

BRD, FRD AND PRD

Subscription Management Platform

1. Business Requirement Document (BRD)

1.1 Business Objective

The objective of this project is to build a subscription management platform that helps users track, manage, and control their recurring subscriptions. The platform aims to reduce unwanted auto-renewals, improve transparency of monthly spending, and enhance user trust and engagement.

1.2 Problem Statement

Users often forget subscription renewal dates, resulting in unexpected charges and financial loss. Existing solutions lack proactive reminders, spending insights, and easy control over subscription tracking.

1.3 Goals & Success Metrics (KPIs)

- Reduce unwanted subscription renewals by 30%
- Increase monthly active users by 25%
- Improve feature adoption rate by 20%
- Reduce user churn by 15%

1.4 Stakeholders

- Product Owner
- End Users (Customers)
- Development Team
- QA Team
- Business Stakeholders

1.5 In Scope

- User onboarding and authentication
- Subscription add/edit/delete
- Renewal reminder notifications
- Spending dashboard and insights

- Subscription pause/cancel controls

1.6 Out of Scope

- Direct subscription cancellation with third-party providers
- Payment processing
- Integration with bank accounts (Phase 2)

1.7 Assumptions

- Users will manually add subscription details
- Internet access is required for notifications

1.8 Constraints

- Limited development resources
 - Mobile notification dependency
-

2. Functional Requirement Document (FRD)

2.1 User Onboarding

- System shall allow users to register using email and password
- System shall authenticate users securely
- System shall allow password reset via email

2.2 Subscription Management

- System shall allow users to add a subscription with mandatory fields (name, renewal date)
- System shall allow users to edit subscription details
- System shall allow users to delete subscriptions

2.3 Notifications

- System shall send renewal reminders 7, 3, and 1 day before renewal
- System shall allow users to enable or disable notifications

2.4 Dashboard & Analytics

- System shall display all active subscriptions
- System shall calculate and display total monthly spending

2.5 Settings & Controls

- System shall allow users to pause subscription tracking
- System shall allow users to mark subscriptions as cancelled

2.6 Error Handling

- System shall display validation messages for incorrect inputs
 - System shall log errors for system failures
-

3. Product Requirement Document (PRD)

3.1 Product Overview

The Subscription Management Platform is a user-centric product designed to give individuals full control over their recurring subscriptions through tracking, reminders, and spending insights.

3.2 User Personas

Persona 1: Working Professional – Manages multiple OTT and utility subscriptions

Persona 2: Student – Budget-conscious with education subscriptions

3.3 User Journeys

1. User signs up and logs in
2. User adds subscriptions
3. User receives renewal reminders
4. User reviews dashboard insights
5. User pauses or cancels tracking if needed

3.4 Features & Requirements

Feature 1: User Onboarding

- Simple signup and login flow
- Secure authentication

Feature 2: Subscription Tracking

- Add, edit, delete subscriptions
- Categorization of subscriptions

Feature 3: Smart Notifications

- Configurable reminder alerts
- Timely renewal notifications

Feature 4: Dashboard & Insights

- Subscription overview
- Monthly spending summary

Feature 5: User Controls

- Pause reminders
- Cancel tracking

3.5 Non-Functional Requirements

- System should support up to 100k users
- Page load time < 3 seconds
- Data encryption for sensitive information

3.6 Risks & Mitigation

- Risk: Users forget to add subscriptions
 - Mitigation: Simple UI and onboarding tips

3.7 Release Plan

- Release 1: MVP with core subscription tracking
 - Release 2: Analytics enhancements
 - Release 3: Integrations (future scope)
-

4. Definition of Done

- All acceptance criteria met
- QA testing completed
- Features demoed in sprint review
- Product Owner approval received