Project Planning & Management

Project Proposal

Project Overview

Kite is a one-vendor mobile e-commerce application built using Flutter. The app provides users with a smooth shopping experience, enabling them to browse products, search, add items to favourites or cart, place orders, communicate with the admin, and manage their profile.

The application aims to deliver:

- A modern and intuitive shopping UI.
- High performance and smooth navigation.
- Easy and secure checkout process.
- Clear communication between users and admin.
- Reliable order tracking.

Project Goals & Objectives

- Build a scalable and user-friendly e-commerce app using Flutter.
- Provide all essential e-commerce functionalities in a single unified system.
- Create a smooth buying process from product discovery to checkout.
- Implement chat support for real-time communication with the admin.
- Support order tracking and product reviews.
- Ensure data security and high performance.

Project Scope

A. User App – In-Scope Items

- **Development of a Flutter mobile app (Android, optionally iOS).**
- **User authentication and profile management.**
- **Product browsing, searching, filtering, and viewing.**
- **Product interactions: add/remove cart, add/remove favourites.**
- **Checkout and order placement.**
- **Product reviews and ratings.**
- **Real-time chat with admin.**
- **Support/Help center.**
- **Integration with Firestore.**

B. Admin App – In-Scope Items

- **Add new products.**
- **Edit existing products.**
- **Delete products.**
- **View orders made by users.**
- **Change order status (Pending, Processing, Shipped, Delivered, Cancelled).**
- **Chat with users (reply to messages).**
- **Manage discount codes (later).**
- **View basic statistics (later).**

Out-of-Scope Items (For Both Apps)

- **Multi-vendor functionality.**
- **Advanced payment gateways.**
- **AI-powered recommendations.**
- **Web dashboard for admin.**
- **GPS delivery tracking.**
- **Order tracking.**

Project Plan

The **Kite Project Plan** outlines the phases, responsibilities, and workflow for both the **User App** and the **Admin App**. It ensures the project is completed on time (25 Aug 2025 – 5 Dec 2025) by a team of 4 developers.

1. Planning & Requirements Phase (25 Aug – 5 Sep)

User App:

- Gather functional requirements (browsing, cart, checkout, chat, profile management).
- Define user personas and usage scenarios.
- Prepare initial app flow and navigation.

Admin App:

- Define admin roles and permissions.
- Requirements for product management, order handling, and chat.

Team Tasks:

- Create project proposal.
- Complete documentation (Scope, SRS, Architecture).

2. UI/UX Design Phase (5 Sep – 20 Sep)

User App:

- Wireframes for home, search, product details, cart, checkout, profile.
- Design modern e-commerce interface.

Admin App:

- Simple and clean dashboard interface.
- Screens for product management and order control.

Team Tasks:

- Align color palettes, typography, and brand identity.
- Build Figma prototypes for both apps.

3. Frontend Development Phase (20 Sep – 20 Oct)

User App:

- Implement UI screens.
- Navigation and state management.

- Integrate cart, favourites, reviews, orders.

Admin App:

- Build CRUD product screens.
- Implement order control UI.
- Build chat interface.

Team Tasks:

- Code structure using clean architecture.
- Reusable shared components.

4. Firebase Integration (20 Oct – 10 Nov)

User App:

- Firestore setup for users, orders, cart, product reviews.
- Chat system with admin.
- Payment method logic.

Admin App:

- Product collection control.
- Real-time order updates.
- Admin-side chat integration.

Team Tasks:

- Security rules setup.
- Authentication methods.

5. Testing & QA (10 Nov – 25 Nov)

User App:

- Test navigation, checkout, search, chat.
- Review performance and loading.

Admin App:

- Test product CRUD, order updates, chat.

Team Tasks:

- Fix bugs.
- Improve UI consistency.
- Stress test Firestore operations.

6. Final Review & Delivery (25 Nov – 5 Dec)

- Final documentation.
- Final presentation.
- Build APK for both apps.
- Prepare project for submission.

Risk Assessment & Mitigation Plan

1. Technical Risks

Risk: Firebase performance issues or quota limits.
Impact: Slow loading, app crashes.
Mitigation:

- Optimize Firestore queries.
- Use pagination & caching.
- Monitor Firebase usage regularly.

Risk: State management complexity leading to bugs.
Mitigation:

- Use a clear architecture (Core & Feature Structure).
- Apply consistent coding standards.

2. Development Risks

Risk: Delay in UI/UX design approval.
Mitigation:

- Set early milestone reviews.
- Use quick iterations on Figma.

Risk: Developers unfamiliar with some features (chat, real-time updates).
Mitigation:

- Allocate research time.
- Divide tasks based on strengths.

3. Scope & Requirements Risks

Risk: Adding new features late in the project.
Mitigation:

- Freeze scope after planning phase.
- Document all changes formally.

Risk: Misunderstanding admin app requirements.
Mitigation:

- Separate scope sections for user vs admin.
- Early demos of admin tools.

Stakeholders

1. Customers (End Users)

Primary users of the application. They browse products, place orders, make payments, track shipments, and provide reviews. Their satisfaction directly determines the platform's success.

2. Admin (Platform Administrator)

Responsible for managing the overall system. Admins handle product moderation, user management, order tracking, analytics monitoring, and ensuring the platform operates smoothly and securely.

3. Customer Support Team

Handles user inquiries, complaints, product return requests, and order issues. They are essential for maintaining customer trust and satisfaction.

4. Development & IT Team

Includes mobile developers, backend engineers, UI/UX designers, DevOps, and QA testers. They design, implement, test, and maintain the application and ensure system stability, performance, and security.

5. Business Owners / Project Stakeholders

Decision-makers who define the business model, set priorities, allocate budget, establish partnerships, and monitor KPIs to ensure long-term growth.

Firebase Database Design

1) Users Collection

Document ID: userId (String)

Field	Type	Description
name	**String**	**Full name of user**
email	**String**	**User email**
photoURL	**String**	**Profile picture URL**
address	**List[String]**	**List of addresses**
favorite	**List [String]**	**List of product IDs marked favorite**
cart	**List [ProductSelected]**	**Products currently in cart**
createdAt	**Timestamp**	**Registration date**
ordersId	**List [String]**	**List of order IDs**

2) Products Collection

Document ID: productId (String)

Field	Type	Description
name	String	Product name
price	double	Price
photoURL	String	Image URL
comments	**List** [String]	List of review IDs
rate	double	Average rating
reviews	int	Total number of reviews
quantity	int	Available quantity
brand	String	Brand name
category	String	Product category
description	String	Product description
instruction	**List** [String]	Usage instructions
productAttributeType	String (enum)	Attribute type (size, color…)
stock	Map<String,dynamic>	Stock details per variant

date	Timestamp	Date added

3) Reviews Collection

Document ID: reviewId (String)

Field	Type	Description
productId	String	Product reference
date	Timestamp	Review date
senderId	String	User ID of reviewer
message	String	Review message
reactNum	int	Number of likes/reactions
rate	int	Rating value (1–5)

4) Orders Collection

Document ID: orderId (String)

Field	Type	Description
userId	String	Buyer user ID
productIds	**List** [String]	List of purchased product IDs
totalPrice	double	Total price
date	Timestamp	Order date
status	String (enum)	Order status (pending/shipped/delivered)
orderNumber	int	Static order number
paymentMethod	String (enum)	Payment method

5) Chat Collection

Document ID: chatId (String)

Field	Type	Description
senderId	String	User ID of sender
receiverId	String	User ID of receiver
message	String	Chat message
date	Timestamp	Message date

6) ProductSelected Structure

Document ID: ProductId (String)

Field	Type	Description
id	String	Product ID
ProductId	String	Selected product ID
name	String	Product name
price	double	Product price
photoURL	String	Image URL
brand	String	Brand name
category	String	Product category
productDetails	Map<String,dynamic>	Variant, size, color, or other dynamic details

UI/UX Design & Prototyping

Figma Design:

https://www.figma.com/make/RtqBvqfSabjT0F3PEl2AIj/Modern-Shopping-App-Design?t=rTEUZFZXWyvNmY2y&fullscreen=1

Prototyping & Mockup:

https://www.canva.com/design/DAG4t2F3tww/E_YlyycZFV-yj8_qU0m9NA/edit?utm_content=DAG4t2F3tww&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

UI/UX Guidelines: Color Scheme & Typography Standards

Color Scheme Standards

Color Name	Hex Code	Purpose / Context
Primary	#2E7D32	Main brand color, used for primary actions and key elements.
Accent	#81C784	Secondary brand color, used for less critical actions or highlights.
Foreground / Dark	#1A1A1A	Main text and foreground elements (High Contrast).
Muted	#4B5563	Secondary text, hints, or disabled elements (Low Contrast).
Success	#16A34A	Used for successful actions or positive confirmations (e.g., status messages).
Destructive / Red	#DC2626	Used for critical or irreversible actions (e.g., Delete buttons).
Background (Gradient Start)	#E8F5E9	Lighter start color for background gradients.
Background (Gradient End)	#B9FBC0	Darker end color for background gradients.
Background (Auth)	#D6EFD8	Specific background color for authentication screens (Login/Signup).
Primary Text Color	#80AF81	Used specifically for certain text styles to match the primary tone.

Typography Standards

Font Family: Montserrat

Size (px)	Weight	Context/Use	Example Style Name
30	Bold (w700)	Main page titles, high-impact branding.	styleBold30Primary
24	Semi-Bold (w600) & Bold (w700)	Section Titles, Headers (H1).	styleSemiBold24Dark, styleBold24Dark
20	Semi-Bold (w600) & Bold (w700)	Secondary Headers (H2), Important Metrics.	styleSemiBold20Dark, styleBold20Primary
18	Semi-Bold (w600) & Medium (w500)	Sub-headers, Primary action text.	styleSemiBold18Dark, styleMedium18Muted
16	Medium (w500) & Regular (w400)	Standard body text, Key list items.	styleMedium16Dark, styleRegular16Muted
14	Medium (w500) & Regular (w400)	Secondary body text, Labels, Captions.	styleMedium14Dark, styleRegular14Muted
12	Medium (w500), Regular (w400) & Semi-Bold (w600)	Smallest functional text, Footer notes, Destructive actions.	styleRegular12Muted, styleSemiBold12Red