure 4.

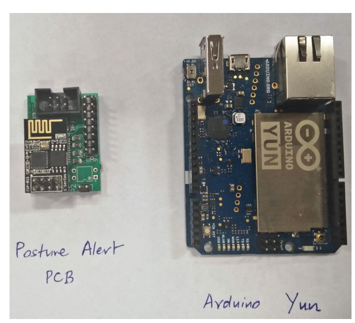


Fig. 4. Size comparison of custom PCB and Arduino Yun

A small piezoelectric buzzer can also be attached to the chair to produce an alert in case the user does not use a mobile phone. The system is capable of sending alerts to the controller side as well.

B. Software

Amazon Web Services (AWS) is a popular computation platform which offers secure and robust cloud computing services. This platform is used to deploy the Node.JS server. The specific service used is referred to as Elastic Cloud Compute (EC2). Amazon EC2 allows to set up and configure an operating system. For this application, the Linux OS Ubuntu was used due to its widespread use. The web server is used for communication with the mobile phone application and the data acquisition system. A Node.JS application was developed and uploaded onto the EC2 instance. An Internet protocol (IP) address was used to access the application running on the server. EC2 also allows MySQL database to run in the cloud which allows remote access. Initially, the backend database was implemented using MongoDB, which was preferred over MySQL, as MySQL required a pre-defined schema and relationships between fields.

While testing our machine learning algorithm, it was recognised that the algorithm worked very accurately when only a single user's sensor readings were present in the database. Eventually, as multiple user's sensor readings were added, accuracy of the algorithm dropped dramatically which is explained in the Section V. MySQL was chosen for the final implementation as it proved to be more scalable and allowed relations between tables, such as the 'Users' and 'SensorReadings' tables. New user data could be added to the database without affecting the performance of the machine learning algorithm. The new relational type database is shown in Fig. 5.

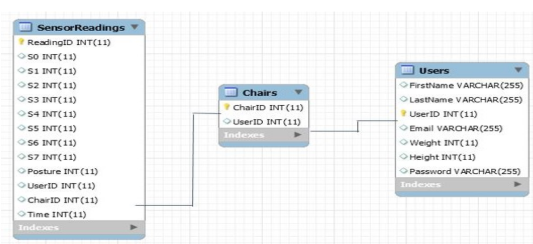
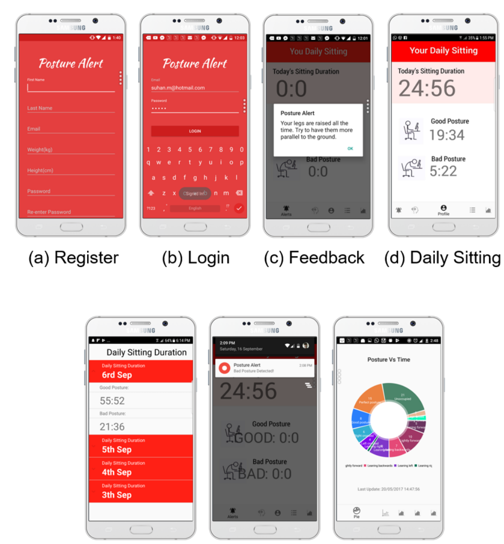


Fig. 5. Relational database

The data stored in the cloud can be retrieved by any client application such as by Android. As mentioned earlier, the Android application needs to issue HTTP GET requests to the web server. Different screen-shots of the Posture Alert Application are shown in Fig. 6. When a user downloads the application for the first time, they would have to create an account and register themselves as shown in Fig. 6a. In order to create a new account, the users are required to enter their details such as name, surname and email address. This information is used to create a new profile for the user. After the registration process is complete, the user may log in by using the login screen as shown in Fig. 6(b). Given that a user is not logging in for the first time, the user is welcomed with a pop-up message indicating the problem with posture detected based on the previous data. This feedback can help users to improve their posture by avoiding the same posture. Fig. 6(d) reports the user sitting information where the user can monitor how long they have spent seated in a good and bad posture throughout the day. Fig. 6(e) allows the users to compare their current day with their history and see how their good to bad ratio is changing over the days. Fig. 6(g) presents the same information graphically such as using pie, bar, and line charts. Finally, Fig. 6(f) demonstrates an example of a posture alert notification received on the screen accompanied by a beep. Whenever a user remains seated in a bad posture for more than 15 minutes or any preset value entered by the user, a notification will be sent which can be used as a negative reinforcement to avoid bad posture. Alternatively, if the user remains seated in a good posture for an extended period of time as defined by the user, again an alert message is sent to the user suggesting them to take a break. The user can set the times based on his nature of the job to avoid unnecessary distractions.

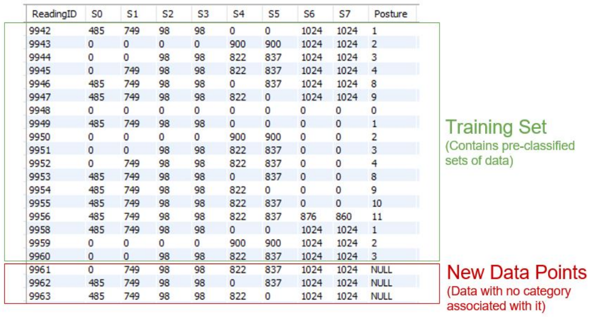


(a) Register (b) Login (c) Feedback (d) Daily Sitting

(e) Sitting History (f) Notification (g) Graph

Fig. 6. Screenshots of Android application

Three main machine learning algorithms considered for recognizing the posture include Decision Tree (DT), K Nearest Neighbours (KNN) and Support Vector Machine (SVM). DT algorithm is easy to set up and train but it is not powerful enough for complex data whereas SVM was found to have high algorithmic complexity and required excessive memory. KNN was chosen due to its robustness to noisy training data and remained effective for large training dataset. Finally, Python's popular machine learning library 'scikit-learn' was used to implement the algorithm [17]. Whenever new sensor data arrives into the API, it is stored in the database with posture attribute assigned to 'NULL'. Datapoints with a posture not assigned to 'NULL' are



ReadingID	S0	S1	S2	S3	S4	S5	S6	S7	Posture
9942	485	749	98	98	0	0	1024	1024	1
9943	0	0	0	0	900	900	1024	1024	2
9944	0	0	98	98	822	837	1024	1024	3
9945	0	749	98	98	822	837	1024	1024	4
9946	485	749	98	98	0	837	1024	1024	8
9947	485	749	98	98	822	0	1024	1024	9
9948	0	0	0	0	0	0	0	0	0
9949	485	749	98	98	0	0	0	0	1
9950	0	0	0	0	900	900	0	0	2
9951	0	0	98	98	822	837	0	0	3
9952	0	749	98	98	822	837	0	0	4
9953	485	749	98	98	0	837	0	0	8
9954	485	749	98	98	822	0	0	0	9
9955	485	749	98	98	822	837	0	0	10
9956	485	749	98	98	822	837	876	860	11
9958	485	749	98	98	0	0	1024	1024	1
9959	0	0	0	0	900	900	1024	1024	2
9960	0	0	98	98	822	837	1024	1024	3
9961	0	749	98	98	822	837	1024	1024	NULL
9962	485	749	98	98	0	837	1024	1024	NULL
9963	485	749	98	98	822	0	1024	1024	NULL

Training Set
(Contains pre-classified sets of data)

New Data Points
(Data with no category associated with it)

Fig. 7. How machine learning algorithm uses MySQL

pre-classified sets of data and used as the training set. This is illustrated in Fig. 7.

V. Methodology

In order to recognize a posture, the first step is to identify what a good and a bad posture looked like. We noted down all the different ways of sitting on a chair. This is represented in Fig. 8. We concluded that postures 8a-8k were not the ideal positions and we needed to find a way to identify them. The sensors actively collect data which is then matched to posture templates as shown in Table II. Depending on the state detected by template matching, the appropriate message would be generated to alert the user.

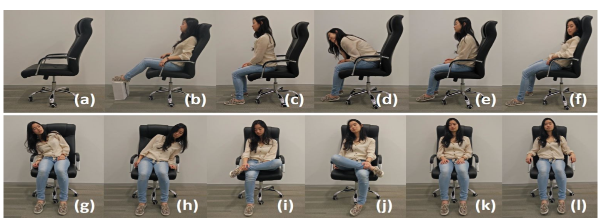


Fig. 8. Different seating positions

TABLE II
Posture Values

Posture	S0	S1	S2	S3	S4	S5	S6	S7
Fig. 8 a	0	0	0	0	0	0	-	-
Fig. 8 b	M	M	H	H	0	0	-	-
Fig. 8 c	0	0	0	0	H	H	-	-
Fig. 8 d	0	0	M	M	H	H	-	-
Fig. 8 e	0	M	M	M	M	M	-	-
Fig. 8 f	M	M	0	0	H	H	-	-
Fig. 8 g	0	M	H	L	L/M	M/H	-	-
Fig. 8 h	0	M	L	H	M/H	L/M	-	-
Fig. 8 i	M	M	M	M	0	L/M	-	-
Fig. 8 j	M	M	M	M	L/M	0	-	-
Fig. 8 k	M	M	M	M	M	M	0	0
Fig. 8 l	M	M	M	M	M	M	1	1

The postures shown in Fig. 8 are explained below:
a) Seat not in use (unoccupied)
b) User is in front of chair with legs raised
c) No contact with back (sitting forward)
d) Torso forward, no contact with back
e) Torso is lightly tilted forward (lightly forward)
f) Front of chair, leaning backward
g) Leaning left or forward left (leaning left)
h) Leaning right or forward right (leaning right)
i) Left leg crossed (left crossed)
j) Right leg crossed (right crossed)
k) Torso and legs are correctly aligned (good posture)
l) Torso/legs correctly aligned, shoulders relaxed (perfect posture)

TABLE II above shows the approximate sensor readings which were expected to be captured when seated in a particular posture. ADC sensor readings were divided into L (low), M (medium) and H (high) values. The ATmega32A has an ADC of 10 bits; this meant the values ranged from 0 - 1023. These values are normalized and mapped to three different levels, L (0 - 341), M (342 - 682) and H (684 - 1023). Two different pressure sensors were used in the project - FlexiForce A502 and FSR 406, both having different ranges. The range of these two sensors was vastly different which were also mapped to three different levels in the similar fashion. Additionally, in the table 0/1 means not-touching/touching, respectively. All the postures are shown in Fig. 8 can be determined using the arrangement shown in Fig.3. Finally, ADCs were tried to find typical readings for the L, M and H values. It was found that for average sized women the L, M, and H values were in the range of 0-200, 201-500, 501-1023 respectively. Whereas, for a large men L, M and H values were in the range of 0-400, 401-700, 701-1023 respectively. It can be seen that there is overlap between the ADC values and this makes it difficult to define hard limits for each of the L, M and H values. To fix this, machine learning was used to classify the sensor ADC values into postures. Different machine algorithms such as Decision Tree (DT), Support Vector Machine (SVM) and K Nearest Neighbour (KNN). DT was tried but it was abandoned due to inaccuracy in prediction as the complexity of the dataset increased. Similarly, SVM was not feasible as it requires higher training time. KNN was adopted due to its simplicity, ease of training and accuracy. The KNN algorithm works by taking input consisting of k closest training examples in the feature space [17]. KNN classification was used. Posture is identified by a majority vote of its neighbours, with the object being assigned to the class most common among its K . It requires that a K value must be chosen for the algorithm. A K value is an integer value which defines the number of nearest neighbour nodes a new data point will be compared with. Next, all the points in the training set, as well as the new point to be classified, are plotted in an n-dimensional space. N refers to the number of features present in the dataset. The K number of nearest neighbours are analyzed and the new member is classified into the class with the most frequent nearest neighbours.

VI. RESULTS

A usable prototype has been developed which is connected to an Android application as seen in Fig. 9. In order to evaluate the accuracy and effectiveness of our posture alert system, we performed testing with different users. The system was able to correctly classify the postures shown in Fig. 8. In the background, a machine learning algorithm runs which categorizes the sensor readings into a posture. Machine learning proved to be a critical component of this project. A few of the machine learning algorithms implemented had their results compared and evaluated. TABLE III illustrates the accuracy of these algorithms. A test set with an 80 - 20 split was used to obtain these results. Where 80% of the data was used to train the system, and 20% was used for testing. Each algorithm underwent this method 10 times, and the accuracy was averaged out. The random algorithm has been added to the table to define what the accuracy would be if the postures were categorized randomly. Finally, personally trained KNN was used as it had the greatest accuracy at 95.42%. We also conducted the experimental tests on two objects to observe the effectiveness of the system. The users were observed during the work hours and an improvement in the time spent sitting in a good posture was observed.

TABLE III
ACCURACY OF DIFFERENT ALGORITHMS

Software Algorithm	**Posture Accuracy**
Random	8.33%
Decision Tree	43.98%
Personally Untrained KNN	72.86%
Personally Trained KNN	95.42%

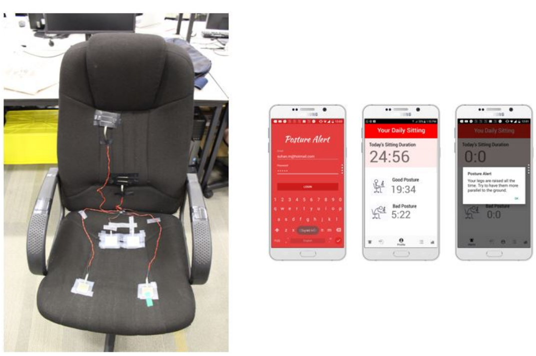


Fig. 9. Prototype and Android applcation

VII. CONCLUSION

In this paper, we presented a solution for sitting posture recognition based on pressure sensors and IoT. Twelve of the most common postures individuals usually sit in were identified, out of which ten were considered bad and had to be mitigated. The whole application was integrated with the cloud, allowing it to be used from anywhere around the globe. An aesthetically pleasing Android application was created which alerted users when seated in a bad posture for prolonged amounts of time. The accuracy of the algorithm was tested on different sized people and was found to be around 95% accurate. Additional features can be added to the Android application to make Posture Alert a fully-fledged product. An example of this is adding personalized posture based exercise routines which aim to reduce pain for particular parts of the body. Alternatively, a website could be designed where users can log in and view their daily sitting information. This could be useful for users who do not own a smart-phone but still choose to use the application.

REFERENCES

[1] H. Siddique, Office employees should be on feet for four hours of working day, study says, The Guardian, 01-Jun-2015.

[2] Similiar Sit Correct Computer Posture Keywords, KeywordSuggests. [Online].

[3] L. G. Writer, The Effects of Bad Posture on the Skeletal System, Back, 29-Sep-2016. [Online]. Available: http://healthyliving.azcentral.com/effects-bad-posture-skeletal-system-1858.html. [Accessed: 20-May-2017].

[4] L. Treadmill "Desks:Cracking the Code on Low Back Pain,life span fitness."[Online].Available: https://www.lifespanfitness.com/workplace/resources/articles/ treadmill -desks-lower-back-pain.[Accessed:19-May-2017].

[5] G. Alexey,"Poor posture," Oct. 2011. http://www.codeproject.com/Articles/260741/Sitting- posture-recognition-with-Kinnect-sensor.

[6] C.F. Juang, C.M. Chang," Human body posture classification by a neural fuzzy network and home care system application," IEEE Trans. Syst. Man Cybern. Part A Syst. Hum., 37 (2007), pp. 984-994

[7] R. Samiei-Zonouz, H. Memarzadeh-Tehran, R. Rahmani, "Smartphone-centric human posture monitoring system," In: 2014 IEEE Canada International Humanitarian Technology Conference-(IHTC). IEEE (2014)

[8] J. Estrada, L. Vea, Sitting posture recognition for computer users using smartphones and a web camera. TENCON 2017 - 2017 IEEE Region 10 Conference (2017): 1520-1525.

[9] H. Harms, O. Amft, M. Appert, R. Mller, and G. Troster," Wearable therapist: Sensing garments for supporting children improve posture," In Ubicomp 2009: Proceedings of the 11th International Conference on Ubiquitous Computing, ACM press.

[10] Upright posture trainers, UPRIGHT. [Online]. Available: https://www.uprightpose.com/. [Accessed: 20-May-2017].

[11] C. Mattmann, O. Amft, H. Harms, C. Troster, and F. Clemens, "Recognizing upper body postures using textile strain sensors," In ISWC 2007: Proceedings of the 11th IEEE International Symposium on Wearable Computers, pages 2936. IEEE Press.

[12] W. Y. Wong and M. S. Wong, "Detecting spinal posture change in sitting positions with tri-axial accelerometers," Gait & Posture, 27(1):168171, Jan. 2008.

[13] K. Kamiya, K. Mineichi, H. Nonaka and J. Toyama, Sitting posture analysis by pressure sensors. ICPR (2008).

[14] G. Liang, J. Cao, X. Liu, and X. Han, Cushionware, Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems - CHI EA 14, 2014.

[15] K. Ishac, K. Suzuki, "A Smart Cushion System with Vibrotactile Feedback for Active Posture Correction Haptic Interaction," Springer Singapore, 2018, 453-459

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, J. Vanderplas, "Scikit-learn: Machine learning in Python. Journal of machine learning research," (2011), 2825-2830.