Athletix

A Fitness Intelligence Technology

Software Requirements Specification (SRS)

Team Members:

Khawaja Mohammad Abdullah Ishaq	22P-9375
Azhan Shoaib	22P-9054
Subhan Tariq	22P-9067

Supervised By:

Dr. Muhammad Amin

October 21, 2025

Contents

1 Introduction 2

1.1 Purpose 2

1.2 Scope 2

1.3 Definitions, Acronyms, and Abbreviations 2

1.4 References 3

2 Overall Description 4

2.1 Product Perspective 4

2.2 Product Functions 4

2.3 User Characteristics 5

2.4 Constraints 5

2.5 Assumptions and Dependencies 5

3 Specific Requirements 6

3.1 Functional Requirements 6

3.2 Non-Functional Requirements 6

3.3 Software Interfaces 6

3.4 Hardware Interfaces 7

3.5 Communication Interfaces 7

4 System Design Constraints 7

5 Future Enhancements 7

6 Appendices 7

6.1 Tools & Technologies 7

1 Introduction

1.1 Purpose

The purpose of this document is to define the software requirements for **Athletix – Fitness Intelligence Technology**, a cross-platform AI-powered mobile application that serves as a Virtual Fitness Trainer. It provides personalized workouts, culturally adaptable diet plans, and real-time form analysis using Computer Vision, Machine Learning, Reinforcement Learning, and Natural Language Processing.

The document serves as a reference for the development team, stakeholders, and evaluators to ensure a clear understanding of project objectives, functional requirements, and system behavior.

1.2 Scope

Athletix aims to bridge the gap between traditional personal training and AI-driven digital health solutions by providing:

- Real-time posture correction and exercise analysis via HRNet-based pose estimation.
- Adaptive workout planning and diet generation based on user goals, body attributes, and preferences.
- Reinforcement learning-enabled personalization that evolves as the user progresses.
- Conversational AI trainer (RAG + fine-tuned LLM) for guidance, exercise substitutions, and recovery tips.
- Culturally and medically adaptable diet planning (e.g., halal, diabetic, vegan options).
- Progress tracking and gamified visualization for engagement.

1.3 Definitions, Acronyms, and Abbreviations

Term	**Description**
AI	Artificial Intelligence
ML	Machine Learning
CV	Computer Vision
RL	Reinforcement Learning
LLM	Large Language Model
RAG	Retrieval-Augmented Generation
FAISS	Facebook AI Similarity Search (Vector Database)
HRNet	High-Resolution Network (Pose Estimation)
TFLite	TensorFlow Lite (for mobile inference)

1.4 References

1. Lewis et al. (2020), *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*, NeurIPS.

2. Ahmad et al. (2022), *Mobile Apps for Human Nutrition: A Review.*

3. Riccio et al. (2021), *Real-Time Fitness Exercise Classification and Counting from Video Frames.*

4. United States Department of Agriculture (2023), *FoodData Central.*

5. World Health Organization (2024), *Physical Activity Guidelines.*

6. Expert Consultation: Dr. Znair Azam – Certified Fitness Trainer & Nutritionist.

2 Overall Description

2.1 Product Perspective

Athletix is an intelligent fitness assistant that operates as a standalone mobile app. It integrates:

- **Frontend:** React Native (cross-platform)
- **Backend:** FastAPI with PostgreSQL & FAISS vector DB
- **AI Modules:** TensorFlow Lite, PyTorch, MediaPipe/HRNet, and Hugging Face Transformers

The system interacts with:

- Camera for real-time pose detection
- Nutrition databases (USDA API)
- Local data storage and cloud backup for user profiles and progress

2.2 Product Functions

1. **User Authentication:** Secure signup/login with profile setup.
2. **User Profiling:** Collect body attributes (height, weight, age, gender, fitness goal, medical/cultural constraints).
3. **Form Analysis Module:**
   - Detect exercise form using HRNet keypoints.
   - Compare with biomechanics-based correct pose templates.
   - Provide instant feedback and correction suggestions.
4. **Workout Planner:**
   - Generate customized workout plans using ML models based on user profile and goal.
   - Reinforcement Learning adapts plans weekly based on progress.
5. **Diet Planner:**
   - Generate diet plans via constraint-based optimization and contextual bandits.
   - Ensure macro balance and cultural/medical compliance.
6. **Virtual AI Trainer (RAG + LLM):**
   - Provide natural language answers to user queries.
   - Offer modifications, recovery tips, and machine usage guidance.
7. **Progress Tracker:**

- Visual graphs showing progress in body metrics, calorie intake, and workout adherence.
- Gamified badges for motivation.

8. **Offline Mode:**

- On-device inference using TensorFlow Lite for real-time analysis without cloud dependency.

2.3 User Characteristics

User Type	**Description**
Beginner	Users new to fitness who need step-by-step guidance and visual form feedback.
Intermediate	Users with experience seeking adaptive plans and diet tracking.
Medical/Cultural Users	Users requiring restrictions (e.g., halal, diabetic).

2.4 Constraints

- Real-time processing limited by mobile hardware.
- Requires camera permissions for pose detection.
- Limited connectivity may affect RAG model responses.
- Dataset collection requires diversity to avoid bias.

2.5 Assumptions and Dependencies

- Users provide accurate input for body attributes.
- External APIs (USDA, WHO) remain accessible.
- TensorFlow Lite supports optimized HRNet on targeted devices.

3 Specific Requirements

3.1 Functional Requirements

ID	Requirement	Description
FR-1	User Authentication	Login and registration via email or Google account.
FR-2	User Profiling	Collect personal and fitness data for plan generation.
FR-3	Pose Detection	Capture live video, extract body keypoints, analyze form using HRNet.
FR-4	Workout Generation	ML-based personalized workout plan based on goal.
FR-5	Diet Recommendation	Constraint-based optimizer generates daily diet plan.
FR-6	Virtual Trainer	RAG + fine-tuned LLM handles all fitness-related Q&A.
FR-7	Reinforcement Learning Engine	Adapt workout/diet plans weekly based on feedback and progress.
FR-8	Progress Tracking	Visual analytics dashboard for calories, weight, workout adherence.
FR-9	Data Storage	PostgreSQL for structured data; FAISS for vector retrieval.
FR-10	Offline Functionality	On-device inference for pose and progress tracking.

3.2 Non-Functional Requirements

Category	Description
Performance	App should provide feedback within 500ms after each pose frame.
Security	Encrypt all user data; JWT authentication for API access.
Scalability	Support multiple concurrent users via asynchronous FastAPI.
Portability	Run on both Android and iOS via React Native.
Usability	Simple, intuitive UI with real-time visual feedback.
Reliability	Maintain ¿95% uptime during usage sessions.

3.3 Software Interfaces

- **Frontend ↔ Backend:** REST APIs (FastAPI)
- **Backend ↔ AI Models:** TensorFlow Lite and PyTorch

- **Backend ↔ Database:** PostgreSQL (user data), FAISS (LLM retrieval)
- **External APIs:** USDA, WHO, Nutritionix

3.4 Hardware Interfaces

- Mobile camera (for form analysis)
- Microphone (for future voice-based trainer queries)
- Local storage (for temporary data caching)

3.5 Communication Interfaces

- HTTPS for all API communication
- WebSocket for real-time updates

4 System Design Constraints

- Model quantization and pruning required for HRNet on mobile.
- RAG responses must be filtered for medical accuracy.
- Offline caching for rural or low-connectivity users.

5 Future Enhancements

- Integration of wearable sensor data (heart rate, steps).
- Voice-based interaction for hands-free training.
- Community and leaderboard features.
- Multi-language support.

6 Appendices

6.1 Tools & Technologies

Component	**Technology**
Frontend	React Native
Backend	FastAPI
Database	PostgreSQL + FAISS
AI Models	HRNet, MediaPipe, Hugging Face Transformers
ML Frameworks	TensorFlow Lite, PyTorch
Optimization	Scipy, Pyomo
Version Control	GitHub