**CHAPTER 5: PROJECT MANAGEMENT**

AttendEase is a face recognition-based student attendance system developed over a three-month period. The system automates attendance tracking in academic settings using facial recognition technology, eliminating traditional manual roll calls and paper-based signing methods that are time-consuming and prone to errors. The project employed agile methodology for development, consisting of design, frontend and backend phases.

### 5.1 Budget Planning

**Development Costs:**

* **Software Tools:** Ksh 40,000 for premium versions of development tools, plus utilization of open-source frameworks and libraries (Django, TensorFlow, Face\_Recognition API).
* **Hardware Requirements:** Ksh 60,000 for testing devices, including webcam-equipped laptops for facial recognition trials.
* **Human Resources:** Ksh 150,000 for a small team of student developers with allocated roles across design, frontend, and backend.
* **Training Data:** Ksh 75,000 for collection and processing of facial data samples for system training and testing.
* **Cloud Services:** Ksh 50,000 for three months of cloud hosting and database services during development.

**Budget Allocation:**

* **Development Environment:** Ksh 120,000 (15% of budget) for development tools and testing environment setup.
* **Human Resources:** Ksh 600,000 (70% of the total budget) allocated to team members based on role complexity and time commitment.
* **Contingency Reserve:** Ksh 120,000 (15% of total budget) reserved for unforeseen technical challenges and additional requirements.
* **Total Project Budget:** Ksh 860,000.

**Additional Expenses:**

* **Software Licenses:** Ksh 80,000 for specialized IDE and development tools.
* **Testing Hardware:** Ksh 90,000 for additional webcam units with different specifications.
* **Team Training:** Ksh 60,000 for online courses on facial recognition implementation.
* **Documentation Tools:** Ksh 30,000 for project management and documentation software.

### 5.2 Time Management and Milestones

**Sprint Planning:**

* **Sprint Duration:** Two-week sprints following agile methodology.
* **Daily Stand-ups:** 15-minute meetings to discuss progress, blockers, and next steps.

**Key Milestones:**

1. **Weeks 1-2: Planning and Design**
   * Requirement gathering and analysis.
   * Wireframing and UI/UX design using tools like Figma.
   * Database design and system architecture planning.
2. **Weeks 3-6: Frontend Development**
   * Development of the user interface using HTML, CSS (Tailwind), and JavaScript.
   * Integration of Face\_API.js for real-time face detection.
   * Testing the frontend for responsiveness and usability.
3. **Weeks 7-10: Backend Development**
   * Setting up Django for backend logic.
   * Integrating TensorFlow and Face\_Recognition for facial recognition.
   * Database integration for storing attendance records.
4. **Weeks 11-12: Testing and Deployment**
   * System testing and bug fixing.
   * Deployment on a local server or cloud platform.
   * Final documentation and handover.

### 5.3 Resource Allocation and Usage

**Human Resources:**

* **Project Manager (James Ngandu):** Overseeing the timeline, coordinating team activities, and managing stakeholder communications.
* **UI/UX Designer (Jacinta Atieno):** Creating intuitive user interfaces and user experience flows.
* **Frontend Developer (James Ngandu):** Implementing responsive web interfaces using HTML, Tailwind CSS, and JavaScript.
* **Backend Developer (Henry Ouma):** Building server-side logic, database structure, and API endpoints with Django.
* **ML Engineer (David Wambua):** Implementing and fine-tuning facial recognition algorithms.

**Technical Resources:**

* **Development Environment:** Local development setups with necessary software installations.
* **Version Control:** Git repository for collaborative development and code versioning.
* **Testing Equipment:** Laptops with webcams for facial recognition testing across different lighting conditions.
* **Database Server:** Local database instance for development and testing.

**Resource Optimization Strategies:**

* Cross-training team members to handle multiple responsibilities.
* Using containerization for consistent development environments.
* Implementing reusable code components to improve development efficiency.

### 5.4 Risk Management

**Technical Risks:**

1. **Facial Recognition Accuracy:**
   * **Risk:** Poor recognition in varying lighting conditions or with facial accessories.
   * **Mitigation:** Implementation of preprocessing techniques and extensive testing with diverse sample data.
2. **System Performance:**
   * **Risk:** Slow processing when handling multiple simultaneous recognition requests.
   * **Mitigation:** Optimization of algorithms and implementation of queue-based processing.
3. **Data Security:**
   * **Risk:** Privacy concerns regarding storage of facial data.
   * **Mitigation:** Implementing data encryption, secure storage protocols, and user consent mechanisms.

**Project Management Risks:**

1. **Scope Creep:**
   * **Risk:** Expanding requirements beyond the initial project boundaries.
   * **Mitigation:** Clear documentation of project scope and change request process.
2. **Schedule Delays:**
   * **Risk:** Missed deadlines due to technical challenges.
   * **Mitigation:** Buffer time incorporated into sprint planning and regular progress tracking.
3. **Team Coordination:**
   * **Risk:** Communication gaps between frontend, backend, and ML components.
   * **Mitigation:** Regular integration meetings and collaborative documentation.

**Operational Risks:**

1. **User Adoption:**
   * **Risk:** Resistance from faculty or students to the new system.
   * **Mitigation:** Early stakeholder involvement, intuitive UI design, and comprehensive training.
2. **System Integration:**
   * **Risk:** Challenges integrating with existing academic management systems.
   * **Mitigation:** Modular design approach and well-documented APIs for future integration.