**Technical Specification Document (TSD)**

**📌 Project Overview**

* **Project Name:** Menu.X
* **Version:** 1.0
* **Date:** July 3, 2025
* **Prepared by:** [Your Name]
* **Status:** Final Draft

**1. Introduction**

**1.1 Purpose**

This document provides a comprehensive technical overview of the Menu.X platform, detailing its architecture, components, and interactions. It serves as a blueprint for developers, testers, and stakeholders to understand the system's design and implementation.

**1.2 Scope**

Menu.X is an AI-powered platform designed to assist restaurants in creating dynamic digital menus. The system leverages AI to analyze customer preferences and suggest optimal menu items, enhancing the dining experience and operational efficiency.

**2. System Architecture**

**2.1 Overview**

The Menu.X platform follows a modular architecture comprising the following layers:

* **Frontend:** User interfaces for restaurant owners and diners.
* **Backend:** Server-side logic, APIs, and database management.
* **AI Engine:** Machine learning models for menu optimization and forecasting.
* **Third-Party Integrations:** Payment gateways, analytics tools, and other external services.

**2.2 Component Diagram**

[Insert a component diagram illustrating the interactions between the frontend, backend, AI engine, and third-party integrations.][helpjuice.com+4meegle.com+4slite.com+4](https://www.meegle.com/en_us/advanced-templates/full_stack_development/technical_specification_template?utm_source=chatgpt.com)

**3. Technology Stack**

**3.1 Frontend**

* **Framework:** React Native (for cross-platform mobile development)
* **State Management:** Redux
* **UI Components:** Material-UI[smartsheet.com](https://www.smartsheet.com/free-technical-specification-templates?utm_source=chatgpt.com)

**3.2 Backend**

* **Framework:** Node.js with Express
* **Database:** Firebase Firestore
* **Authentication:** Firebase Authentication

**3.3 AI Engine**

* **Model:** Google Gemini AI
* **Integration:** Firebase ML Kit

**3.4 Third-Party Integrations**

* **Payment Gateway:** Stripe
* **Analytics:** Google Analytics
* **Cloud Hosting:** Firebase Hosting[archbee.com+1helpjuice.com+1](https://www.archbee.com/blog/technical-specification?utm_source=chatgpt.com)

**4. Functional Requirements**

**4.1 User Roles**

* **Restaurant Owner:** Manages menu items, views analytics, and configures settings.
* **Diner:** Browses digital menus, places orders, and provides feedback.

**4.2 Key Features**

* **AI-Powered Menu Generation:** Automatically suggests menu items based on customer preferences and trends.
* **Dynamic Menu Updates:** Real-time updates to the menu based on inventory and customer feedback.
* **Order Management:** Integration with POS systems to manage orders efficiently.
* **Analytics Dashboard:** Insights into customer preferences, sales trends, and inventory status.

**5. Non-Functional Requirements**

* **Performance:** The system should handle up to 10,000 concurrent users without degradation in performance.
* **Scalability:** The architecture should support horizontal scaling to accommodate growth.
* **Security:** Data encryption at rest and in transit; compliance with GDPR and other relevant regulations.
* **Availability:** 99.9% uptime with automated failover mechanisms.

**6. Data Flow Diagrams**

**6.1 Level 1 DFD**

[Insert a Level 1 Data Flow Diagram illustrating the high-level data interactions between users, the frontend, backend, and AI engine.]

**6.2 Level 2 DFD**

[Insert a Level 2 Data Flow Diagram detailing the interactions within the backend, including database operations and API calls.]

**7. API Specifications**

**7.1 Authentication API**

* **Endpoint:** POST /api/auth/login
* **Request Body:** { "email": "string", "password": "string" }
* **Response:** { "token": "string" }
* **Description:** Authenticates a user and returns a JWT token.[archbee.com+8blog.monday.com+8smartsheet.com+8](https://blog.monday.com/blog/rnd/technical-specification/?utm_source=chatgpt.com)

**7.2 Menu Management API**

* **Endpoint:** GET /api/menu
* **Response:** [{ "id": "string", "name": "string", "price": "number", "description": "string" }]
* **Description:** Retrieves the list of menu items.

**7.3 Order API**

* **Endpoint:** POST /api/order
* **Request Body:** { "items": [{"id": "string", "quantity": "number"}], "total": "number" }
* **Response:** { "status": "string", "orderId": "string" }
* **Description:** Places a new order.

**8. Security Considerations**

* **Authentication:** JWT tokens for secure user authentication.
* **Authorization:** Role-based access control (RBAC) to restrict access to resources.
* **Data Protection:** Use of HTTPS for all communications; data encryption using AES-256.
* **Compliance:** Adherence to GDPR, CCPA, and other relevant data protection regulations.

**9. Testing Strategy**

**9.1 Unit Testing**

* **Tools:** Jest for JavaScript testing.
* **Coverage:** 80% code coverage for all modules.

**9.2 Integration Testing**

* **Tools:** Mocha and Chai for API testing.
* **Focus:** Ensure correct interaction between frontend, backend, and AI engine.

**9.3 User Acceptance Testing (UAT)**

* **Participants:** Selected restaurant owners and diners.
* **Objective:** Validate the system against business requirements and user expectations.

**10. Deployment Plan**

* **Staging Environment:** Firebase Hosting for pre-production testing.
* **Production Environment:** Firebase Hosting with CDN for global delivery.
* **CI/CD Pipeline:** GitHub Actions for automated testing and deployment.
* **Monitoring:** Firebase Analytics and Google Analytics for performance tracking.

**11. Maintenance and Support**

* **Monitoring:** 24/7 system monitoring using Firebase Crashlytics and Google Cloud Monitoring.
* **Incident Response:** Defined SLA for issue resolution; dedicated support team.
* **Updates:** Regular updates for security patches and feature enhancements.

**12. Glossary**

* **JWT:** JSON Web Token, a compact and self-contained way to represent information between parties.
* **RBAC:** Role-Based Access Control, a method of restricting system access based on users' roles.
* **AES-256:** Advanced Encryption Standard with a 256-bit key size, used for securing data.[meegle.com](https://www.meegle.com/en_us/advanced-templates/full_stack_development/technical_specification_template?utm_source=chatgpt.com)

**13. References**

* Stack Overflow's guide on writing technical specs: [A practical guide to writing technical specs](https://stackoverflow.blog/2020/04/06/a-practical-guide-to-writing-technical-specs)
* Slite's software design documentation template: [Software Design Documentation Template](https://slite.com/templates/software-design-documentation)
* Meegle's technical specification template: [Technical Specification Template](https://www.meegle.com/en_us/advanced-templates/full_stack_development/technical_specification_template)[stackoverflow.blog](https://stackoverflow.blog/2020/04/06/a-practical-guide-to-writing-technical-specs?utm_source=chatgpt.com)[slite.com+1slite.com+1](https://slite.com/templates/software-design-documentation?utm_source=chatgpt.com)[meegle.com](https://www.meegle.com/en_us/advanced-templates/full_stack_development/technical_specification_template?utm_source=chatgpt.com)

**Menu.X – Project Bible (v6.1)**

**Status:** Final  
**Date:** July 3, 2025

**🌟 PART 1: STRATEGIC OVERVIEW**

**1.1 Vision & Identity**

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Name** | Menu.X |
| **Tagline** | Simple Menu, Faster Service. |
| **Vision** | Become the leading digital ordering & dining-experience platform in Bangladesh, celebrated for elegance, simplicity, and intelligence. |

**1.2 Go-to-Market Strategy**

* **Target Audience:** Urban cafés/casual diners in Bangladesh
* **Model:** Freemium – Free Plan + Pro Plan launch
* **Scope:** "All‑in‑One" release (full feature set, excluding auto-payments)
* **User Acquisition:** 7–30‑day Pro trial offered to all new signups

**1.3 Initial Business Operations**

* **Pro Price:** ৳1,500/month
* **Payment:** Manual via MFS (bKash/Nagad)
* **Diner Model:** "Order Now, Request Bill"—physical payment at table

**1.4 Technical Overview**

* **Architecture:** SPA + Firebase BaaS
* **Stack:** React/Vue, Firestore, Auth, Functions, Hosting
* **Launch Domain:** menux-app.web.app

**👤 PART 2: THE APPLICATION**

**2.1 Personas**

* **Mr. Kabir:** Restaurant owner using management & AI tools
* **Ayesha:** Diner using smart, convenient ordering features

**2.2 Free Plan 🔒**

1. Secure Email/Google login
2. Dashboard: menu views & item count + QR code link
3. Menu management (CRUD + drag‑reorder) up to 25 items
4. Mobile menu view with “Powered by Menu.X” footer

**2.3 Pro Plan 🚀**

**Core Enhancements**

* Unlimited menu items, remove branding footer
* Pro trial banner shows remaining days

**Smart Ordering System**

* Diners: order & request bill
* Owners: live update dashboard + alerts

**AI Menu Architect**

* OCR-based menu upload
* Smart Description Writer

**AI Business Forecaster**

* Predictive insights on sales & inventory

**AI Smart Sommelier**

* Chatbot for personalized diner recommendations

**Advanced Analytics Dashboard**

* Revenue & top items charts
* Peak hours visualization

**👤 PART 3: SUPER ADMIN DASHBOARD**

**3.1 Purpose & Access**

Support team interface for user management, monetization, and analytics

**3.2 Tech Strategy**

Built with Retool/Forest Admin + Firebase backend

**3.3 Features**

**3.3.1 Business Dashboard**

* KPIs: Total users, plan breakdown, MRR
* Sign‑up growth graph

**3.3.2 User Management**  
*Table Schema:*

* Restaurant Name | Owner Email | Plan | Status | End Date  
  *Actions per user:*
* View Details
* Login as User
* Edit Plan
* Start Pro Trial

**3.3.3 Subscription Workflow**

* **Paid:** Confirm MFS payment → Edit plan to Pro, Active, +1 mo
* **Trial:** Click Start Trial → Set type to Pro, Trialing, +X days

**🔐 PART 4: CORE TECHNICAL SPECS**

**4.1 Firestore Data Model**

* **restaurants**: ownerUid, name, planType, status, subscriptionEndDate, createdAt
* **menuItems** (sub‑collection): itemName, price, description, orderIndex
* **orders** (sub‑collection): tableId, items[], totalAmount, status, createdAt

**4.2 Security Model**

* Firebase Auth for all logins
* Firestore rules to allow only owners to read/write their data

**🛠 PART 5: POST-LAUNCH ROADMAP**

**5.1 Future Enhancements**

* Full **Automated Payment Gateway** (e.g. SSLCOMMERZ)
* **Subscription Lifecycle Management** via scheduled Cloud Functions
* Incremental improvements per roadmap