REVA UNIVERSITY
Bengaluru, India

SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY

REPORT ON

Accurate body measurement for tailoring and alteration

BACHELOR OF TECHNOLOGY IN INFORMATION SCIENCE AND ENGINEERING

Submitted by:

Spoorthi Devraj-R23EQ111

Sinchana MU-R23EQ107

Sinchana S-R23EQ108

Spandana M-R23EQ110

Bhoomika-R24EQ802

Rishith M-R23EQ087

Shrihari-R23EQ104

Shiva-R23EQ101

Sangamesh-R23EQ097

Under the guidance of Prof .Ramaraj

Product Requirement Specifications

1. Clarity & Scope

□ **Scope of the System:**

The AccurateBody Measurements (AAD) system uses Artificial Intelligence to estimate a person's body measurements (such as height, chest, waist, and hips) from two digital images — a front and a side view — captured using a standard smartphone camera.

The system provides accurate, contactless measurements that can be used for online shopping, tailoring, and fitness tracking.

□ **In-Scope Features:**

- Uploading of front and side photos.
- Automatic detection of body landmarks.
- Use of a reference object (e.g., ruler or marked line) for scale calibration.
- Generation of accurate measurement values in cm and inches.
- Secure data handling with automatic deletion after processing.
- Integration API for e-commerce platforms or tailoring services.

□ **Out-of-Scope Features:**

- Real-time video measurement or 3D body scanning.
- Virtual try-on or clothing recommendation visualization.
- Use of photos without a reference scale.
- Manual measurement correction or editing by the user.
- AI model training using user photos (without explicit consent).

2. Functional Requirements

□ **Photo Upload:**

The user can upload front and side photos for body measurement.

- System should accept only clear, full-body images.
- Results must have at least 90–95% accuracy.

- **Scale Reference:**

  The user must provide a known reference object (like a ruler or marked line) to help the AI calculate real dimensions.
  - If missing, system asks the user to re-upload or confirm scale length.

- **Measurement Calculation:**

  The system processes the images using an AI model to estimate measurements such as height, chest, waist, and hips.
  - The process should complete in under 2 seconds.
  - Results displayed in both cm and inches.

- **Privacy & Consent:**

  User photos are processed only after consent.
  - Data and images are deleted automatically after results are generated.

- **Result Display & Report:**

  The app shows all calculated measurements clearly.
  - User can download or share a measurement report.

- Error Handling:
  - Invalid or blurry photo → show error message.
  - Missing scale reference → ask for correction.
  - Server error → show “Try again later.”
  -

3. Non-Functional Requirements (NFRs)

- **Performance:**
  - Image analysis should take less than 2 seconds.
  - The system should handle at least 1000 users per minute.
- **Scalability:**
  - Should support 1 lakh+ users during peak shopping seasons.
- **Security:**

  - All photos uploaded via secure HTTPS.
  - User data encrypted.
  - No data shared with third parties.
- **Reliability:**
  - 99.5% uptime.
  - Automatic backups and failover support.
- **Usability:**
  - Simple and mobile-friendly interface.
  - Measurement visible within 3 clicks.
- **Privacy & Compliance:**
  - Follows data privacy norms (GDPR-like).
  - Users can delete all stored data anytime.
- **Portability:**
  - Works on Android, iOS, and web browsers.

4. AI-Specific Considerations

1. **Model Accuracy:**
   The AI model should maintain at least 95% accuracy across diverse body types, lighting conditions, and clothing styles.

2. **Bias Mitigation:**
   The dataset should include diverse gender, age, and skin tone samples to minimize algorithmic bias.

3. **Continuous Learning:**
   The model should allow retraining and fine-tuning as new data is collected to improve accuracy over time.

4. **Explainability:**
 The system should provide a confidence score (e.g., 0–1) for each measurement to help identify uncertain results.

5. **Ethical AI Use:**
 User photos should never be reused for AI training without explicit consent.
 Faces should be anonymized or blurred automatically before model inference.

6. **Performance Monitoring:**

AI model performance metrics (accuracy, latency, and error rate) should be logged and reviewed periodically.

Performance scale

Parameter	**Expected Standard**	**Description**
● Accuracy	● ≥ 95%	● Measurement results should match actual dimensions within ±5%.
● Response Time	● ≤ 2 seconds	● Time taken to process and return measurements after photo upload.
● Throughput	● ≥ 1000 requests/minute	● Number of user requests that can be handled efficiently per minute.
● Uptime	● ≥ 99.5%	● System availability for users across all hours.
● Scalability	● Up to 1 lakh concurrent users	● Should support heavy usage during festivals or sales.

● Storage Efficiency	● Temporary (≤ 24 hrs)	● Uploaded photos and data deleted automatically after use.
● Error Tolerance	● ≤ 2%	● Maximum acceptable rate of failed or inaccurate results.
● Network Latency	● ≤ 200 ms	● Maximum delay for UI response to user actions.

5. Traceability & Testability

➢ **Traceability:**

- Every functional requirement should be linked to corresponding test cases and user stories.
- A Requirements Traceability Matrix (RTM) should map:
    - Requirement ID → Design Component → Test Case → Deployment Feature.

➢ **Testability:**

- All major functions (photo upload, scaling, measurement output, report generation) must be testable through automation.
- Automated testing includes:
    - Unit testing (model components)
    - Integration testing (AI + UI + backend)
    - End-to-end workflow testing
- Accuracy validation to be performed using a standardized benchmark dataset.
- Change and Dependency Management

➢ **Change Control Process:**

- Any modification in AI model, UI, or backend API should go through version control and documented approval.
- Maintain a changelog describing the nature, reason, and impact of each change.

➢ **Dependency Management:**

- Track all dependent libraries, AI frameworks (e.g., TensorFlow, PyTorch), and APIs.
- Verify compatibility during upgrades to avoid conflicts.
- Use environment isolation (like Docker or virtual environments) to manage dependencies safely.

➢ **Versioning:**

- Use semantic versioning (v1.0, v1.1, etc.) for releases.
- Ensure backward compatibility for integrated systems (e-commerce API, mobile app).

6. Change & Dependency Management

Effective change and dependency management ensures that all modifications in the AI system, backend services, or UI components are handled systematically without disrupting functionality or introducing new errors.

Dependencies:

- The system depends on several external components such as AI frameworks (TensorFlow or PyTorch), image processing libraries, cloud storage services, and APIs for integration with e-commerce platforms.

- Each dependency should be explicitly listed in a dependency register, mentioning version, usage scope, and ownership.

- Any dependency on third-party software or tools should include monitoring for updates, patches, or end-of-life announcements to prevent compatibility issues.

- Cross-team dependencies (e.g., UI team, AI model team, backend team) must be documented with clear communication protocols to ensure synchronized updates.

Assumptions & Constraints:

- All body measurements are derived from static front and side images with a reference scale visible.

- It is assumed that users will provide clear, full-body photos with minimal background clutter.

- Constraints include limited lighting variation tolerance, dependence on accurate scale reference, and mobile camera resolution quality.

- Assumptions and constraints must be reviewed during every iteration to identify potential risks and hidden blockers early in the development cycle.

Versioning & Change Control:

- A version control mechanism (e.g., Git) should be used for tracking changes to code, AI model weights, and configuration files.

- Semantic versioning (v1.0.0, v1.1.0, etc.) must be adopted for each release, ensuring traceability of improvements and backward compatibility with integrated systems.

- All modifications — whether related to the model accuracy, UI changes, or API updates — should follow a defined approval process, documented through change requests.

- A changelog must be maintained for each release, specifying the nature, purpose, and impact of the change on overall system performance.

- When major design or logic changes are made, regression testing and validation must be carried out to ensure stability.

By maintaining strict control over dependencies and implementing robust change management, the system can evolve continuously while maintaining performance, reliability, and consistency across all platforms.

7. Delivery Alignment

Delivery alignment ensures that the development and deployment of the Accurate Body Measurement System remain consistent with business goals, user needs, and Agile best practices. It guarantees that the product roadmap, backlog prioritization, and sprint activities are well-coordinated across teams.

Prioritization & Requirement Ordering:

- All PRS items should be reviewed and prioritized based on business value, user impact, and technical feasibility.

- Requirements should be categorized into MVP (Minimum Viable Product) and Future Enhancements.

    - *MVP features* include image upload, scale reference, and accurate measurement generation.

    - *Future features* include 3D visualization, personalized clothing recommendations, and integration with fitness apps.

- Product Owners and Scrum Masters must ensure that only validated, high-priority features enter the active sprint backlog.

Definition of Done (DoD):

- Each requirement should have a clearly defined DoD that includes acceptance criteria, testing completion, documentation updates, and stakeholder approval.

- The DoD must ensure that no task is considered “done” unless it meets quality benchmarks such as 95% measurement accuracy, proper data deletion, and user interface responsiveness under load.

- Automated test reports and user validation results must be attached as evidence of completion before closing any PRS item.

Agile Process Compliance:

- The development cycle must adhere to core Agile ceremonies including backlog refinement, sprint planning, daily stand-ups, sprint reviews, and retrospectives.

- During backlog refinement, PRS items should be broken down into smaller, testable user stories that can be completed within a sprint.

- Sprint planning should align delivery milestones with measurable KPIs such as accuracy improvement or latency reduction.

- Continuous integration and delivery (CI/CD) pipelines must be maintained to enable frequent, reliable deployments without manual intervention.

- Regular stakeholder demos should be conducted to validate progress and align with evolving user expectations.