

Data Presentation and Management

Sub-System Technical Design

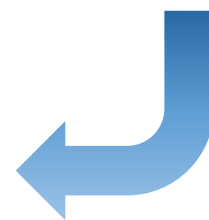
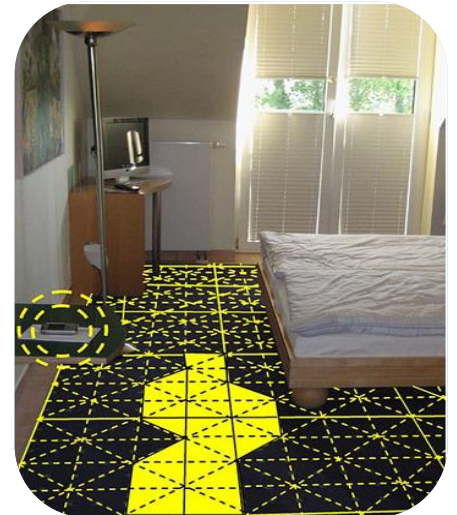
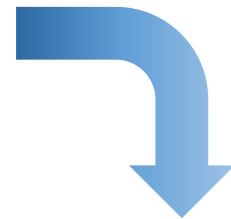
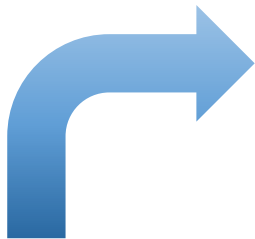
Candidate Number: 189025267

Candidate Name: Benjamin Scriven

Integrated Mechanical and Electrical engineering

GDBP

Group 7



ALUS PENSE [Overview](#) [Full Information](#) [Call Analysis](#)

Full Name: James Cook Care Home Location: Bristol
Room Number: 4 DOB: 2000-09-20

Start Date: 20-02-2019 Number of records: 3

End Date: 19-05-2022

Date	Time	X/Y	Fall Flag
2019-02-20	10:30:11	2500	
2019-02-20	10:30:12	2500	
2019-02-20	10:30:13	2500	
2019-02-20	10:30:14	2500	
2019-02-20	10:30:15	2500	
2019-02-20	10:30:16	2500	
2019-02-20	10:30:17	2500	
2019-02-20	10:30:18	2500	
2019-02-20	10:30:19	2500	
2019-02-20	10:30:20	2500	
2019-02-20	10:30:21	2500	
2019-02-20	10:30:22	2500	
2019-02-20	10:30:23	2500	
2019-02-20	10:30:24	2500	
2019-02-20	10:30:25	2500	
2019-02-20	10:30:26	2500	
2019-02-20	10:30:27	2500	
2019-02-20	10:30:28	2500	
2019-02-20	10:30:29	2500	
2019-02-20	10:30:30	2500	
2019-02-20	10:30:31	2500	
2019-02-20	10:30:32	2500	
2019-02-20	10:30:33	2500	
2019-02-20	10:30:34	2500	
2019-02-20	10:30:35	2500	
2019-02-20	10:30:36	2500	
2019-02-20	10:30:37	2500	
2019-02-20	10:30:38	2500	
2019-02-20	10:30:39	2500	
2019-02-20	10:30:40	2500	
2019-02-20	10:30:41	2500	
2019-02-20	10:30:42	2500	
2019-02-20	10:30:43	2500	
2019-02-20	10:30:44	2500	
2019-02-20	10:30:45	2500	
2019-02-20	10:30:46	2500	
2019-02-20	10:30:47	2500	
2019-02-20	10:30:48	2500	
2019-02-20	10:30:49	2500	
2019-02-20	10:30:50	2500	
2019-02-20	10:30:51	2500	
2019-02-20	10:30:52	2500	
2019-02-20	10:30:53	2500	
2019-02-20	10:30:54	2500	
2019-02-20	10:30:55	2500	
2019-02-20	10:30:56	2500	
2019-02-20	10:30:57	2500	
2019-02-20	10:30:58	2500	
2019-02-20	10:30:59	2500	
2019-02-20	10:31:00	2500	
2019-02-20	10:31:01	2500	
2019-02-20	10:31:02	2500	
2019-02-20	10:31:03	2500	
2019-02-20	10:31:04	2500	
2019-02-20	10:31:05	2500	
2019-02-20	10:31:06	2500	
2019-02-20	10:31:07	2500	
2019-02-20	10:31:08	2500	
2019-02-20	10:31:09	2500	
2019-02-20	10:31:10	2500	
2019-02-20	10:31:11	2500	
2019-02-20	10:31:12	2500	
2019-02-20	10:31:13	2500	
2019-02-20	10:31:14	2500	
2019-02-20	10:31:15	2500	
2019-02-20	10:31:16	2500	
2019-02-20	10:31:17	2500	
2019-02-20	10:31:18	2500	
2019-02-20	10:31:19	2500	
2019-02-20	10:31:20	2500	
2019-02-20	10:31:21	2500	
2019-02-20	10:31:22	2500	
2019-02-20	10:31:23	2500	
2019-02-20	10:31:24	2500	
2019-02-20	10:31:25	2500	
2019-02-20	10:31:26	2500	
2019-02-20	10:31:27	2500	
2019-02-20	10:31:28	2500	
2019-02-20	10:31:29	2500	
2019-02-20	10:31:30	2500	
2019-02-20	10:31:31	2500	
2019-02-20	10:31:32	2500	
2019-02-20	10:31:33	2500	
2019-02-20	10:31:34	2500	
2019-02-20	10:31:35	2500	
2019-02-20	10:31:36	2500	
2019-02-20	10:31:37	2500	
2019-02-20	10:31:38	2500	
2019-02-20	10:31:39	2500	
2019-02-20	10:31:40	2500	
2019-02-20	10:31:41	2500	
2019-02-20	10:31:42	2500	
2019-02-20	10:31:43	2500	
2019-02-20	10:31:44	2500	
2019-02-20	10:31:45	2500	
2019-02-20	10:31:46	2500	
2019-02-20	10:31:47	2500	
2019-02-20	10:31:48	2500	
2019-02-20	10:31:49	2500	
2019-02-20	10:31:50	2500	
2019-02-20	10:31:51	2500	
2019-02-20	10:31:52	2500	
2019-02-20	10:31:53	2500	
2019-02-20	10:31:54	2500	
2019-02-20	10:31:55	2500	
2019-02-20	10:31:56	2500	
2019-02-20	10:31:57	2500	
2019-02-20	10:31:58	2500	
2019-02-20	10:31:59	2500	
2019-02-20	10:32:00	2500	
2019-02-20	10:32:01	2500	
2019-02-20	10:32:02	2500	
2019-02-20	10:32:03	2500	
2019-02-20	10:32:04	2500	
2019-02-20	10:32:05	2500	
2019-02-20	10:32:06	2500	
2019-02-20	10:32:07	2500	
2019-02-20	10:32:08	2500	
2019-02-20	10:32:09	2500	
2019-02-20	10:32:10	2500	
2019-02-20	10:32:11	2500	
2019-02-20	10:32:12	2500	
2019-02-20	10:32:13	2500	
2019-02-20	10:32:14	2500	
2019-02-20	10:32:15	2500	
2019-02-20	10:32:16	2500	
2019-02-20	10:32:17	2500	
2019-02-20	10:32:18	2500	
2019-02-20	10:32:19	2500	
2019-02-20	10:32:20	2500	
2019-02-20	10:32:21	2500	
2019-02-20	10:32:22	2500	
2019-02-20	10:32:23	2500	
2019-02-20	10:32:24	2500	
2019-02-20	10:32:25	2500	
2019-02-20	10:32:26	2500	
2019-02-20	10:32:27	2500	
2019-02-20	10:32:28	2500	
2019-02-20	10:32:29	2500	
2019-02-20	10:32:30	2500	
2019-02-20	10:32:31	2500	
2019-02-20	10:32:32	2500	
2019-02-20	10:32:33	2500	
2019-02-20	10:32:34	2500	
2019-02-20	10:32:35	2500	
2019-02-20	10:32:36	2500	
2019-02-20	10:32:37	2500	
2019-02-20	10:32:38	2500	
2019-02-20	10:32:39	2500	
2019-02-20	10:32:40	2500	
2019-02-20	10:32:41	2500	
2019-02-20	10:32:42	2500	
2019-02-20	10:32:43	2500	
2019-02-20	10:32:44	2500	
2019-02-20	10:32:45	2500	
2019-02-20	10:32:46	2500	
2019-02-20	10:32:47	2500	
2019-02-20	10:32:48	2500	
2019-02-20	10:32:49	2500	
2019-02-20	10:32:50	2500	
2019-02-20	10:32:51	2500	
2019-02-20	10:32:52	2500	
2019-02-20	10:32:53	2500	
2019-02-20	10:32:54	2500	
2019-02-20	10:32:55	2500	
2019-02-20	10:32:56	2500	
2019-02-20	10:32:57	2500	
2019-02-20	10:32:58	2500	
2019-02-20	10:32:59	2500	
2019-02-20	10:33:00	2500	
2019-02-20	10:33:01	2500	
2019-02-20	10:33:02	2500	
2019-02-20	10:33:03	2500	
2019-02-20	10:33:04	2500	
2019-02-20	10:33:05	2500	
2019-02-20	10:33:06	2500	
2019-02-20	10:33:07	2500	
2019-02-20	10:33:08	2500	
2019-02-20	10:33:09	2500	
2019-02-20	10:33:10	2500	
2019-02-20	10:33:11	2500	
2019-02-20	10:33:12	2500	
2019-02-20	10:33:13	2500	
2019-02-20	10:33:14	2500	
2019-02-20	10:33:15	2500	
2019-02-20	10:33:16	2500	
2019-02-20	10:33:17	2500	
2019-02-20	10:33:18	2500	
2019-02-20	10:33:19	2500	
2019-02-20	10:33:20	2500	
2019-02-20	10:33:21	2500	
2019-02-20	10:33:22	2500	
2019-02-20	10:33:23	2500	
2019-02-20	10:33:24	2500	
2019-02-20	10:33:25	2500	
2019-02-20	10:33:26	2500	
2019-02-20	10:33:27	2500	
2019-02-20	10:33:28	2500	
2019-02-20	10:33:29	2500	
2019-02-20	10:33:30	2500	
2019-02-20	10:33:31	2500	
2019-02-20	10:33:32	2500	
2019-02-20	10:33:33	2500	
2019-02-20	10:33:34	2500	
2019-02-20	10:33:35	2500	
2019-02-20	10:33:36	2500	
2019-02-20	10:33:37	2500	
2019-02-20	10:33:38	2500	
2019-02-20	10:33:39	2500	
2019-02-20	10:33:40	2500	
2019-02-20	10:33:41	2500	
2019-02-20	10:33:42	2500	
2019-02-20	10:33:43	2500	
2019-02-20	10:33:44	2500	
2019-02-20	10:33:45	2500	
2019-02-20	10:33:46	2500	
2019-02-20	10:33:47	2500	
2019-02-20	10:33:48	2500	
2019-02-20			



1 Table of Contents

2	EXECUTIVE SUMMARY	5
3	OVERVIEW	6
3.1	PRODUCT DESCRIPTION	6
3.2	THE DESIGN TASK	6
3.2.1	SUB-SYSTEM BREAKDOWN	6
3.2.2	INTERFACE DESIGN SPECIFICATIONS	7
4	DESIGN SPECIFICATION	8
4.1	DATA MANAGEMENT	8
4.2	USER INTERFACE	9
5	DESIGN ALTERNATIVES	10
5.1	ALTERNATIVE DESIGNS OR OPTIONS	10
5.1.1	DATA MANAGEMENT	10
5.1.1.1	Private Cloud	10
5.1.1.2	Public Cloud	11
5.1.1.3	Hybrid Cloud	12
5.1.1.4	Colocation	12
5.1.2	USER INTERFACE	13
5.1.2.1	Web Application	13
5.1.2.2	Native Application	14
5.1.2.3	Hybrid Application	14
5.2	ASSESSMENT OF ALTERNATIVES	15
5.2.1	PRIMARY CONSIDERATIONS	15
5.2.1.1	Data Management Solution	15
5.2.1.2	User Interface Platform	16
5.2.2	SECONDARY CONSIDERATIONS	17
5.2.2.1	Cloud Computing Service	17
5.3	DESIGN VALIDATION	19
5.4	TEST SPECIFICATION	21
5.4.1	DATA MANAGEMENT	21
5.4.2	USER INTERFACE	22
5.4.3	HEURISTIC TESTING	24
6	THE FINAL DESIGN	25
6.1	DESIGN SUMMARY	25
6.2	USER EXPERIENCE FLOWCHART	25
6.3	APPLICATION DEVELOPMENT LANGUAGES	29
6.3.1	HTML	29
6.3.2	CSS	29



6.3.3	JAVASCRIPT	29
6.3.4	PHP	29
6.3.5	SQL	29
6.4	RESPONSIVE WEB DESIGN	30
6.5	DATA STORAGE	30
6.5.1	DATABASE MANAGEMENT SYSTEM	31
6.5.2	FILE SYSTEM	31
6.6	SECURITY CONSIDERATIONS	32
6.6.1	SECURITY THREATS	32
6.6.1.1	Data Leaks	32
6.6.1.2	Cyberattacks	32
6.6.1.3	Lack of awareness	32
6.6.2	PROTECTION METHODOLOGY	32
6.6.2.1	Data and Network Encryption	32
6.6.2.2	Hashing and Salting	32
6.6.2.3	Access Control	32
6.6.2.4	Cloud Access Security Broker (CASB)	33
6.7	ACCESSIBILITY FEATURES	33
6.7.1	COLOUR COMBINATIONS WITH GOOD CONTRAST	33
6.7.2	SIMPLE LANGUAGE	33
6.7.3	KEYBOARD ONLY NAVIGATION	34
6.7.4	LARGE LINKS, BUTTONS AND TEXT	34
6.8	POTENTIAL FUTURE IMPROVEMENTS	34
6.8.1	WEBSITE GREETING PAGE	34
6.8.2	ROOM CALIBRATION	34
6.8.3	NATIVE APP DEVELOPMENT	35
6.8.4	PURCHASE SERVER INFRASTRUCTURE	35
7	REFERENCES	36
8	APPENDICES	38
8.1	.HTML FILES	38
8.1.1	LOGIN_PAGE.HTML	38
8.1.2	PASSWORD_RECOVERY.HTML	39
8.1.3	RESET_LOGIN.HTML	40
8.2	.PHP FILES	41
8.2.1	CONNECT.PHP	41
8.2.2	CAREHOME_ROOM_SELECT.PHP	42
8.2.3	GET_FALL_DATA.PHP	43
8.2.4	GET_RAW_DATA.PHP	44
8.2.5	GET_RESIDENT_INFO.PHP	45
8.2.6	GET_RESIDENTS.PHP	46
8.2.7	RESIDENT_FALL_INFO.PHP	47
8.2.8	RESIDENT_OVERVIEW.PHP	48
8.2.9	ROOM_SELECT.PHP	49
8.2.10	VERIFY_LOGIN.PHP	50

1.1 Table of Tables

TABLE 1 - SUB-SYSTEM ROLES	6
TABLE 2 - DATA MANAGEMENT DESIGN SPECIFICATIONS.....	8
TABLE 3 - USER INTERFACE DESIGN SPECIFICATIONS	9
TABLE 4 - RANKING CRITERIA WEIGHTINGS - DATA MANAGEMENT	15
TABLE 5 - WEIGHTED RANKINGS - DATA MANAGEMT	15
TABLE 6 - RANKING CRITERIA WEIGHTINGS - USER INTERFACE.....	16
TABLE 7 - WEIGHTED RANKINGS - USER INTERFACE	16
TABLE 8 - RANKING CRITERIA WEIGHTINGS – CLOUD COMPUTING SERVICE	18
TABLE 9 - WEIGHTED RANKINGS - CLOUD COMPUTING SERVICE	18
TABLE 10 - TEST SPECIFICATIONS - DATA MANAGEMENT	21
TABLE 11 - TEST SPECIFICATIONS - USER INTERFACE	23
TABLE 12 - SURVEY FOR COLLECTING USER FEEDBACK ON USER INTERFACE	24

1.2 Table of Figures

FIGURE 1 - SUB-SYSTEM BREAKDOWN.....	6
FIGURE 2 - SUB-SYSTEM INTERFACES BREAKDOWN	7
FIGURE 3 - CLOUD COMPUTING SERVICES.....	17
FIGURE 4 - FALL INFORMATION DESKTOP VIEW	19
FIGURE 5 - GAIT ANALYSIS DESKTOP VIEW	19
FIGURE 6 - MYSQL DATABASE CONTAINING SAMPLE RAW DATA	20
FIGURE 7 - FALL INFORMATION MOBILE VIEW	20
FIGURE 8 - GAIT ANALYSIS MOBILE VIEW.....	20
FIGURE 9 - 10 HEURISTIC PRINCIPLES FOR USER INTERFACE DESIGN	24
FIGURE 10 - INFRASTRUCTURE AS A SERVICE DIAGRAM	25
FIGURE 11 - USER EXPERIENCE FLOW CHART	26
FIGURE 12 - WEB APPLICATION PROTOTYPE ON MOBILE DISPLAY.....	27
FIGURE 13 - WEB APPLICATION PROTOTYPE ON DESKTOP DISPLAY.....	28
FIGURE 14 - HTML EXAMPLE CODE	29
FIGURE 15 - CSS CODE EXAMPLE.....	29
FIGURE 16 - JAVASCRIPT CODE EXAMPLE	29
FIGURE 17 - PHP CODE EXAMPLE.....	29
FIGURE 18 - SQL CODE EXAMPLE	30
FIGURE 19 - DESKTOP LOGIN PAGE.....	30
FIGURE 20 - MOBILE LOGIN PAGE.....	30
FIGURE 21 - DATABASE STRUCTURE.....	31
FIGURE 22 - FILE SYSTEM STRUCTURE.....	32
FIGURE 23 - HASHING AND SALTING PROCESS	32
FIGURE 24 - ACCESS LEVELS.....	33
FIGURE 25 - GOOD AND POOR COLOUR CONTRASTS EXAMPLE.....	33

2 Executive Summary

Salus Sense is a smart flooring product designed to be installed in health care environments for the purposes of fall detection, gait analysis and collection of walking pattern data.

This report aims to address the design of the data management and data presentation sub-systems. Although the sub-systems could be considered separately, they are highly integrated and therefore it is difficult to make a detailed design of one without considering the other. For this reason, the following report will present design challenges, alternatives, and solutions for both sub-systems.

As physicist Niels Bohr once remarked [1], “prediction is very difficult, especially about the future.” And this is particularly true when it comes to data storage requirements. Before a full-scale working prototype has been developed and testing commences it is difficult to estimate storage requirements. Therefore, it was decided that scalability is of paramount importance in the selection of a data storage solution.

- ❖ With potentially huge amounts of data requiring storage, security is a major design challenge. Not only is a company’s reputation on the line when it comes to protecting their user’s information but so too is its future. Data breaches can lead to mistrust between businesses and customers which would make finding new clients very challenging.
- ❖ Having selected a web application as the best choice for our user-interface there are more design challenges to take into consideration. Websites must be hosted by a server with high-speed internet connection which calls into question where the server space will come from and whether it is a worthy investment for the company to purchase servers as this time.
- ❖ With a largely elderly user base it is vital that the user interface is extremely accessible. This means including features to help those with physical, visual

and cognitive impairments such as screen-reader compatibility and an uncluttered user interface.

The final design manages to address all of the design challenges outlined above. Opting for cloud computing and, more specifically, Infrastructure as a Service the data storage and processing requirements become a smaller issue. With the ability to increase storage capacity and computing power as and when it is required, the business can feel safe in the knowledge that the front end will be unlikely to experience any bottlenecking or performance issues. Furthermore, using cloud computing means that the application may be hosted using the service providers web servers.

Although the prototype shown in this report does not demonstrate many security or accessibility features due to being in the early stages of development, this report clearly outlines the features to be included in future iterations of the design. Security features such as encryption, access control and the use of a Cloud Access Security Broker (CASB) [2] will mitigate the risk of data breaches and leaks immensely. Accessibility features such as good colour contrast options, keyboard only navigation and large, clear, well-spaced page elements [3] will ensure that even the least technically proficient user can navigate the web application with ease.

3 Overview

3.1 Product Description

Salus Sense is a fall detection and gait analysis product. What sets Salus Sense's smart floor solution apart from the competition, however, is the specially designed sensor layer sandwiched between a vinyl top layer and an acoustic underlay. Equipped with 400 piezo-electric sensors per m², the product is capable of tracking footsteps and detecting falls. By gathering this data over a prolonged period and analysing patterns the product can produce a wide range of information pertaining to the overall health of patients. This is known as Assisted Ambient Living (AAL) [4] in the healthcare industry.

3.2 The Design Task

3.2.1 Sub-System Breakdown

The whole system may be broken down into 7 individual sub-systems each of which presents its own unique design challenges. *Figure 1* shows the breakdown of all 7 sub-systems and *figure 2* describes the interfaces between each of them. By clearly defining the sub-systems and their interfaces, it is possible to split the system into individual sub-systems and assign each team member responsibility for the technical design of one sub-system. The aim is that each technical report should outline the technical considerations, alternative designs and specific information being passed from one sub-system to the next.

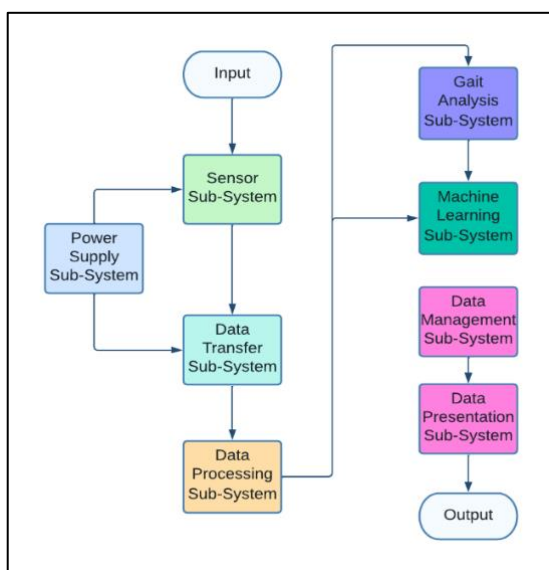


Figure 1 - Sub-System Breakdown

Sub-System	Group Member
Sensor Interfacing	Matthew Dickinson
Responsible for interfacing the microcontroller to read raw sensor output and provide the data to the multiplexer node for I ² C interface. Ensuring reliable connection with insulated wiring, filtering noise.	
Future Iterations: scale design into functional whole system prototype and increase security measures.	
Power Supply	Mackenzie Cox
Create a custom power solution to power all device components. Consider a backup battery in the event of a power outage. Include safety features to prevent overheating etc.	
Future Iterations: Ensure scalability of power solution for future prototypes.	
Data Transfer	Unassigned
The hardware design of the transceiver involves a microcontroller collating data from the sensor array to be encoded, organised, and relayed to the cloud. Data is transferred wirelessly using IoT methods.	
Future Iterations: Optimise the microcontroller for faster detection and relay operation while keeping design costs minimal.	
Data Processing	Declan Bridge
Software based subsystem where the data stored locally is filtered, key readings are isolated and passed through fall detection algorithm. Falls are immediately reported to carers.	
Future Iterations: Implementing machine learning locally.	
Gait Analysis	Keiran Ratnayake
Data pertaining to gait analysis is transferred to cloud storage where it is processed to determine useful gait statistics: movement pattern, stride length, speed etc.	
Future Iterations: By using machine learning, individuals unique gaits will be differentiated.	
Machine Learning	Zechi Nwankwo
Data from the gait analysis subsystem is used to run a deep learning algorithm to identify walking patterns of deteriorating health and determine fall likelihood. New data is collated, improving future functionality.	
Future Iterations: As the available learning data increases the fall likelihood algorithm should improve in accuracy.	
Data Presentation and Management	Ben Scriven
All local data is uploaded to secure non-local storage. Falls and gait analysis information is presented in an intuitive user-friendly format.	
Future Iterations: User Interface will include more graphics pertaining to health.	

Table 1 - Sub-System Roles

3.2.2 Interface Design Specifications

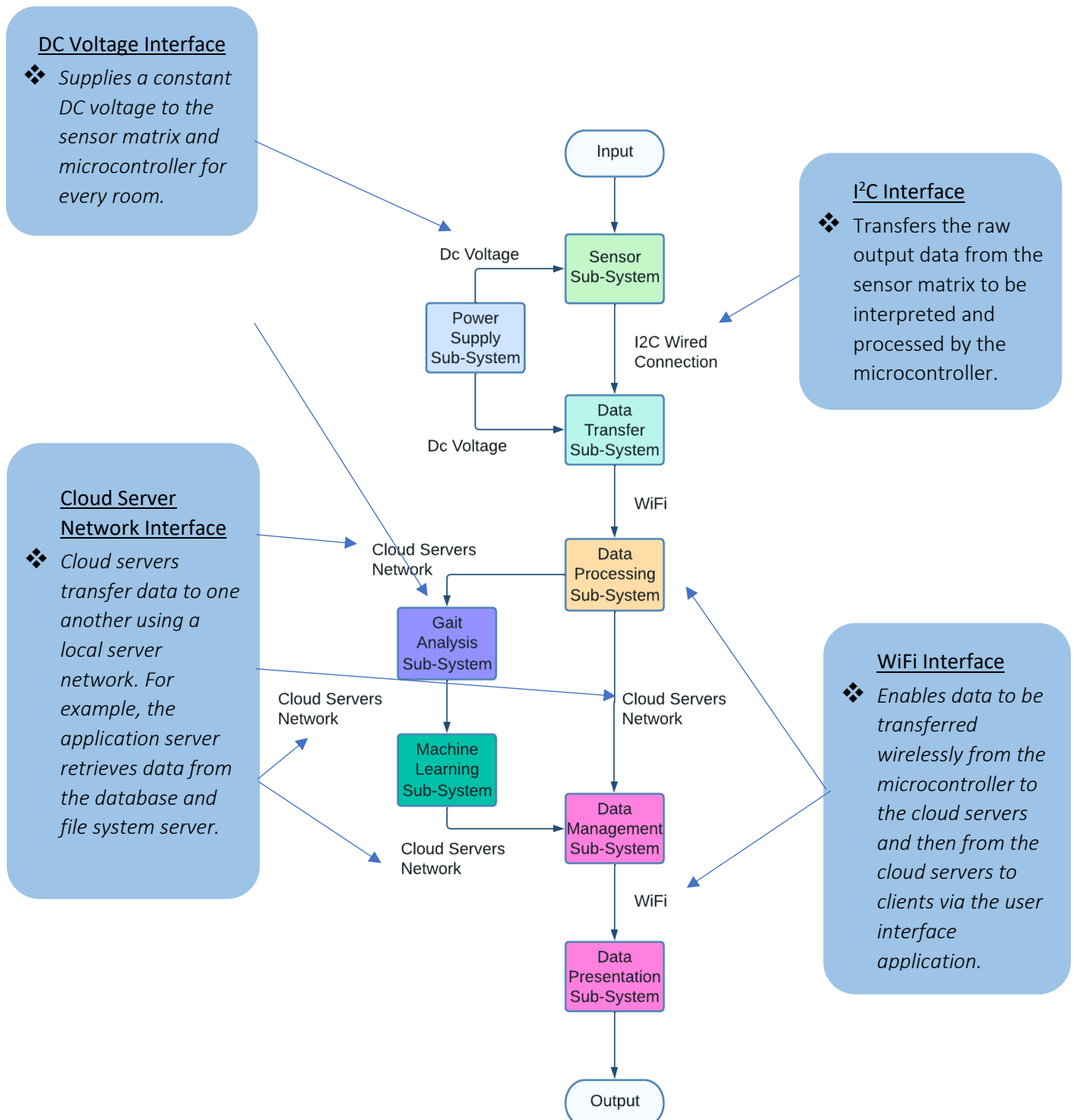


Figure 2 - Sub-System Interfaces Breakdown

4 Design Specification

4.1 Data Management

<i>Number</i>	<i>Design Specification</i>	<i>Description</i>
1	Storage capacity should be scalable	As the business expands the quantity of data to manage will increase exponentially and therefore it is essential that the selected management solution can be upgraded to provide greater capacity. Ideally the upgrades will be easily implemented and the costs minimal.
2	should support a range of data security features	There are many security features available including encryption, firewalls and access control. Beyond legal requirements, customers will feel more comfortable using our product if they can trust that their personal data is protected. Any breach to our storage could pose a huge threat to the future of the business as customer trust would be diminished.
3	Fast data transfer speeds	The user experience can be diminished massively when a user regularly experiences prolonged periods on loading screens [5] Therefore, it is key that our data management solution boasts impressive transmission speeds.
4	upfront costs should be as low as possible	As a start-up the business, we would initially invest as little as possible while still meeting all regulations and requirements. With this small initial investment, the business will purchase sufficient capacity for the first few clients with the intention of upgrading the capacity once it is required.
5	Minimal physical space required	Office space can be a huge expense for a new start-up, and it is therefore vital that the data management solution selected doesn't take up too much space thereby pushing up our monthly rent.
6	Data must be remotely accessible	As the user interface should be accessible from any internet connected device, it is key that all data is remotely accessible provided that the user device has an internet connection.

Table 2 - Data Management Design Specifications

4.2 User Interface

<i>Number</i>	<i>Design Specification</i>	<i>Description</i>
1	Operating system independent	The user interface will ideally be available to access from any internet connected device and it is therefore key that the application is not designed to work on only one operating system such as Windows. Building an application for Windows, Mac OS, Android and IOS can become extremely costly with all 4 platforms running on different operating systems.
2	Minimal setup	With a significant portion of care home residents lacking in technical ability and experience it is important that the user interface can be accessed with minimal setup. For example, navigating to a website, selecting the correct version of the app to download and then installing it correctly can be challenging for some elderly persons.
3	Minimal system requirements	As it is entirely possible that many of the residents will have computers that are over a decade old, it is vital the user interface is not reliant on new technology or fast processors. The app should be designed such that any heavy calculations or processes are executed remotely, leaving the users device with only the task of displaying results.
4	Login page with varying levels of admin access	To keep residents' personal data private, the application should have a login page which grants varying levels of access. The lowest level will be for residents who want to view their own personal data. The middle level will be available for care home staff to view statistics about all residents. The top level of admin access will be reserved for Salus Sense employees and will provide access to all data collected by our products across the UK.
5	System status indicator with sync status and last update time stamp	The UI should provide some type of indicator that lets the user know when the data they are viewing was last updated. Along with this, there should be a refresh button which enables the user to update the data to view the most recent measurements.
6	UI should provide a range of accessibility features	With a customer base largely consisting of elderly people, the UI should include accessibility features such as Alt text for images, options to increase font sizes and high contrast colours to improve readability.
7	Navigation made simple through icons	All navigation buttons should include a small image that represents its purpose. This makes navigating the app more intuitive and can drastically improve a user's perception of the product
8	Errors should be prevented or flagged and where this is not possible, the cause and solution should be presented to user clearly	As the app will be entirely new to all our customers it is very important that errors are caught before they lead to a fatal error in the program. Where an error is encountered it should be flagged to the user and provide a clear method for the user to resolve the issue.
9	Minimalist aesthetic to prevent confusion on the part of the user	A cluttered page is considerably more difficult to navigate for novice users. By keeping the screens appearance clean and uncluttered the user is less likely to feel overwhelmed by options and navigation will be more intuitive

Table 3 - User Interface Design Specifications

5 Design Alternatives

5.1 Alternative designs or Options

5.1.1 Data Management

5.1.1.1 Private Cloud

A private cloud, alternatively referred to as either an internal or enterprise cloud [6], is built and maintained for the exclusive use of a single organisation. Typically, the organisation is responsible for the operation of a private cloud as if it were a traditional data management centre. This includes maintenance, upgrades, operating system patches and application software management. A private cloud is generally hosted at an organisation's own data centre although it may also be hosted at a third-party colocation facility as described in *section 5.1.1.4*.

Pros	Cons
<ul style="list-style-type: none"> ❖ Security – Private cloud solutions are dedicated to a single organisation and therefore the hardware, data storage and connections are all designed to assure higher levels of security ❖ Performance – A private cloud resides within a company's intranet network meaning it is protected at all times by a firewall. It may provide access to the same resources as a public cloud but with less exposure to internet security risks. ❖ Full Control – Since the organisation owns the private cloud servers, they have complete control over security measures and reconfiguring software exactly the way they want it. This is particularly true for on premises servers. 	<ul style="list-style-type: none"> ❖ Cost - A private cloud is by far the most expensive solution because it requires specialised hardware and IT staff to ensure smooth operation. ❖ No External Support – As the private cloud is not maintained by a third-party provider but instead belongs entirely to the organisation, it is up to the company to install new security updates themselves and provide disaster recovery backup systems. These updates are cheaper from a public cloud provider as they are rolled out on a large scale.

5.1.1.2 Public Cloud

Public Clouds are server space rented from third-party provider such as Amazon or Microsoft. These providers are responsible for maintenance, security and backup provisions in the case of disasters such as fires or floods. They also configure software and roll out regular security updates to ensure the safety of the data on their servers.

Pros	Cons
<ul style="list-style-type: none"> ❖ Cheap – By renting server space from a third-party, an organisation may avoid the significant costs that come with purchasing server hardware and hiring IT staff. You also avoid incurred costs that come with software and security updates as well as server hardware maintenance. ❖ Scalability – An organisation may rent more server space from the provider as and when it is required. This provides an almost infinitely scalable data management solution. ❖ Backups – Public cloud providers also offer disaster recovery measures. All your data is stored and backed-up on an external server by the provider. 	<ul style="list-style-type: none"> ❖ Security – Generally public cloud solutions offer less security than a private cloud. However, public cloud providers do offer a range of security measures including password protection and encryption. Private clouds are only more secure provided that the initial investment in security measures is sufficiently large. Many small companies fail to achieve better security than public cloud providers. ❖ Lack of Control – Since the data is held by a third-party that the client does not control, it is not possible to customise your data storage setup which may prove problematic for large companies with complex storage requirements. ❖ Difficult to change provider – Once you have selected a provider and stored data with them it can prove difficult to transfer massive amounts of data to another public cloud provider securely. This means the organisation may have a cost-efficient deal to begin with but quickly become stuck with an expensive storage solution with no way of switching.

5.1.1.3 Hybrid Cloud

A hybrid cloud setup splits company's data between a private cloud infrastructure and a public cloud infrastructure [7]. A hybrid cloud puts your confidential and sensitive data on a highly secure private cloud, while everything else

Pros	Cons
<ul style="list-style-type: none"> ❖ Cheaper than private cloud – A hybrid cloud solution means that the organisation must only buy and maintain servers to store their most sensitive data. The rest of the data may be held in a more cost-efficient public cloud ❖ Control where it's needed – By maintaining one's own servers an organisation has complete control over the security of their most confidential data. They can also reconfigure software the way they want it on private servers, while also using public servers and getting automatic updates for less security sensitive software. 	<ul style="list-style-type: none"> ❖ More expensive than public cloud – The cost of setting up a private cloud, even a small one, is not insignificant which means it should only be considered as an option if security is a genuine concern. ❖ More maintenance than public cloud - Given that a hybrid cloud combines two solutions, using a hybrid cloud is a more complicated solution. Therefore, it is necessary to hire an IT team who can run and maintain the private servers. The IT team may be more compact than what is required for an entirely private cloud-based solution.

goes on the more economical public cloud.

5.1.1.4 Colocation

Colocation falls into the private cloud family. With a colocation service, firms are required to purchase their own servers, networking equipment, software and rack space, all of which reside in a data centre. With this option, clients have control over the brand and configuration of server hardware and are responsible for the installation, maintenance, software licensing and backups. The cost of retaining a colocation services provider includes power, climate control, storage and bandwidth [8].

Pros	Cons
<ul style="list-style-type: none"> ❖ Time and Cost Savings – When sharing a data centre with other businesses, the company may avoid spending on the construction of a custom build location. The shared facility will invest in the best cooling systems, security options and customised racks as well as uninterrupted power. All of these are significant costs for a business that chooses to invest in data centre construction ❖ Flexibility – As the business expands it will begin to need additional bandwidth. Colocation is an affordable choice as a business may simply upgrade their plan and enjoy business continuity. Upgrading a proprietary data centre can be an extremely costly endeavour. 	<ul style="list-style-type: none"> ❖ More costly than public cloud hosting – Compared to public cloud solutions, which provide all the infrastructure for a monthly subscription fee, colocation requires the business to purchase all its own server hardware and infrastructure for installation at a third-party data centre which can be a large expense for a new business. ❖ Maintenance Restrictions – As the server hardware is housed on separate facilities to the clients' offices, it can be time consuming and costly to carry out server maintenance as an IT employee must travel to the facility in order for repairs or updates to be carried out.

5.1.2 User Interface

5.1.2.1 Web Application

A web application may be described as a 'responsive' website. They can launch in a browser on a desktop computer, tablet, or mobile device. Web apps are designed to look good regardless of the device display and can work on any operating system. As a result, web apps are more cost effective and can be developed quickly [9].

Pros	Cons
<ul style="list-style-type: none"> ❖ Easy to maintain – Since web apps use standard website development tools, they are quick and easy to maintain and update. ❖ Fast and cost-effective development – Only one set of code is needed for all devices, so development costs are lower, and the development process takes less time and requires less specialised knowledge. ❖ Speed to market – Web apps can be built faster and, due to the fact that they do not require approval from app stores or marketplaces, they can be released to the market more quickly. 	<ul style="list-style-type: none"> ❖ Instability – Differences between and changes to browsers can cause issues with running a cross-platform web application. The regular updates needed to combat this can be quite costly particularly for a small business ❖ Internet connection required – As the codebase is stored on cloud servers, web apps cannot be accessed without a Wi-Fi or cellular data signal. ❖ Limited device feature accessibility – Since web apps are run from a browser, they do not have access to a device's native features such as the camera or GPS.

5.1.2.2 Native Application

Native applications are developed specifically for one platform or operating system. This means that they run faster and, if developed efficiently, deliver superior performance when compared to web applications. Native apps must be downloaded from an app store and are not accessible through a web browser [10].

Pros	Cons
<ul style="list-style-type: none"> ❖ Improved Usability – Since native apps are normally designed and developed for use on a single device, they are generally more intuitive, easy-to-use and provide the best user experience. ❖ Optimal Performance – Developing a native app provides developers with access to the full feature for the operating system which means the app run smoothly with fewer glitches. ❖ One codebase per platform – Since the developer is coding for only one platform during native app development, the process can be simpler and more streamlined. 	<ul style="list-style-type: none"> ❖ Specialisation required – Building a native app relies on specialist knowledge about app development for each platform e.g. IOS, Android, Windows, Mac OS, Linux etc. ❖ Separate development efforts – Each platform will require its own development effort. As a result, development time may be greater if the app is being developed more multiple platforms.

5.1.2.3 Hybrid Application

A hybrid app combines features from both a native app and a web app. It may use HTML, CSS, JavaScript and more modern languages such as Google's Flutter. It forms a single app that works on all platforms, including Apple and Android, and may be installed from an app store [10].

Pros	Cons
<ul style="list-style-type: none"> ❖ Fast Development – Developing a single codebase speeds up both the initial build time of a hybrid app, its maintenance, and its' up-keep. Cross-application development tools assist in enabling this process. ❖ Ease of Development – Developers don't need to learn multiple languages to create a hybrid app and thus the process is faster and easier for in-house developers ❖ Increased reach for less cost – Creating an app for both Android and IOS can provide access to a much wider audience. Although Androids market share dominates globally, in the US it is almost 50/50 and therefore creating an app for both platforms can almost double your audience. 	<ul style="list-style-type: none"> ❖ Poorer user experience – Usability is often not as good when compared to a native app. They can't take advantage of all the devices native features and are often slower. ❖ Customization challenges – The more customization an app requires, the more likely it is to also require native coding. This negates the development time and cost advantages of a hybrid app. ❖ New Feature Delays – Developers need to wait for cross-platform frameworks to develop compatibility for new features that are released natively.

5.2 Assessment of Alternatives

5.2.1 Primary Considerations

5.2.1.1 Data Management Solution

In order to determine the most appropriate data storage solution and user interface platform for Salus Sense the options were ranked on key considerations for the business. Each category upon which the options were ranked was assigned an importance weighting to reflect how important the assessment criteria is to the business.

For example, for the data storage solution *transmission speed* was given an importance weighting of 5 because the communication speeds required are modest whereas *scalability* was given a 10 because the storage capacity needed for the business is expected to increase exponentially and therefore scalability is a key consideration.

Ranking Criteria	Weighting	Evaluation	
		Low (0)	High (10)
Scalability	10	Can not increase size	Can increase size easily
Hardware Acquisition	8	Unaffordable	Free
Storage Capacity	6	No capacity	Unlimited capacity
Reliability	10	Never functions correctly	Never malfunctions
Maintenance	6	Impossible to maintain	Easy to maintain
Application Management	8	No control	Complete control
Physical space requirements	4	Enormous	No physical space required
Data Security	10	No security	extremely secure
Networking	10	No networking capability	Unlimited networking capability
Power and Cooling	7	Unaffordable	Free

Table 4 - Ranking Criteria Weightings - Data Management

Type	Scalability	Hardware Acquisition	Storage Capacity	Reliability	Maintenance	Application Management	Physical Space Requirements	Data Security	Networking	Power and Cooling	Total
Private Cloud	5	2	6	9	5	10	5	10	6	4	510
Public Cloud	10	10	10	6	8	5	10	6	8	8	624
Hybrid Cloud	8	6	8	8	7	8	8	8	7	6	586
Colocation	6	4	6	8	7	8	7	8	8	7	551

Table 5 - Weighted Rankings - Data Management

As *table 5* demonstrates, public cloud storage was chosen as the ideal data storage solution for Salus Sense at this point in time. This is due, in large part, to the scalability, capacity and physical space requirements. While the public cloud performed well in these categories, the alternatives performed poorly due to the massive costs that both a capacity increase and server room installation would incur.

5.2.1.2 User Interface Platform

Ranking Criteria	Weighting	Evaluation	
		Low	High
Setup Difficulty	8	Expertise required	No experience required
Cross-platform compatibility	8	Not compatible	Easily compatible
Native feature accessibility	3	No Accessibility	Unlimited Accessibility
Maintenance	7	No maintenance needed	Constant maintenance needed
Development costs	9	Expensive	Free
Stability	10	Never stable	always stable
Internet requirement	2	Needs constant connection	No connection required
Performance	7	Unusably slow	Instant

Table 6 - Ranking Criteria Weightings - User Interface

Type	Setup Difficulty	Cross-platform compatibility	Native feature accessibility	Maintenance	Development Costs	Stability	Internet Requirement	Performance	Total
Web App	10	10	5	8	8	6	0	6	405
Native App	6	2	10	6	4	10	10	8	348
Hybrid App	8	6	8	7	6	8	5	7	378

Table 7 - Weighted Rankings - User Interface

Table 7 demonstrates a web application was selected as the optimal solution for our business. While a web app boasts features like cross-platform compatibility and low and low development and maintenance costs, the alternatives offer limited or no cross-compatibility, and greater development costs.

Given the findings of the weighted ranking *tables 5 and 7*, Salus Sense will be moving forward with the development of a web application which will act as the user interface for clients to view their data. The raw data will be processed within the microcontroller of each room and the processed data will then be transferred to a shared storage location via the internet. This storage location will be a public cloud server which we Salus Sense will rent from a third-party provider.

5.2.2 Secondary Considerations

5.2.2.1 Cloud Computing Service

Having elected to move forward with the development of a web application using public cloud servers, the next consider is which cloud computing service best suits the needs of Salus Sense.

Figure 3 provide a brief description of the three cloud computing options available to companies and also some examples of popular products on the market for each option .

- **Infrastructure as a Service** – IaaS allows resources to be delivered to organisations virtually (or through the cloud). IaaS tools help organisation build and manage servers, networks, operating systems and data storage without the necessity to buy hardware.
- **Platform as a Service** – A PaaS provides developers with a framework they can use to build custom applications. PaaS doesn't deliver software but rather a platform for developers to create online software and apps.
- **Software as a Service** – SaaS refers to cloud-based software that is hosted by online by a third-party company. It is available for purchase on a subscription basis and is delivered to clients over the internet. *Table 8* shows key ranking considerations for Salus Sense when selecting a cloud computing service and also includes an associated importance weighting for each category [11].

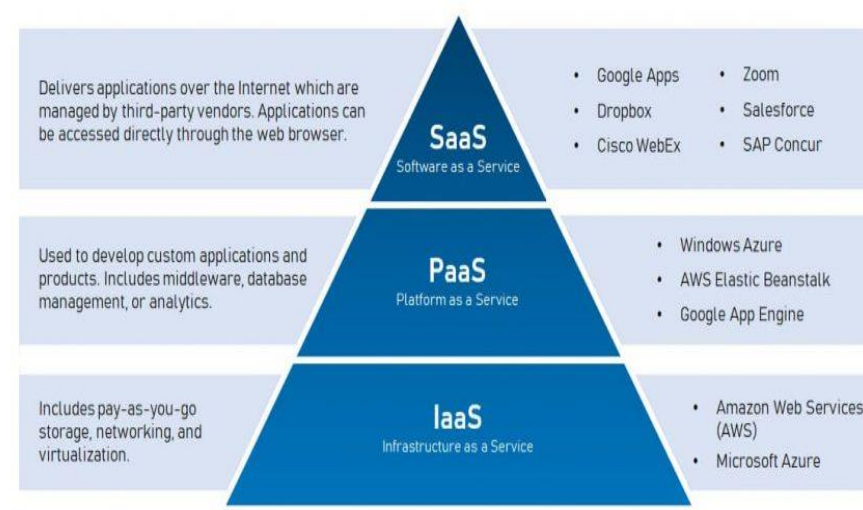


Figure 3 - Cloud Computing Services [15]

Ranking Criteria	Weighting	Evaluation	
		Low	High
Control	10	No Control	Complete Control
Scalability	10	Impossible to increase	Unlimited Increase
Included Security Features	8	No Security Features	Every Security Feature
Backup Systems	8	No Backup System	Unlimited Backups
Ease of use	7	Impossible to use	Easy to use
Collaboration Capabilities	6	No collaboration possible	Collaboration easy
Cross-Compatibility	8	No cross-compatibility	Compatible with anything

Table 8 - Ranking Criteria Weightings – Cloud Computing Service

Type	Control	Scalability	Included Security Features	Backup Systems	Ease of Use	Collaboration Capabilities	Cross-Compatibility	Total
IaaS	8	10	4	5	6	10	10	434
PaaS	5	10	7	7	7	8	8	423
SaaS	3	10	8	8	8	4	5	378

Table 9 - Weighted Rankings - Cloud Computing Service

As shown in *table 9*, the best cloud computing service for Salus Sense is Infrastructure as a Service. Although the PaaS and SaaS performed well in categories such as included security features and backup systems, IaaS outperformed the alternatives in more important areas like control and developer collaboration capabilities.

5.3 Design Validation

The web application, connected to our database through a public cloud server, ultimately will be available to access through any browser to all users. In order to make the web application available to the end-users, however, the company would have to invest in some form of web hosting service and a domain. The purchase/rental of these services can be expensive and for a small company just entering the product development stages there are better options for prototyping.

Instead of purchasing a cloud computing service and domain to test and develop the app, it is possible to do so on a personal computer (localhost). Through the use of a software stack, in this report 'WAMP' [12] was used, a developer can host a virtual server on their personal computer which allows testing of application features without consequences as it is localised and not connected to the web. 'WAMP' is so named as it consists of 4 key elements:

- 'W' for Windows as the software is designed to run on a windows OS, LAMP and MAMP are also available for developers using Linux or MacOS.
- 'A' for Apache. Apache is the server software that is responsible for serving web pages. When you request a page to be seen by you, Apache grants your request over HTTP and shows you the site.
- 'M' stands for MySQL. MySQL's job is to be the database management system for your server. It stores all the relevant information like your site's content, user profiles, etc.
- 'P' stands for either PHP, Perl or Python. These are all scripting languages they act, essentially, as the glue that holds the software stack together. It runs in conjunction with Apache while communicating with MySQL.

The combination of these pieces of software allows developers to develop, test and refine their web site/application and its database communication protocols without the need to pay for a cloud computing service and domain. Furthermore, it offers advantages for prototyping that speed up the development process further. Firstly, as it is localized to the developers PC, there is no need to wait for files to upload to a cloud

service before testing and secondly, files can be backed-up more easily.

Figures 4-8 demonstrate communication between a web application, the MySQL database management system and the local file system. The data used for generating the tables and graphics shown in figures 4-8 was generated randomly and placed inside the database shown in figure 6. Figures 4 and 5 demonstrate a preliminary layout for the user interface in desktop view while figures 7 and 8 demonstrate a preliminary user interface for mobile displays. To achieve automatically changing layouts depending on screen size, Responsive Web Design was implemented into the development of the application as discussed in section 6.3.

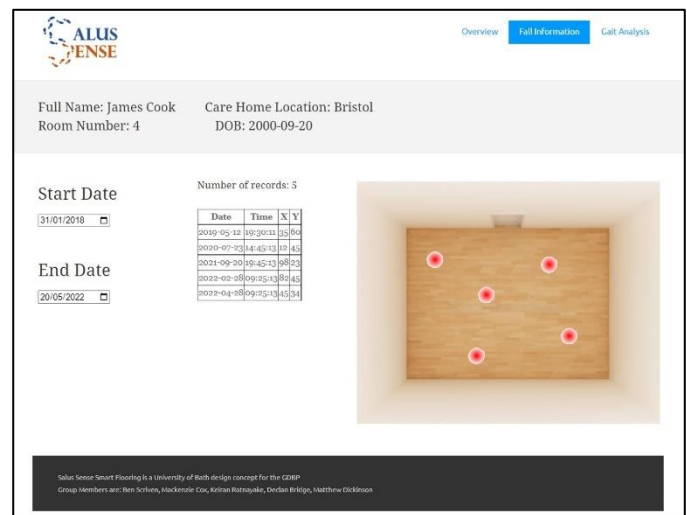


Figure 4 - Fall Information Desktop View

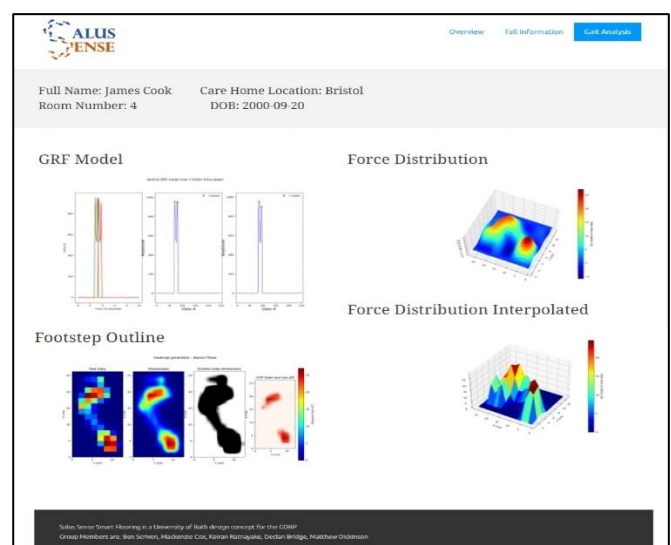
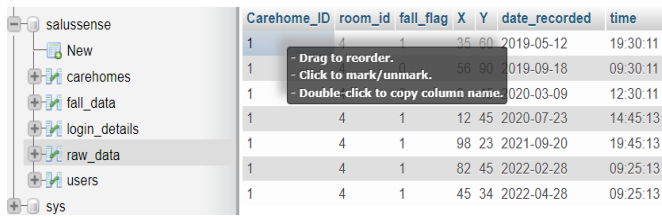
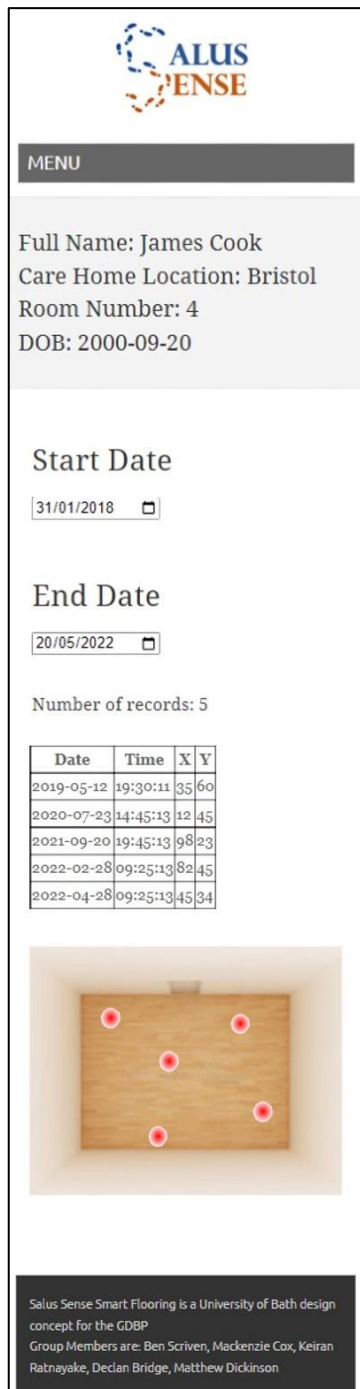


Figure 5 - Gait Analysis Desktop View



Carehome_ID	room_id	fall_flag	X	Y	date_recorded	time
1	4	1	35	60	2019-05-12	19:30:11
1	4	1	56	90	2019-09-18	09:30:11
1	4	1	12	45	2020-03-09	12:30:11
1	4	1	12	45	2020-07-23	14:45:13
1	4	1	98	23	2021-09-20	19:45:13
1	4	1	82	45	2022-02-28	09:25:13
1	4	1	45	34	2022-04-28	09:25:13

Figure 6 - MySQL Database containing sample raw data



ALUS ENSE

MENU

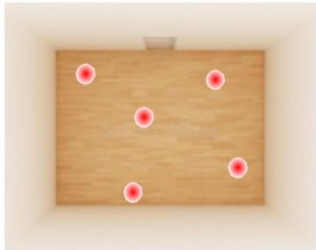
Full Name: James Cook
Care Home Location: Bristol
Room Number: 4
DOB: 2000-09-20

Start Date
31/01/2018

End Date
20/05/2022

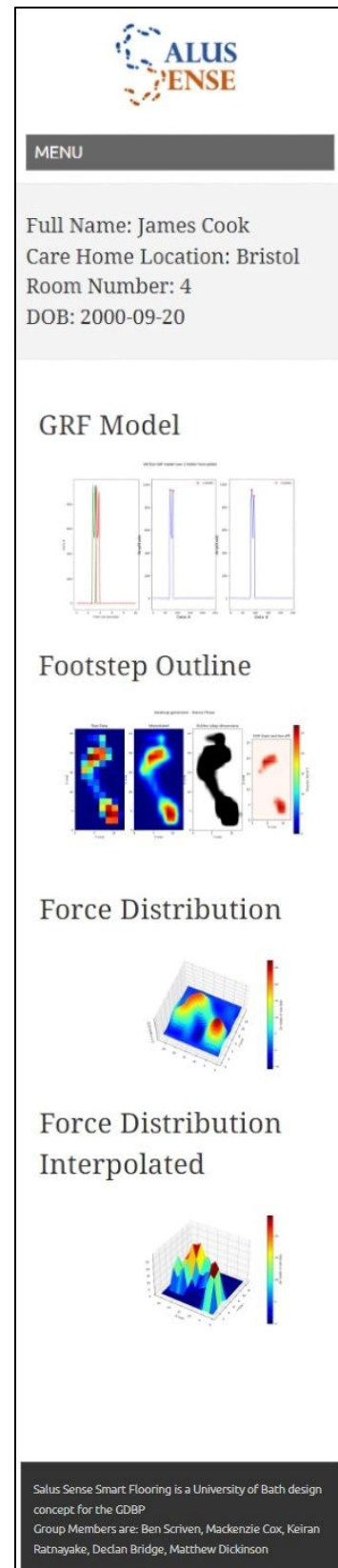
Number of records: 5

Date	Time	X	Y
2019-05-12	19:30:11	35	60
2020-07-23	14:45:13	12	45
2021-09-20	19:45:13	98	23
2022-02-28	09:25:13	82	45
2022-04-28	09:25:13	45	34



Salus Sense Smart Flooring is a University of Bath design concept for the GDBP
Group Members are: Ben Scriven, Mackenzie Cox, Keiran Ratnayake, Declan Bridge, Matthew Dickinson

Figure 7 - Fall Information Mobile View

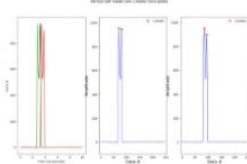


ALUS ENSE

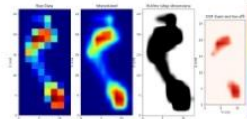
MENU

Full Name: James Cook
Care Home Location: Bristol
Room Number: 4
DOB: 2000-09-20

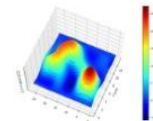
GRF Model



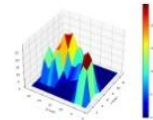
Footstep Outline



Force Distribution



Force Distribution Interpolated



Salus Sense Smart Flooring is a University of Bath design concept for the GDBP
Group Members are: Ben Scriven, Mackenzie Cox, Keiran Ratnayake, Declan Bridge, Matthew Dickinson

Figure 8 - Gait Analysis Mobile View

5.4 Test Specification

5.4.1 Data Management

Design Specification	Test Specification
Storage capacity should be scalable	This specification may be confirmed through online research. If a third-party data storage option is selected, a discussion with the provider will reveal specific costs for increasing capacity
Storage option should offer a range of data security features	Researching the storage option that is selected to see what security features are offered will reveal whether the option is suitable. The solution will be deemed to meet the design spec if it is compatible with the security options discussed in <i>section 6.5.2</i>
Fast data transfer speeds	This may be tested by establishing a full working prototype of the application and asking users to perform simple tasks. The survey discussed in section XXX will ask users how they felt about the loading times. The test will be successful if users generally have no issue with loading times
upfront costs should be as low as possible	By establishing a ranking table for the data management options and then weighting the criteria to reflect importance to the company we can ensure that upfront costs are low. This will be successful if the chosen data management option does indeed offer low setup costs
Minimal physical space required	By establishing a ranking table for the data management options and then weighting the criteria to reflect importance to the company we can ensure that physical space requirements are low. This will be successful if the chosen data management option does indeed require minimal physical space
Data must be remotely accessible	By establishing a ranking table for the data management options and then weighting the criteria to reflect importance to the company we can ensure that data is remotely accessible. This will be successful if the data can be accessed remotely.

Table 10 - Test Specifications - Data Management

5.4.2 User Interface

Design Specification	Test Specification
Operating system independent	The application will be accessed using a browser on various devices running different operating systems. The test will be deemed successful if the application opens correctly on all devices. This means all elements should be visible and usable.
Minimal setup	In order to test this we will ask users to open the application using a device they are completely unfamiliar with. This will test whether the setup process is complicated for users with minimal experience using technology. The results will be collected using the survey from <i>section 5.4.3</i>
Minimal system requirements	The application will be tested using a series of devices with limited processing power. This will include computers running outdated versions of operating systems and cheap mobile phones with slow processors. The test will be successful if the application runs without any glitches.
Login page with varying levels of admin access	Using a sample database of login details as shown in <i>section 5.4.3</i> the prototype of the web application can be tested. By creating various login-details of varying access levels the application can be used to confirm whether different pages are displayed. The test will be successful if different pages are displayed for different access levels.

System status indicator with sync status and last update time stamp	Once a prototype of the final product has been developed a system status indicator can be added to the web application prototype. This will be successful when a timestamp can be displayed to indicate when the flooring sensors last uploaded data to the database.
UI should provide a range of accessibility features	Using technology designed for additional needs users such as a screen reader, it is possible to test the web application is accessible. The test will be successful if all elements can be used and produce the same results that it would without a screen reader.
Navigation made simple through icons	This specification can be met by simply observing the user interface of the web application and confirming whether icons are used. As discussed in <i>section 5.4.3</i> , the best way of confirming this is to use a survey.
Errors should be prevented or flagged and where this is not possible, the cause and solution should be presented to user clearly	The web-application can be released to beta testers to test the functionality of all the features. The results can be collected using the survey discussed in <i>section 5.4.3</i> . The test will be successful if any and all errors are caught and a message explaining the solution is displayed.
Minimalist aesthetic to prevent confusion on the part of the user	This will be tested by asking users to rank how the aesthetic appears to them. The results will be collected using the survey in <i>section 5.4.3</i> . The test will be successful if most users think the appearance is easy to use.

Table 11 - Test Specifications - User Interface

5.4.3 Heuristic Testing

One of the many challenges that User Interface designers face before, during and after designing or improving a feature or page, is how to know if we're on the right path. While there are many methods of design testing, "Jakob Nielsen's 10 heuristics for user interface design" [6] are one of the must-does when designing, wireframing or evaluating a design. *Figure 9* shows the 10 heuristics as outlined by Nielsen.

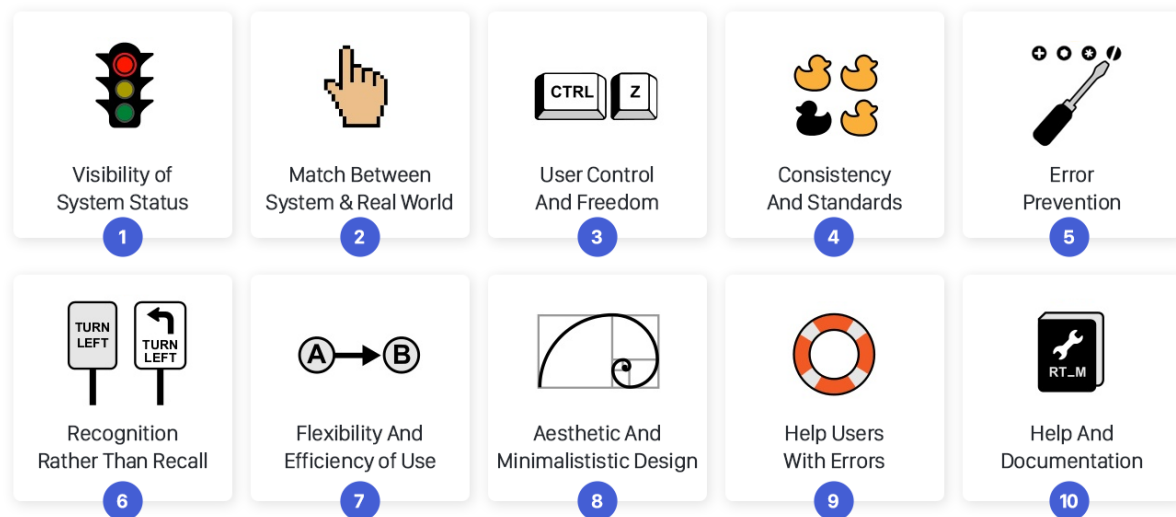


Figure 9 - 10 Heuristic Principles for User Interface Design

Once the design and development of the web application reaches the beta testing phase, a group of people will be asked to provide feedback. The group of testers should include a range of ages, disabilities and experience levels with technology. This should ensure that all issues can be reported back to the developer to be fixed in future iterations of the application. *Table 12* provides questions and answer formats for the survey that users will be asked to complete after testing.

Question Number	Question	Answer Format
1	How difficult or easy did you find the setup process?	Rate from 1 - 10
2	Did you experience any glitches or performance issues?	Yes/No + Explanation
3	Were you kept updated about the system sync status through an indicator light or timestamp?	Yes/No + Explanation
4	Do you use a screen-reader or another accessibility product?	Yes/No + Explanation
5	Did you find the UI to be accessible?	Yes/No + Explanation
6	Did all of the navigation buttons convey their purpose through an indicative icon?	Yes/No + Explanation
7	Did you experience any error messages when using the application?	Yes/No
7.1	If Yes to question 7: Did the error messages make it clear what you could do to rectify the issue?	Yes/No
7.2	If Yes to question 7: Did the suggested solution solve the problem?	Yes/No
8	Did you find the overall appearance appealing and easy to use?	Yes/No + Explanation

Table 12 - Survey for collecting user feedback on user interface

6 The Final Design

6.1 Design Summary

To summarise the results of the alternative design assessment, Salus Sense will be opting for a cloud computing solution and using a web application user interface in order to provide a platform for users to access and view their personal data. Beyond just selecting cloud computing, Infrastructure as a Service has been identified as the optimal solution for our data management requirements. With the option to increase bandwidth, computing power, storage capacity and many other options on demand, this model offers huge amounts of flexibility and means that the company may make considerable cost savings by avoiding investing in our own server infrastructure. *Figure 10* shows an outline of how an IaaS works. A web application was identified to be the best option as a platform for our user interface. Building our application as a web application means that it must only be designed, developed and tested once as the code is compatible across operating systems unlike the native alternatives. This single development process again saves the business spending money to develop the app for at least 5 operating systems (Windows, MacOS, Linux, Android, IOS).

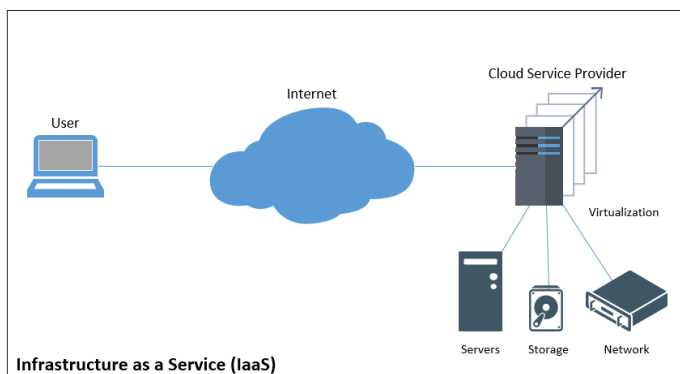


Figure 10 - Infrastructure as a Service Diagram

6.2 User Experience Flowchart

Figure 11 shows a flowchart of the web application user experience [7]. *Figures 12 and 13* show a preliminary design of the web application on a mobile display and a desktop display respectively. After arriving at the Salus Sense website users will be greeted with a login page.

This login page will include a password recovery option should any user misplace their login details.

After entering login details, the application will communicate with the database manager and determine 3 things:

- ❖ Are the login details correct?
- ❖ Is this the first time the user has logged in?
- ❖ What level of admin access does the user have?

If the user's login details are incorrect a pop-up box will appear to inform the user that they have entered their details incorrectly.

If the user is logging into the web application for the first time, they will be redirected to another page on which they will have the opportunity to update their username and password to whatever they choose.

Once the user has successfully logged in they will be greeted with one of three home pages depending on their level of administrator access.

The top-level access (reserved for Salus Sense employees) will present users with the option to select any room in any of the Salus Sense partner care homes.

The mid-level access (reserved for care home management staff) will present users with the option to select any room within the care home that they work at.

The lowest-level access will direct users straight to the room associated with their login details.

Once users have been directed to a specific room, they will be presented with an overview page which allows them to view records of all footsteps, falls or any other recorded impacts. This page will include 2 date input boxes so that the user may select a custom time interval of the records they would like to view. All records will be displayed in a user-friendly graphic.

The overview page will also contain a menu bar. The menu will allow users to navigate away from the overview page to either the fall information page or the gait analysis page.

The fall information page will present all falls recorded in a table along with location information and a timestamp of when the fall occurred. The fall location will be overlaid on a stock image of a room.

The gait analysis page will 4 images detailing the foot shape, force distribution GRF model during the step and two 3D force distribution graph.

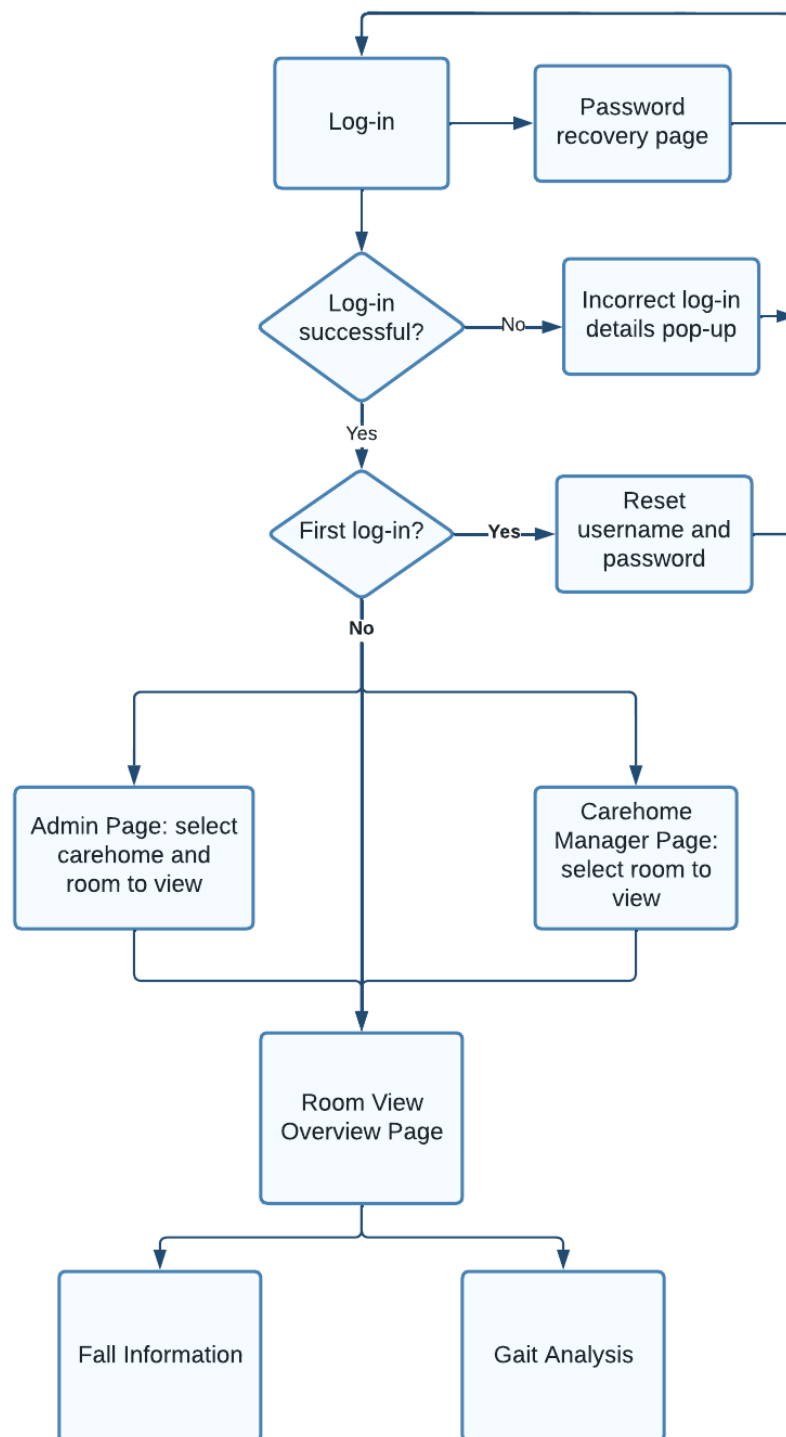


Figure 11 - User Experience Flow Chart

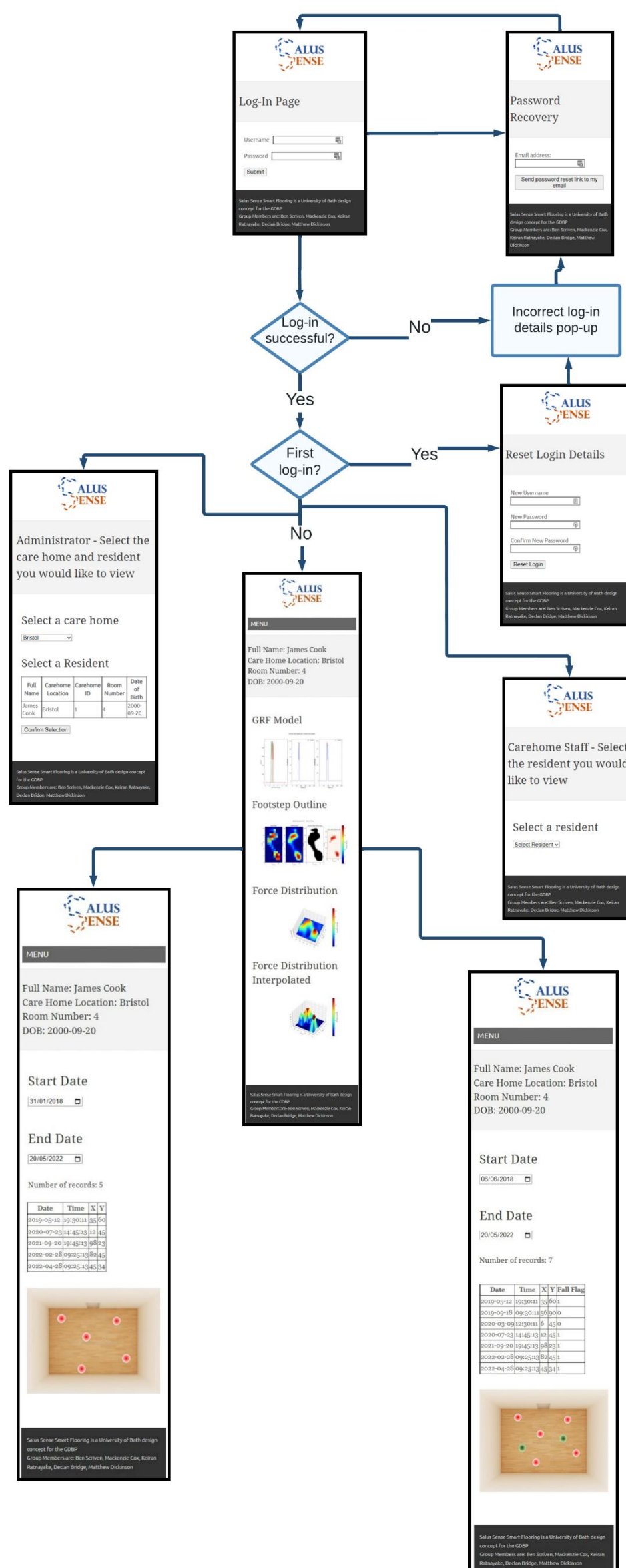


Figure 12 - Web Application Prototype on Mobile Display

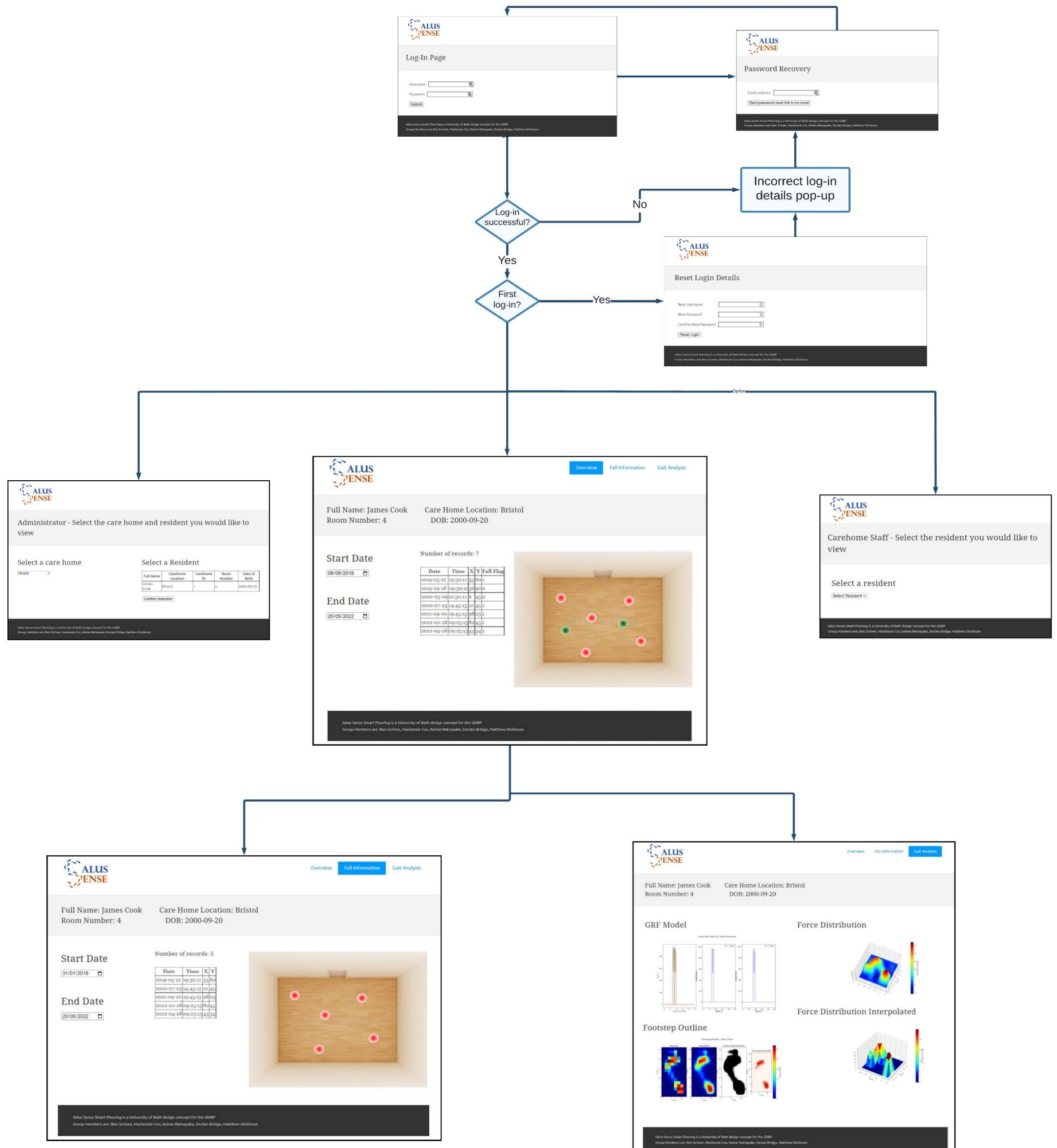


Figure 13 - Web Application Prototype on Desktop Display

6.3 Application Development Languages

6.3.1 HTML

HTML is the basis of all web development languages. For either a website or web app, HTML provides a solid foundation for website development. Since most websites get built on top of HTML, developers can style their code to fit users specific needs [15].

In the case of the prototype demonstrated in this report, HTML is responsible for the general layout of the web pages as well as the text, images and input boxes. *Figure 14* shows HTML used to display username and password input boxes in the web application prototype.

```

61      <label for="username">Username &nbsp;  </label>
62      <input type="text" id="username" name="username"/>
63      <br>
64      <br>
65      <label for="password">Password &nbsp;  </label>
66      <input type="text" id="password" name="password"/>

```

Figure 14 - HTML Example Code

6.3.2 CSS

CSS (Cascading Style Sheets) is a language that describes how to display HTML elements. When it comes to modern web design, CSS is vital; using CSS enables designers to separate style from content [15].

In the case of the prototype demonstrated in this report, CSS is responsible for the flexible containers (key to responsive web design) and the style of the text, images and input boxes. *Figure 15* shows CSS used to apply styles to the main body of the web pages.

```

18  body{
19      background:#fff;
20      color:#666;
21      font-family:'Droid Serif', Arial, Helvetica, sans-serif;
22      font-size:1em;
23      line-height:1.4em;
24      font-weight:normal;
25  }

```

Figure 15 - CSS Code Example

6.3.3 JavaScript

JavaScript is a lightweight, interpreted scripting language widely used to enhance web page content. It

is used in various ways such as adding simple functions or buttons to creating complex animations. JavaScript runs on top of a browser's built-in engine and adds functionality with simple commands placed inside `<script>` tags within an HTML document [15].

In the case of the prototype demonstrated in this report, JavaScript is responsible for the dynamic elements like updating drop down options or displaying new content based on user input. *Figure 16* shows JavaScript used to update the main overview page to display one specific residents' details.

```

127  <script>
128  var full_name = sessionStorage.getItem("full_name");
129  var care_home = sessionStorage.getItem("care_home");
130  var carehome_ID = sessionStorage.getItem("carehome_ID")
131  var room_num = sessionStorage.getItem("room_num");
132  var dob = sessionStorage.getItem("dob");
133

```

Figure 16 - JavaScript Code Example

6.3.4 PHP

Used widely by website developers, PHP (Hypertext Pre-processor) is one of the most popular server-side scripting languages. PHP allows web developers to create dynamic content and, importantly, interact with databases [15].

In the case of the prototype demonstrated in this report, PHP is responsible for sending SQL query statements to the database. *Figure 17* shows PHP used to connect to the MySQL database server and retrieve the username and password from the login form.

```

9      <?php
10      require_once 'connect.php';
11
12      $username = $_POST['username'];
13      $password = $_POST['password'];
14      // gets value sent from login form

```

Figure 17 - PHP Code Example

6.3.5 SQL

SQL (Structured Query Language) allows web developers to manage databases. From this language you can create, manipulate, remove or edit data contained within a database

In the case of the prototype demonstrated in this report, SQL is used to send instructions to the database [15]. This instruction may be in the form of queries which specify what data to retrieve from the database and how the data may be ordered. *Figure 18* shows an SQL query used to retrieve resident details based on the login details provided.

```
$sql = "SELECT * FROM login_details WHERE username LIKE '$username' AND password LIKE '$password'";
```

Figure 18 - SQL Code Example

6.4 Responsive Web Design

Responsive Web design (RWD) [8] is an approach to web design which proposes that web pages should respond to the user's behaviour and environment based on screen size, platform and orientation.

The practice consists of a mix of flexible grids and layouts, images and an intelligent use of CSS queries. As the user switches from their laptop to iPad, for example, the website should automatically switch layout to accommodate for resolution, image size and scripting abilities. *Figures 19 and 20* demonstrate a dynamic layout which automatically stretches/shrinks to fit the available display.



Figure 19 - Desktop Login Page

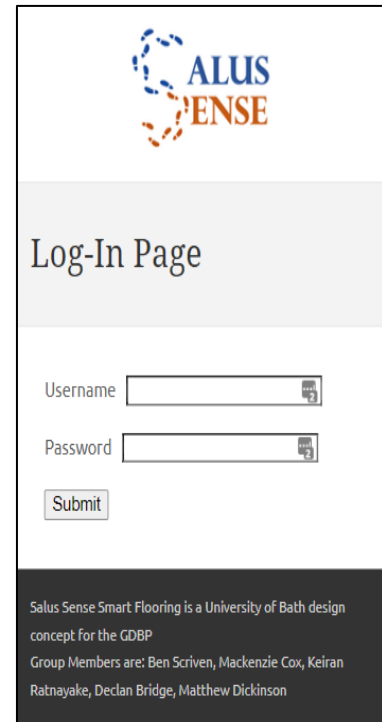


Figure 20 - Mobile Login Page

6.5 Data Storage

Given that the Salus Sense sensor product has a resolution of approximately 5cm², each footstep or fall that is recorded can exert a measurable force on anywhere between 5 and 1000 of the individual sensors. This level of detail is considerably greater than what is required for the user interface and therefore a considerable amount of data processing is carried out on the microcontroller before it is transferred to the cloud for storage.

For example, instead of recording the exact location of all 10 sensors that are triggered by a single footstep, the microcontroller processes the information and determines one single centre point for the footstep and transfers this for storage in the cloud instead. In doing so, the storage needed for this footstep instance is reduced by 10 times and, for falls where more than 1000 sensors are triggered, the storage savings can be even greater.

As part of the data processing, gait analysis is carried out on the raw sensor data before the excess sensor data is

discarded. As a result of the gait analysis 4 images are produced:

1. A GRF model will be produced and converted to an image file format. This may be a .JPG or .PNG and will be presented in the user interface without any further modification.
2. A foot outline image will be produced. This will either be a .JPG or .PNG file.
3. A 3D force distribution graph will be produced which clearly shows the outline of a footstep and where the force is exerted. Again, this will either be a .JPG or .PNG file.
4. An interpolated 3D force distribution graph will be produced which clearly shows the outline of a footstep and where the force is exerted. Again, this will either be a .JPG or .PNG file.

In terms of data storage options Salus Sense considered both a file system and a database management system [9]. A file system is a software application that organizes and maintains files on a storage device. It manages the storage and retrieval of data. A database management system, or DBMS, is a software application that allows you to access, create, and manage databases. While both options offer benefits, there are a few key differences that must be considered when selecting the appropriate choice for the business.

- ❖ Query Processing – While file systems offer no efficient method of processing queries, a database management service may be queried efficiently using a query language such as SQL (a popular choice).
- Crash Recovery - The file system doesn't have a crash recovery mechanism on the other hand, DBMS provides a crash recovery mechanism.
- ❖ Image storage – Images stored in a file system can be accessed very quickly by a web server whereas images stored in a database may take longer to download due to the increased complexity of communicating with the DBMS.

After considering both options, Salus Sense will be opting for a hybrid storage solution. Sensor information, fall records and user details etc. will be stored within a relational database whereas the files produced by the gait analysis will be stored using a file system. This

solution offers key advantages – sensor information and fall records may be queried efficiently through the use of a DBMS while image files stored in the file system may be called for display by the web application without a complex query.

6.5.1 Database Management System

Salus Sense will be using a DBMS called MySQL. MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL). *Figure 21* shows the database structure.



Figure 21 - Database Structure

6.5.2 File System

We will be using a hierarchical file system to store the files generated through gait analysis. A hierarchical file system is how drives, folders, files, and other storage devices are organized and displayed on an operating system. In a hierarchical file system, the drives, folders, and files are displayed in groups, which allows the user to see only the files they're interested in seeing. *Figure 22* shows how the file system would be structured.



Figure 22 - File System Structure

6.6 Security Considerations

6.6.1 Security Threats

6.6.1.1 Data Leaks

Cloud data faces the same threats as traditional infrastructures. Due to the large amount of data, cloud platforms are an attractive target for attackers. Data leaks often lead to a chain of unfortunate events for companies and IaaS providers. Leaking personal data about users can greatly diminish the trust between end-users and providers and should be avoided at all costs [18].

6.6.1.2 Cyberattacks

Targeted cyberattacks are a common sense and experienced attackers who have established a presence in their target infrastructure can be very difficult to detect. Despite Denial of Service (DOS) attacks having a long history, the rise of cloud computing has increased their commonality. A DOS attack can cause critical services to slow down or stop completely and, only through a deep understanding of the vulnerabilities presented by web servers, databases and applications, may they be prevented.

6.6.1.3 Lack of awareness

When establishing a new application on the cloud it is vital that the IT specialists are familiar with the features of cloud technologies and principles of deploying a cloud-based application. Without this knowledge, operational and architectural issues will arise that will lead to down time and, more seriously, to data breaches.

6.6.2 Protection Methodology

6.6.2.1 Data and Network Encryption

Encryption is the main and most popular method of data and network protection. Meticulously managing security and encryption key storage control is essential to maintaining the security benefits that come with encryption.

6.6.2.2 Hashing and Salting

Hashing is a one-way process that converts a password to ciphertext using what is known as a hash algorithm. A hashed password cannot be decrypted, but given a sufficiently large volume of hashed passwords, a hacker can use brute force computing to try and reverse engineer it. Password salting adds random characters before or after a password prior to hashing to obfuscate the actual password. Adding 'salt' makes it infinitely more difficult for hackers to decrypt passwords and is widely used in industry to protect user's login details [18]. Figure 23 shows the process of hashing, salting and storing passwords.

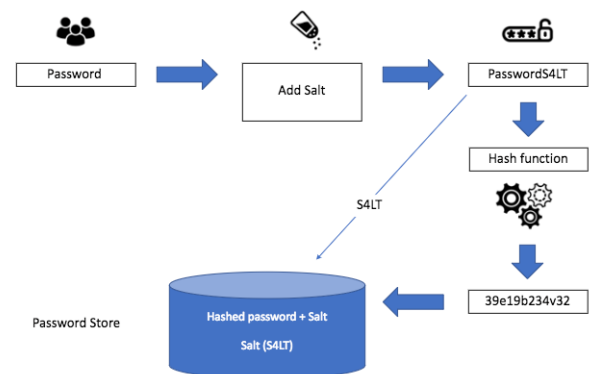


Figure 23 - Hashing and Salting Process

6.6.2.3 Access Control

Access control refers to the selective restriction of access to a place or other resource. In our case this means restricting access to administrator level features such as application source code, databases, and the network. The use of multi-factor authentication, including OTP, tokens, smart cards, etc., will significantly reduce the risks of unauthorized access to the infrastructure. Hardware and virtual firewalls provide a means of providing network access control.

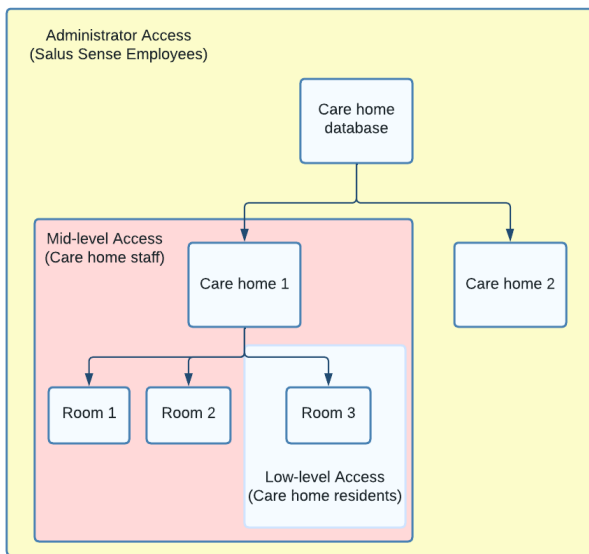


Figure 24 - Access Levels

6.6.2.4 Cloud Access Security Broker (CASB)

A CASB is a security tool which allows administrators to identify potential data loss risks and ensure a high level of protection. The solution works in conjunction with the IaaS provider's cloud infrastructure by enabling users to monitor shared files and prevent data leakage. This way, administrators know where important content is stored and who has access to the data [20].

6.7 Accessibility Features

The product produced by Salus Sense is intended for use in care homes which means a large proportion of users will be elderly. With visual, motor and cognitive impairment being extremely common among the elderly population it is vital that the web application is as accessible as possible. Furthermore, the elderly population are generally less experienced with technology and so the navigation of the web page should be as intuitive as possible to prevent any unnecessary confusion [21].

6.7.1 Colour Combinations with Good Contrast



Figure 25 - Good and Poor Colour Contrasts Example

It is important that the web application includes good colour contrast between elements including but not limited to text, backgrounds, links and buttons. Figure 25 shows an example of good colour contrast vs poor colour contrast.

This feature is essential for:

- ❖ People with low vision or a visual impairment
- ❖ People with colour deficiencies who struggle to distinguish between certain colours

This feature is useful for:

- ❖ Elderly people and anyone over the age of 50, as the colour contrast sensitivity in our eyes naturally declines with age.
- ❖ People with temporary disabilities like cataracts
- ❖ People in difficult lighting conditions e.g., experiencing glare on a mobile phone screen in bright sunlight

It is important to note that people with dyslexia or migraine sensitivity may prefer lower colour contrasts and, therefore, an option to change the colour contrasts is key so that they best suit the needs of the user.

6.7.2 Simple Language

The average reading of age in the UK is 9 and therefore writing in clear, plain English is essential for ensuring that the website content is accessible to as many people as possible.

This feature is essential for:

- ❖ People with learning difficulties who may be unable to understand complex language and vocabulary
- ❖ People with a cognitive impairment who struggle to focus on large blocks of text
- ❖ People with autism who make take sayings or phrases literally
- ❖ People with British Sign Language (BSL) as their first language
- ❖ People with dyslexia
- ❖ People with chronic fatigue

This feature is useful for:

- ❖ People who speak English, but it is not their first language
- ❖ People who are in a hurry

6.7.3 Keyboard Only Navigation

Ensuring that the web application may be navigated using a keyboard only is a fundamental accessibility principle. It affects anyone who uses assistive technology, including screen readers, to navigate web pages. Keyboard-only users typically navigate through web page elements like links, buttons and forms using the tab key.

This feature is essential for:

- ❖ People with motor, sensory or cognitive impairments who rely on a keyboard instead of a mouse to navigate web pages
- ❖ Blind and visually impaired users who rely on a screen reader

This feature is useful for:

- ❖ People with tremors
- ❖ People with temporary injuries such as a broken arm
- ❖ People who are multi-tasking
- ❖ Users who prefer keyboard shortcuts

6.7.4 Large Links, Buttons and Text

Elements on a web page should be sufficiently large that people with motor impairments can use them. Small buttons or elements placed close to one another are difficult for many people to use and it is therefore

important to space elements out as much as possible. This is an especially important feature for users with small displays such as mobile phones or tablets.

This feature is essential for:

- ❖ People with motor impairments including those with conditions like Cerebral Palsy, Parkinson's or Motor Neuron Disease

This feature is useful for:

- ❖ Blind and visually impaired users
- ❖ Elderly people
- ❖ People using mobile and tablet devices
- ❖ People that aren't confident with technology

6.8 Potential Future Improvements

6.8.1 Website Greeting Page

Currently the first page that users are greeted with when visiting the web application prototype is a login screen. Ideally in the future, the website would be expanded to include a much wider range of information about the business as a whole. Currently only user with login details can use the web app however the intention is that the Salus Sense web application will be expanded to include information such as an about as page, details of the product offered by Salus Sense, a description of the technology used in the product, pricing information for those interested in purchasing, customer reviews, an FAQ section and much more.

6.8.2 Room calibration

Currently the product is not designed to include a calibration process for each room which results in a default rectangle being used in the user interface to display results. Ideally in future software updates an option would be added to calibrate the shape of a room during installation so that the results shown in the user interface are more closely related to the real-life conditions. A potential method of doing this would be to add a calibration mode to the product which asks the user to walk around the available space multiple times. The product could then save this information for future reference and use the data gathered to generate more realistic graphics within the user interface.

6.8.3 Native App Development

As the business expands and operating costs increase, Salus Sense would consider investing in the development of a native app for IOS, Android, Windows and MacOS to start with.

Native apps offer better performance and the ability to run without an internet connection, both of which are currently drawbacks of the web application. Data could be stored locally in the native app meaning that users could download graphics and data to view offline at a later date.

6.8.4 Purchase server infrastructure

Although cloud computing services offer huge amounts of scalability, Salus Sense will consider investing in a server infrastructure in the future. Investing in servers would offers various benefits to the company including increased control over data security and software configuration. The investment in servers would also entail the hiring of a dedicated IT staff with experience in private cloud hosting, database management and web development.

7 References

- [N. Bohr, "Quotes of famous people," [Online].
1 Available: <https://quotepark.com/quotes/1736295-niels-bohr-prediction-is-very-difficult-especially-about-the/>. [Accessed 15 05 2022].
- [M. Sovetkin, "IaaS Security: Threats and Protection
2 Methodologies," [Online]. Available:
] <https://www.esecurityplanet.com/networks/iaas-security-threats-and-protection-methodologies/>. [Accessed 15 05 2022].
- [M. Dudharejia, "9 Ways You Can Make Your Website
3 More Accessible," [Online]. Available:
] <https://www.searchenginejournal.com/make-website-more-accessible/347450/#close>. [Accessed 13 05 2022].
- ["AMBIENT ASSISTED LIVING (AAL): TECHNOLOGY
4 FOR INDEPENDENT LIFE," globalRPH, [Online].
] Available: [https://globalrph.com/general-consumer-links/ambient-assisted-living-aal-technology-for-independent-life/#:~:text=Ambient%20Assisted%20Living%20\(AAL\)%20involves,make%20aging%20in%20place%20possible..](https://globalrph.com/general-consumer-links/ambient-assisted-living-aal-technology-for-independent-life/#:~:text=Ambient%20Assisted%20Living%20(AAL)%20involves,make%20aging%20in%20place%20possible..) [Accessed 04 05 2022].
- [M. Monaghan, "Website Load Time Statistics: Why
5 Speed Matters in 2022," [Online]. Available:
] <https://www.websitebuilderexpert.com/building-websites/website-load-time-statistics/>. [Accessed 2022 05 09].
- [V. Krasteva, "Private vs Public Cloud: Pros and Cons,"
6 [Online]. Available:
] <https://composity.com/post/private-vs-public-cloud-pros-and-cons>. [Accessed 14 05 2022].
- [TelecomsGuides, "The Pros and Cons of Public,
7 Private and Hybrid Cloud," [Online]. Available:
] <https://www.icompario.com/en-gb/telecoms/the-pros-and-cons-of-public-private-and-hybrid-cloud/>. [Accessed 05 05 2022].
- ["On-premises vs cloud vs colocation: is hybrid the
8 best?," TeleHouse, [Online]. Available:
] <https://www.telehouse.net/blog/on-premises-vs-cloud-vs-colocation/>. [Accessed 07 05 2022].
- [R. Sanyal, "Difference Between Web vs Hybrid vs
9 Native Apps," LambdaTest, [Online]. Available:
] <https://www.lambdatest.com/blog/web-vs-hybrid-vs-native-apps/#:~:text=Native%20applications,-Brief&text=Web%20apps%20are%20responsive%20websites%20that%20are%20dependent%20on%20browsers,HTML5%20and%20languages%20like%20Java..> [Accessed 15 05 2022].
- [K. Shields, "Native App vs. Web App vs. Hybrid App:
1 Which Is Best?," 04 01 2021. [Online]. Available:
0 [https://designli.co/blog/native-app-vs-web-app-vs-](https://designli.co/blog/native-app-vs-web-app-vs-hybrid-app-which-is-best/)
] [hybrid-app-which-is-best/](https://designli.co/blog/native-app-vs-web-app-vs-hybrid-app-which-is-best/). [Accessed 17 05 2022].
- [M. R. Stephen Watts, "SaaS vs PaaS vs IaaS: What's
1 The Difference & How To Choose," 15 06 2019.
1 [Online]. Available:
] <https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/>. [Accessed 16 05 2022].
- [E. Y., "What Is WAMP – a Friendly Guide for
1 Beginners," 18 03 2022. [Online]. Available:
2 [https://www.hostinger.co.uk/tutorials/what-is-](https://www.hostinger.co.uk/tutorials/what-is-wamp)
] [wamp](https://www.hostinger.co.uk/tutorials/what-is-wamp). [Accessed 03 05 2022].
- ["10 Things to pay attention in evaluating user
1 interface," CognitiveGroup, [Online]. Available:
3 [https://cognitivegroup.com/10-things-to-pay-](https://cognitivegroup.com/10-things-to-pay-attention-in-user-interface/)
] [attention-in-user-interface/](https://cognitivegroup.com/10-things-to-pay-attention-in-user-interface/). [Accessed 23 05 2022].
- [P. Boag, "Better UX Flow Charts: An Easy to Follow
1 How-to Guide in 6 Steps," 08 08 2021. [Online].
4 Available: [https://boagworld.com/how-to/ux-flow-](https://boagworld.com/how-to/ux-flow-charts/)
] [charts/](https://boagworld.com/how-to/ux-flow-charts/). [Accessed 05 05 2022].
- [[https://medium.com/javarevisited/top-5-](https://medium.com/javarevisited/top-5-programming-languages-for-web-development-in-2021-f6fd4f564eb6)
1 [programming-languages-for-web-development-in-](https://medium.com/javarevisited/top-5-programming-languages-for-web-development-in-2021-f6fd4f564eb6)
5 [2021-f6fd4f564eb6](https://medium.com/javarevisited/top-5-programming-languages-for-web-development-in-2021-f6fd4f564eb6), "Top 5 Programming languages
] for Web development in 2022," 13 01 2021.
[Online]. Available:
[https://medium.com/javarevisited/top-5-](https://medium.com/javarevisited/top-5-programming-languages-for-web-development-in-2021-f6fd4f564eb6)

programming-languages-for-web-development-in-2021-f6fd4f564eb6. [Accessed 22 05 2022].

[V. Friedman, "Responsive Web Design - What It Is And How To Use It," [Online]. Available: 6 <https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/>. [Accessed 20 05 2022].

["Difference between File System and DBMS," 26 04 1 2022. [Online]. Available: 7 <https://www.geeksforgeeks.org/difference-between-file-system-and-dbms/>. [Accessed 02 05 2022].

[C. Fanchi, "Database Security Best Practices," 10 12 1 2021. [Online]. Available: 8 <https://backendless.com/database-security-best-practices/>. [Accessed 14 05 2022].

[J. Jung, "What is Password Hashing and Salting?," 21 1 05 2021. [Online]. Available: 9 <https://www.okta.com/uk/blog/2019/03/what-are-salted-passwords-and-password-hashing/>. [Accessed 18 05 2022].

["IaaS Security: Threats and Protection 2 Methodologies," 16 03 2017. [Online]. Available: 0 <https://www.esecurityplanet.com/networks/iaas-security-threats-and-protection-methodologies/>. [Accessed 13 05 2022].

[M. Dudharejia, "9 Ways You Can Make Your Website 2 More Accessible," 01 08 2021. [Online]. Available: 1 [https://www.searchenginejournal.com/make-website-more-accessible/347450/#close](https://www.searchenginejournal.com/make-your-website-more-accessible/347450/#close). [Accessed 19 05 2022].

["SaaS vs IaaS vs PaaS: Which cloud service is right for 2 your business?," [Online]. Available: 2 <https://www.businesstechweekly.com/operational-efficiency/cloud-computing/saas-vs-paas-vs-iaas/>. [Accessed 21 05 2022].

8 Appendices

8.1 .HTML Files

8.1.1 Login_page.html

```
<!DOCTYPE html>

<html class="no-js">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

<title>Simple Responsive Template</title>
<meta name="description" content="Simple Responsive Template is a template for responsive web
design. Mobile first, responsive grid layout, toggle menu, navigation bar with unlimited drop downs,
responsive slideshow">
<meta name="keywords" content="">

<!-- Mobile viewport -->
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=yes">

<link rel="shortcut icon" href="images/salus_sense_icon.ico" type="image/x-icon">

<!-- CSS-->
<!-- Google web fonts. You can get your own bundle at http://www.google.com/fonts. Don't forget
to update the CSS accordingly!-->
<link href='http://fonts.googleapis.com/css?family=Droid+Serif|Ubuntu' rel='stylesheet'
type='text/css'>

<link rel="stylesheet" href="css/normalize.css">
<link rel="stylesheet" href="js/flexslider/flexslider.css">
<link rel="stylesheet" href="css/basic-style.css">

<!-- end CSS-->

<!-- JS-->
<script src="js/libs/modernizr-2.6.2.min.js"></script>
```

8.1.2 Password_recovery.html

```
<!DOCTYPE html>

<html class="no-js">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

<title>Simple Responsive Template</title>
<meta name="description" content="Simple Responsive Template is a template for responsive web
design. Mobile first, responsive grid layout, toggle menu, navigation bar with unlimited drop downs,
responsive slideshow">
<meta name="keywords" content="">

<!-- Mobile viewport -->
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=yes">

<link rel="shortcut icon" href="images/salus_sense_icon.ico" type="image/x-icon">

<!-- CSS-->

<!-- Google web fonts. You can get your own bundle at http://www.google.com/fonts. Don't forget
to update the CSS accordingly!-->
<link href='http://fonts.googleapis.com/css?family=Droid+Serif|Ubuntu' rel='stylesheet'
type='text/css'>

<link rel="stylesheet" href="css/normalize.css">
<link rel="stylesheet" href="js/flexslider/flexslider.css">
<link rel="stylesheet" href="css/basic-style.css">

<!-- end CSS-->

<!-- JS-->
<script src="js/libs/modernizr-2.6.2.min.js"></script>
```

8.1.3 Reset_login.html

```
<!DOCTYPE html>

<html class="no-js">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

<title>Simple Responsive Template</title>
<meta name="description" content="Simple Responsive Template is a template for responsive web
design. Mobile first, responsive grid layout, toggle menu, navigation bar with unlimited drop downs,
responsive slideshow">
<meta name="keywords" content="">

<!-- Mobile viewport -->
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=yes">

<link rel="shortcut icon" href="images/salus_sense_icon.ico" type="image/x-icon">

<!-- CSS-->
<!-- Google web fonts. You can get your own bundle at http://www.google.com/fonts. Don't forget
to update the CSS accordingly!-->
<link href='http://fonts.googleapis.com/css?family=Droid+Serif|Ubuntu' rel='stylesheet'
type='text/css'>

<link rel="stylesheet" href="css/normalize.css">
<link rel="stylesheet" href="js/flexslider/flexslider.css">
<link rel="stylesheet" href="css/basic-style.css">

<!-- end CSS-->

<!-- JS-->
<script src="js/libs/modernizr-2.6.2.min.js"></script>
```

8.2 .PHP Files

8.2.1 Connect.PHP

```
<?php
```

```
$user = 'root';
```

```
$password = 'User123';
```

```
$db = 'salussense';
```

```
$host = 'localhost';
```

```
$conn = new mysqli($host, $user, $password, $db) or die("Connect failed: %s\n". $conn -> error);
```

```
?>
```

8.2.2 Carehome_Room_Select.php

```
<!DOCTYPE html>

<html class="no-js">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

<title>Simple Responsive Template</title>
<meta name="description" content="Simple Responsive Template is a template for responsive web
design. Mobile first, responsive grid layout, toggle menu, navigation bar with unlimited drop downs,
responsive slideshow">
<meta name="keywords" content="">

<!-- Mobile viewport -->
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=yes">

<link rel="shortcut icon" href="images/salus_sense_icon.ico" type="image/x-icon">

<!-- CSS-->
<!-- Google web fonts. You can get your own bundle at http://www.google.com/fonts. Don't forget
to update the CSS accordingly!-->
<link href='http://fonts.googleapis.com/css?family=Droid+Serif|Ubuntu' rel='stylesheet'
type='text/css'>

<link rel="stylesheet" href="css/normalize.css">
<link rel="stylesheet" href="js/flexslider/flexslider.css">
<link rel="stylesheet" href="css/basic-style.css">

<!-- end CSS-->
```

8.2.3 Get_fall_data.php

```
<?php
$start_date = ($_GET['x']);
$end_date = ($_GET['y']);
$carehome_ID = ($_GET['z']);
$room_ID = ($_GET['q']);

require_once("connect.php");

$sql="SELECT * FROM raw_data WHERE date_recorded BETWEEN '$start_date' AND '$end_date'
AND Carehome_ID = '$carehome_ID' AND room_ID = '$room_ID' AND fall_flag = '1'";
$result = mysqli_query($conn,$sql);
$total_results=mysqli_num_rows($result);

if($total_results>0){

    //echo"success";

    ?>

    <h3>Number of records: <?php echo $total_results;?></h3>

    <br>

    <div id="raw_results" style="height:120px #ccc;font:16px/26px Georgia, Garamond, Serif;overflow-
y:auto;overflow-x:auto;overflow:auto">

    <table style=font-size:16px>

    <tbody>

    <tr>

    <th>Date</th>

    <th>Time</th>

    <th>X</th>

    <th>Y</th>

    </tr>

    <tr>

    <?php
```


8.2.4 Get_raw_data.php

```
<?php
$start_date = ($_GET['x']);
$end_date = ($_GET['y']);
$carehome_ID = ($_GET['z']);
$room_ID = ($_GET['q']);

require_once("connect.php");

$sql="SELECT * FROM raw_data WHERE date_recorded BETWEEN '$start_date' AND '$end_date'
AND Carehome_ID = '$carehome_ID' AND room_ID = '$room_ID'";

$result = mysqli_query($conn,$sql);

$total_results=mysqli_num_rows($result);

if($total_results>0){

    //echo"success";

    ?>

    <h3>Number of records: <?php echo $total_results;?></h3>

    <br>

    <div id="raw_results" style="height:120px #ccc;font:16px/26px Georgia, Garamond, Serif;overflow-
y:auto;overflow-x:auto;overflow:auto">

    <table style=font-size:16px>

    <tbody>

    <tr>

    <th>Date</th>

    <th>Time</th>

    <th>X</th>

    <th>Y</th>

    <th>Fall Flag</th>

    </tr>

    <tr>
```



8.2.5 Get_resident_info.php

```
<!DOCTYPE html>

<html class="no-js">

<?php
$full_name = ($_GET['x']);

$Carehome_Location = ($_GET['y']);

require_once("connect.php");

$sql="SELECT * FROM users WHERE full_name = '".$_full_name.'";
$result = mysqli_query($conn,$sql);

if(mysqli_num_rows($result)>0){

    echo "<table>";
    while($row = mysqli_fetch_array($result)) {

        ?>
        <tbody>
        <tr>
        <th>Full Name</th>
        <th>Carehome Location</th>
        <th>Carehome ID</th>
        <th>Room Number</th>
        <th>Date of Birth</th>
        </tr>
```

8.2.6 Get_residents.php

```
<?php
$q = ($_GET['q']);

require_once("connect.php");

//mysqli_select_db($conn,"salussense");
$sql="SELECT Carehome_ID FROM Carehomes WHERE Carehome_Location = '".$_q.'";
$result = mysqli_query($conn,$sql);

while($row = mysqli_fetch_array($result)) {
    $care_homeid = $row['Carehome_ID'];
}

$sql2="SELECT full_name FROM users WHERE Carehome_ID = '".$care_homeid.'";
$result2 = mysqli_query($conn,$sql2);

$resident_options = mysqli_fetch_all($result2,MYSQLI_ASSOC);

?>

<select name="Residents" onchange="UpdateRoomSelection()" id="Residents">
<option value="room selection">Select Resident</option>

<?php
foreach ($resident_options as $option) {
    ?>

    <option><?php echo $option['full_name']; ?> </option>
```

8.2.7 Resident_fall_info.php

```
<!DOCTYPE html>
```

```
<html class="no-js">
```

```
<head>
```

```
    <meta charset="utf-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
```

```
    <title>Simple Responsive Template</title>
```

```
    <meta name="description" content="Simple Responsive Template is a template for  
responsive web design. Mobile first, responsive grid layout, toggle menu, navigation bar with  
unlimited drop downs, responsive slideshow">
```

```
    <meta name="keywords" content="">
```

```
    <!-- Mobile viewport -->
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0, user-  
scalable=yes">
```

```
    <link rel="shortcut icon" href="images/salus_sense_icon.ico" type="image/x-icon">
```

```
    <!-- CSS-->
```

```
    <!-- Google web fonts. You can get your own bundle at http://www.google.com/fonts. Don't  
forget to update the CSS accordingly!-->
```

```
    <link href='http://fonts.googleapis.com/css?family=Droid+Serif|Ubuntu' rel='stylesheet'  
type='text/css'>
```

```
    <link rel="stylesheet" href="css/normalize.css">
```

```
    <link rel="stylesheet" href="js/flexslider/flexslider.css">
```

```
    <link rel="stylesheet" href="css/basic-style.css">
```

```
    <!-- end CSS-->
```

```
    <!-- JS-->
```

8.2.8 Resident_overview.php

```
<!DOCTYPE html>
```

```
<html class="no-js">
```

```
<head>
```

```
    <meta charset="utf-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
```

```
    <title>Simple Responsive Template</title>
```

```
    <meta name="description" content="Simple Responsive Template is a template for  
responsive web design. Mobile first, responsive grid layout, toggle menu, navigation bar with  
unlimited drop downs, responsive slideshow">
```

```
    <meta name="keywords" content="">
```

```
    <!-- Mobile viewport -->
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0, user-  
scalable=yes">
```

```
    <link rel="shortcut icon" href="images/salus_sense_icon.ico" type="image/x-icon">
```

```
    <!-- CSS-->
```

```
    <!-- Google web fonts. You can get your own bundle at http://www.google.com/fonts. Don't  
forget to update the CSS accordingly!-->
```

```
    <link href='http://fonts.googleapis.com/css?family=Droid+Serif|Ubuntu' rel='stylesheet'  
type='text/css'>
```

```
    <link rel="stylesheet" href="css/normalize.css">
```

```
    <link rel="stylesheet" href="js/flexslider/flexslider.css">
```

```
    <link rel="stylesheet" href="css/basic-style.css">
```

```
    <!-- end CSS-->
```

```
    <!-- JS-->
```

8.2.9 Room_select.php

```
<!DOCTYPE html>

<html class="no-js">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

<title>Simple Responsive Template</title>
<meta name="description" content="Simple Responsive Template is a template for responsive web
design. Mobile first, responsive grid layout, toggle menu, navigation bar with unlimited drop downs,
responsive slideshow">
<meta name="keywords" content="">

<!-- Mobile viewport -->
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=yes">

<link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon">

<!-- CSS-->

<!-- Google web fonts. You can get your own bundle at http://www.google.com/fonts. Don't forget
to update the CSS accordingly!-->
<link href='http://fonts.googleapis.com/css?family=Droid+Serif|Ubuntu' rel='stylesheet'
type='text/css'>

<link rel="stylesheet" href="css/normalize.css">
<link rel="stylesheet" href="js/flexslider/flexslider.css">
<link rel="stylesheet" href="css/basic-style.css">

<!-- end CSS-->
```

8.2.10 Verify_login.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

    <title>Search results</title>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

    <link rel="stylesheet" type="text/css" href="style.css"/>

</head>

<body>

    <?php

    require_once 'connect.php';

    $username = $_POST['username'];

    $password = $_POST['password'];

    // gets value sent from login form

    $username = htmlspecialchars($username);

    $password = htmlspecialchars($password);

    // changes characters used in html to their equivalents, for example: < to &lt;

    $username = mysqli_real_escape_string($conn, $username);

    $password = mysqli_real_escape_string($conn, $password);

    // makes sure nobody uses SQL injection

    $sql = "SELECT * FROM login_details WHERE username LIKE '$username' AND password LIKE
'$password'";

    $raw_results = mysqli_query($conn, $sql);

    $results_count = mysqli_num_rows($raw_results);

    // * means that it selects all fields, you can also write: `id`, `title`, `text`
```