

Marketplace Technical Foundation For Software House

1. System Architecture Document

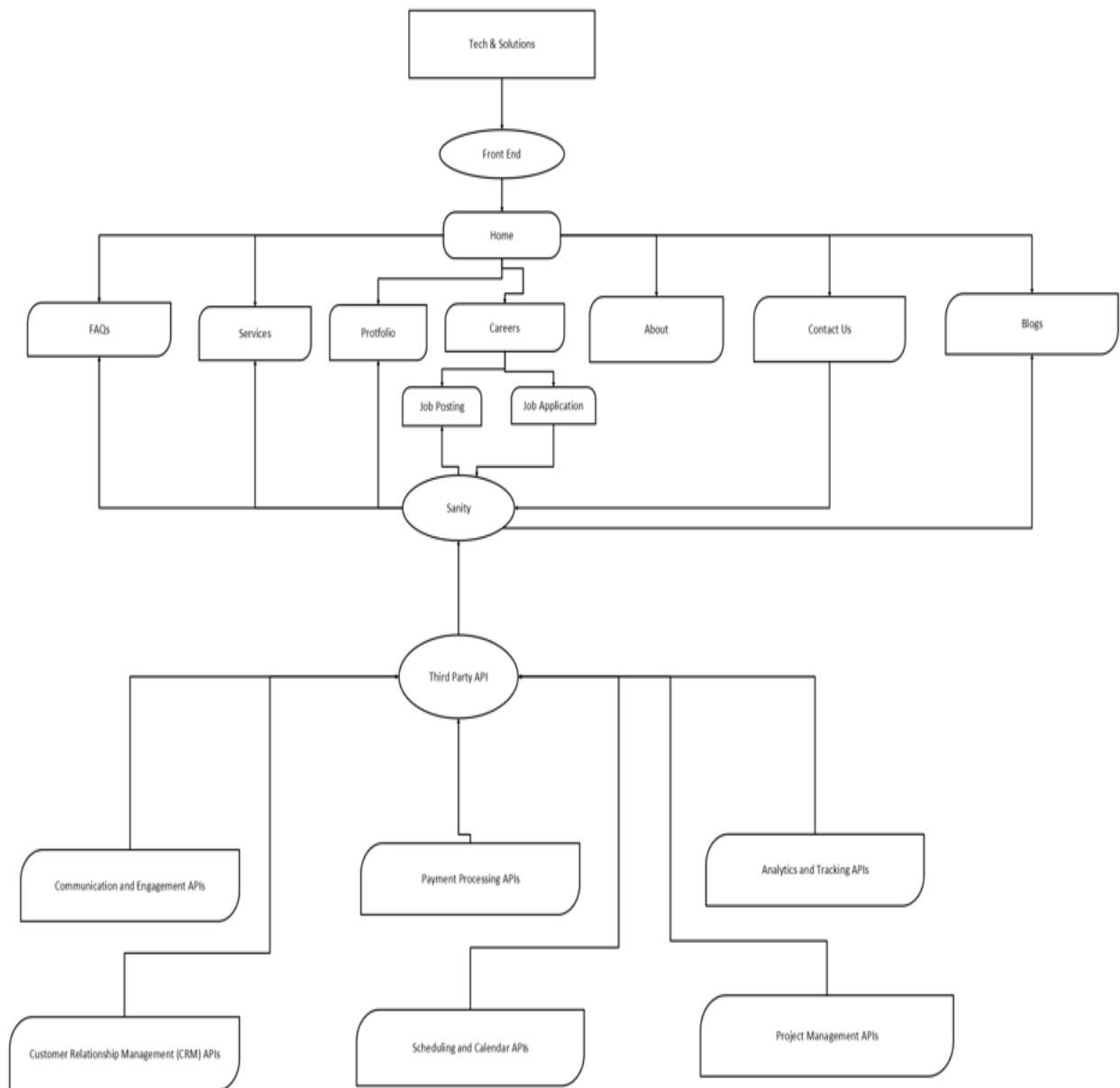
Overview

This document outlines the system architecture for the software house's website, detailing the interaction between the frontend, Sanity CMS, and third-party APIs.

Components

- **Frontend:**
 - Developed using modern web technologies (Next.js with Tailwind CSS).
 - Responsible for rendering the user interface and handling client-side interactions.
- **Sanity CMS:**
 - Acts as the backend content management system for managing dynamic content like blog posts, portfolios, and product information.
- **Third-Party APIs:**
 - Integrates external services for specific functionalities like payment processing, live chat, and analytics.

System Architecture Diagram



Roles of Components

1. Frontend:

- Handles user input and displays data fetched from Sanity CMS or third-party APIs.
- Provides a seamless and responsive user experience.

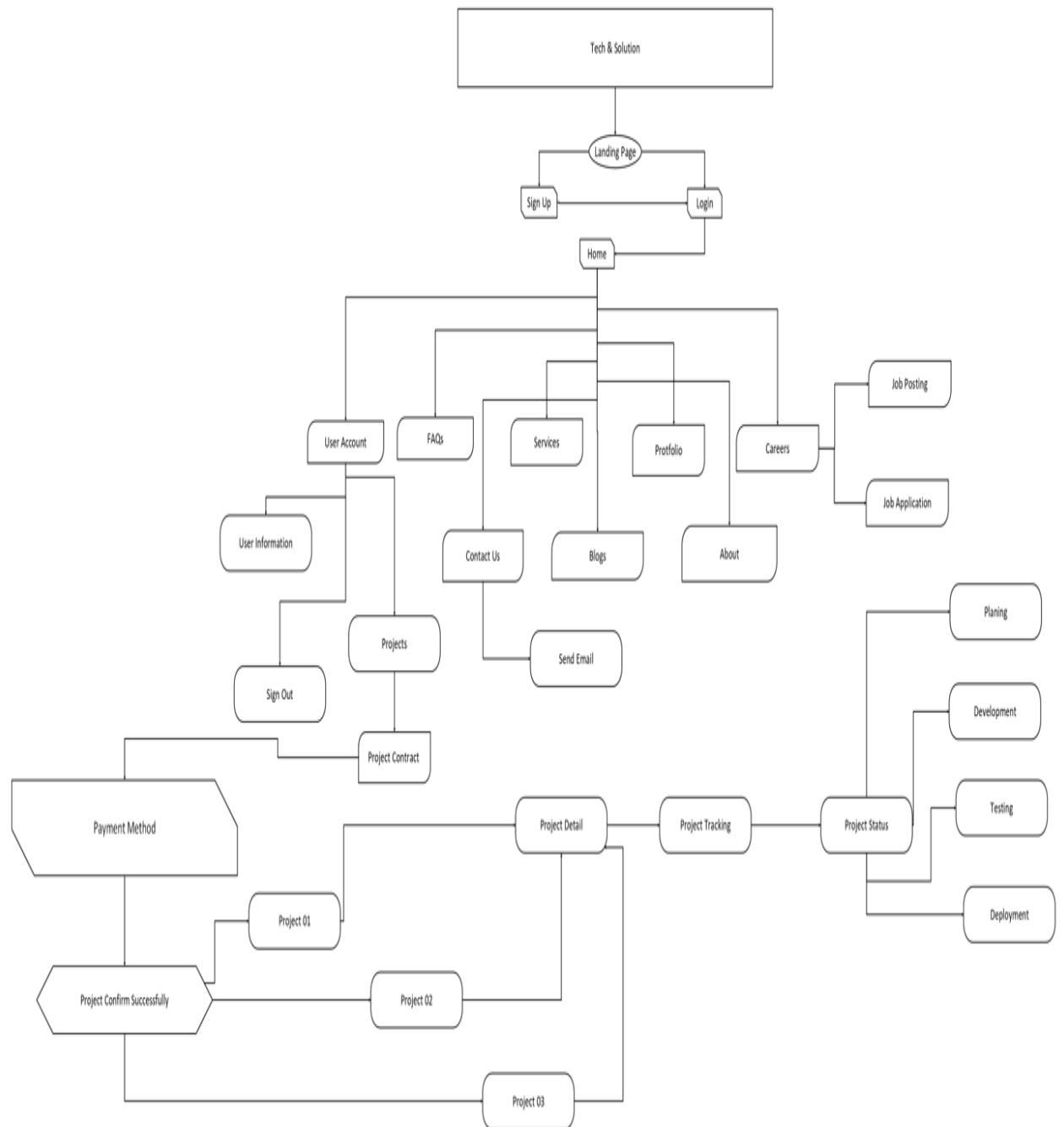
2. Sanity CMS:

- Stores and manages structured content for the website.
- Serves as the single source of truth for all dynamic data.

3. Third-Party APIs:

- **Payment APIs** (e.g., Stripe): Process client payments.
- **Communication APIs** (e.g., Twilio): Enable real-time communication.
- **Analytics APIs** (e.g., Google Analytics): Monitor website performance and user behavior.

3. Workflow Diagram



5. Technical Roadmap

Milestones

1. Frontend Development

- Design and implement the website using Next.js and Tailwind CSS.
- Integrate dynamic content fetching from Sanity CMS.

2. Sanity CMS Integration

- Set up schemas for users, projects, services, portfolio, contact, blogs, faqs and careers.
- Configure Sanity dashboard for content management.

3. Third-Party API Integration

- Integrate payment processing APIs (e.g., Stripe).
- Communication and Engagement APIs
- Customer Relationship Management (CRM) APIs
- Scheduling and Calendar APIs
- Project Management APIs
- Analytics and Tracking APIs
- Connect shipment tracking API.

4. Testing and QA

- Perform unit testing for API endpoints.
- Conduct end-to-end testing for workflows.

5. Deployment

- Deploy the website on Vercel.
- Monitor performance and ensure scalability.

Deliverables

- Fully functional and responsive website.
- Integrated CMS and third-party APIs.
- Comprehensive documentation and testing reports.