University of Sheffield

Building a Visualisation Interface for Gait Analysis in People with Multiple Sclerosis

COM3610

Weng Wong (George) Hum

Supervisor: Vita Lanfranchi

A report submitted in partial fulfilment of the requirements for the degree of BSc in Computer Science by Weng Wong (George) Hum

in the

School of Computer Science

May 12, 2025

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Weng Wong (George) Hum

Signature: 

Date: May 12, 2025

Abstract

One of the main concerns of people with multiple sclerosis is the deterioration of their mobility, leading to overall poorer quality of life.

However, maintaining mobility comes with a need for ways to measure it. Measuring gait is often considered as a good representation of overall mobility, but many gold-standard procedures are lab-based (not representative of real-world walking) and expensive. The Mobilise-D consortium of experts are on the mission to clinically validate the use of wearable digital devices to track and extract digital parameters measuring every-day walking bouts.

While validation is in progress, this project aims to provide a useful tool for clinicians, allowing them to efficiently extract, visualise and interpret these digital parameters to aid gait analysis and mobility treatment in their patients with multiple sclerosis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Multiple sclerosis (MS) is an auto-immune condition that affects 2.9 million people across the world (National Multiple Sclerosis Society, no date). The body attacks its own nervous system and the body has a hard time sending messages throughout, causing basic functions such as vision and *mobility* to deteriorate.

Impaired mobility has detrimental effects to a person's overall quality of life, therefore the maintenance of mobility is prioritised in MS patients (Sutliff, 2009). To measure mobility, current gold-standard approaches are expensive in terms of both the equipment and time. They are also typically lab-based and therefore not representative of the variability in real-world walking.

The Mobilise-D consortium of experts aim to validate the use of wearable digital sensors to help understand and analyse gait in clinical settings. Parameters known as digital mobility outcomes (DMOs) are extracted from sensor data (specifically in this project, gait-related DMOs), and analysed to understand the patient's ability to walk. This is a representative measure of the patient's overall disability.

With most of this being in the validation stage, there is little work in the actual application of these DMOs. The main focus of this project is on applying these DMOs, by creating a visualisation tool for clinicians that shall help them form conclusions, from DMOs extracted from a single IMU sensor worn at the lower-back, about the gait quality of their MS patients. With this, the clinician can then make more informed decisions and provide better treatment.

1.2 Aims and Objectives

The main aim is to create a web visualisation interface for DMO data, for clinicians to easily make conclusions about their MS patients' gait. This involves allowing clinicians to easily upload patient walking data collected from a single wearable inertial measurement unit (IMU) sensor, worn at the lower-back. DMOs should then be automatically extracted and visualised in ways that aid clinician decisions surrounding treatment of an MS patient's gait and thus overall mobility. Features must meet real clinician needs and wants during gait analysis.

1.3 Overview of the Report

This report is structured as follows:

1. **Introduction:** some background about the project, aims and objectives and report structure.
2. **Literature survey:** summary of all preliminary research. Topics explore MS as a condition, DMOs and their validity, and visualisation tools and techniques.
3. **Requirements and analysis:** requirements (functional, non-functional and input data). Also, the iterative design process and results, analyses on target user, required tools, ethical considerations, potential risks and final evaluation method.
4. **Design:** exploration of data storage requirements, proposed application flow and visual design mock-ups.
5. **Implementation and Testing:** showcase of implemented features with explained code snippets. Testing methods (unit testing, manual testing and user testing) are also described.
6. **Results and Discussion:** success of the project in terms of meeting requirements and satisfying the target user, and suggestions for possible future work.

Chapter 2

Literature Survey

This chapter presents preliminary research. First, this chapter introduces what MS is and how it affects the gait of people with MS (pwMS). Then, the validity and extraction of DMOs are analysed, alongside identification of relevant DMOs. Finally, visualisation tools and techniques are explored to determine relevant features for the final system.

2.1 Multiple sclerosis

2.1.1 What is multiple sclerosis?

Multiple sclerosis is an autoimmune condition that affects the central nervous system, which consists of the brain and spinal cord. The immune system becomes self-destructive and attacks the protective cover of nerves, called myelin. Myelin aids in communication between neurons and, as a result, patients often experience a deterioration of basic functions such as sight and *movement*. (Cleveland Clinic, 2024)

2.1.2 Main types of multiple sclerosis

MS can be categorised into three different main types. These types are not indicative of condition severity, but rather how the patient's symptoms occur.

The first is relapsing remitting MS (RRMS). This is often the initial type that patients are diagnosed with. This is when patients experience symptoms in cycles of relapses (worsen) and remissions (get better).

The next type usually comes after RRMS - secondary progressive MS (SPMS). This is when

the patient doesn't experience cycles of relapsing and remitting symptoms anymore. The symptoms remain steady and gradually worsen.

Finally, the third main type is primary progressive MS (PPMS), where symptoms gradually worsen from the very start (MS Society, no date).

2.1.3 Effect of multiple sclerosis on gait

One of the main problems of MS is the deterioration of gait quality. Online surveys were conducted to evaluate the impact of gait impairment from the perspective of pwMS and their caretakers (Larocca, 2011). 41% of those with MS reported difficulty walking, of which 70% found that it was the most challenging part of having the disease. Consequently, maintaining mobility is one of the highest priorities for pwMS (Sutliff, 2009).

More specifically, studies commonly reported a decrease in gait speed, and step and stride length (see difference in figure 2.1). Some studies also reported decreases in swing period duration, increases in stance period percentage and increases in step width (see figure 2.1) (Coca-Tapia et al., 2021). These are all examples of gait-related DMOs, which will be investigated further in the sections below.

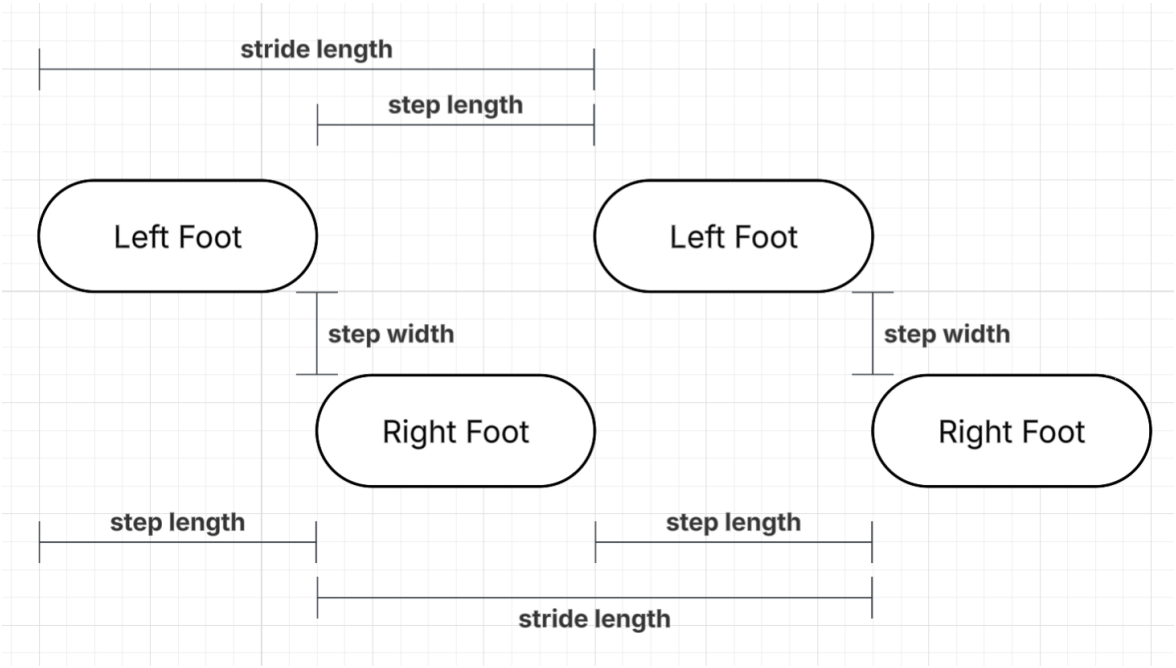


Figure 2.1: Illustration of step width, step length and stride length

2.2 Digital mobility outcomes

2.2.1 Validity in assessing multiple sclerosis

There is ongoing work in validating the use of DMOs in monitoring diseases that impair gait, including MS. The main advantage of DMOs is that they can be extracted from simple, wearable sensors. This makes gait assessment less expensive (than e.g. complex camera systems) and the hope is that results can be more representative of real-world walking as data can be collected while patients go about their daily lives. Thus, successful validation of DMOs will greatly advance treatments of gait-impairing diseases like MS.

A longitudinal study of 600 pwMS (Brittain et al., 2023) found that gait-related DMOs were able to differentiate between the different types of MS. People with PPMS and SPMS, compared to those with RRMS, had shorter walking bouts (WBs), fewer turns, slower gait speeds, shorter stride lengths and step durations.

To add, there is an older scoping review (Polhemus et al., 2021) over existing studies that sought to assess the clinical utility of DMOs, across different conditions (including MS). For MS, this review found several DMOs were able to distinguish between different, known levels of gait. There was also good association between many DMOs and other validated measures of disease severity. However, there was limited evidence for the predictive validity, ecological validity, and responsiveness of DMOs.

At the time of writing, the official Mobilise-D clinical validation study (Mikolaizak et al., 2022) is still ongoing. It is a longitudinal study of 2400 participants that is looking to solidify the clinical utility of DMOs in measuring and monitoring impaired mobility.

There is clearly still work to be done on validating the use of DMOs in real-world, clinical settings. This project can therefore be viewed as 'preparation' for when DMOs are fully validated - a tool that is readily available for clinicians to easily make sense of these DMOs and provide more informed treatments for mobility in pwMS.

2.2.2 Core algorithms for extraction from inertial measurement unit data

Patients can wear an inertial measurement unit (IMU) sensor and continue living as normal, as data is continuously collected. However, IMU data itself is meaningless. DMOs, such as gait speed, must be extracted from this IMU data using specific algorithms.

Technical validation of algorithms that extract gait-related DMOs from single IMU sensors worn at the lower back has been carried out by Mobilise-D (Micó-Amigo et al., 2023). This study evaluated different algorithms for estimating gait sequence detection (GSD) and foot

initial contact detection (ICD). These are imperative for identifying periods of walking and steps taken. These are prerequisites for extracting any kind of DMO. The study also evaluates estimation algorithms for cadence (CAD) and stride length (SL), which are DMOs. These can be interpreted as the "main" algorithms for extracting DMOs that will be used to derive other DMOs. The top performing algorithms in MS for GSD, ICD and CAD had relatively high overall performance scores, while the top performing algorithm for SL remained on the lower side. This highlights that the algorithms on which this project will be built upon, will be generally reliable, though for the SL algorithm comparatively less so. In particular, this hints that temporal DMOs (e.g. CAD) may be more reliable than spatial DMOs (e.g. SL) displayed in the final system. The study mentions that the poor performance for SL could be due to turns and non-straight walking patterns of real-world walking. However, this is unjustified, since the main aim of DMOs is to better represent real-world mobility. Nonetheless, more research on SL algorithms is required for the final system of this project to present better estimates of SL.

Referencing the same study, it is useful to consider that the algorithms for CAD, ICD and SL performed generally worse for shorter duration WBs and slower gait speeds. Higher errors (greater than 50%) in the study were observed in durations and speeds specified in table 2.1. The study observed that algorithms performed significantly worse when the subject's walking speed was below 0.5 m/s. This suggests that this project's final system would be most suitable for longer and higher speed gait assessments than values shown in table 2.1. This may be unsuitable for patients with more severe disability. However, this is crucial to consider when fairly evaluating the results displayed by the final system. This highlights the need for more research to create algorithms that avoid this problem, to make results displayed by this project's final system more reliable.

DMO	**Walking bout duration**	**Gait speed**
Step duration (from ICD)	8.37 ± 4.71s	0.44 ± 0.24m/s
CAD	8.88 ± 5.97s	0.28 ± 0.09m/s
SL	13.03 ± 10.53s	0.36 ± 0.13m/s

Table 2.1: Means ± standard deviations of walking bout durations and gait speeds that had relative errors greater than 50% (compared with values from the reference system) for each DMO. (Micó-Amigo et al., 2023)

2.2.3 Gait speed estimation and Mobilise-D algorithm pipelines

The Mobilise-D algorithm pipelines were created from results of the study analysed in subsection 2.2.2. There are two different recommended pipelines, constructed from the top-performing algorithms for each gait-impairing condition. Thus, there is a recommended pipeline specifically for MS.

A study was conducted on the validation of gait speed estimation with the Mobilise-D algorithm pipelines (Kirk et al., 2024). The study found overall positive results. As the study describes, gait speed is a "composite measure", because it is derived from final results from the pipeline. This study therefore supports the reliability of the overall Mobilise-D algorithm pipelines in extracting accurate DMOs. However, it was also discussed that some results were negatively impacted by slow WBs (as seen in sub-section 2.2.2), which further emphasises the need for more research to improve algorithms for patients with severely impaired gait.

2.2.4 Implications on the validity of data displayed on this project

Although these research findings are not completely positive for the project, expectations can be set on the overall usefulness and validity of the final system. Until DMOs are completely validated and the gaps in the performances of current algorithms are addressed, the validity of the results visualised in the final system can be challenged.

2.2.5 Relevant digital mobility outcomes for MS

DMOs that best highlight mobility impairments and disease severity in MS should be identified. There are many DMOs that can be extracted. Shah et al. (2020) verifies that DMOs discriminate mobility impairments differently across different conditions, so to better analyse gait in pwMS, only a select group of DMOs should be focused on.

Shah et al. (2020) tested which DMOs best discriminated between pwMS and a healthy control group. Results can be found in figure 2.2. Nine DMOs with Area under Curves (AUC) scores greater or equal to 0.80 are considered to have good discriminatory ability and the final implementation should prioritise these DMOs.

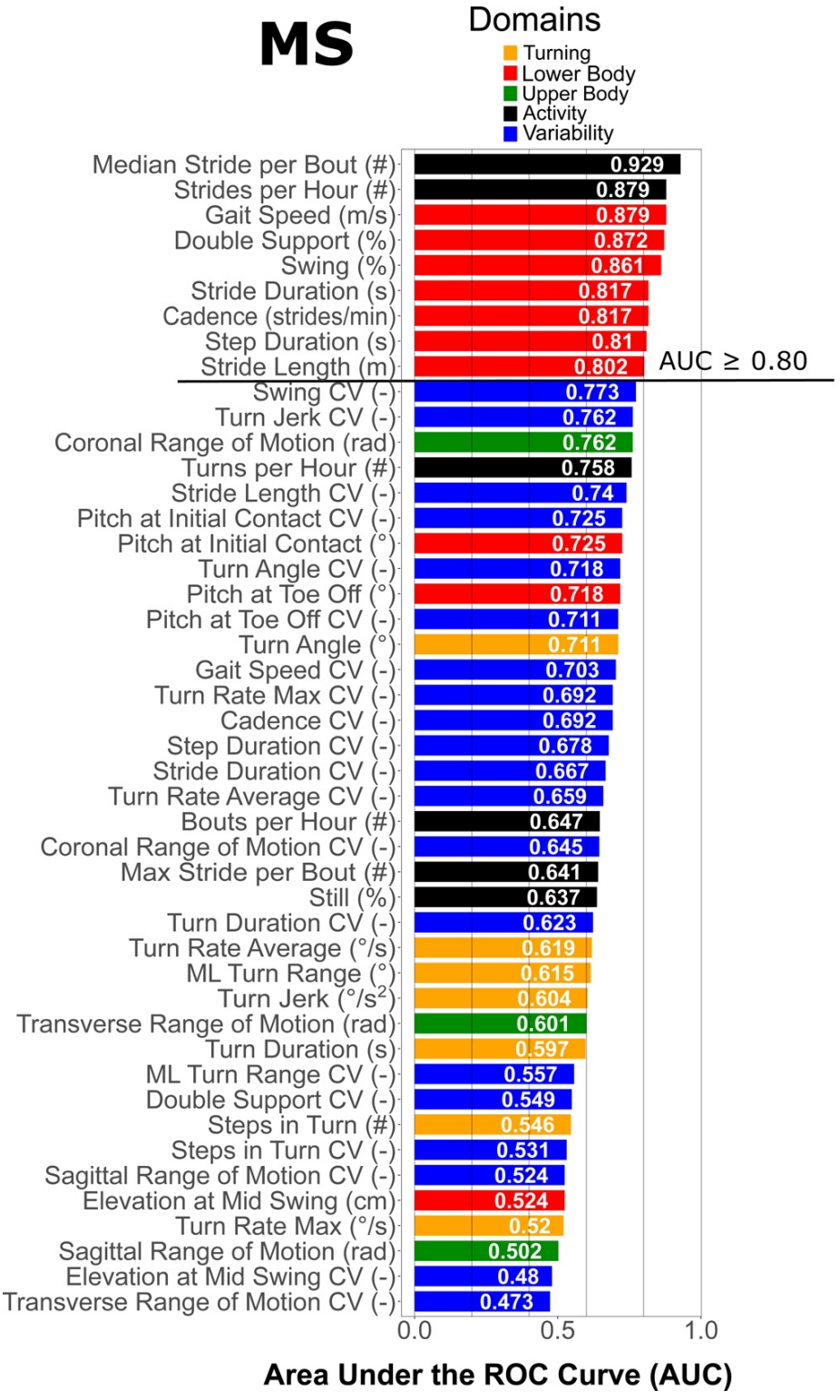


Figure 2.2: Discriminatory ability of DMOs between pwMS and a healthy control group. Higher AUC value indicates better discrimination. (Shah et al., 2020)

The same study investigated correlations between these discriminative DMOs and MS severity (represented by existing clinical scores/measures). Significant correlations were found between the Patient-Reported Expanded Disability Status Scale and double support percentage, swing percentage, pitch at initial contact and pitch at toe off CV (representing the variability of pitch at toe off). Although only close to statistical significance, there were also correlations between the median strides per bout and The Modified Fatigue Index Scale. Finally, there were also correlations between gait speed and the Timed 25-Foot Walk test. Therefore, these DMOs should be noted, as there is some evidence of being reflective of MS severity.

2.3 Visualisation

2.3.1 Data types involved and visualisation requirements

To establish how to best visualise data, understanding the type of data concerned is crucial. The data involved in this project is multivariate and heterogenous. There are many numerical gait parameters (DMOs) that can be extracted and analysed together. DMOs can also be defined on many levels - aggregated metrics (e.g. mean) across all identified walking bouts (WB), per WB or per stride of a given WB. Additionally, these are associated with patients of different MS category types and backgrounds. This project therefore requires visualisation techniques for analysing multivariate and heterogenous data. Furthermore, techniques focusing on the progression of parameter values over time will be useful. Sub-sections below will uncover useful data visualisation techniques, beyond simple approaches such as a bar chart or line graph.

2.3.2 Parallel coordinate plots

Parallel coordinate plots (PCP) are useful for analysing multiple numerical variables. Each variable has its own axis with different or normalised scales. Each record would have values for each variable, with points plotted on respective axes and connected to create a polyline across all axes. The axes order can be shifted to bring desired variables closer for analysis and specific polylines can be highlighted to focus on specific records (called brushing). Correlations between variables can then be identified depending collective positioning of polylines (The Data Visualisation Catalogue, no date-a).

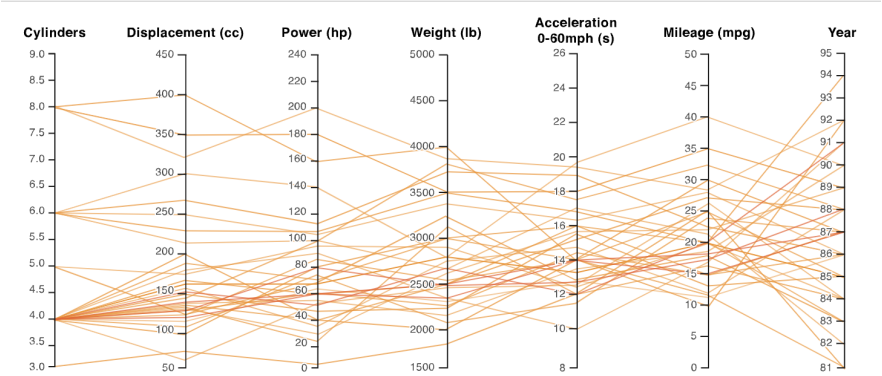


Figure 2.3: PCP for vehicle stats (The Data Visualisation Catalogue, no date-a)

This will be useful in finding correlations between multiple DMOs at once, or comparing DMO values for different selected strides/walking bouts.

2.3.3 Parallel set plots

Parallel set plots (PSPs) are useful for analysing multiple categorical variables (e.g. gender). An axis is divided into separate parts for each sub-category (e.g. male, female) with widths proportional to their prominence within the respective category in the dataset. Each "ribbon" (coloured shapes in figure 2.4) is connected across all dimensions, representing the proportion of records that are in each category. As one analyses further down the plot, percentage intersections between the categories are revealed (The Data Visualisation Catalogue, no date-b).

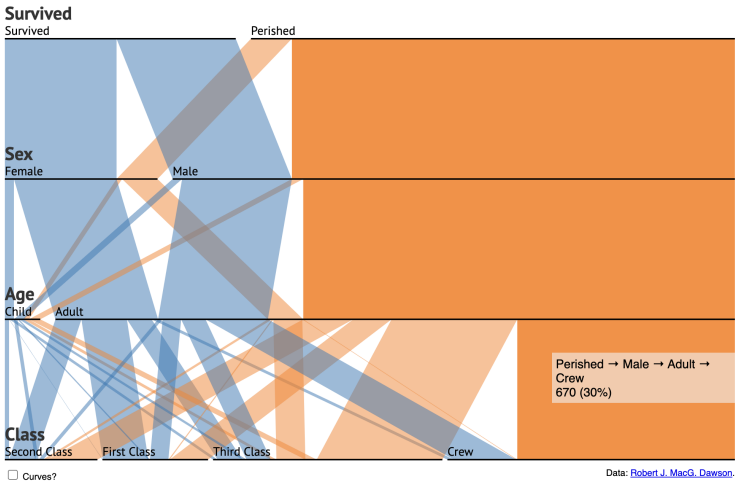


Figure 2.4: PSP representing categorical distribution within Titanic survivors (Jason Davies, no date)

PSPs would be useful when working with system-wide data, where more records with categorical data are available. They may be useful for providing additional context to assist in any hypotheses the clinicians may have. For example, of all male patients in the system, how many suffer from PPMS? This offers insights into the likelihood of a male patient actually having PPMS.

2.3.4 Mosaic plots

Mosaic plots are another way of visualising multivariate categorical data. They represent the proportion of records in the dataset within each combination of categories. However, PSPs show how each category splits off into other categories, while mosaic plots highlight the data distribution across category sets more clearly through cell area. Mosaic plots are however typically limited to two to three variables; anything more makes analysis difficult. Thus, mosaic plots can be used as an extension to PSPs, offering a more focused analysis of categorical data distribution with fewer variables in isolation.

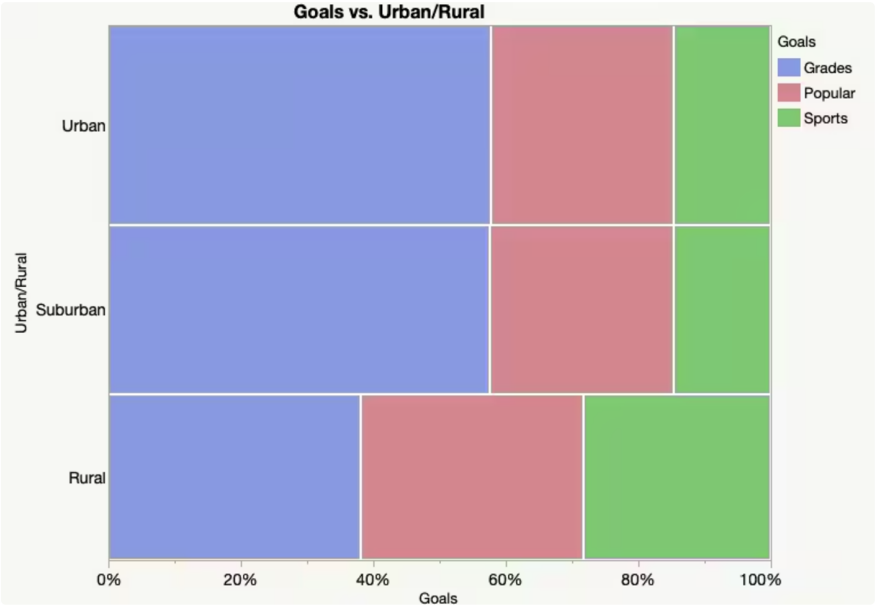


Figure 2.5: Mosaic plot of student goals at different locations (JMP, no date)

2.3.5 Box plots

Box plots quickly show key distribution figures for a given variable (maximum, minimum, upper and lower quartiles, median). They will be useful for summarising the values of a specific DMO across several identified walking bouts or strides.

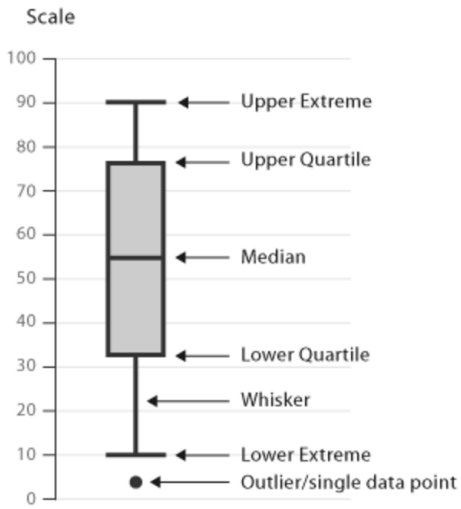


Figure 2.6: Key regions of a box plot (The Data Visualisation Catalogue, no date-d)

2.3.6 Radar charts

Radar charts might be useful in comparing overall gait performance between patients or walking bouts. Multiple DMOs can be added as separate axes on the radar chart, with each record's DMO values plotted on the axes and connected to form polygons. The shape and area of a polygon can be interpreted as an overview of overall performance. It is possible to have more than one (translucent) polygon on the radar chart to compare different records easily, but this should be limited to aid readability. There may be a need to normalise the scale across all DMOs, which have different units and value ranges, to avoid misinterpretation.

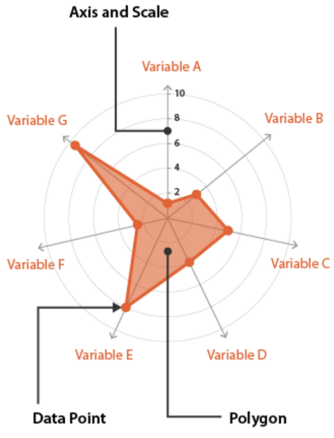


Figure 2.7: Components of a radar chart (The Data Visualisation Catalogue, no date-e)

2.3.7 Heatmaps

A quick graphical way to spot areas of concern or correlation as variable values are displayed with colours from a scale. This technique offers straightforward visual comparisons between DMO values. Heatmaps are good for multivariate data. Typically the rows and columns represent categorical variables, while the cells contain either numerical or categorical data (The Data Visualisation Catalogue, no date-c).

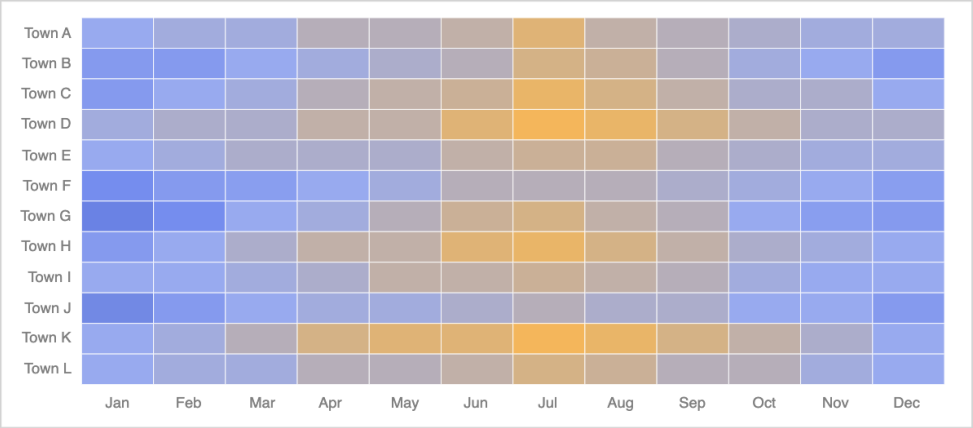


Figure 2.8: Heatmap of temperatures throughout the year for different towns (The Data Visualisation Catalogue, no date-c)

2.3.8 Data requirements and visualisation in healthcare

The main aim of clinicians with data visualisation is to make quick and well-informed conclusions on the patient's current and projected condition (here, on the patient's mobility) so that they can provide suitable treatments.

One data source in healthcare is electronic health records (EHRs), which are patient profiles with information such as medical and treatment histories. This concept can be replicated in the system to provide familiarity for clinicians and is also a more organised storage of patient data. Additionally, as this data is sensitive, substantial consideration has to be placed on its secure storage.

Dashboards are common in healthcare, offering complete overviews of key figures, charts and graphs. Combining dashboards with EHRs, patient outcome dashboards provide a well-rounded health overview of specific patients (Medesk, no date). A cluttered design should be avoided and clinicians should be able to explore components of the dashboard more deeply if need be (Vaniukov, 2024; Dunskiy, 2023).

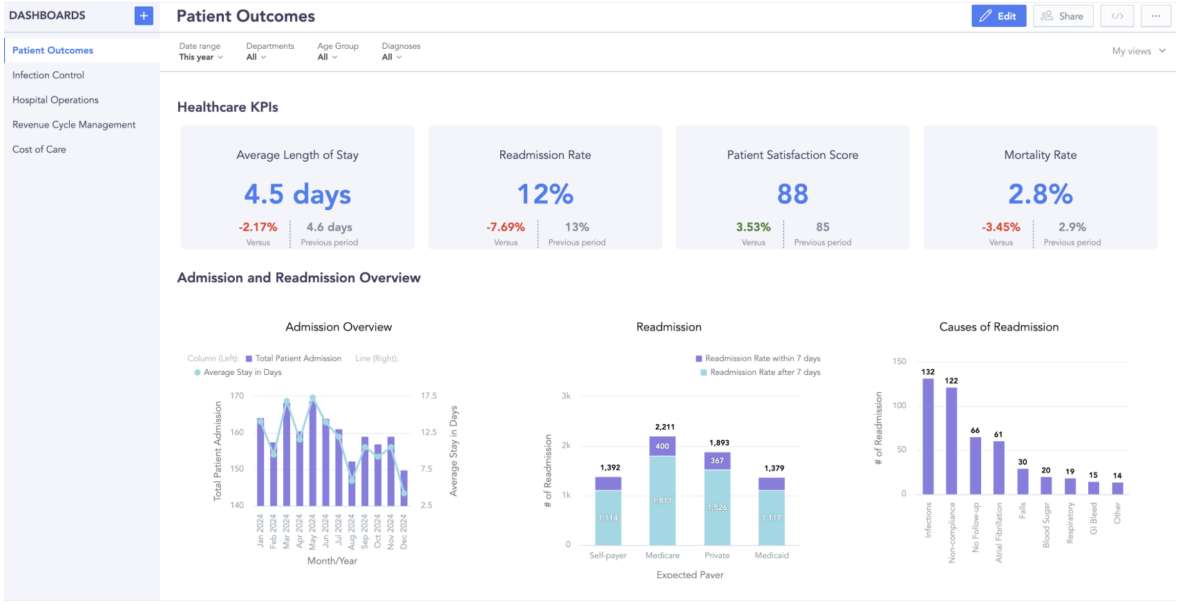


Figure 2.9: Patient outcome dashboard (Czaban, 2024)

Another common concept is interactivity. Clinicians often require many visualisations to deepen their patient analysis, and it is not efficient to try to show every single possible graph. It is simpler to allow the clinicians to manually manipulate existing graphs and dataset filter settings (Dunskiy, 2023).

When visiting any clinic/hospital, infographics and posters can be seen throughout hallways because they offer concise and illustrated views of medical concepts (e.g. body parts affected by disease). Although this project is more clinician-facing and infographics/posters are often more targeted towards the public, these visualisation techniques are worth considering due to their easily-digestible nature.

Figure 2.10: Infographic of the symptoms of MS (Northwestern Medicine, 2022)

2.3.9 Review of existing visualisation tools

Evaluation has been conducted on existing tools that create visualisations and analyse gait data. Table 2.2 provides a summary of noteworthy strengths and gaps of the systems analysed. These are considered during requirements gathering and design stages.

System	Strengths	Gaps
RunScribe	• Summary dashboard of key gait stats and community data. • Gauge charts to classify data (e.g. high/low). • Change visualisation options like data smoothing or units used. • Grouping related metrics and colour-coding. • Time series line graphs partitioned into stance and swing phases. • Connect with other users and manage/view their data. • Comparison tool between custom stat sets, with different views (e.g. graph, precise values).	Mainly uses time series line graphs, lacking other graph and chart types that may provide more insight.
Proto Kinetics Movement Analysis Software	• Can run video recording of walking test with a running visualisation of corresponding footsteps. • Table of gait measurements can be easily exported into Excel to create visualisations. • Select a record in table and highlights corresponding point in visualisation. • Colour coding of pressure in steps. • Show preset time series line graphs in separate windows as needed and organise in view for analysis.	• Relies on Excel visualisations. • Old software and interface.

System	Strengths	Gaps
GaitSmart	• Labels data at respective positions where sensors are placed, in an illustration of the body. • Provides aggregate percentage scores for the assessment. • Values highlighted with traffic light colour coding to highlight if in/outside normal range. • Personalised exercise suggestions. • Patient demographic and recording details.	• Lacking graphs or charts. • In a generated report format, not an app so lacks interactivity and customisation.

Table 2.2: Analysis of strengths and gaps of existing visualisation tools

Information sourced from RunScribe (`https://runscribe.com`), ProtoKinetics (`https://protokinetics.com`), and GaitSmart (`https://www.gaitsmart.com`).

2.4 Summary

Multiple sclerosis worsens the gait of pwMS, and this can be measured by DMOs extracted from an IMU sensor worn in everyday life. Certain DMOs better than others at representing MS and its severity. Validation of DMOs is still in-progress and estimation algorithms are not completely perfect. Therefore, data that will be visualised may not be 100% reliable until validation is complete and improvements to algorithms are made. Still, this project readily provides a tool for clinicians to visualise and make sense of patient gait. Finally, there are many useful visualisation techniques used in existing systems that should be considered.

Chapter 3

Analysis

Firstly, the target users of this project are defined. Then, the iterative approach is described for collecting project requirements that satisfy the target users. IMU sensor input data, functional and non-functional requirements are outlined, which are then prioritised. Technical details for implementation are explored, specifically required tools and technologies. This chapter then concludes with relevant ethical issues, possible project risks and a planned final method of evaluation.

3.1 Target user and needs

The target users are clinicians that have patients with MS and are looking to assess their gait through data collected by a single IMU sensor worn at the lower back. The values produced by the IMU sensor are uninformative. Clinicians need this data extracted into a more meaningful form (DMOs) and visualised, so they can create useful conclusions about gait. With insight into the patient's ability to walk and thus mobility, the clinician can recommend treatments for maintaining mobility with MS.

3.2 Iterative requirements gathering and design

To ensure that this project implements features that are useful to clinicians, an iterative approach is used for gathering requirements and finalising designs. Due to time constraints, only one iterative step will be done, but this would still be sufficient in capturing clinician needs and wants.

3.2.1 Methodology

Initial requirements and prototypes were created using research from chapter 2. A questionnaire was then created on Google Forms to gather feedback on these initial designs and requirements. This was then sent out to clinicians that have worked with patients with MS. Results were then considered when finalising project requirements (section 3.3) and designs (chapter 4). Feedback was collected under the approval of The University of Sheffield's Ethics board. Copies of the questionnaire, participant information sheet, consent form and ethics approval letter are available in appendices A, B, C and D respectively.

The feedback questionnaire was structured into individual sections to initially build an understanding of the project before collecting feedback:

Section one firstly includes a brief introduction and a reference to the participant information sheet, which contains essential information about the overall project, the questionnaire's purpose and content.

Section two integrates points from the consent form, so responses are only considered if all consent points are agreed to.

Section three verifies that they are clinicians working with patients diagnosed with multiple sclerosis (MS), or if not, their related profession.

Section four presents the initial lists of data requirements, and prioritised functional and non-functional requirements. This is to build an initial understanding of planned system features.

Section five contains a reference to the Figma design file with the mock-ups (Hum, 2025-a), with one short 5-minute video briefly explaining each mock-up page (Hum, 2025-b). This is accompanied by a longer, optional 25-minute video with more in-depth explanations (Hum, 2025-c). These visual resources develop on the initial understanding.

The final section six contains open questions to collect feedback about the initial requirements, mock-ups and intended visualisations. Then, respondents are asked to give a rating on the overall usefulness of the system.

3.2.2 Results

Five clinicians, who have experience working with patients with MS, filled out the questionnaire. As the questionnaire is dominated by open questions, summaries of answers are provided in this sub-section.

Category	**Points**

Functional requirements	• Change from individual clinician accounts to health centre accounts, as clinicians may share patients.
	• Clinicians are unfamiliar with generating CSV files. Better to integrate with wearable devices to extract data from them directly.
	  • Offering data transformations into correct units and coordinate system are must-haves.
	• Reference ranges are a must-have to interpret results in context.
	  • Highlight areas of interest. For example, the presence of any abnormalities.
	• Creating new custom visualisations are a must-have.
	• Integration with electronic health systems rather than standalone additional system (could-have).
	• Add non gait/MS-related data, such as patient injuries and feelings during gait assessment, to reference alongside gait results.

Non-functional requirements	• Emphasis on secure storage of data, because medical data is sensitive information. • Ensure colours used are colour-blind friendly.

Visualisations	• Three clinicians expressed approval of planned visualisations in initial mock-ups.
	  • Particularly appreciated the different displays of data, progression of values and comparisons with previous recordings.
	• Each plot requires sufficient textual description as clinicians are unlikely trained in statistics to be able to form interpretations.
	  • Technical wording should be simplified. For example, "aggregate" to "overview" and "per walking-bout" to "each walk".

Usefulness in forming practical conclusions about patient mobility	Average score of 7.6/10.

Table 3.1: Summary of points received from feedback questionnaire about initial requirements and mockups

These results have been considered in the final set of requirements (section 3.3) and during development. Some of the suggestions were infeasible for the scope and duration for this dissertation project, but are considered as potential future work.

3.3 Project Requirements

IMU sensor data, functional and non-functional requirements are listed below. IMU sensor data requirements involve expectations surrounding the core input data for extracting DMOs. Functional and non-functional requirements define features and attributes of the system, and they are assigned priorities according to the MoSCoW method (M = must have, S = should have, C = could have, W = won't have). Those classified as "won't have" are less realistic given time and resources, but will be considered as future work.

3.3.1 IMU sensor data requirements

These are requirements on the IMU sensor data uploaded into the system, so that expectations are set on what types of input data the system can handle. Specifically, these are expected by the DMO extraction library used in this project, *mobgap* (Küderle et. al, 2024).

#	**Requirement**
1	Input data should consist of raw values collected by an IMU sensor worn on the lower back by a person with MS.
2	Input data should contain acceleration values in m/s^2 and angular velocity values in deg/s, for all x, y, z directions.
3	IMU sensor must be worn such that the resulting coordinate system aligns with mobgap's expectations (see sub-sub-section 3.4.1.1).
4	Basic patient and recording metadata of the input data known. Sampling rate (hz), sensor height (m), patient height (m), and measurement setting (lab or real-world) are mandatory.

Table 3.2: IMU sensor data requirements

3.3.2 Functional requirements

Functional requirements are about the concrete functions and features of the system.

#	Requirement	Priority
1	For a new gait analysis, the user is able to upload a CSV of sensor data and input other details (for data requirements and additional descriptions) to automatically generate relevant, pre-defined data visualisations.	M
2	There should a section detailing important input data requirements to ensure reliable information about patient gait is extracted.	M
3	Where applicable, the user is able to edit the settings of a given data visualisation (e.g. pick specific data records to visualise).	M
4	The user is able to create their own account that will hold all their patient and gait analysis data.	S
5	The user is able to create new patients to store their background information (e.g. age and sex) and gait analyses.	S
6	The user is able to effectively filter through gait analyses and patients.	S
7	The user is able to select subsets of data to compare (e.g. between average DMOs of two patients). Different visualisations should be used for comparison (e.g. graphs or table of precise values).	S
8	A reference range should be provided with colour coding for DMO values to provide context for analysis (e.g. what is considered low vs. fast gait speed).	C
9	Each visualisation can be exported into an image format and saved in the user's device.	C
10	The user is able to create new, custom visualisations, outside of the automatically generated ones.	C
11	There are useful infographics and posters conveying general information about MS and gait, that may aid analysis.	C
12	The user is able to automatically transform their CSV data into the appropriate format. Specifically: converting into the correct units and coordinate system.	C
13	For a given patient, the user can view a patient outcome dashboard/electronic health record summarising their current patient's gait condition and historical analyses.	C
14	The user is able to organise the layout of visualisations.	C

Table 3.3: Functional requirements

There was feedback (sub-section 3.2.2) emphasising the need for custom visualisations and direct integration with measurement devices. However, this is unrealistic with the time available. Reference ranges were also desired by a clinician questionnaire respondent, but due to the novelty of DMOs, there is insufficient information on reliable reference values.

Hence, these are assigned lower priority.

Visualisations are the focus for this project, while features like user accounts and creating an all-in-one gait analysis application are additional.

3.3.3 Non-functional requirements

Non-functional functional requirements relate to the quality of the system. This covers areas like speed, security and ease of use.

#	**Requirement**	**Priority**
1	Data uploads are optimised and as fast as possible.	M
2	The interface is optimised for desktop screens.	M
3	Descriptions should be as simplified as possible for clinicians to easily understand.	M
4	All information displayed is reliable and accurate for clinical analysis.	M
5	All data is stored securely and can only be accessed by the right users.	M
6	Navigation throughout and use of the system are easy, intuitive and assisted if necessary.	M
7	Invalid interactions are correctly discouraged and the user is guided towards the right paths.	M
8	The overall design and theme are modern, organised and aesthetic.	S
9	Pages load quickly.	S
10	Colours used are colour-blind friendly.	C

Table 3.4: Non-functional requirements

3.4 Tools and technologies

In modern day software development, there are many frameworks and libraries that can be leveraged to streamline implementation. This section analyses the core areas of implementation and what technologies will be used in each.

3.4.1 Digital mobility outcome extraction

DMOs are the main data that will be visualised. Mobgap (Küderle et. al, 2024) is the official library developed by Mobilise-D for extracting DMOs and thus is the chosen library

for extracting gait-related DMOs from IMU sensor data.

Mobgap provides Python implementations for both individual algorithms for extracting specific gait-related DMOs and complete high-level end-to-end pipelines for extracting the main DMOs (cadence, stride length, stride duration, walking speed) from which other DMOs can be derived. These pipelines were validated by the studies seen in chapter 2 (Micó-Amigo et al., 2023; Kirk et al., 2024), and thus will be the main method of extracting DMOs. Particularly, the recommended pipeline for MS (P2 in Kirk, et al., 2024) will be used.

3.4.1.1 Coordinate system

Attention has to be paid towards the inputs expected by mobgap, as these inputs will naturally also have to be requested from the users. These have been outlined in section 3.3.1, but the specifics on the coordinate system will be described here.

The coordinate system associated to the uploaded IMU sensor data must align with mobgap's expected coordinate system. Otherwise, the algorithms will not produce the correct output. Put simply, the IMU sensor must be worn correctly - such that the orientation aligns with the expected coordinate system.

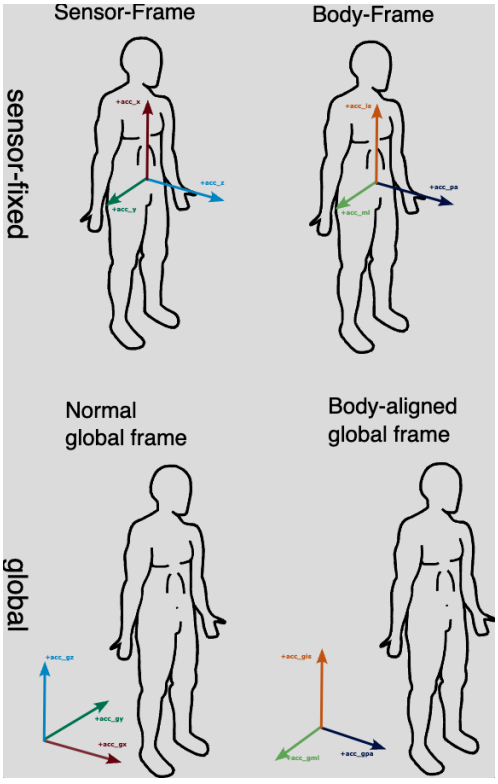


Figure 3.1: Mobgap coordinate systems (Mobgap, no date-a)

As seen in figure 3.1, there are four different coordinate systems that each define a set of axes. However, only the sensor frame is important, which creates implications for how the IMU should be worn. From the point of view of the wearer, the x axis points upwards, the y axis right and z axis forward.

Functional requirement 12 involves converting the uploaded data into the right coordinate system. There are two approaches for transformation, depending on whether the mounting orientation of the sensor is known. This can be found in the sensor/manufacturer's documentation or with a test recording (Gaitmap, no date). This is dependent on the worn sensor unit. Thus, the clinician has to personally determine the transformations. If mounting orientation is known, transformation is simple by just using a fixed rotation matrix, otherwise this rotation matrix has to be estimated using the data. Only when the coordinate systems have aligned, then can the data be correctly transformed into the body frame (using functions in mobgap) and ran through the pipeline. Due to the complexity of this process, the requirement is low in priority.

3.4.2 Web development framework

The visualisation interface will be web-based, and there are many technologies that simplify web-development. For this project, Next.js is used, a modern full-stack React framework. In addition to familiarity, it has many advantages, from file-based routing to its React-implementation. Additionally if required, this project be deployed easily using Vercel, a platform for hosting Next.js projects.

3.4.3 Data visualisation

There are lots of front-end libraries available for creating and customising data visualisations, such as D3.js, Chart.js and Plotly.js.

Customisability and interactivity are needed. For example, brushing and axes manipulation in a PCP. Therefore, D3.js has been chosen, as it offers ample control and customisation to developers. Example work to reference also exist on different plots and graphs.

3.4.4 Interface components

Given the strict time constraints, it is imperative to consider features that should and shouldn't be created from scratch. Component libraries, such as Shadcn UI, Chakra UI and Ant Design, are available. These make it easy to add and customise attractive UI components (such as drop-downs).

Shadcn is chosen, due to its minimalistic and modern aesthetic, and ease of component customisation.

3.4.5 Data storage

Data is extremely relevant in this project, from DMO values to account logins. Depending on the project progression, either local storage on the user's browser or a database will be implemented. Features requiring user logins prompt the need for a standalone database. Local storage inherently ensures security, while specific measures will need to be considered for a database.

In a database, data must be linked; for example patients to their DMO data. Thus, a relational MySQL database is the project's choice. It is compatible with Prisma, which is an object relational mapper frequently used to connect databases to Next.js projects. Specifically, the free tier of Amazon Relational Database Service (RDS) would be used, a cloud-based MySQL database. This is sufficient for academic projects, but also allows for an easy transition to real-world deployment.

3.4.6 Connection between front-end and back-end

Due to the decision to use Next.js, a special connection is needed between the front-end JavaScript framework and the backend in Python (to use mobgap). An API endpoint must be created in Python, which can be called from the front-end. There are many web frameworks for building APIs in Python, such as Flask, FastAPI and Starlette.

Due to the abundance of resources online about the integration between FastAPI and Next.js, FastAPI has been selected.

3.4.7 Testing

Testing is required to ensure that implemented features function correctly. In addition to user testing (section 3.7), automated unit tests will be created. React Testing Library with Jest will be used to test front-end components function and display properly. Pytest will be used to test back-end functionality such as DMO extraction. Finally, manual integration testing will be conducted to ensure edge cases are covered, CSV files properly upload, visualisations correctly generated and interactions work.

3.5 Ethical, professional and legal issues

3.5.1 Feedback collection

The requirements gathering and evaluation processes will involve feedback collection from real clinicians. These introduce ethical considerations.

An ethics application was created for the first feedback questionnaire, and approved by The University of Sheffield Ethics Board. Copies of the consent form, participant information sheet, and approval letter can be found in appendices B, C and D respectively.

This application was followed by an ethics amendment application, specifying additional information for the evaluation process with clinicians (section 3.7). Copies of the amendment, and modified portions of the consent form and participant information sheet can found in appendices F, G and H respectively.

Surrounding personal information, questionnaires only collect email addresses and descriptions of occupational background to identify the respondents. Additionally, identifying information (e.g. names) are not revealed in the discussion of results in this report. Responses are also destroyed within a month of this project's completion date.

3.5.2 Sensitive data

As this project revolves around sensitive patient data, only information required for the system's functions should be stored. Only clinicians can access their own data that is stored in local browser storage. For a standalone database, encryption should be enabled, and user authentication and authorisation should be properly implemented.

For the scope of this dissertation project which is focused on visualisation, only synthetic sample datasets provided by the developers of mobgap are used during development.

3.5.3 Tools and technologies

Acknowledgment and licensing of tools used are important. Specifically, use of the mobgap library is described as demanded by its creators. All the tools and technologies mentioned are also free to use.

3.5.4 Research data

Finally, as per functional requirements 8 and 11, data collected in other studies may be presented. These will have to be cited and referenced accordingly in the system, to state the source and attribute work done to its owners.

3.6 Risks analysis

Table 3.4 shows the potential risks concerning this project and mitigation measures that will be taken to prevent their occurrence.

Risk	**Mitigation measure**
Clinicians don't find the system useful.	Iterative approach to gathering requirements and system design will establish their needs and wants early. Prioritisation of requirements also ensures most important features are delivered.
Mobgap is still under development and new, unexpected updates may affect progress.	Contact developers to get an idea of plans for the library throughout project duration, and adjust project plan if necessary.
Final system is buggy or hard to use.	Thorough testing procedures involving automated unit testing, user testing, and integration testing. Maintain passing test cases.
Use of unfamiliar tools (e.g. D3.js) take too long to learn.	Plan a dedicated learning period for learning the basics. Work with these tools early to develop understanding.
Final system is incomplete.	Prioritise establishing final requirements and designs early, so that development can begin early.

Table 3.5: Potential project risks and respective mitigation measures

3.7 Evaluation

As this project is targeted towards clinician use, their satisfaction and perceived usefulness of the final system is important. Thus, user testing involving relevant individuals will be conducted, where they will assess the system's usability and efficacy.

User testing will be conducted either in-person or remotely using Chrome Remote Desktop

(CRD). CRD will allow the clinician to view the screen of another computer, with cursor control to navigate through and use the locally-hosted interface.

Basic understanding of this project's aims is expected before interacting with the system. Additionally, sample IMU CSV data files with associated metadata will be provided.

Clinicians should then freely use the system without intervention, to mimic an actual user. They are encouraged to describe their thoughts and decisions aloud as they run through the system. The entire process will be voice and screen-recorded for reference afterwards.

After satisfied with their use, they will be asked to fill out a questionnaire (appendix E). This will contain the integrated consent form (appendix G), participant information sheet (appendix H) , and questions about system usability, adherence to project requirements and system effectiveness.

An amendment of the initial feedback ethics application (appendix F) is sufficient for this evaluation exercise. Results from the recordings and questionnaire are then analysed to suggest future improvements.

3.8 Summary

In this chapter, clinicians have been identified as the core target users. Initial requirements and designs have been iterated on with feedback received from clinicians. Data, functional and non-functional requirements have been established and prioritised. Relevant tools and technologies for implementation have been identified and chosen. Finally, consideration has been performed on ethical issues, risks and the final evaluation process.

Chapter 4

Design and Planning

Planning the appearance and features of the system is crucial before implementation. Equipped with the knowledge of useful visualisation techniques and clinician needs and wants, this chapter presents specific data storage requirements, application flow and initial visual mockups.

4.1 Data storage

The first step is to properly establish what sort of data will be stored and visualised.

The core data are DMOs, which are extracted through the mobgap library. Through analysis of mobgap's documentation (Mobgap, no date-a), it has been determined that gait parameters (or DMOs) are extracted by the pipelines on three different levels: aggregate, per-walking bout (WB) and per-stride.

4.1.1 Per walking-bout level DMOs

A gait recording might run throughout an entire day, but the patient is not necessarily walking at all times. Mobgap therefore identifies valid WBs using the GSD and ICD algorithms discussed in sub-section 2.2.2.

Per-WB level DMOs are defined under each identified WB. Specifically, the *total number of strides, WB duration (s), cadence (steps per minute), stride length (m), and walking speed (m/s).*

4.1.2 Aggregate-level DMOs

These are summarized metrics of the per-WB level DMOs. Mobgap pipelines extract 24 in total, under specific time intervals (Mobgap, no date-b). For example, the average cadence over all identified WBs, or isolated to only WBs that are at least 30 seconds long.

However, there are inconsistencies in the defined aggregated parameters by mobgap. For instance, maximum WB duration is defined over all WBs, but not for the WBs within the interval of 10 to 30 seconds. Küderle (2025) (a mobgap contributor) stated that these are validated aggregations for clinical reliability and consistency. The paper verifying this has not been published at the time of writing.

Therefore, custom aggregations will alternatively be calculated. Specifically, the *maximums, minimums, averages and variances* of the gait parameters mentioned in sub-section 4.1.1, over all WBs. Total duration of all WBs and WB count are also useful and will be incorporated.

4.1.3 Stride-level DMOs

Finally, WBs are made up of individual strides, so gait parameters are also defined under individual strides of each WB. Notably, *a left/right label, stride duration (s), cadence (steps per minute), stride length (m), stride duration (s), and walking speed (m/s).*

4.1.4 Database diagram

A database diagram (figure 4.1) was created using the Unified Modelling Language (UML). It includes the storage of all parameters mentioned in previous sub-sections, as well as personal information such as user accounts and supplementary patient data. This covers all the data required to implement all functional requirements.

However, the main project focus is visualisation, and an entire database is complex. Thus, this entire diagram will only be implemented if time permits.

An initial implementation of data storage using the browser local storage and JavaScript Object Notation (JSON) will be prioritised. This will involve a subset of figure 4.1 (surrounded by the red box). Specifically, only data fields from the tables *analyses, per_stride_parameters, per_wb_parameters, and aggregated_parameters.* As local storage should only store limited data, there will only be one analysis (only one set of gait parameters at each analysis level) saved at a time. This is sufficient in creating most visualisations and fulfilling the must-have requirements. On early completion, it should be extended to a database to store more data and expand possibilities for visualisations.

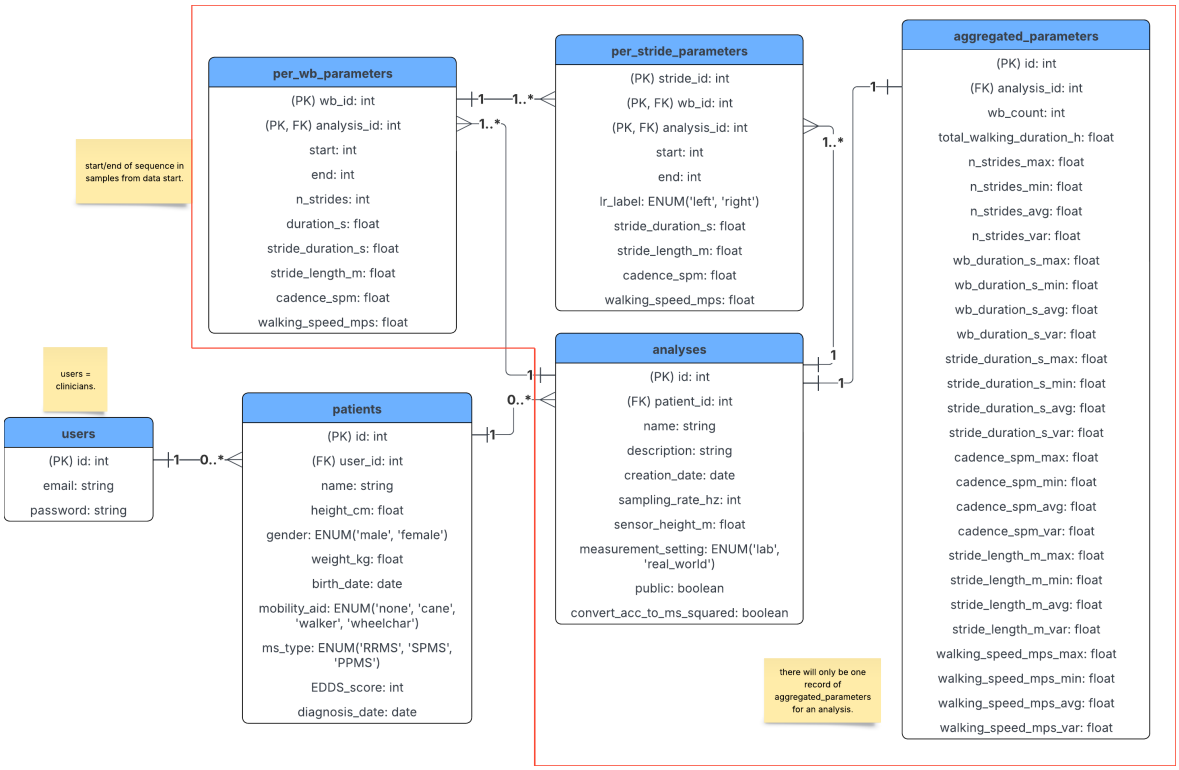


Figure 4.1: Database design of planned tables, fields and associations. Area inside red box signifies data fields that will be prioritised and included in the initial JSON implementation. PK and FK represent primary and foreign keys respectively.

4.2 Application flow

Figure 4.2 shows the planned application flow for the complete implementation of all requirements. Each state (box) represents a page. It starts with basic authentication functionality, such as logins/registration and password resets. Once the user is logged in, they can access and manipulate their list of patients. Clinicians can then select for a given patient, to view the gait analyses of, which they can manipulate and create new ones of. Finally, clinicians can select a specific gait analysis to look into, which can be explored on the three different levels (section 4.1), where the relevant visualisations can be examined and manipulated.

To account for time constraints, priority will be assigned to the components inside the red box. This segment is sufficient to fulfill the main focus of visualisations and must-have requirements. Instead of storing many patients with several gait analyses, there will be one form for the user to submit a CSV file and other essential metadata. Then, only one concurrent set of DMOs will be stored and visualised on the different levels.

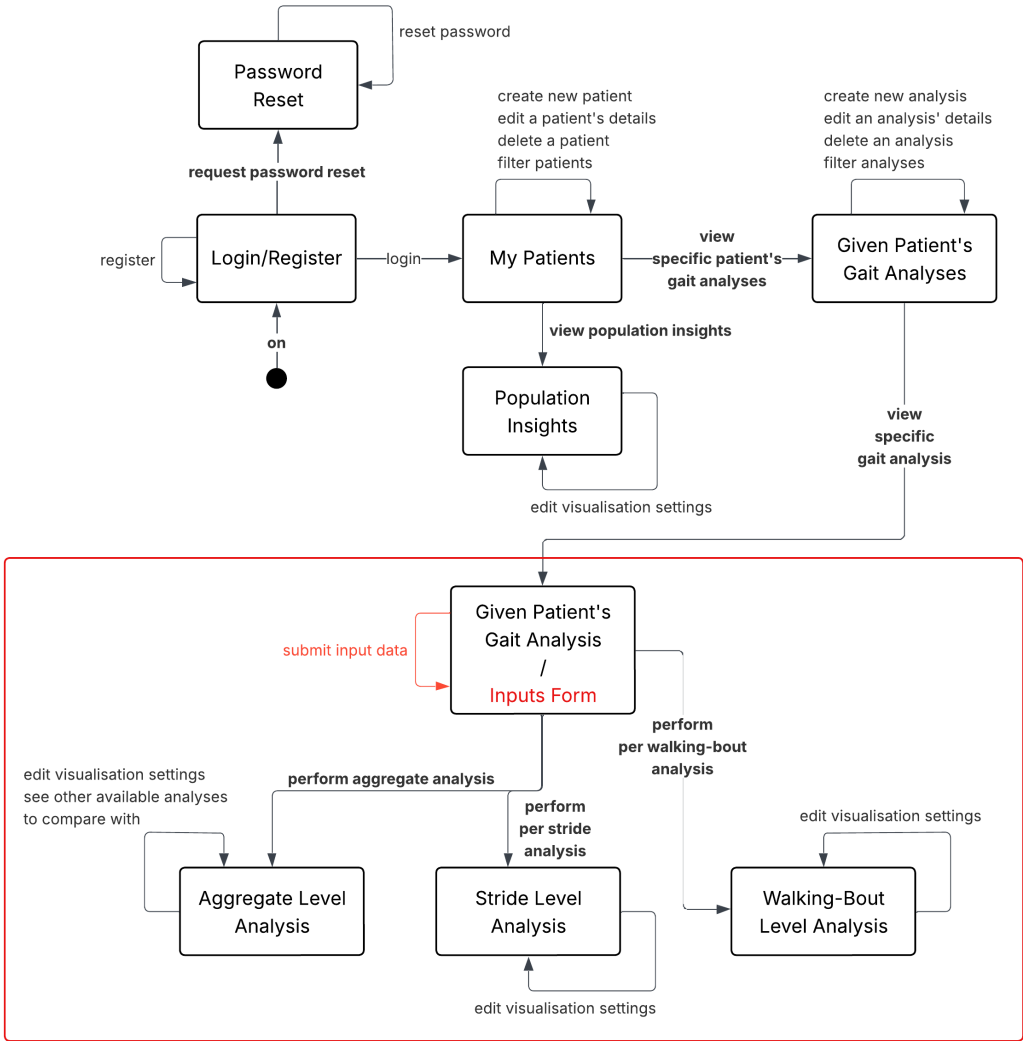


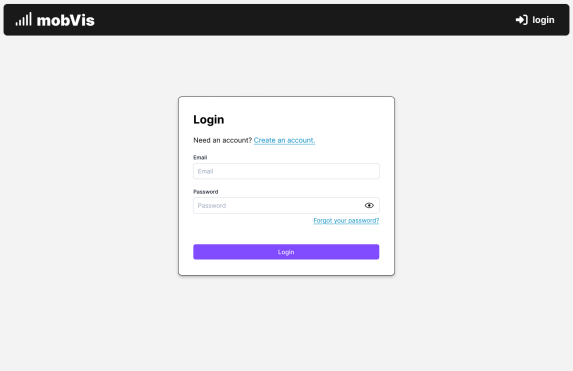
Figure 4.2: Application flow state machine diagram. The area inside the red box, and the text coloured red are must-have, high priority features.

4.3 Mockups

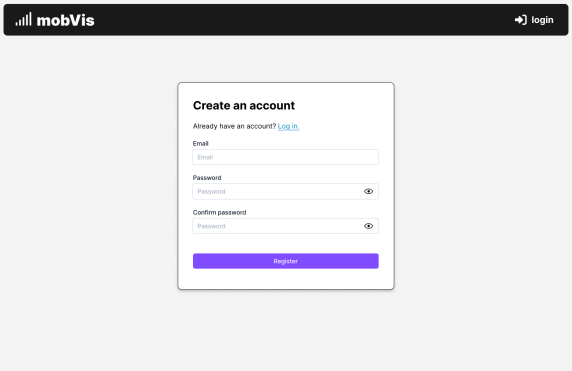
This section presents the initial mockups that were designed to develop a visual understanding of the interface's features. These mockups were also presented in the initial feedback questionnaire. They attempt to fulfill all the requirements (section 3.3). However, emphasis has to be placed on the focus of visualisation. Sections below highlight the designs that are essential to this focus - those that are not labeled "(high priority)" are more optional and may be left as future work.

4.3.1 Clinician authentication

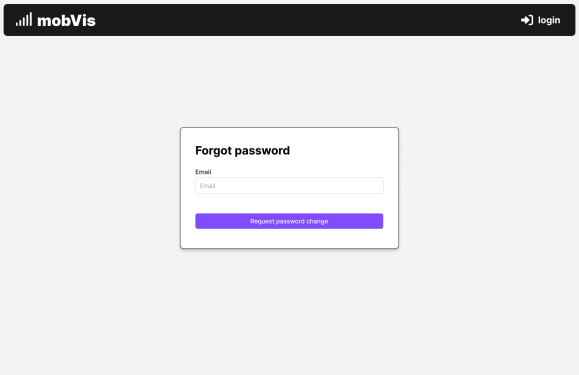
Basic pages for authenticating clinicians as valid users of the system. These include typical login and registration forms, and password reset functionality implemented by a customary reset request email.



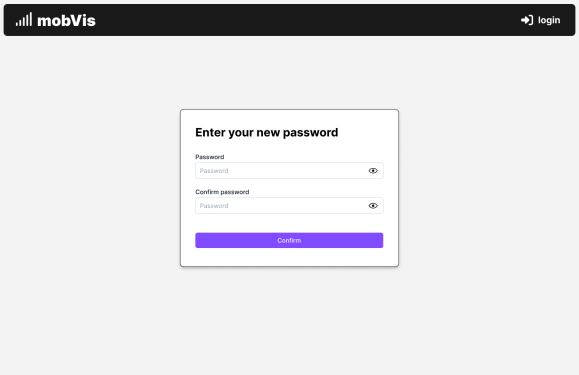
(a) Login



(b) Registration



(c) Forgot password



(d) Password reset form

Figure 4.3: Authentication pages

4.3.2 Patients

4.3.2.1 Patients list

When the clinician logs in, they are greeted by a paginated list of their patients. This page includes all the useful background data on each patient, with functionality to filter specific patients according to their background.

From here, clinicians can create, edit, and delete patients. Additionally, by clicking on "analyses", they are directed to a list of the given patient's analyses (sub-section 4.3.3).

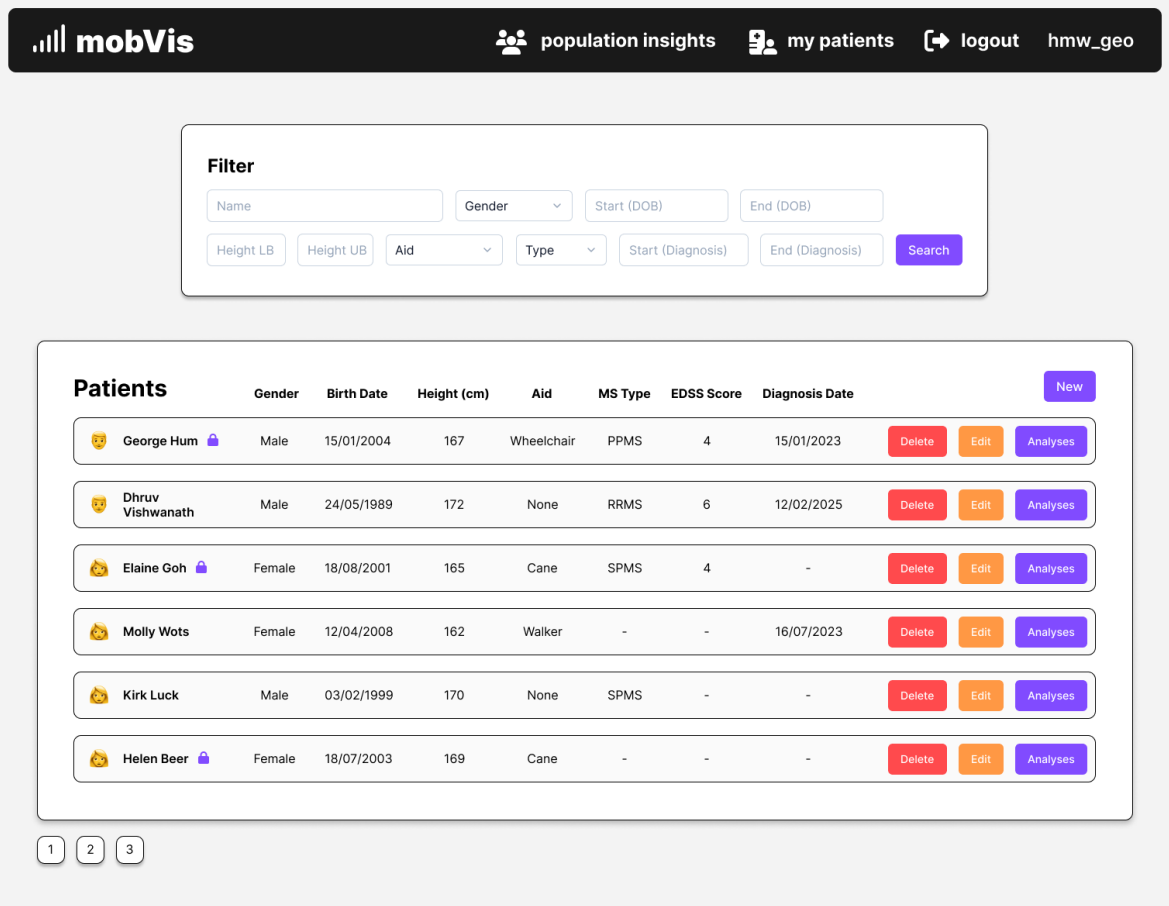


Figure 4.4: List of clinician's patients

4.3.2.2 Creating a new patient

This modal window pops up when the "new" button (figure 4.4) is clicked. It encloses a form for creating a new patient. Essential fields are highlighted with "*", while fields like "diagnosis date" may be unknown and left empty. By checking the "public?" checkbox, the clinician grants the use of the patient's data in the "population insights" feature (sub-section 4.3.4).

The form for editing a patient will be the same, differing by merely the pre-filling of fields with existing patient data.

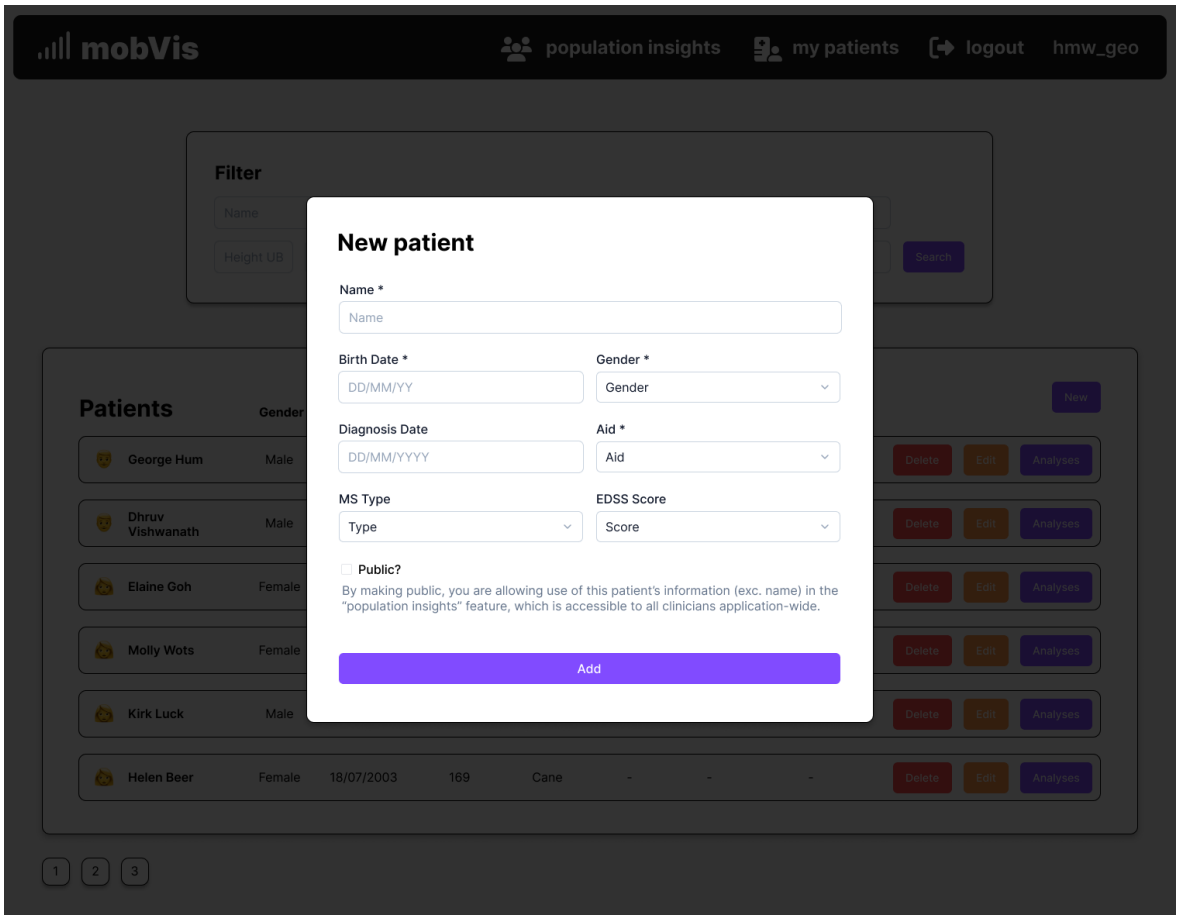


Figure 4.5: New patient form

4.3.3 A patient's analyses

4.3.3.1 List of analyses for a given patient

Similar to the patients list, this is a list of the analyses created for a given patient that is accessed by clicking on the "analyses" button (figure 4.4). It displays basic information about each analysis, with filtering functionality. The clinician can also create a new analysis, and update and delete existing ones. The "analyse" button will direct the clinician towards selecting a given analysis level and then viewing the actual visualisations (sub-section 4.3.5).

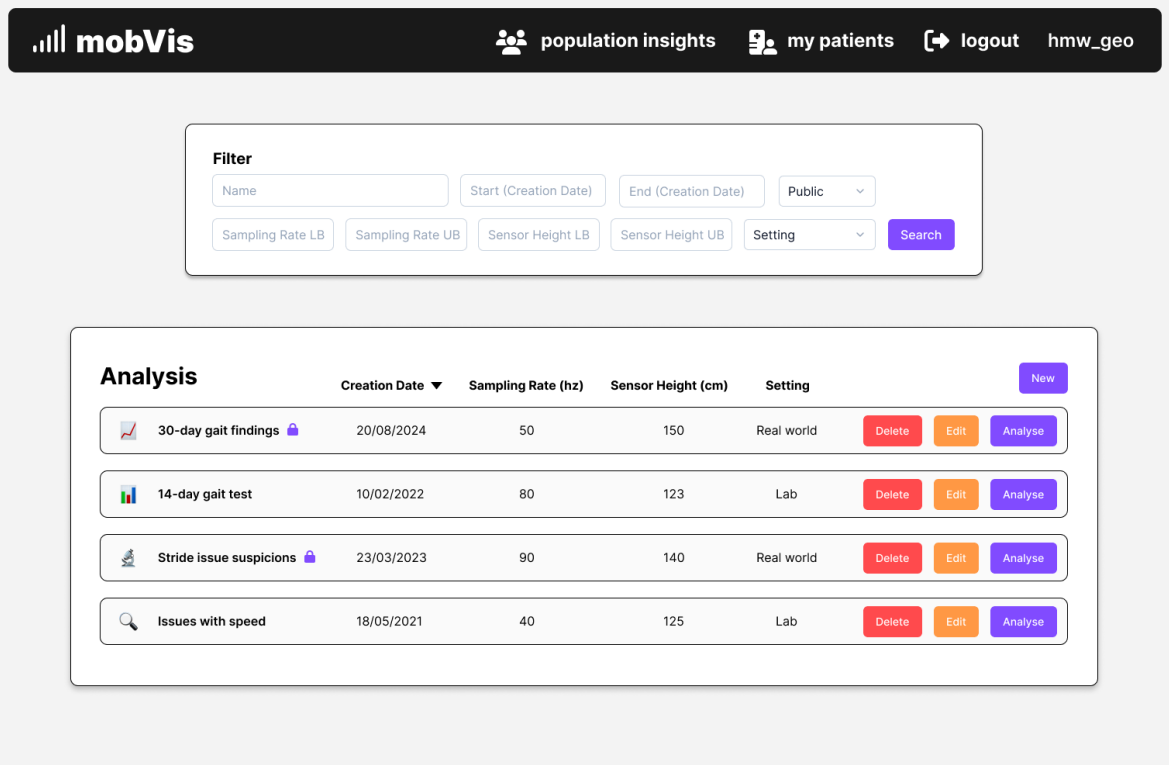


Figure 4.6: List of analyses for a given patient

4.3.3.2 Creating a new analysis (high priority)

Figure 4.7 features the form that appears when clicking on the "new" button in figure 4.6. This requests additional data, such as name and description, and mandatory fields for the mobgap pipelines like the CSV file, sampling rate and measurement setting (real-world or laboratory). The checkbox to publicise the analysis, which allows other clinicians to use this analysis for comparison (see sub-section 4.3.6).

This form is of high priority, to collect necessary information for DMO extraction.

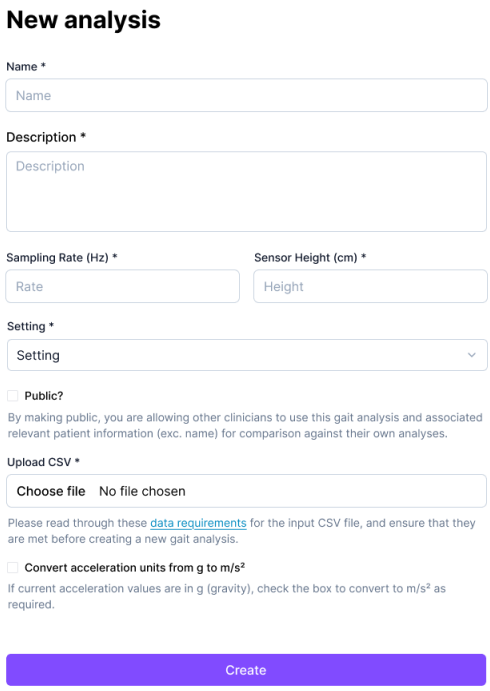


Figure 4.7: New gait analysis form

4.3.3.3 Editing an analysis

Only the fields seen in figure 4.8 can be modified for a given analysis. Since DMOs will be extracted upon analysis creation and the CSV file won't be stored (to minimise storage requirements), pipeline-relevant fields, such as frequency, cannot be modified.

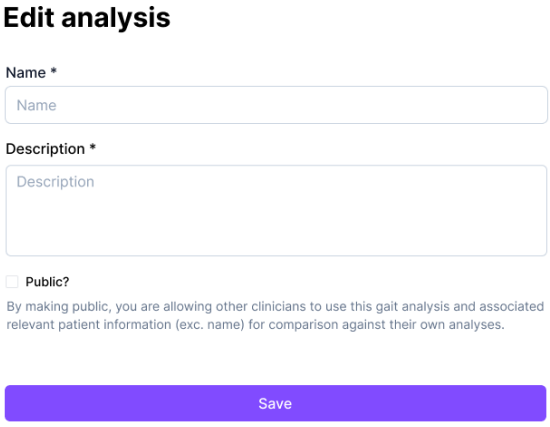


Figure 4.8: Form for editing an analysis

4.3.3.4 CSV data file requirements

This popup appears when the "data requirements" link on the new analysis form is clicked (figure 4.7). These detail format and data expectations of the CSV file. There is also a link to download an example file.

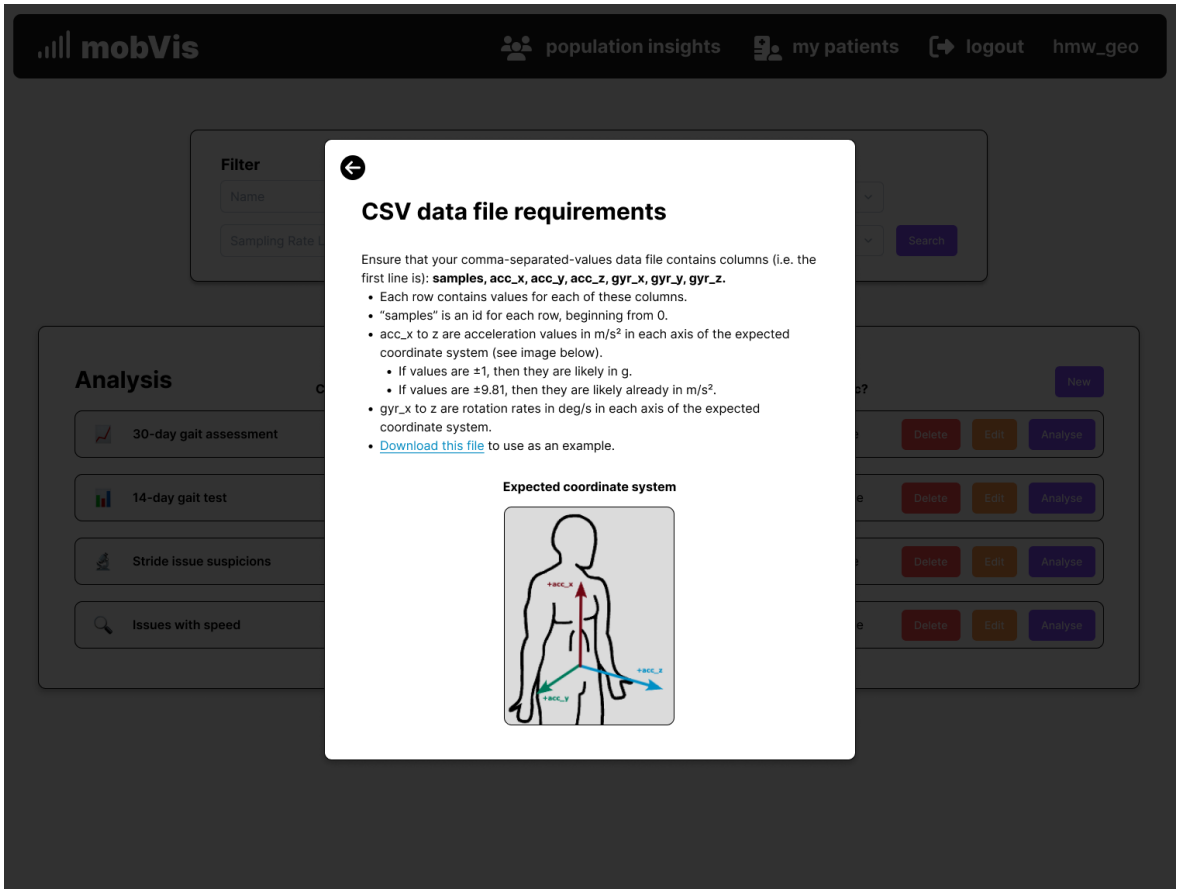


Figure 4.9: CSV data file requirements popup

4.3.4 Population insights

This page (figure 4.10) focuses on categorical patient data (e.g. age, gender) to provide more context to any gait-related conclusions made by the clinician.

The first visualisation is a parallel set plot, which illustrates the distribution of patient background categories in the system, and how they intersect with each other.

The second visualisation is a mosaic plot comparing a patient background category against corresponding average values for a focus DMO. This serves as a reference for expected DMO

values given a demographic.

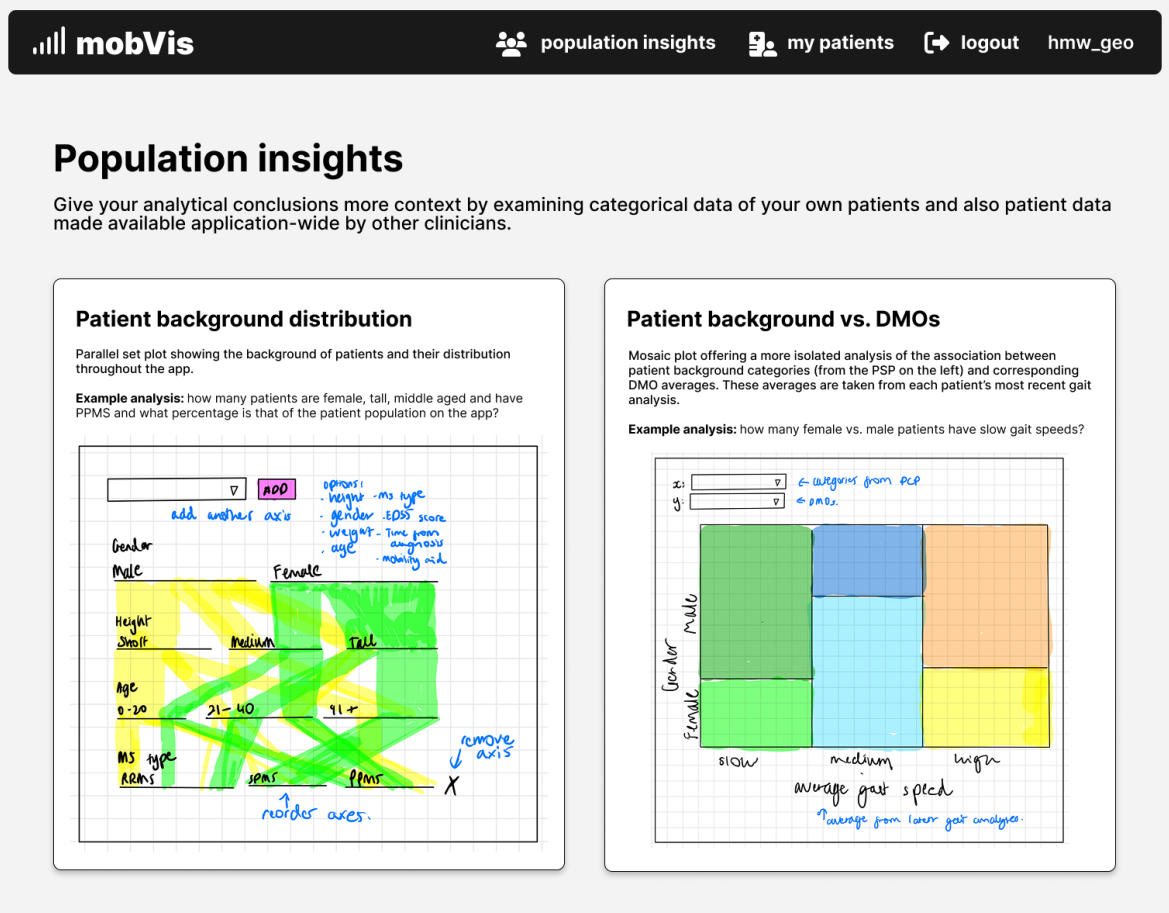


Figure 4.10: Parallel set and mosaic plot visualisations for insights into system-wide patient background information

4.3.5 Pick an analysis level

When the clinician clicks on one of the "analyse" buttons (figure 4.6), they are greeted with the page in figure 4.11. The clinician can see the inputs they submitted for the gait analysis, and as well as a menu for diving deeper into a specific analysis level that contains the actual visualisations.

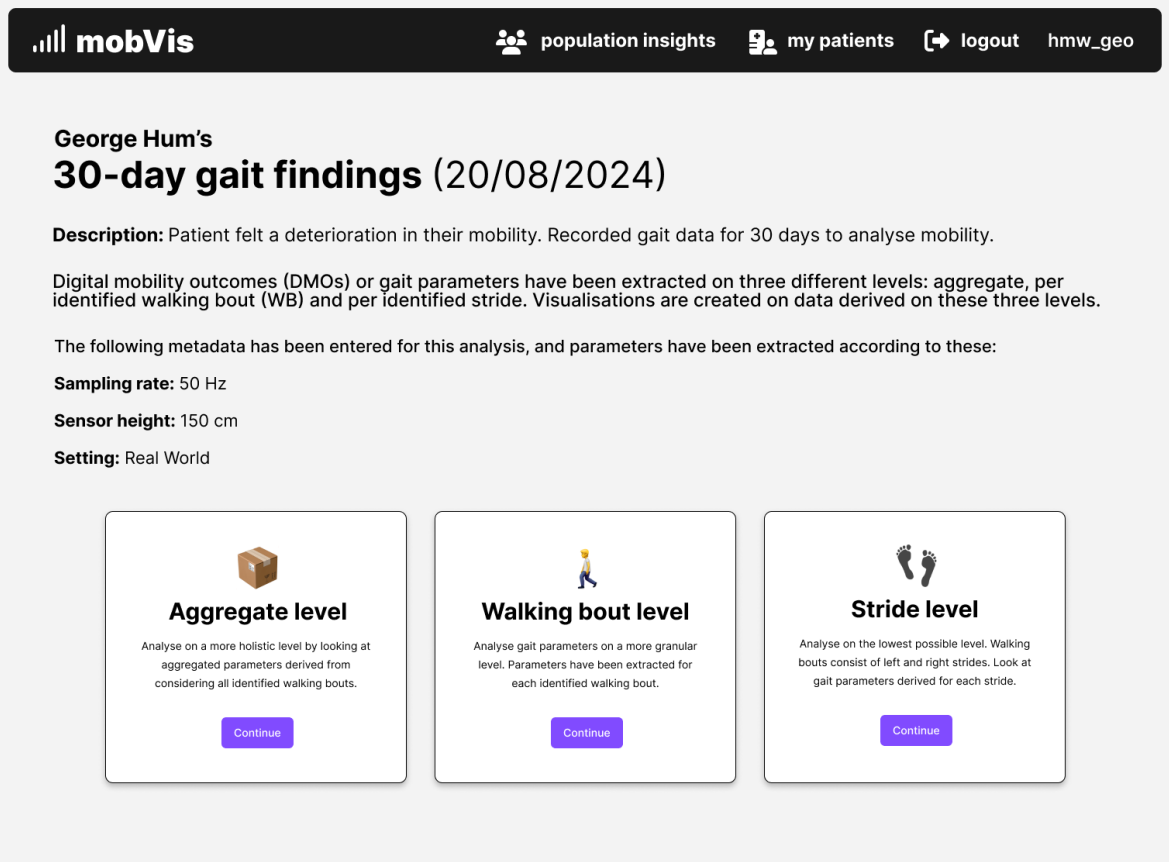


Figure 4.11: Details page of a given gait analysis, with a menu for diving into a specific analysis level

4.3.6 Aggregate/summary level analysis (high priority)

This page contains visualisations that summarise the current, selected gait analysis. For example, it contains stat cards on total walking bouts and duration (figure 4.12). The focus is on summarising all the DMO values across all identified WBs.

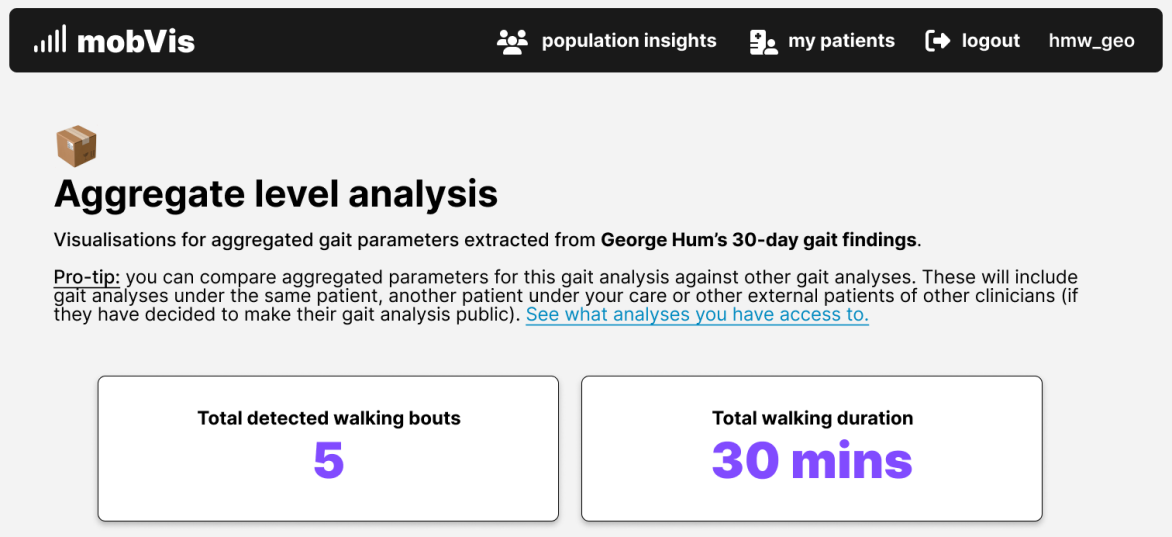


Figure 4.12: Top of "aggregate analysis" page with useful stat cards about detected WBs

A useful feature is the ability to compare with aggregate metrics of other gait analyses in the system (also explained in figure 4.12). By clicking on the blue hyperlink, the user is greeted by the window in figure 4.13. It contains a list of available analyses to compare with the current analysis. Each analysis can be expanded to show more details about it. The analyses include ones owned by the logged-in clinician, or by other clinicians who have decided to make their analyses public.

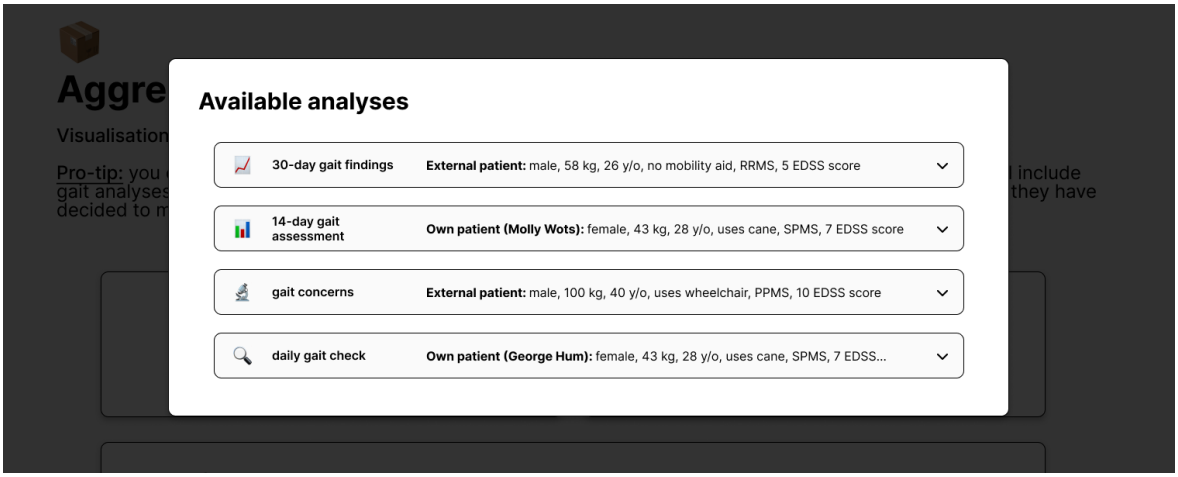


Figure 4.13: List of analyses available to compare the current analysis with

4.3.6.1 Table of all aggregated parameters

This table contains precise values on aggregate metrics (maximum, minimum, average and variance) for each DMO. The clinician can add aggregates for other analyses and DMOs, using the drop-downs.

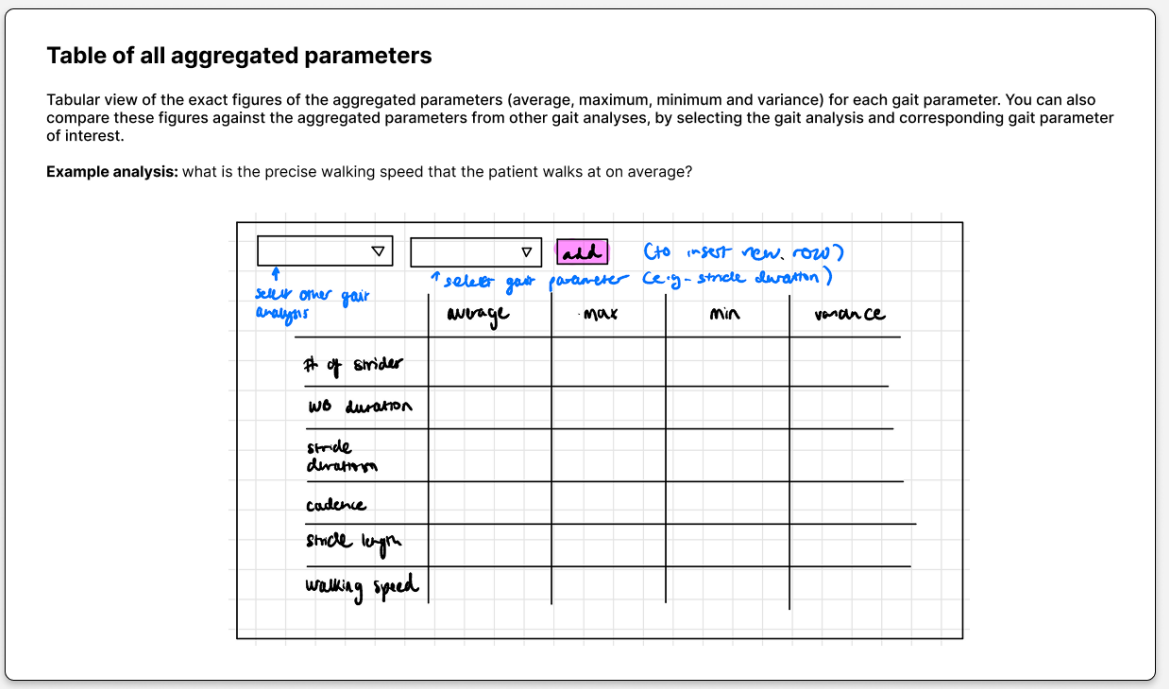


Figure 4.14: Table of precise values for aggregate metrics of DMOs

4.3.6.2 Radar chart of overall gait performance

Gait analyses are compared by their shape. This includes the current analysis, with other selected, comparable analyses from figure 4.13. Their corresponding average values for each DMO are plotted on each axis. The axes can be re-ordered by dragging them around, to offer a different perspective. As with many of the future visualisations, there is a limit set (3) to reduce clutter.

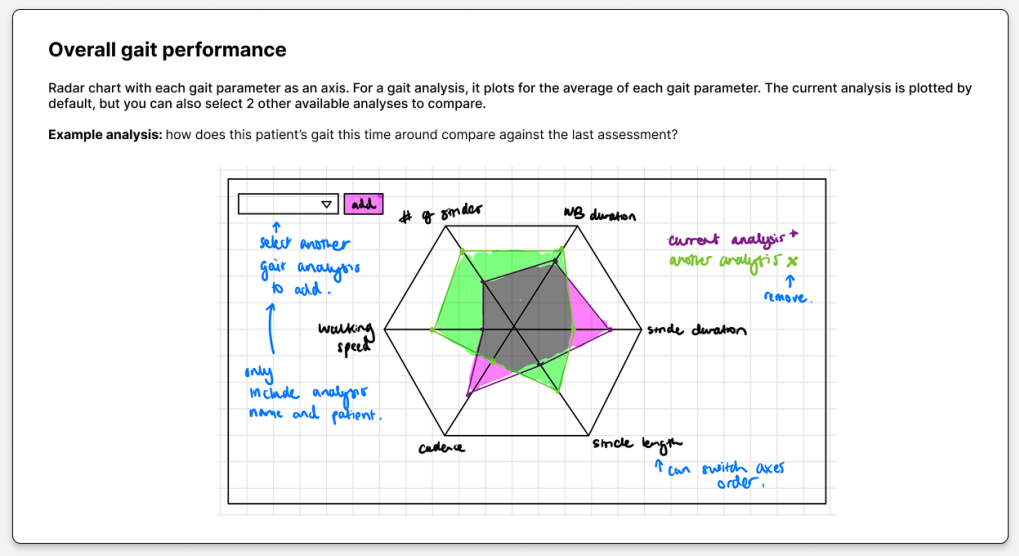


Figure 4.15: Radar chart comparing the overall performances between different gait analyses

4.3.6.3 Violin/box plot of DMO distribution

This is a combined violin and box plot which provides insight into the distribution of values for a focus DMO defined under walking bouts of different analyses. Similarly, the gait analyses and focus DMO can be selected using the dropdowns.

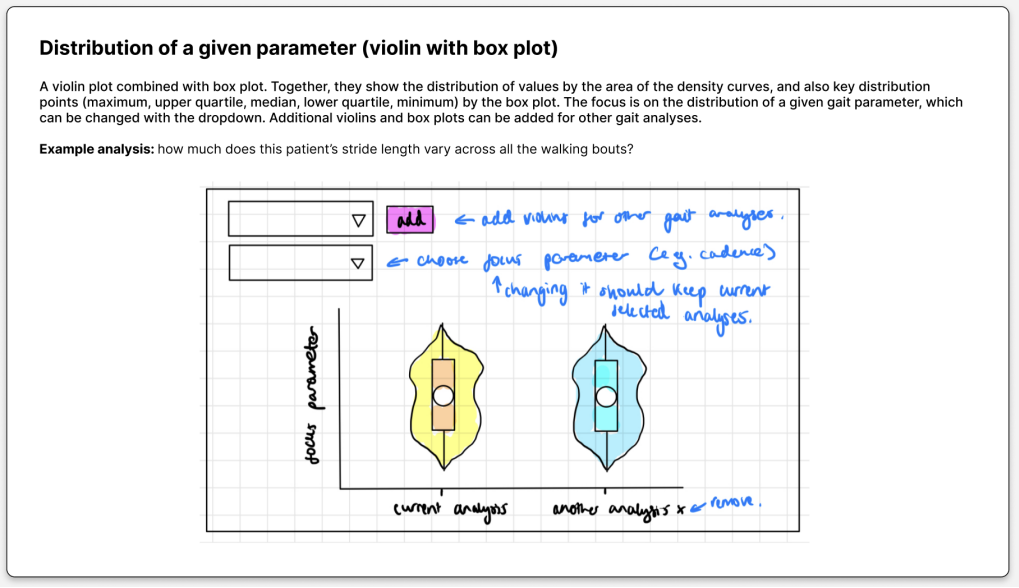


Figure 4.16: Combined violin and box plot depicting the distribution of DMO values across all walking bouts for selected gait analyses

4.3.6.4 Histogram of DMO distribution

This provides another view of distribution through a histogram, which offers a more concrete view of value intervals and corresponding frequencies. Similarly, intersecting distributions can be plotted for a limited set of different analyses.

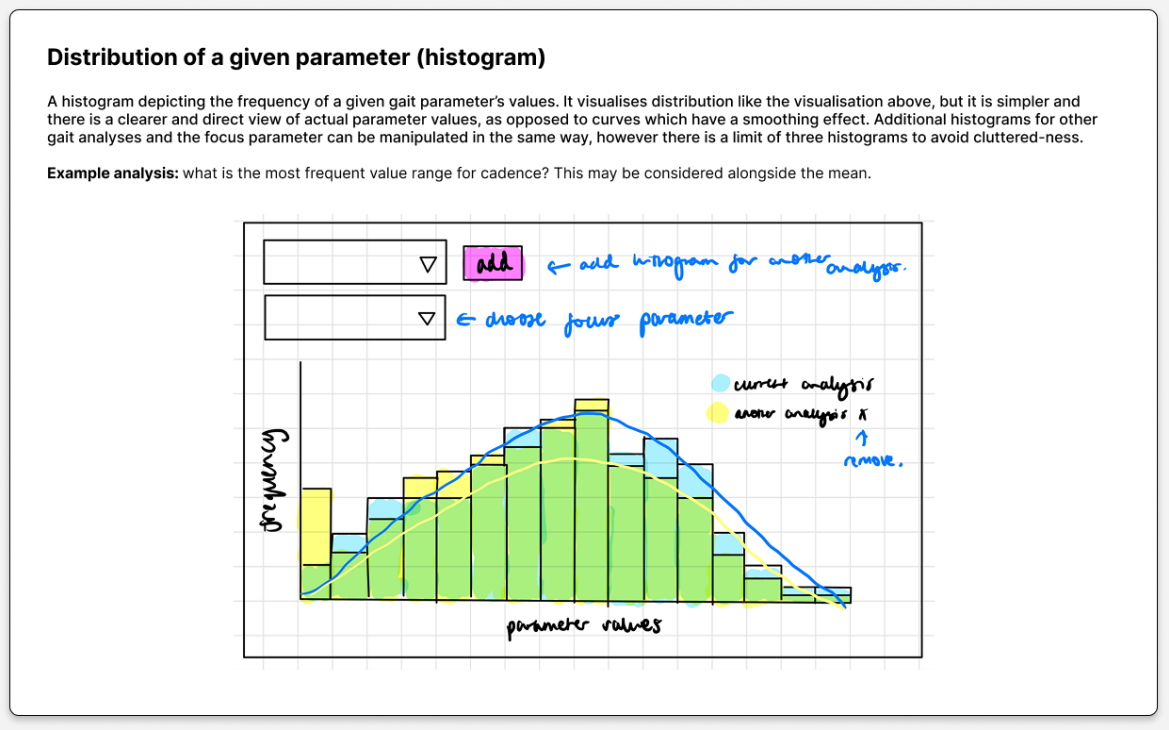


Figure 4.17: Histogram depicting the distribution of DMO values (with visible value intervals and frequencies) across all walking bouts of selected analyses

4.3.7 Walking-bout level analysis

While the aggregate level examined DMO values across all WBs, this level of analysis contains visualisations which focus on DMO values under specific walking bouts. There will be similar visualisations to before, that simply plot specific WBs rather than aggregations over all WBs. These will not be described as they serve the same function, but just offer analysis on a different level.

4.3.7.1 Table of all DMO values

This table presents the exact figures of the DMO values for each identified walking bout from the uploaded data recording.

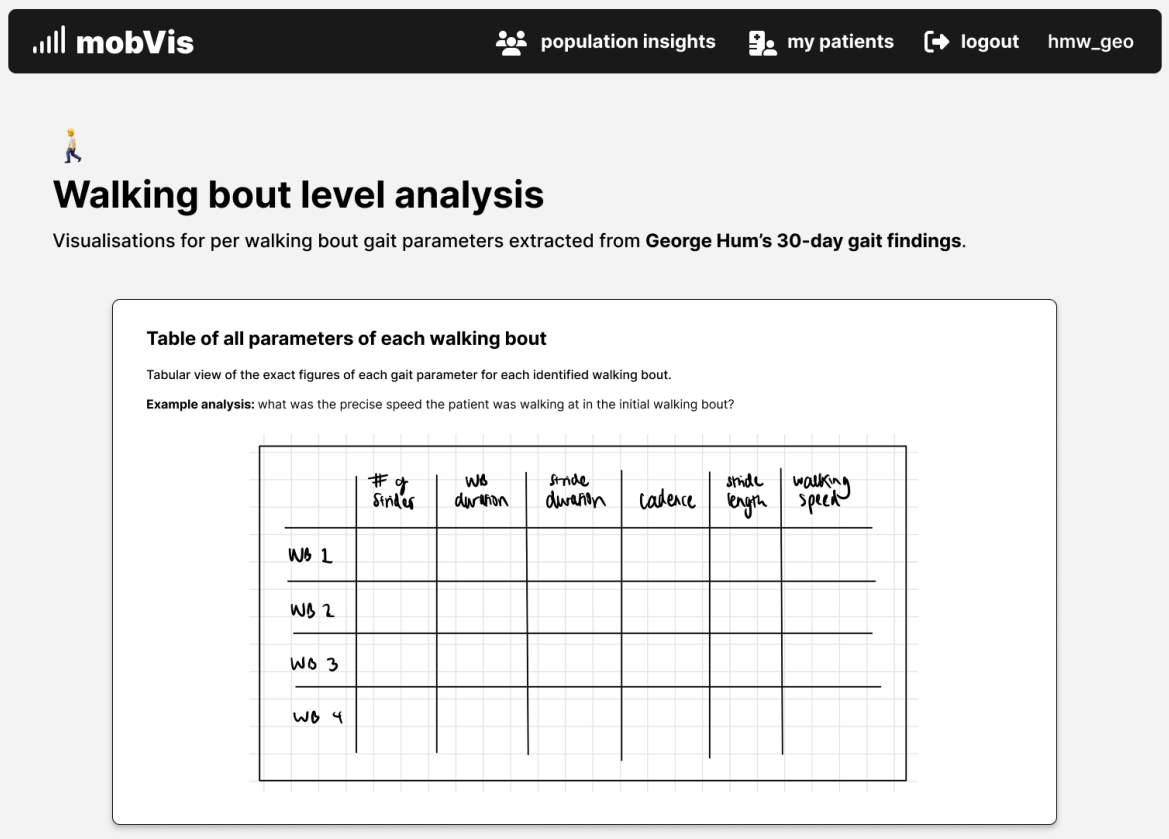


Figure 4.18: Top of the WB level analysis page, with the table of DMO values for each WB

4.3.7.2 Scatter plot and bar chart of DMO progression

These visualisations highlight how a DMO's values changes over time by ordering the WBs chronologically.

A scatter plot and bar chart are used to offer different views of the temporal relationship. The scatter plot can be switched to a step plot by ticking a checkbox.

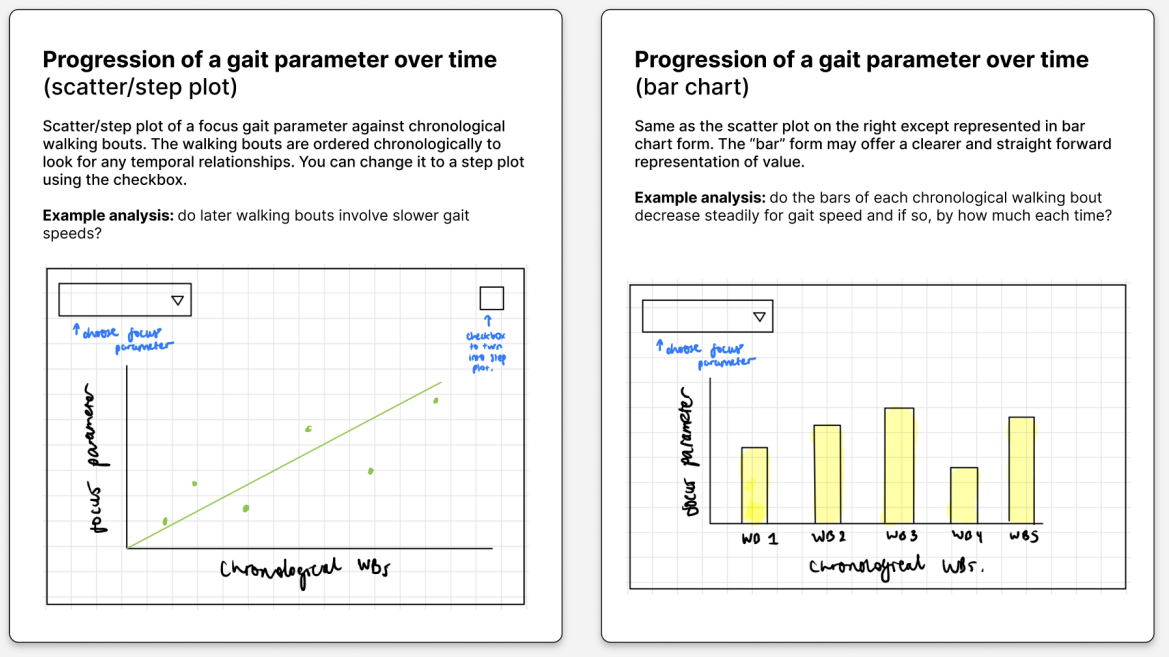


Figure 4.19: Scatter/step plot and bar graph depicting the progression of DMO values per-WB over time

4.3.7.3 Parallel coordinates plot of relationships between all DMOs

Each DMO is plotted as a vertical axis and each WB is a data line connected across its corresponding values on each DMO axis.

Brushing will be implemented, where clinicians can colour to highlight selected data lines. Additionally, hovering on specific data lines will expose more information about the corresponding WB. Axes can also be shifted around assign more focus to the relationship between two specific DMOs.

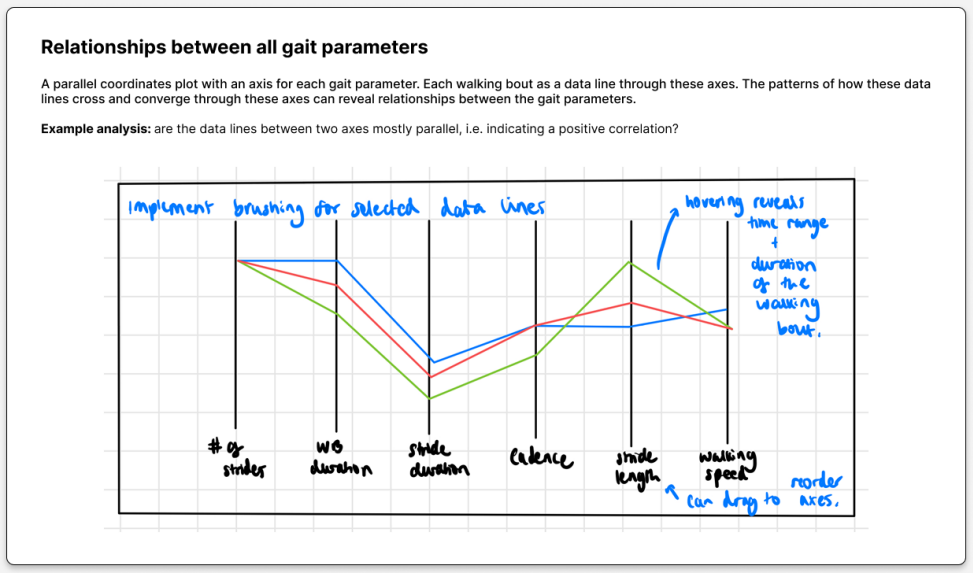


Figure 4.20: Parallel coordinates plot showing the relationships between all the DMOs

4.3.7.4 Scatter plot of the relationship between two DMOs

In contrast with the PCP, this concentrates on the relationship between just two specific DMOs. A trend-line will be plotted, which can be hovered to identify the level of significance of the correlation.

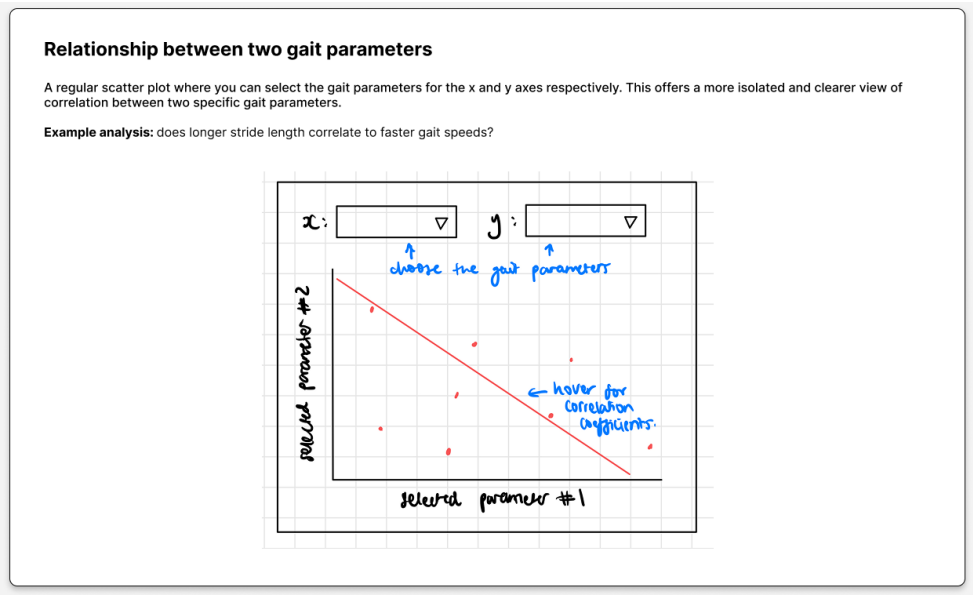


Figure 4.21: Scatter plot with a trend-line between two DMOs to depict their relationship

4.3.7.5 Radar chart for walking-bout comparison

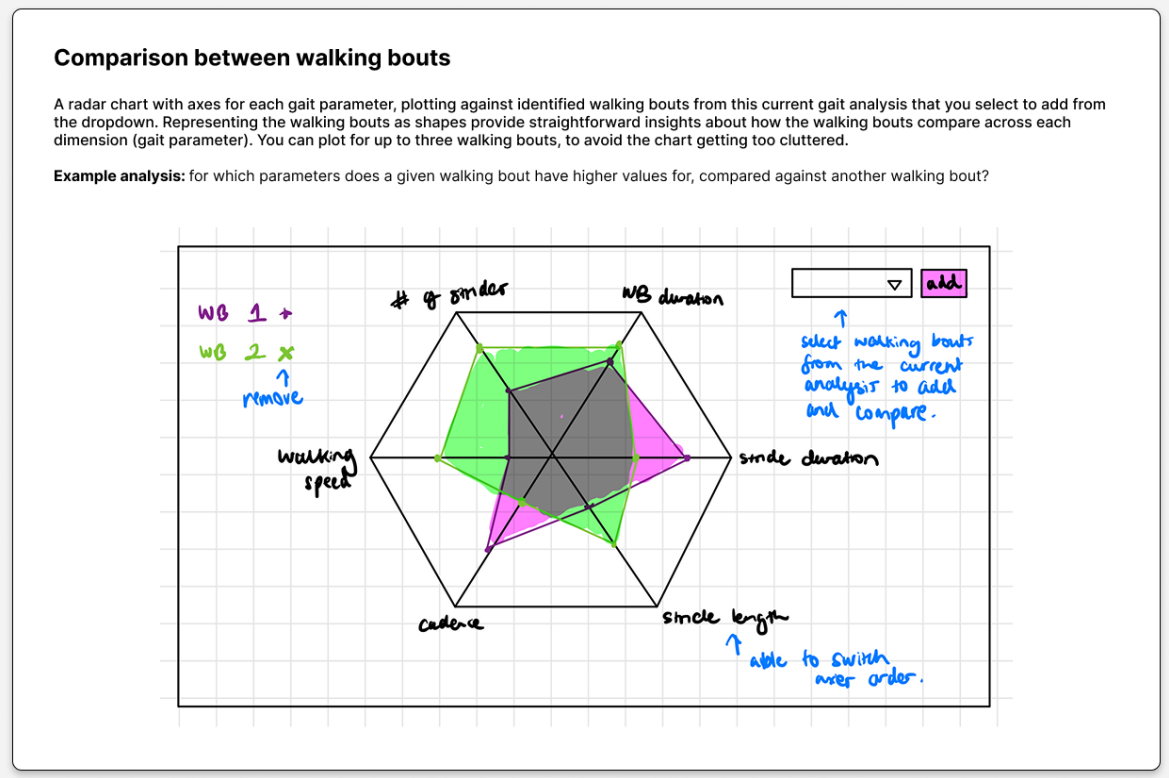


Figure 4.22: Radar chart for comparing the DMO values between different WBs

4.3.8 Stride level analysis

This is the most granular level of analysis available. Intuitively, each WB consists of strides between the left and right leg. DMOs are thus also defined for each stride. Upcoming visualisations will involve per-stride DMOs. Similarly, most visualisations will be reused from the previous two levels, with the main differentiating factor being that DMOs are plotted per-stride of given WBs. As such, repeated descriptions of these will be avoided.

4.3.8.1 Table of DMO values for each stride

In this table, strides are related to their corresponding WB by spanning the WB cell across relevant rows.

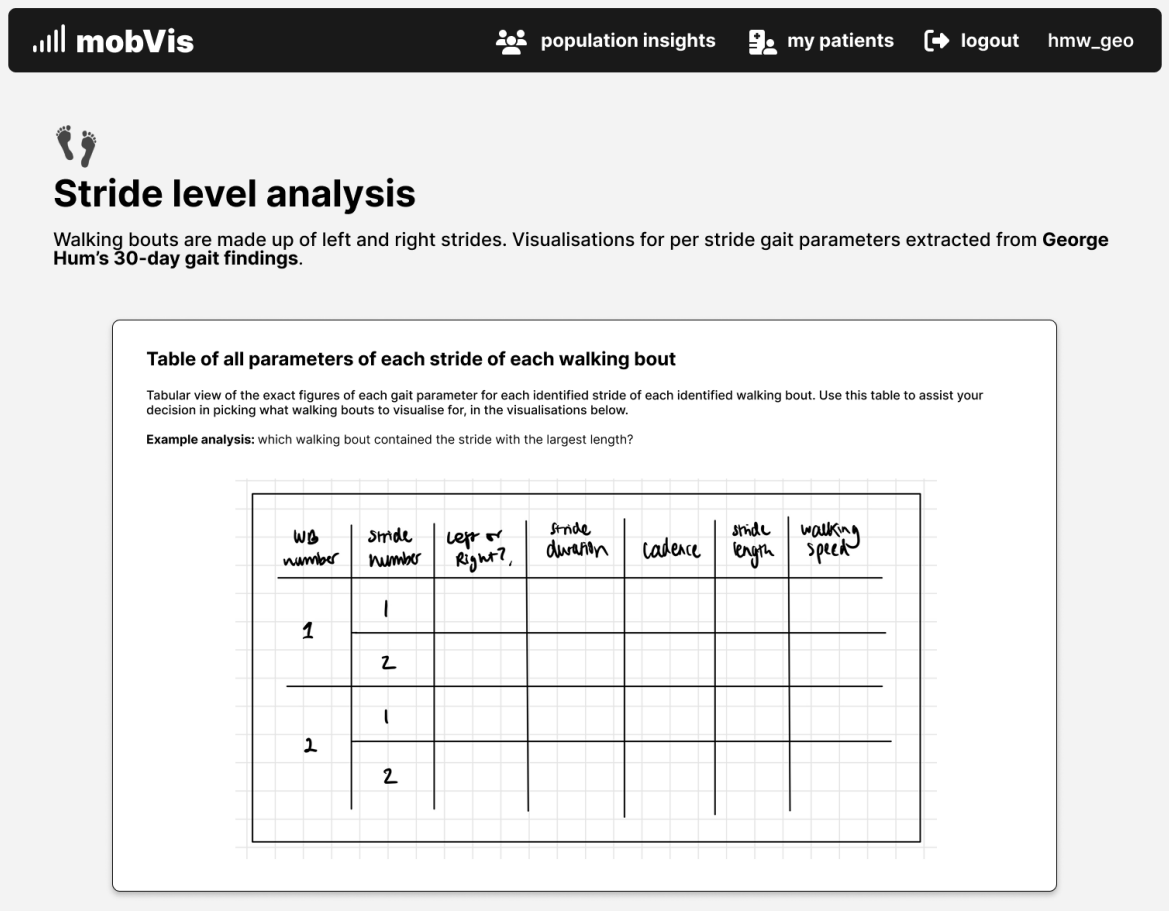


Figure 4.23: Start of the stride level analysis page, with the table of DMO values for each stride linked to each identified corresponding WB

4.3.8.2 Scatter/step plot and bar chart of the progression of a DMO over time

A useful addition to these repeated visualisations is that since strides can be identified as left or right, colour coding is used to differentiate between them. This will help to identify any asymmetry between the left and right leg.

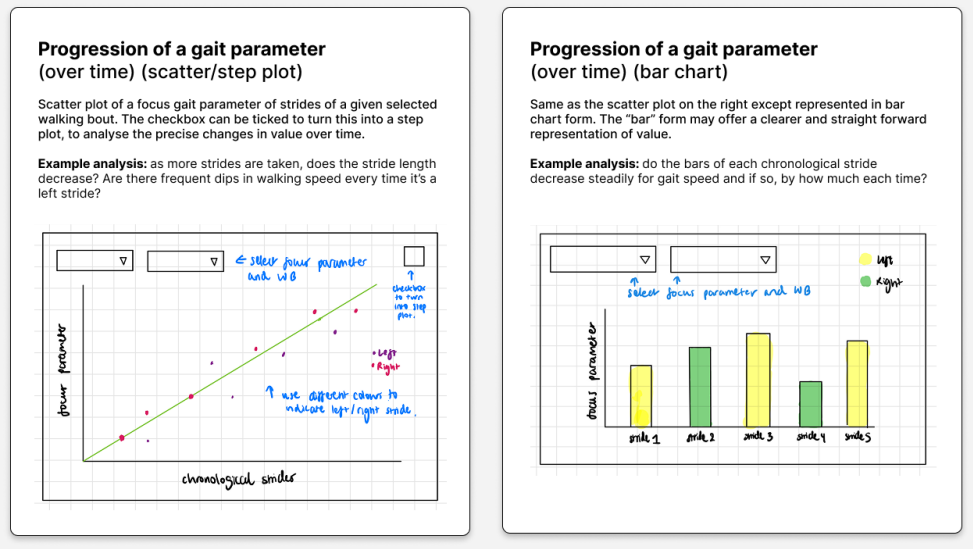


Figure 4.24: Scatter/step plot and bar graph displaying the progression over time of DMO values per-stride of a given WB

4.3.8.3 Violin with box plot of distribution of a DMO

A convenient add-on to previous combined violin and box plots is a checkbox that can be activated to split each WB into two sets of violins/boxes; one for left strides and another for right strides.

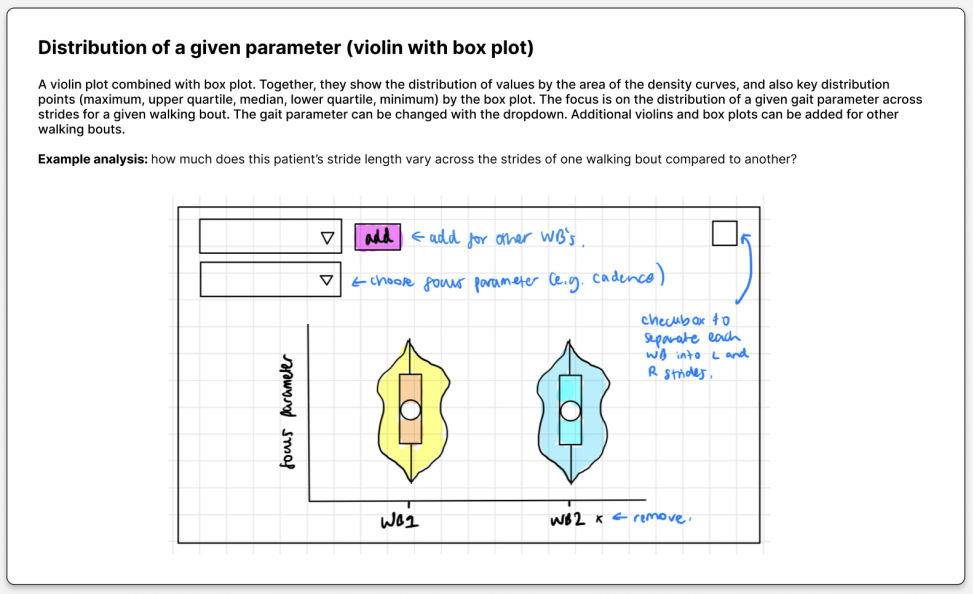


Figure 4.25: Violin combined with box plot showing the distribution of a focus DMO's values for strides of specific WBs

4.3.8.4 Histogram of the distribution of a DMO's values

The user can also split the histogram into left and right strides, but this is restricted to when only one WB is displayed to maintain focus and tidiness in the visualisation.

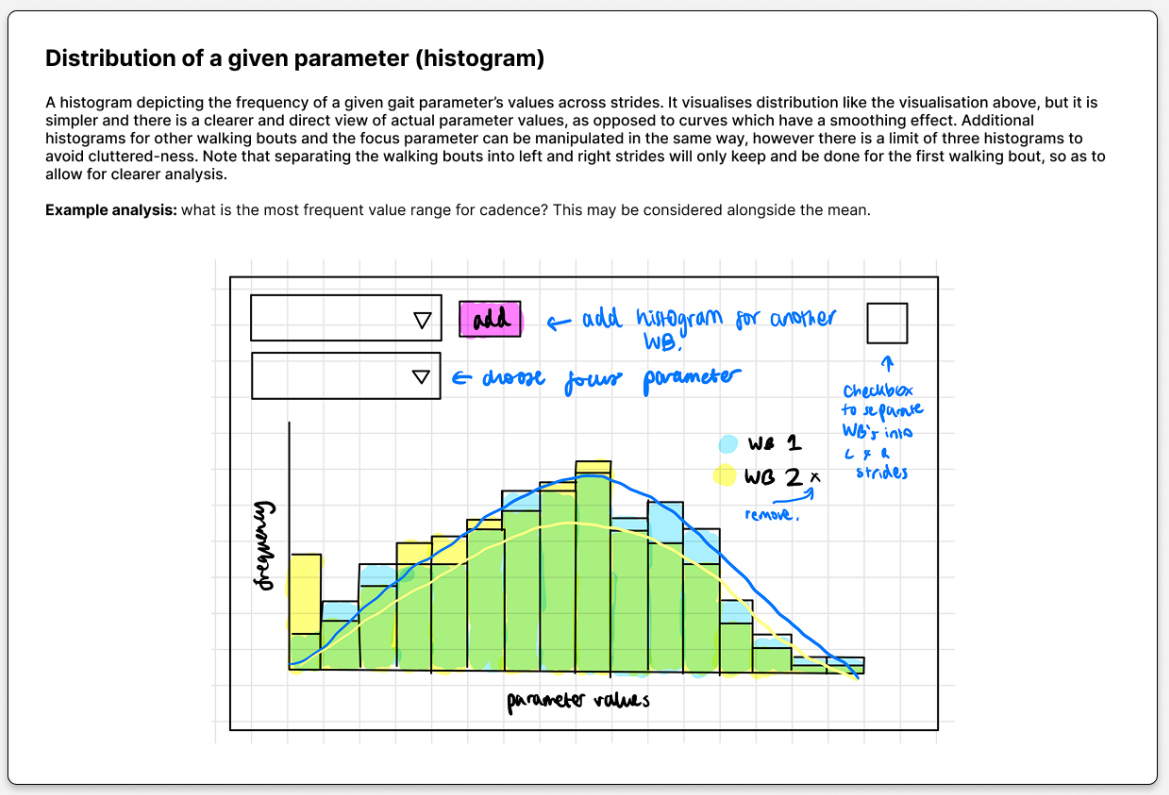


Figure 4.26: Histogram of the distribution of a given DMO's values for strides of selected WBs

4.3.8.5 Parallel coordinates plot of the relationships between all DMOs

As strides are grouped under specific WBs, colour coding is used to differentiate between WBs. This colour coding can be extended to differentiating left and right strides with the checkbox. However, this is also restricted by the condition of one WB being displayed, to avoid overloading the visualisation.

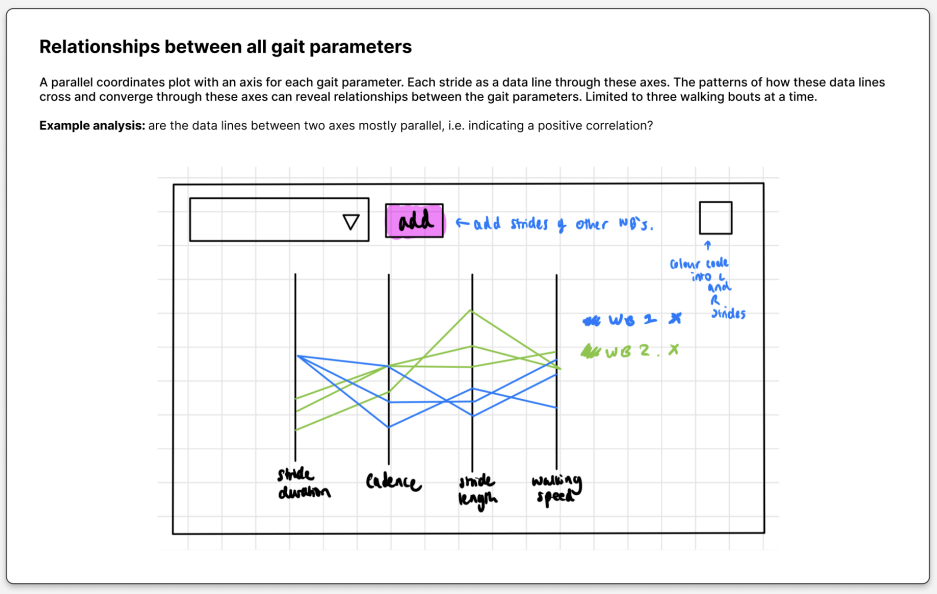


Figure 4.27: Parallel coordinates plot of DMO values for all the strides under selected WBs

4.3.8.6 Radar chart comparing DMO values of selected strides

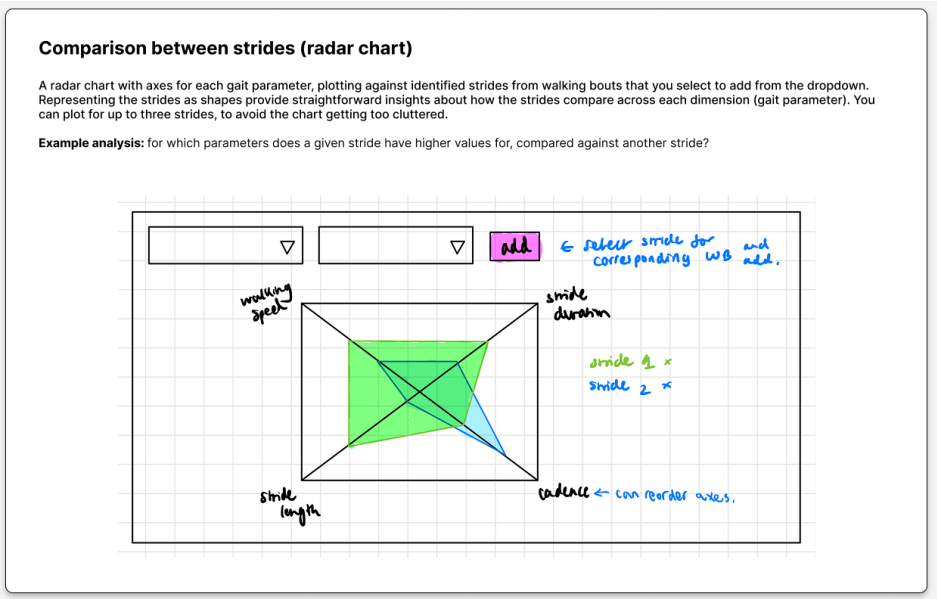


Figure 4.28: Radar chart comparing the values of DMOs of selected strides of different/identical WBs

4.3.8.7 Heatmap comparing DMO values across strides of selected WBs

This is a new visualisation, providing a different view for the comparison between the chronological strides of chosen WBs. The cells are coloured by the magnitude of the focus DMO. The difference in colour shades offer a more straightforward, efficient method of comparison.

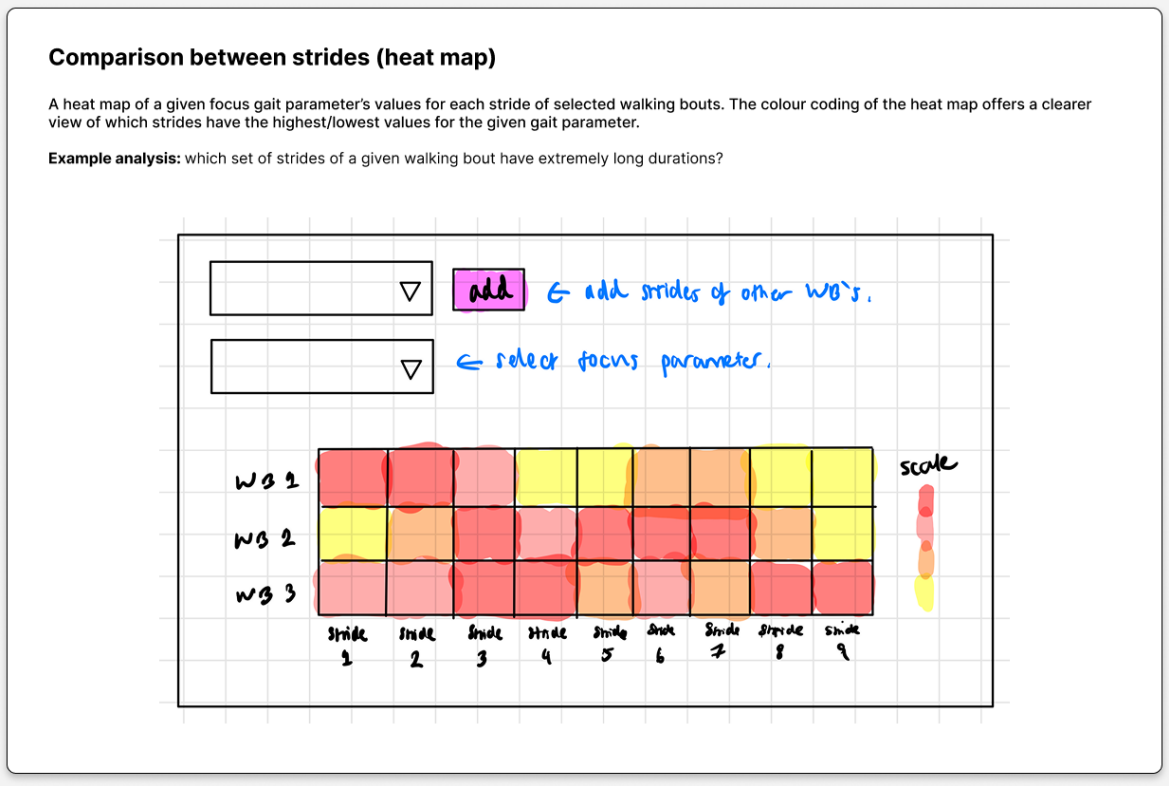


Figure 4.29: Heatmap comparing the values of a DMO under each chronological stride of selected WBs

4.4 Summary

In this chapter, data storage requirements have been identified, followed by the design of required database tables. An overall application flow diagram was also created. Sub-sections of these design diagrams are prioritised for the project's main focus of visualisation and in consideration of time constraints. Then, initial mock-ups of individual pages and visualisations were designed and explained.

Chapter 5

Implementation and Testing

This chapter will describe the final system with screenshots of implemented features, visualisations, and the back-bone code snippets that underpins everything. Additionally, the approach taken to testing will be explained, including automated unit test cases, manual testing, and user evaluation testing.

5.1 Back-end extraction of digital mobility outcomes

5.1.1 Core DMO extraction functions

The first task was to determine how to properly use the Python mobgap library and its pipeline for extracting DMOs from gait assessments of MS patients. All the relevant code for this can be found in `scripts/dmo_extraction/core.py`.

The main function that coordinates the entire process is `extract_dmos`, which calls helper functions, located within the same file, that each performs a mandatory step of the DMO extraction process.

```python
def extract_dmos(file: Union[Path,  SpooledTemporaryFile], sensor_height_m: float, height_m: float, measurement_condition: str,
sampling_rate_hz: int, convertAccFromGToMs: bool = False) -> MobilisedPipelineImpaired:
  # check that measurement condition is one of two types
  is_valid_measurement_condition(measurement_condition)

  print("Beginning DMO extraction process...")

  # load data, convert to m/s^2 if needed
  data = load_csv(file, convertAccFromGToMs)
  print("Data from CSV file has been loaded successfully.")

  # create dataset from dataframe
  dataset = create_dataset_from_dataframe(data, sensor_height_m, height_m, measurement_condition, sampling_rate_hz)
  print ("Gait dataset required for the pipeline has been created successfully.")

  # run the single record through the pipeline
  record = dataset[0]
  pipeline = MobilisedPipelineImpaired().run(record)
  print("Pipeline has been executed successfully.")

  # pipeline object with attributes for gait parameters.
  return pipeline
```

Figure 5.1: `extract_dmos` function

Figure 5.1 shows the code snippet for this function. First, `is_valid_measurement_condition` checks that the parameter `measurement_condition` is valid (i.e. `free_living` or `laboratory`), otherwise it throws an error to the developer. Then, `load_csv` loads in the CSV file and converts the acceleration columns to the correct units if necessary. `create_dataset_from_dataframe` creates a `GaitDataset` object, expected by the pipelines, that stores the single recording data. Then, this is passed into the recommended pipeline for MS to return a pipeline object, that possesses the desired DMOs as attributes.

5.1.2 API for extracting DMOs from front-end

This project is mainly based on Next.js. Typically, Next.js API routes are used to implement back-end API endpoints, which are written in JavaScript. This means that an external Python API endpoint had to be created to manage DMO extraction requests from the Next.js front-end.

```python
@app.post("/api/py/dmo_extraction")
def dmo_extraction(name: Annotated[str, Form()], description: Annotated[str, Form()], samplingRate: Annotated
[int, Form()], sensorHeight: Annotated[float, Form()], patientHeight: Annotated[float, Form()], setting: Annotated
[str, Form()], convertToMs: Annotated[bool, Form()], csvFile: UploadFile):
  # validation handled on the FE, TODO: future work, add validation here.
  try:
    results = extract_dmos(csvFile.file, sensorHeight, patientHeight, setting, samplingRate, convertToMs)

    # check if actual results are empty
    if (results.per_wb_parameters_.empty):
      raise ValueError(general_error_message)

    # use per_wb and per_stride parameters given from the pipeline.
    per_wb_parameters = results.per_wb_parameters_
    per_wb_parameters = per_wb_parameters.drop(columns=["rule_name", "rule_obj"]).replace(np.nan, 0)
    per_wb_parameters["wb_id"] = per_wb_parameters.index

    per_stride_parameters = results.per_stride_parameters_
    per_stride_parameters = per_stride_parameters.drop(columns=["original_gs_id"]).replace(np.nan, 0)
    # add s_id which contains wb_id but also the stride index.
    per_stride_parameters["s_id"] = per_stride_parameters.index.get_level_values("s_id")
    # adjust s_id to only include the stride index, not wb
    per_stride_parameters["s_id"] = per_stride_parameters["s_id"].apply(lambda s_id: int(s_id.split("_")[1]))
    # add corresponding wb_id to easier access strides for a wb
    per_stride_parameters["wb_id"] = per_stride_parameters.index.get_level_values("wb_id")

    # calculate aggregate params from per_wb params
    aggregate_parameters = calculate_aggregate_parameters(per_wb_parameters).replace(np.nan, 0)

    # 0 is an okay alternative for NaN because the parameters should be non-zero when calculated.

    response = {
      "total_walking_duration": results.aggregated_parameters_.loc["all_wbs","total_walking_duration_h"],
      "per_wb_parameters": per_wb_parameters.to_dict(orient="records"),
      "per_stride_parameters": per_stride_parameters.to_dict(orient="records"),
      "aggregate_parameters": aggregate_parameters.to_dict(orient="records")
    }

    return response
```

Figure 5.2: DMO extraction in FastAPI Python endpoint (a)

```python
  except Exception as e:
    # handle common interpolation error involving invalid sampling rate.
    if ("x_new is below the interpolation range's minimum value" in str(e)):
      e = ValueError("The sampling rate may be too high.")

    csvFile.file.close()
    raise HTTPException(status_code=400, detail=str(e))
  finally:
    csvFile.file.close()
```

Figure 5.3: Sending errors to the front-end from the FastAPI Python endpoint (b)

Figure 5.2 features the main code in the endpoint for extracting the DMOs. It first calls the `extract_dmos` method from sub-section 5.1.1 and stores the returned pipeline object, which contains the per-WB, per-stride and predefined aggregate DMOs. The per-WB and per-stride DMOs are taken from the pipeline, but modified to remove unnecessary columns, add columns for efficient indexing, and change `NaN` values to 0 (for JSON storage). Custom aggregate parameters are calculated using `calculate_aggregate_parameters` in `core.py`,

which calculates the maximums, minimums, averages and variances for all DMOs across the per-WB level. Total walking duration is also extracted from the pipelines and stored. Finally, the data is stored in JSON using LocalStorage in the user's browser (see figures 5.4 - 5.6).

Figure 5.3 shows the code for sending to the front-end, any sort of errors raised by the pipeline or preparatory operations. Notably, after manual testing of edge cases, there was a recurring error that was the result of an excessively high sampling rate, so a clearer message is passed to the front-end accordingly.

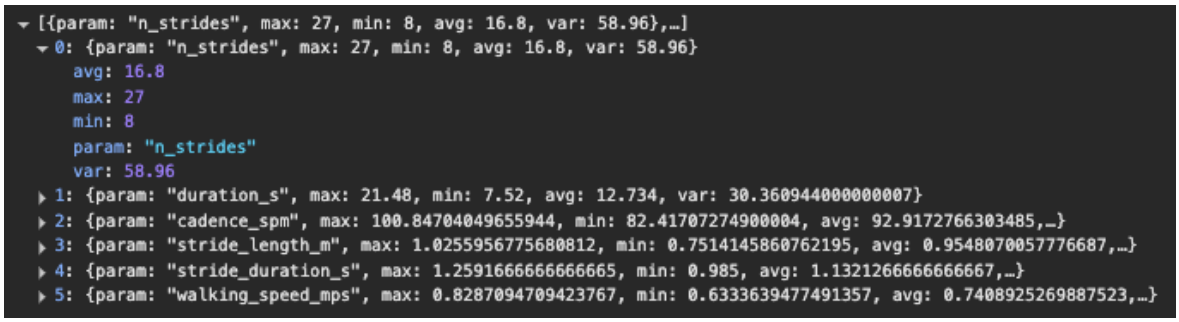


Figure 5.4: Example JSON storage of aggregate parameters

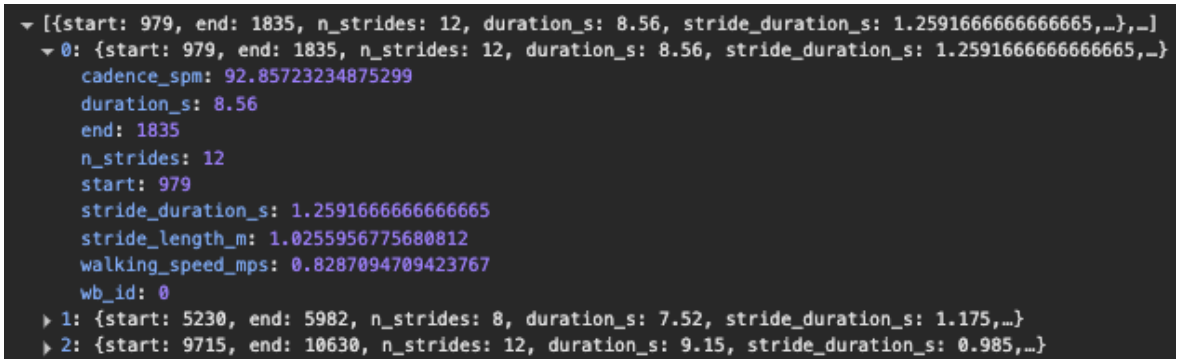


Figure 5.5: Example JSON storage of per-WB parameters

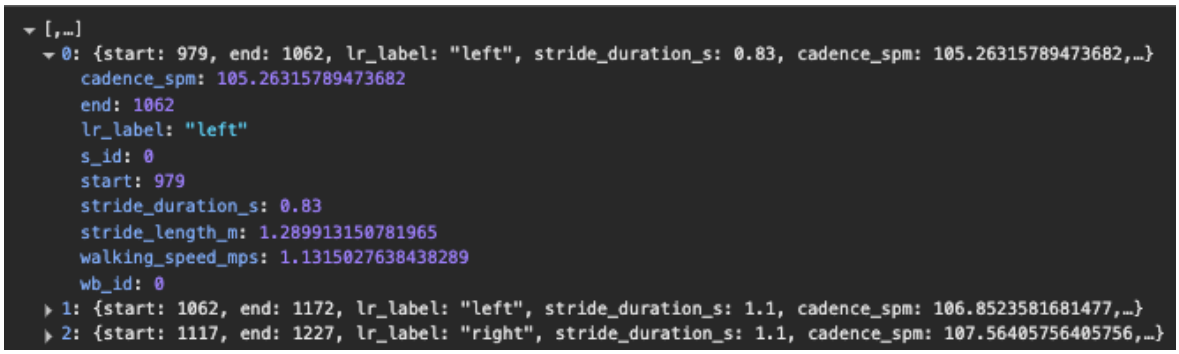


Figure 5.6: Example JSON storage of per-stride parameters

These are then accessible on all pages of the system, and used for visualisation.

5.2 Implemented features

Due to time constraints, only the requirements that were labeled must-have (most relevant to the focus of visualisation) were completed.

5.2.1 Navigation

The application spans across only a few pages. The "new" link sends the clinician to the home page with the inputs form (subsection 5.2.2). Then, there are links for each analysis level, with simplified naming changes (e.g. "aggregate" to "summary") as requested by clinician feedback.

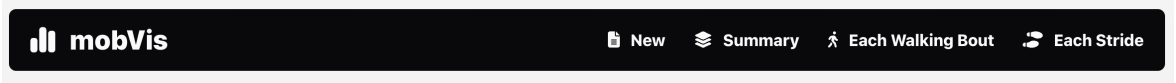


Figure 5.7: Top navigation menu bar

5.2.2 Home page with main form

On launch, the clinician is first greeted with a page (figure 5.8) consisting of useful overview information about what the system is about, who the target users are, what sort of inputs are applicable, how to extract DMOs and how that has been implemented in the back-end. These are imperative in providing the clinician with sufficient, initial understanding on how to use the system.

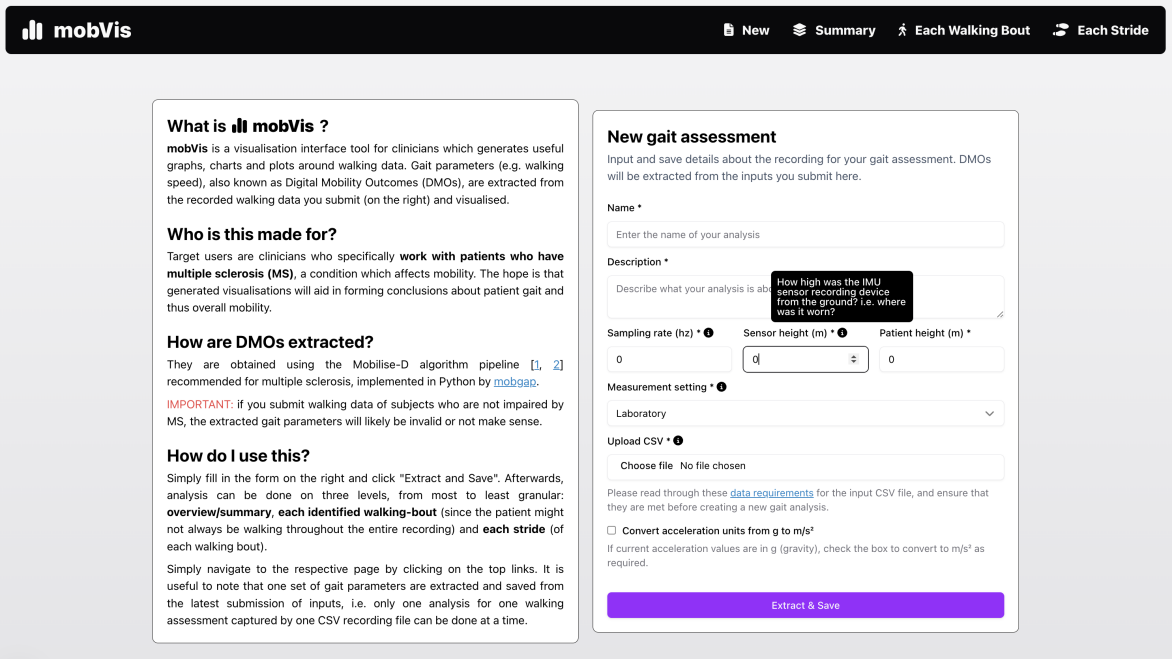


Figure 5.8: Home page with information on how to use the system and the main form for submitting input data

On the right of the background information card, there exists another card housing the main form for submitting essential input data for the current gait assessment. Useful information tool tips have been added to provide a better understanding of the expectations of specific fields.

It is useful to note the last paragraph of the background information. Visualisations are created for the latest set of inputs that are submitted through this form. The storage, comparison between and visualisation of multiple gait analyses could not be implemented due to time constraints.

5.2.2.1 CSV data requirements dialog

This dialog window (figure 5.9) appears after clicking on the blue "data requirements" hyperlink in figure 5.8. It details all the expectations for the uploaded CSV file, while providing an actual downloadable sample file for clarification.

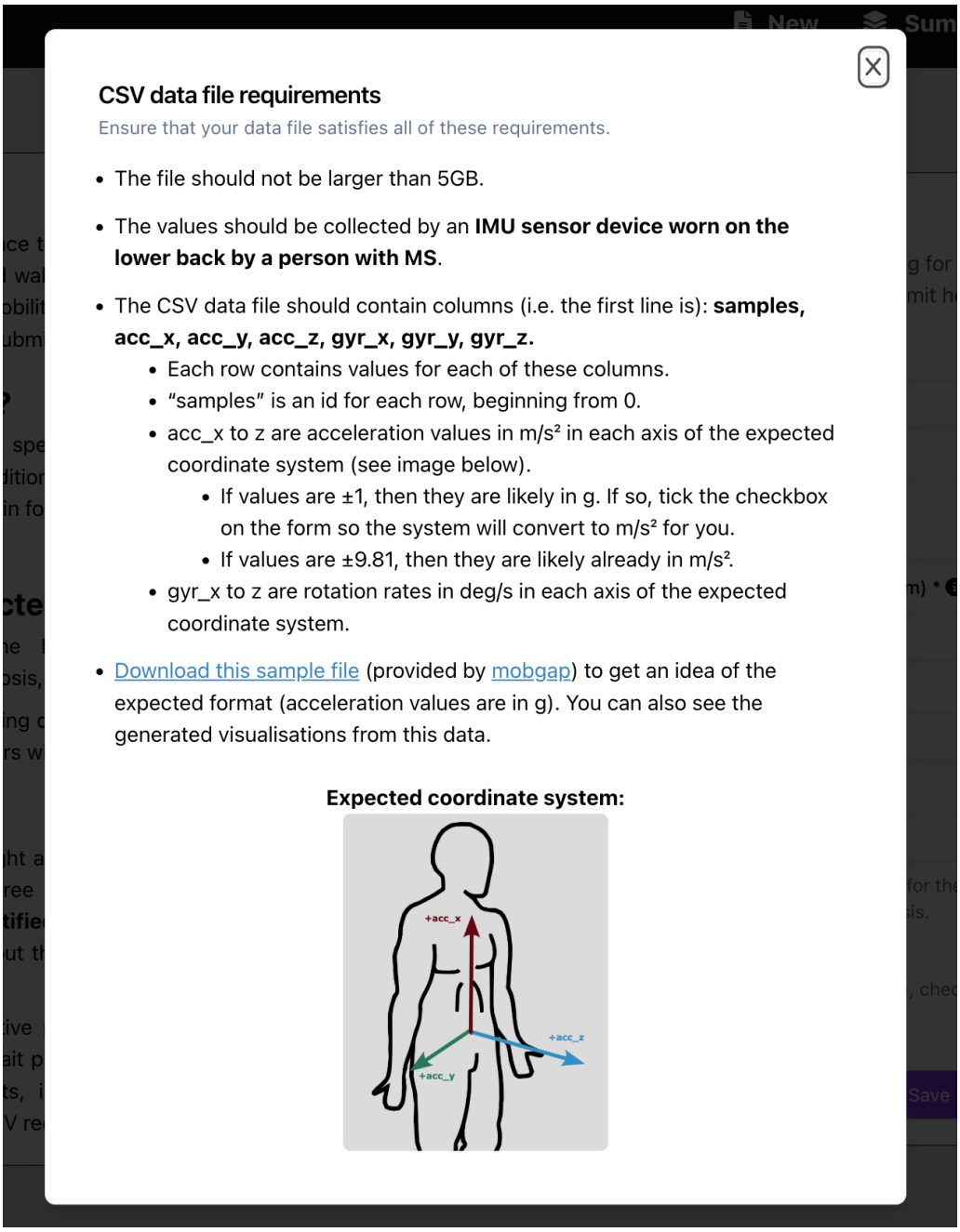


Figure 5.9: Dialog window outlining CSV data requirements

5.2.2.2 Front-end form validation and status messages

Non-functional requirement 7 was fulfilled through standard validation error messages on the form (e.g. presence and threshold checks) (figure 5.10) and form submission status messages (figure 5.11).

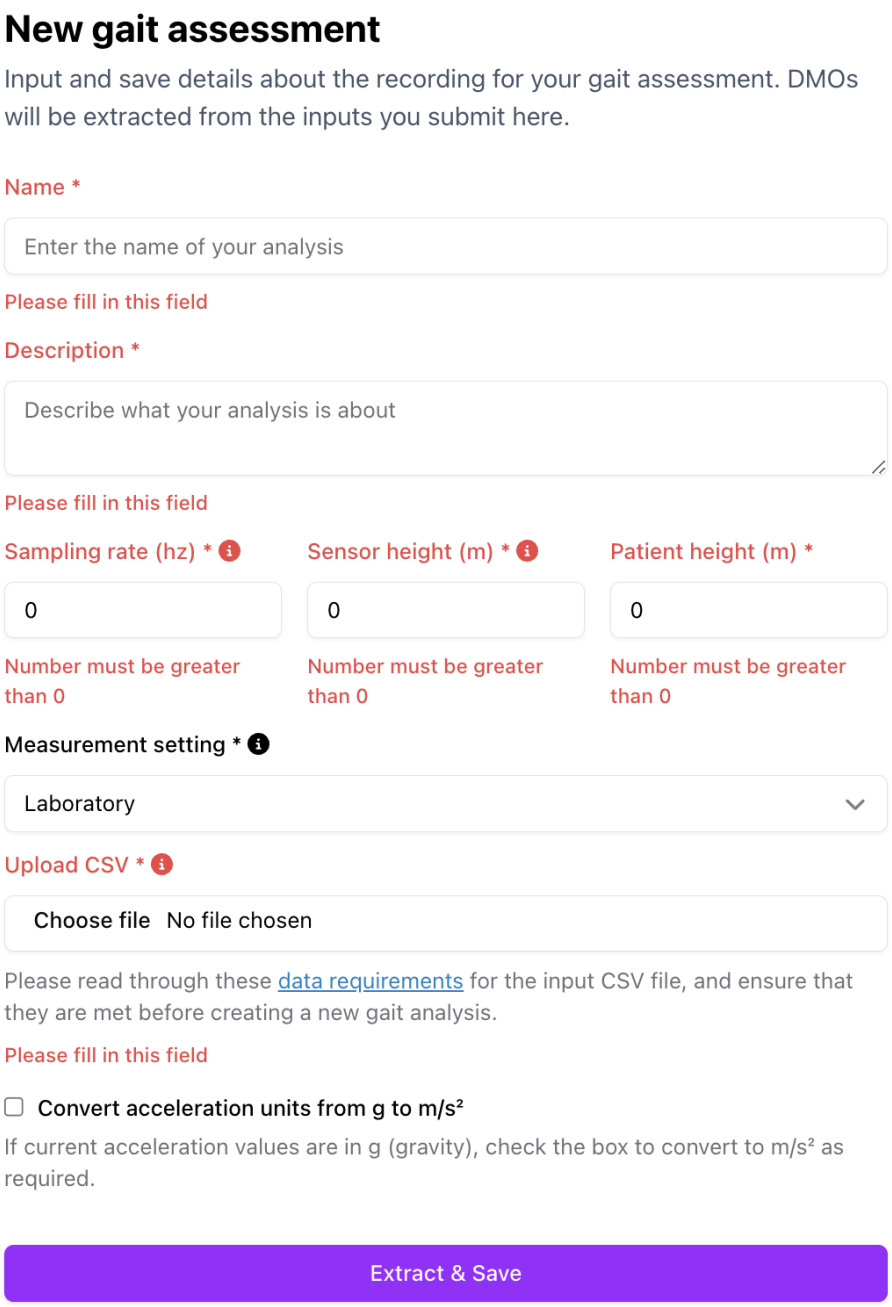


Figure 5.10: Front-end form validation error messages for presence and invalid value checks

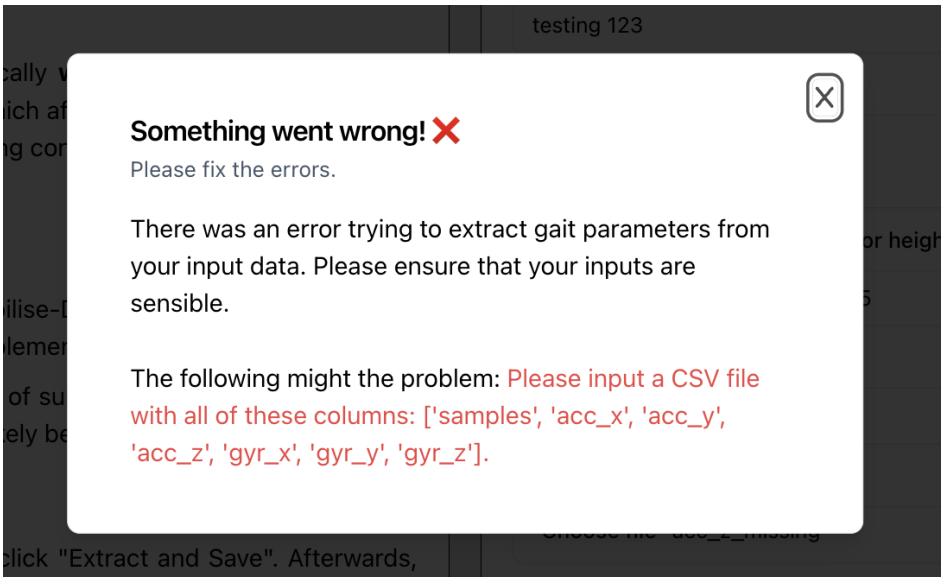


Figure 5.11: Error message dialog on trying to submit CSV with missing acc_z column

5.2.3 Current inputs dialog

At the top of each analysis page, there is a hyperlink which displays a dialog on click, with the current set of values that were submitted for each field.

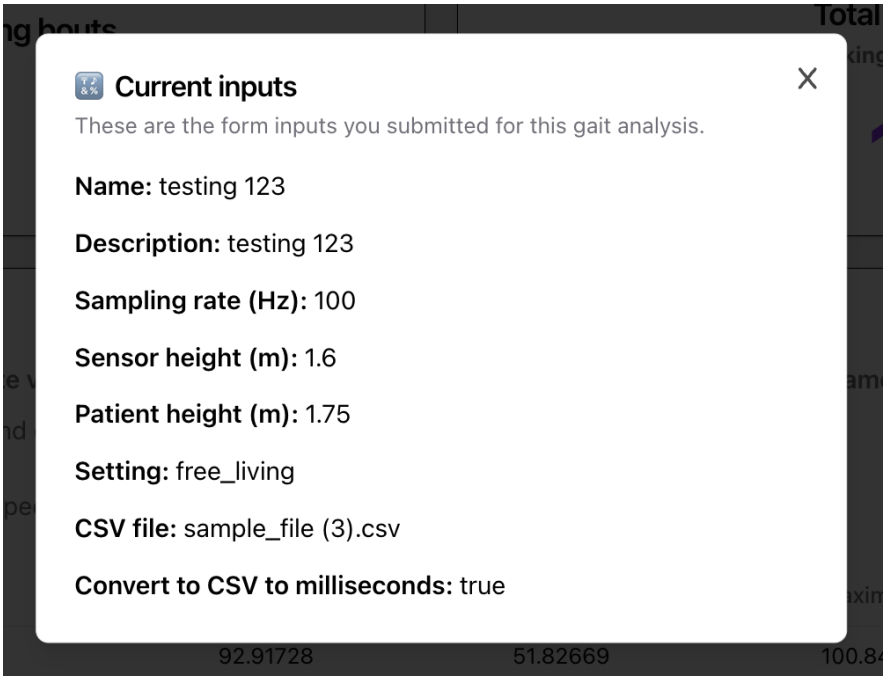


Figure 5.12: Dialog with details on all submitted current inputs

5.2.4 Visualisation card layout

To assist busy clinicians with limited data analytical knowledge, a consistent informational layout is used in each visualisation. The first line is always the title. This is followed by a bolded description of the purpose and basics of the visualisation type. Following un-bolded lines describe possible interactive functions. Finally, there is a line of an example analytical conclusion that can be reached with that visualisation. An example of this can be seen in figure 5.14.

5.2.5 Summary level analysis

Only a few of the planned visualisations were implemented, as most of them required multiple comparable gait analyses to be useful. Therefore, a few additional visualisations were created in place of these.

5.2.5.1 Stat cards

Simple stat cards from the mock-ups are shown at the top. One for the integral count of detected walking bouts. Another for the total walking duration summed across all walking bouts, rounded to the nearest appropriate time unit.

Figure 5.13: Stat cards for total detected walking bouts and duration

5.2.5.2 Table of aggregate parameters

As described by its title, it's a simple table with concrete values for each aggregation metric (mean, variance, maximum and minimum). Like all tables in the system, the rows are re-orderable through dragging. This is to allow for more direct analysis between specific rows/gait parameters.

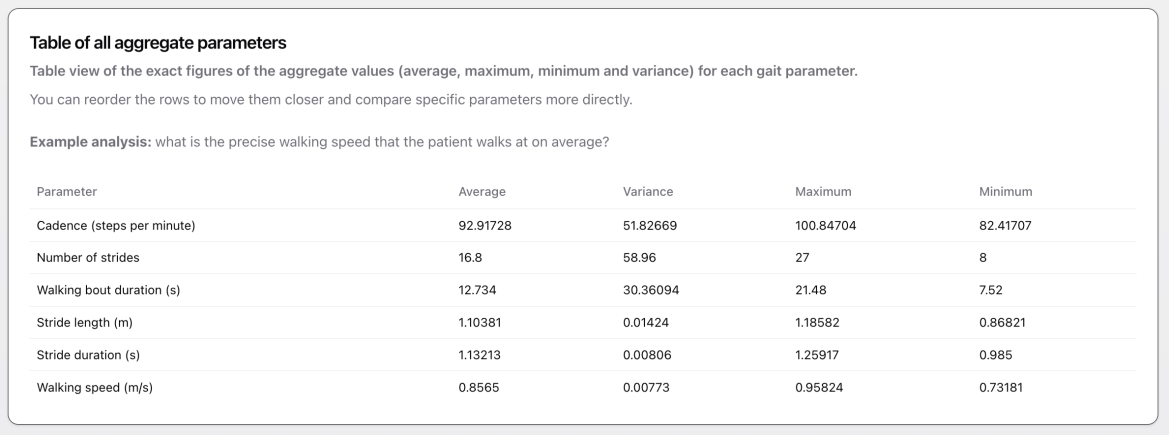


Table of all aggregate parameters

Table view of the exact figures of the aggregate values (average, maximum, minimum and variance) for each gait parameter.

You can reorder the rows to move them closer and compare specific parameters more directly.

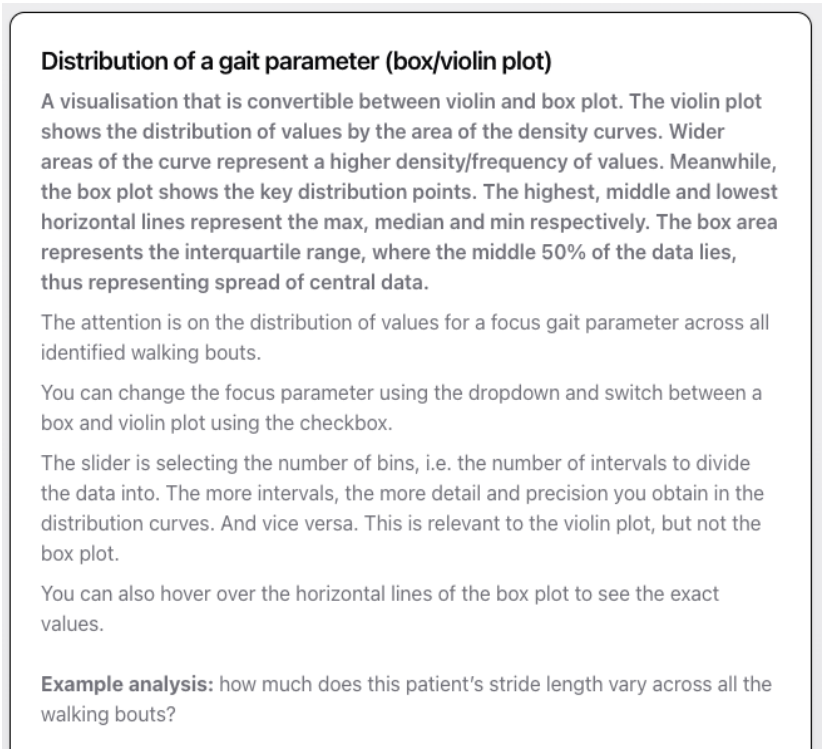
Example analysis: what is the precise walking speed that the patient walks at on average?

Parameter	Average	Variance	Maximum	Minimum
Cadence (steps per minute)	92.91728	51.82669	100.84704	82.41707
Number of strides	16.8	58.96	27	8
Walking bout duration (s)	12.734	30.36094	21.48	7.52
Stride length (m)	1.10381	0.01424	1.18582	0.86821
Stride duration (s)	1.13213	0.00806	1.25917	0.985
Walking speed (m/s)	0.8565	0.00773	0.95824	0.73181

Figure 5.14: Table of averages, variances, maximums and minimums of each gait parameter across all WBs

5.2.5.3 Distribution of a gait parameter

Different views are offered of the distribution of a focus gait parameter's values across all WBs.



Distribution of a gait parameter (box/violin plot)

A visualisation that is convertible between violin and box plot. The violin plot shows the distribution of values by the area of the density curves. Wider areas of the curve represent a higher density/frequency of values. Meanwhile, the box plot shows the key distribution points. The highest, middle and lowest horizontal lines represent the max, median and min respectively. The box area represents the interquartile range, where the middle 50% of the data lies, thus representing spread of central data.

The attention is on the distribution of values for a focus gait parameter across all identified walking bouts.

You can change the focus parameter using the dropdown and switch between a box and violin plot using the checkbox.

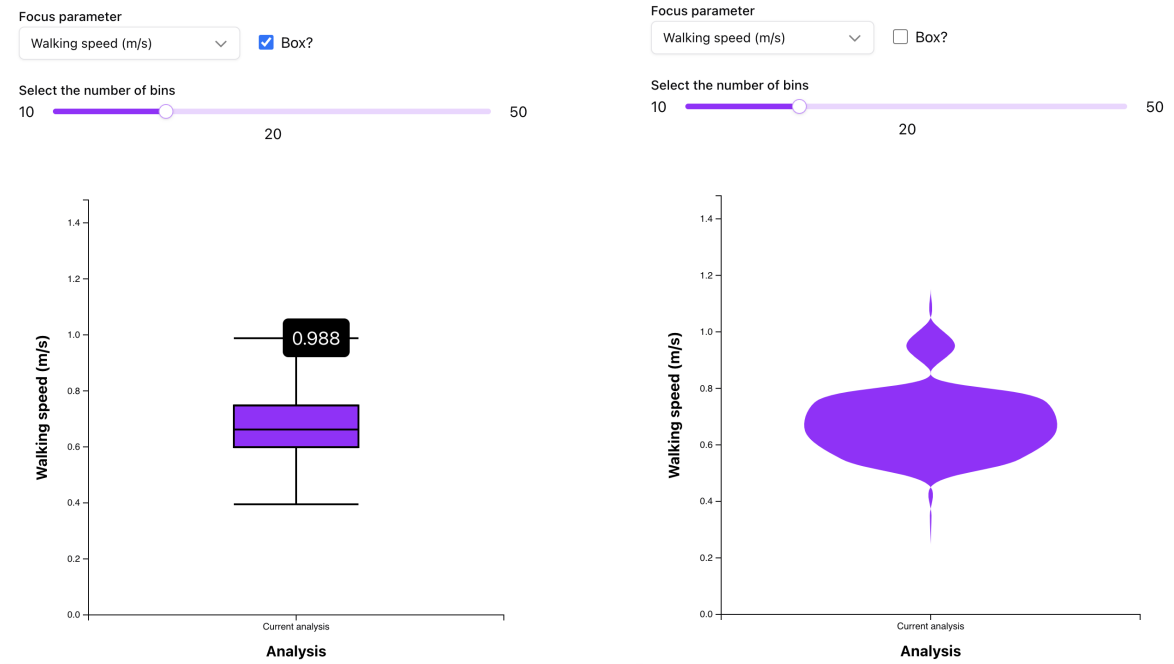
The slider is selecting the number of bins, i.e. the number of intervals to divide the data into. The more intervals, the more detail and precision you obtain in the distribution curves. And vice versa. This is relevant to the violin plot, but not the box plot.

You can also hover over the horizontal lines of the box plot to see the exact values.

Example analysis: how much does this patient's stride length vary across all the walking bouts?

Figure 5.15: Description of box/violin distribution plot

Figures 5.15 and 5.16 show a plot that is interchangeable between a box and violin plot. These were not combined within each other (as in the mock-ups), so as to prevent congestion. Key lines (e.g. maximum) are hover-able to view exact values.



(a) Box plot

(b) Violin plot

Figure 5.16: Combined violin and box plot to show a focus gait parameter's distribution across all WBs

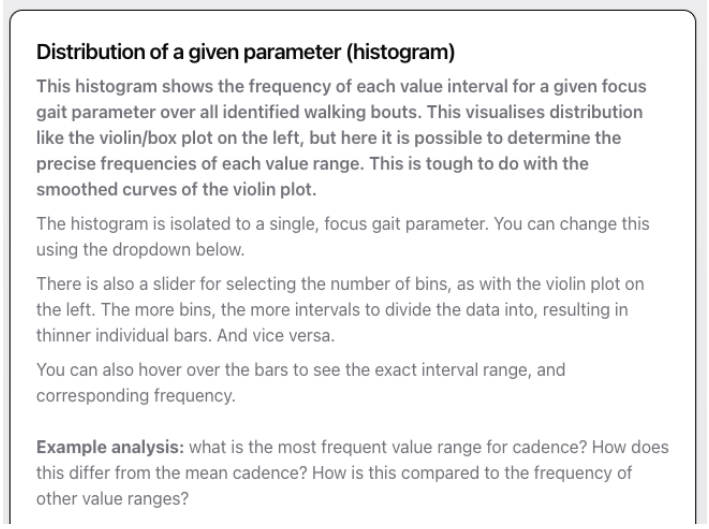


Figure 5.17: Description of histogram distribution plot

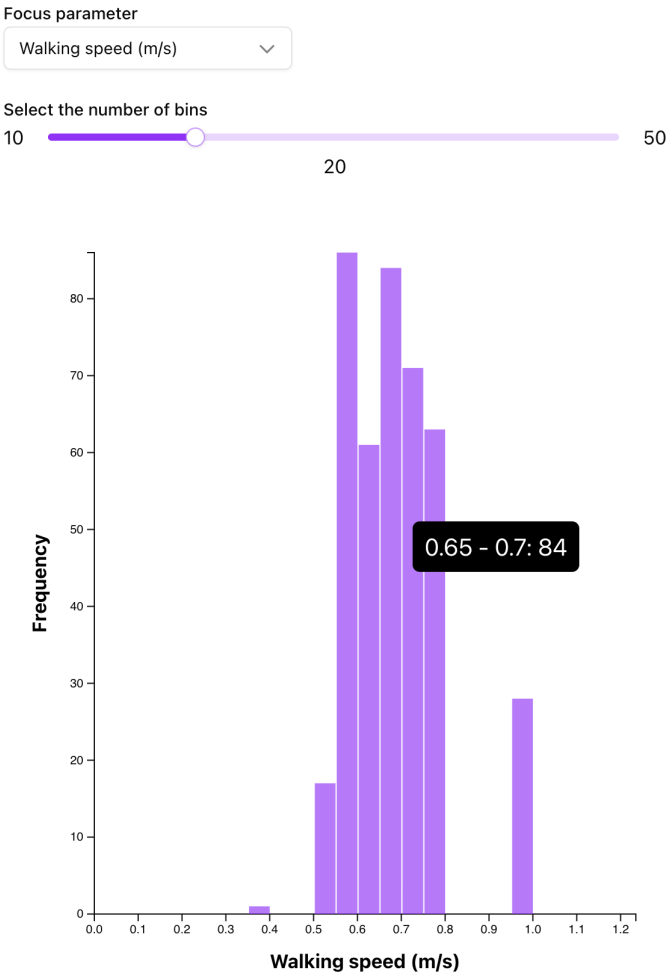


Figure 5.18: Histogram to show a focus gait parameter's distribution over all WBs

Figures 5.17 and 5.18 make up the histogram view of distribution. Clinicians can see the exact frequencies of interval ranges, which become visible on hover of individual bars.

Both distribution plots can be manipulated by their focus parameter and the number of bins to separate the interval ranges into.

5.2.6 Walking bout level analysis

These are all the same visualisations from the mockups, with some modifications that were found to be more appropriate during implementation. As descriptions can be found in subsection 4.3 of mock-ups, only additional features or alterations will be explained.

5.2.6.1 Table of all gait parameters under each individual walking bout

Clinicians can sort the records by individual columns by clicking on the button beside the headers. Additionally, the records can be divided into groups of a preferred size using the number field, and navigation between groups is available at the bottom.

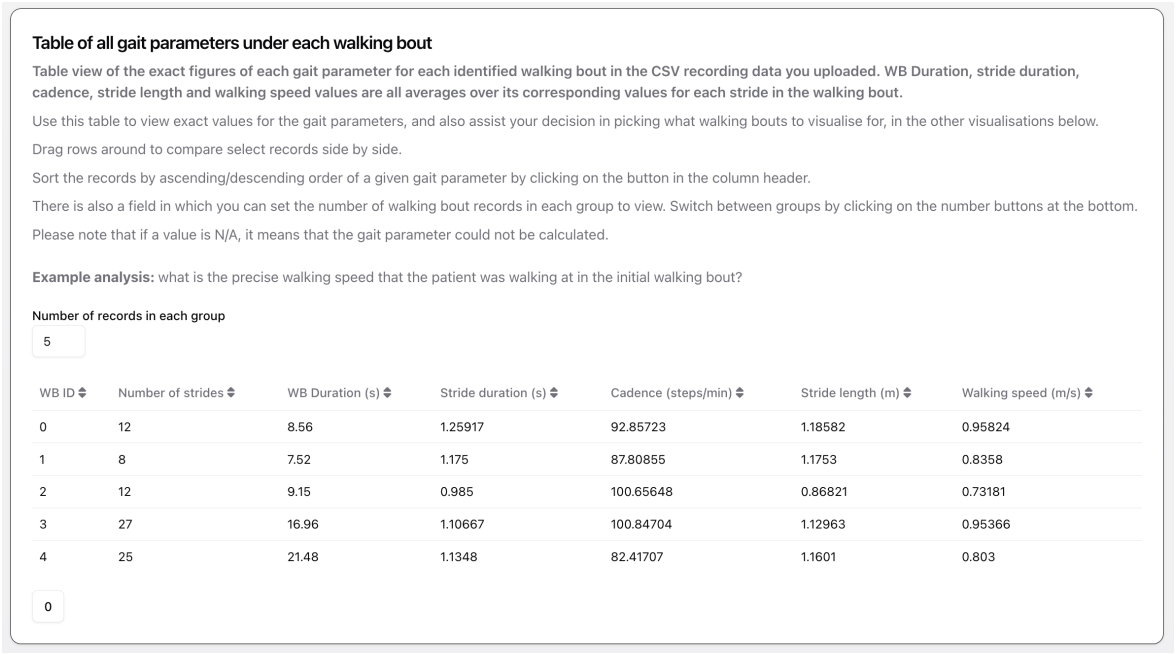


Table of all gait parameters under each walking bout

Table view of the exact figures of each gait parameter for each identified walking bout in the CSV recording data you uploaded. WB Duration, stride duration, cadence, stride length and walking speed values are all averages over its corresponding values for each stride in the walking bout.

Use this table to view exact values for the gait parameters, and also assist your decision in picking what walking bouts to visualise for, in the other visualisations below.

Drag rows around to compare select records side by side.

Sort the records by ascending/descending order of a given gait parameter by clicking on the button in the column header.

There is also a field in which you can set the number of walking bout records in each group to view. Switch between groups by clicking on the number buttons at the bottom.

Please note that if a value is N/A, it means that the gait parameter could not be calculated.

Example analysis: what is the precise walking speed that the patient was walking at in the initial walking bout?

Number of records in each group

5

WB ID	Number of strides	WB Duration (s)	Stride duration (s)	Cadence (steps/min)	Stride length (m)	Walking speed (m/s)
0	12	8.56	1.25917	92.85723	1.18582	0.95824
1	8	7.52	1.175	87.80855	1.1753	0.8358
2	12	9.15	0.985	100.65648	0.86821	0.73181
3	27	16.96	1.10667	100.84704	1.12963	0.95366
4	25	21.48	1.1348	82.41707	1.1601	0.803

0

Figure 5.19: Table of gait parameter values for each WB

5.2.6.2 Progression of a gait parameter over time

A connected scatter/step plot (alternate using a checkbox) and bar chart are used to determine how values of a certain gait parameter evolve over time. Each data point is hover-able to display exact values.

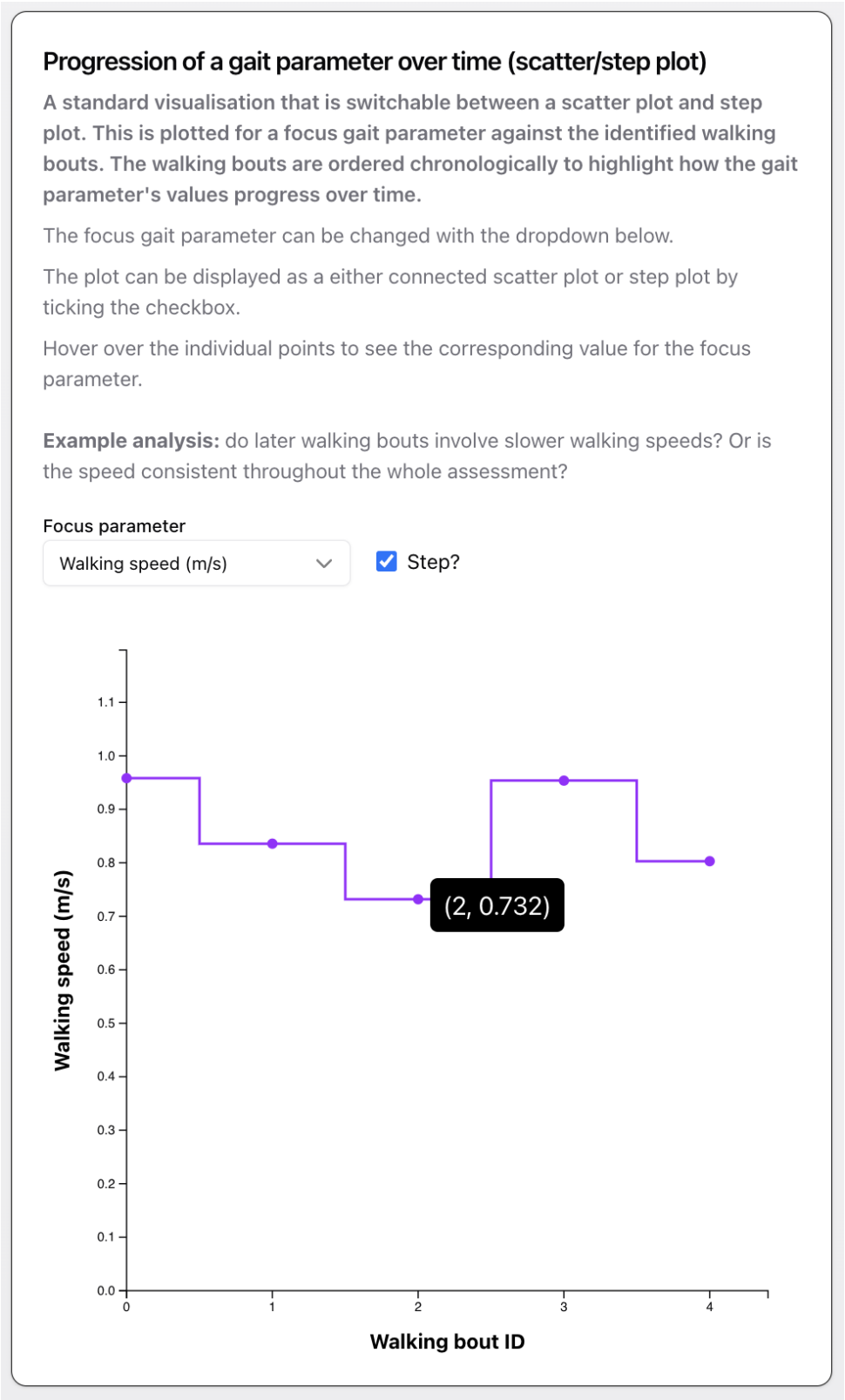


Figure 5.20: An alternating step and connected scatter plot to show value progression of a gait parameter over chronological WBs

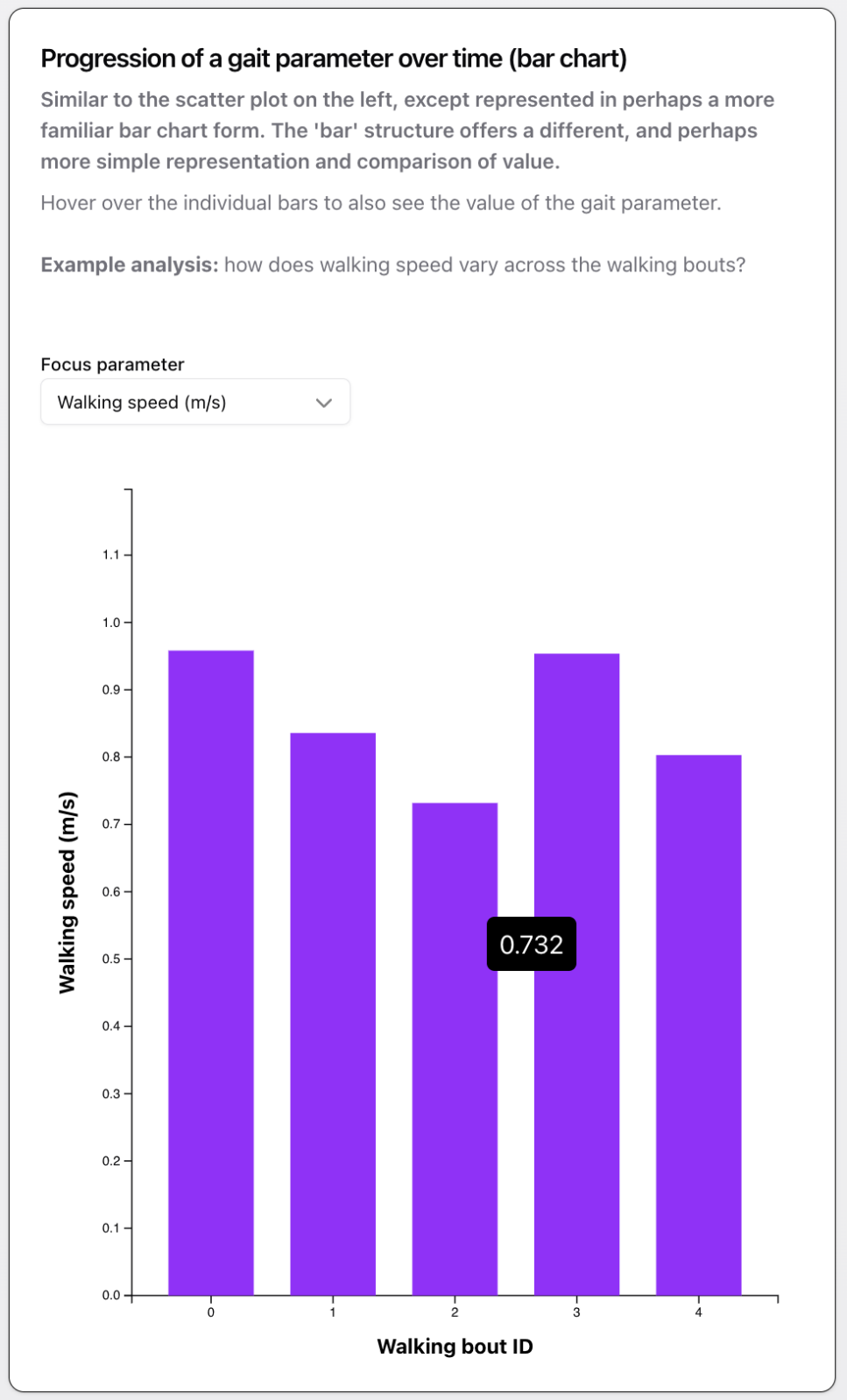


Figure 5.21: Bar chart showing the value progression of a gait parameter over chronological WBs

5.2.6.3 Relationship between gait parameters

A PCP and scatter plot are used to expose relationships between gait parameters.

The PCP axes can be shifted using drop-downs, selecting the axis and new position. Data lines can be brushed with a selected colour, and hovered to reveal the corresponding WB ID.

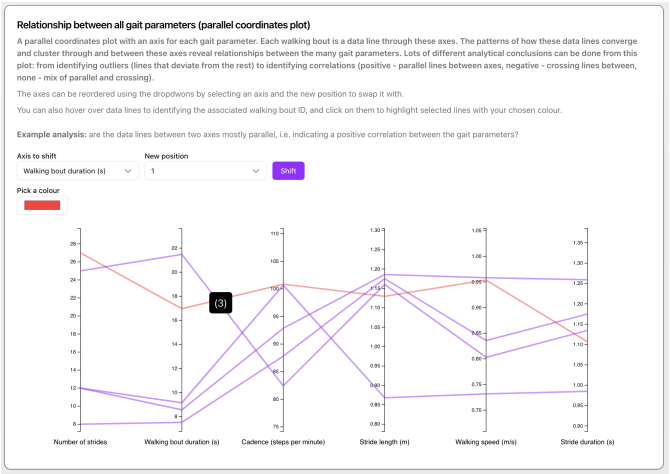


Figure 5.22: Parallel coordinates plot with data lines for each WB

There is a trend-line on the scatter plot plotted using the least squares regression formula. Hovering over it shows the Pearson correlation coefficient, representing the significance of the correlation.

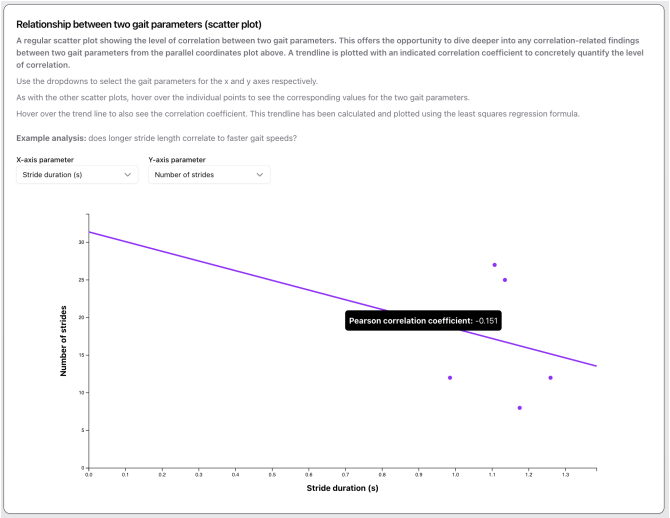


Figure 5.23: Scatter plot of the relationship between two selected parameters with its significance represented by line of best fit (uses least squares regression) and Pearson correlation coefficient

5.2.6.4 Comparison between walking bouts

The axes order in the radar chart is manipulated using drop-downs, due to the difficulty of implementing dragging behaviour.

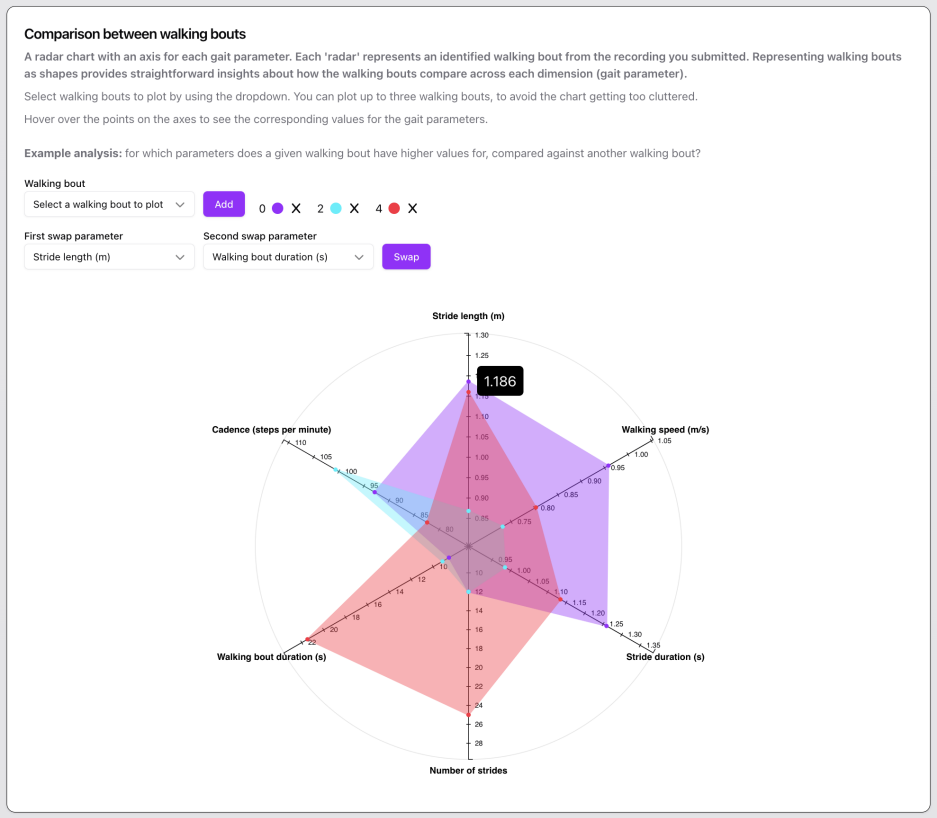


Figure 5.24: Radar chart comparing different WBs

5.2.7 Stride level analysis

Most visualisations are also repeated from previous analysis levels, and also closely follow the mock-ups. Therefore, only new implementation details will be described.

5.2.7.1 Table of gait parameter values

The clinician can choose between WBs to display the strides for using the arrows. The plan was initially to display all strides of all WBs, and relate the strides to WBs with a row-spanning column. However, this was difficult to achieve alongside choosing how many records to display.

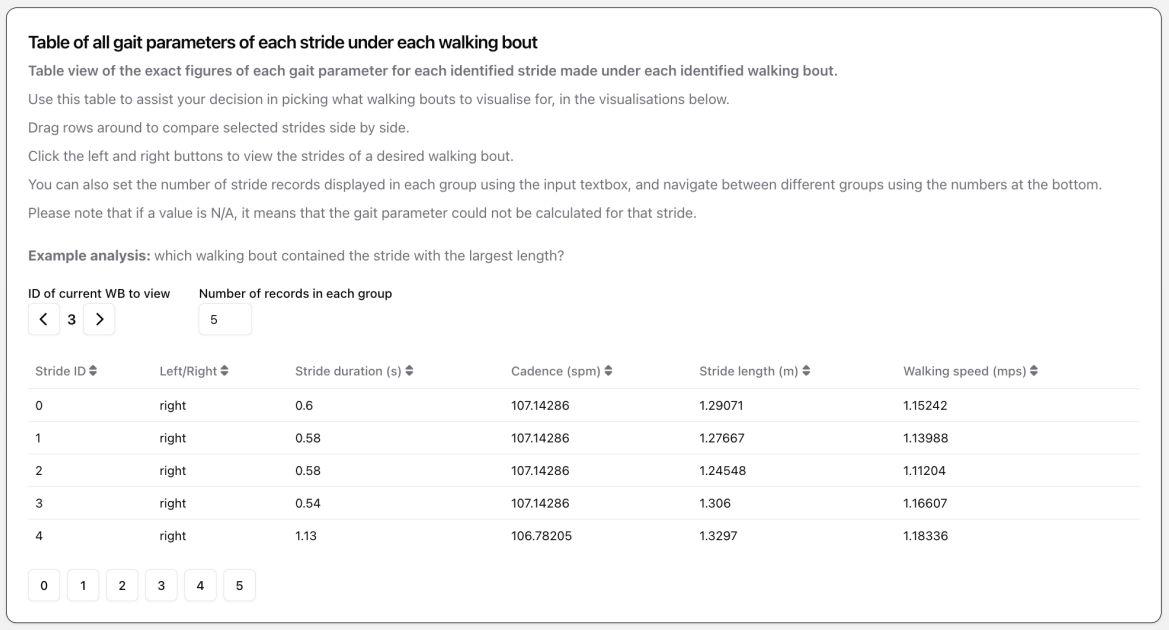


Figure 5.25: Table of gait parameter values for each stride

5.2.7.2 Progression of a gait parameter's values

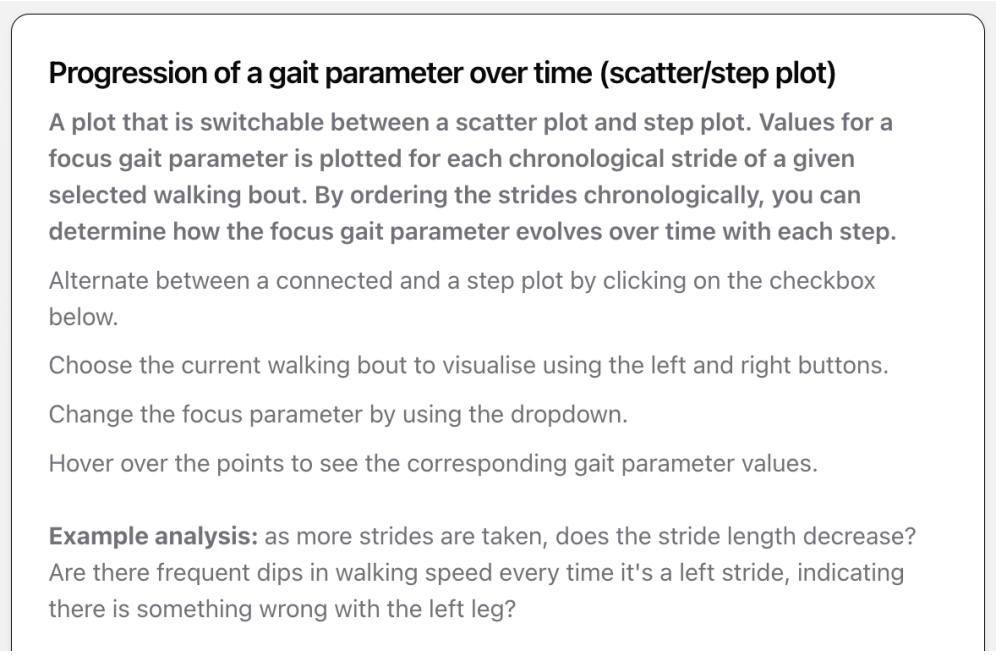


Figure 5.26: Description of scatter plot showing the progression of a gait parameter across strides

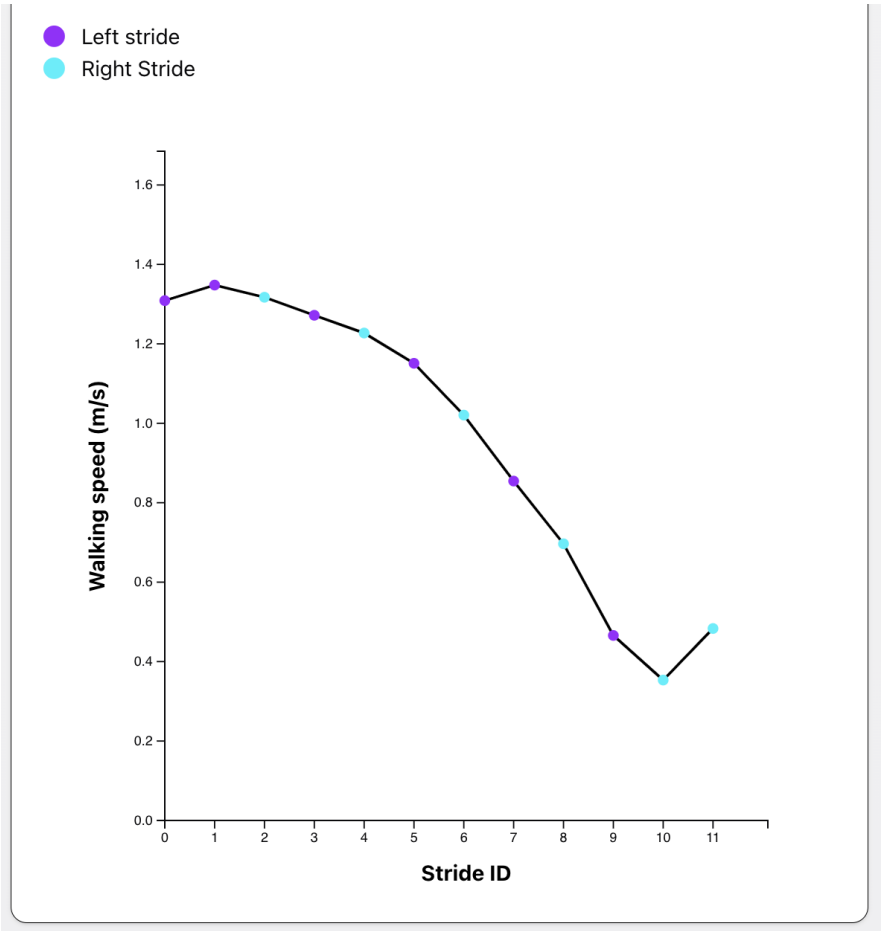


Figure 5.27: Scatter plot showing the progression of a gait parameter across strides

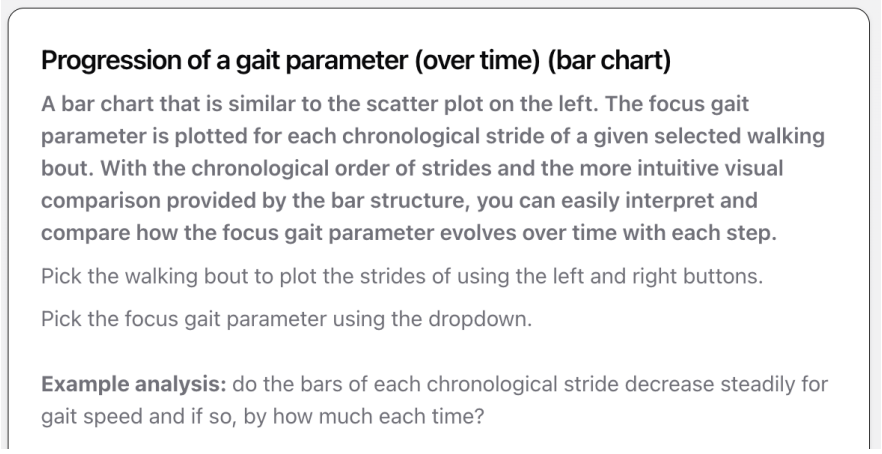


Figure 5.28: Description of bar chart showing the progression of a gait parameter across strides

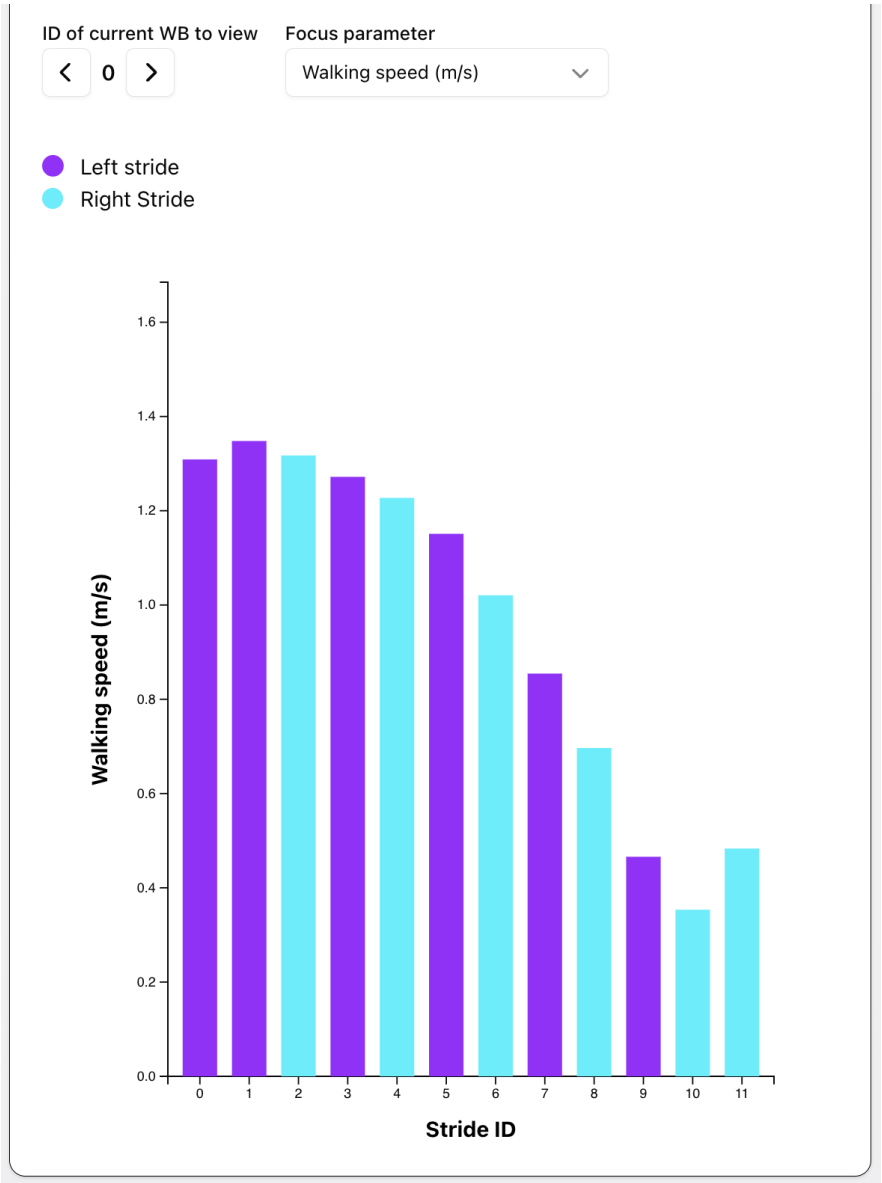


Figure 5.29: Bar chart showing the progression of a gait parameter across strides

5.2.7.3 Distribution of a given parameter by a box and violin plot

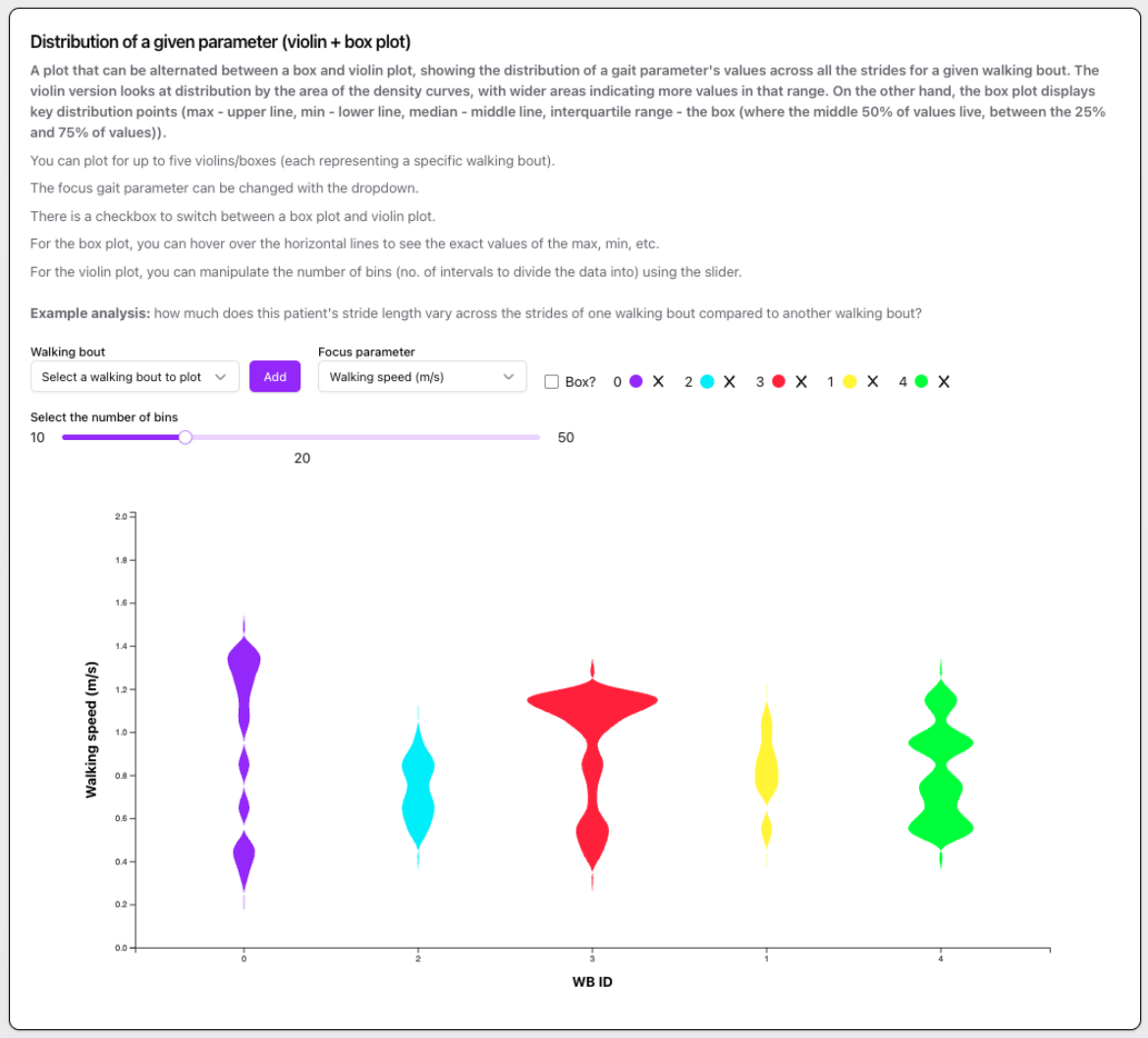


Figure 5.30: Violin/box plot showing the distribution of a gait parameter for strides of multiple WBs

5.2.7.4 Distribution of a given parameter by histogram

It is important to note that hovering on overlapping bars will only show the range and frequency of the histogram plotted last.

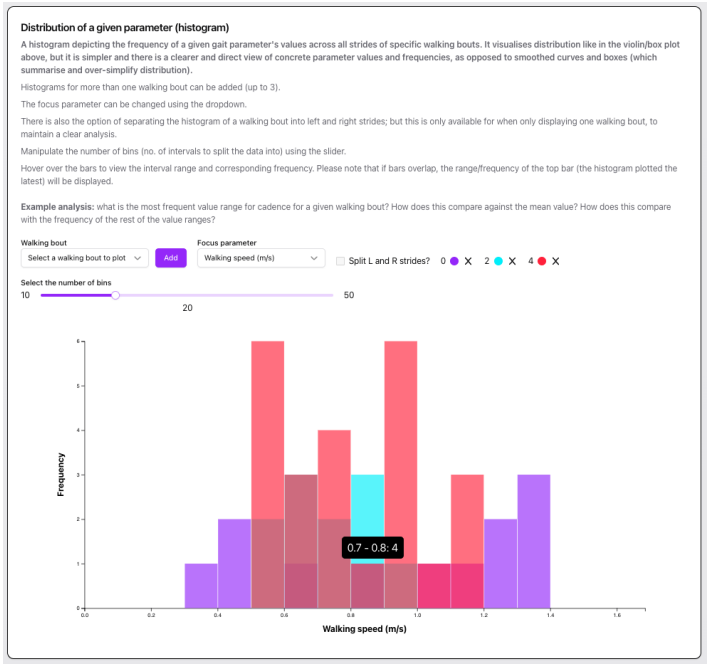


Figure 5.31: (a) Distribution of a given parameter across strides of multiple WBs by a histogram

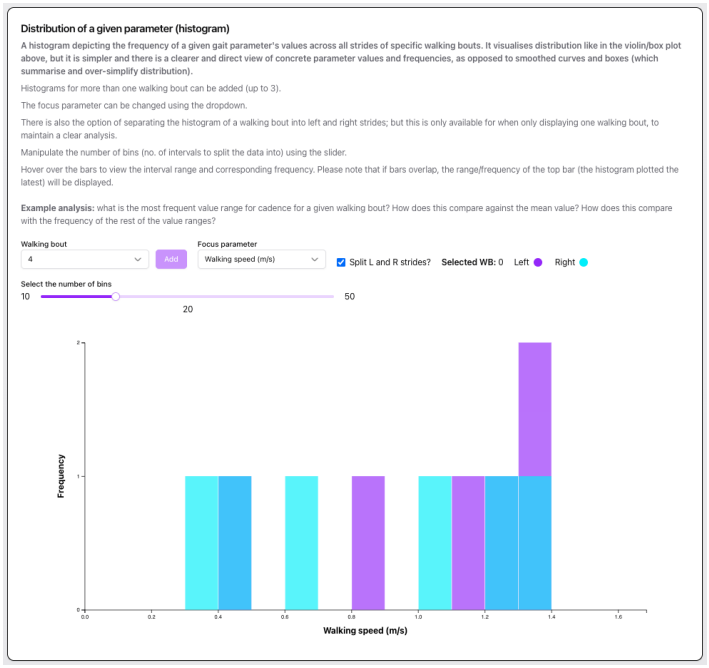


Figure 5.32: (b) Distribution of a given parameter across split left and right strides of a singular WB by a histogram

5.2.7.5 Relationship between all gait parameter by a PCP

Hovering on a specific data line will display the corresponding WB and stride.

Figure 5.33: (a) Parallel coordinates plot for the per-stride DMO values of multiple selected WBs

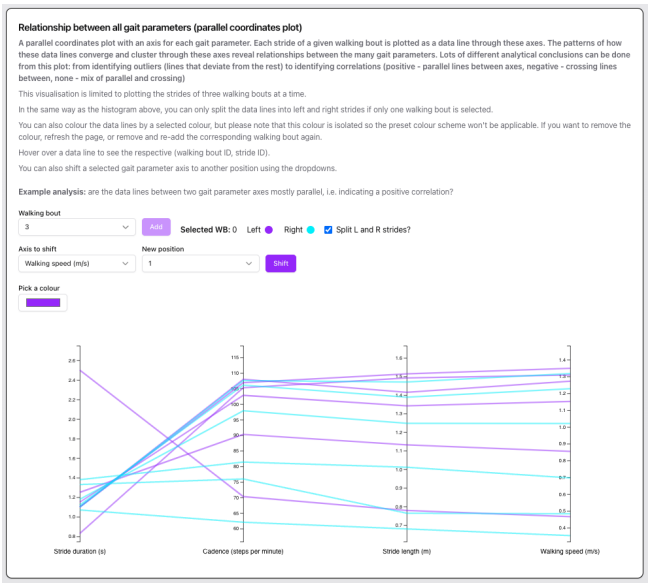


Figure 5.34: (b) Parallel coordinates plot for the per-stride DMO values of left and right strides of a given WB

5.2.7.6 Comparison between specific strides by a radar chart

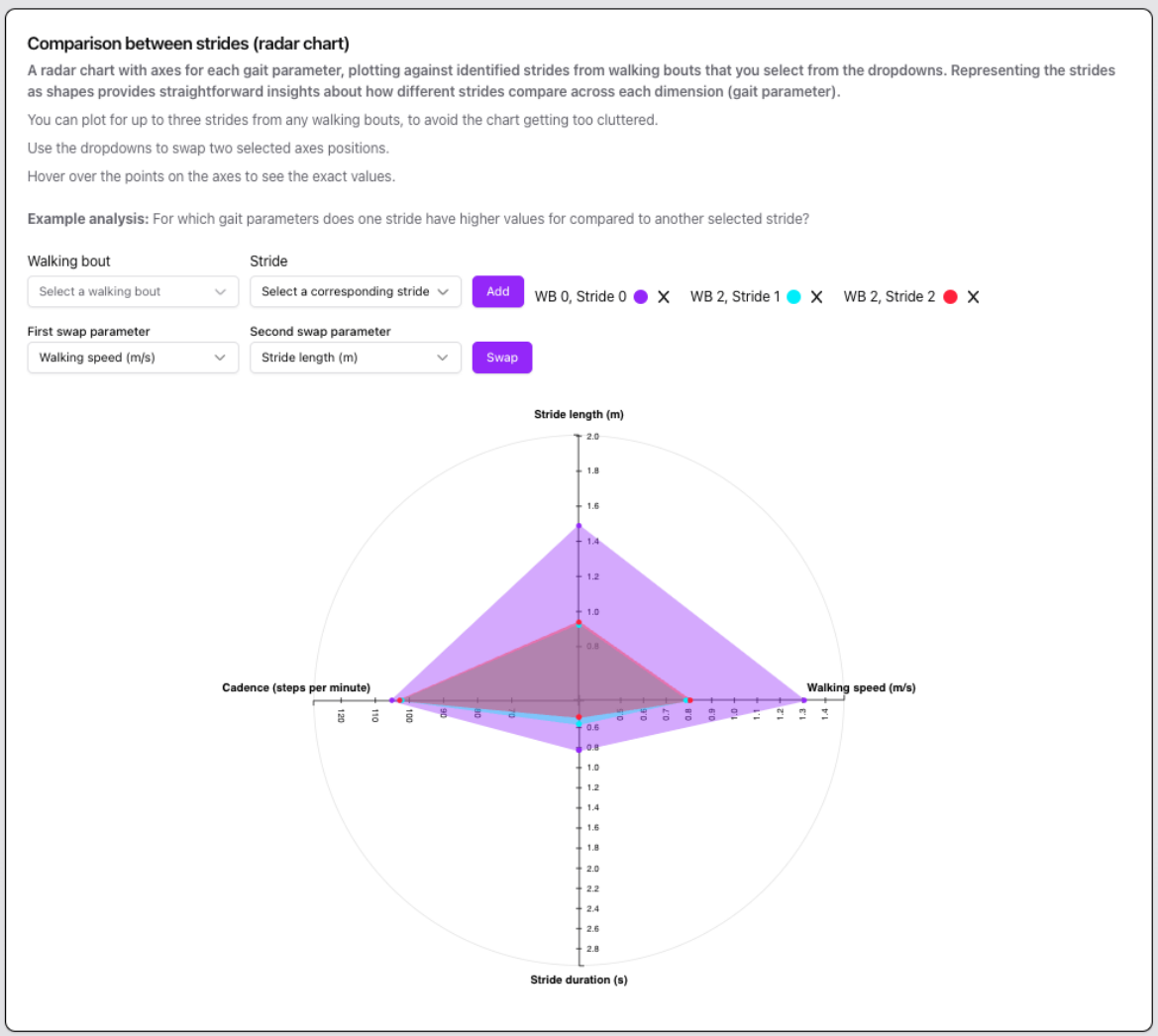


Figure 5.35: Comparison between selected strides of multiple WBs with a radar chart

5.2.7.7 Comparison between specific strides by a heat map

Comparison is intended between chronological strides of multiple WBs, so some cells are blacked out to account for unequal stride counts between WBs.

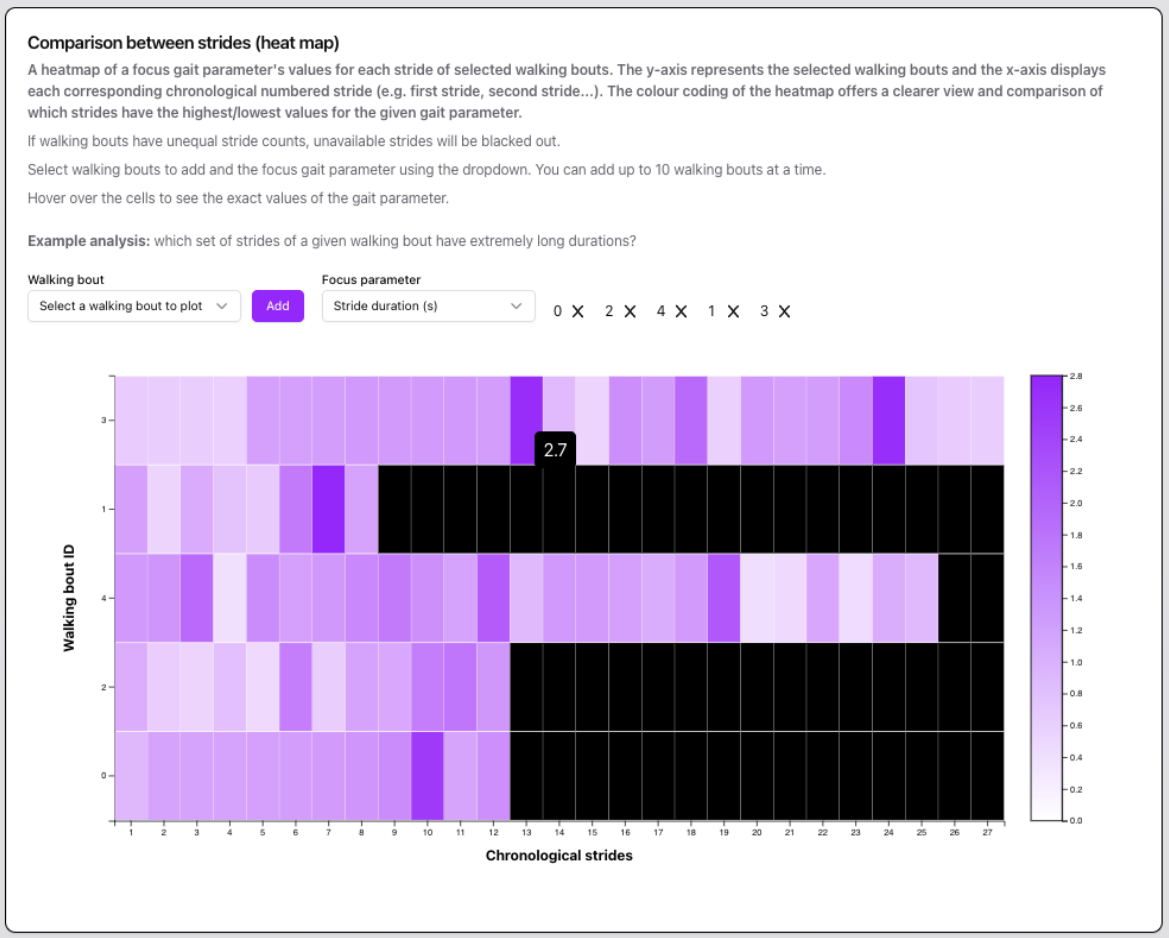


Figure 5.36: Comparison between chronological strides of multiple WBs by a heat map

5.2.8 d3.js implementation

It is useful to showcase an exemplar code snippet of how these visualisations were created using d3.js. The overall approach to creating each visualisation type is the same, differing only by some additional operations to draw components unique to each type. Additionally, with React, reusable components were created for each visualisation type, so plots and graphs for different datasets could be easily generated.

The implementation of the bar chart component (`src/components/viz/charts&graphs/bar-chart.tsx`) will be used to detail the use of d3.js in this sub-section.

5.2.8.1 Basic template

All visualisation components use this base code structure (figures 5.37 and 5.38). Firstly, the definition of common component properties so that it can be reused for different visualisations of the same type. Then, variables are defined for establishing dimensions and data values. Figure 5.38 features the actual SVG HTML element that gets progressively "drawn on" in the custom `draw()` method.

```
7   interface Props {
8     width: number;
9     height: number;
10    margin: Margin;
11    className: string;
12    data: [string, number][];
13    xLabel: string;
14    yLabel: string;
15    tiltXLabels?: boolean;
16    differentColours?: number[][];
17  }
18  export default function BarChart({
19    width,
20    height,
21    margin,
22    className,
23    data,
24    xLabel,
25    yLabel,
26    tiltXLabels = false,
27    differentColours,
28  }: Props) {
29    const ref = useRef(null); // reference to SVG HTML element
30
31    // set dimensions of graph
32    const totalHeight = height + margin.top + margin.bottom;
33    const totalWidth = width + margin.left + margin.right;
34
35    // split up x and y values.
36    const xValues = data.map((d) => d[0]);
37    const yValues = data.map((d) => d[1]);
38
39    useEffect(() => {
40      draw();
41    });
```

Figure 5.37: Template (a): basic definitions of component's properties, SVG element reference, dimensions and data

```
142   return (
143     <svg width={width} height={height} ref={ref} className={className}></svg>
144   );
145 }
```

Figure 5.38: Template (b): main HTML SVG element

5.2.8.2 Custom `draw()` method

The `draw()` method defines the actual operations for drawing the actual components of the visualisation. There are still a few common operations, such as the creation of the x and y axes (figures 5.39 and 5.40), but with slight differences depending on the visualisation. For example, the bar chart uses `.scaleBand()` for the x axis, as we're dealing with discrete values, whereas the scatter plot would use `.scaleLinear()` as values are continuous. Finally, figure 5.41 shows the creation of the actual bars, which is specific to the bar chart.

```
43  function draw() {
44    // get svg element and clear canvas
45    const svg = d3.select(ref.current);
46    svg.select("*").remove();
47
48    // template for the main plot
49    const plot = svg
50      .attr("width", totalWidth)
51      .attr("height", totalHeight)
52      .append("g")
53      .attr("transform", "translate(" + margin.left + "," + margin.top + ")");
54
55    // create the x axis
56    const x = d3
57      .scaleBand()
58      .range([0, width])
59      .domain(data.map((datapoint) => datapoint[0]))
60      .padding(0.3); // spacing between bars
61    const xAxis = d3.axisBottom(x);
62
63    // limit x labels if there are too many
64    if (xValues.length > 30) {
65      xAxis.tickValues(x.domain().filter((_, i) => i % 5 === 0));
66    }
67
68    // draw the x axis
69    const plottedXAxis = plot
70      .append("g")
71      .attr("transform", "translate(0," + height + ")")
72      .call(xAxis);
73    if (tiltXLabels) {
74      plottedXAxis
75        .selectAll("text")
76        .attr("transform", "translate(12,20)rotate(90)");
77    }
```

Figure 5.39: `draw()` method (a): Definition of the actual plot area, and creation of x axis.

```
79      // create and draw the y axis
80      const maxY = Math.max(...yValues);
81      const y = d3
82        .scaleLinear()
83        .domain([0, maxY + maxY * 0.1])
84        .range([height, 0]);
85      const yAxis = d3.axisLeft(y);
86      plot.append("g").call(yAxis);
87
88      // add the x and y axis labels
89      plot
90        .append("text")
91        .attr("y", height + 55)
92        .attr("x", width / 2)
93        .style("text-anchor", "middle")
94        .attr("font-weight", 700)
95        .text(xLabel);
96      plot
97        .append("text")
98        .attr("transform", "rotate(-90)")
99        .attr("x", "-" + height / 2)
100       .attr("y", "-45")
101       .style("text-anchor", "middle")
102       .text(yLabel)
103       .attr("font-weight", 700);
```

Figure 5.40: `draw()` method (b): Creation of y axis and axes labels

```
117     // add the bars for each data point
118     const bars = plot
119       .selectAll("bar")
120       .data(data)
121       .enter()
122       .append("rect")
123       .attr("x", (d) => x(d[0])!)
124       .attr("y", (d) => y(d[1])!)
125       .attr("width", x.bandwidth())
126       .attr("height", (d) => height - y(d[1]))
127       // hover data display functionality
128       .on("mouseover", (event, d) => {
129         tooltip
130           .html(`(${d[0]}, ${roundToNDpIfNeeded(d[1], 3)})`)
131           .style("display", "block")
132           .style("left", event.pageX + 10 + "px")
133           .style("top", event.pageY - 20 + "px");
134       })
135       .on("mouseout", () => {
136         tooltip.style("display", "none");
137       });
```

Figure 5.41: `draw()` method (c): Drawing the actual bars of the bar chart

5.3 Testing

The creation of automated unit tests was minimised, due to limited time constraints and aim of delivering all the basic functionality. Where a given function was simple and confidence could be easily placed on its correctness, a test case would not be created. Additionally, it is difficult to create test cases for visualisations. For example, it is difficult to assert that a bar of a bar graph is of the correct height given the value. The time cost of producing such test cases is substantial, compared with simple, sufficient manual testing through visual observation.

Nevertheless, this section presents exemplar unit tests where they were created, and the user testing conducted to evaluate the final system.

5.3.1 Pytest back-end testing

Unit tests were created with Pytest to ensure that core functions and the API endpoint for DMO extraction behave as expected. These can be found in `py_tests/dmo_extraction` and `py_tests/api` respectively.

5.3.1.1 Core DMO extraction functions

Figure 5.42 features an example unit test for the `create_dataset_from_dataframe` function. It takes a dataset in the form of a `pandas` (Python data analysis library) `DataFrame`, alongside other metadata, and outputs a `GaitDatasetFromData` object that is expected by the `mobgap` pipelines. The test asserts that the returned object truly contains the dataset that was passed in.

```python
class TestCreateDatasetFromDataframe:
  def test_returns_valid_gait_dataset_with_data(self):
    # create data/parameters to pass in
    data = pd.DataFrame({
      "samples": [0, 1, 2],
      "acc_x": [0.2834, 0.234, 0.234],
      "acc_y": [0.345, 0.345, 0.8723],
      "acc_z": [0.7234, 0.2347, 0.7234],
      "gyr_x": [0.2347, 0.123, 0.123],
      "gyr_y": [0.473, 0.234, 0.7123],
      "gyr_z": [0.8123, 0.1238, 0.8712],
    })
    sensor_height_m = 1.6
    height_m = 1.8
    measurement_condition = "laboratory"
    sampling_rate_hz = 100

    # get the result of the function
    dataset = create_dataset_from_dataframe(data, sensor_height_m, height_m, measurement_condition,
    sampling_rate_hz)

    assert isinstance(dataset, GaitDatasetFromData)
    # check the single record matches our data
    single_record = dataset[0]
    assert single_record.data_ss.equals(data) == True
```

Figure 5.42: Unit test for `create_dataset_from_dataframe` function

5.3.1.2 API endpoint

An example unit test case for the DMO extraction API endpoint can be seen in figure 5.43. It passes in valid inputs for all of the endpoint's parameters except for the sampling rate, which should therefore return an error. The error is asserted in the last two lines, by the failing status code and expected error message to be returned.

```python
  def test_invalid_sampling_rate_returns_error_message(self):
    file_path = Path(__file__).parent.parent.parent / "sample_data" / "TimeMeasure1_Test11_Trial1.csv"
    files = {"csvFile": open(file_path, "rb")}
    data = {
      "name": "Test extraction",
      "description": "Test description",
      "public": True,
      "samplingRate": 1, # invalid sampling rate.
      "sensorHeight": 1.65,
      "patientHeight": 1.80,
      "setting": "laboratory",
      "convertToMs": True,
    }

    response = client.post("/api/py/dmo_extraction", data=data, files=files)
    assert response.status_code == 400
    assert response.json()["detail"] == general_error_message
```

Figure 5.43: API endpoint unit test for invalid input

5.3.2 Jest front-end testing

Jest was used to test the front-end functionality written in JavaScript. It provides customary assertion unit test statements. This is used in combination with React Testing Library to test the correct rendering of created components, and in isolation to test for utility functions written in vanilla JavaScript. These can be found in the `__tests__` folder.

5.3.2.1 Component testing

Figure 5.44 features the unit test for checking that the main form component for submitting analysis inputs, properly calls the submission handler with the correct values. A stub function is used to assert that it is called upon clicking submit, and with the correct input values.

```
77  it("calls submission handler with the correct data", async () => {
78    const values = {
79      name: "Testing analysis",
80      description: "This is the best analysis",
81      samplingRate: 100,
82      sensorHeight: 1.69,
83      patientHeight: 1.8,
84      setting: "laboratory",
85      csvFile: new File([""], "test.csv", { type: "text/csv" }),
86      convertToMs: false,
87    };
88
89    cleanup(); // get rid of the previous render to render our component with mock handler.
90    const submissionHandler = jest.fn();
91    render(<NewAnalysisForm submissionHandler={submissionHandler} />);
92
93    await fillForm(values);
94    await submitForm();
95
96    await waitFor(() => {
97      expect(submissionHandler).toHaveBeenCalledTimes(1);
98      expect(submissionHandler).toHaveBeenCalledWith(values);
99    });
100 });
101 });
```

Figure 5.44: Submission unit test for new analysis form component

5.3.2.2 Utility function testing

Utility functions for vanilla JavaScript operations (e.g. string manipulation) were also tested. An exemplar can be seen in figure 5.45. It features three unit tests for the different cases, each checking that the function converts the input in hours to the correct time units.

```
describe("convertHoursToReadableForm", () => {
  it("stays at hours if it is larger than 1", () => {
    const result = convertHoursToReadableForm(2.78);
    expect(result).toBe("3 hours");
  });

  it("converts to minutes if the number of hours is less than 1", () => {
    const result = convertHoursToReadableForm(0.46);
    expect(result).toBe("28 mins");
  });

  it("converts to seconds if the number of minutes is less than 1", () => {
    const result = convertHoursToReadableForm(0.01305555556);
    expect(result).toBe("47 secs");
  });
});
```

Figure 5.45: Unit test cases for `convertHoursToReadableForm` function

5.3.3 User testing/evaluation

User testing was conducted with a healthcare researcher and clinician, that each have experience monitoring patients with MS. The methodology has been previously described in sub-section 3.7. Results from the entire evaluation process will be revealed here.

5.3.3.1 System usability

The average score given for overall system usability was 8/10. The general consensus was that all the functionality was intuitive, and there was sufficient information given to understand the more complex visualisations. The modern interface was appreciated.

However, there was disapproval of the large amount of text content across the system. Although explanations are useful, too much information can be off-putting. Participants suggested a separate manual or training stage. Additionally, if there are lots of WBs/strides, filtering through and selecting the correct WB/stride is difficult, especially in drop-down menus.

5.3.3.2 Fulfillment of requirements

There was only sufficient time to attempt a subset of the planned requirements. Only these were included in the questionnaire, and participants were asked how successfully they felt

each had been met. Tables 5.1 and 5.2 show the average success scores for each non-functional and functional requirement.

#	Requirement	Priority	Score
1	For a new gait analysis, the user is able to upload a CSV of sensor data and input other details (to abide by data requirements and also additional descriptions about the gait analysis) to automatically generate relevant, pre-defined data visualisations.	M	7.5
2	There should a section detailing important input data requirements to ensure reliable information about patient gait is extracted.	M	7.5
3	Where applicable, the user is able to edit the settings of a given data visualisation (e.g. pick specific data records to visualise).	M	7.5
7	The user is able to select subsets of data to compare (e.g. between average DMOs of two patients). Different visualisations should be used for comparison (e.g. graphs or table of precise values).	S	8
12	The user is able to automatically transform their CSV data into the appropriate format. Specifically: converting into the correct units and coordinate system.	C	8.5

Table 5.1: Averages of evaluation participant's perceived success scores for each attempted functional requirement

#	Requirement	Priority	Score
1	Data uploads are optimised and as fast as possible.	M	8
2	The interface is optimised for desktop screens.	M	9
4	All information displayed is reliable and accurate for clinical analysis.	M	8.5
6	Navigation throughout and use of the system are easy, intuitive and assisted if necessary.	M	8.5
7	Invalid interactions are correctly discouraged and the user is guided towards the right paths.	M	9
8	The overall design and theme are modern, organised and aesthetic.	S	8.5
9	Pages load quickly.	S	9

Table 5.2: Averages of evaluation participant's perceived success scores for each attempted non-functional requirement

5.3.3.3 User acceptance/system effectiveness

An average score of 8.5 was given on the likeliness of actual, real-world usage. While this is an extremely positive result, this can be further improved by implementing the leftover requirements and considering suggestions made by participants. Lots of feedback were given regarding the effectiveness of the visualisations and overall system. The full list can be found in Appendix I, but the main points will be discussed here.

Participants approved of some visualisations in particular. Specifically, visualisations that split left and right strides will be useful for examining differences between left and right limbs. One participant liked visualisations that looked at relationships between parameters; particularly the ability to view how walking speed has a knock-on effect on other parameters. The same participant has also had experience using radar charts with clinic-based assessments involving pwMS, and approved its usefulness in distinguishing between levels of mobility impairment severity.

There was emphasis on the need for initial simplified view, accompanied by the option to delve into more advanced visualisations and functionality if desired. For example, PCPs may be on the more advanced side, which may be off-putting for clinicians. More co-designing with clinicians was also suggested, to fix the best settings like axes order.

Across evaluations, participants identified the need for reference ranges in the visualisations. The researcher involved in Mobilise-D revealed that there is ongoing research for this data.

There was also the realisation that it was difficult to identify relevant WBs/strides to visualise for. For example, determining which WBs/strides are on a certain day/time, and having the option to filter them out.

Although this was initially planned in functional requirement 5, the desire for longitudinal analysis (e.g. comparison of analyses over 6 months) was also frequently expressed. Individual strides/WBs are less useful, compared to higher-level comparisons of averages over different analyses (e.g. normal vs. impaired) at separate time-points.

Finally, two files were provided (one 200MB and another 20MB), however participants ran into request errors whenever they tried the larger file. Data processing needs to be better optimised for larger CSV inputs.

Chapter 6

Results and Discussion

This chapter will discuss the overall success of the project, in terms of meeting requirements and truly satisfying clinician wants and needs. Then, future improvements to further this project are proposed, including evaluation feedback and personal suggestions.

6.1 Requirements

A positive outcome of this project is that all the must-have functional requirements were implemented, and the core focus of visualisation has been achieved to a high standard.

However, a lot of the planned requirements were not attempted due to the comparatively short duration of the project.

Tables 6.1 and 6.2 present the complete tables of functional and non-functional requirements and whether they have been passed or failed (according to what the system objectively can/cannot do and user evaluation from subsection 5.3.3).

#	**Requirement**	**Priority**	**P/F**
1	For a new gait analysis, the user is able to upload a CSV of sensor data and input other details (to abide by data requirements and also additional descriptions about the gait analysis) to automatically generate relevant, pre-defined data visualisations.	M	P
2	There should a section detailing important input data requirements to ensure reliable information about patient gait is extracted.	M	P

3	Where applicable, the user is able to edit the settings of a given data visualisation (e.g. pick specific data records to visualise).	M	P
4	The user is able to create their own account that will hold all their patient and gait analysis data.	S	F
5	The user is able to create new patients to store their background information (e.g. age and sex) and gait analyses.	S	F
6	The user is able to effectively filter through gait analyses and patients.	S	F
7	The user is able to select subsets of data to compare (e.g. between average DMOs of two patients). Different visualisations should be used for comparison (e.g. graphs or table of precise values).	S	P
8	A reference range should be provided with colour coding for DMO values to provide context for analysis (e.g. what is considered low vs. fast gait speed).	C	F
9	Each visualisation can be exported into an image format and saved in the user's device.	C	F
10	The user is able to create new, custom visualisations, outside of the automatically generated ones, that they seem useful analysis.	C	F
11	There are useful infographics and posters conveying general information about MS and gait, that may aid analysis.	C	F
12	The user is able to automatically transform their CSV data into the appropriate format. Specifically: converting into the correct units and coordinate system.	C	P
13	For a given patient, the user can view a patient outcome dashboard/electronic health record summarising their current patient's gait condition and historical analyses.	C	F
14	The user is able to organise the structure of visualisations.	C	F

Table 6.1: Final passing/failure (P/F) of functional requirements

#	**Requirement**	**Priority**	**P/F**
1	Data uploads are optimised and as fast as possible.	M	F
2	The interface is optimised for desktop screens.	M	P
3	Descriptions and text should be as simplified as possible for clinicians to easily understand.	M	P
4	All information displayed is reliable and accurate for clinical analysis.	M	P

5	All data is stored securely and can only be accessed by the right users.	M	P
6	Navigation throughout and use of the system are easy, intuitive and assisted if necessary.	M	P
7	Invalid interactions are correctly discouraged and the user is guided towards the right paths.	M	P
8	The overall design and theme are modern, organised and aesthetic.	S	P
9	Pages load quickly.	S	P
10	Colours used are colour-blind friendly.	C	F

Table 6.2: Final passing/failure (P/F) of non-functional requirements

It is useful to note that while functional requirements 7 and 12 pass and evaluation participants were satisfied, there is still room for improvement. Users are currently unable to compare data between different patients, and the conversion between coordinate systems is not supported. Non-functional requirement 1 should also be mentioned. The system struggles to extract DMOs for extremely large files, and optimisation should be greatly prioritised in future work.

There was early consideration of the likelihood of not meeting all the requirements. Therefore, priorities had to be assigned by emphasising the main focus on visualisation. The complete set of requirements encompass an ambitious, more-comprehensive application for gait analysis. This would be overall more useful to clinicians, but exceeded the available scope and resources for a dissertation project.

In any case, final efforts were oriented towards improving the usability and flow of the system, and ensuring that attempted requirements were, at minimum, implemented to a high standard with minimal bugs. This is reflected in the passing of the majority of non-functional requirements, and positive opinions on usability during user evaluation.

6.2 User satisfaction

From the evaluation results (sub-section 5.3.3), the overall consensus is that clinicians would be able to easily use the system, and make meaningful conclusions about their patient's gait.

However, there are still lots of immediate features that can be implemented to further enhance the overall usefulness of the system. For example, storage of patient and past analysis data would greatly expand the possible gait conclusions that can be made (e.g. longitudinal

comparison of analyses 6 months apart).

As a visualisation interface, this project is successful as it offers practical graph and plot types for gait analysis. Nevertheless, more data can be extracted and stored to improve the usefulness of the visualisations themselves and the analyses that can be made with them.

6.3 Future work

There are many avenues for future improvements, in the facet of unaccomplished requirements and evaluation feedback.

6.3.1 Unfinished requirements

There are many requirements that remain unattempted. Namely, functional requirement 4 would have improved the usefulness of the system substantially, in that multiple gait analyses can be compared between different patients and time-points. However, this was a technically heavy requirement, because it would involve creating a new database, authentication and authorisation, and all while ensuring proper testing of everything.

Fortunately, mock-ups have been designed to fulfill all the requirements, so they can be used for reference to guide implementation.

6.3.2 Evaluation feedback

During evaluation, lots of feedback was received on the system usability and effectiveness. While the general opinion was positive, many areas for improvement were identified, both during observation of participant use and from the questionnaire feedback afterwards. All the findings can be found in Appendix I. Some suggested changes were smaller like pre-filling mandatory fields with expected/typical data. Other changes were larger like segregating the system into a simplified and more advanced part.

6.4 Conclusion

Overall, the project has been successful and offers clinicians a way to visualise gait data for MS patients. Clinicians can use the provided visualisations to suggest possible remedies for impaired mobility. While the planned requirements could be considered as too ambitious given the short-time frame of a dissertation project, there exist mock-ups that can be used

to implement unfinished requirements and offer a more complete application with more effective visualisations. Given the feedback from the evaluation, a second iteration can also be produced. Then, this iterative process with clinicians can be continued until clinicians are adequately satisfied to apply the system in their regular assessments and treatments of MS patients.

Bibliography

Aggarwal, A., Gupta, R. and Agarwal, R. (2018) 'Design and Development of Integrated Insole System for Gait Analysis', 2018 Eleventh International Conference on Contemporary Computing (IC3). Noida, India, 2-4 August, pp. 1-5. doi: https://doi.org/10.1109/IC3.2018.8530543.

Brittain, G. et al. (2023) 'Mobilise-D: The largest observational trial of digital mobility outcome measures in multiple sclerosis', Journal of the Neurological Sciences, 455(Supplement), p. 121059. doi: https://doi.org/10.1016/j.jns.2023.121059.

Cleveland Clinic (2024) Multiple Sclerosis (MS). Available at: https://my.clevelandclinic.org/health/diseases/17248-multiple-sclerosis (Accessed: 22 November 2024)

Coca-Tapia, M. et al. (2021) 'Gait Pattern in People with Multiple Sclerosis: A Systematic Review', Diagnostics (Basel, Switzerland), 11(4), p. 584. doi: https://doi.org/10.3390/diagnostics11040584.

Czaban, T. (2024) Healthcare Dashboard Examples: Different Use Cases and Their Benefits'. Available at: https://www.gooddata.com/blog/healthcare-dashboards-examples-use-cases-and-benefits/ (Accessed 14 April 2025)

Dunskiy, I. (2023) 'Healthcare Data Visualisation: Examples & Key Benefits', Demigos, 18th July. Available at: https://demigos.com/blog-post/healthcare-data-visualization/ (Accessed: 28 November 2024).

Gaitmap (no date) Prepare your own data. Available at: https://gaitmap.readthedocs.io/en/latest/source/user_guide/prepare_data.html (Accessed: 05 December 2024).

Hum, W. W. G. (2025-a) mobVis mockups. Available at: https://www.figma.com/design/RzQce25CumIgblkhCWRwk0/dissertation?node-id=0-1&t=IDY4UuB24ripHJrd-1 (Accessed: 28 April 2024).

Hum, W. W. G. (2025-b) Shorter overview of gait visualisation interface. Available at: https://www.loom.com/share/76d1a35e6e0f4d1e99d04f47db2912a5?sid=16bad18f-b66

e-4f92-942d-725952ef6fa4 (Accessed: 28 April 2024).

Hum, W. W. G. (2025-c) Longer overview of gait visualisation interface. Available at: https://www.loom.com/share/35c7d9b9c1754889b44aba9a036d7f26?sid=b444890e-990c-4dfe-8557-3f1d3c0e80ad (Accessed: 28 April 2024).

Jason Davies (no date) Parallel Sets. Available at: https://www.jasondavies.com/parallel-sets/ (Accessed: 29 November 2024).

JMP (no date) Mosaic Plot. Available at: https://www.jmp.com/en_gb/statistics-knowledge-portal/exploratory-data-analysis/mosaic-plot.html (Accessed: 23 November 2024).

Kirk, C. et al. (2024) 'Mobilise-D insights to estimate real-world walking speed in multiple conditions with a wearable device', Sci Rep, 14(1754). doi: https://doi.org/10.1038/s41598-024-51766-5.

Küderle, A., et al. (2024), Mobgap, 0.10.0 [https://github.com/mobilise-d/mobgap]. GitHub. Available at: https://pypi.org/project/mobgap/ (Accessed: 06 December 2024).

Kúderle, A., (2025) 'Aggregated parameters question', mobgap, 07 Februrary. Available at: https://github.com/mobilise-d/mobgap/discussions/206 (Accessed: 07 April 2025).

Larocca, N. G. (2011) 'Impact of walking impairment in multiple sclerosis: perspectives of patients and care partners', The patient, 4(3), pp. 189-201. doi: https://doi.org/10.2165/11591150-000000000-00000.

Medesk (no date) Key Aspects of Healthcare Data Visualization. Available at: https://www.medesk.net/en/blog/data-visualization-in-healthcare/ (Accessed: 28 November 2024).

Micó-Amigo, M. E. et al. (2023) 'Assessing real-world gait with digital technology? Validation, insights and recommendations from the Mobilise-D consortium', Journal of NeuroEngineering and Rehabilitation, 20(78). doi: https://doi.org/10.1186/s12984-023-01198-5.

Mikolaizak, A. S. et al. (2022) 'Connecting real-world digital mobility assessment to clinical outcomes for regulatory and clinical endorsement-the Mobilise-D study protocol', PLoS ONE, 17(10). doi: https://doi.org/10.1371/journal.pone.0269615.

Mobgap (no date-a) Coordinate System. Available at: https://mobgap.readthedocs.io/en/latest/guides/coordinate_system.html (Accessed: 4 December 2024).

Mobgap (no date-b) Preconfigured Mobilised Pipelines. Available at: https://mobgap.readthedocs.io/en/latest/auto_examples/pipeline/_03_preconfigured_mobilised_pip

elines.html (Accessed: 7 April 2025).

MS Society (no date) Types of MS. Available at: https://www.mssociety.org.uk/about-ms/types-of-ms (Accessed: 22 November 2024)

National Multiple Sclerosis Society (no date) How Many People Live With Multiple Sclerosis? Available at: https://www.nationalmssociety.org/understanding-ms/what-is-ms/who-gets-ms/how-many-people (Accessed: 22 November 2024)

Northwestern Medicine (2022) What is Multiple Sclerosis? [Infographic]. Available at: https://www.nm.org/healthbeat/healthy-tips/What-Is-Multiple-Sclerosis-infographic (Accessed: 28 November 2024).

Polhemus, A. et al. (2021) 'Walking on common ground: a cross-disciplinary scoping review on the clinical utility of digital mobility outcomes', npj Digit. Med., 4(149). doi: https://doi.org/10.1038/s41746-021-00513-5.

Shah, V. V. et al. (2020) 'Quantity and quality of gait and turning in people with multiple sclerosis, Parkinson's disease and matched controls during daily living' Journal of neurology, 267(4), pp. 1188-1196. doi: https://doi.org/10.1007/s00415-020-09696-5.

Sutliff, M. H. (2009) 'Contribution of impaired mobility to patient burden in multiple sclerosis', Current Medical Research and Opinion, 26(1), pp. 109–119. doi: https://doi.org/10.1185/03007990903433528.

The Data Visualisation Catalogue (no date-a) Parallel Coordinates Plot. Available at: https://datavizcatalogue.com/methods/parallel_coordinates.html (Accessed: 23 November 2024).

The Data Visualisation Catalogue (no date-b) Parallel Sets. Available at: https://datavizcatalogue.com/methods/parallel_sets.html (Accessed: 23 November 2024).

The Data Visualisation Catalogue (no date-c) Heatmap (Matrix). Available at: https://datavizcatalogue.com/methods/heatmap.html (Accessed: 28 November 2024).

The Data Visualisation Catalogue (no date-d) Box and Whisker Plot. Available at: https://datavizcatalogue.com/methods/box_plot.html (Accessed: 23 November 2024).

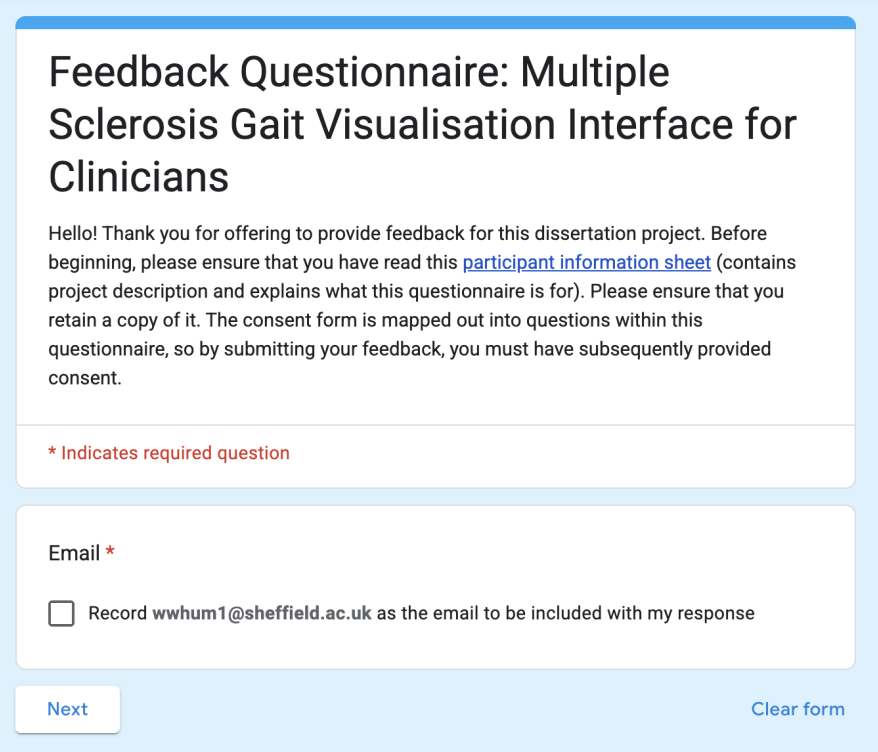
The Data Visualisation Catalogue (no date-e) Radar Chart. Available at: https://datavizcatalogue.com/methods/radar_chart.html (Accessed: 23 November 2024).

Vaniukov, S. (2024) 'The Impact of Data Visualisation in Healthcare: A Pathway to Enhanced Patient Care', Medium, 7th February. Available at: https://medium.com/@vaniukov.s/the-impact-of-data-visualization-in-healthcare-a-pathway-to-enhanced-patient-care-7e555550a403 (Accessed: 28 November 2024).

Appendices

Appendix A

Iterative feedback questionnaire



Feedback Questionnaire: Multiple Sclerosis Gait Visualisation Interface for Clinicians

Hello! Thank you for offering to provide feedback for this dissertation project. Before beginning, please ensure that you have read this participant information sheet (contains project description and explains what this questionnaire is for). Please ensure that you retain a copy of it. The consent form is mapped out into questions within this questionnaire, so by submitting your feedback, you must have subsequently provided consent.

* Indicates required question

Email *

Record wwhum1@sheffield.ac.uk as the email to be included with my response

Next

Clear form

Please let us know a little about who you are.

This is just to understand who the feedback is coming from.

Are you a clinician that works with patients with multiple sclerosis (MS)? *

- ◯ Yes
- ◯ No

If not, what do you do or what is your relevant background/experience? (e.g. worked with and analysed MS patient gait data)

Your answer

Back Next Clear form

STEP 1: Read through the list of requirements for this project.

Please go through the list of requirements below to get an understanding of what sort of functionality has been planned for implementation into the final system. Don't spend too much time on this, you don't need to understand it completely. Just have a general idea for each requirement, and hopefully any questions you have may be cleared up by the next section, where we go through mockups that will hopefully offer a visual understanding.

1) Data requirements

These are requirements that have been established around the structure/content of the input data that should be supplied by users (clinicians). These are important to clarify what sort of the input data the final system will support.

These requirements are to align with the coding library that we will use to extract gait parameters from the IMU data.

These are all likely to be implemented into the final system.

Table of data requirements

#	Requirement
1	The input data should consist of raw values collected by an IMU sensor worn on the lower back by a person with MS.
2	The input data should contain acceleration values in m/s^2 and angular velocity in deg/s for all x, y and z directions.
3	The input data must align with the expected coordinate system ***(explained further in the mockups).***
4	The samplnig rate (Hz), sensor height (m), patient height (m) and measurement setting (real world or laboratory) must be known.

2) Functional requirements

These requirements about actual functions that users can use and have been prioritised as follows:

* **Must have** = the basic functions of the system (the bare minimum that has to be implemented).
* **Should have** = extremely useful functions that are still integral, but might be complex to implement given time constraints. Only considered if there is enough time and all must-haves have been implemented.
* **Could have** = extra functions that improve the quality and usefulness of the system.

ALSO NOTE: "user" is equivalent to clinician (this is the main end-user).

Table of must-have functional requirements

#	Requirement	Priority	Additional comments
1	For a new gait analysis, the user is able to upload a CSV of sensor data and input other details (according to the data requirements) to automatically generate relevant, pre-defined data visualisations.	Must have	
2	There should be a section detailing important input data requirements to ensure reliable information about patient gait is extracted.	Must have	
3	Where applicable, the user is able to edit the settings of a given data visualisation (e.g. pick which set of records are visualised).	Must have	

Table of should-have functional requirements

4	The user is able to create their own account that will hold all their patient and gait analysis data.	Should have	All these should have's are functionalities involving adding data storage/persistence.
5	The user is able to create new patients to store their background information and gait analyses.	Should have	
6	The user is able to effectively filter through gait analyses and patients.	Should have	
7	The user is able to select subsets of data to compare (e.g. between average DMOs for current patient and other patients). There should be different views available for comparison (e.g. graphs or table of precise values).	Should have	For this to be possible, we need to also implement clinician user accounts (the other requirements)

Table of could-have functional requirements

8	A reference range should be provided with colour coding for DMO values to provide context for analysis (e.g. what is considered low vs. fast gait speed).	Could have	Requires research to verify valid reference ranges, don't want to provide clinicians with incorrect information.
9	Each visualisation can be exported into an image format and saved in the user's device.	Could have	Even though low priority, quite simple and likely to fulfil.
10	The user is able to create new, custom visualisations, outside of the automatically generated ones, that they seem useful analysis.	Could have	
11	There are useful infographics and posters conveying general information about MS and gait, that may aid analysis.	Could have	Might be useful as an efficient medium of conveying information.
12	The user is able to automatically transform their CSV data into the appropriate format. Specifically: converting into the correct units and coordinate system.	Could have	Currently partially in plans. Conversion from g to m/s^2 is in plans. However, allowing to shift into the correct coordinate system is quite a bit of extra work to accommodate. Currently prioritise clarifying how input file/data should look like and rely on users manipulating their data and supplying it as required.
13	For a given patient, the user can view a patient outcome dashboard/electronic health record summarising their current patient's gait condition and historical analyses.	Could have	Good for familarity as common conventions in healthcare, but considered additional work.
14	The user is a ble to organise the structure of visualisations.	Could have	Difficulty in ensuring legibility while moving things around.

3) Non-functional requirements

These are more quality-of-life requirements surrounding usability and satisfaction when using the system. More like considerations to keep in mind during actual implementation and coding.

Table of non-functional requirements

#	Requirement	Priority	Additional comments
1	Data uploads are optimised and as fast as possible.	Must have	
2	The interface is optimised for desktop screens.	Must have	Analysis makes more sense and is much easier on bigger desktop screens.
3	All information displayed is reliable and accurate for clinical analysis.	Must have	
4	All data is stored securely and can only be accessed by the right users.	Must have	
5	Navigation throughout and use of the system are easy, intutiive and assisted if necessary.	Must have	
6	Invalid interactions are correctly discouraged and the user is guided towards the right paths.	Must have	
7	The overall design and theme are modern, organised and aesthetic.	Should have	
8	Pages load quickly.	Should have	

STEP 2: Watch the video explaining mockups of the system.

Now that you have an overview of what the system should be able to do, please reinforce your understanding by going through the video explaining visual mockups of the system.

There are two videos that have been created. I am aware as clinicians time is limited, so I've created a video that is 5 minutes long and it skims through the mockups. You may then quickly skim through the design file to develop a quicker understanding on your own.

There is a longer video, that describes each mockup more thoroughly. Please note that this is **not mandatory** to watch, and is additional material to better your understanding if you feel that is required.

The design file as promised can be reached here.

STEP 3: Enter your feedback.

Please provide your feedback by answering the questions below. Please try to provide your own initial immediate thoughts - your first impressions. What comes to mind when you read the requirements or watched the video. There are a couple questions that are not marked as required, as it is understandable that you may not have an answer.

What are your thoughts on the functional requirements? Do you find any useful in particular? Is there anything that might be missing or useful to add? *

Your answer

Do you have any comments on the data requirements? Any suggested additions/omissions?

Your answer

Do you have any comments on the non-functional requirements? Any suggested additions/omissions?

Your answer

Would you make any adjustments to the current set of visualisations? If so, what changes would you make? *

Your answer

Are there any other visualisations that you would find particularly useful to add for analysing your patients' gait? *

Your answer

Are there any changes that you would make to any specific pages/features presented in the mockups? *

Your answer

How useful would you find this system in forming real, practical conclusions about your patients' gait/mobility? (1 - least useful, 10 - super duper useful) *

1 2 3 4 5 6 7 8 9 10

☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆ ☆

Can you please explain your rating above? *

Your answer

Anything else you would like to add?

Your answer

Appendix B

Iterative feedback consent form

CONSENT FORM: Please verify that you agree with all of the points of consent below.

If you answer "No" to any of the points below, please do not submit the questionnaire.

I have read and understood the participant information sheet dated 24/02/2025 or the project has been fully explained to me. (If you answer No to this question please do not fill out the rest of the form until you understand what taking part in the project will mean.) *

- ◯ Yes
- ◯ No

I have been given the chance to ask questions about the project. *

- ◯ Yes
- ◯ No

I understand how to raise a concern or make a complaint. *

- ◯ Yes
- ◯ No

I agree to take part in the project. I understand that taking part in the project will involve taking the time to understand project plans (will be described in the questionnaire) and providing feedback by filling in the questionnaire. *

- ◯ Yes
- ◯ No

I understand that by choosing to participate as a volunteer in this research, this does not create a legally binding agreement nor is it intended to create an employment relationship with the University of Sheffield. *

- ◯ Yes
- ◯ No

I understand that I can withdraw from the research/study at any time. I understand that I do not have to give any reasons for this and there will be no negative consequences if I choose to withdraw. *

- ◯ Yes
- ◯ No

I understand my identifying information such as name, email address etc. will not be seen by anyone outside the project. *

- ◯ Yes
- ◯ No

I understand and agree that my words/feedback alongside a vague identification of my background/experience may be described in the final dissertation report. I understand that I will not be named in these outputs. e.g. "a clinician that works with MS patients did not like the visualisations" *

- ◯ Yes
- ◯ No

Back Next Clear form

Appendix C

Iterative feedback questionnaire

24th February 2025

Building a Visualisation Interface for Gait Analysis in People with Multiple Sclerosis
Participant Information Form for Feedback Collection Questionnaire

Research team:

- **Dissertation project author:** Weng Wong (George) Hum (wwhum1@sheffield.ac.uk)
- **Dissertation project supervisor:** Vita Lanfranchi (v.lanfranchi@sheffield.ac.uk)

Invitation
Hi! We are inviting you to take part and provide feedback about a first iteration of a system for a final-year dissertation project. Before you decide if you would like to take part, it is important for you to understand what the project and this questionnaire are about. Read the following information carefully and feel free to ask if there is anything that you don't understand or if you would like more information. Take your time to decide whether you want to take part.

What is the project's purpose?
This is for an undergraduate final-year dissertation project at The University of Sheffield. The aim is to build a visualization interface/app that clinicians can use to visualize walking data collected by their patients that have multiple sclerosis (MS).

Mobility is deemed a high-priority issue for most patients with MS, and the ability to walk is a good representation of overall mobility. Patients can collect data on their day-to-day walking (gait) using a single inertial measurement unit (IMU) sensor worn on their lower back.

This project involves building a system where clinicians can upload data extracted from this sensor (in the form of a comma-separated-values (CSV) file). Then, useful gait parameters (also called digital mobility outcomes (DMOs)) (e.g. walking speed, cadence) are extracted from this data, and graphs and charts about these parameters are automatically created and presented to the clinician. The hope is that these visualisations are useful to the clinician for forming conclusions about their patient with MS' mobility and then they can offer tailored treatments accordingly.

The deadline for this dissertation project is mid-May.

What is this questionnaire for?
We are currently collecting feedback for a first iteration of requirements (planned features for the system) and mockups, to

ensure the final system is one that is useful to the target users (clinicians).

Why have I been chosen?
You have been chosen to offer feedback because you have clinical experience/knowledge working with patients that have multiple sclerosis. The plan is to collect feedback from a total of around three fitting participants.

Do I have to take part?
This questionnaire is entirely voluntary and there are no negative consequences to refusing to fill in the questionnaire. Additionally, if at any point in the process you wish to withdraw, then this is entirely possible. Please express your wish to withdraw in an email to George, and any sort of data collected from you will be destroyed.

It is important to note that by participating, you will not create a legally binding agreement or any form of employment with the University of Sheffield.

What will happen to me if I take part? What do I have to do?
The questionnaire will be sent to you, and you should aim to submit this form within five days. The overall process should not take more than 25-30 minutes. This aim is to ensure feedback is collected and adjustments can be made to the design and requirements in a timely manner.

The questionnaire will first ask that you give consent to being a part of this research. Then it will include step-by-step instructions on understanding the project plans and overall system. This will entail descriptions of the data, functional and non-functional requirements and how these will be prioritised. You will then watch a video that goes through mockups of the system, to help you better understand the overall flow of the interface and what you can do as a user. Then, a few questions (mostly free text) will follow, for you to express your opinions on the requirements and mockups.

What are the possible disadvantages and risks of taking part?
The only real disadvantage is that it may take some time to understand the project before being able to provide your opinions. This will however be mitigated by providing a clear view of the requirements and priorities, and a video explaining the mockups and the intentions behind them.

What are the possible benefits of taking part?
Whilst there are no immediate benefits for taking part, your feedback will be useful towards creating a final system that will

provide real value to clinicians when treating patients with MS.

Will my taking part in this project be kept confidential?
The questionnaire will simply collect your email address, so that we are able to identify your answers to you. Additionally, we ask that you specify your occupational background. These are only for us to understand who is submitting the feedback. No other personal data will be collected. However, only your opinions and occupational background will be used and summarized in the final dissertation report. These won't be explicitly linked to your identity. For example, "a clinician that works with MS patients did not find the visualizations useful."

Data protection
Responses collected through the questionnaire will only be accessible by the dissertation author (George) and supervisor (Vita). These will be destroyed within a month of the dissertation completion date.

The University of Sheffield will act as the Data Controller for this study. This means that the University is responsible for looking after your information and using it properly.

What happens to the results of the research?
You should be sent a copy of your responses. Your opinions will be considered, and necessary adjustments to the project requirements and mockups will be made accordingly.

Your submitted opinions will be summarized in the dissertation report, and they will not be linked back to your identity (e.g. name/email address), but a vague label of your occupational background may be used (e.g. a clinician that works with MS patients).

Who has ethically reviewed this project?
This project has been ethically approved by the School of Computer Science at the University of Sheffield.

What if something goes wrong and I wish to complain about the research or report a concern or incident?
If you have any questions about the research/questionnaire, please contact George (wwhum1@sheffield.ac.uk). If you would like to complain about any aspect of the research or if you feel that you have been exploited, abused or harmed as a result of taking part in the research, please contact Vita (v.lanfranchi@sheffield.ac.uk). If you feel your complaint has not been handled in a satisfactory way, you can contact the Head of School of Computer Science (Heidi Christensen; heidi.christenson@sheffield.ac.uk).

If you have concerns regarding how the personal data is being handled, information on how to raise a complaint can be found in the University's Privacy Notice.

Final comments
Please retain a copy of this information sheet for future reference. Thank you for taking part in the project!

Appendix D

Iterative feedback ethics approval

The University Of Sheffield.

Downloaded: 28/04/2025
Approved: 17/03/2025

George Hum
Registration number: 220132750
Computer Science
Programme: BSc Computer Science

Dear George

PROJECT TITLE: Building a Visualisation Interface for Gait Analysis in People with Multiple Sclerosis
APPLICATION: Reference Number 066644

On behalf of the University ethics reviewers who reviewed your project, I am pleased to inform you that on 17/03/2025 the above-named project was **approved** on ethics grounds, on the basis that you will adhere to the following documentation that you submitted for ethics review:

- University research ethics application form 066644 (form submission date: 13/03/2025); (expected project end date: 14/05/2025).
- Participant information sheet 1149638 version 6 (27/02/2025).
- Participant consent form 1149639 version 3 (26/02/2025).

The following amendments to this application have been approved:

- Amendment approved: 30/03/2025

If during the course of the project you need to deviate significantly from the above-approved documentation please inform me since written approval will be required.

Your responsibilities in delivering this research project are set out at the end of this letter.

Yours sincerely

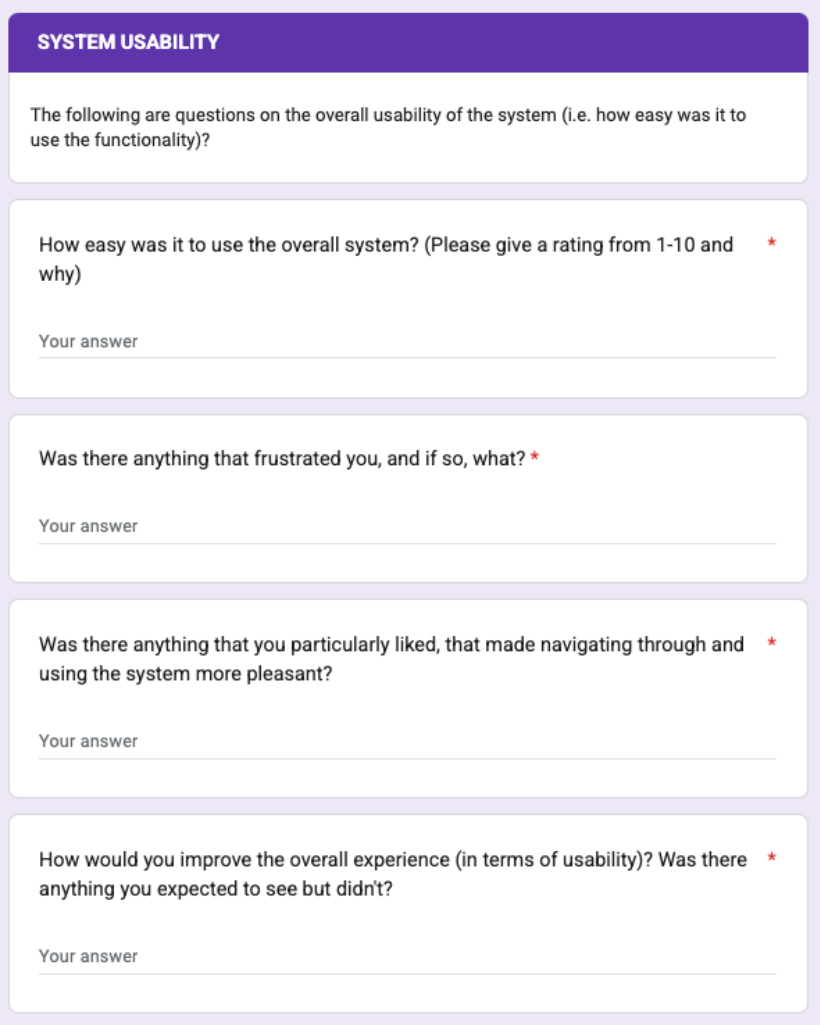
Luke Whitham
Ethics Admin
Computer Science

Please note the following responsibilities of the researcher in delivering the research project:

- The project must abide by the University's Research Ethics Policy: https://www.sheffield.ac.uk/research-services/ethics-integrity/policy
- The project must abide by the University's Good Research & Innovation Practices Policy: https://www.sheffield.ac.uk/polopoly_fs/1.671066!/file/GRIPPolicy.pdf
- The researcher must inform their supervisor (in the case of a student) or Ethics Admin (in the case of a member of staff) of any significant changes to the project or the approved documentation.
- The researcher must comply with the requirements of the law and relevant guidelines relating to security and confidentiality of personal data.
- The researcher is responsible for effectively managing the data collected both during and after the end of the project in line with best practice, and any relevant legislative, regulatory or contractual requirements.

Appendix E

Evaluation questionnaire



SYSTEM USABILITY

The following are questions on the overall usability of the system (i.e. how easy was it to use the functionality)?

How easy was it to use the overall system? (Please give a rating from 1-10 and why) *

Your answer

Was there anything that frustrated you, and if so, what? *

Your answer

Was there anything that you particularly liked, that made navigating through and using the system more pleasant? *

Your answer

How would you improve the overall experience (in terms of usability)? Was there anything you expected to see but didn't? *

Your answer

HOW WELL WERE THE PROJECT REQUIREMENTS MET?

A list of the attempted project requirements will follow. Please rate (1-10) how well the system achieves each of these requirements and provide a short explanation for your answer.

For a new gait analysis, the user is able to upload a CSV of sensor data and input other details (according to the data requirements) to automatically generate relevant, pre-defined data visualisations. *

Your answer

There should be a section detailing important input data requirements to ensure reliable information about patient gait is extracted. *

Your answer

Where applicable, the user is able to edit the settings of a given data visualisation (e.g. pick which set of records are visualised). *

Your answer

The user is able to select subsets of data to compare (e.g. between average DMOs for current patient and other patients). There should be different views available for comparison (e.g. graphs or table of precise values). *

Your answer

The user is able to automatically transform their CSV data into the appropriate format. Specifically: converting into the correct units and coordinate system. *

Your answer

Data uploads are optimised and as fast as possible. *

Your answer

The interface is optimised for desktop screens. *

Your answer

All information displayed is reliable and accurate for clinical analysis. *

Your answer

Navigation throughout and use of the system are easy, intutiive and assisted if necessary. *

Your answer

Invalid interactions are correctly discouraged and the user is guided towards the right paths. *

Your answer

The overall design and theme are modern, organised and aesthetic. *

Your answer

Pages load quickly. *

Your answer

USER ACCEPTANCE/SYSTEM EFFECTIVENESS

The following questions are on the overall success of the system - how useful would it be in the real world?

Were there any visualisations that you found particularly useful in forming useful conclusions about patient gait? *

Your answer

Were there any visualisations that you thought were unnecessary or not useful at all? *

Your answer

Are there any types of visualisations you would add to expand the possible conclusions that could be made about patient gait? *

Your answer

How likely are you to use this system to analyse MS patient gait? (Please give a rating from 1-10 and explain why) *

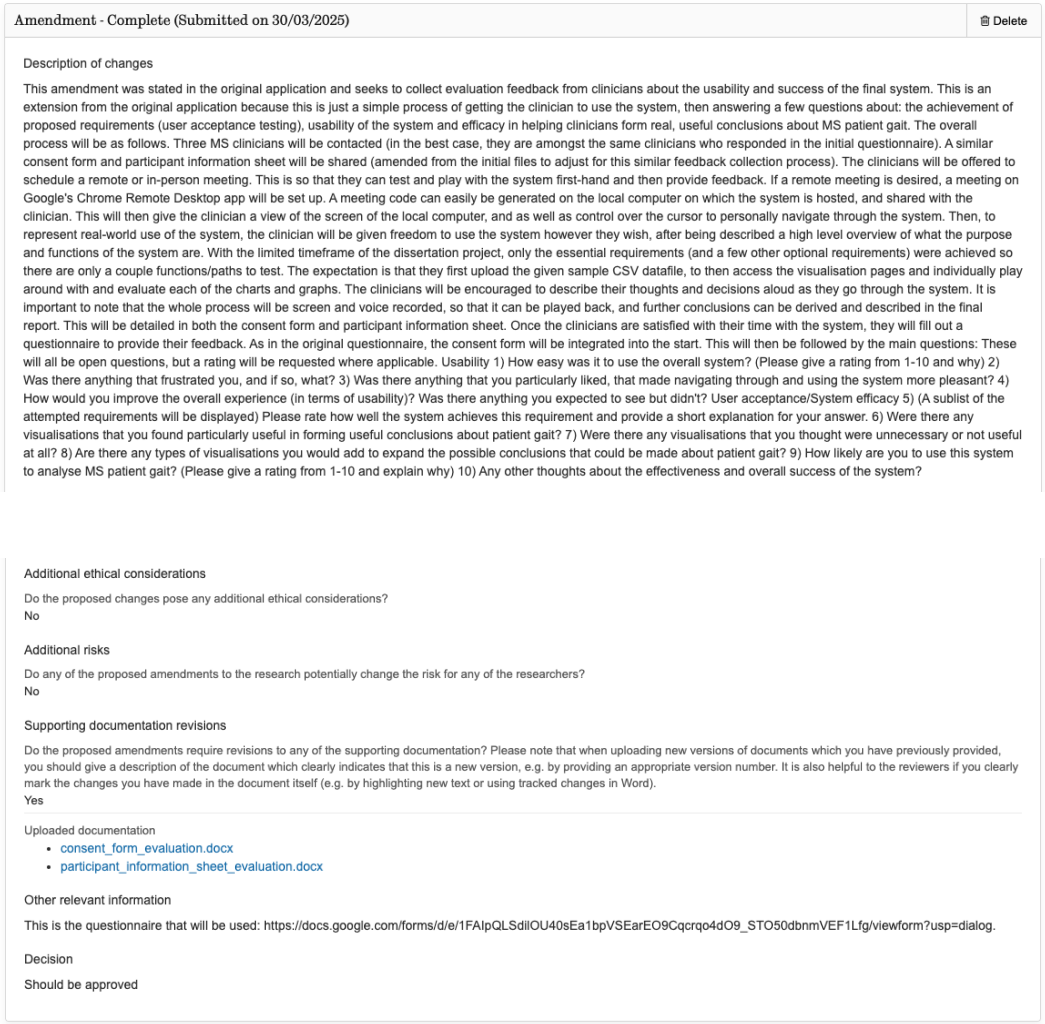
Your answer

Any other thoughts about the effectiveness and overall success of the system? *

Your answer

Appendix F

Ethics amendment for evaluation



Amendment - Complete (Submitted on 30/03/2025)

Delete

Description of changes

This amendment was stated in the original application and seeks to collect evaluation feedback from clinicians about the usability and success of the final system. This is an extension from the original application because this is just a simple process of getting the clinician to use the system, then answering a few questions about: the achievement of proposed requirements (user acceptance testing), usability of the system and efficacy in helping clinicians form real, useful conclusions about MS patient gait. The overall process will be as follows. Three MS clinicians will be contacted (in the best case, they are amongst the same clinicians who responded in the initial questionnaire). A similar consent form and participant information sheet will be shared (amended from the initial files to adjust for this similar feedback collection process). The clinicians will be offered to schedule a remote or in-person meeting. This is so that they can test and play with the system first-hand and then provide feedback. If a remote meeting is desired, a meeting on Google's Chrome Remote Desktop app will be set up. A meeting code can easily be generated on the local computer on which the system is hosted, and shared with the clinician. This will then give the clinician a view of the screen of the local computer, and as well as control over the cursor to personally navigate through the system. Then, to represent real-world use of the system, the clinician will be given freedom to use the system however they wish, after being described a high level overview of what the purpose and functions of the system are. With the limited timeframe of the dissertation project, only the essential requirements (and a few other optional requirements) were achieved so there are only a couple functions/paths to test. The expectation is that they first upload the given sample CSV datafile, to then access the visualisation pages and individually play around with and evaluate each of the charts and graphs. The clinicians will be encouraged to describe their thoughts and decisions aloud as they go through the system. It is important to note that the whole process will be screen and voice recorded, so that it can be played back, and further conclusions can be derived and described in the final report. This will be detailed in both the consent form and participant information sheet. Once the clinicians are satisfied with their time with the system, they will fill out a questionnaire to provide their feedback. As in the original questionnaire, the consent form will be integrated into the start. This will then be followed by the main questions: These will all be open questions, but a rating will be requested where applicable. Usability 1) How easy was it to use the overall system? (Please give a rating from 1-10 and why) 2) Was there anything that frustrated you, and if so, what? 3) Was there anything that you particularly liked, that made navigating through and using the system more pleasant? 4) How would you improve the overall experience (in terms of usability)? Was there anything you expected to see but didn't? User acceptance/System efficacy 5) (A sublist of the attempted requirements will be displayed) Please rate how well the system achieves this requirement and provide a short explanation for your answer. 6) Were there any visualisations that you found particularly useful in forming useful conclusions about patient gait? 7) Were there any visualisations that you thought were unnecessary or not useful at all? 8) Are there any types of visualisations you would add to expand the possible conclusions that could be made about patient gait? 9) How likely are you to use this system to analyse MS patient gait? (Please give a rating from 1-10 and explain why) 10) Any other thoughts about the effectiveness and overall success of the system?

Additional ethical considerations

Do the proposed changes pose any additional ethical considerations?
No

Additional risks

Do any of the proposed amendments to the research potentially change the risk for any of the researchers?
No

Supporting documentation revisions

Do the proposed amendments require revisions to any of the supporting documentation? Please note that when uploading new versions of documents which you have previously provided, you should give a description of the document which clearly indicates that this is a new version, e.g. by providing an appropriate version number. It is also helpful to the reviewers if you clearly mark the changes you have made in the document itself (e.g. by highlighting new text or using tracked changes in Word).
Yes

Uploaded documentation

- consent_form_evaluation.docx
- participant_information_sheet_evaluation.docx

Other relevant information

This is the questionnaire that will be used: https://docs.google.com/forms/d/e/1FAIpQLSdilOU40sEa1bpVSEarEO9Cqcrqo4dO9_STO50dbnmVEF1Lfg/viewform?usp=dialog.

Decision

Should be approved

Appendix G

Evaluation consent form (modified)

I agree to take part in the project. I understand that taking part in the project will involve taking the time to schedule a meeting, use the system and provide feedback by filling in the questionnaire. *

- ◯ Yes
- ◯ No

I understand that the meeting and my use of the system will be voice and screen recorded for reference when writing about the system's evaluation in the final dissertation report.

- ◯ Yes
- ◯ No

Appendix H

Evaluation participant information form (modified)

Why have I been chosen?
You have been chosen to offer feedback because you have clinical experience/knowledge working with patients that have multiple sclerosis. The plan is to collect feedback from a total of around three fitting participants.

You may have also provided feedback in the questionnaire for the first iteration of the system, in which case you can also help assess the success of the incorporation of your feedback.

What will happen to me if I take part? What do I have to do?
You can choose to schedule either a remote meeting or an in-person meeting with George at a location at/around The University of Sheffield.

If you choose to do a remote meeting, at the start of this meeting, you will be given a code to enter into Chrome Remote Desktop. This is because the system is not deployed on the web and is hosted on a local computer. Chrome Remote Desktop will show you the screen of the local computer, and as well as give you mouse control to navigate through and use the final system.

If you choose to do an in-person meeting, you will be given access to the local computer which hosts the system.

All you need to know beforehand are the project's purpose (above), and that final system essentially just generates gait-related visualisations from input walking data that you give it. For this, a sample CSV file and relevant information for inputs will be provided.

You will then be given the freedom to navigate and use the final system as you wish. It is encouraged that you describe your

thoughts and decisions out loud as you go through the system. It is important that your process through the system will be screen and voice recorded so that it can be watched back to derive conclusions for the final dissertation report.

Once you are satisfied with your time with the system, you will fill out a questionnaire to provide your feedback about the system.

Appendix I

User evaluation results

Category	Points
System usability	• Given average score of 8. • Intuitive and information is sufficient to understand how to use. • The interface is modern and clean. • Colours are easy to see. • It might be useful to pre-fill/pre-select mandatory fields with expected data. • Colour-blind friendly colours should be used. • Might be useful to redirect to summary page after form submission. • Lots of text is off-putting. Consider a user manual or training stage? • Consider mobile/tablet views in the future. • Most clinicians will just tap around (trial and error) to learn rather than read the information. • Lots of WBs if the recording is large. These are hard to filter through.

Meeting project requirements and scores / 10	• Functional requirement 1: 7, 8 • Functional requirement 2: 7, 8 • Functional requirement 3: 7, 8 • Functional requirement 7: 7, 9 • Functional requirement 12: 8, 9 • Non-functional requirement 1: 7, 9 • Non-functional requirement 2: 9, 9 • Non-functional requirement 4: 8, 9 • Non-functional requirement 6: 8, 9 • Non-functional requirement 7: 9, 9 • Non-functional requirement 8: 8, 9 • Non-functional requirement 9: 8, 10

System effectiveness	• Data input/form submission should be done before clinic, by healthcare assistants. Clinicians shouldn't see this screen. • Need simple overview screen as first results page (1-2 minute read). Then have a "show more" or advanced settings option. • Labelling of left and right strides is useful for examining differences between left and right limbs. • Need timestamps on walking bouts to identify which are actually useful. • Need reference ranges to determine which values are healthy/unhealthy. There is ongoing research on this. • PCP/scatter plot shows useful relationships between parameters. Particularly links to walking speed which usually has knock-on effects on other parameters. • Radar chart was used with pwMS in clinic-based assessments in the past. Found that could see patterns in disease severity. • It might be useful to co-design with clinicians to tighten axes orders (what generally makes the most sense). • Need for storage of more analyses to perform longitudinal analysis. • Overlap between progression bar and scatter graphs. No need for both. Scatter might be an easier read. • Individual strides/bouts are not really useful. Weekly/daily averages (higher level) or trends over a week are better (longitudinal analysis with averages). • Ability to filter out WBs of specific length, or certain time of day or day of the week. • Need to be able to work with real, larger datasets, to be able to properly evaluate clinical usefulness. • Inclusion of other DMOs e.g. on pace, rhythm and variability. • Link indicators like EDSS with visualisations. • Limits to overlapping shapes and plots depend on context. • Integration of visualisations with electronic patient records. • Given scores 8 and 9 on likeliness of using the system. More work is needed to implement into practice but the project is a good start.

Table I.1: Points received from evaluation feedback questionnaire about final implemented system