**Software Requirements**

**Specification**

**for**

# Yoga Pose Detection System

**Version 1.0 approved**

**Prepared by Shubham Pochhali**

**Parthsarthi Gupt**

**Divyam**

**Vivekanand Institute of Professional Studies**

**Oct 24, 2024**

***Software Requirements Specification for YPDS Page i***

## Table of Contents

**Table of Contents ................................................................................................................................... 2**

**Revision History ..................................................................................................................................... 2**

### Table of Contents

1. *Introduction* 
   1. *Purpose*
   2. *Document Conventions*
   3. *Intended Audience and Reading Suggestions*
   4. *Product Scope*
   5. *References*
2. *Overall Description* 
   1. *Product Perspective*
   2. *Product Functions*
   3. *User Classes and Characteristics*
   4. *Operating Environment*
   5. *Design and Implementation Constraints*
   6. *User Documentation*
   7. *Assumptions and Dependencies*
3. *External Interface Requirements* 
   1. *User Interfaces*
   2. *Hardware Interfaces*
   3. *Software Interfaces*
   4. *Communications Interfaces*
4. *System Features* 
   1. *Pose Detection*
   2. *Real-Time Feedback*
   3. *User Pose Data Storage*
5. *Other Nonfunctional Requirements* 
   1. *Performance Requirements*
   2. *Safety Requirements*
   3. *Security Requirements*
   4. *Software Quality Attributes*
6. *Other Requirements* 
   1. *Legal and Regulatory Compliance*
   2. *Data Management*
   3. *Internationalization*

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Shubham Pochhali *Parthsarthi Gupt Divyam* | 24th Oct,  2024 | This Version sets the foundation, defining core functions and user requirements for the Yoga Poses Detection System | 1.0 |
|  |  |  |  |

### 1. Introduction

#### 1.1 Purpose

The purpose of this document is to define the software requirements for the Yoga Pose Detection System. This system enables real-time yoga pose detection and corrective feedback to users, facilitating accurate posture without a live trainer.

#### 1.2 Document Conventions

This document follows the IEEE Software Requirements Specification template. Each requirement has a unique identifier, ensuring traceability. "TBD" serves as a placeholder for information that is pending or not yet available.

#### 1.3 Intended Audience and Reading Suggestions

This document is intended for software developers, project managers, and stakeholders in the field of fitness and health technology. It is recommended to read the document sequentially, starting with the Introduction, followed by the Overall Description, System Features, External Interface Requirements, and Nonfunctional Requirements.

#### 1.4 Product Scope

The Yoga Pose Detection System aims to identify and analyze user poses, providing real-time feedback to enhance yoga practice. Key features include yoga pose recognition, feedback generation, and historical data storage.

**1.5 References** • IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications

* Research papers and articles on pose detection and computer vision (e.g., "Yoga Pose Detection and Classification Using Deep Learning," IJISRT)
* OpenPose documentation and developer resources

### 2. Overall Description

#### 2.1 Product Perspective

The Yoga Pose Detection System is a standalone application utilizing computer vision to detect key body points and analyze yoga poses. Built with OpenPose and OpenCV, the system integrates with Django for backend processes and stores data in a MySQL database.

#### 2.2 Product Functions

* ***Pose Detection****: Identifies specific yoga poses using machine learning algorithms.*
* ***Real-Time Feedback****: Provides corrective feedback on user poses.*
* ***Pose Data Storage****: Maintains a record of previous poses and feedback for progress tracking.*

#### 2.3 User Classes and Characteristics

* ***Practitioners****: Individuals performing yoga independently without an instructor.*
* ***Fitness Enthusiasts****: Users interested in improving their yoga technique and alignment.*

#### 2.4 Operating Environment

* ***Hardware****: Desktop or laptop with a webcam.*
* ***Software****: Operating systems compatible with OpenCV and OpenPose libraries, such as Windows 7+ or equivalent Linux distributions.*

#### 2.5 Design and Implementation Constraints

The system relies on OpenPose for pose detection and requires sufficient lighting and camera resolution for optimal performance. It must meet hardware specifications, including a minimum of 4 GB RAM and i3 processor.

#### 2.6 User Documentation

Documentation will include user manuals, online help resources, and tutorials to guide users on system functionality, including uploading images or using live video for pose detection.

#### 2.7 Assumptions and Dependencies

The system assumes a stable internet connection and access to camera hardware. Its performance depends on suitable lighting conditions and camera quality.

### 3. External Interface Requirements

#### 3.1 User Interfaces

The system interface allows users to:

* *Upload an image or stream live video for pose detection.*
* *Receive visual and textual feedback on their poses.*
* *View past performance data for self-assessment.*

**3.2 Hardware Interfaces**

The system interfaces with webcams to capture user poses and transmits data to the server for analysis.

#### 3.3 Software Interfaces

The system integrates with OpenPose and OpenCV for image processing and uses Django as the backend framework, interacting with a MySQL database for data storage.

#### 3.4 Communications Interfaces

The system requires secure protocols for data transmission between client applications and the server, especially for pose data and feedback.

### 4. System Features

#### 4.1 Pose Detection

* ***Description and Priority****: The system detects specific yoga poses based on user input (image or video). Priority: High.*
* ***Stimulus/Response Sequences****:* 
  1. *User submits a pose through image or video.*
  2. *System identifies and displays detected pose.*
* ***Functional Requirements****:* 
  + *FR-1: The system shall accurately detect eight predefined yoga poses.*
  + *FR-2: The system shall use OpenPose to identify key points on the body for pose classification.*

#### 4.2 Real-Time Feedback

* ***Description and Priority****: Provides feedback based on pose accuracy. Priority: High.*
* ***Stimulus/Response Sequences****:* 
  1. *User holds a pose in front of the camera.*
  2. *System analyzes pose and provides corrective feedback.*
* ***Functional Requirements****:* 
  + *FR-3: The system shall generate corrective feedback within 2-3 seconds of pose detection.*
  + *FR-4: The system shall include both visual and textual feedback to assist with pose correction.*

#### 4.3 User Pose Data Storage

* ***Description and Priority****: Stores user pose data and feedback for future reference. Priority:*

*Medium.*

* ***Stimulus/Response Sequences****:* 
  1. *User completes a session with multiple poses.*
  2. *System stores data for later review.*
* ***Functional Requirements****:*

o *FR-5: The system shall record each detected pose along with feedback.* o *FR-6: The system shall allow users to review their past poses and feedback.*

### 5. Other Nonfunctional Requirements

#### 5.1 Performance Requirements

* ***Real-time Processing****: The system shall process images or videos within 2-3 seconds for realtime feedback.*
* ***Accuracy****: The system shall have at least a 90% accuracy rate in identifying yoga poses.*

#### 5.2 Safety Requirements

* ***User Safety****: The system should ensure accurate feedback to prevent physical harm due to incorrect poses.*
* ***Error Handling****: The system should handle errors gracefully, alerting the user if a pose cannot be detected.*

#### 5.3 Security Requirements

* ***Data Security****: All user data should be securely stored in the database with encryption where necessary.*
* ***Access Control****: Only authenticated users should be able to access their session history.*

#### 5.4 Software Quality Attributes

* ***Reliability****: The system should maintain an uptime of 99%, ensuring continuous availability.*
* ***Usability****: The user interface should be intuitive, enabling easy navigation.*
* ***Maintainability****: The system should allow easy updates to accommodate additional yoga poses or adjustments in feedback algorithms.*

### 6. Other Requirements

**6.1 Legal and Regulatory Compliance**

• *The system shall comply with relevant privacy laws regarding user data storage and handling.*

#### 6.2 Data Management

• *The system shall implement regular data backups and provide recovery options to ensure data integrity and accessibility.*

#### 6.3 Internationalization

• *The system shall be developed to support multiple languages, allowing users worldwide to access the application in their preferred language.*

### *Conclusion*

*The Yoga Pose Detection System is designed to improve yoga practice for individuals by offering realtime pose detection and corrective feedback. This SRS document outlines the functional and*

*nonfunctional requirements essential for building a reliable, efficient, and user-friendly system. The development of this system will allow users to practice yoga safely and effectively without needing an inperson instructor.*