Secure Wearable Medical Interface Device

Callum Alexander Joseph Owen-Bridge

Award of Submission: BSc Computer Science and Electronic Engineering

Liverpool Hope University

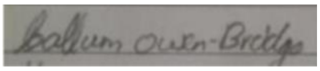
Department of Mathematics and Computer Science

Dissertation Supervisors: Dr Stewart Blakeway and Dr Emanuele Lindo Secco

Submitted to Liverpool Hope University, April 2017

Declaration

I hereby declare that this is entirely my own work and that it has not been submitted as an exercise for the award of a degree at this or any other University. I agree that the Library may lend or copy this dissertation on request.



CALLUM OWEN-BRIDGE 16/04/2017

Summary

A secure medical wearable device and web application was constructed to address the problem of access to medical healthcare. This aims to connect patients with specialist provision where necessary. This system helps engage the patient with their own health care and aims to promote independence over their own health. The issue to be resolved is low funding in health care, so the time to see patients are reduced. There are too many patients whom require appointments compared to the number of appointments available. There are too few doctors in clinics. There is a small population which take responsibility over their own health. It was found that many countries have been using eHealth strategies for a long period of time with success and aim to improve and further implement the strategies they currently use.

Abstract

This dissertation applies the use of eHealth to improve health care services through use of a secure medical wearable device. eHealth is a system which allows access to health care using the internet. A medical wearable device records and stores data about the body's physical parameters. A web application to view recorded data. Health services are struggling to diagnose human illnesses accurately and quickly enough, due to lack of data, large queues for appointments and long waiting times for results. Humans are becoming less proactive and unaware of their current health status. Many people struggle to allocate time to see a health care practitioner due to their lifestyle and some elderly people struggle to see health care professionals due to their physical health. A secure medical wearable device with a web application is a solution for this issue. The device records their vital body parameters, throughout the day. The web application allows the user to upload data, which can then be reviewed by a health care practitioner for review. To conclude, the research project was successful in creating a solution for the issue. However, there are issues to be resolved and many improvements required for the device and software for them to integrate more easily within current lifestyles.

Acknowledgements

I would like to thank my parents, partner, Dr Stewart Blakeway and Dr Emanuele Lindo Secco in helping me with this research project, they have helped keep me on track and motivated.

Table of Contents

List of Appendices

List of tables and illustrations

List of Abbreviations

1) AES (Asynchronous Encrypted Signal)
2) ASSL (Ambient Assisted Living)
3) CDN (Content Delivery Network)
4) CHF (Congestive Heart Failure)
5) CSV (Comma Separated Values)
6) ECG (Electrocardiogram)
7) EMG (Electromyography)
8) HCP (Health Care Professional)
9) HIPAA (Health Insurance Portability and Accountability)
10) UHC (Universal Health Coverage)
11) WBAN (Wireless Body Area Network)
12) WHO (World Health Organisation)

1.0 Introduction

The aim is to create a secure medical wearable device for humans, to help support in the monitoring of physical parameters in the daily life and in the Ambient Assisted Living (AAL) context. This dissertation will be focused on the development of software and the hardware of the device.

The research undertook reviews population health, eHealth, types of eHealth monitors and security, privacy and legal issues surrounding medical wearables. The research has shown high expectation for eHealth strategies and the use of medical wearable devices, as long as the security, privacy and motivation aspects surrounding these systems are successful.

In the materials and methods chapter is the construction of a secure medical device and a web application is discussed. This explains how the device is constructed, and what hardware was used. The creation of the web application is explained as well as what software and languages were used. The security and ethical issues of the web application and secure medical device is then discussed.

The results section contains a review of what has been created. The discussion reviews and compares what has been created to what the research shows. The conclusion shows the overall thought of a secure medical wearable device. The future work chapter discusses the future of this system and possible improvements.

2.0 Literature Review

In this chapter subjects surrounding the topic of eHealth will be reviewed.

2.1 Health and Initiatives

In this subchapter the health and wellness of the population will be discussed and any initiatives that have been used to encourage health improvement.

2.1.1 Health of the population

The causes of death in 2015, globally, related to the cardiovascular system and respiratory system, has risen significantly since the year 2000 as shown in figure 1 (World Health Organisation, 2017). Ischaemic heart diseases had risen by 2 million in 2015. Comparing figures 1 and 2 shows that new diseases became more common for causing deaths in 2015, such as diabetes mellitus causing around 1.7 million deaths and diseases related to the respiratory system causing around 1.8 million deaths.

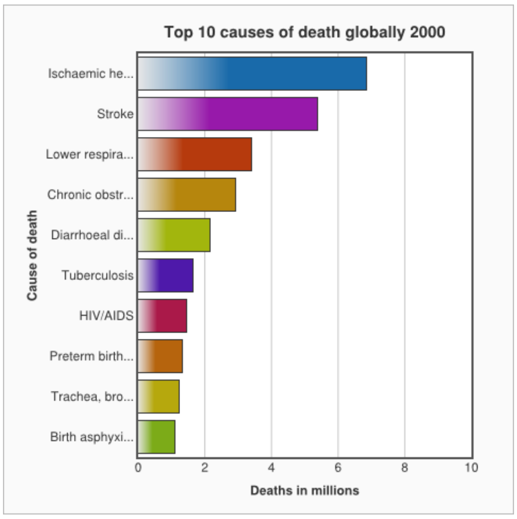


Figure 1Top 10 causes of death globally in the year 2000. (World Health Organization. (2017). The top 10 causes of death.)

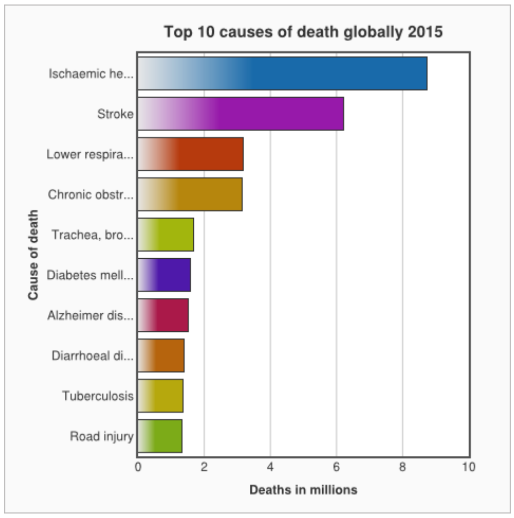


Figure 2 Top 10 causes of death globally in the year 2015. (World Health Organization. (2017). The top 10 causes of death.)

The results depicted in figure 2 show that the most common deaths are caused by the cardiovascular system (heart, blood vessels, and blood) and respiratory system. Therefore, a device that can record data about the cardiovascular system and respiratory system would be beneficial. As the device could provide data to help reduce illness caused not only by diseases related to the cardiovascular system and respiratory system but illnesses that affect the rest of the body because heart and lung diseases can contribute to other illnesses.

From the year 2000 to 2015 ischaemic heart disease has risen by 2 million deaths and there has been a rise in deaths caused by diabetes mellitus and Alzheimer's disease. These two diseases were not present in the top 10 causes of death in the year 2000. The rise in diabetes mellitus and Alzheimer's disease could be due to the type of lifestyle people are having. With the use of a health monitor, these diseases could be either reduced or prevented as well as other diseases.

2.1.2 Wellness and Initiatives

In this subchapter, the effect of health initiatives on the population and the importance of wellness will be discussed.

2.1.2.1 Wellness

Wellness is defined as "the state of being in good physical and mental health" (Reference, n.d). Constant monitoring and maintenance of personal wellness are important for people to acknowledge, as this can keep them aware of their health conditions in different environments and their current health.

2.1.2.2 Initiatives

The use of health-related initiatives, help to educate the population about their health and others. The aim of these initiatives is to make people more aware of their health, to be more proactive and take more control over their health.

Initiatives such as 'This Girl Can', have been created to motivate, encourage and help women to become more proactive with their lifestyle and become more involved in sports activities.

Many fitness applications on mobile devices, use a technique to create competition called gamification. Gamification is the use of game aspects of an application, for example, a fitness application could create competition between multiple people by allowing them to share the statistics of a current exercise they have completed (Lister et al., 2014).

2.2 eHealth

In this subchapter, eHealth will be explained and reviewed.

eHealth (electronic health) describes health services and information enhanced by the internet and technology. eHealth is a field involved in medical informatics and public health (Eysenbach, 2001). The WHO in 2016 developed the definition to include workforce training using eLearning, using electronic health records to obtain up to date information enabling improved diagnosis and treatment of patients, improvements to the healthcare systems in terms of time and money and enhance health care for remote communities (World Health Organization., 2017).

eHealth has been shown to provide better support for patients at home and can have a dramatic effect on cost and efficiency of healthcare. Portable health monitors have shown to increase self-care, quality of life, and improve medical care (Alnosayan, N., et al., 2014).

Below is a list of aspects of eHealth, these are required for eHealth to be successful and what eHealth will provide:

- **Efficiency:** medical costs can be decreased by increasing the efficiency of health care. This could be done by preventing duplicated data and unnecessary diagnosis by improving the communication between healthcare premises and increasing the involvement of patients (Eysenbach, 2001).

- **Improving the quality of care:** eHealth could improve quality of care by providing comparisons between providers and consumers and so directing patients to the best providers (Eysenbach, 2001).

- **Evidence-based:** the effectiveness and efficiency of eHealth interventions should be evaluated scientifically (Eysenbach, 2001).

- **Empowerment of consumers and patients**: eHealth provides new opportunities for medicine and medical care which is patient-centred, giving them more choice. This is done by making patient records and the knowledge of medicine, available for consumers (Eysenbach, 2001).

- **Encouragement:** to improve the relationship between the patient and health care professionals, so that collaborative decisions are made (Eysenbach, 2001).

- **Education:** provide up to date information for both physicians and patients via online sources (Eysenbach, 2001).

- **Enabling:** allows for information exchange between health care premises, in a more standardised manner (Eysenbach, 2001).

- **Extending:** this allows consumers to gain access to health services, knowledge, interventions and products provided by wider-reaching providers (Eysenbach, 2001).

- **Ethics:** eHealth has issues with privacy, consent, equity, professionalism, and ethics (Eysenbach, 2001).

- **Equity:** until a political measure ensures that everyone will have equitable access, eHealth could have equitable issues because only people who have money to enable them to have access to computers and/or networks are more likely to have access to eHealth. Not having access to eHealth further deepens the divide between those who have and have not (Eysenbach, 2001).

Therefore, eHealth is a broad term used to describe the combination of technology and medical knowledge to provide a service to the population which is more efficient than current schemes and can provide a much more enhanced experience with medical care and allows for people to independently monitor their health. This helps to improve the medical knowledge people have, which can improve the confidence and social aspects of people’s lives (Eysenbach, 2001).

However, the eHealth scheme can only be successful if it is implemented properly to allow all people to have access. This can be an issue for undeveloped countries which cannot afford this type of care (Eysenbach, 2001).

The World Health Organisation (WHO) states that only 58% of member states (countries that are a member of WHO) have an eHealth strategy, 55% of countries have legalisation to protect electronic patient data and 87% of countries report having one or more national initiatives (World Health Organization., 2017).

A report created by WHO in 2016 shows how much eHealth has been implemented, as it states that 58% of member states have an eHealth strategy and of 90% of these strategies there are references to Universal Health Coverage. Universal Health Coverage (UHC) is a scheme so that all people and communities receive quality health services they need without suffering financial hardship (World Health Organization, 2017).

WHO states that 90% of countries with eHealth strategies have a specific funding for eHealth services; this shows the importance of eHealth and that countries see that eHealth is a useful scheme. Since 2005 WHO has had eHealth as the top priority (World Health Organization., 2017)

As the following graph shows, eHealth is a growing health scheme that more and more countries are adopting. Below is a timeline of countries from 1990 to 2015 that have adopted eHealth schemes.

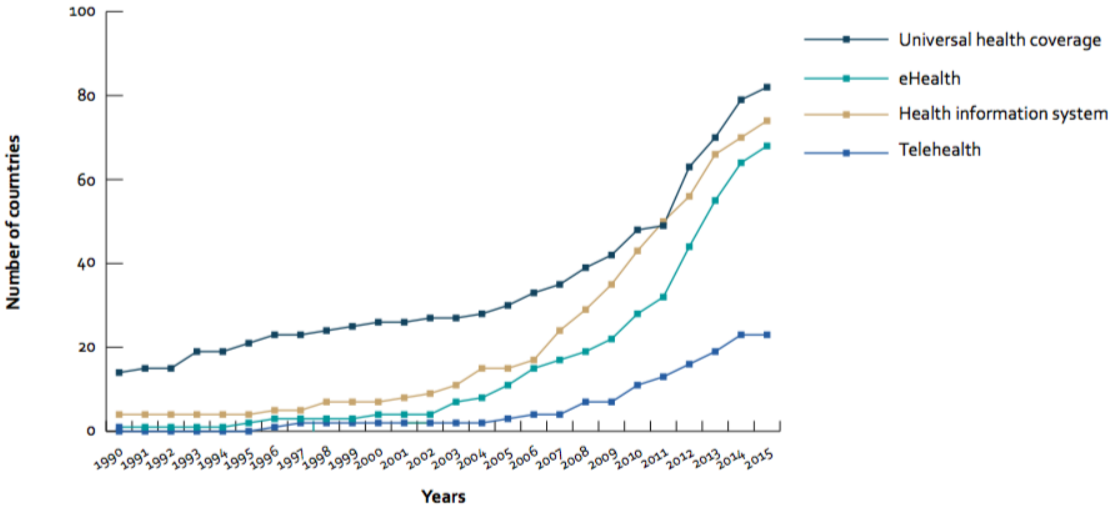


Figure 3 line graph for the number of countries which have adopted Universal Health Coverage(UHC), eHealth, Health Information System, and Telehealth from 1990 to 2015. (World Health Organization., 2017)

Figure 3 shows a line graph of results, taken by the World Health Organisation, for the number of countries which have adopted Universal Health Coverage (UHC), eHealth, Health Information System, and Telehealth from 1990 to 2015. The graph in figure 3 shows a steady and then sudden increase in the adoption rates of eHealth strategies. The gradual increase is shown between 1998 and 2006, where the adoption of eHealth went from about 5 countries to about 19 countries.

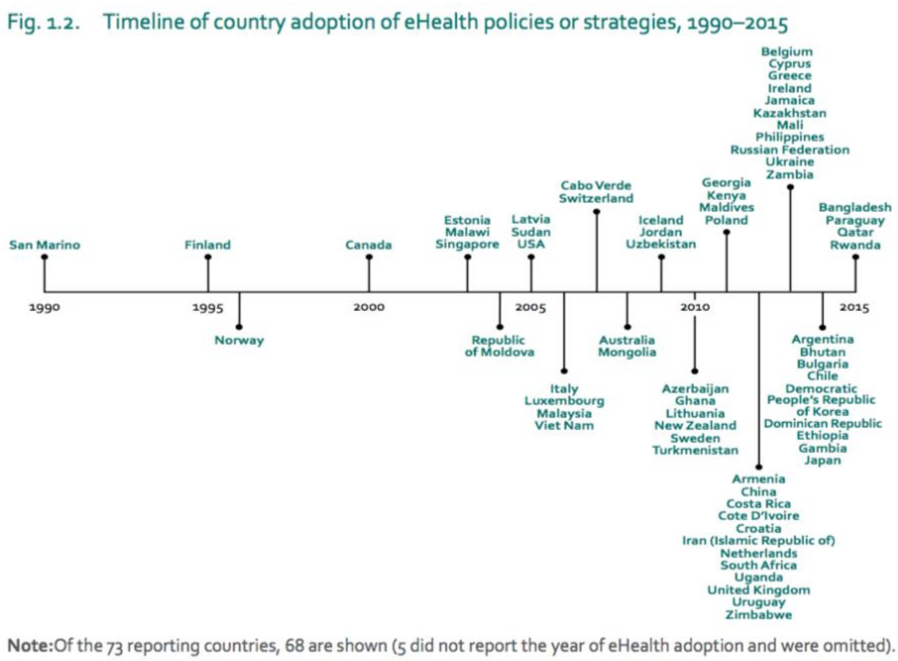


Figure 4 timeline showing the number of countries which have adopted eHealth strategies from 1990 to 2015. (World Health Organization., 2017)

Figure 4 is a timeline of when countries adopted eHealth, the results were taken by the World Health Organisation. This timeline spans from 1990 to 2015. The higher adoption rates made by countries occurred between 2003 and 2015.

There is a high potential, shown by the eHealth system, to reduce medical costs, improve diagnosis and help solve the on-demand health care challenges.

There have been recent advancements in Wireless Body Area Networks (WBANs), which allows for sensors to be placed around and in the body for continuous monitoring of physiological parameters (World Health Organization, 2017).

With the use of wearable medical eHealth devices, the patient's parameters can be measured and monitored, continuously, in real-time and remotely. This removes the constant requirement of face to face measurements, therefore increasing the number of appointments that can be used for other medical requirements (Li, M., et al., 2010).

Doctors usually assess a patient's condition at regular intervals, creating discrete data. However, the patient conditions could change between the intervals or new symptoms may form or a symptom may only occur during the period the patient is not assessed. The use of eHealth devices to record patient parameters creates continuous data so providing doctors with a complete medical history (Omoogun, M., et al., 2017).

2.2.1 Benefits and challenges of eHealth:

The Netherlands adopted eHealth schemes in 2012 and more than 90% of doctors recommended the use of eHealth monitoring (Krijgsman et al., 2015).
The following are the reasons why the Netherland Government encourage the use of eHealth in health care:

1. **Time.** Services like telehealth help to save time as appointments can be made through an online provider or arrange an online consultation using a video link (Government.nl, 2018).

2. **Aware of their own health.** Digital health environments allow people to view their health more easily and can share their recorded health data with their health care provider. This provides much more effective and efficient health care, by the patient's and helps the health care professional diagnose the patient's health problem more quickly. With this awareness and control, patients can gain more control over their health (Government.nl, 2018).

3. **Reduced paperwork.** This requires less administration as health care professionals can share data digitally, more easily and more securely (Government.nl, 2018).

4. **Continuous data**. The use of medical recording devices helps health care professionals to receive continuous data. So, the health care professional can more precisely diagnose the patient, as they will have a timeline of the patient's health, spanning throughout the day or week, rather than a specific time of a day. Therefore, reducing gaps normally found in current data collection systems (Virone et al., 2006).

5. **Reduced medical errors**. eHealth could reduce the number of patients affected by medical errors such as software failures, insufficient data entries and technology affected by human fault errors (Ammenwerth and Shaw, 2005).

6. **Reduced cost**. Use of medical monitors can reduce costs to the health care budget, as they help increase the efficiency of health resources used by the health care provider and decrease the time taken to detect health conditions (Liang et al., 2012)

The challenges of eHealth are:

1. **Software.** Applications associated with eHealth can form health risks, privacy and security risks. This is due to the wider access to patient data and other third-party services could be allowed access to this data. Applications also have the potential to be accessed by cybercriminals (Catwell and Sheikh, 2009).

2. **Investments.** Investments in eHealth systems could reduce funding in other areas of the health system (Catwell and Sheikh, 2009).

3. **Adoption by physicians.** A factor which could affect the adoption of eHealth technology is the use of the technology by physicians. Physicians want the use of eHealth technology to be simple and user-friendly. Patients want to keep their data secure and some patients do not want to share their health records. Physicians are concerned about how expenses with device maintenance are covered. Physicians are concerned about reduced patient contact (de Grood et al., 2016).

2.3 Medical Wearable Devices

Medical wearable devices are part of eHealth as they can communicate via the internet. In this chapter examples of medical wearable solutions will be summarised and analysed.

2.3.1 MyHeart

A telehealth system designed to help patients with Congestive Heart Failure (CHF) care, to transition from hospital to their home. MyHeart consists of a mobile application and wireless health devices, used by the patient (Alnosayan, N., et al., 2014).

The aim of MyHeart is to achieve an early diagnosis. The wireless health devices used by MyHeart is embedded in textiles so that the collection of sensors can be worn comfortably. There are four concepts within MyHeart which aid in therapy support and prevention (Gacek, A. and Pedrycz, W., 2012). MyHeart brings new health technologies to health care by using devices in clothing, use of body electronics, algorithm development and interaction with professionals and users (Harris, M. and Habetha, J., 2007).

The aim of MyHeart is to prevent cardiovascular disease by having early diagnosis. It can help patients lead a healthier lifestyle as well as enable them to cope with cardiovascular diseases. The European Commission project involved 33 partners, such as Vodafone and Philips, from 10 countries (Harris, M. and Habetha, J., 2007).

The mobile application encourages self-care by displaying messages, by collecting and displaying patient data. The data displayed on the patient dashboard is coloured to highlight data points. Each colour represents low, medium and high-risk values tailored to the specific patient. The data is also displayed as a line graph, as this helps to form a correlation between the health data and the time it was recorded (Alnosayan, N., et al., 2014). A rule-based system with data aggregator is used to process the patient data and sends notifications to health care providers about patient health status.

The communication is via email or SMS messages and google cloud messaging to communicate with the patient's mobile. A clinician has a dashboard application to view the vital signs, symptoms and health risks of the patient. The software is designed to improve and enhance communication between patients and their healthcare provider (Alnosayan, N., et al., 2014).

Data transmission is encrypted between the patient's home and Cloud database, and between the clinician and Cloud database, to keep the patient data secure (Alnosayan, N., et al., 2014).

The patient's vital signs are collected daily by the device, which uses A&D and MyGlucoHealth devices to record blood pressure, weight and blood glucose. MyHeart connects to these devices by Bluetooth. The device then connects to a 2Net communication hub, which has a cellular connection to the healthcare provider (Alnosayan, N., et al., 2014).

The database used by MyHeart keeps its data secure by using a unique key generated by the patient's mobile application. Every time the patient's mobile application communicates with the database it uses both the patient's mobile number and the generated unique key, to allow access to data. The MyGlucoHealth device uses an automated method to establish a connection with the database. The security for the MyGlucoHealth device is maintained at the seller's location (Alnosayan, N., et al., 2014).

Data collected by MyHeart is stored on Microsoft SQL servers using Windows authentication to keep it secure. The database design uses a data mart design to send and display information. This is used as it provides efficient speeds and allows for data security and traceability. The database stores patient data in a traditional transactional design to allow for scalability. This reduces system impact (Alnosayan, N., et al., 2014).

2.3.1.1The benefits of MyHeart

- The monitor system of MyHeart can be incorporated in clothing such as a t-shirt, therefore making it simple to be worn and is unobtrusive.

- The Myheart system uses a hub to connect the application and devices together and communicates with the HCP via a cellular medium, therefore removing the requirement for Wi-Fi and allows for automated uploads.

- MyHeart collects patient data daily and displays it for both the patient and HCP, this allows the patient to see the health of their body throughout the day and understand how their body functions with the environment.

- A rule-based aggregator is used to process patient data and send alerts to the HCP about patient health status. It also sends messages to the patient about their health to encourage self-care, specific to their health and profile.

- MyHeart keeps patient data secure by encrypting data transmissions and using unique identifiers to access data, so keeping patient data secure and private.

2.3.1.2 The Weaknesses of MyHeart

- A rule-based aggregator is used to decide whether the HCP should be notified of abnormal health statuses, this could either send false notifications or no notifications when one is required.

- A mobile is required to run the application, therefore limiting the use of the system as portions of the population may be unable to use a mobile or be able to afford one.

- The hub used for communication with the health monitoring sensors and devices is required for data uploads and requires a mobile connection, this is not practical as the patient may be away from the hub for a long period of time or the hub may not establish a connection reliably or for a long period of time, therefore, hindering the patients' health care.

2.3.1.3 Conclusion

To conclude, MyHeart is a very useful system, it can record many aspects of the human body using unobtrusive methods and is able to use multiple sensors in a modular way. The patient and HCP can communicate with each other, allowing collaborative care and allowing the patient to remain in their comforting environment.

However, the flaws with the MyHeart system make it complex for patients and possibly HCP’s and health care providers require equipment that is highly reliable and simple to maintain. The use of a mobile limits the type of patients this system could be used for.

2.3.2 AMON (Alert Portable Telemedical Monitor)

AMON was designed for high risk, cardiac and respiratory patients. The device records continuous data for vital parameters. AMON can detect medical emergencies and has a connection to the telemedicine service via cellular networks (Anliker, U., et al., 2004).

AMON is an unobtrusive wrist-worn device as shown in figure 5, this is to reduce the interference in the patient's lifestyle and mobility. All of the sensors are incorporated into the device (Anliker, U., et al., 2004).

This device measures:

- Blood pressure and blood oxygen (SpO2)
- Skin temperature: experimental purpose, as the sensor location may not provide a reliable body temperature.
- Activity
- 1-lead ECG
- Additional sensors using interface:
    - 12 lead ECG

The device can recognise the patient's activity by using a 2-axis accelerometer. It then uses this information to correlate with body vital sign data.

The online system collects the patient data and displays them in a graphical form for both the user and telemedicine centre.

This device has emergency detection, which uses the patient profile and recognised activities to reduce false alarms (Anliker, U., et al., 2004). The software, allows authorised users to access patient medical records as well as the data the patient has uploaded. The system allows authorised users to request a measurement and communicate with patients (Anliker, U., et al., 2004).

Figure 5 AMON wrist worn device, (Anliker, U., et al., 2004).

The device displays messages to communicate its current working status. There are messages for:

- Ok measurements. Data is transmitted three times a day to the TMC.
- Parameters out of range. This causes a re-measurement. The device then decides whether the TMC needs to be notified.
- Multiple parameters out of range. TMC is notified and receives the recorded data (Anliker, U., et al., 2004).

If communication between the device and TMC is broken. The data can be held on the device for up to 4 days. The connection uses a GSM cellular signal; however, the user is asked to remain within good area coverage.

The device has a battery which lasts up to 24 hours. The device can also receive text messages from health services (Lukowicz, P., et al., 2002).

2.3.2.1 The benefits of AMON

Compact unobtrusive wearable. This device is all contained inside a large bracket that is worn on the wrist, so, the device is simple to set up by the patient and use. There are no cables reducing any health risks and simplifying setup.

AMON can record data for 4 days if a connection to the healthcare provider is lost. The device displays simple messages to indicate its working status, therefore preventing the patient from being confused.

The connection between the device and telemedicine centre is by a cellular signal, this ensures the patient has a more reliable connection and does not have to have access to Wi-Fi. This also allows the TMC to receive data more frequently, three times a day, therefore noticing if the patient requires medical help.
AMON can recognise patient activity, therefore reducing the number of false alarms.

AMON has a battery which lasts up to 24 hours. Allowing the patient to be monitored for a day. AMON allows health services to communicate with the patient, by sending the patient messages to their wrist-worn device, which could comfort the patient.

The device allows the patient to be more aware of their health and encourage self-care, therefore reducing the pressure on health care providers for medical help.

2.3.2.2 The Weaknesses of AMON

If a cellular connection is not made after the 4 days, the device may delete previously recorded data that could be detrimental to the patients' health care and the patient may require assistance if they are stranded for example elderly or disabled people.

The device is unable to record body temperature and many other aspects of the body such as blood glucose, this reduces the use of the device for people as it does not have Bluetooth to communicate with other devices.

The device is bulky, so patients may feel embarrassed to wear the device. The device can only use a cellular connection, this inhibits where the device can be used, which could prevent the patient from staying in an area longer than 4 days.

2.3.2.3 Conclusion

To conclude, this device is very useful and practical in the area it has been created for, cardiac and respiratory patients. The device records many useful body parameters, however, not all the body vital signs are recorded and parameters such as the patients breathing would be required for respiratory patients.
The bulky aspect of the device may persuade the user not to wear the device as it may be viewable by other people. AMON allows the patient to be contacted via the wrist-worn device using text messages, this keeps the patient more aware of what is happening and may comfort the patient.

In comparison, MyHeart allows the patient and HCP to monitor more specific parameters of the human body, as sensors can be added or removed, whereas AMON has a specific set of sensors to monitor cardiac and respiratory patients. AMON uses many algorithms, patient activity, and patient profile to depict what the patient is doing to prevent sending the HCP false notification, whereas MyHeart uses a data aggregator to determine whether the HCP should be notified.

2.3.3 Overall

Overall, both AMON and MyHeart are useful and well-created systems. However, with the system created in this project will improve and avoid on the flaws found in AMON and MyHeart to create a system reliable for health care providers but is also simple to use and maintain.

2.4 Healthcare Systems

In this chapter current healthcare systems used to manage patient health records and allows a form of patient independence over their health will be analysed.

2.4.1 Patient Access

This is a service which the NHS currently uses and is provided by EMIS. This allows patients to access their health records online, view doctor-patient notes, create appointments and review their prescriptions. This service also allows patient self-monitoring, by allowing patients to upload health data via the HealthKit app created by Apple. This reduces the number of appointments used and provides clinicians with more detailed information. Patient Access uses encryption within the app and website, keeping patient data secure (Emishealth., 2016)

Patients can view why, where and when their records have been accessed by General Practitioners; if available (Emishealth., 2016)

2.4.2 Babylon

This is an NHS application. This app allows people to talk to GMC certified doctors with 10 years' experience. People can either do a video or voice call with doctors. This app allows people to view doctors without visiting surgeries, this allows people to still see doctors without the need of travel, which is useful for those with disabilities and can save time for people who cannot make appointments. These appointments can be made within minutes at any time of the day and week (babylon health, 2017)

This app also contains Artificial Intelligence (AI). This is used as a symptom checker to help people to self-diagnose and the AI will recommend appointments if necessary. This process can be done within 60 seconds (babylon health, 2017)

Soon the app will contain the ability to monitor health and share the data with doctors, similar to the approach used with Patient Access (babylon health, 2017)

The app allows people to choose the GP clinics they would like to use. Patients can review consultation videos and notes, NHS prescriptions, referrals and fit notes (babylon health, 2017)

Babylon uses encryption within the app and website. Babylon is currently under testing and so can only be used within London (babylon health, 2017)

2.4.3 Healthcare Systems Review

Both Patient Access and Babylon are useful pieces of software, simplifying health care for patients, by quick appointment time and access, quicker access to prescriptions and access to their own medical records.

Both applications provide security within the system using encryption to hide transmitted data.

Although, Patient Access and Babylon are quite similar Babylon is far more superior, professional and modern. Babylon allows people to see a doctor at any time of the day and week either through video or voice calls. The application contains an AI service for the symptom checker, allowing people to understand their health. Soon, the application will have a monitor aspect with connections with the AI allowing people to take more control over their health and wellness and become more proactive in keeping healthy.

Compared to the web application made in this research project, the web application provides the patient with the ability to view their health data in graphical form and the patient can access this web application from any device with internet access. Whereas, currently Patient Access only allows the user to upload data from an iOS device with the HealthKit app which is both the Apple iPhone and Apple iPod Touch, this limits this feature greatly. Babylon currently does not allow for this health data sharing. This research project uses a secure medical device which is modular and so is not limited to what body parameters it could measure, whereas with a mobile device is limited to the devices that support it.

2.5 Security, privacy and legal concerns surrounding a Medical wearable device

In this chapter, the Security, privacy and legal concerns surrounding a Medical wearable device will be discussed and how it affects different people.

It is essential that the data recorded and stored by medical wearable devices are secured as this holds personal information for patient diagnosis. Not securing the data may prevent the patient from being treated effectively, as this provides the potential for unauthorised users to alter information. One of the security issues with wireless medical wearable devices is the data transmission to the database via the internet because it is possible that the data can be lost or eavesdropped (Li, M., et al., 2010).

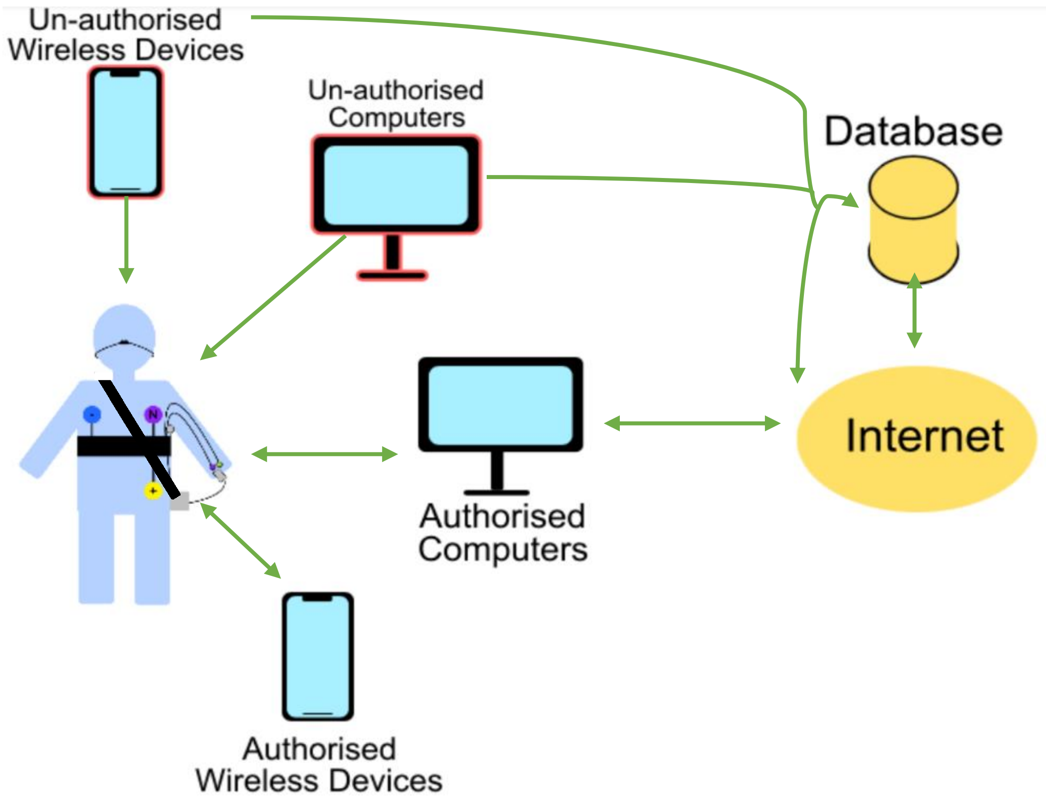


Figure 6 security risks between the medical wearable device and other devices.

Figure 6 shows which devices could be involved with the wireless medical device and the links between the devices are possible parts of the system that could be compromised by an unauthorised user. The devices highlighted in red are devices, unknown to the user of the wireless medical device, which could intercept wireless connections with the device or could access the data related to the device.

2.5.1 Security of Medical Wearable Devices

Security issues that are concerned with medical wearable devices and associated software:

- **Eavesdropping:** a network hack that occurs during transmission of data between two end systems. The attacker intercepts this data during transmission (Omoogun, M., et al., 2017).

- **Data Access**: this is when an unauthorised user accesses data while the data is being transferred between end systems. This is an issue because the user can then modify the data causing health risks for patients (Meingast, M., et al., 2006).

- **Denial of Service**: this is when the attacker overloads servers this prevents the servers from being accessible (Omoogun, M., et al., 2017).

- **Access to the device**: unauthorised access to the device recording the patient parameters, could allow the user to access personal data of the patient and could locate the patient (Omoogun, M., et al., 2017).

- **Data ownership**: This is about who can have the authority to delete, edit and add information to health records. Do patients have ownership of their own data? Do insurance providers own patient data? This can cause many issues, because if insurance providers have access to this information then they can refuse to cover expenses for treatment (Omoogun, M., et al., 2017).

- **Type of Data stored**: the type of data stored should be minimal, to prevent patient identification and prevent unnecessary data from being stored, but still provide optimal health care (Omoogun, M., et al., 2017).

- **Storage of data**: This is about where should the patient data be stored. Should the data be stored locally or at a central database such as a Hospital? (Omoogun, M., et al., 2017).

- **Patient record access**: There are two types of people who have access to electronic patient records those who can read and edit the record and those who are only allowed to read the record. Some people may want to restrict what can be seen by those with only read privileges. The patient record can be sent to other health care practitioners without the patients' consent (Omoogun, M., et al., 2017).

- **Data mining**: When mining medical data, it may be possible to profile patients, using basic information such as age and gender. Therefore, increasing the possibility of discrimination and exclusion of people. So, this data must be collected and stored to prevent the ability to profile a patient (Omoogun, M., et al., 2017).

Security solutions for medical wearable devices and associated software:

- **Encryption:** this is a method that converts data into a form which is unreadable. This is then transmitted across the network and decrypted by the destination system. Encryption and decryption involve the use of a unique key which only intended users know. This key is used to decrypt an encrypted file, without the key the file will take a substantial amount of time to decrypt. By making the data unreadable, it prevents the data from being susceptible to eavesdropping (Omoogun, M., et al., 2017).

- **Authentication mechanisms:** this ensures that the data is being used by authorised people and that the data comes from people that they claim to be. Examples of this method are digital signatures, usernames, passwords and challenge-response protocols (Meingast, M., et al., 2006).

- **Unique Identification:** use of identification numbers to hide personal patient data, will help prevent eavesdropping and patient identification.

2.5.2 Privacy of Medical Wearable Devices

In the United States of America, a governmental initiative was created in 1996 called Health Insurance Portability and Accountability (HIPAA). This initiative has created a set of privacy rules for wireless medical devices to help protect sensitive personally identifiable medical data. A form of encryption and cryptography is required for the data as a node within the internet can be compromised, so this will help keep patient data private (Li, M., et al., 2010).

2.6 Research Project Uniqueness

The difference between this research project and those already created are:

- The ability to upload and send the recorded data to a health care professional, who can view the data.
- The data is kept secure and encrypted on the device and within the database.
- The user and health care professional can view the recorded data in graphical form on the website.
- The device is modular, therefore can record as many human body parameters as wanted.
- The sensors are worn under clothing.

2.7 Overall

Overall, the health of the population is becoming worse in diseases which can be monitored, diagnosed early and prevented with the early diagnosis.

The use of eHealth systems such as online applications, software, and medical monitors have shown to be useful in personal health monitoring; as described in 2.2.1 Benefits and challenges of eHealth:. As health care professionals saw that patients liked the eHealth systems being used and found beneficial uses of the system. The challenges for eHealth could further delay full implementation and use of systems. However, other countries such as the Netherlands found that eHealth has proven to be useful and beneficial as it allows people to become more independent over their health. There is security, privacy and legal issues with eHealth and its technologies, however, there are many solutions which can be implemented to reduce, prevent or remove these issues.

3.0 Materials and Methods

In this chapter, the creation of the secure medical wearable device and software application will be discussed and shown. The goal was to create a secure medical wearable device for use on humans, to allow for independent health monitoring, health awareness, and improved health care, through use of eHealth.

3.1 Secure Medical Wearable Device Creation

In this chapter, the method of creating the secure medical wearable device is shown and explained; including any hardware and software required.

3.1.1 Arduino Board, Shields and Sensors

In this subchapter, the hardware used will be discussed.

3.1.1.1 Arduino board and shields

The hardware that was used to create the secure medical wearable device was:

- An Arduino UNO development board. This will be used to collect, store and send the sensor readings, shown in figure 7. This was used because it provides the ability to connect multiple electronic components, to allow for communication with other devices, it is small and can be made portable.

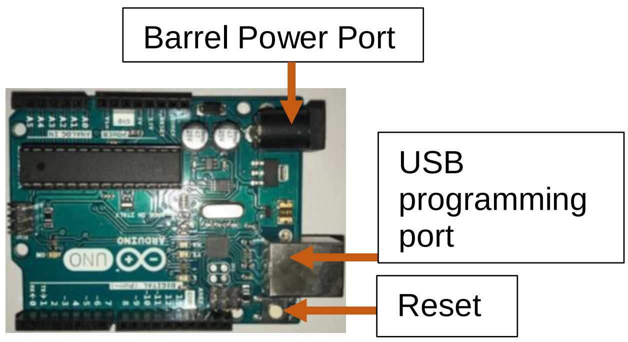


Figure 7 Arduino board

- Cooking Hacks eHealth kit. This contains a shield for the Arduino and the sensors required to record the body vitals. This kit is a development kit for research, so it is not Medically Certified. This kit provides sensors to measure human body parameters. These sensors are blood oxygen, heart rate, body temperature, blood glucose, galvanic skin response, electrocardiogram, electromyography and breathing rate. The shield collects the data from the sensors and processes the data for the Arduino to use, shown in figure 8.

Figure 8 eHealth shield V2.0

- A Bluetooth communication shield. The shield communicates with a device to send and receive information. Shown in figure 9. A small board containing a Bluetooth transmitter and receiver antenna. This shield allows the Arduino board to send and receive data via the Bluetooth frequency. The Bluetooth shield can be programmed to use encryption. In this project, this shield was used to communicate with an application, using encryption, to send data stored on the SD card to the application for processing and is also able to receive information from the application to interact with the SD card data. This shield uses Bluetooth version is 2.1 + EDR Class 2 and uses AES (Asynchronous Encryption Standard) for data connections. This shield is by XBEE. The signal range is between 10 to 50 meters depending on environmental factors.

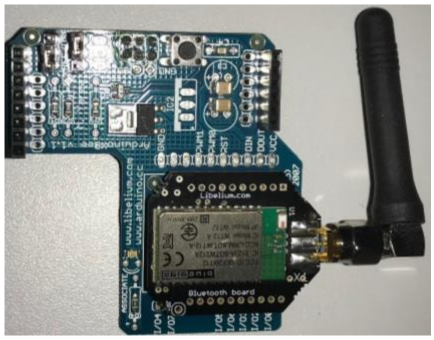


Figure 9 Bluetooth Communication Shield

- SD Card shield. Provides the ability to store the sensor information on a removable SD card and contains a real-time clock which provides the real-time and date, even when power to the device is removed. Shown in figure 10. This allows the Arduino to store and read information from an SD card. For this project, the SD card shield will be used to store the information collected from the sensors, the real-time and date the sensors data was recorded, onto a removable SD card.

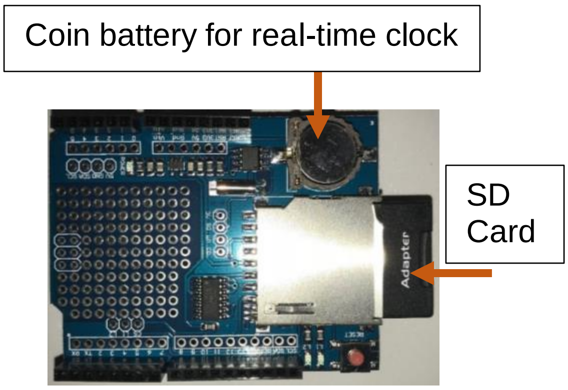


Figure 10 SD card Shield

  This method allows the user to be able to upload data from the device to the application via the SD card, if the user does not have a Bluetooth connection or cable. A 2GB SD card was used.

This monitoring device will be encased in a box for protection for both the circuits and user. A shield is a small electronic board that can be stacked on top of another board. The shields provide the Arduino board with more customisability and function.

3.1.1.2 Sensors

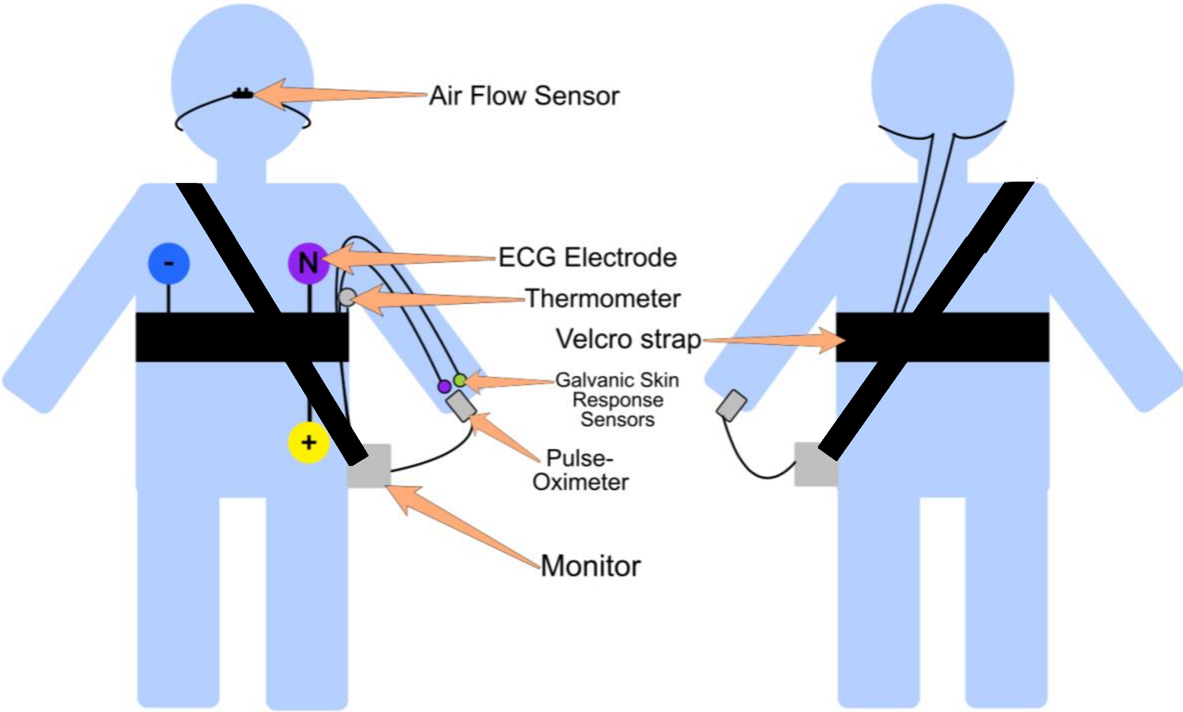


Figure 11 health monitor connected to the body

Figure 11 shows where all the sensors are attached to the human body.
In the cooking hacks eHealth kit, there are seven sensors that were used:

- **ECG (3 lead)/EMG:** this is a sensor that produces an electrocardiogram or an electromyography. This sensor is used to measure the electrical pulses from the heart (ECG), which is required as this could show abnormalities with heart rhythm and with the heart chamber functions. When using the ECG, three electrode pads are required to attach the electrodes to the skin; there are different types of electrode pads, but a pre-gelled electrode pad was used in the project.

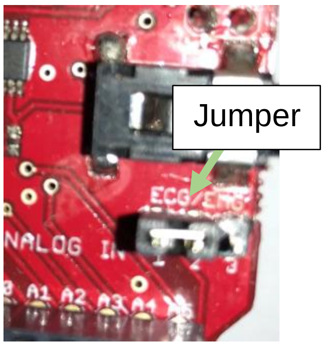


Figure 12 EMG/ECG jumper position

  The yellow electrode is positive and is placed on the lower left abdomen. The purple electrode is neutral and is placed on the upper left chest. The blue electrode is negative and is placed on the upper right chest. The electrode placement is shown in figure 11. EMG measures the electrical pulses generated by muscles, when the muscle cells are electrically activated, such as a change in position.

This sensor can help to identify any abnormalities in the activation levels, (Neuromuscular disease, assessing lower back pain, motor control). This uses the same sensors used for the ECG, so to use the sensors the jumper shown in figure 12 must be set to EMG. The blue electrode of the EMG attaches close to the elbow area using a pre-gelled electrode. The purple electrode attaches to the top of the forearm near the inner elbow and the yellow electrode attaches to the top of the mid-upper arm. These positions are shown in figure 13 (Cooking-hacks.com, 2017).

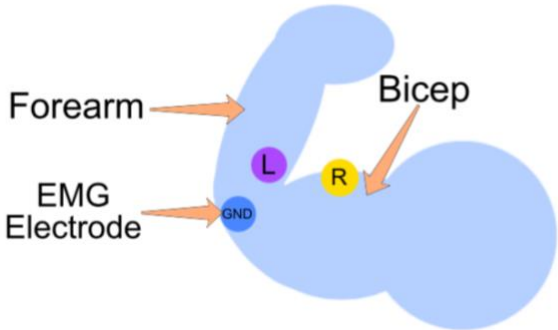


Figure 13 EMG electrodes positioned on an arm

- **Thermometer:** this is a thermistor which is placed at the axillary point on the body as shown in figure 12. This measures the body temperature. This sensor is required because a change in body temperature can be an indication of illness, for example, if the body temperature falls below 35°C the body will experience hypothermia or if the body temperature rises above 38°C the body experiences hyperthermia. This sensor measures temperature to one decimal place and has a maximum deviation of 0.1, so this sensor is very accurate.

- **Glucometer:** this is a device which measures the blood glucose content, shown in figure 14. It measures the blood glucose content in Millimoles Per Litre (mmol/L). The glucometer is a separate battery-operated device. The glucometer uses a lancet and lancet device to extract blood, the lancet is used to break the skin on the fingertip sides to extract blood, so it can be added to the test strips.

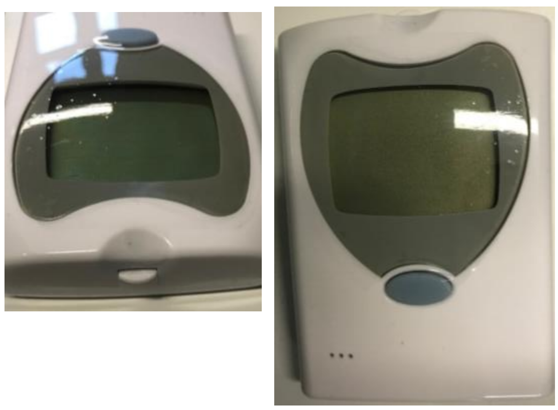


Figure 14 Glucometer

To use the glucometer the user pierces the skin using the lancet device on one of the points. The user then inserts the strip into the glucometer and adds the round drop of blood that comes from the skin, onto the strip. If the blood drop smears another blood drop must be re-drawn and added to a new test strip. The glucometer will then display the result. This sensor is required as it can help aid in removing type 2 diabetes or help aid people who have diabetes mellitus and require blood glucose monitoring.

- **Air flow:** this is a sensor made up of two thermistors. The sensor measures the breathing rate of the patient. This is done by measuring the change in temperature when breathing; breathing out will produce warm air and breathing in will produce cold air over the thermistors. This sensor is placed on the upper lip of the patient and the cables are placed over the ears and down the neck as shown in figure 11. This sensor can be useful for users affected by respiratory diseases or cardiovascular diseases.

- **Galvanic skin response:** this is used to measure the electrical conductance of the skin which changes with the skin moisture levels. This sensor can show an indication of psychological or physiological responses to the current environment, due to the sweat glands being controlled by the sympathetic nervous system. The skin resistance is also measured. Low resistance indicates the user is sweating whereas a higher resistance indicates the user has drier or dry skin. This sensor is often used in polygraphs (Cooking-hacks.com, 2017). This consists of two sensors placed on two separate fingers as shown in figure 11.

- **Pulse oximeter:** this is a non-invasive small battery-operated device which attaches to the index finger as shown in figure 11. The pulse oximeter contains a light source and light detector. The pulse oximeter is used to measure the oxygen levels in the blood, by finding the rate at which red light and infra-red light are absorbed. A photodiode, which is against the opposite side of the finger, is used to find how much of this light is absorbed, by the intensity of the light. Haemoglobin absorbs red light and oxyhaemoglobin absorbs Infra-red light (Ates, G. and Polat, K., 2012). The pulse oximeter displays the oxygen level as a percentage, a value below 90 percent is considered as abnormal.

3.1.2 Construction of the monitor

The device was built by stacking the data logger shield onto the Arduino board, stacking the eHealth shield onto the data logger shield and then stacking the Bluetooth communication shield onto the eHealth shield. All the sensors were then connected to their relevant ports, on the eHealth shield. The battery was then plugged into the device, providing power. The device was then enclosed in the constructed case. The sensors were threaded through the Velcro strap.

3.1.3 Programming the Medical Wearable Device

Programming the health monitor involved programming both the Bluetooth shield and the Arduino board.

The Arduino board was programmed to read information from the eHealth shield, process this information to securely store on the SD card, in the data logger, and send the stored information via Bluetooth to a computer to be processed by a web application.

3.1.3.1 Data logger and serial communication code

The Arduino code to store data, date and time to the SD card via the data logger shield is shown within appendices chapter 1.0 Arduino Code lines (7 – 14, 21 – 78 and 79 – 113). Lines (194 – 139) was used for serial communication with the device.

To use the data logger shield three libraries are required:

1. SD library
2. Wire library
3. SPI library
4. RTC library

These libraries were added to the Arduino IDE library folder and sketch. These libraries allow the Arduino to read and write files on an SD card and program the data logger clock to allow the real date and time to be stored.

The SD card was formatted to the FAT32 file system. A text file was created and stored on the SD card using the 8.3 naming system for a file. The file was named data00.txt, as the numbers can be incremented, so file identification can be done. These values change, when the date that sensor data is being recorded changes. The file contained a line with the patient database ID and date of birth separated by a vertical bar delimiter, as shown in figure 30. This allows for SD card identification with the device.

In lines (7 – 14), variables have been defined. Line 12 was used for communication with the data logger. Line 13, creates a file object to allow Arduino to read and write to files. Lines (25 – 30) was used to match the SD card to the file by comparing the first line present in the file with the line (line 35) stored on the health monitor. A mismatch will prevent data logging.

Error detection functions were added to the setup block, shown in lines (22 – 30 and 31 – 42). Lines (22 – 24) detects if the SD card is present or working. Lines (25 – 30) are used to verify if the specified file on the SD card exists and if the real-time clock is functioning. If these functions become true an error message is displayed in the serial monitor, notifying what the error is.

This error notification is achieved using the error function created, lines (123 – 127). This prints a string in the serial monitor and the while (1) loop prevents code from being executed.

The code within void loop block, lines (45 – 113), was used to store data to the SD card. Lines (54 – 78), retrieves the date (MM, DD, YY) and time (HH: MM) from the real-time clock and stores this on a new line within the file on the SD card. Lines (79 – 106), was used to gather sensor readings from the eHealth shield and append them to a string variable named output. This variable is then printed to the file once all sensor readings have been collected. The serial is flushed to remove any data remaining in serial, so duplicates are not produced. A one second delay is introduced before the sensor readings, so that one bunch of data is collected. Sensor readings are collected every minute, by using the ‘if’ statement at line 80.

Lines (128 – 139) was used to allow for communication with the health monitor. These lines contain a serial event block. This block is called when data is received via the serial port. When the health monitor receives a ‘1’, it sends a message notifying the user that data logging has stopped, and file data will be sent. The file on the SD card will be opened and each line sent to the serial port. If the device receives ‘2’ it will erase the sensor data on the SD card file and start data logging.

3.1.3.2 Coding for eHealth shield

The Arduino code to read the sensor data from the eHealth shield can be found in appendices chapter 1.0 Arduino Code lines (79 – 106 and 114 – 122).

An eHealth library is required to provide the Arduino IDE with code to communicate with the eHealth shield, this was installed into the Arduino IDE. Library version 2.0 was downloaded from the Libelium webpage, an added to the Arduino library folder.

The pulse oximeter requires extra code, as the pulse oximeter does not provide continuous data and so needs to interrupt the Arduino with its results.
A PinChangeInt.h library was added to the Arduino IDE library, this provided the ability to interrupt the Arduino. The interrupt was connected to pin 6 of the Arduino, where the pulse oximeter will be sending the data.

The interrupt will be triggered on a rising edge of a digital signal and will call the function readPulsioximeter(). This function gets 50 measures from the pulse oximeter to reduce latency, this is shown in lines (114 – 122).

A baud rate of 115200 is used, this is the speed at which the Arduino sends information through the serial port. A string was created to output the sensor data, this was created by concatenating the sensor data together forming a single string. Each sensor output required a delay to allow the programme to receive and process the information in time. So, a delay of 10 milliseconds was added between each concatenation of sensor data.

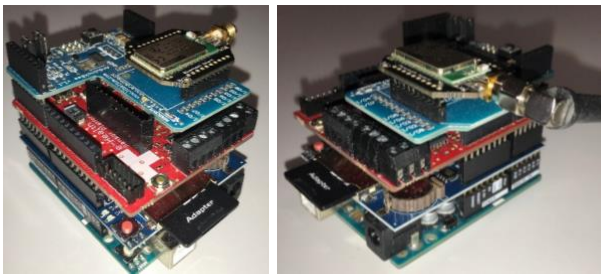


Figure 15 boards stacked

The e-Health shield was stacked onto the Arduino board as shown in figure 15. A sensor was then attached one at a time to the e-Health shield to be tested for functionality; this was done for each sensor. The code was tested by looking for the sensor data in the Arduino IDE serial monitor, as the code outputted a string of sensor data; this code is shown in lines (79 – 106). Once, all sensors completed the test, they were all attached to the e-Health shield.

3.1.3.3 Coding for Bluetooth shield

The Bluetooth shield was then programmed and the code for communication via Bluetooth was then written for the Arduino.

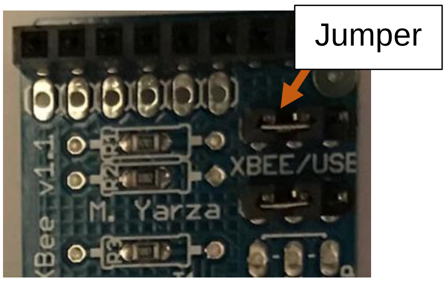


Figure 16 Bluetooth jumper position in USB (XBEE/USB)

The Bluetooth shield was stacked on top of the e-Health shield as shown in figure 15. The function of the Bluetooth shield was tested.

To test the Bluetooth communication shield, the jumpers as shown in figure 16 should be put to the XBEE position; this changes the path of serial communication from the USB to the Bluetooth transmitter and receiver. The Bluetooth shield was then connected to the computer via Bluetooth. If the device is successful, then the shield will have connected, and the Arduino IDE serial monitor will have the output string printing.

To programme the Bluetooth shield, the jumpers shown in figure 16, were both set to USB. The ATMEGA chip was then removed from the Arduino board, to allow direct connection to the Bluetooth shield as shown in figure 17. This setup is known as gateway mode.

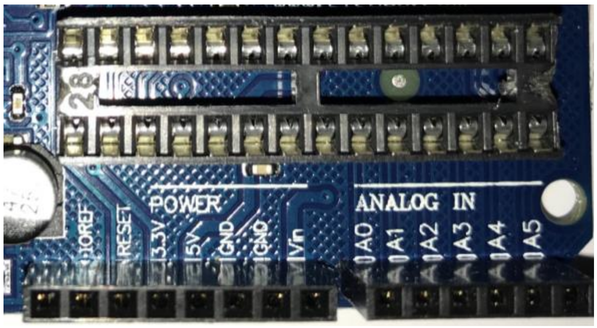


Figure 17 Arduino board without ATMEGA chip

To connect to the Bluetooth shield, an AT command is sent to the shield via the Arduino IDE serial monitor, at baud rate 115200bps, as shown in figure 18. A Bluetooth was set to be visible by typing 'SET BT PAGEMODE 4' and a pin was set by typing 'SET BT AUTH * 1234'. The jumpers were placed to XBEE position allowing for Bluetooth communication.

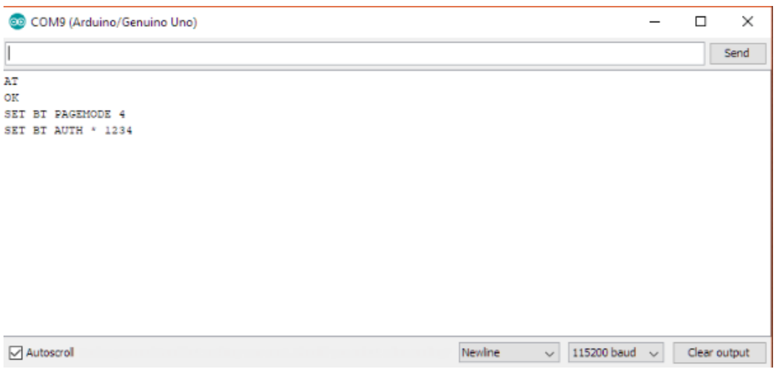


Figure 18 Arduino IDE serial monitor Bluetooth commands

3.1.4 Communication with the device

A communication with the health monitor is required to transfer sensor data from the device via Bluetooth or USB cable. This was done using Processing IDE, to send commands to the Arduino and code for the Arduino to respond to the commands.

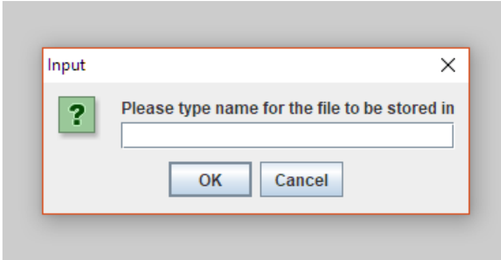


Figure 19 Processing Start Screen

In the code, in appendices chapter,

2.0 Processing Code, was used to connect to the health monitor and send commands to retrieve sensor data.

Line (20) was used to create a manual connection to the health monitor. The manual connection requires the port number to connect to the Arduino. Once connected, the buttons can be used to send commands to the Arduino.

Lines (20 – 22) creates a pop-up window which asks the user to name the file that the data will be written to. This file is stored inside the Processing program file.

Lines (14 – 31) creates a user interface. With this interface, the user can send commands to the health monitor via clickable buttons as shown in figure 20. Button one sends a ‘1’ to the Arduino, this causes the Arduino to send data from the SD card file. The data received is stored in the named file. Button 2 closes the file to save the data. Button 3 sends ‘22’ to the Arduino which removes the file on the SD card, creates a new file and restarts the sensor recording process.

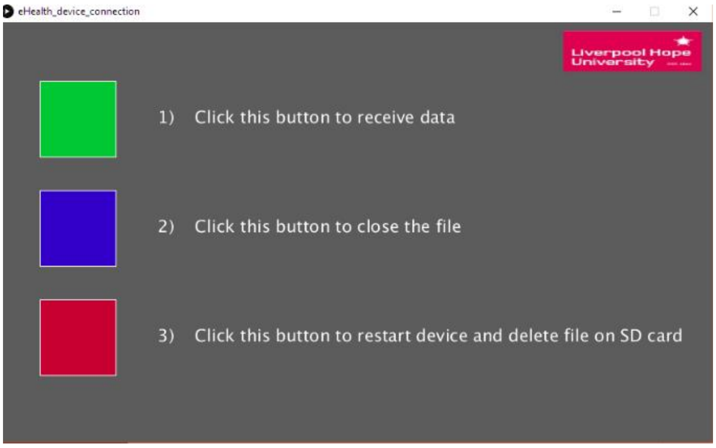


Figure 20 Processing Program Options

3.1.5 The medical wearable Case, Velcro Strap, and Battery

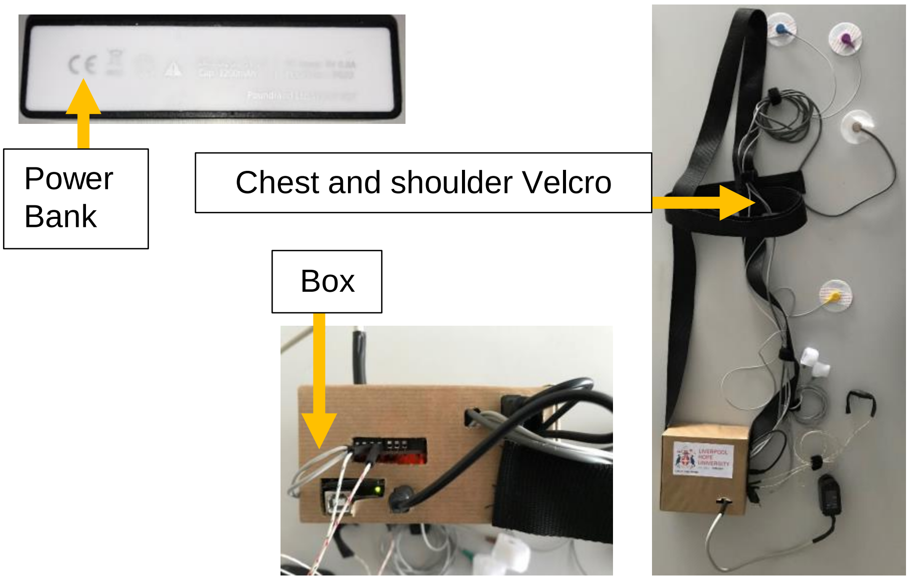


Figure 21 the monitor case, Velcro strap and battery

3.1.5.1 Battery

A power bank was used which provides the Arduino board with 5 volts and 1 Ampere (A) input, this is shown in figure 21. This power bank can be recharged using a micro USB cable providing 0.8(A) and 5v input. The power bank capacity is 1200mAh and the device requires 94 mA and 5 volts to run. The power bank has an LED, solid red indicates the power bank is charging, solid blue indicates the power bank is providing power and flashing red indicates the power bank is charging.

3.1.5.2 Case

The case was created by cutting slots out in a cardboard box, to allow access to, the sensor cables, the power bank ports, the SD card and the ports on the Arduino. A shoulder strap was added to the case, using a sewing machine. The case contains all the boards and power bank. The case helps to protect the device. The case is shown in figure 21.

3.1.5.3 Velcro Strap Created

The Velcro strap was created by using a material called webbing and sewing Velcro to the strap. The strap was made to go around the shoulder, around the chest and attaches to the sides of the box, as shown in figure 21.

3.2 Web Application

A secure web application was created to communicate with the secure medical wearable device and display the data from the device. The web application will be created using a combination of coding languages; MySQL, HTML, CSS, PHP, and JavaScript.

3.2.1 Database Creation and Server

For the client side of the web application, a database was created to store patient information and the data created by the health monitor device. This was created using a WampServer version 3.1.0 (Bourdon, 2018). As this allows for storage of a MySQL database, web application, and communication between them. The WampServer is using a MySQL database server version 5.7.19, Apache version 2.4.27 and PHP version 5.6.31.

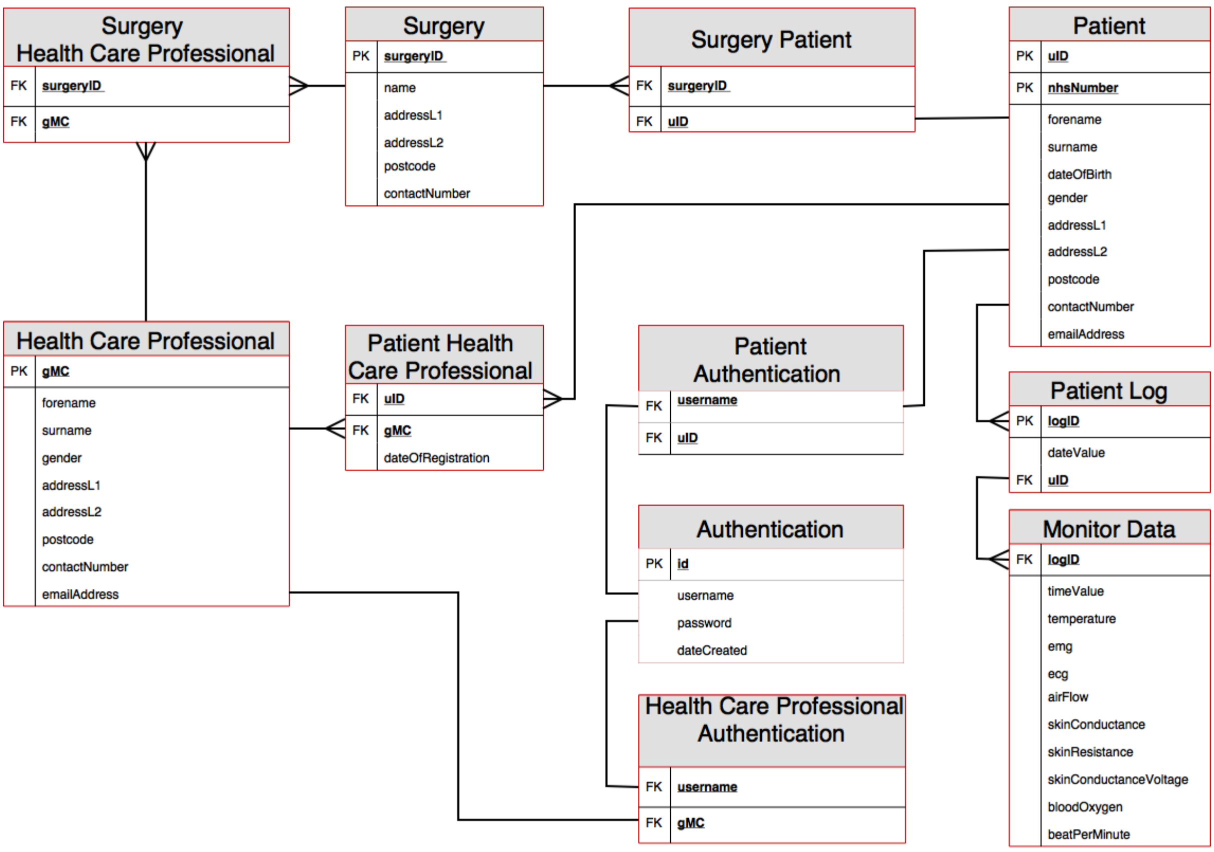


Figure 22 Database logical design

The database was created using MySQL. The database was designed before being implemented, using a logical design. The logical design contains all the tables required, their fields, their keys and the relationships between the tables.

Tables were created to store data for surgery details, relationship between surgery and patient, relationship between surgery and health care professional, patient details, health care professional details, a relationship between the health care professional and patient, authentication, a relationship between the health care professional and authentication, a relationship between authentication and the patient, patient log and monitor data. This logical design is shown in figure 22; a many relationship is shown using a split line, for example, the relationship between patient log and monitor data is one too many.

These tables allow the web application to provide security of data, display and retrieve appropriate information.

A database was created and named health_database. MySQL commands were used to create the database, in the WampSever MySQL console. The tables in figure 22 were then created, using MySQL commands, and added to the database; this is shown in Database Tables MySQL of appendices chapter 4.0 MySQL Code. Random data was then created and added to the database to test the database and webpage, this is shown in Database data of appendices chapter 4.0 MySQL Code.

When inserting usernames and passwords, each password was hashed as shown in line 124 of appendices chapter 4.0 MySQL Code, Database Data Inserts MySQL. This uses SHA2 hashing at 512 bits.

Users were created to restrict the MySQL commands for the type of user logged in, shown in Database Users MySQL of appendices chapter 4.0 MySQL Code.

The authentication table and its relation tables, patient authentication and health care professional authentication, allow the website to keep data secure and provide a personal experience for patients and health care practitioners. The authentication table holds username data and passwords, which the web application uses to prevent wrong data being shown and provide data privacy.

3.2.3 Website

This web application was created using HTML, CSS, PHP and JavaScript languages. These languages allow the web application to become dynamic and establish server communications.

3.2.3.1 Login Screen

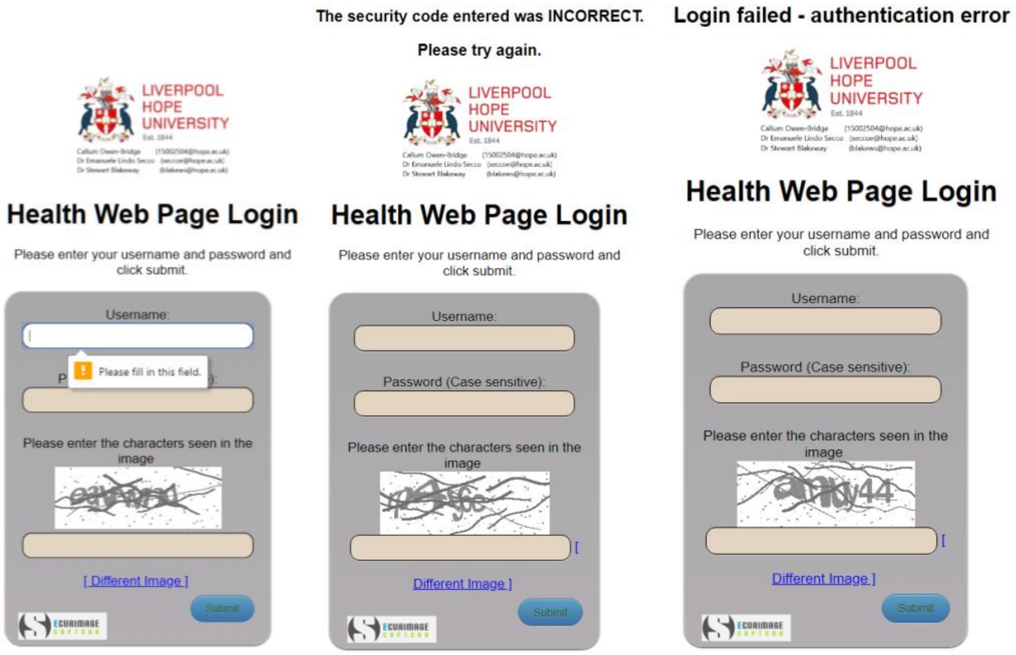


Figure 23 Login page

The login page shown in figure 23 was created using PHP, HTML and CSS, shown in appendices chapter 3.0 Web page Code within login_form.php. The login design was created using CSS form (Cascading Style Sheet), the inclusion of this form can be found on line 80.

The centre image of figure 23, shows the login page the user views when accessing the webpage. The submit button is blue and when the cursor hovers over it or clicks the button, the button turns green to indicate an action was taken place. This effect was created using the CSS form.

The password is hidden by dots, when the user types in the password field, this was achieved by making the input type password, found on line 106. A post method is used when the user clicks submit because it is more secure then the get method and the PHP section of the webpage requires this information to see if the credentials match those in the database.

Function $_SERVER["PHP_SELF"] is used as the credentials entered must be posted to the same page for the PHP to process and if an error occurs the user will be notified. A function called htmlspecialchars() is used as this prevents the webpage from being exploited. This can be found in lines (97 – 119).

If the user enters incorrect credentials the user is notified as shown in the third image in figure 23, this can be found on line (59). If the user leaves a field blank they are notified of this when they click enter, as shown in the first image of figure 23. This can be found in lines (101 and 106).

Before the user can submit this must enter the characters shown in the image below the password field. This is a secure captcha image, the code for this can be found on lines (62 – 66 and 109 – 117). If the user enters incorrect values an error message is displayed as shown in the second image of figure 23, this can be found on lines (62 – 66). The secure image was created by phpcaptcha, (Phillips, D., 2005).

This login page is used by both the patient and health care professional. The website determines the user by looking for either an ‘h’ or ‘p’ as the fourth character if the username entered.

3.2.3.2 Patient Welcome Page

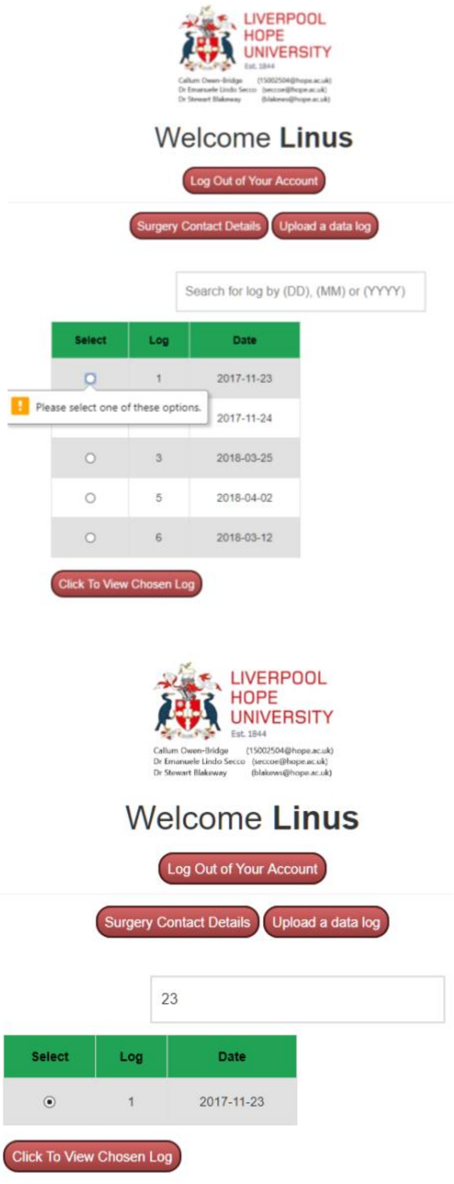


Figure 24 Patient page

This can be found in appendices chapter 3.0 Web page code, patient_welcome.php.

There are two blocks of code a PHP block and an HTML block. The PHP block is used to connect to the database and complete server-side tasks such as getting data. When the PHP code receives posted data from the HTML part of the webpage, a MySQL statement is executed searching the database for the credentials entered by the user. If there is a match the username is copied into a variable from the database. The fourth character of the username is searched for either an 'h' or 'p' character. If 'h' is present the user is directed to the HCP web page if 'p' is present the user is directed to the patient webpage. If not match occurs an error message is displayed. This can be found within lines (3 – 35).

Lines 3 – 16, creates a database connection and session. The session allows a variable set from the previous page to be used, shown on line 20. Lines 25 – 35, contains MySQL statements and database tasks. They retrieve logs and patient name.

Two CSS files are used. welcome_form_CSS.css creates the design for the buttons and a bootstrap CDN (Content Delivery Network) CSS form version 4.0 (Joshua Mervine, 2018), shown on line 50, is used to create the style for the text and headings. An internal CSS is also created to style the log table.

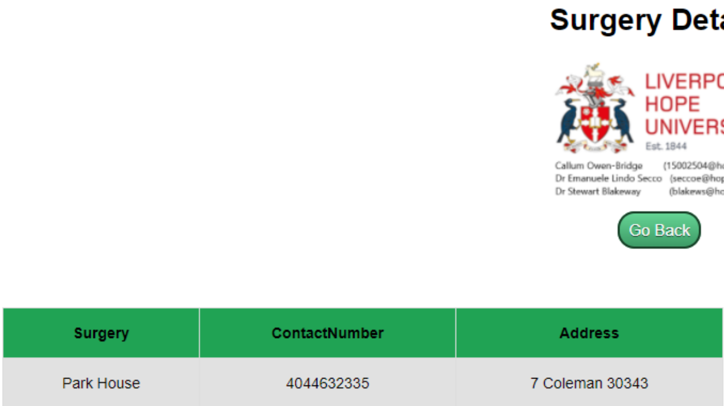
When a patient logs in they see the first image in figure 24. This is used in a MySQL statement to find the patient's ID number. Using the ID number, the patient's logs and forename is gathered from the database.

Buttons were created for surgery details, log out of the user account, upload a log and to view a chosen log, as shown in the first image in figure 24. Clicking surgery details takes the user to a webpage containing a table with the details of the surgery they are in as shown in figure 25. Clicking log out logs the user out and directs them to the login page. Clicking 'Upload a data log', will direct the user to the upload webpage. Clicking 'click to view chosen log' takes the user to graphical data webpage if a data log is selected. The code for these buttons is shown within lines (99 – 112).

A search bar was created to find data logs more easily, the code for this is within lines (114 – 158). Any part of the date can be entered, into the search bar refining the number of data logs shown, as shown in the second image of figure 24. when data is entered into the search bar function searchLogTable() is called which uses JavaScript to get input data change the characters to upper case, search for the input data in logs table in each row. The search bar is created on line 116. The search is restricted to the data column, shown on line 145. Each row which is not relevant to the input data is temporarily removed, shown on line 153. Lines 134 – 158, is JavaScript code used to search the table created for the value entered in the search bar. It searches column 2 which is specified on line 145.

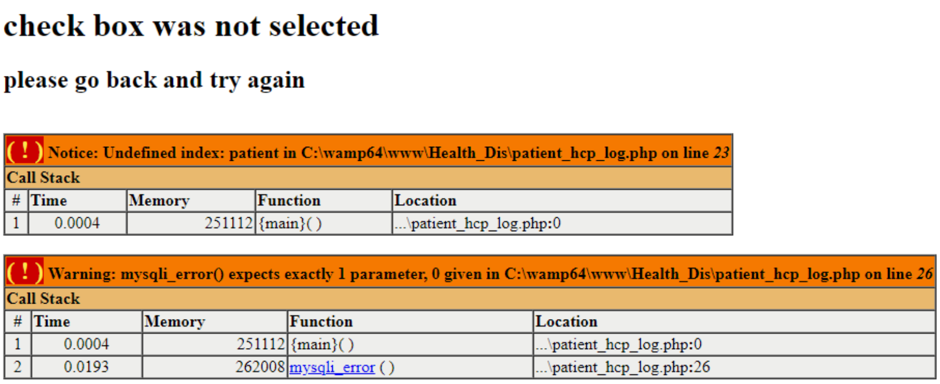
A data log table was created, shown in lines 114 – 133. This table contains all the logs the patient has uploaded to the database. Each row of the logs table is highlighted when the cursor hovers over it and each row is a different shade of grey to allow the user to differentiate between the rows.

Radial buttons are used within the logs table to select a data log. The table gains a scroll bar for the y-axis when many logs are present; this is done automatically. If the user clicks the ‘click to view chosen log’ button without selecting a log they will be shown an error message on the graphical data webpage as shown in figure 26, shown on line 15.



Surgery	ContactNumber	Address
Park House	4044632335	7 Coleman 30343

Figure 25 Surgery Details page



check box was not selected

please go back and try again

(!) Notice: Undefined index: patient in C:\wamp64\www\Health_Dis\patient_hcp_log.php on line 23				
Call Stack				
#	Time	Memory	Function	Location
1	0.0004	251112	{main}( )	...\patient_hcp_log.php:0

(!) Warning: mysqli_error() expects exactly 1 parameter, 0 given in C:\wamp64\www\Health_Dis\patient_hcp_log.php on line 26				
Call Stack				
#	Time	Memory	Function	Location
1	0.0004	251112	{main}( )	...\patient_hcp_log.php:0
2	0.0193	262008	mysqli_error ( )	...\patient_hcp_log.php:26

Figure 26 Selection error page

3.2.3.3 Upload Webpage

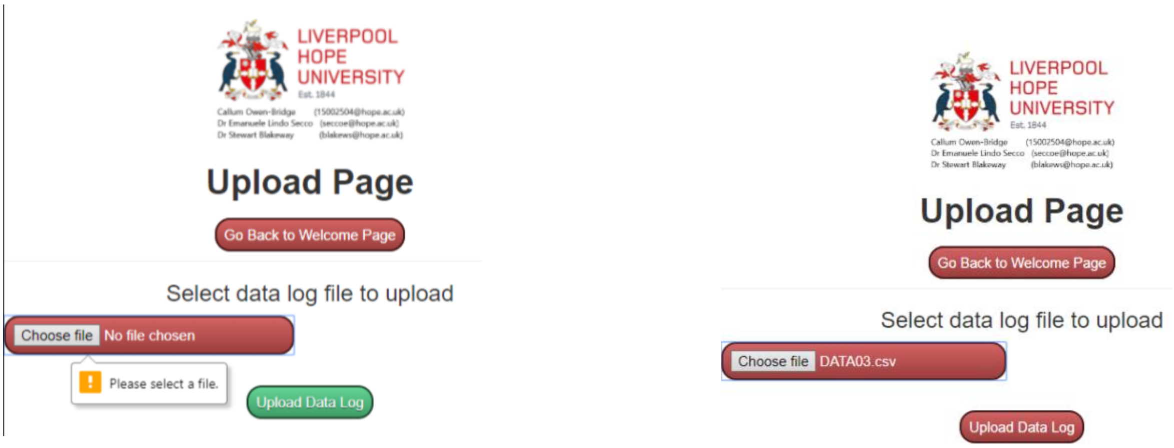


Figure 27 Upload page

The code used to create the upload webpage is shown in appendices chapter 3.0 Webpage Code, upload.php. This webpage is accessed from the patient welcome page. This page allows the user to browse for a CSV (Comma Separated Values) file and upload the file to the server temporarily. The data is read from the file and stored in the database tables. Once stored the server file is removed.

Lines 20 – 74, copies the selected CSV file and adds the file to server folder named uploads. A message is displayed if the file currently exists shown on line 31. Line 36 – 40 restricts the file type uploaded to txt and csv.

Once the file uploads, a message is displayed as shown in figure 27. Lines 61 – 75, opens the file within the server and reads every line of the file into an array named lines. Lines 77 – 109, creates a new patient log and adds the data from the file to the server database. Once complete the file is removed shown in lines 110 and 111.

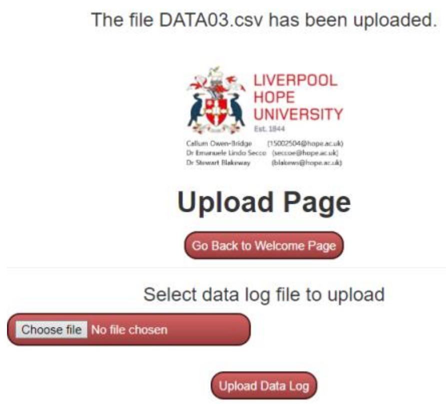


Figure 28 Upload notification

Lines 117 – 154, is HTML code used to design the webpage. Lines 149 – 152, creates the option to browse for a file and submit the file chosen. Line 150, restricts the user from uploading no file, as shown in figure 28.

3.2.3.4 Graphical Webpage

Figure 29 shows the top section of the graphical data webpage. This is the graphical_data.php file, containing code blocks for PHP, HTML and JavaScript. This webpage displays data log data in graphical form and provides links to a symptom checker and medical information. The code for this webpage is in appendices chapter 3.0 Webpage Code, graphical_data.php

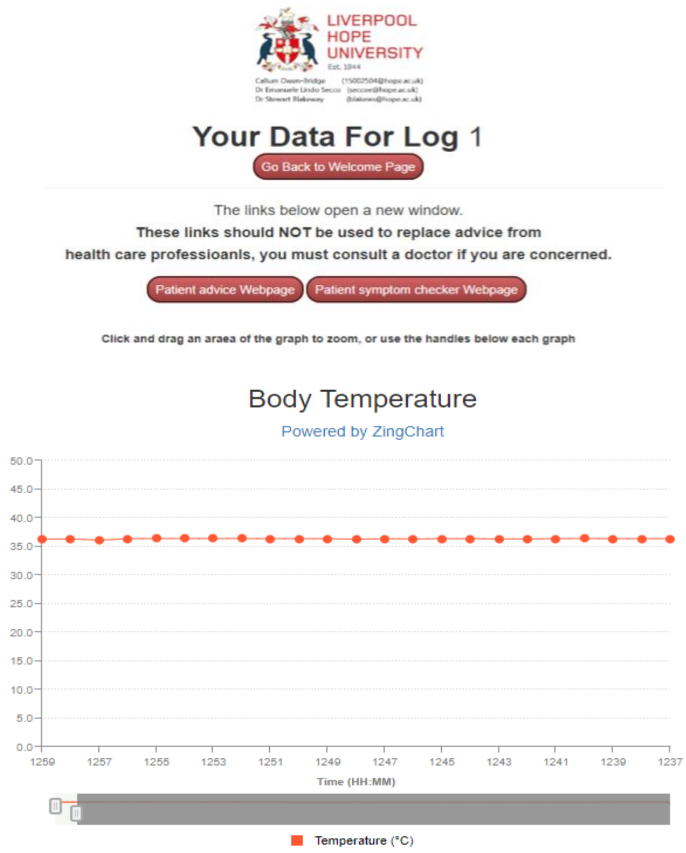


Figure 29 Graphical Data page

In the PHP block, a connection to the database is made. Once connected, the posted logID from the patient webpage is used to find the relevant monitor data for the patient and log chosen. The logID value is used within the title of the page to indicate to the user which log data they are currently viewing. This can be found within lines 9 – 20.

Lines 18 – 44 gathers monitor data for the patient and log ID chosen. Line 18 and 19 uses MySQL commands to retrieve this information and stores the data into a variable named, monitor_data, line 19. The monitor data is entered in separate arrays for each sensor, shown in lines 31 – 43. The number of rows found in the database is stored in a variable, which is used within the HTML block, in line 20.

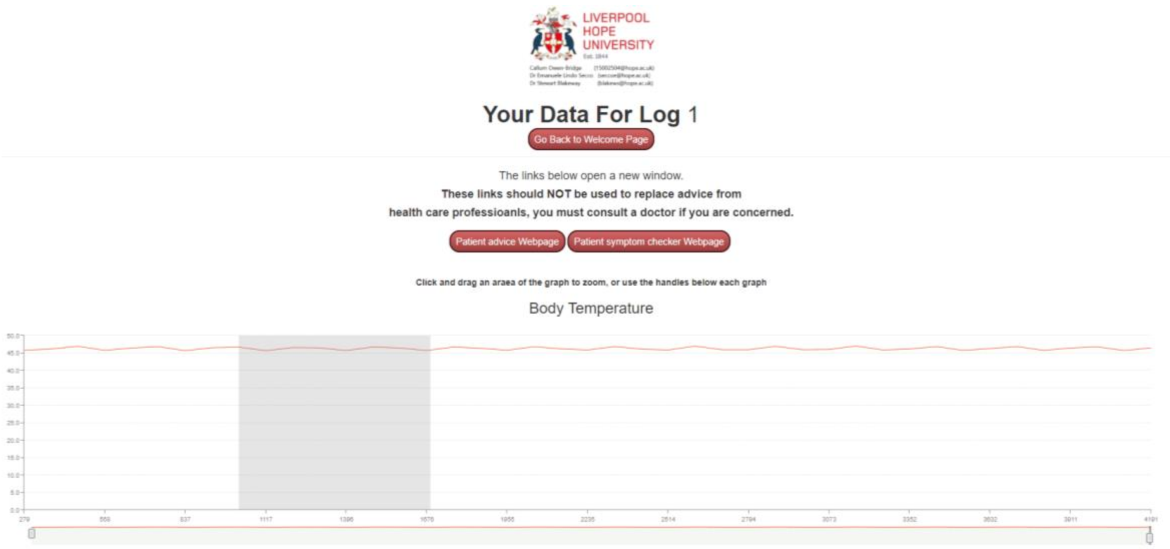


Figure 30 Selection tool

The HTML block uses a button_form_CSS.css file to design the buttons and webpage. A CDN CSS form is used for text styling. A CDN JavaScript library called zingchart is used to create charts on the webpage. This can be found on lines 52 – 54. A PHP block within the HTML body is used to direct the user to the correct page when clicking the back button. This was achieved by searching for an ‘h’ or ‘p’ as the fourth character if the username. This is because the patient and HCP use this webpage to view monitor data. This can be found on the line (152 (lines 1 - 3)).

Lines 152 (vii – xvii), are links to a symptom checker and medical information web pages. This also includes a message notifying the user that these webpages must not be used for diagnoses.

Lines 152 (xviii - xxi), is a window prompt to notify the user, that JavaScript must be enabled for the page to function properly. Lines 152 (xxiv - xli), are page dividers for each graph created for each sensor.

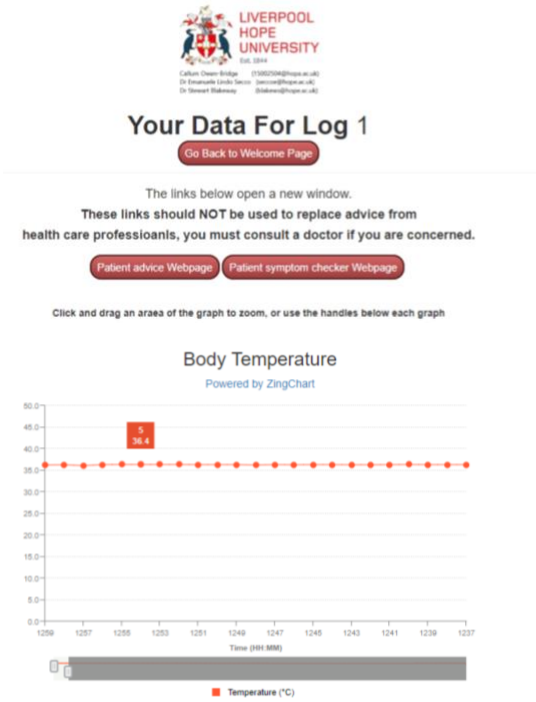


Figure 31 Graph values

In the JavaScript block within the head tag, arrays are created for each sensor that is used on the monitor. This is to copy each value gathered from the database by the PHP block. This process occurs in a PHP block within the JavaScript block. A for loop is used to iterate through the rows gathered from the database. This can be found within lines 58 – 72.

A line graph for each sensor is then created using the JavaScript library. Each sensor type is plotted against time (HH:MM). As shown in figure 29. If the user hovers the cursor over a specific data point on the graph, its data value is displayed as shown in figure 31. The code for creating the graphs can be found on lines 73 – 150. This code creates the graphs and allows the graphs to be customised.

The graphs have an ability to zoom in and pan across the graph as shown in 30. The user can click and highlight a section of the graph to zoom in and view the data points for a more precise time. The user can then pan across the graph using the handles on the selection tool below the graph. These handles can be expanded or contracted. This function is added by including a preview block within the graph, an example is on line 74 (ix – x).

Lines 94 – 149, was used to bind the graph data to a div tag and dimensions. This renders the graphs on page loading.

3.2.3.5 HCP Welcome Web page

This is a welcome page created for the HCP. The code for this can be found in appendices 3.0 webpage code, hcp_welcome.php. This file contains a PHP and HTML blocks of code. In the PHP code block, a session is created to get the username variable, to display on the webpage. A connection is made to the database.

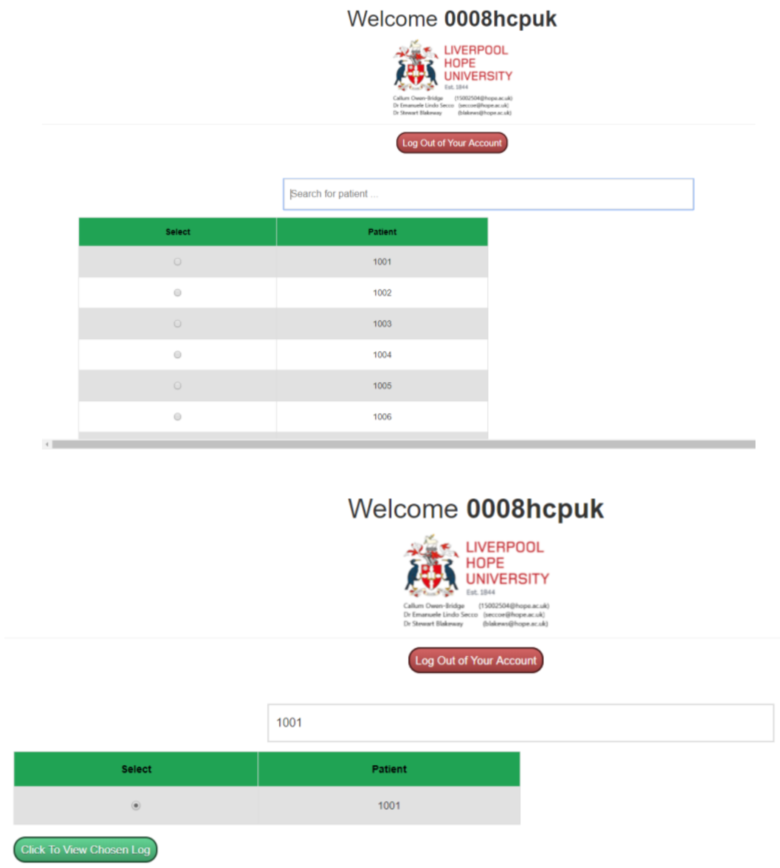


Figure 48 HCP pages

A MySQL statement is executed to get all the patients present at the surgery, shown on lines 19 and 20. This gathers the patient ID numbers in the database.

In the HTML block, two external CSS forms are used. welcome_form_CSS.css is used to design the page and buttons and a CDN CSS form is used, (Joshua Mervine, 2018), for text styling. This is on lines 35 and 36. An internal CSS form is used to design the table, shown in lines 47 – 78. Line 68, alters the shade of each line, as shown in figure 32. Line 69, darkens the line to indicate which line the cursor is hovering over.

A search bar is created to search for specific patient IDs, as shown in the second image of figure 32. A table of patient IDs is created with a radial button. This is shown on lines 85 – 124.

When the HCP selects a patient and clicks ‘click to view chosen patient’, they will be directed to the patient_hcp_log.php webpage shown in figure 33.

This page displays the logs created by the chosen patient. Otherwise, a message will appear indicating the patient has no logs present shown in the first image of figure 29. If no patient is selected an error message will appear as shown in figure 26.

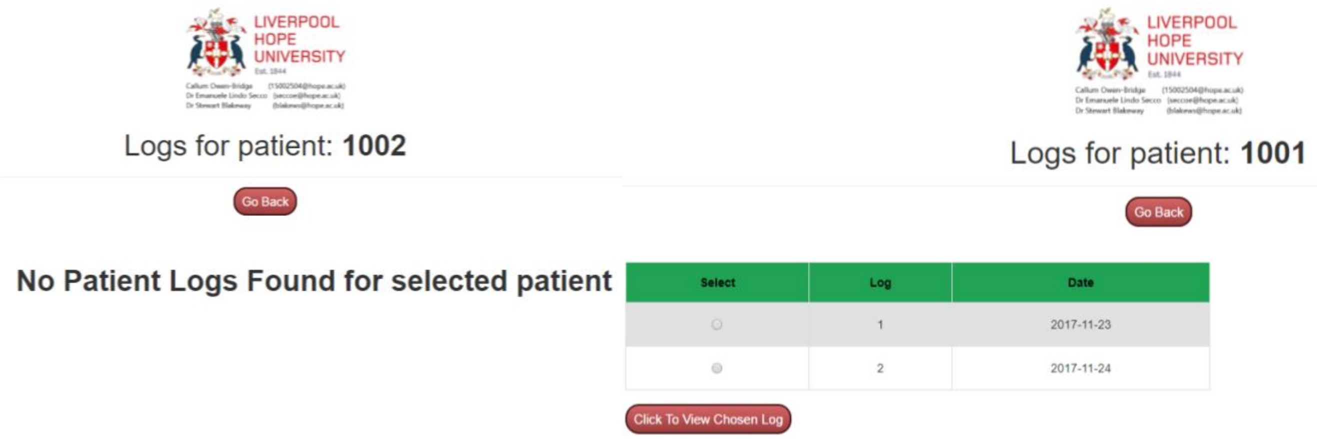


Figure 51 HCP pages

3.2.3 Client web application

Figure 34 shows the possible paths the client could take on the website. The client logs in using their provided password and username. If the credentials are correct the client will see their unique welcome page. Here the user can view all the data that they have logged. The user could then view their surgery details or select a log and view the chosen log data in a graphical form or the user can log out of their account.

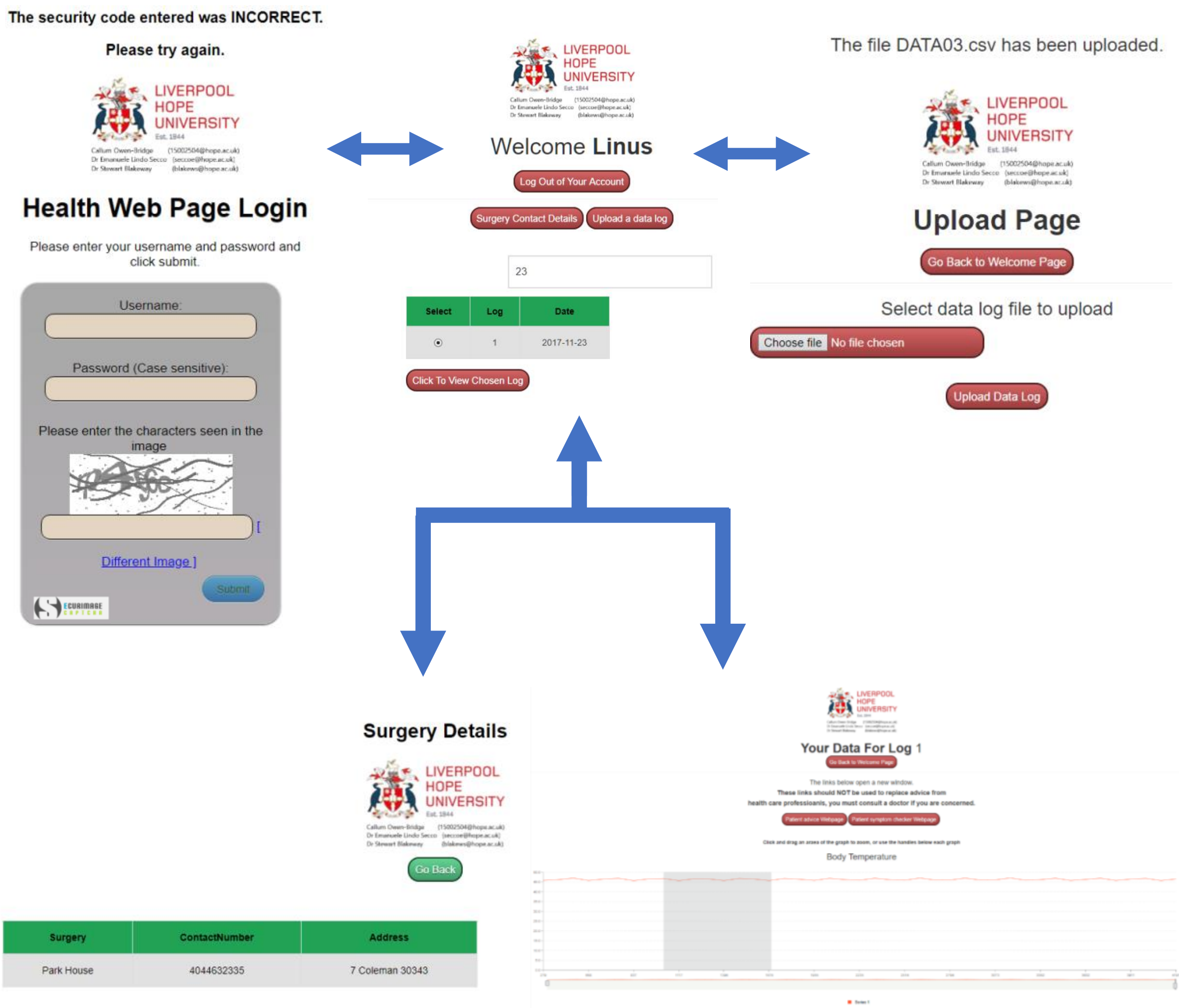


Figure 52 Client webpage user interface and possible paths taken

3.2.4 Practitioner Web Application

Figure 35 shows the possible paths the health care practitioner (HCP) could take on the website. The HCP logs in using their provided password and username. If the credentials are correct the HCP will see their welcome page. Here the HCP can view all their patients. The HCP can select a patient or log out of their account. When the HCP selects a patient, they can then view that patient's logs, from there the HCP can either go back to the patient select page or select a log. When a log is selected the chosen log, data is shown in graphical form. From the graphical page, the HCP can go back to the patient selection page.

Figure 53 Practitioner webpage user interface and possible paths taken

3.3 Summary

A secure medical wearable device was created using an Arduino board, data logger shield, eHealth shield and Bluetooth shield. The device was created to securely store recorded data with the real time it was recorded, using a data logger shield, SD card and encryption. The Bluetooth shield was added to allow the device to communicate via Bluetooth. The eHealth shield was added to have the ability to record human body parameters using many sensors such as ECG, pulse oximeter, and thermometer.

4.0 Results

In this chapter, the results of creating the secure medical wearable device and web application will be discussed.

4.1 Secure Medical Wearable Device Completed

The secure medical wearable device was created successfully.

The device can measure the human body vital parameters and store the data with the date and time they are measured. The data was successfully stored on the SD card, as shown in figure 36. Figure 36, shows a small section of the data recorded by the health monitor, it stores the date and time, temperature, EMG, ECG, Air Flow, galvanic skin response, skin conductance, skin resistance, skin conductance voltage, beats per minute and blood oxygen levels. This data is accessible using a computer to connect to the wearable device via Bluetooth connection, USB cable or accessing the file on the SD card.

DATA01 - Notepad

File Edit Format View Help

```
1001|1957/04/11

2018-04-11,2009,31.80,395,1.92,392,28.61,34950.46,1.93,0,90
2018-04-11,2010,31.75,396,1.93,394,28.71,34831.46,1.94,0,90
2018-04-11,2010,31.25,408,1.98,403,29.39,34020.62,1.97,0,90
2018-04-11,2011,35.72,397,1.94,385,19.42,51745.07,1.47,0,90
2018-04-11,2012,35.81,394,1.93,382,19.42,51745.07,1.47,0,90
2018-04-11,2013,36.25,388,1.90,376,19.03,52813.63,1.45,0,90
2018-04-11,2020,32.12,390,1.90,377,27.15,36838.32,1.86,0,90
2018-04-11,2021,31.77,396,1.93,395,29.00,34479.27,1.95,0,90
2018-04-11,2022,36.19,387,1.89,379,19.13,52542.37,1.45,0,90
2018-04-11,2023,36.22,386,1.89,378,19.03,52813.63,1.45,0,90
2018-04-11,2024,36.22,386,1.89,377,19.03,52813.63,1.45,0,90
2018-04-11,2025,36.22,385,1.88,376,19.03,52813.63,1.45,0,90
2018-04-11,2026,31.51,410,2.00,401,29.00,34479.27,1.95,0,90
2018-04-11,2027,36.28,394,1.93,379,19.23,52273.88,1.46,0,90
2018-04-11,2028,36.25,383,1.87,374,18.93,53087.70,1.44,0,90
2018-04-11,2029,36.25,383,1.87,374,18.93,52813.63,1.45,0,90
2018-04-11,2030,36.25,383,1.87,373,18.93,53087.70,1.44,0,90
2018-04-11,2112,31.88,397,1.93,371,27.15,36838.32,1.86,0,90
2018-04-11,2113,36.28,379,1.85,353,18.15,55387.11,1.40,0,90
2018-04-11,2114,36.28,379,1.85,354,18.15,55088.86,1.41,0,90
2018-04-11,2115,31.83,397,1.93,374,27.34,36574.90,1.87,0,90
2018-04-11,2116,36.31,388,1.90,358,18.35,54793.79,1.41,0,90
2018-04-11,2117,31.83,397,1.93,374,27.34,36574.90,1.87,0,90
2018-04-11,2118,36.28,380,1.85,354,18.15,55387.11,1.40,0,90
2018-04-11,2119,36.28,380,1.86,354,18.15,55387.11,1.40,0,90
2018-04-11,2120,36.31,390,1.91,358,18.35,54501.87,1.42,0,90
2018-04-11,2121,36.28,380,1.86,353,18.15,55387.11,1.40,0,90
2018-04-11,2122,36.28,379,1.85,353,18.15,55387.11,1.40,0,90
2018-04-11,2123,36.28,379,1.85,352,18.05,55387.11,1.40,0,90
```

Figure 54 Snippet of sensor data stored on SD card

The data was not removed or overwritten unless told to. If the file was asked to be removed, it was removed from the SD card. The SD card and device are matched using patient ID and date of birth as authentication. This is to prevent the wrong device or SD card from being used. If the data did not match the device would not write to the SD card. The File can be saved on the computer and uploaded to the database using the web application.

A box was created to hold and protect the circuit boards of the device. The box has an over the shoulder and chest strap. The chest strap was created to hold and keep cables tidy around the body, this is shown in figure 21.

Tests on the device consumption shown that the device requires 94 milliamps and 5 Volts to run. Therefore, the battery used is inadequate and can only run the device for around seven minutes.

The processing application works effectively. The program was opened and started, each button completed the task it was created for.

4.2 Web application Created

The web application was created successfully. The patient can log into their account, view their logs and select a log to see in graphical form. The web application keeps data secure, private and only accessing data for the specific user logged in. The user can view surgery details and are able to upload data log files to the database. The HCP can access and view patient logs for specifically chosen patients. The HCP is also able to view this data in graphical form.

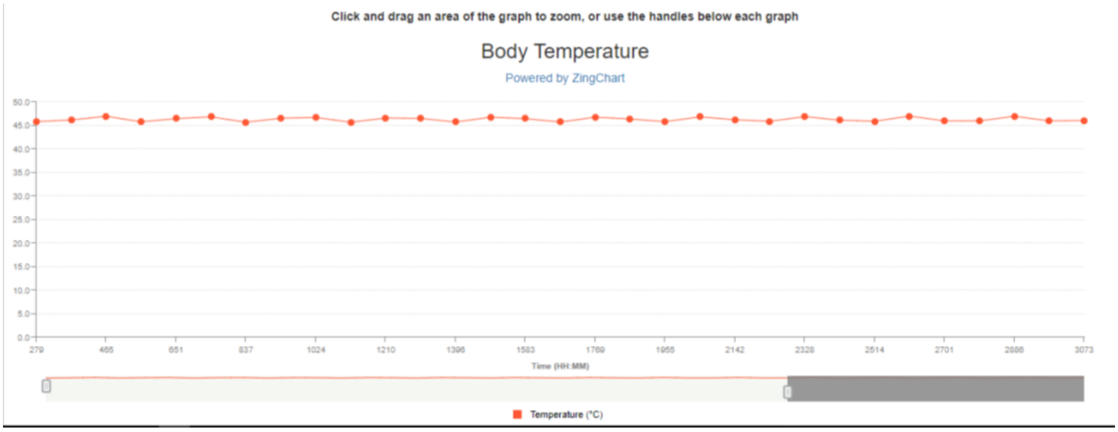


Figure 57 Temperature graph

Figure 37 shows, the raw data recorded for body temperature. Each circle represents a data point recorded. Temperature is plotted against time, to allow the user to understand when the value was recorded and allow them to solve any issues they may have had. The line graph created allows the user and health care professional to view connections between data points, identify patterns in the recorded data and identify increasing or decreasing values.

Due to a signed ethical declaration, the device could not be tested on a human subject. Therefore, the device data cannot be discussed as the data showed will be raw data from the sensors.

4.3 Security Features Used

The website has the following security features:

- Passwords are hidden when typed in the password field, so the characters are not visible.

- Passwords have been hashed with SHA-2, within the database and once entered in the login web page, to keep passwords hidden and secure.

- HTML special chars are used to prevent the web pages from being exploited with code injections.

- A secure image capture was added to reduce brute force attempts.

The medical wearable device is kept secure because:

- The Bluetooth connection is AES encrypted and requires a PIN.

- The SD card must be matched to the device before the device starts recording data.

4.4 Ethical Issues with the System

There were no human subjects used to test the medical device created in this dissertation, as an ethical form was signed.

However, the ethical issues surrounding the system created in this dissertation are:

- The collected data could be used against the patient by health insurance companies or wellness programs run by employers.

- The collected data could be shared with the doctor’s clinic without the patient knowing. The collected data could be misused to harm the patient.

- This device could be misused to keep track of people. For example, a business company or insurance company could force employees or customers to wear this device to keep track of them, instead of using the device for its purpose of wellness improvement and health awareness.

5.0 Discussion

In this chapter, this research project will be analysed and reviewed.

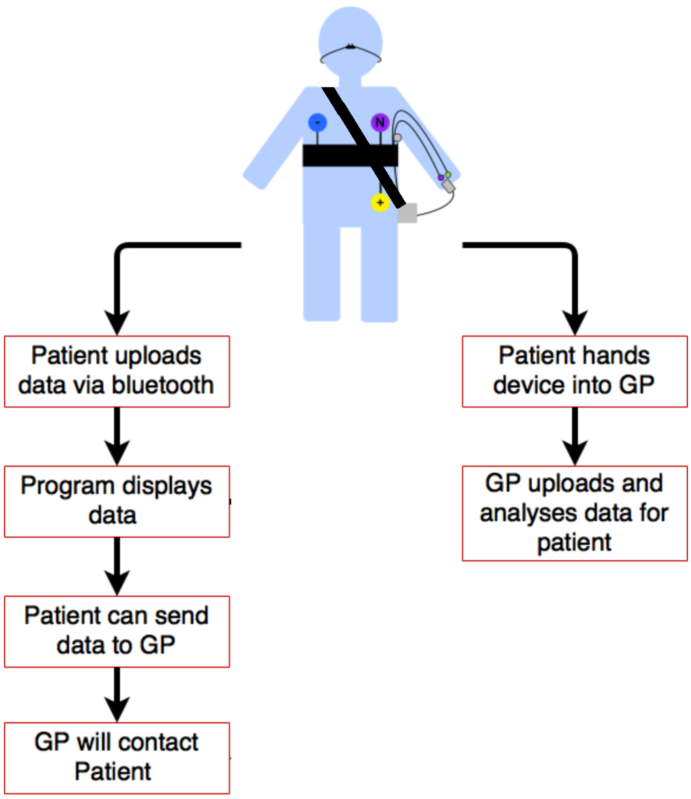


Figure 58 flow diagram for the eHealth system created

Figure 38 shows, in step form, the overview of how the patient and health care practitioner are involved with the web application created. For the first step, the patient wears the device for a period. Next, the patient uploads the data collected from the monitor to the program via the SD card, USB cable, Bluetooth, or they can provide the SD card to a Health Care Professional where they will upload the data to the patient's profile. If the USB cable or Bluetooth is used as a method of data transfer, the processing application is used. The user opens the application, enters the port number for the device and clicks the play button. The user then enters a file name and presses enter. The user then clicks the buttons in order and the file transfer is complete. The copied file is found in the same folder as the Processing application.

Once uploaded, the programme displays selected data logs in visual graphs. The Health Care Professional can view the data, of patients in the surgery. The data recorded is kept secure when data transfers occur using Bluetooth, due to the AES encryption.

This system allows the patient to be more aware of their own health and allows them to take more independence over their health. The device and web application are simple to use.

5.1 Secure Medical Wearable Device

Figure 39 shows the secure medical monitor created. The device weighs 470grams, so the wearable device is lightweight and easy to carry. Therefore, people who have physical disabilities will be able to wear the device safely. It does not have to be connected to a computer to function, so, allowing people to continue with their day. However, the battery currently lasts up to 7 minutes, so would require multiple charges throughout the day.

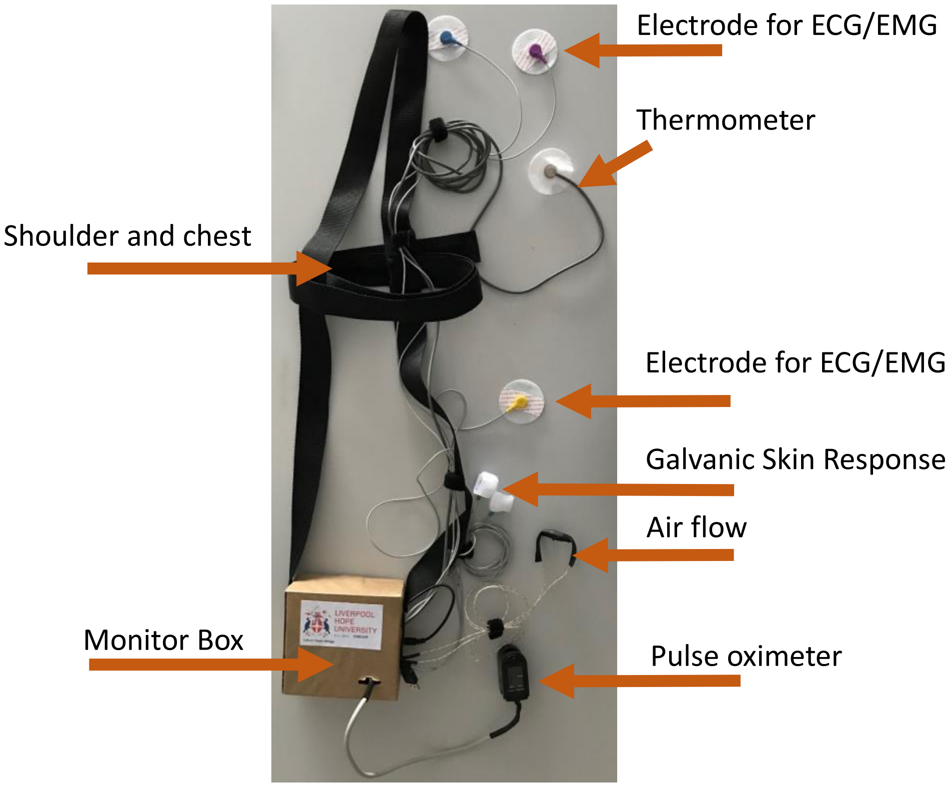


Figure 59 the monitor with all the sensors and shields connected

In comparison to AMON, this monitor can record more body parameters such as airflow and body temperature, which are important for diagnosis. AMON records skin temperature at the wrist, which is not accurate as the environment can affect this result. This medical device records body temperature at the auxiliary point of the human body, as this area is more accurate and is least affected by the environment.

This monitor does not require any service connection to function. The SD card can multiple days of recorded data, as one hour of recording is 2KB.

The data recorded by the monitor is uploaded to the database, by uploading data from the device to a computer via Bluetooth or SD card or USB cable. From here the data is uploaded to the database using the web application. Therefore, not requiring a cellular connection such as 3G to upload data.

However, AMON is more user-friendly as the data is automatically uploaded and is worn on the wrist, so is less obtrusive.

In comparison to MyHeart, this monitor is similar as it does not require all sensors to function, as more or fewer sensors can be added. However, MyHeart can be less obtrusive as it can be incorporated into clothing. It does not require Wi-Fi as it uses a cellular connection, which some people may prefer if they do not have Wi-Fi installed.

5.2 Web Application

The Web application created works successfully. Patients can upload their data to the database and view the data in graphical form. This data can also be viewed by health care professionals for advice on health care.

This website helps provide insight into a person's health, as continuous data is recorded and displayed. The webpage is clear and simple to use. The use of the Patient website links, on the graphical data webpage, allows the user to look up symptoms or research for medical advice.

The webpage is secure and private as it keeps passwords hidden, displayed data cannot be used to identify patients, within the health care professional's web pages.

In comparison to babylon, the webpage does not allow communication with health care professionals and does not provide a reliable symptom checker. These two features are useful and would be a future improvement to this website.

Currently, babylon does not allow health data uploads, however, in the future it allows artificial intelligence to provide health tips from the data uploaded and would be able to accept uploads from mobile devices.

In comparison to Patient Access, this website does not allow for appointment creations or prescription insights. This is because the main aim of this website is to improve the patient health through graphical displays of health data and personal health awareness and independence.

The Processing application achieves its purpose of data transfer however, this method makes the data transfer process more difficult and complicated.

5.3 Future Improvements

In this subchapter, the improvements to this research project will be discussed.

5.3.1 Secure Medical Wearable Device

To make the wearable monitor more unobtrusive the size and weight must be reduced. Due to its large size, the monitor is visible outside of clothing which can be an issue for some people who may be image conscious. This improvement can be done by creating sensor modules that are small and compact, this will reduce the number and length of cables used. A hub such as a smartwatch or mobile device would be used to connect to these modules via a wireless medium such as Bluetooth, so there is no need for a box to be worn. This will make the system more unobtrusive and wearable.

The monitor created does not have a screen, so if the monitor stops working the user may not know and cannot see their health data in real time. Therefore, by adding a hub such as a smartwatch or mobile device, the user can be notified of any issues and view their health data in real-time.

Encryption could not be added to the device due to lack of on-board memory on the Arduino UNO. This could be improved by using a customised circuit board. The Bluetooth module is difficult to customise due to unknown bugs, therefore a custom-built Bluetooth module with encryption and Bluetooth 4.1 would be more efficient.

The battery used will only last up to 7 minutes, therefore the user will have to charge the device multiple times throughout the day. A more compact and larger capacity battery would allow the user to wear the device for longer and reduce the number of recharges required. More energy efficient boards with lower power consumption will improve the energy consumption of the device.

The glucometer was not added to the device due to lack of time and bugs with the code. Adding the glucometer to the device would allow the user to upload the stored glucometer data, allowing the health care professional easier access to the data. However, the glucometer can still be used as a separate device to view blood glucose readings.

5.3.2 Web Application

Currently, the web application allows a user to log in, view their logs, upload data to the database securely and view this data in graphical form. The web application allows health care professionals to view this data for feedback on the patient's health.

To improve this system, artificial intelligence could be used to provide the user with health tips and advice on health care. This will increase health independence and could reduce the number of appointments used or wasted.

A social interaction will help keep people more proactive and motivated to improve their wellness. Such as, sharing data on achievements made in exercises or health habits.

The ability to view real-time data from the monitor, so that the user can see what could be causing an issue they are experiencing. A more direct connection to the health monitor, to make data uploads more automatic and simplistic.

Currently, the web application is unable to connect directly to the device and store the file. Being able to connect to the device using a single application will make this system more user-friendly and simpler to use.

A more efficient graphical library would be used, due to long load times caused by the current library.

The use of encryption on the website and stronger security will help keep patient data secure and private. Gaining an HTTPS certificate will provide a secure encrypted connection to the site. Using a much more improved hashing method such as bcrypt will prevent passwords from being seen.

5.3.3 Overall

Overall, the system created works successfully and solves the issues found. However, these improvements will allow the device to become more successful in its use as it will become more easily integrated into a person's lifestyle.

Trials on using the device will help improve the system as human feedback from the public and health care professionals, will help improve the device much quicker and more efficiently.

Although, the system could not be tested or used on human subjects, the system shows how eHealth can be used to improve medical care. As the system allows the user to record continuous data, showing a clearer insight into their health.

6.0 Conclusion

To conclude, a solution has been made successfully using a health monitor and web application, to solve the issue. The secure medical wearable device allows people to record their body parameters securely, allow them to view their data in a graphical form and allow health care professionals to view this data for feedback.

This monitor will help diagnose illnesses early and reduce diagnosis times, due to data being recorded throughout the day. The number of appointments available may increase because the patient can use the device outside of the clinic and upload the data from their own home. The wearable device provides an insight into the user's own health, therefore allowing them to be more knowledgeable of their current health and become more proactive with independent health care.

However, this solution can be further improved so it can become more unobtrusive and provide more information about the human health.

Appendices

References

1. Alnosayan, N., Lee, E., Alluhaidan, A., Chatterjee, S., Houston-Feenstra, L., Kagoda, M. and Dysinger, W. (2014). MyHeart: An intelligent mHealth home monitoring system supporting heart failure self-care. *2014 IEEE 16th International Conference on e-Health Networking, Applications and Services (Healthcom)*. [online] Available at: http://ieeexplore.ieee.org/document/7001860/ [Accessed 11 Jan. 2018].

2. Ammenwerth, E. and Shaw, N. (2005). Bad health informatics can kill - Is evaluation. Methods of Information in Medicine 44, [online] pp.1-3. Available at: https://www.researchgate.net/publication/7955842_Bad_health_informatics_can_kill_-_Is_evaluation_the_answer [Accessed 19 Mar. 2018].

3. Anliker, U., Ward, J., Lukowicz, P., Troster, G., Dolveck, F., Baer, M., Keita, F., Schenker, E., Catarsi, F., Coluccini, L., Belardinelli, A., Shklarski, D., Alon, M., Hirt, E., Schmid, R. and Vuskovic, M. (2004). AMON: A Wearable Multiparameter Medical Monitoring and Alert System. *IEEE Transactions on Information Technology in Biomedicine*, [online] 8(4), pp.415-427. Available at: http://ieeexplore.ieee.org/document/1362650/ [Accessed 11 Jan. 2018].

4. Ates, G. and Polat, K. (2012). Measuring of oxygen saturation using pulse oximeter based on fuzzy logic. 2012 IEEE International Symposium on Medical Measurements and Applications Proceedings.

5. babylon health. (2017). *Online Doctor Consultations & Advice | babylon health*. [online] Available at: https://www.babylonhealth.com/ [Accessed 19 Mar. 2018].

6. Bourdon, R. (2018). WampServer. Alter Way.

7. Catwell, L. and Sheikh, A. (2009). Evaluating eHealth Interventions: The Need for Continuous Systemic Evaluation. *PLoS Medicine*, 6(8), p.e1000126.

8. Cooking-hacks.com. (2018). e-Health Sensor Platform V2.0 for Arduino and Raspberry Pi [Biometric / Medical Applications]. [online] Available at: https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical [Accessed 22 Mar. 2018].

9. de Grood, C., Raissi, A., Kwon, Y. and Santana, M. (2016). Adoption of e-health technology by physicians: a scoping review. *Journal of Multidisciplinary Healthcare*, Volume 9, pp.335-344.

10. Emishealth. (2016). Patient Access. [online] Available at: https://www.emishealth.com/products/patient-access/?tab=empowering-patients [Accessed 19 Mar. 2018].

11. Eysenbach, G. (2001). What is e-health?. *Journal of Medical Internet Research*, [online] 3(2), p.e20. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1761894/ [Accessed 6 Dec. 2017].

12. Gacek, A. and Pedrycz, W. (2012). *ECG Signal Processing, Classification and Interpretation*. London: Springer-Verlag London Limited. Pages 120-121

13. Government.nl. (2018). *Benefits of eHealth*. [online] Available at: https://www.government.nl/topics/ehealth/benefits-of-ehealth [Accessed 1 Mar. 2018].

14. Harris, M. and Habetha, J. (2007). The MyHeart project: A framework for personal health care applications. *2007 Computers in Cardiology*. [online] Available at: http://ieeexplore.ieee.org/abstract/document/4745440/ [Accessed 11 Jan. 2018].

15. Joshua Mervine, X. (2018). Quick Start. [online] Bootstrapcdn.com. Available at: https://www.bootstrapcdn.com/ [Accessed 10 Feb. 2018].

16. Krijgsman, J., Peeters, J., Burghouts, A., Brabers, A., de Jong, J., Mollx, T., Friele, R. and van Gennip, L. (2015). *Between the spark and the flame 2015 eHealth monitor*. [online] Nictiz.nl. Available at: https://www.nictiz.nl/SiteCollectionDocuments/Rapporten/eHealth-monitor%20ENGELS%202015.pdf [Accessed 1 Mar. 2018].

17. Liang, X., Barua, M., Chen, L., Lu, R., Shen, X., Li, X. and Luo, H. (2012). Enabling pervasive healthcare through continuous remote health monitoring. IEEE Wireless Communications, 19(6), pp.10-18.

18. Lister, C., West, J., Cannon, B., Sax, T. and Brodegard, D. (2014). Just a Fad? Gamification in Health and Fitness Apps. *JMIR Serious Games*, 2(2), p.e9.

19. Li, M., Lou, W. and Ren, K. (2010). Data security and privacy in wireless body area networks. *IEEE Wireless Communications*, [online] 17(1), pp.51-58. Available at: http://ieeexplore.ieee.org/document/5416350/ [Accessed 11 Jan. 2018].

20. Lukowicz, P., Anliker, U., Ward, J., Troster, G., Hirt, E. and Neufelt, C. (2002). AMON: a wearable medical computer for high risk patients. *Proceedings. Sixth International Symposium on Wearable Computers*, pp.133-134. doi: 10.1109/ISWC.2002.1167230

21. Meingast, M., Roosta, T. and Sastry, S. (2006). Security and Privacy Issues with Health Care Information Technology. *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. [online] Available at: http://ieeexplore.ieee.org/document/4463039/ [Accessed 11 Jan. 2018].

22. Omoogun, M., Seeam, P., Ramsurrun, V., Bellekens, X. and Seeam, A. (2017). When eHealth meets the internet of things: Pervasive security and

privacy challenges. *2017 International Conference on Cyber Security And Protection Of Digital Services (Cyber Security)*. [online] Available at: http://ieeexplore.ieee.org/document/8074857/ [Accessed 11 Jan. 2018].

23. Phillips, D. (2005). *Securimage*. phpcaptcha. [online] Available at: https://www.phpcaptcha.org
24. Powell, T. (2009). *ZingChart*. San Diego: ZingChart. [online] Available at: https://www.zingchart.com
25. Reference. (n.d.) In: OXFORD DICTIONARY OF ENGLISH [online] Oxford: Oxford University Press, Inc. Available at: https://www.collinsdictionary.com/dictionary/english/wellness [Accessed 19 Mar. 2018].
26. Virone, G., Wood, A., Selavo, L., Cao, Q., Fang, L., Doan, T., He, Z., Stoleru, R., Lin, S. and A. Stankovic, J. (2006). An Advanced Wireless Sensor Network for Health Monitoring. Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2).. [online] Available at: https://www.researchgate.net/publication/242382910_An_Advanced_Wireless_Sensor_Network_for_Health_Monitoring [Accessed 19 Mar. 2018].
27. World Health Organization. (2017). *Global Diffusion of Ehealth*. Geneva: World Health Organization.
28. World Health Organization. (2017). *The top 10 causes of death*. [online] Available at: [Accessed 6 Dec. 2017].

Code:

1.0 Arduino Code

```
#include <PinChangeInt.h>
#include <eHealth.h>
#include <Wire.h>
#include <SPI.h>
#include <SD.h>
#include "RTClib.h"

#define ECHO_TO_SERIAL   1 // echo data to serial port
#define WAIT_TO_START    0 // Wait for serial input in setup()

RTC_DS1307 RTC; // define the Real Time Clock object
const int chipSelect = 10;// for the data logging shield, we use digital pin 10 for the SD cs line
File logfile;// the logging file
String output, patient;
int i = 0, logData = 1;
char filename[] = "DATA01.csv";

void setup()
{
eHealth.initPulsioximeter();
PCintPort::attachInterrupt(6, readPulsioximeter, RISING);

Serial.begin(115200);
pinMode(10, OUTPUT);

// see if the card is present and can be initialized:
if (!SD.begin(chipSelect))
{
    error("Card failed, or not present");
    return;
}

if (SD.exists(filename))
{
    char patient[] = "1001|1957/04/11";
    logfile = SD.open(filename);
    String line = logfile.readStringUntil('\n');
    char match[16];
    line.toCharArray(match,16);
    for(int i=0; i<line.length(); i++)
    {
        if (!match[i] == patient[i])
        {
        error("Patient Data Match Failed");
```

```
}
}
Serial.println("Patient Data Matched");
}
else
{
error("couldnt open file");
}


Wire.begin();
if (!RTC.begin())
{
error("RTC failed");
}

if (! RTC.begin()) {
Serial.println("Couldn't find RTC");
while (1);
}

if (! RTC.isrunning()) {
Serial.println("RTC is NOT running!");
RTC.adjust(DateTime(F(__DATE__), F(__TIME__)));
}

logfile.close();
}

void loop()
{
DateTime now = RTC.now();
if (logData == 1 && now.second()==00)
{
logfile=SD.open(filename,FILE_WRITE);
logfile.print(now.year(), DEC);
logfile.print("-");
if(now.month() < 10)
{
logfile.print("0" + String(now.month()));
}
else
{
logfile.print(now.month(), DEC);
}
logfile.print("-");
if(now.day() < 10)
{
logfile.print("0" + String(now.day()));
}
else
{
logfile.print(now.day(), DEC);
```

```
    }
    logfile.print(",");
    if(now.hour() < 10)
    {
        logfile.print("0" + String(now.hour()));
    }
    else
    {
        logfile.print(now.hour(), DEC);
    }
    if(now.minute() < 10)
    {
        logfile.print("0" + String(now.minute()));
    }
    else
    {
        logfile.print(now.minute(), DEC);
    }
    delay(1000); //One data per second
    output += ',';
    output += String(eHealth.getTemperature());
    delay(5);
    output += ',';
    output += String(eHealth.getEMG());
    delay(5);
    output += ',';
    output += String(eHealth.getECG());
    delay(5);
    output += ',';
    output += String(eHealth.getAirFlow());
    delay(5);
    output += ',';
    output += String(eHealth.getSkinConductance());
    delay(5);
    output += ',';
    output += String(eHealth.getSkinResistance());
    delay(5);
    output += ',';
    output += String(eHealth.getSkinConductanceVoltage());
    delay(5);
    output += ',';
    output += String(eHealth.getBPM());
    delay(5);
    output += ',';
    output += String(eHealth.getOxygenSaturation() + 90);
    delay(5);

    logfile.println(output);
    Serial.println(output);
    logfile.close();
    Serial.flush();
```

```
output ="";
}
}

void readPulsioximeter()
{
i ++;
if (i == 50)
{
eHealth.readPulsioximeter();
i = 0;
}
}

void error(char *str)
{
Serial.print('error: ' + str);
while(1);
}

void serialEvent()
{
while (Serial.available())
{
if(Serial.read() == '1')
{
delay(2000);
logData = 0;
logfile.close();
logfile= SD.open(filename);
patient = logfile.readStringUntil('\n');
while (logfile.available())
{
    String line = logfile.readStringUntil('\n');
    Serial.println(line);
}
logfile.close();
break;
}

if(Serial.read() == '2') // 22 is required for this if to become true
{
delay(2000);
SD.remove("DATA01.csv");
logfile = SD.open("DATA01.csv", FILE_WRITE);
logfile.println(patient);
Serial.println(patient);
logfile.close();
patient = "";
logData = 1;
break;
}
```

```
    }
}
```

2.0 Processing Code

```
import processing.serial.*; //import the Serial library
import javax.swing.JOptionPane;
PrintWriter output; // creating variable to write data to a file
Serial ehealthPort; // assisgning the name my port to serial port
String val, file_name, serial_input;
PImage img;
boolean firstContact = false;

int rectX, rectY, rectX_2, rectY_2, rectX_3, rectY_3;
int rectSize = 90;     // Diameter of rect
color rectColor, rectColor_2, rectColor_3, rectHighlight;
boolean rectOver = false, rectOver_2= false, rectOver_3= false;

void setup() {
  size(850, 500);
  img = loadImage("hope_logo.jpg");
  img.resize(850, 500);
  background(img);

  file_name= JOptionPane.showInputDialog("Please type name for the file to be stored in");
  file_name = file_name + ".csv"; // adds file type to name entered
  output = createWriter( file_name ); // writes to text file

  ehealthPort = new Serial(this, Serial.list()[0], 115200);
  ehealthPort.bufferUntil('\n');

  rectColor = color(0, 200, 50);
  rectColor_2 = color(50, 0, 200);
  rectColor_3 = color(200, 0, 50);
  rectHighlight = color(150, 0, 150);
  rectX = 50;
  rectY = 70;
  rectX_2 = 50;
  rectY_2 = 200;
  rectX_3 = 50;
  rectY_3 = 330;

  textSize(20);
  text("1)   Click this button to receive data", 190, 120);
  text("2)   Click this button to close the file", 190, 250);
```

```
  text("3)    Click this button to restart device and delete file on SD card", 190, 380);
  fill(0, 102, 153);
}


void draw()
{

  ///BUTTONS
  update(mouseX, mouseY);

  if (rectOver)
  {
    fill(rectHighlight);
  } else
  {
    fill(rectColor);
  }
  stroke(255);
  rect(rectX, rectY, rectSize, rectSize);

  if (rectOver_2)
  {
    fill(rectHighlight);
  } else
  {
    fill(rectColor_2);
  }
  stroke(255);
  rect(rectX_2, rectY_2, rectSize, rectSize);

  if (rectOver_3)
  {
    fill(rectHighlight);
  } else
  {
    fill(rectColor_3);
  }
  stroke(255);
  rect(rectX_3, rectY_3, rectSize, rectSize);
}


void serialEvent( Serial ehealthPort)
{
  val = ehealthPort.readStringUntil('\n');
  if (val != null)
  {
    val = trim(val);
    println(val);
    output.println(val);
  }
```

```
  }

  void update(int x, int y)
  {
    if ( overRect(rectX, rectY, rectSize) )
    {
      rectOver = true;
      rectOver_2 = false;
      rectOver_3 = false;
    }
    else if ( overRect_2(rectX_2, rectY_2, rectSize) )
    {
      rectOver = false;
      rectOver_3 = false;
      rectOver_2 = true;
    }
    else if ( overRect_3(rectX_3, rectY_3, rectSize) )
    {
      rectOver = false;
      rectOver_2 = false;
      rectOver_3 = true;
    }
    else
    {
      rectOver_3 = rectOver_2 = rectOver = false;
    }
  }


  // code which saves stored data in text file when a keyboard key is pressed
  void keyPressed() {
  }

  void mousePressed()
  {
    if (rectOver)
    {
      ehealthPort.write('1');
      println("Sending data, Restart device to keep file");
    }

    if (rectOver_2)
    {
      output.flush();  // Writes the remaining data to the file
      output.close();  // Finishes the file
      println("Closing File: " + file_name);
    }

```

```
    if (rectOver_3)
    {
      ehealthPort.write("22");
      println("Removing file and restarting");
    }
  }

  boolean overRect(int x, int y, int height)
  {
    if (mouseX >= x && mouseX <= x+width && mouseY >= y && mouseY <= y+height)
    {
      return true;
    } else
    {
      return false;
    }
  }

  boolean overRect_2(int x, int y, int height)
  {
    if (mouseX >= x && mouseX <= x+width && mouseY >= y && mouseY <= y+height)
    {
      return true;
    } else
    {
      return false;
    }
  }

  boolean overRect_3(int x, int y, int height)
  {
    if (mouseX >= x && mouseX <= x+width && mouseY >= y && mouseY <= y+height)
    {
      return true;
    } else
    {
      return false;
    }
  }
```

3.0 Web page Code

login_form.php

```
<?php

session_start();

if (isset($_POST['username']))
{
	doAuthenticate();
}


	function doAuthenticate()
	{
	// login and use database
	  $conn = mysqli_connect("localhost", "root", "Root#123", "health_database");

	// error check for dtabase connection
	if (!$conn)
	{
		die("Connection Failed: " . mysqli_connect_error());
	}

	include_once $_SERVER['DOCUMENT_ROOT'] . '/securimage/securimage.php';
	$securimage = new Securimage();

	$username = trim($_POST['username']);
	$password = trim($_POST['password']);
	$password = hash('sha512', $password);


	$sql = "SELECT username, password FROM authentication WHERE username = '$username' AND password= '$password' ";

	$result = mysqli_query($conn,$sql) or die(mysqli_error($conn));
	$numOfRows = mysqli_num_rows($result);



	if ($numOfRows == 1 && $securimage->check($_POST['captcha_code']) == true)
	{

			$line = mysqli_fetch_array($result);
```

```
                $_SESSION['username'] = $line['username'];

                $find_user = $line['username'];
                $username_length = strlen($find_user);

                $char = str_split($find_user, 1);
                if ($char[4] == "h")
                {
                              header("location: hcp_welcome.php");
                }
                elseif ($char[4] == "p")
               {
                              header("location: patient_welcome.php");
               }
     }

     if ($numOfRows != 1)
     {
               echo ("<h2>Login failed - authentication error</h2>");
     }

     if ($securimage->check($_POST['captcha_code']) == false)
     {
     echo ("<h3>The security code entered was INCORRECT.<br></h3>");
     echo ("<h3>Please try again.</h3>");
     }

     }

     ?>




     <!DOCTYPE html>
     <html lang="en">
     <head>
     <meta charset="UTF-8">
     <title>Login</title>
     <link rel="stylesheet" type="text/css" href="login_form_CSS.css">
     <style type="text/css">
     body{ font: 17px sans-serif; text-align: center; }
     .wrapper
     {
                margin-left: auto;
                margin-right: auto;
```

```
            width: 380px;
    }
    </style>
    </head>
    <body>
    <div class="wrapper">
    <img src="images/logo.png" alt="logo" width = "200" height = "135">
    <div class="form-title"><h1> Health Web Page Login</h1></div>
    <p>Please enter your username and password and click submit.</p>

    <form class="form-container" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post" autocomplete = "off">

    <div class="form-title">
    <label>Username:</label>
    <input type="text" name="username" class="form-field" value= "" required>
    </div>

    <div class="form-title">
    <label>Password (Case sensitive):</label>
    <input type="password" name="password" class="form-field" required>
    </div>

    <label>Please enter the characters seen in the image</label>
    <br>
    <img id="captcha" src="securimage/securimage_show.php" alt="CAPTCHA Image" />
    <input class="form-field" type="text" name="captcha_code" size="10" maxlength="6" required/>
    <a href="#" onclick="document.getElementById('captcha').src = 'securimage/securimage_show.php?' + Math.random(); return false">[ Different Image ]</a>
    <br>

    <div class="submit-container">
                <a href="http://phpcaptcha.org/" target="_blank"><img src="images/securimage.png" width = "100" height = "30" align ="left"></a>
                <input class="submit-button" type="submit" value="Submit" />
            </div>
    </form>

    </div>
```

```
</body>
</html>
```

patient_welcome.php

```
<?php

session_start();

if(!isset($_SESSION['username']) || empty($_SESSION['username']))
{
header("location: login_form.php");
exit;
}

	$conn = mysqli_connect("localhost", "patient", "patientHOPEuniversity#2018", "health_database");

	if (!$conn)
	{
	die("Connection Failed: " . mysqli_connect_error());
	}

	// get an array of logs to display as a tbale on welcome page

	$user = $_SESSION['username'];

	// getting patient name to display on welcome page


	$get_patientUID = "SELECT uID FROM patient_authentication WHERE username = '$user'";

	$patientUID = mysqli_query($conn, $get_patientUID) or die(mysqli_error());
	$line = mysqli_fetch_array($patientUID);

	$get_patientLog = "SELECT * FROM patient_log WHERE uID = $line[uID]";
	$patient_logs = mysqli_query($conn, $get_patientLog) or die(mysqli_error());

	$get_patientName= "SELECT forename FROM patient WHERE uID = $line[uID]";
	$patientNames = mysqli_query($conn, $get_patientName) or die(mysqli_error());
	$patientName = mysqli_fetch_array($patientNames);

	mysqli_close($conn);
	?>
```

```
<!DOCTYPE html>
<html lang="en">
<head>

<meta charset="UTF-8">

<title>Welcome</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">
<link rel="stylesheet" type="text/css" href="welcome_form_CSS.css">


<style type="text/css">
body{ font: 14px sans-serif; text-align: center; }

#input
{
width: 50%;
font-size: 16px;
padding: 15px 10px 15px 10px;
border: 2px solid #ddd;
margin-bottom: 12px;
position: relative;
}
#logs
{
border-collapse: collapse;
width: 50%;
}

#logs th, #logs td
{
padding: 15px;
border: 1px solid #ddd;
text-align: center;
}
#logs th
{
background-color:#20a354;
color: black;
}

#logs tr:nth-child(even) {background-color: #e1e1e1;}
#logs tr:hover {background-color:#f1f1f1;}

div.log_table
{
position: relative;
left: 60px;
```

```
bottom: -50px;
}
</style>
</head>
<body>
<div class="page-header">
<img src="images /logo.png" alt="logo" width = "200" height = "135">
<h1>Welcome <b><?php echo $patientName['forename']; ?></b></h1>
<br>
<p>
<a href="logout.php" class="button">Log Out of Your Account</a>
</p>
</div>
<div>
<a href="surgery.php" class="button">Surgery Contact Details</a>
<a href="upload.php" class="button">Upload a data log</a>
</div>
<div class = log_table style = "overflow-y: auto;">
<form action = "graphical_data.php" method = "post">
<input type="text" id="input" onkeyup="searchLogTable()" placeholder="Search for log by (DD), (MM) or (YYYY)" title="Type in a date">
<table id = "logs">
<th> Select </th><th> Log </th> <th> Date </th>
<?php
while ($lines = mysqli_fetch_array($patient_logs))
{
echo ("<tr><td><input type='radio' name='logID' value=".$lines['logID']." required></td>");
echo ("<td>" . $lines['logID']. "</td> ");
echo ("<td>" . $lines['dateValue']. "</td></tr>");
}
?>
</table>
<br>
<input type = "submit" class = "button" value = "Click To View Chosen Log" style= "float:left">
</br>
</form>
```

```
</div>
<script>
function searchLogTable()
{
var input, filter, table, tr, td, i;
input = document.getElementById("input");
filter = input.value.toUpperCase();
table = document.getElementById("logs");
tr = table.getElementsByTagName("tr");

for (i = 0; i < tr.length; i++)
{
td = tr[i].getElementsByTagName("td")[2];
if (td)
{
if (td.innerHTML.toUpperCase().indexOf(filter) > -1)
{
tr[i].style.display = "";
} else
{
tr[i].style.display = "none";
}
}
}
}
</script>
</body>
</html>
```

surgery.php

```
<?php

session_start();

if(!isset($_SESSION['username']) || empty($_SESSION['username']))
{
header("location: patient_welcome.php");
exit;
}

$conn = mysqli_connect("localhost", "root", "Root#123", "health_database");

if (!$conn)
{
die("Connection Failed: " . mysqli_connect_error());
}

// get an array of logs to display as a tbale on welcome page
$get_surgeryDetails = "SELECT * FROM surgery";
$surgeryDetails = mysqli_query($conn, $get_surgeryDetails) or die(mysqli_error());
?>



<!DOCTYPE html>
<html lang="en">
<head>

<meta charset="UTF-8">

<title>Welcome</title>

<link rel="stylesheet" type="text/css" href="welcome_form_CSS.css">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">


<style type="text/css">
body{ font: 14px sans-serif; text-align: center; }

#surgery
{
border-collapse: collapse;
width: 50%;
```

```
}

#surgery th, #surgery td
{
padding: 15px;
border: 1px solid #ddd;
text-align: center;
}
#surgery th
{
background-color:#20a354;
color: black;
}

#surgery tr:nth-child(even) {background-color: #e1e1e1;}

div.surgery_table
{
position: relative;
left: 60px;
bottom: -50px;
}

</style>

</head>
<body>

<div class="page-header">
<img src="images/logo.png" alt="logo" width = "200" height = "135">
<h1><b></b> Surgery Details </h1>
<p><a href="patient_welcome.php" class="button">Go Back</a></p>
</div>

<div class = surgery_table>
<table id = "surgery">
<th> Surgery </th> <th> ContactNumber </th> <th> Address </th>
<?php
while ($lines = mysqli_fetch_array($surgeryDetails))
{
echo ("<tr><td>" . $lines['name'] . "</td>");
echo ("<td>" . $lines['contactNumber']. "</td>");
echo ("<td>" . $lines['addressL1'] ." ".$lines['addressL2'] ."   ". $lines['postcode'] ." ". "</td></tr>");
}
?>
</table>
```

```
    </div>
    </body>
</html>
```

logout.php

```
<?php

session_start();

$_SESSION = array();

session_destroy();

header("location: login_form.php");

    exit;

    ?>
```

upload.php

```php
<?php

session_start();

if(!isset($_SESSION['username']) || empty($_SESSION['username']))
{
header("location: login_form.php");
exit;
}

	$conn = mysqli_connect("localhost", "root", "Root#123", "health_database");

	if (!$conn)
	{
	die("Connection Failed: " . mysqli_connect_error());
	}


	// Check if image file is a actual image or fake image
	if(isset($_POST["submit"]))
	{

	$target_directory = "uploads/";
	$target_file = $target_directory . basename($_FILES['csv']['name']);
	$uploadOk = 1;
	$datalog_fileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));

	// Check if file already exists
	if (file_exists($target_file))
	{
	echo "Sorry, file already exists.";
	$uploadOk = 0;
	}

	// Allow certain file formats
	if($datalog_fileType != "txt" && $datalog_fileType != "csv")
	{
	echo "Sorry, only txt and csv files are allowed.";
	$uploadOk = 0;
	}

	// Check if $uploadOk is set to 0 by an error
	if ($uploadOk == 0)
	{
```

```
echo "Sorry, your file was not uploaded.";
// if everything is ok, try to upload file
}

else
{
if (move_uploaded_file($_FILES["csv"]["tmp_name"], $target_file))
{
echo ("<h3>The file ". basename( $_FILES["csv"]["name"]). " has been uploaded.</h3>");
}
else
{
echo "Sorry, there was an error uploading your file.";
}
}

$handle = fopen($target_file,"r");
$line = fgetcsv($handle,",");
// required to prevent first line of data log file from being uploaded as this data is only for use by the health monitor itself.
$date = "";
for ($i = 0; $i<1; $i++)
{
$line = fgetcsv($handle,",");
if($line [$i] == "")
{}
else
{
$date = $line [$i];
}
}
$user = $_SESSION['username'];

$get_patientUID= "SELECT uID FROM patient_authentication WHERE username = '$user'";
$get_uID = mysqli_query($conn, $get_patientUID)or die(mysqli_error());
$uID = mysqli_fetch_array($get_uID);
$patientUID = $uID [0];

mysqli_query($conn, "INSERT INTO patient_log (dateValue, uID) VALUES ('$date', '$patientUID')") or die(mysqli_error());


$get_datalogID = "SELECT logID FROM patient_log WHERE uID = '$patientUID' AND dateValue ='$date'";
$get_logID = mysqli_query($conn, $get_datalogID)or die(mysqli_error());
```

```
$logID = mysqli_fetch_array($get_logID);
$patient_logID = $logID [0];

mysqli_free_result($get_uID);
mysqli_free_result($get_logID);

do {
$timeValue = $line [1];
$temperature = $line [2];
$emg = $line [3];
$ecg = $line [4];
$airFlow = $line [5];
$skinConductance = $line [6];
$skinResistance = $line [7];
$skinConductanceVoltage = $line [8];
$beatsPerMinute = $line [9];
$bloodOxygen = $line [10];
$query = "INSERT INTO monitor_data (logID, timeValue, temperature, emg, ecg, airFlow, skinConductance, skinResistance, skinConductanceVoltage, beatsPerMinute, bloodOxygen)
VALUES
('$patient_logID','$timeValue','$temperature','$emg','$ecg','$airFlow','$skinConductance', '$skinResistance', '$skinConductanceVoltage', '$beatsPerMinute', '$bloodOxygen')";
mysqli_query($conn, $query) or die(mysqli_error());

} while ($line = fgetcsv($handle,","));
fclose($handle);
unlink($target_file);
}
?>



<!DOCTYPE html>
<html lang="en">
<head>

<meta charset="UTF-8">

<title>Welcome</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">
<link rel="stylesheet" type="text/css" href="welcome_form_CSS.css">


<style type="text/css">
```

```
body{ font: 14px sans-serif; text-align: center; }
</style>

</head>
<body>

<div class="page-header">
<img src="images/logo.png" alt="logo" width = "200" height = "135">
<h1><b>Upload Page</b></h1>  <br>
<p>
<a href="patient_welcome.php" class="button">Go Back to Welcome Page</a>
</p>
</div>

<div>
<h3><p>
Select data log file to upload
</p></h3>
</div>
<form action="" method="post" enctype="multipart/form-data">
<input type="file" class="button" name="csv" id="csv" required><br><br>
<input type="submit"  class="button" value="Upload Data Log" name="submit">
</form>
</body>
</html>
```

graphical_data.php

```php
<?php

session_start();

if(!isset($_SESSION['username']) || empty($_SESSION['username']))
{
header("location: login_form.php");
exit;
}

$conn = mysqli_connect("localhost", "root", "", "health_database");

if (!$conn)
	{
	die("Connection Failed: " . mysqli_connect_error());
	}

	if(isset($_POST) && 0 === count($_POST))
	{
	echo ("<h1>check box was not selected</h1><p><h2>please go back and try again</h2></p>");
	}

	$logid = $_POST["logID"];


	$get_monitorData = "SELECT * FROM monitor_data WHERE logID = $logid";// use log selected by user
	$monitor_data = mysqli_query($conn, $get_monitorData) or die(mysqli_error());
	$rows = mysqli_num_rows($monitor_data);

	$temp = array();
	$time = array();
	$emg = array();
	$ecg = array();
	$airFlow = array();
	$skinConductance = array();
	$skinResistance = array();
	$skinConductanceVoltage = array();
	$beatsPerMinute = array();
	$bloodOxygen = array();

	while($array = mysqli_fetch_array($monitor_data))
	{
	$temp[] = $array['temperature'];
	$time[] = $array['timeValue'];
```

```
$emg[] = $array['ecg'];
$ecg[] = $array['emg'];
$airFlow[] = $array['airFlow'];
$skinConductance[] = $array['skinConductance'];
$skinResistance[] = $array['skinResistance'];
$skinConductanceVoltage[] =
$array['skinConductanceVoltage'];
$beatsPerMinute[] = $array['beatsPerMinute'];
$bloodOxygen[] = $array['bloodOxygen'];
}

mysqli_free_result($monitor_data);
mysqli_close($conn);

?>



<!DOCTYPE html>
<html lang="en">
<head>

<meta charset="UTF-8">

<title>Graphical Data</title>

<link rel="stylesheet" href="button_form_CSS.css">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">
<script
src="https://cdn.zingchart.com/zingchart.min.js"></script>

<style type="text/css">
    body{ font: 14px sans-serif; text-align:
    center; }

    #myChart {
    height:100%;
    width:100%;
    min-height:300px;
    }

</style>

<script>


var temp = [];
```

```
var ecg = [];
var emg = [];
var airFlow = [];
var skinConductance = [];
var skinResistance = [];
var skinConductanceVoltage = [];
var beatsPerMinute = [];
var bloodOxygen = [];
var time = [];


<?php
for ($i=0 ; $i<$rows ; $i++)
{
    echo "temp[$i] = $temp[$i];";
    echo "emg[$i] = $emg[$i];";
    echo "ecg[$i] = $ecg[$i];";
    echo "airFlow[$i] = $airFlow[$i];";
    echo "skinConductance[$i] =
    $skinConductance[$i];";
    echo "skinResistance[$i] =
    $skinResistance[$i];";
    echo "skinConductanceVoltage[$i] =
    $skinConductanceVoltage[$i];";
    echo "bloodOxygen[$i] = $bloodOxygen[$i];";
    echo "beatsPerMinute[$i] =
    $beatsPerMinute[$i];";
    echo "time[$i] = $time[$i];";
}
?>

var myChart1 =
{
    "graphset":[
    {
    "background-color":"#ffffff",
    "type":"line",
    "title": {
        "text": ""
    },
    "legend":{
        "toggle-action":"remove",
        "align":"center",
        "adjust-layout":false,
        "borderWidth":0,
        "verticalAlign":"bottom",
        "fontColor":"#7d7d7d",
        "font-size":10,
        "marginRight":20,
        "marginBottom":0,
        "marginTop":0,
        "marker":{
            "type":"square"
```

```
            12.     },
        viii. },
          ix. "preview":{
              1. "background-color":"#F5F7F3",
              2. "border-width":0,
              3. "height":30,
              4. "preserve-zoom":false,
              5. "mask":{
                  a. "backgroundColor":"grey",
                  b. "alpha":0.8
              6. },
              7. "handle":{
                  a. "border-width":2
              8. },
              9. "y":"85%"
           x. },
          xi. "scale-x":{
              1. "zooming":true,
              2. "zoom-to":[0,30],
              3. "item":{
                  a. "font-size":10,
                  b. "font-color":"#7d7d7d"
              4. },
              5. "tick":{
                  a. "visible":true
              6. },
              7. "labels": time,
              8.                          "label":{
              9.                          "text":"Time
                 (HH:MM)",
              10.                                 "font-
                 size":"10px",
              11.                                 "color":"grey"
              12.                          },
         xii. },
        xiii. "scale-y":{
              1. "zooming":false,
              2. "values":"0:50:5",
              3. "guide":{
                  a. "line-style":"dotted"
              4. },
              5. "item":{
                  a. "font-size":10,
                  b. "font-color":"#7d7d7d"
              6. },
              7. "tick":{
                  a. "visible":true
              8. },
              9. "format":"",
              10.     "decimals":1
         xiv. },
          xv. "zoom":{
              1. "active":true,
```

```
                2. "preserve-zoom":true
         xvi. },
        xvii. "plotarea":{
                1. "margin-top":"20",
                2. "margin-left":"20",
                3. "margin":"dynamic 50 105 dynamic"
       xviii. },
         xix. "plot":{
                1. "data-append-selection":true,
                2. "mode":"normal",
                3. "line-width":1,
                4. "line-color":"#FF5733",
                5. "background-color":"#1e88e5",
                6. "marker":{
                     a. "size":3,
                     b. "background-color":"#FF5733",
                     c. "border-width":2,
                     d. "border-color":"#FF5733"
                7. },
                8. "tooltip":{
                     a. "visible":true,
                     b. "text":"%kv<br>%vv"
                9. },
                10.    "selection-mode":"multiple",
                11.    "selected-state":{
                     a. "background-color":"#ffa726",
                     b. "border-width":0
                12.    }
          xx. },
         xxi. "series":[
                1. {
                     a. "values":temp,
                     b. "text":"Temperature (°C)"
                2. }
        xxii. ]
       xxiii. }
       b. ]
75.      };

76.      var myChart2 = {
           i. "graphset":[
          ii. {
         iii. "background-color":"#ffffff",
          iv. "type":"line",
           v. "title": {
                1. "text": ""
          vi. },
         vii. "legend":{
                1. "toggle-action":"remove",
                2. "align":"center",
                3. "adjust-layout":false,
                4. "borderWidth":0,
                5. "verticalAlign":"bottom",
```

```
            6. "fontColor":"#7d7d7d",
            7. "font-size":10,
            8. "marginRight":20,
            9. "marginBottom":0,
            10.     "marginTop":0,
            11.     "marker":{
                 a. "type":"square"
            12.     },
    viii. },
      ix. "preview":{
            1. "background-color":"#F5F7F3",
            2. "border-width":0,
            3. "height":30,
            4. "preserve-zoom":false,
            5. "mask":{
                 a. "backgroundColor":"grey",
                 b. "alpha":0.8
            6. },
            7. "handle":{
                 a. "border-width":2
            8. },
            9. "y":"85%"
       x. },
      xi. "scale-x":{
            1. "zooming":true,
            2. "zoom-to":[0,30],
            3. "item":{
                 a. "font-size":10,
                 b. "font-color":"#7d7d7d"
            4. },
            5. "tick":{
                 a. "visible":true
            6. },
            7. "labels": time,
            8.                     "label":{
            9.                     "text":"Time
              (HH:MM)",
            10.                         "font-
              size":"10px",
            11.                         "color":"grey"
            12.                      },
     xii. },
    xiii. "scale-y":{
            1. "zooming":false,
            2. "values":"0:1000:10",
            3. "guide":{
                 a. "line-style":"dotted"
            4. },
            5. "item":{
                 a. "font-size":10,
                 b. "font-color":"#7d7d7d"
            6. },
            7. "tick":{
```

```
                a. "visible":true
            8. },
            9. "format":"",
            10.     "decimals":1
     xiv. },
      xv. "zoom":{
            1. "active":true,
            2. "preserve-zoom":true
     xvi. },
    xvii. "plotarea":{
            1. "margin-top":"20",
            2. "margin-left":"20",
            3. "margin":"dynamic 50 105 dynamic"
   xviii. },
     xix. "plot":{
            1. "data-append-selection":true,
            2. "mode":"normal",
            3. "line-width":1,
            4. "line-color":"#FFAF33",
            5. "background-color":"#1e88e5",
            6. "marker":{
                a. "size":3,
                b. "background-color":"#FFAF33",
                c. "border-width":2,
                d. "border-color":"#FFAF33"
            7. },
            8. "tooltip":{
                a. "visible":true,
                b. "text":"%kv<br>%vv"
            9. },
            10.     "selection-mode":"multiple",
            11.     "selected-state":{
                a. "background-color":"#ffa726",
                b. "border-width":0
            12.     }
      xx. },
     xxi. "series":[
            1. {
                a. "values":ecg,
                b. "text":"ECG"
            2. }
    xxii. ]
   xxiii. }
    xxiv. ]
77.     };

78.     var myChart3 = {
       i. "graphset":[
      ii. {
     iii. "background-color":"#ffffff",
      iv. "type":"line",
       v. "title": {
            1. "text": ""
```

vi. },
vii. "legend":{
1. "toggle-action":"remove",
2. "align":"center",
3. "adjust-layout":false,
4. "borderWidth":0,
5. "verticalAlign":"bottom",
6. "fontColor":"#7d7d7d",
7. "font-size":10,
8. "marginRight":20,
9. "marginBottom":0,
10. "marginTop":0,
11. "marker":{
a. "type":"square"
12. },
viii. },
ix. "preview":{
1. "background-color":"#F5F7F3",
2. "border-width":0,
3. "height":30,
4. "preserve-zoom":false,
5. "mask":{
a. "backgroundColor":"grey",
b. "alpha":0.8
6. },
7. "handle":{
a. "border-width":2
8. },
9. "y":"85%"
x. },
xi. "scale-x":{
1. "zooming":true,
2. "zoom-to":[0,30],
3. "item":{
a. "font-size":10,
b. "font-color":"#7d7d7d"
4. },
5. "tick":{
a. "visible":true
6. },
7. "labels": time,
8. "label":{
9. "text":"Time (HH:MM)",
10. "font-size":"10px",
11. "color":"grey"
12. },
xii. },
xiii. "scale-y":{
1. "zooming":false,
2. "values":"0:1000:10",
3. "guide":{

```
                a. "line-style":"dotted"
            4. },
            5. "item":{
                a. "font-size":10,
                b. "font-color":"#7d7d7d"
            6. },
            7. "tick":{
                a. "visible":true
            8. },
            9. "format":"",
            10.     "decimals":1
        xiv. },
         xv. "zoom":{
            1. "active":true,
            2. "preserve-zoom":true
        xvi. },
       xvii. "plotarea":{
            1. "margin-top":"20",
            2. "margin-left":"20",
            3. "margin":"dynamic 50 105 dynamic"
      xviii. },
        xix. "plot":{
            1. "data-append-selection":true,
            2. "mode":"normal",
            3. "line-width":1,
            4. "line-color":"#33B0FF",
            5. "background-color":"#1e88e5",
            6. "marker":{
                a. "size":3,
                b. "background-color":"#33B0FF",
                c. "border-width":2,
                d. "border-color":"#33B0FF"
            7. },
            8. "tooltip":{
                a. "visible":true,
                b. "text":"%kv<br>%vv"
            9. },
            10.     "selection-mode":"multiple",
            11.     "selected-state":{
                a. "background-color":"#ffa726",
                b. "border-width":0
            12.     }
         xx. },
        xxi. "series":[
            1. {
                a. "values":emg,
                b. "text":"EMG"
            2. }
       xxii. ]
      xxiii. }
       xxiv. ]
79.    };
```

```
80.     var myChart4 = {
81.     "graphset":[
          i. {
         ii. "background-color":"#ffffff",
        iii. "type":"line",
         iv. "title": {
                1. "text": ""
          v. },
         vi. "legend":{
                1. "toggle-action":"remove",
                2. "align":"center",
                3. "adjust-layout":false,
                4. "borderWidth":0,
                5. "verticalAlign":"bottom",
                6. "fontColor":"#7d7d7d",
                7. "font-size":10,
                8. "marginRight":20,
                9. "marginBottom":0,
                10.     "marginTop":0,
                11.     "marker":{
                     a. "type":"square"
                12.     },
        vii. },
       viii. "preview":{
                1. "background-color":"#F5F7F3",
                2. "border-width":0,
                3. "height":30,
                4. "preserve-zoom":false,
                5. "mask":{
                     a. "backgroundColor":"grey",
                     b. "alpha":0.8
                6. },
                7. "handle":{
                     a. "border-width":2
                8. },
                9. "y":"85%"
         ix. },
          x. "scale-x":{
                1. "zooming":true,
                2. "zoom-to":[0,30],
                3. "item":{
                     a. "font-size":10,
                     b. "font-color":"#7d7d7d"
                4. },
                5. "tick":{
                     a. "visible":true
                6. },
                7. "labels": time,
                8. "label":{
                9.                      "text":"Time
                   (HH:MM)",
                10.                          "font-
                  size":"10px",
```

```
            11.                         "color":"grey"
            12.                     },
      xi. },
     xii. "scale-y":{
            1. "zooming":false,
            2. "values":"-0:1000:10",
            3. "guide":{
                 a. "line-style":"dotted"
            4. },
            5. "item":{
                 a. "font-size":10,
                 b. "font-color":"#7d7d7d"
            6. },
            7. "tick":{
                 a. "visible":true
            8. },
            9. "format":"",
            10.     "decimals":1
    xiii. },
     xiv. "zoom":{
            1. "active":true,
            2. "preserve-zoom":true
      xv. },
     xvi. "plotarea":{
            1. "margin-top":"20",
            2. "margin-left":"20",
            3. "margin":"dynamic 50 105 dynamic"
    xvii. },
   xviii. "plot":{
            1. "data-append-selection":true,
            2. "mode":"normal",
            3. "line-width":1,
            4. "line-color":"#5433FF",
            5. "background-color":"#1e88e5",
            6. "marker":{
                 a. "size":3,
                 b. "background-color":"#5433FF",
                 c. "border-width":2,
                 d. "border-color":"#5433FF"
            7. },
            8. "tooltip":{
                 a. "visible":true,
                 b. "text":"%kv<br>%vv"
            9. },
            10.     "selection-mode":"multiple",
            11.     "selected-state":{
                 a. "background-color":"#ffa726",
                 b. "border-width":0
            12.     }
     xix. },
      xx. "series":[
            1. {
                 a. "values":airFlow,
```

```
                    b. "text":"Air Flow"
               2. }
        xxi. ]
       xxii. }
      xxiii. ]
82.     };

83.     var myChart5 = {
          i. "graphset":[
         ii. {
        iii. "background-color":"#ffffff",
         iv. "type":"line",
          v. "title": {
               1. "text": ""
         vi. },
        vii. "legend":{
               1. "toggle-action":"remove",
               2. "align":"center",
               3. "adjust-layout":false,
               4. "borderWidth":0,
               5. "verticalAlign":"bottom",
               6. "fontColor":"#7d7d7d",
               7. "font-size":10,
               8. "marginRight":20,
               9. "marginBottom":0,
               10.    "marginTop":0,
               11.    "marker":{
                    a. "type":"square"
               12.    },
       viii. },
         ix. "preview":{
               1. "background-color":"#F5F7F3",
               2. "border-width":0,
               3. "height":30,
               4. "preserve-zoom":false,
               5. "mask":{
                    a. "backgroundColor":"grey",
                    b. "alpha":0.8
               6. },
               7. "handle":{
                    a. "border-width":2
               8. },
               9. "y":"85%"
          x. },
         xi. "scale-x":{
               1. "zooming":true,
               2. "zoom-to":[0,30],
               3. "item":{
                    a. "font-size":10,
                    b. "font-color":"#7d7d7d"
               4. },
               5. "tick":{
                    a. "visible":true
```

```
          6. },
          7. "labels": time,
          8. "label":{
          9.                     "text":"Time
             (HH:MM)",
          10.                        "font-
             size":"10px",
          11.                        "color":"grey"
          12.                    },
  xii. },
 xiii. "scale-y":{
          1. "zooming":false,
          2. "values":"0:5000000:10",
          3. "guide":{
               a. "line-style":"dotted"
          4. },
          5. "item":{
               a. "font-size":10,
               b. "font-color":"#7d7d7d"
          6. },
          7. "tick":{
               a. "visible":true
          8. },
          9. "format":"",
          10.    "decimals":1
  xiv. },
   xv. "zoom":{
          1. "active":true,
          2. "preserve-zoom":true
  xvi. },
 xvii. "plotarea":{
          1. "margin-top":"20",
          2. "margin-left":"20",
          3. "margin":"dynamic 50 105 dynamic"
xviii. },
  xix. "plot":{
          1. "data-append-selection":true,
          2. "mode":"normal",
          3. "line-width":1,
          4. "line-color":"#EB2E2E",
          5. "background-color":"#1e88e5",
          6. "marker":{
               a. "size":3,
               b. "background-color":"#EB2E2E",
               c. "border-width":2,
               d. "border-color":"#EB2E2E"
          7. },
          8. "tooltip":{
               a. "visible":true,
               b. "text":"%kv<br>%vv"
          9. },
          10.    "selection-mode":"multiple",
          11.    "selected-state":{
```

```
                    a. "background-color":"#ffa726",
                    b. "border-width":0
               12.     }
          xx. },
         xxi. "series":[
               1. {
                    a. "values":skinConductance,
                    b. "text":"Skin Conductance"
               2. }
        xxii. ]
       xxiii. }
        xxiv. ]
84.      };

85.      var myChart6 = {
           i. "graphset":[
          ii. {
         iii. "background-color":"#ffffff",
          iv. "type":"line",
           v. "title": {
               1. "text": ""
          vi. },
         vii. "legend":{
               1. "toggle-action":"remove",
               2. "align":"center",
               3. "adjust-layout":false,
               4. "borderWidth":0,
               5. "verticalAlign":"bottom",
               6. "fontColor":"#7d7d7d",
               7. "font-size":10,
               8. "marginRight":20,
               9. "marginBottom":0,
               10.     "marginTop":0,
               11.     "marker":{
                    a. "type":"square"
               12.     },
        viii. },
          ix. "preview":{
               1. "background-color":"#F5F7F3",
               2. "border-width":0,
               3. "height":30,
               4. "preserve-zoom":false,
               5. "mask":{
                    a. "backgroundColor":"grey",
                    b. "alpha":0.8
               6. },
               7. "handle":{
                    a. "border-width":2
               8. },
               9. "y":"85%"
           x. },
          xi. "scale-x":{
               1. "zooming":true,
```

```
            2. "zoom-to":[0,30],
            3. "item":{
                a. "font-size":10,
                b. "font-color":"#7d7d7d"
            4. },
            5. "tick":{
                a. "visible":true
            6. },
            7. "labels": time,
            8. "label":{
            9.                      "text":"Time
               (HH:MM)",
            10.                          "font-
               size":"10px",
            11.                          "color":"grey"
            12.                      },
    xii. },
   xiii. "scale-y":{
            1. "zooming":false,
            2. "values":"0:5000000:10",
            3. "guide":{
                a. "line-style":"dotted"
            4. },
            5. "item":{
                a. "font-size":10,
                b. "font-color":"#7d7d7d"
            6. },
            7. "tick":{
                a. "visible":true
            8. },
            9. "format":"",
            10.    "decimals":1
    xiv. },
     xv. "zoom":{
            1. "active":true,
            2. "preserve-zoom":true
    xvi. },
   xvii. "plotarea":{
            1. "margin-top":"20",
            2. "margin-left":"20",
            3. "margin":"dynamic 50 105 dynamic"
  xviii. },
    xix. "plot":{
            1. "data-append-selection":true,
            2. "mode":"normal",
            3. "line-width":1,
            4. "line-color":"#4d9900",
            5. "background-color":"#1e88e5",
            6. "marker":{
                a. "size":3,
                b. "background-color":"#4d9900",
                c. "border-width":1,
                d. "border-color":"#4d9900"
```

```
                7. },
                8. "tooltip":{
                    a. "visible":true,
                    b. "text":"%kv<br>%vv"
                9. },
                10.    "selection-mode":"multiple",
                11.    "selected-state":{
                    a. "background-color":"#ffa726",
                    b. "border-width":0
                12.    }
          xx. },
         xxi. "series":[
                1. {
                    a. "values":skinResistance,
                    b. "text":"Skin Resistance (ohm)"
                2. }
        xxii. ]
       xxiii. }
        xxiv. ]
86.     };

87.     var myChart7 = {
           i. "graphset":[
          ii. {
         iii. "background-color":"#ffffff",
          iv. "type":"line",
           v. "title": {
                1. "text": ""
          vi. },
         vii. "legend":{
                1. "toggle-action":"remove",
                2. "align":"center",
                3. "adjust-layout":false,
                4. "borderWidth":0,
                5. "verticalAlign":"bottom",
                6. "fontColor":"#7d7d7d",
                7. "font-size":10,
                8. "marginRight":20,
                9. "marginBottom":0,
                10.    "marginTop":0,
                11.    "marker":{
                    a. "type":"square"
                12.    },
        viii. },
          ix. "preview":{
                1. "background-color":"#F5F7F3",
                2. "border-width":0,
                3. "height":30,
                4. "preserve-zoom":false,
                5. "mask":{
                    a. "backgroundColor":"grey",
                    b. "alpha":0.8
                6. },
```

```
            7. "handle":{
                 a. "border-width":2
            8. },
            9. "y":"85%"
      x. },
     xi. "scale-x":{
            1. "zooming":true,
            2. "zoom-to":[0,30],
            3. "item":{
                 a. "font-size":10,
                 b. "font-color":"#7d7d7d"
            4. },
            5. "tick":{
                 a. "visible":true
            6. },
            7. "labels": time,
            8. "label":{
            9.                     "text":"Time
               (HH:MM)",
            10.                         "font-
               size":"10px",
            11.                         "color":"grey"
            12.                       },
    xii. },
   xiii. "scale-y":{
            1. "zooming":false,
            2. "values":"0:5000000:10",
            3. "guide":{
                 a. "line-style":"dotted"
            4. },
            5. "item":{
                 a. "font-size":10,
                 b. "font-color":"#7d7d7d"
            6. },
            7. "tick":{
                 a. "visible":true
            8. },
            9. "format":"",
            10.    "decimals":1
    xiv. },
     xv. "zoom":{
            1. "active":true,
            2. "preserve-zoom":true
    xvi. },
   xvii. "plotarea":{
            1. "margin-top":"20",
            2. "margin-left":"20",
            3. "margin":"dynamic 50 105 dynamic"
  xviii. },
    xix. "plot":{
            1. "data-append-selection":true,
            2. "mode":"normal",
            3. "line-width":1,
```

```
                4. "line-color":"#3DDA33",
                5. "background-color":"#1e88e5",
                6. "marker":{
                     a. "size":3,
                     b. "background-color":"#3DDA33",
                     c. "border-width":2,
                     d. "border-color":"#3DDA33"
                7. },
                8. "tooltip":{
                     a. "visible":true,
                     b. "text":"%kv<br>%vv"
                9. },
                10.    "selection-mode":"multiple",
                11.    "selected-state":{
                     a. "background-color":"#ffa726",
                     b. "border-width":0
                12.    }
         xx. },
        xxi. "series":[
                1. {
                     a. "values":skinConductanceVoltage,
                     b. "text":"SKin Conductance Voltage
                        (V)"
                2. }
       xxii. ]
      xxiii. }
       xxiv. ]
88.     };

89.     var myChart8 = {
90.     "graphset":[
          i. {
         ii. "background-color":"#ffffff",
        iii. "type":"line",
         iv. "title": {
                1. "text": ""
          v. },
         vi. "legend":{
                1. "toggle-action":"remove",
                2. "align":"center",
                3. "adjust-layout":false,
                4. "borderWidth":0,
                5. "verticalAlign":"bottom",
                6. "fontColor":"#7d7d7d",
                7. "font-size":10,
                8. "marginRight":20,
                9. "marginBottom":0,
                10.    "marginTop":0,
                11.    "marker":{
                     a. "type":"square"
                12.    },
        vii. },
       viii. "preview":{
```

```
        1. "background-color":"#F5F7F3",
        2. "border-width":0,
        3. "height":30,
        4. "preserve-zoom":false,
        5. "mask":{
            a. "backgroundColor":"grey",
            b. "alpha":0.8
        6. },
        7. "handle":{
            a. "border-width":2
        8. },
        9. "y":"85%"
  ix. },
   x. "scale-x":{
        1. "zooming":true,
        2. "zoom-to":[0,30],
        3. "item":{
            a. "font-size":10,
            b. "font-color":"#7d7d7d"
        4. },
        5. "tick":{
            a. "visible":true
        6. },
        7. "labels": time,
        8. "label":{
        9.                 "text":"Time
           (HH:MM)",
        10.                     "font-
           size":"10px",
        11.                     "color":"grey"
        12.                 },
  xi. },
 xii. "scale-y":{
        1. "zooming":false,
        2. "values":"0:230:10",
        3. "guide":{
            a. "line-style":"dotted"
        4. },
        5. "item":{
            a. "font-size":10,
            b. "font-color":"#7d7d7d"
        6. },
        7. "tick":{
            a. "visible":true
        8. },
        9. "format":"",
        10.    "decimals":1
xiii. },
 xiv. "zoom":{
        1. "active":true,
        2. "preserve-zoom":true
  xv. },
 xvi. "plotarea":{
```

```
                1. "margin-top":"20",
                2. "margin-left":"20",
                3. "margin":"dynamic 50 105 dynamic"
      xvii. },
     xviii. "plot":{
                1. "data-append-selection":true,
                2. "mode":"normal",
                3. "line-width":1,
                4. "line-color":"#E1DE2B",
                5. "background-color":"#1e88e5",
                6. "marker":{
                     a. "size":3,
                     b. "background-color":"#E1DE2B",
                     c. "border-width":2,
                     d. "border-color":"#E1DE2B"
                7. },
                8. "tooltip":{
                     a. "visible":true,
                     b. "text":"%kv<br>%vv"
                9. },
                10.    "selection-mode":"multiple",
                11.    "selected-state":{
                     a. "background-color":"#ffa726",
                     b. "border-width":0
                12.    }
       xix. },
        xx. "series":[
                1. {
                     a. "values":beatsPerMinute,
                     b. "text":"Beats Per Minute"
                2. }
       xxi. ]
      xxii. }
     xxiii. ]
91.    };

92.    var myChart9 = {
         i. "graphset":[
        ii. {
       iii. "background-color":"#ffffff",
        iv. "type":"line",
         v. "title": {
                1. "text": ""
        vi. },
       vii. "legend":{
                1. "toggle-action":"remove",
                2. "align":"center",
                3. "adjust-layout":false,
                4. "borderWidth":0,
                5. "verticalAlign":"bottom",
                6. "fontColor":"#7d7d7d",
                7. "font-size":10,
                8. "marginRight":20,
```

9. "marginBottom":0,
10. "marginTop":0,
11. "marker":{
    a. "type":"square"
12. },

viii. },
ix. "preview":{
1. "background-color":"#F5F7F3",
2. "border-width":0,
3. "height":30,
4. "preserve-zoom":false,
5. "mask":{
    a. "backgroundColor":"grey",
    b. "alpha":0.8
6. },
7. "handle":{
    a. "border-width":2
8. },
9. "y":"85%"

x. },
xi. "scale-x":{
1. "zooming":true,
2. "zoom-to":[0,30],
3. "item":{
    a. "font-size":10,
    b. "font-color":"#7d7d7d"
4. },
5. "tick":{
    a. "visible":true
6. },
7. "labels": time,
8. "label":{
9. "text":"Time (HH:MM)",
10. "font-size":"10px",
11. "color":"grey"
12. },

xii. },
xiii. "scale-y":{
1. "zooming":false,
2. "values":"0:100:10",
3. "guide":{
    a. "line-style":"dotted"
4. },
5. "item":{
    a. "font-size":10,
    b. "font-color":"#7d7d7d"
6. },
7. "tick":{
    a. "visible":true
8. },
9. "format":"",

```
                10.    "decimals":1
        xiv. },
         xv. "zoom":{
                1. "active":true,
                2. "preserve-zoom":true
        xvi. },
       xvii. "plotarea":{
                1. "margin-top":"20",
                2. "margin-left":"20",
                3. "margin":"dynamic 50 105 dynamic"
      xviii. },
        xix. "plot":{
                1. "data-append-selection":true,
                2. "mode":"normal",
                3. "line-width":1,
                4. "line-color":"#E12B2B",
                5. "background-color":"#1e88e5",
                6. "marker":{
                     a. "size":3,
                     b. "background-color":"#E12B2B",
                     c. "border-width":2,
                     d. "border-color":"#E12B2B"
                7. },
                8. "tooltip":{
                     a. "visible":true,
                     b. "text":"%kv<br>%vv"
                9. },
                10.    "selection-mode":"multiple",
                11.    "selected-state":{
                     a. "background-color":"#ffa726",
                     b. "border-width":0
                12.    }
         xx. },
        xxi. "series":[
                1. {
                     a. "values":bloodOxygen,
                     b. "text":"Blood Oxygen (%)"
                2. }
       xxii. ]
      xxiii. }
       xxiv. ]
      };


      window.onload=function(){


      zingchart.render({
      id:'chartDiv1',
      data:myChart1,
      height: 400,
      width: '100%'
```

```
});

zingchart.render({
id:'chartDiv2',
data:myChart2,
height: 400,
width: '100%'
});

zingchart.render({
id:'chartDiv3',
data:myChart3,
height: 400,
width: '100%'
});

zingchart.render({
id:'chartDiv4',
data:myChart4,
height: 400,
width: '100%'
});

zingchart.render({
id:'chartDiv5',
data:myChart5,
height: 400,
width: '100%'
});

zingchart.render({
id:'chartDiv6',
data:myChart6,
height: 400,
width: '100%'
});

zingchart.render({
id:'chartDiv7',
data:myChart7,
height: 400,
width: '100%'
});

zingchart.render({
id:'chartDiv8',
data:myChart8,
height: 400,
width: '100%'
});

zingchart.render({
```

```
id:'chartDiv9',
data:myChart9,
height: 400,
width: '100%'
});
};
</script>

</head>
<body>

    <div class="page-header">
    <img src="images/logo.png" alt="logo" width =
    "200" height = "135">
    <h1><b>Your Data For Log </b> <?php echo
    $logid; ?> </h1>
    <p>
        <?php
            $userPage ="";
            $find_user = $_SESSION['username'];
            $username_length =
            strlen($find_user);

            $char = str_split($find_user, 1);
            if ($char[4] == "h")
            {
                $userPage="hcp_welcome.php";
            }
            elseif ($char[4] == "p")
            {
                $userPage="patient_welcome.php
                ";
            }
        ?>
        <a href="<?php echo $userPage ?>"
        class="button">Go Back to Welcome
        Page</a>
    </p>
    </div>


    <h4><p>
    The links below open a new window.
    </p>

    <b><p>
    These links should NOT be used to replace
    advice from
    </p>
    health care professioanls, you must consult a
    doctor if you are concerned.
    </b></h4>
    <br>
```

```
xvi.     <a href="https://patient.info"
         target="_blank" class="button">Patient advice
         Webpage</a>
xvii.    <a href="https://patient.info/symptom-
         checker" target="_blank"
         class="button">Patient symptom checker
         Webpage</a>
xviii.   <noscript>
xix.     <h2>Sorry, javascript is disabled. <br><br>
xx.      Please enable javascript to improve your
         experince with this webpage.</h2>
xxi.     </noscript>

xxii.    <br><br><br><br>
xxiii.   <b><p>Click and drag an araea of the graph to
         zoom, or use the handles below each
         graph</p></b>

xxiv.    <h3>Body Temperature</h3>
xxv.     <div id='chartDiv1'><a
         href="https://www.zingchart.com">Powered by
         ZingChart</a></div>

xxvi.    <h3>Electromyography</h3>
xxvii.   <div id='chartDiv2'><a
         href="https://www.zingchart.com">Powered by
         ZingChart</a></div>

xxviii.  <h3>Electrocardiogram</h3>
xxix.    <div id='chartDiv3'><a
         href="https://www.zingchart.com">Powered by
         ZingChart</a></div>

xxx.     <h3>Air Flow</h3>
xxxi.    <div id='chartDiv4'><a
         href="https://www.zingchart.com">Powered by
         ZingChart</a></div>

xxxii.   <h3>Skin Conductance</h3>
xxxiii.  <div id='chartDiv5'><a
         href="https://www.zingchart.com">Powered by
         ZingChart</a></div>

xxxiv.   <h3>Skin Resistance</h3>
xxxv.    <div id='chartDiv6'><a
         href="https://www.zingchart.com">Powered by
         ZingChart</a></div>

xxxvi.   <h3>Skin Conductance Voltage</h3>
xxxvii.  <div id='chartDiv7'><a
         href="https://www.zingchart.com">Powered by
         ZingChart</a></div>
```

```
xxxviii. <h3>Beats Per Minute</h3>
xxxix. <div id='chartDiv8'><a
       href="https://www.zingchart.com">Powered by
       ZingChart</a></div>

xl. <h3>Blood Oxygen</h3>
b. <div id='chartDiv9'><a
   href="https://www.zingchart.com">Powered by
   ZingChart</a></div>

153.    </body>
154.    </html>
```

hcp_welcome.php

```
<?php

session_start();

if(!isset($_SESSION['username']) || empty($_SESSION['username']))
{
header("location: login_form.php");
exit;
}

    $conn = mysqli_connect("localhost", "healthCareProfessional", "hcpHOPEuniversity#2018", "health_database");

    if (!$conn)
    {
    die("Connection Failed: " . mysqli_connect_error());
    }

    // get an array of logs to display as a table on welcome page
    $get_patients = "SELECT uID FROM patient"; // change to UID
    $patients = mysqli_query($conn, $get_patients) or die(mysqli_error());

    mysqli_close($conn);
    ?>



    <!DOCTYPE html>
    <html lang="en">
    <head>

    <meta charset="UTF-8">

    <title>Welcome</title>

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">
    <link rel="stylesheet" type="text/css" href="welcome_form_CSS.css">

    <style type="text/css">
    body{ font: 14px sans-serif; text-align: center; }

    #input
```

```
{
width: 50%;
font-size: 16px;
padding: 15px 10px 15px 10px;
border: 2px solid #ddd;
margin-bottom: 12px;
position: relative;
}
#patients
{
border-collapse: collapse;
width: 50%;
}

#patients th, #patients td
{
padding: 15px;
border: 1px solid #ddd;
text-align: center;
}
#patients th
{
background-color:#20a354;
color: black;
}

#patients tr:nth-child(even) {background-color: #e1e1e1;}
#patients tr:hover {background-color:#f1f1f1;}

div.patient_table
{
position: relative;
left: 60px;
bottom: -50px;
}

</style>

</head>
<body>

<div class="page-header">
<img src="images/logo.png" alt="logo" width = "200" height = "135">
<h1>Welcome <b><?php echo $_SESSION['username']; ?></b></h1>
</div>

<div>
<a href="logout.php" class="button">Log Out of Your Account</a>
</div>
```

```

<div class = patient_table style = "overflow-y: auto;">
<form action = "patient_hcp_log.php" method = "post" >

<input type="text" id="input" onkeyup="searchPatientTable()" placeholder="Search for patient ..." title="Type in a patient ID">

<table id = "patients">
<th> Select </th> <th> Patient </th>
<?php
while ($lines = mysqli_fetch_array($patients))
{
echo ("<tr><td><input type='radio' name='patient' value=".$lines['uID']." required></td>");
echo ("<td>" . $lines['uID']. "</td></tr>");
}
?>
</table>
<br>
<input type = "submit" class = "button" value = "Click To View Chosen Log" style= "float:left">
</br>
</form>
</div>
<script>
function searchPatientTable()
{
var input, filter, table, tr, td, i;
input = document.getElementById("input");
filter = input.value.toUpperCase();
table = document.getElementById("patients");
tr = table.getElementsByTagName("tr");

for (i = 0; i < tr.length; i++)
{
td = tr[i].getElementsByTagName("td")[0];
if (td)
{
if (td.innerHTML.toUpperCase().indexOf(filter) > -1)
{
tr[i].style.display = "";
} else
{
tr[i].style.display = "none";
}
}
}
}
</script>
```

```
</body>
</html>
```

patient_hcp_log.php

```
<?php

session_start();

if(!isset($_SESSION['username']) || empty($_SESSION['username']))
{
header("location: hcp_welcome.php");
exit;
}

	$conn = mysqli_connect("localhost", "healthCareProfessional", "hcpHOPEuniversity#2018", "health_database");

	if (!$conn)
	{
	die("Connection Failed: " . mysqli_connect_error());
	}

	if(isset($_POST) && 0 === count($_POST))
	{
	echo ("<h1>Check Box was NOT Selected</h1><p><h2>Please go back and try again</h2></p>");
	}

	$chosen_patient = $_POST["patient"];
	// get an array of logs to display as a tbale on welcome page
	$get_patientLog = "SELECT * FROM patient_log WHERE uID = $chosen_patient";
	$patient_logs = mysqli_query($conn, $get_patientLog) or die(mysqli_error());
	$numOfRows = mysqli_num_rows($patient_logs);

	mysqli_close($conn);
	?>



	<!DOCTYPE html>
	<html lang="en">
	<head>

	<meta charset="UTF-8">

	<title>Welcome</title>

```

```
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">
    <link rel="stylesheet" type="text/css" href="welcome_form_CSS.css">
    <style type="text/css">
    body{ font: 14px sans-serif; text-align: center; }
    #logs
    {
    border-collapse: collapse;
    width: 50%;
    }
    #logs th, #logs td
    {
    padding: 15px;
    border: 1px solid #ddd;
    text-align: center;
    }
    #logs th
    {
    background-color:#20a354;
    color: black;
    }
    #logs tr:nth-child(even) {background-color: #e1e1e1;}
    #logs tr:hover {background-color:#f1f1f1;}
    div.log_table
    {
    position: relative;
    left: 60px;
    bottom: -50px;
    }
    </style>
    </head>
    <body>
    <div class="page-header">
    <img src="images/logo.png" alt="logo" width = "200" height = "135">
    <h1>Logs for patient: <b><?php echo $chosen_patient; ?></b></h1>
    </div>
    <div>
```

```
<a href="hcp_welcome.php" class="button">Go Back</a>
</div>
<div class = log_table style = "overflow-y: auto;">
<form action = "graphical_data.php" method = "post">
<?php
if($numOfRows >0)
{
echo("<table id = 'logs'> ");
echo("<th> Select </th><th> Log </th> <th> Date </th>");
while ($lines = mysqli_fetch_array($patient_logs))
{
echo ("<tr><td><input type='radio' name='logID' value=".$lines['logID']." required></td>");
echo ("<td>" . $lines['logID']. "</td> ");
echo ("<td>" . $lines['dateValue']. "</td></tr>");
}
echo(" </table>
<br>
<input type = 'submit' class = 'button' value = 'Click To View Chosen Log' style= 'float:left'>
</br>");
}
elseif ($numOfRows == 0)
{?>
<h1><b>No Patient Logs Found for selected patient</b></h1>
<?php
}
?>
</form>
</div>
</body>
</html>
```

4.0 MySQL Code

Database Tables MySQL

```
CREATE DATABASE health_database;

USE health_database;

CREATE TABLE surgery (
surgeryID  INTEGER NOT NULL AUTO_INCREMENT,
name  VARCHAR(30) NOT NULL,
addressL1  VARCHAR(100) NOT NULL,
addressL2  VARCHAR(100) NOT NULL,
postcode  VARCHAR(7) NOT NULL,
contactNumber BIGINT,

PRIMARY KEY (surgeryID)

);

CREATE TABLE health_care_professional (
gmc INTEGER (7) NOT NULL UNIQUE,
forename VARCHAR(30) NOT NULL,
surname VARCHAR(30) NOT NULL,
gender char NOT NULL,
addressL1 VARCHAR(50) NOT NULL,
addressL2 VARCHAR(50) NOT NULL,
postcode VARCHAR(7) NOT NULL,
contactNumber BIGINT,
emailAddress VARCHAR(255),

PRIMARY KEY (gmc)

);

CREATE TABLE surgery_HCP_relation (
surgeryID INTEGER NOT NULL,
gmc INTEGER NOT NULL,

FOREIGN KEY (surgeryID) REFERENCES surgery(surgeryID)
ON UPDATE CASCADE ON DELETE RESTRICT,
FOREIGN KEY (gmc) REFERENCES health_care_professional(gmc)
ON UPDATE CASCADE ON DELETE RESTRICT

);

CREATE TABLE patient (
uID INTEGER NOT NULL UNIQUE AUTO_INCREMENT,
nhsNumber BIGINT NOT NULL UNIQUE,
forename VARCHAR(30) NOT NULL,
surname VARCHAR(30) NOT NULL,
```

```
dateOfBirth DATE NOT NULL,
gender CHAR NOT NULL,
addressL1 VARCHAR(50) NOT NULL,
addressL2 VARCHAR(50) NOT NULL,
postcode VARCHAR(7) NOT NULL,
contactNumber BIGINT,
emailAddress VARCHAR(255),
PRIMARY KEY (uID)
);
ALTER TABLE patient AUTO_INCREMENT = 1001;
CREATE TABLE surgery_Patient_relation (
surgeryID INTEGER NOT NULL,
uID INTEGER NOT NULL,
FOREIGN KEY (surgeryID) REFERENCES surgery(surgeryID)
ON UPDATE CASCADE ON DELETE RESTRICT,
FOREIGN KEY (uID) REFERENCES patient(uID)
ON UPDATE CASCADE ON DELETE RESTRICT
);
CREATE TABLE patient_HCP_relation (
uID INTEGER NOT NULL,
gmc INTEGER NOT NULL,
dateOfRegistration DATE NOT NULL,
FOREIGN KEY (uID) REFERENCES patient(uID)
ON UPDATE CASCADE ON DELETE RESTRICT,
FOREIGN KEY (gmc) REFERENCES health_care_professional(gmc)
ON UPDATE CASCADE ON DELETE RESTRICT
);
CREATE TABLE patient_log (
logID INTEGER NOT NULL AUTO_INCREMENT,
dateValue DATE NOT NULL,
uID INTEGER NOT NULL,
PRIMARY KEY (logID),
FOREIGN KEY (uID) REFERENCES patient(uID)
ON UPDATE CASCADE ON DELETE RESTRICT
);
CREATE TABLE monitor_data (
id INTEGER NOT NULL AUTO_INCREMENT,
logID INTEGER NOT NULL,
timeValue VARCHAR(9) NOT NULL,
```

```
temperature FLOAT,
emg INTEGER,
ecg FLOAT,
airFlow INTEGER,
skinConductance FLOAT,
skinResistance FLOAT,
skinConductanceVoltage FLOAT,
beatsPerMinute INTEGER,
bloodOxygen TINYINT(101),
PRIMARY KEY (id),
FOREIGN KEY (logID) REFERENCES patient_log(logID)
ON UPDATE CASCADE ON DELETE RESTRICT
);
CREATE TABLE authentication(
id INTEGER NOT NULL AUTO_INCREMENT,
username VARCHAR(20) NOT NULL UNIQUE,
password VARCHAR(255) NOT NULL,
dateCreated DATETIME DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY (id)
);
CREATE TABLE patient_authentication (
id INTEGER NOT NULL AUTO_INCREMENT,
username VARCHAR(20) NOT NULL,
uID INTEGER NOT NULL,
PRIMARY KEY (id),
FOREIGN KEY (username) REFERENCES authentication(username)
ON UPDATE CASCADE ON DELETE RESTRICT,
FOREIGN KEY (uID) REFERENCES patient(uID)
ON UPDATE CASCADE ON DELETE RESTRICT
);
CREATE TABLE hcp_authentication (
id INTEGER NOT NULL AUTO_INCREMENT,
username VARCHAR(20) NOT NULL,
gmc INTEGER NOT NULL,
PRIMARY KEY (id),
FOREIGN KEY (username) REFERENCES authentication(username)
ON UPDATE CASCADE ON DELETE RESTRICT,
FOREIGN KEY (gmc) REFERENCES health_care_professional(gmc)
```

```
ON UPDATE CASCADE ON DELETE RESTRICT

);
```

Database Users MySQL

1. CREATE USER 'healthCareProfessional'@'localhost' IDENTIFIED BY 'hcpHOPEuniversity#2018';
2. CREATE USER 'patient'@'localhost' IDENTIFIED BY 'patientHOPEuniversity#2018';
3. CREATE USER 'admin'@'localhost' IDENTIFIED BY 'adminHOPEuniversity#2018';

4. GRANT ALL ON health_database.* TO 'admin'@'localhost';

5. GRANT SELECT ON health_database.surgery TO 'healthCareProfessional'@'localhost';
6. GRANT SELECT ON health_database.health_care_professional TO 'healthCareProfessional'@'localhost';
7. GRANT SELECT ON health_database.surgery_HCP_relation TO 'healthCareProfessional'@'localhost';
8. GRANT SELECT ON health_database.patient TO 'healthCareProfessional'@'localhost';
9. GRANT SELECT ON health_database.surgery_Patient_relation TO 'healthCareProfessional'@'localhost';
10. GRANT SELECT ON health_database.patient_HCP_relation TO 'healthCareProfessional'@'localhost';
11. GRANT SELECT, INSERT, DELETE ON health_database.patient_log TO 'healthCareProfessional'@'localhost';
12. GRANT SELECT, INSERT, DELETE ON health_database.monitor_data TO 'healthCareProfessional'@'localhost';

13. GRANT SELECT ON health_database.surgery TO 'patient'@'localhost';
14. GRANT SELECT ON health_database.patient TO 'patient'@'localhost';
15. GRANT SELECT ON health_database.patient_authentication TO 'patient'@'localhost';
16. GRANT SELECT ON health_database.patient_HCP_relation TO 'patient'@'localhost';
17. GRANT SELECT, INSERT ON health_database.patient_log TO 'patient'@'localhost';
18. GRANT SELECT, INSERT ON health_database.monitor_data TO 'patient'@'localhost';

Database Data Inserts MySQL

```
1. INSERT INTO surgery (surgeryID, name, addressL1,
   addressL2, postcode, contactNumber) VALUES ('1','Park
   House', '7', 'Coleman', '30343', '4044632335');
2.
3. INSERT INTO health_care_professional (gmc, forename,
   surname, gender, addressL1, addressL2, postcode,
   contactNumber, emailAddress) VALUES (2116803, 'Meagan',
   'Cosker', 'F', '35843', 'Buhler', '6454', '6903658831',
   'mcosker1@ucla.edu');
4. INSERT INTO health_care_professional (gmc, forename,
   surname, gender, addressL1, addressL2, postcode,
   contactNumber, emailAddress) VALUES (5224053, 'Jere',
   'Velte', 'M', '01', 'Duke', '4411', '4296770517',
   'jvelte2@wp.com');
5. INSERT INTO health_care_professional (gmc, forename,
   surname, gender, addressL1, addressL2, postcode,
   contactNumber, emailAddress) VALUES (4481522, 'Udell',
   'Kynsey', 'M', '01', 'Hayes', 'ff44ff', '6212308702',
   'ukynsey6@jimdo.com');
6.
7. INSERT INTO surgery_HCP_relation
8. (surgeryID, gmc)
9.
10.     VALUES
11.     ('1', '2026667'), ('1', '2116803'), ('1',
   '5224053'), ('1', '5312210'), ('1', '6204696'), ('1',
   '3379341'), ('1', '4481522');

12.     INSERT INTO patient (nhsNumber, forename, surname,
   dateOfBirth, gender, addressL1, addressL2, postcode,
   contactNumber, emailAddress) VALUES ('4405798222',
   'Linus', 'Gudger', '1957/04/11', 'M', '7', 'Hagan',
   'hh555gg', '8935283617', 'lgudger0@phoca.cz');
13.     INSERT INTO patient (nhsNumber, forename, surname,
   dateOfBirth, gender, addressL1, addressL2, postcode,
   contactNumber, emailAddress) VALUES ('5912891348',
   'Kylynn', 'Escalera', '2006/12/06', 'F', '512',
   'Becker', '618742', '7226204482',
   'kescalera1@indiegogo.com');
14.     INSERT INTO patient (nhsNumber, forename, surname,
   dateOfBirth, gender, addressL1, addressL2, postcode,
   contactNumber, emailAddress) VALUES ('8462672279',
   'Hamil', 'Francescotti', '1949/10/25', 'M', '21855',
   'Westerfield', 'BD23', '7726386118',
   'hfrancescotti2@dion.ne.jp');
15.     INSERT INTO patient (nhsNumber, forename, surname,
   dateOfBirth, gender, addressL1, addressL2, postcode,
   contactNumber, emailAddress) VALUES ('1055547479',
   'Romain', 'Ondra', '1916/07/19', 'M', '02', 'Golf
   View','jj77hhh', '3768499646', 'rondra3@dot.gov');
```

```
    INSERT INTO patient (nhsNumber, forename, surname, dateOfBirth, gender, addressL1, addressL2, postcode, contactNumber, emailAddress) VALUES ('3490429613', 'Marylee', 'Lampett', '1906/01/06', 'F', '85822', 'Heffernan', '5555555', '6316080802', 'mlampett4@state.gov');
    INSERT INTO patient (nhsNumber, forename, surname, dateOfBirth, gender, addressL1, addressL2, postcode, contactNumber, emailAddress) VALUES ('3481738862', 'Ruthann', 'Croxall', '1953/07/02', 'F', '62152', 'Eagan', 'yt55ff', '7082702598', 'rcroxall5@illinois.edu');
    INSERT INTO patient (nhsNumber, forename, surname, dateOfBirth, gender, addressL1, addressL2, postcode, contactNumber, emailAddress) VALUES ('4037178559', 'Emiline', 'Mokes', '1908/05/10', 'F', '21308', 'Talisman', '33-115', '9818641745', 'emokes6@macromedia.com');
    INSERT INTO patient (nhsNumber, forename, surname, dateOfBirth, gender, addressL1, addressL2, postcode, contactNumber, emailAddress) VALUES ('5298427391', 'Lidia', 'Boland', '1964/06/21', 'F', '2233', 'Center', '44273', '2858244874', 'lboland7@japanpost.jp');
    INSERT INTO patient (nhsNumber, forename, surname, dateOfBirth, gender, addressL1, addressL2, postcode, contactNumber, emailAddress) VALUES ('0516620789', 'Ninnette', 'Salvadore', '1955/03/11', 'F', '50124', 'Schmedeman', 'ffyy66', '2824117254', 'nsalvadore8@wiley.com');
    INSERT INTO patient (nhsNumber, forename, surname, dateOfBirth, gender, addressL1, addressL2, postcode, contactNumber, emailAddress) VALUES ('7395584163', 'Sayer', 'Hanlon', '1925/10/11', 'M', '51842', 'Anniversary', '271145', '8533813787', 'shanlon9@gmpg.org');

    INSERT INTO surgery_patient_relation (surgeryID, uID) VALUES ('1', '1001'),('1', '1002'),('1', '1003'),('1', '1004'),('1', '1005'),('1', '1006'),('1', '1007'),('1', '1008'),('1', '1009'),('1', '1010');


    INSERT INTO patient_HCP_relation
    (uID, gmc, dateOfRegistration)

    VALUES
    ('1001', '2116803', '2007/01/01'), ('1002', '5224053', '2000/02/02'), ('1003', '4481522', '2001/03/03'), ('1008','2116803', '1990/03/09'), ('1007', '5224053', '2016/10/04'), ('1004', '4481522', '2018/05/05'), ('1009', '2116803', '2017/11/05'), ('1010', '5224053', '2005/06/07'), ('1005', '4481522', '2010/01/07'), ('1006','4481522','2011/09/03');
```

```
INSERT INTO patient_log
(dateValue, uID)

VALUES
('2017/11/23', '1001'), ('2017/11/24', '1001');

INSERT INTO monitor_data
(logID, timeValue, temperature, emg, ecg, airFlow, skinConductance, skinResistance, skinConductanceVoltage, beatsPerMinute, bloodOxygen)

VALUES
('1','279','45.80','105','0.55','130','-1.00','-1.00','0.26','0','90'),
('1','372','46.16','105','0.53','125','-1.00','-1.00','0.33','0','90'),
('1','465','46.93','103','0.50','126','-1.00','14614300.00','0.50','0','90'),
('1','558','45.76','101','0.53','121','-1.00','-1.00','0.22','0','90'),
('1','651','46.44','103','0.51','121','-1.00','-1.00','0.40','0','90'),
('1','744','46.81','104','0.50','125','-1.00','3788892.50','0.48','0','90'),
('1','837','45.68','102','0.54','120','-1.00','-1.00','0.22','0','90'),
('1','930','46.52','103','0.51','122','-1.00','-1.00','0.43','0','90'),
('1','1024','46.68','104','0.50','125','-1.00','2764867.50','0.46','0','90'),
('1','1117','45.68','101','0.54','119','-1.00','-1.00','0.20','0','90'),
('1','1210','46.56','104','0.51','123','-1.00','-1.00','0.46','0','90'),
('1','1303','46.48','104','0.51','125','-1.00','1526864.40','0.44','0','90'),
('1','1396','45.72','101','0.53','119','-1.00','-1.00','0.21','0','90'),
('1','1489','46.73','104','0.50','123','-1.00','-1.00','0.48','0','90'),
('1','1583','46.44','103','0.51','126','-1.00','2176597.80','0.39','0','90'),
('1','1676','45.72','100','0.53','118','-1.00','-1.00','0.22','0','90'),
('1','1769','46.73','104','0.50','124','-1.00','-1.00','0.51','0','90'),
('1','1862','46.36','104','0.51','124','-1.00','6017652.50','0.39','0','90'),
```

```
('1','1955','45.80','101','0.53','117','-1.00','-1.00','0.23','0','90'),
('1','2049','46.81','103','0.50','123','-1.00','-1.00','0.52','0','90'),
('1','2142','46.20','103','0.52','124','-1.00','6017652.50','0.36','0','90'),
('1','2235','45.88','101','0.53','119','-1.00','-1.00','0.23','0','90'),
('1','2328','46.85','104','0.50','124','-1.00','-1.00','0.50','0','90'),
('1','2421','46.12','103','0.52','123','-1.00','14614300.00','0.34','0','90'),
('1','2514','45.88','102','0.53','119','-1.00','-1.00','0.24','0','90'),
('1','2608','46.97','104','0.50','123','-1.00','-1.00','0.52','0','90'),
('1','2701','45.96','104','0.53','123','-1.00','-1.00','0.31','0','90'),
('1','2794','45.96','102','0.53','119','-1.00','-1.00','0.26','0','90'),
('1','2886','46.93','103','0.49','124','-1.00','-1.00','0.53','0','90'),
('1','2979','45.96','104','0.53','123','-1.00','-1.00','0.28','0','90'),
('1','3073','46.04','103','0.53','120','-1.00','-1.00','0.31','0','90'),
('1','3166','47.01','104','0.50','125','-1.00','-1.00','0.52','0','90'),
('1','3259','45.88','102','0.53','120','-1.00','-1.00','0.25','0','90'),
('1','3352','46.16','103','0.53','120','-1.00','-1.00','0.35','0','90'),
('1','3445','46.85','104','0.50','125','-1.00','-1.00','0.51','0','90'),
('1','3538','45.76','102','0.53','121','-1.00','-1.00','0.23','0','90'),
('1','3632','46.28','104','0.52','120','-1.00','-1.00','0.39','0','90'),
('1','3725','46.85','104','0.50','125','-1.00','6017652.50','0.50','0','90'),
('1','3818','45.76','101','0.52','120','-1.00','-1.00','0.22','0','90'),
('1','3911','46.44','102','0.51','121','-1.00','-1.00','0.41','0','90'),
('1','4004','46.77','105','0.51','125','-1.00','2176597.80','0.49','0','90'),
('1','4098','45.72','102','0.54','120','-1.00','-1.00','0.21','0','90'),
('1','4191','46.48','104','0.52','122','-1.00','-1.00','0.43','0','90');


INSERT INTO monitor_data
```

```
    (logID, timeValue, temperature, emg, ecg, airFlow, skinConductance, skinResistance, skinConductanceVoltage, beatsPerMinute, bloodOxygen)

    VALUES
    ('2','0','31.15','414','3.39','0','1.63','-1.00','0.59','0','90'),
('2','58','31.12','312','3.05','1','1.73','-1.00','0.57','0','90'),
    ('2','115','31.12','269','2.45','0','1.53','1175861.60','0.50','0','90'),
    ('2','174','31.15','296','1.96','0','-1.00','651592.44','0.45','0','90'),
    ('2','232','31.15','344','1.44','0','-1.00','612575.00','0.41','0','90'),
    ('2','290','31.15','443','1.32','1','-1.00','577966.25','0.33','0','90'),
    ('2','348','31.12','456','1.20','0','-1.00','746715.38','0.33','0','90'),
    ('2','406','31.12','668','1.87','0','-1.00','6017652.50','0.38','0','90'),
    ('2','464','31.09','685','2.21','0','-1.00','-1.00','0.49','0','90'),
    ('2','523','31.12','611','2.67','2','-1.00','-1.00','0.58','0','90'),
    ('2','580','31.15','507','3.18','0','1.34','-1.00','0.60','0','90'),
    ('2','638','31.12','404','3.36','0','2.12','-1.00','0.59','0','90'),
    ('2','697','31.12','300','2.92','0','1.73','-1.00','0.57','0','90'),
    ('2','754','31.12','280','2.43','1','1.93','1526864.40','0.51','0','90'),
    ('2','813','31.12','300','1.81','2','-1.00','695918.44','0.46','0','90'),
    ('2','871','31.12','382','1.51','9','-1.00','651592.44','0.41','0','90'),
    ('2','929','31.12','484','1.47','0','-1.00','612575.00','0.35','0','90'),
    ('2','987','31.12','627','1.98','0','-1.00','874358.88','0.35','0','90'),
    ('2','1045','31.12','668','1.95','0','-1.00','14614300.00','0.39','0','90'),
    ('2','1103','31.12','670','2.29','0','-1.00','-1.00','0.47','0','90'),
    ('2','1162','31.12','602','2.71','0','-1.00','-1.00','0.56','0','90'),
    ('2','1219','31.12','485','3.10','10','1.05','-1.00','0.58','0','90'),
    ('2','1277','31.12','374','3.20','0','1.34','-1.00','0.58','0','90'),
    ('2','1336','31.12','295','2.84','1','1.53','-1.00','0.57','0','90'),
```

```
('2','1394','31.12','259','2.21','2','1.44','14614300.00','0.52','0','90'),
('2','1452','31.12','286','1.77','0','-1.00','746715.38','0.48','0','90'),
('2','1510','31.12','353','1.46','11','-1.00','651592.44','0.43','0','90'),
('2','1568','31.12','164','0.89','0','-1.00','651592.44','0.36','0','90'),
('2','1627','31.12','538','1.51','1','-1.00','746715.38','0.34','0','90'),
('2','1685','31.12','600','1.66','0','-1.00','2176597.80','0.36','0','90'),
('2','1742','31.12','641','1.99','0','-1.00','-1.00','0.44','0','90'),
('2','1801','31.12','627','2.49','0','-1.00','-1.00','0.55','0','90'),
('2','1859','31.04','536','2.94','0','-1.00','-1.00','0.57','0','90'),
('2','1917','31.12','411','3.19','2','1.34','-1.00','0.58','0','90'),
('2','1975','31.12','309','2.95','0','1.44','-1.00','0.58','0','90'),
('2','2033','31.15','272','2.54','0','1.44','-1.00','0.54','0','90'),
('2','2092','31.12','284','2.00','0','1.05','547059.00','0.49','0','90'),
('2','2150','31.15','134','1.10','1','-1.00','651592.44','0.44','0','90'),
('2','2207','31.12','421','1.46','1','-1.00','651592.44','0.38','0','90'),
('2','2266','31.12','495','1.37','0','-1.00','805511.81','0.34','0','90'),
('2','2324','31.02','576','1.57','0','-1.00','1526864.40','0.36','0','90'),
('2','2382','31.12','646','1.95','3','-1.00','2176597.80','0.42','0','90'),
('2','2441','31.15','652','2.39','0','-1.00','-1.00','0.52','0','90'),
('2','2498','31.12','566','2.84','0','-1.00','-1.00','0.57','0','90'),
('2','2556','31.12','434','3.09','0','1.34','-1.00','0.58','0','90'),
('2','2615','31.12','359','3.22','0','1.53','-1.00','0.57','0','90');

INSERT INTO authentication
(username, password)

VALUES
('0001plg',  SHA2('plg', 512)), ('0002pke', SHA2('pke', 512)), ('0003phf',  SHA2('phf', 512)), ('0004pro',  SHA2('pro', 512)), ('0005pml', SHA2('pml', 512)), ('0006prc',  SHA2('prc', 512)),
```

```
('0007pem',  SHA2('pem', 512)), ('0008plb',
SHA2('plb', 512)), ('0009pns',  SHA2('pns', 512)),
('0010psh',  SHA2('psh', 512)), ('0011hcpmc',
SHA2('hcpmc', 512)), ('0009hcpjv',  SHA2('hcpjv',
512)), ('0008hcpuk',  SHA2('hcpuk', 512));

INSERT INTO patient_authentication
(uID, username)

VALUES
('1001', '0001plg'), ('1002', '0002pke'), ('1003',
'0003phf'), ('1008', '0004pro'), ('1007', '0005pml'),
('1004', '0006prc'), ('1010', '0007pem'), ('1005',
'0008plb'), ('1006', '0009pns'), ('1009', '0010psh');

INSERT INTO hcp_authentication
(gmc, username)

VALUES
('2116803', '0011hcpmc'), ('5224053',
'0009hcpjv'), ('4481522', '0008hcpuk');
```