PROPOSAL FOR QUEUE MANAGEMENT SYSTEM IMPLEMENTATION: WGGA EYE CENTER

ACE-TECH

Bole in front of Mega
building
Addis Ababa
Ethiopia

Ace-Tech Software Solutions

---

1 DOCUMENT CONTROL

1.1 Change Record

Date	Author	Version	Notes
October 24, 2024	Yafet Surafel	1.0	Initial Draft

1.2 Reviewers

Name	Position
Elias	IT staff

---

2 **CONTENTS**

3 **EXECUTIVE SUMMARY**

This proposal outlines the implementation plan for a comprehensive Queue Management System at WGGA Eye Center. The system aims to enhance patient flow, reduce wait times, and improve overall service efficiency by utilizing advanced hardware and software components. Key functionalities include real-time queue display, ticketing kiosks, centralized server management, and optional enhancements like audio announcements and professional media content.

---

4 **SYSTEM OVERVIEW**

The proposed system leverages state-of-the-art technology to streamline patient management processes within the WGGA Eye Center. It encompasses display screens powered by Raspberry Pi devices for real-time queue information, touch screen kiosks for ticket generation, and a central server running a custom Flask application integrated with Gize for backend management. The system is designed with scalability, security, and user-friendliness in mind, ensuring a seamless experience for both patients and staff.

---

5 **KEY FUNCTIONAL AREAS**

5.1 **Display Screens & Media Players**

- **Hardware Components:**
  - 4 x 43" Full HD Display Screens
  - 4 x Raspberry Pi 4 Model B (4GB RAM)
- **Purpose:**
  - Display real-time queue information and announcements.
- **Features:**
  - High-definition visual output for clarity.
  - Raspberry Pi devices run the front-end Flask application.
  - Communication with the central server for live updates.

5.2 **Input Devices for Ticketing**

- **Hardware Components:**
  - 2 x Touch Screen Kiosks with Integrated Ticket Printers
  - 2 x Barcode Scanners (Optional)
- **Purpose:**
  - Allow patients to generate queue tickets for different services (Doctors, Refraction).
- **Features:**
  - User-friendly interface for ticket generation.

    - Optional barcode scanners for efficient check-in and tracking.
    - Integration with the central system for real-time data synchronization.

5.3 **Cables, Splitters & Network**

- **Hardware Components:**
    - 4 x HDMI 2.0 Cables
    - 1 x HDMI Splitter (1x4)
    - 4 x Ethernet Cables (Cat6)
- **Purpose:**
    - Ensure reliable connectivity between devices.
- **Features:**
    - HDMI cables connect Raspberry Pis to display screens.
    - HDMI splitter for mirroring content across multiple screens.
    - Ethernet cables for network connectivity and data transmission.

5.4 **Server & Software**

- **Hardware Components:**
    - 1 x Central Queue Server (Intel i5 or higher, 16GB RAM, 512GB SSD)
- **Software Components:**
    - Custom Flask Application Integrated with Gize
- **Purpose:**
    - Manage queues, handle backend processes, and facilitate communication between devices.
- **Features:**
    - Robust server specifications for optimal performance.
    - Scalable software architecture for future expansions.
    - Secure data handling and storage.

5.5 **Optional Add-Ons**

- **Hardware Components:**
    - 1 x Blackmagic ATEM Mini
    - 4 x External Speakers
- **Purpose:**
    - Enhance media capabilities and provide audio announcements.
- **Features:**
    - Professional-grade video switching for dynamic content.
    - High-quality speakers for clear audio notifications.

5.6 **Accessories & Power**

- **Hardware Components:**
    - 4 x Wall Mounts for Display Screens
    - 4 x UPS Units

    - Cable Management Kit
    - Label Maker (Optional)
- **Purpose:**
    - Ensure a neat, organized, and uninterrupted system operation.
- **Features:**
    - Secure mounting solutions for displays.
    - Uninterruptible power supplies for critical components.
    - Efficient cable organization and device labeling for maintenance ease.

---

6 **SECURITY AND IMPLEMENTATION**

6.1 **User Permissions**

- **Role-Based Access Control (RBAC):**
    - Define roles such as Admin, Receptionist, Technician.
    - Assign permissions based on roles to restrict unauthorized access.
- **Implementation:**
    - Admins have full control over the system settings and data.
    - Receptionists can manage queues and assist patients.
    - Technicians handle hardware maintenance and monitoring.

6.2 **Secure Login and Authentication**

- **Authentication Mechanisms:**
    - Secure login for administrative interfaces.
    - Strong password policies and periodic resets.
- **Multi-Factor Authentication (MFA):**
    - Implement MFA for admin accounts to enhance security.
- **Session Management:**
    - Automatic logout after periods of inactivity.

6.3 **Data Encryption**

- **Encryption in Transit:**
    - Use SSL/TLS protocols for data transmission between devices and server.
- **Encryption at Rest:**
    - Encrypt sensitive data stored on the server using AES-256 encryption.
- **Data Protection:**
    - Regular backups and secure storage solutions.

6.4 **Network Security**

- **Firewall Implementation:**
    - Configure firewalls to protect against unauthorized access.
- **Network Segmentation:**
    - Isolate the queue management system network from the general hospital network.
- **Regular Security Audits:**
    - Conduct periodic vulnerability assessments and penetration testing.

6.5 **Hardware Security**

- **Physical Security Measures:**
    - Secure mounting of devices to prevent tampering.
    - Lockable enclosures for kiosks and server hardware.
- **UPS Protection:**
    - Uninterruptible power supplies to prevent data loss during power outages.
- **Device Monitoring:**
    - Regular hardware checks and maintenance schedules.

---

7 **TECHNICAL PROPOSAL**

7.1 **System Architecture**

7.1.1 **Frontend**

- **Technology:**
    - Flask framework serving dynamic web pages.
- **Devices:**
    - Raspberry Pi units connected to display screens.
- **Features:**
    - Real-time queue updates.
    - User-friendly interface for patients.

7.1.2 **Backend**

- **Technology:**
    - Flask application running on the central server.
- **Functionality:**
    - Manages queue logic, user data, and system configurations.
- **Integration:**
    - Communicates with frontend devices and databases.

7.1.3 **Database**

- **Technology:**
  - PostgreSQL or MySQL for robust data handling.
- **Data Management:**
  - Stores queue data, user information, and system logs.
- **Security:**
  - Implements data encryption and access controls.

7.2 **Integration Points**

7.2.1 **Hardware Integration**

- **Display Systems:**
  - Raspberry Pis connected via HDMI to display screens.
- **Input Devices:**
  - Kiosks interfacing with ticket printers and optional barcode scanners.
- **Network Connectivity:**
  - Ethernet connections for reliable communication.

7.2.2 **Software Integration**

- **Flask Application:**
  - Integrated with Gize for enhanced functionality.
- **API Development:**
  - Custom APIs for communication between server and devices.
- **Third-Party Services:**
  - Optional integration with paging systems or mobile applications.

7.3 **Deployment Plan**

- **Phase 1: Preparation**
  - Site survey and planning.
  - Hardware inventory and organization.
- **Phase 2: Installation**
  - Mounting of display screens and kiosks.
  - Hardware setup and configuration.
- **Phase 3: Network Setup**
  - Network configuration and security setup.
- **Phase 4: Software Deployment**
  - Installation of server and client applications.
  - System testing and validation.

---

8 **IMPLEMENTATION STRATEGY**

8.1 **Project Phases**

Phase 1: Requirements Analysis & Planning

- **Duration:** 3 days
- **Activities:**
  - Confirm project scope and objectives.
  - Develop a detailed implementation roadmap.
- **Deliverables:**
  - Project plan with timelines and milestones.

Phase 2: System Design

- **Duration:** 2 days
- **Activities:**
  - Design system architecture and hardware layout.
  - Define network topology and security measures.
- **Deliverables:**
  - Detailed design documents and schematics.

Phase 3: Hardware Installation

- **Duration:** 1 week
- **Activities:**
  - Install display screens, kiosks, and server.
  - Set up Raspberry Pi devices and peripherals.
- **Deliverables:**
  - Fully installed hardware components.

Phase 4: Software Configuration

- **Duration:** 1 week
- **Activities:**
  - Deploy server software and configure applications.
  - Install and configure client-side software on Raspberry Pis.
- **Deliverables:**
  - Operational software environment.

Phase 5: Testing & Quality Assurance

- **Duration:** 3 days
- **Activities:**
  - Conduct comprehensive system testing.
  - Perform user acceptance testing with staff.

- **Deliverables:**
  - Testing reports and resolved issues.

Phase 6: Training and Handover

- **Duration:** 2 days
- **Activities:**
  - Train staff on system usage and administration.
  - Provide documentation and support materials.
- **Deliverables:**
  - Trained personnel and handover documents.

Phase 7: Go-Live and Support

- **Duration:** Ongoing
- **Activities:**
  - Monitor system performance.
  - Provide technical support and maintenance.
- **Deliverables:**
  - Fully operational system with support mechanisms.

8.2 **Deployment Methodology**

- **Approach:**
  - Utilize a phased rollout to minimize disruptions.
- **Testing Environment:**
  - Set up a testing environment mirroring the production setup.
- **Continuous Integration:**
  - Implement CI/CD pipelines for software updates.

8.3 **Change Management**

- **Process:**
  - Document and approve all changes through a formal process.
- **Communication:**
  - Keep stakeholders informed of changes and impacts.
- **Version Control:**
  - Maintain version control for software and configurations.

8.4 **Training and Handover**

- **Training Sessions:**
  - Conduct workshops for different user groups.
- **Documentation:**
  - Provide user manuals and technical guides.
- **Support Transition:**

    - Outline support procedures post-implementation.

---

9 **STAKEHOLDER ROLES AND RESPONSIBILITIES**

9.1 **Project Sponsor**

- **Role:**
    - Oversee project funding and alignment with organizational goals.
- **Responsibilities:**
    - Approve budgets and resources.
    - Provide strategic direction.

9.2 **Project Manager**

- **Role:**
    - Coordinate all aspects of the project.
- **Responsibilities:**
    - Manage timelines, resources, and communication.
    - Ensure project milestones are met.

9.3 **Implementation Team**

- **Roles:**
    - Technicians, Software Developers, Network Engineers.
- **Responsibilities:**
    - Execute installation, configuration, and testing.
    - Develop and deploy software components.
- **Team Breakdown:**
    - **Lead Technician:** Oversees hardware setup.
    - **Software Developer:** Handles application development.
    - **Network Engineer:** Manages network configuration.
    - **QA Specialist:** Conducts testing and quality assurance.

9.4 **Communication and Reporting Structure**

- **Regular Meetings:**
    - Weekly progress meetings with stakeholders.
- **Reporting:**
    - Status updates and reports on project milestones.
- **Issue Escalation:**
    - Defined channels for addressing project challenges.

Further Documents Required as Attachments

- **Floor Plan with Device Locations - Elias**
- **Network Topology Diagram - Elias**
- **User Manuals and Training Materials - Ace-Tech**
- **Maintenance Schedule - Ace-Tech**