**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Topic** | **Page No** |
| **1** | **Introduction** | 2 |
| 1.1 | Purpose | 2 |
| 1.2 | Scope | 2 |
| 1.3 | Definitions, acronyms, and abbreviations |  |
| **2** | **System architecture** |  |
| 2.1 | Data Collection Components | 3 |
| 2.2 | Data Preprocessing Component | 3 |
| 2.3 | Feature Extraction Components | 3 |
| 2.4 | Gait Classification Component | 3 |
| 2.5 | Inherent Attribute Identification Component | 3 |
| 2.6 | User Interface Component | 4 |
| 2.7 | Database Component | 4 |
| 2.8 | Communication Component | 4 |
| 2.9 | System Integration Component | 4 |
| **3** | **System Testing** | 4 |
| **4** | **Functional Requirement** |  |
| 4.1 | Data Collection | 6 |
| 4.2 | Data Preprocessing | 6 |
| 4.3 | Feature Extraction | 6 |
| 4.4 | Gait Classification | 6 |
| 4.5 | Inherent Attribute Identification | 6 |
| 4.6 | Machine Learning Algorithm | 6 |
| 4.7 | Real-Time Processing | 7 |
| 4.8 | User Interface | 7 |
| 4.9 | Database Management | 7 |
| 4.10 | Performance Metrics | 7 |
| 4.11 | Security | 7 |
| 4.12 | Scalability | 7 |
| 4.13 | Compatibility | 8 |
| 4.14 | Documentation | 8 |
| **5** | **Non-Functional Requirements** |  |
| 5.1 | Performance | 8 |
| 5.2 | Accuracy | 8 |
| 5.3 | Reliability | 8 |
| 5.4 | Security | 8 |
| 5.5 | Usability | 9 |
| 5.6 | Compatibility | 9 |
| 5.7 | Maintability | 9 |
| 5.8 | Portability | 9 |
| 5.9 | Performance Scalability | 9 |
| 5.10 | Cost | 9 |

**1.Introduction:**

**1.1 Purpose**

Gait analysis has become an essential tool for assessing human movement in various fields such as rehabilitation, sports medicine, and biomechanics research. The identification of gait patterns and inherent attributes such as step length, cadence, and gait symmetry can provide valuable information for clinical decision-making and injury prevention.Wearable sensors such as inertial measurement units (IMUs) placed on the ankles have been widely used to collect kinematic data during gait analysis. The ankle joint plays a crucial role in gait, and kinematic data collected from the ankle joint can provide insights into the gait patterns and inherent attributes of individuals.Gait classification involves the categorization of gait patterns into different types such as normal gait, limping gait, shuffling gait, and toe walking. Inherent attribute identification involves the measurement of gait parameters such as step length, cadence, and gait symmetry.In this context, this project proposes a system for Gait Classification with Gait Inherent Attribute Identification from Ankle's Kinematics. The system will utilize wearable sensors placed on the ankles to collect kinematic data during walking or running. The data will then be pre-processed, and relevant features will be extracted to identify gait patterns and inherent attributes using machine learning algorithms. The proposed system has the potential to provide valuable information for clinical decision-making in rehabilitation and sports medicine. It can also be used for biomechanics research to better understand human movement and improve injury prevention strategies.

**1.2 Scope**

The scope of this system is to collect, preprocess, extract features, classify gait types, identify inherent attributes, and provide a user interface for users to interact with the system. The system will be designed to work with data collected from wearable sensors attached to the ankles of individuals during walking or running.

**1.3 Definitions, acronyms, and abbreviations**

**SRS**: Software Requirement Specification

**Gait**: A person's manner of walking or running

**Wearable sensors:** Devices that can be worn on the body to collect data

**Kinematics:** The study of motion without considering the forces that cause the motion

**Machine learning:** A type of artificial intelligence that allows a computer system to learn from data without being explicitly programmed.

**2. System architecture:**

**2.1Data Collection Component:**

This component is responsible for acquiring data from wearable sensors placed on the ankles of individuals during walking or running. The sensors should be able to capture ankle kinematics such as joint angles, angular velocity, and acceleration. The data collection component should provide a mechanism for calibrating the sensors before data collection.

**2.2Data Preprocessing Component:**

This component is responsible for performing a range of preprocessing tasks to ensure that the data is suitable for analysis. This should include filtering the data to remove noise and artifacts, resampling the data to ensure uniform time intervals between data points, and segmenting the data into individual strides.

**2.3Feature Extraction Component:**

This component is responsible for extracting relevant features from the preprocessed data. The features should be related to gait classification and inherent attribute identification. Examples of features include joint angle range, stride length, and foot strike pattern. The system should use well-established feature extraction techniques that have been shown to be effective in previous research.

**2.4Gait Classification Component:**

This component is responsible for classifying different gait types using machine learning algorithms. The system should use supervised learning techniques to train the classification model using a labeled dataset. The gait types to be classified should include normal gait, limping gait, shuffling gait, and toe walking. The system should be able to handle real-time classification of gait.

**2.5Inherent Attribute Identification Component:**

This component is responsible for identifying inherent gait attributes such as cadence, step length, and gait symmetry using machine learning algorithms. The system should use well-established machine learning techniques to identify these attributes. The module should be able to handle real-time identification of gait attributes.

**2.6User Interface Component:**

This component is responsible for providing an easy-to-use interface that allows users to input data, visualize results, and export data in different formats. The system should provide visual feedback to users to help them understand the results of the gait classification and inherent attribute identification.

**2.****7Database Component:**

This component is responsible for storing the data collected from the wearable sensors, preprocessed data, extracted features, and results of gait classification and inherent attribute identification. The database component should be able to handle large datasets and provide secure storage for user data.

**2.8Communication Component:**

This component is responsible for establishing communication between the wearable sensors and the system. The communication component should use secure communication protocols to ensure that data transmitted between the sensors and the system is encrypted.

**2.9System Integration Component:**

This component is responsible for integrating all the other components of the system architecture to achieve the overall system goal. The system integration component should ensure that the system is reliable, accurate, and easy to use. It should also ensure that the system meets the non-functional requirements such as performance, usability, and security.

**3 System Testing:**

1. **Unit testing:** This involves testing individual modules or components of the system to ensure that they work as intended. This can be done using automated testing frameworks.
2. **Integration testing:** This involves testing the integration of different modules or components of the system to ensure that they work together seamlessly.
3. **System testing:** This involves testing the system as a whole to ensure that it meets the functional and non-functional requirements. This can be done using both manual and automated testing.
4. **Performance testing:** This involves testing the performance of the system under different loads and stress conditions to ensure that it can handle the expected user traffic and data volumes.
5. **Security testing:** This involves testing the security of the system to ensure that it is resistant to attacks and unauthorized access. This can be done using tools such as penetration testing and vulnerability scanning.
6. **User acceptance testing:** This involves testing the system with actual users to ensure that it meets their needs and is easy to use.
7. **Regression testing:** This involves testing the system after making changes or updates to ensure that it still functions as intended and that new features or changes do not introduce new bugs or issues.
8. **Usability testing**: This involves testing the user interface of the system to ensure that it is intuitive and easy to use.
9. **Compatibility testing:** This involves testing the system on different hardware and software configurations to ensure that it is compatible with a wide range of systems.
10. **Maintenance testing:** This involves testing the system after maintenance or updates to ensure that it continues to function as intended and that no new issues have been introduced.

**4 Functional Requirements:**

**4.1Data collection**:

The system should be able to collect kinematic data from wearable sensors with a sampling frequency of at least 100 Hz. The sensors should be lightweight, non-invasive, and waterproof to ensure they do not interfere with natural gait patterns during walking or running.

**4.2Data preprocessing:**

The system should be able to remove noise and artifacts from the raw data, using techniques such as bandpass filtering and signal detrending. The data should be segmented into individual strides based on identifiable gait events such as heel strikes and toe-offs.

**4.3Feature extraction:**

The system should be able to extract relevant features from the preprocessed data, such as joint angles, angular velocities, and accelerations during different phases of gait. These features can be computed using standard algorithms such as inverse kinematics and signal processing techniques.

**4.4Gait classification:**

The system should use machine learning algorithms such as support vector machines (SVMs) or artificial neural networks (ANNs) to classify gait patterns into different types. The system should be able to recognize common gait deviations such as limping, shuffling, and toe walking.

**4.5Inherent attribute identification:**

The system should be able to calculate inherent gait attributes such as step length, cadence, and gait symmetry based on the extracted features. These attributes can provide valuable insights into gait quality and risk for falls.

**4.6Machine learning algorithms**:

The system should be able to use a variety of machine learning algorithms to classify gait patterns and identify inherent attributes, including supervised learning, unsupervised learning, and reinforcement learning. The system should be able to learn from new data and adapt to different gait patterns.

**4.7Real-time processing:**

The system should be able to perform gait classification and inherent attribute identification in real-time, with a latency of less than 100 ms to ensure the system can be used in real-world settings such as clinical or sports environments.

**4.8User interface:**

The system should have an intuitive user interface that allows users to input data, visualize results, and export data in different formats such as CSV or JSON. The interface should provide real-time feedback and allow users to customize the settings of the system such as the classification algorithm or feature extraction methods.

**4.9Database management:**

The system should be able to store the data collected from the wearable sensors, preprocessed data, extracted features, and results of gait classification and inherent attribute identification in a secure and organized manner. The system should also provide tools for data cleaning and data visualization to facilitate data exploration.

**4.10Performance metrics:**

The system should provide performance metrics such as accuracy, precision, recall, and F1-score to evaluate the performance of the classification and identification algorithms. The system should also provide visualizations such as confusion matrices and ROC curves to facilitate performance evaluation.

**4.11Security:**

The system should ensure the security and privacy of user data by using encryption and access control mechanisms. The system should also comply with relevant data privacy regulations such as GDPR or HIPAA.

**4.12Scalability:**

The system should be able to handle large datasets and scale to meet the needs of different users, such as healthcare providers, researchers, or athletes. The system should also be able to integrate with other systems such as electronic health records or wearable sensor platforms.

**4.13Compatibility:**

The system should be compatible with different types of wearable sensors and operating systems, such as Android or iOS. The system should also be able to integrate with different types of data storage systems such as SQL or NoSQL databases.

**4.14Documentation:**

The system should have comprehensive documentation that includes user manuals, system requirements, and technical specifications. The documentation should also provide

**5.Non-Functional Requirements:**

**5.1Performance:**

The system should be able to process and classify gait patterns in real-time with minimal latency. The system should be able to handle large datasets and scale to meet the needs of different users.

**5.2Accuracy:**

The system should provide accurate gait classification and inherent attribute identification results with minimal errors. The system should be able to handle noisy data and adapt to different gait patterns.

**5.3Reliability:**

The system should be reliable and available at all times, with minimal downtime or system failures. The system should be able to recover from failures quickly and without data loss.

**5.4Security:**

The system should ensure the security and privacy of user data by using encryption and access control mechanisms. The system should comply with relevant data privacy regulations such as GDPR or HIPAA.

**5.5Usability:**

The system should have an intuitive user interface that is easy to use and navigate. The interface should provide real-time feedback and allow users to customize the settings of the system such as the classification algorithm or feature extraction methods.

**5.6Compatibility:**

The system should be compatible with different types of wearable sensors and operating systems, such as Android or iOS. The system should also be able to integrate with different types of data storage systems such as SQL or NoSQL databases.

**5.7Maintainability:**

The system should be easy to maintain and update, with clear documentation and code structure. The system should be modular and allow for easy modification or expansion.

**5.8Portability:**

The system should be easily deployable on different platforms and environments, such as cloud services or local servers. The system should also be able to run on different hardware configurations.

**5.9Performance scalability:**

The system should be designed to handle increasing amounts of data and user traffic without a significant drop in performance. The system should be able to scale horizontally or vertically depending on the needs of the user.

**5.10Cost:**

The system should be cost-effective to develop, deploy, and maintain. The system should use open-source software components and infrastructure where possible to reduce costs.