To create a comprehensive specification for an advanced To-Do List application in Flutter that adheres to SOLID principles and clean code architecture, we can break down the project into several key components and features. Based on your requirements, here's a detailed specification:

Project Overview
Project Name: Advanced To-Do List Application

Objective: Develop a robust, feature-rich To-Do list application with task management capabilities, synchronization with local and remote databases, and notification features. Ensure the code adheres to SOLID principles and employs clean code architecture.

Core Features
1. **User Authentication**
   - Sign Up / Sign In
   - Password recovery
   - Social Media Login (Google, Facebook)

2. **Task Management**
   - Create, Read, Update, Delete (CRUD) operations for tasks
   - Task categorization (Work, Personal, Shopping, etc.)
   - Task prioritization (High, Medium, Low)
   - Due dates and reminders
   - Recurring tasks
   - Attachments (images, files) to tasks

3. **Synchronization**
   - Local database (SQLite)
   - Remote database (Firebase Firestore)
   - Offline access with automatic sync when online

4. **Notifications**
   - Push notifications for upcoming tasks
   - Daily summary notifications
   - In-app notifications

5. **User Interface**
   - Responsive design
   - Dark mode support
   - Customizable themes
   - Drag-and-drop for task reordering
   - Search and filter functionality

6. **Analytics**

  - Task completion statistics
  - Weekly and monthly productivity reports

7. **Settings**
  - Profile management
  - Notification settings
  - Data backup and restore
  - Language preferences (localization)

Technical Specifications

1. **Architecture**
  - **Clean Architecture**
    - Presentation Layer (UI): Flutter Widgets
    - Domain Layer (Business Logic): Use cases, Repositories
    - Data Layer (Data Access): Local and remote data sources, Models
  - **SOLID Principles**
    - Single Responsibility Principle (SRP)
    - Open/Closed Principle (OCP)
    - Liskov Substitution Principle (LSP)
    - Interface Segregation Principle (ISP)
    - Dependency Inversion Principle (DIP)

2. **Libraries and Tools**
  - **State Management:** Provider, Riverpod, or Bloc
  - **Database:** SQLite for local storage, Firebase Firestore for remote storage
  - **Notifications:** Firebase Cloud Messaging (FCM), Local Notifications plugin
  - **Authentication:** Firebase Authentication
  - **Networking:** Dio or Http package
  - **Dependency Injection:** GetIt
  - **Routing:** AutoRoute or Fluro
  - **Testing:** Flutter Test, Mockito, Integration Testing

3. **Development Environment**
  - IDE: Android Studio or Visual Studio Code
  - Version Control: Git, GitHub or GitLab for repository management
  - CI/CD: GitHub Actions or GitLab CI

User Stories

1. **User Authentication**
  - As a user, I want to sign up with my email and password.
  - As a user, I want to sign in with my email and password.
  - As a user, I want to recover my password if I forget it.
  - As a user, I want to sign in using my Google or Facebook account.

2. **Task Management**
   - As a user, I want to create a new task with a title, description, category, and priority.
   - As a user, I want to edit an existing task.
   - As a user, I want to delete a task.
   - As a user, I want to view a list of my tasks, sorted by due date or priority.
   - As a user, I want to attach images or files to my tasks.
   - As a user, I want to set due dates and reminders for my tasks.
   - As a user, I want my tasks to sync across my devices.

3. **Notifications**
   - As a user, I want to receive push notifications for upcoming tasks.
   - As a user, I want to receive a daily summary of my tasks.
   - As a user, I want to receive in-app notifications for tasks.

4. **Analytics**
   - As a user, I want to see my task completion statistics.
   - As a user, I want to see weekly and monthly productivity reports.

5. **Settings**
   - As a user, I want to manage my profile information.
   - As a user, I want to customize my notification settings.
   - As a user, I want to back up and restore my data.
   - As a user, I want to change the app's language.

Milestones

1. **Milestone 1:** Project Setup and Basic UI
   - Set up Flutter project
   - Implement basic UI for authentication and task management

2. **Milestone 2:** Local Storage and Task Management
   - Implement SQLite for local storage
   - CRUD operations for tasks
   - Task categorization and prioritization

3. **Milestone 3:** Remote Storage and Synchronization
   - Implement Firebase Firestore for remote storage
   - Synchronize tasks between local and remote databases
   - Handle offline access and automatic sync

4. **Milestone 4:** Notifications
   - Implement push notifications using Firebase Cloud Messaging
   - Implement local notifications for reminders
   - In-app notifications

5. **Milestone 5:** Advanced Features and Settings
   - Implement task attachments
   - Implement analytics and productivity reports
   - Implement user settings and preferences

6. **Milestone 6:** Testing and Deployment
   - Unit and integration testing
   - CI/CD setup
   - App Store and Google Play Store deployment

This detailed specification should guide the development of your advanced To-Do list application in Flutter, ensuring a comprehensive and well-structured approach.