Innhold

[**Tech Stacks** 1](#_Toc201449418)

[**Frontend (Mobile App - Staff + Manager View)** 1](#_Toc201449419)

[**Backend (Firebase)** 2](#_Toc201449420)

[**Admin Panel (Web)** 2](#_Toc201449421)

[**Deployment** 2](#_Toc201449422)

[**Core Features** 2](#_Toc201449423)

[**Manager (Boss Mode)** 2](#_Toc201449424)

[**Staff (Worker Mode)** 3](#_Toc201449425)

[**Access Control & User Rights** 4](#_Toc201449426)

[**Authentication** 4](#_Toc201449427)

[**Development Roadmap** 4](#_Toc201449428)

**SmashPlan – Project Documentation**

**📅** **Overview**

**SmashPlan** is a modern, cross-platform scheduling and task management application built specifically for the **Smashed Restaurant Team**. It streamlines shift planning, task delegation, and internal communication between staff and management.

The app has two primary user roles:

* **Manager (Boss Mode)**
* **Employee (Worker Mode)**

The system consists of:

* **Flutter mobile app** (used by both Managers and Staff)
* **React.js Admin Dashboard** (for advanced management)
* **Firebase backend** for authentication, data storage, and push notifications

**Tech Stacks**

### **Frontend (Mobile App - Staff + Manager View)**

* **Framework**: Flutter
* **Language**: Dart
* **Platforms**: Android & iOS
* **State Management**: Riverpod
* **Routing**: go\_router
* **UI Components**: Flutter Material + Custom Widgets

**Backend (Firebase)**

* **Auth**: Firebase Authentication (Email, Magic Link)
* **Database**: Cloud Firestore (real-time, scalable NoSQL)
* **Notifications**: Firebase Cloud Messaging (Push)
* **Storage**: Firebase Storage (for media or attachments)

**Admin Panel (Web)**

* **Frontend**: React.js
* **Styling**: Tailwind CSS
* **Authentication**: Firebase Auth
* **State Management**: React Query / Zustand
* **UI Frameworks**: ShadCN UI / DaisyUI / Material UI (optional)

**Deployment**

* **Flutter App**: Google Play Store / Apple App Store
* **Admin Panel**: Vercel / Netlify
* **Backend**: Firebase Hosting + Firestore

## **Core Features**

**Manager (Boss Mode)**

1. **Create & Manage Work Schedules**
   * Plan daily, weekly, or monthly shifts
   * Use drag-and-drop interface (mobile + dashboard)
2. **Assign Roles to Staff**
   * Define roles like Kitchen, Cashier, Server, etc.
   * Assign specific people to roles per shift
3. **Real-Time Notifications**
   * Alert staff of schedule updates or changes
   * Send reminders and urgent alerts
4. **Track Staff Availability**
   * View time-off requests
   * Prevent scheduling conflicts
5. **Approve or Deny Shift Change Requests**
   * Get notified of staff swap requests
   * Accept/decline with optional comments
6. **Admin Dashboard (Web Only)**
   * Overview of schedules and analytics
   * Add/remove staff accounts
   * Export reports or shift history
7. **Task Assignment & Tracking**
   * Create task checklists for each shift
   * Assign tasks to specific employees

**Staff (Worker Mode)**

1. **View Personal Schedules**
   * See upcoming shifts, roles, and tasks
   * Mobile-friendly UI with calendar integration
2. **Shift Swap Requests**
   * Send swap requests to teammates
   * Request time off or suggest changes
3. **Receive Notifications**
   * Get alerts for shift assignments and updates
   * Push reminders for upcoming tasks or events
4. **Task Checklists**
   * View tasks for each shift
   * Mark tasks as completed
5. **Profile Management**
   * Update contact info
   * View shift history and performance (if implemented)

**Access Control & User Rights**

**Authentication**

* All users authenticate using Firebase Auth (email/password or magic link)

**Roles & Permissions**

| **Role** | **Permissions** |
| --- | --- |
| **Admin/Manager** | Full access: schedule creation, role assignment, notification sending, shift editing, dashboard access |
| **Staff** | Can view personal schedule, submit shift swaps, get notifications, complete tasks |

## **Development Roadmap**

**Phase 1: Core Setup**

* Firebase project configuration
* Set up Flutter project (Riverpod + go\_router)
* Create Firestore database structure

**Phase 2: Staff App (Flutter)**

* Authentication flow (login/register)
* Schedule view for staff
* Shift swap form
* Task checklist UI

**Phase 3: Manager App (Flutter)**

* Shift creation UI
* Role assignment
* Notifications
* Availability overview

**Phase 4: Admin Dashboard (React)**

* User management
* Schedule editor
* Report generation (optional)

**Phase 5: Polish & Deploy**

* Push notifications
* Testing on Android/iOS
* Publish app
* Launch dashboard