**1. customer App**

**Purpose**: Centralized user management, roles/permissions, advanced authentication, profiles.

1. **User (Custom Django User)**
   * **Description**: Extends Django’s AbstractUser or AbstractBaseUser to store additional fields for KobraSuite users.
   * **Fields**:
     + username: CharField(max\_length=150, unique=True) (if using AbstractUser)
     + email: EmailField(unique=True)
     + phone\_number: CharField(max\_length=20, null=True, blank=True)
     + is\_email\_verified: BooleanField(default=False)
     + is\_phone\_verified: BooleanField(default=False)
     + (plus the standard Django user fields like password, first\_name, last\_name, etc.)
   * **Notes**:
     + Consider making email the unique identifier for authentication.
     + Could integrate with multi-factor or third-party authentication flows.
2. **Role**
   * **Description**: Custom Role-Based Access Control (RBAC) roles (e.g., “Family Admin,” “Finance Manager,” “Work Lead”).
   * **Fields**:
     + name: CharField(max\_length=50, unique=True)
     + description: TextField(null=True, blank=True)
   * **Relationships**:
     + users = ManyToManyField(User, related\_name='roles', blank=True)
3. **MFAConfig**
   * **Description**: Stores multi-factor authentication details for a user.
   * **Fields**:
     + user: OneToOneField(User, on\_delete=CASCADE)
     + mfa\_enabled: BooleanField(default=False)
     + mfa\_type: CharField(max\_length=50, choices=[('TOTP', 'Time-Based OTP'), ('SMS', 'SMS'), ('PUSH', 'Push Notification'), ...])
     + secret\_key: CharField(max\_length=255, null=True, blank=True) (for TOTP or backup codes)
   * **Notes**:
     + Additional fields for tracking enrollment date or last verification time.
4. **UserProfile**
   * **Description**: Extended profile with optional personal details not in the main User model.
   * **Fields**:
     + user: OneToOneField(User, on\_delete=CASCADE)
     + date\_of\_birth: DateField(null=True, blank=True)
     + address: TextField(null=True, blank=True)
     + profile\_picture: ImageField(upload\_to='profile\_pics/', null=True, blank=True)
     + preferences: JSONField(null=True, blank=True) (to store user custom settings)
5. **SecureDocument**
   * **Description**: Stores metadata for user documents (e.g., passports, birth certificates, finance statements).
   * **Fields**:
     + user: ForeignKey(User, on\_delete=CASCADE, related\_name='documents')
     + title: CharField(max\_length=100)
     + file: FileField(upload\_to='secure\_documents/')
     + description: TextField(null=True, blank=True)
     + created\_at: DateTimeField(auto\_now\_add=True)

**2. finances App**

**Purpose**: Budgeting, expense tracking, income management, debt, and other financial features.

1. **BankAccount**
   * **Description**: Links a user to external or internal bank accounts for real-time data or manual tracking.
   * **Fields**:
     + user: ForeignKey(customer.User, on\_delete=CASCADE, related\_name='bank\_accounts')
     + account\_name: CharField(max\_length=100)
     + account\_number: CharField(max\_length=50)
     + institution\_name: CharField(max\_length=100)
     + balance: DecimalField(max\_digits=12, decimal\_places=2, default=0.00)
     + currency: CharField(max\_length=10, default='USD')
     + last\_synced: DateTimeField(null=True, blank=True)
2. **Budget**
   * **Description**: Represents a user’s monthly/annual budget with multiple categories.
   * **Fields**:
     + user: ForeignKey(customer.User, on\_delete=CASCADE, related\_name='budgets')
     + name: CharField(max\_length=100)
     + total\_amount: DecimalField(max\_digits=12, decimal\_places=2)
     + start\_date: DateField()
     + end\_date: DateField()
     + is\_active: BooleanField(default=True)
3. **BudgetCategory**
   * **Description**: Sub-categories within a Budget (e.g., “Groceries,” “Rent,” “Entertainment”).
   * **Fields**:
     + budget: ForeignKey(Budget, on\_delete=CASCADE, related\_name='categories')
     + name: CharField(max\_length=100)
     + allocated\_amount: DecimalField(max\_digits=12, decimal\_places=2, default=0.00)
4. **Transaction**
   * **Description**: Stores financial transactions (expenses, income, transfers).
   * **Fields**:
     + user: ForeignKey(customer.User, on\_delete=CASCADE, related\_name='transactions')
     + bank\_account: ForeignKey(BankAccount, on\_delete=CASCADE, null=True, blank=True)
     + budget\_category: ForeignKey(BudgetCategory, on\_delete=SET\_NULL, null=True, blank=True)
     + transaction\_type: CharField(max\_length=50, choices=[('INCOME','Income'),('EXPENSE','Expense'),('TRANSFER','Transfer')])
     + amount: DecimalField(max\_digits=12, decimal\_places=2)
     + description: TextField(null=True, blank=True)
     + date: DateField()
     + created\_at: DateTimeField(auto\_now\_add=True)
5. **Bill**
   * **Description**: Recurring or one-time bills (utilities, subscriptions, loan payments).
   * **Fields**:
     + user: ForeignKey(customer.User, on\_delete=CASCADE, related\_name='bills')
     + name: CharField(max\_length=100)
     + amount: DecimalField(max\_digits=12, decimal\_places=2)
     + due\_date: DateField()
     + recurring\_interval: CharField(max\_length=50, choices=[('MONTHLY','Monthly'),('YEARLY','Yearly'),('WEEKLY','Weekly'),('NONE','None')], default='NONE')
     + is\_auto\_pay\_enabled: BooleanField(default=False)
6. **Debt**
   * **Description**: Tracks loans, credit card balances, mortgages, etc.
   * **Fields**:
     + user: ForeignKey(customer.User, on\_delete=CASCADE, related\_name='debts')
     + name: CharField(max\_length=100)
     + principal\_amount: DecimalField(max\_digits=12, decimal\_places=2)
     + remaining\_balance: DecimalField(max\_digits=12, decimal\_places=2)
     + interest\_rate: DecimalField(max\_digits=5, decimal\_places=2)
     + due\_date: DateField(null=True, blank=True)
     + debt\_type: CharField(max\_length=50, choices=[('CREDIT\_CARD','Credit Card'),('LOAN','Loan'),('MORTGAGE','Mortgage'), ...])
     + created\_at: DateTimeField(auto\_now\_add=True)
7. **SavingsGoal**
   * **Description**: Tracks a goal-oriented savings target.
   * **Fields**:
     + user: ForeignKey(customer.User, on\_delete=CASCADE, related\_name='savings\_goals')
     + name: CharField(max\_length=100)
     + target\_amount: DecimalField(max\_digits=12, decimal\_places=2)
     + current\_amount: DecimalField(max\_digits=12, decimal\_places=2, default=0.00)
     + deadline: DateField(null=True, blank=True)
     + created\_at: DateTimeField(auto\_now\_add=True)

**3. homelife App**

**Purpose**: Manage family or household tasks, calendars, chores, IoT data, meal planning, etc.

1. **Household**
   * **Description**: Represents a household that can include multiple users (e.g., family members).
   * **Fields**:
     + name: CharField(max\_length=100)
     + created\_at: DateTimeField(auto\_now\_add=True)
   * **Relationships**:
     + members = ManyToManyField(customer.User, related\_name='households', blank=True)
2. **Chore**
   * **Description**: Individual chore or household task (laundry, dishes, etc.).
   * **Fields**:
     + household: ForeignKey(Household, on\_delete=CASCADE, related\_name='chores')
     + title: CharField(max\_length=100)
     + description: TextField(null=True, blank=True)
     + frequency: CharField(max\_length=50, choices=[('DAILY','Daily'),('WEEKLY','Weekly'),('MONTHLY','Monthly'),('ONE\_TIME','One Time')], default='ONE\_TIME')
     + priority: IntegerField(default=1) (or use a choices for priority levels)
     + due\_date: DateField(null=True, blank=True)
     + assigned\_to: ForeignKey(customer.User, on\_delete=SET\_NULL, null=True, blank=True)
3. **SharedCalendarEvent**
   * **Description**: Events on a shared family calendar (e.g., doctor appointments, birthdays).
   * **Fields**:
     + household: ForeignKey(Household, on\_delete=CASCADE, related\_name='calendar\_events')
     + title: CharField(max\_length=100)
     + start\_datetime: DateTimeField()
     + end\_datetime: DateTimeField()
     + description: TextField(null=True, blank=True)
     + location: CharField(max\_length=200, null=True, blank=True)
4. **MealPlan**
   * **Description**: Represents a planned meal (daily/weekly) for the household.
   * **Fields**:
     + household: ForeignKey(Household, on\_delete=CASCADE, related\_name='meal\_plans')
     + date: DateField()
     + meal\_type: CharField(max\_length=50, choices=[('BREAKFAST','Breakfast'),('LUNCH','Lunch'),('DINNER','Dinner'),('SNACK','Snack')])
     + recipe\_name: CharField(max\_length=100)
     + notes: TextField(null=True, blank=True)
5. **GroceryItem**
   * **Description**: Items needed for the household’s grocery list.
   * **Fields**:
     + household: ForeignKey(Household, on\_delete=CASCADE, related\_name='grocery\_items')
     + name: CharField(max\_length=100)
     + quantity: CharField(max\_length=50, null=True, blank=True) (e.g., “2 lbs”, “1 bag”)
     + purchased: BooleanField(default=False)
6. **IoTDevice**
   * **Description**: Represents a connected device in the home (thermostat, camera, fridge, etc.).
   * **Fields**:
     + household: ForeignKey(Household, on\_delete=CASCADE, related\_name='iot\_devices')
     + device\_name: CharField(max\_length=100)
     + device\_type: CharField(max\_length=100) (e.g., “Thermostat,” “Security Camera”)
     + identifier: CharField(max\_length=100, unique=True) (device serial or MAC address)
     + last\_online: DateTimeField(null=True, blank=True)
7. **IoTDataLog**
   * **Description**: Historical sensor data or events from an IoT device.
   * **Fields**:
     + iot\_device: ForeignKey(IoTDevice, on\_delete=CASCADE, related\_name='data\_logs')
     + timestamp: DateTimeField(auto\_now\_add=True)
     + data: JSONField() (could store temperature, motion event, energy usage, etc.)

**4. investing App**

**Purpose**: Portfolio management, transactions, assets, market data, watchlists, etc.

1. **Portfolio**
   * **Description**: A user’s investment portfolio, which can have multiple assets (stocks, crypto, etc.).
   * **Fields**:
     + user: ForeignKey(customer.User, on\_delete=CASCADE, related\_name='portfolios')
     + name: CharField(max\_length=100)
     + created\_at: DateTimeField(auto\_now\_add=True)
     + description: TextField(null=True, blank=True)
2. **Asset**
   * **Description**: Represents an investable asset (stock, ETF, crypto, real estate, etc.).
   * **Fields**:
     + symbol: CharField(max\_length=50, unique=True) (e.g., “AAPL,” “BTC,” “VTSAX”)
     + asset\_type: CharField(max\_length=50, choices=[('STOCK','Stock'),('CRYPTO','Crypto'),('ETF','ETF'),('REIT','REIT'), ...])
     + name: CharField(max\_length=100)
     + description: TextField(null=True, blank=True)
3. **Holding**
   * **Description**: Many-to-many “through” relationship between Portfolio and Asset to track quantity/cost basis.
   * **Fields**:
     + portfolio: ForeignKey(Portfolio, on\_delete=CASCADE, related\_name='holdings')
     + asset: ForeignKey(Asset, on\_delete=CASCADE, related\_name='holdings')
     + quantity: DecimalField(max\_digits=12, decimal\_places=4, default=0.0000)
     + average\_cost: DecimalField(max\_digits=12, decimal\_places=4, default=0.0000)
     + created\_at: DateTimeField(auto\_now\_add=True)
4. **TradeTransaction**
   * **Description**: Records buy/sell transactions within a portfolio.
   * **Fields**:
     + holding: ForeignKey(Holding, on\_delete=CASCADE, related\_name='transactions')
     + transaction\_type: CharField(max\_length=4, choices=[('BUY','Buy'),('SELL','Sell')])
     + quantity: DecimalField(max\_digits=12, decimal\_places=4)
     + price\_per\_unit: DecimalField(max\_digits=12, decimal\_places=4)
     + transaction\_date: DateTimeField()
     + fees: DecimalField(max\_digits=12, decimal\_places=2, default=0.00)
5. **Watchlist**
   * **Description**: A user’s watchlist to track potential assets of interest.
   * **Fields**:
     + user: ForeignKey(customer.User, on\_delete=CASCADE, related\_name='watchlists')
     + name: CharField(max\_length=100)
     + created\_at: DateTimeField(auto\_now\_add=True)
   * **Relationships**:
     + assets = ManyToManyField(Asset, related\_name='watchlisted\_by')
6. **MarketData**
   * **Description**: Stores daily or real-time data for an asset (for charting, analytics).
   * **Fields**:
     + asset: ForeignKey(Asset, on\_delete=CASCADE, related\_name='market\_data')
     + date: DateField()
     + open\_price: DecimalField(max\_digits=12, decimal\_places=4)
     + high\_price: DecimalField(max\_digits=12, decimal\_places=4)
     + low\_price: DecimalField(max\_digits=12, decimal\_places=4)
     + close\_price: DecimalField(max\_digits=12, decimal\_places=4)
     + volume: BigIntegerField()

**5. school App**

**Purpose**: Handle courses, assignments, notes, study tools, collaboration features.

1. **Course**
   * **Description**: Represents a course/class a user is enrolled in.
   * **Fields**:
     + title: CharField(max\_length=200)
     + description: TextField(null=True, blank=True)
     + start\_date: DateField()
     + end\_date: DateField(null=True, blank=True)
   * **Relationships**:
     + students = ManyToManyField(customer.User, related\_name='courses', blank=True)
2. **Assignment**
   * **Description**: Tasks or homework linked to a course.
   * **Fields**:
     + course: ForeignKey(Course, on\_delete=CASCADE, related\_name='assignments')
     + title: CharField(max\_length=200)
     + description: TextField(null=True, blank=True)
     + due\_date: DateTimeField()
     + created\_at: DateTimeField(auto\_now\_add=True)
3. **Submission**
   * **Description**: A student’s submission for an assignment.
   * **Fields**:
     + assignment: ForeignKey(Assignment, on\_delete=CASCADE, related\_name='submissions')
     + student: ForeignKey(customer.User, on\_delete=CASCADE, related\_name='submissions')
     + file: FileField(upload\_to='assignment\_submissions/', null=True, blank=True)
     + text\_answer: TextField(null=True, blank=True)
     + submitted\_at: DateTimeField(auto\_now\_add=True)
     + grade: DecimalField(max\_digits=5, decimal\_places=2, null=True, blank=True)
4. **StudyGroup**
   * **Description**: A small group for collaborative learning.
   * **Fields**:
     + name: CharField(max\_length=100)
     + course: ForeignKey(Course, on\_delete=CASCADE, related\_name='study\_groups')
     + members = ManyToManyField(customer.User, related\_name='study\_groups', blank=True)
5. **StudyNote**
   * **Description**: Notes for a course or an assignment; can be shared with others.
   * **Fields**:
     + author: ForeignKey(customer.User, on\_delete=CASCADE, related\_name='study\_notes')
     + course: ForeignKey(Course, on\_delete=CASCADE, null=True, blank=True, related\_name='notes')
     + assignment: ForeignKey(Assignment, on\_delete=CASCADE, null=True, blank=True, related\_name='notes')
     + content: TextField()
     + created\_at: DateTimeField(auto\_now\_add=True)
     + shared\_with: ManyToManyField(customer.User, blank=True, related\_name='received\_notes')

**6. work App**

**Purpose**: Professional tasks, projects, timesheets, meeting notes, etc.

1. **Team**
   * **Description**: Group of users collaborating on work projects.
   * **Fields**:
     + name: CharField(max\_length=100)
     + description: TextField(null=True, blank=True)
     + created\_at: DateTimeField(auto\_now\_add=True)
   * **Relationships**:
     + members = ManyToManyField(customer.User, related\_name='teams', blank=True)
2. **Project**
   * **Description**: A project within a work context, possibly linked to a Team.
   * **Fields**:
     + team: ForeignKey(Team, on\_delete=CASCADE, null=True, blank=True, related\_name='projects')
     + name: CharField(max\_length=200)
     + description: TextField(null=True, blank=True)
     + start\_date: DateField()
     + end\_date: DateField(null=True, blank=True)
     + is\_active: BooleanField(default=True)
3. **WorkTask**
   * **Description**: Individual tasks under a project.
   * **Fields**:
     + project: ForeignKey(Project, on\_delete=CASCADE, related\_name='tasks')
     + title: CharField(max\_length=200)
     + description: TextField(null=True, blank=True)
     + assignee: ForeignKey(customer.User, on\_delete=SET\_NULL, null=True, blank=True, related\_name='assigned\_tasks')
     + status: CharField(max\_length=50, choices=[('TODO','To Do'),('IN\_PROGRESS','In Progress'),('DONE','Done')], default='TODO')
     + priority: IntegerField(default=1) (or use a choices field)
     + due\_date: DateField(null=True, blank=True)
     + created\_at: DateTimeField(auto\_now\_add=True)
4. **Timesheet**
   * **Description**: Logs hours worked by a user on a specific task or project.
   * **Fields**:
     + user: ForeignKey(customer.User, on\_delete=CASCADE, related\_name='timesheets')
     + project: ForeignKey(Project, on\_delete=CASCADE, related\_name='timesheets')
     + task: ForeignKey(WorkTask, on\_delete=CASCADE, null=True, blank=True, related\_name='timesheets')
     + date: DateField()
     + hours\_worked: DecimalField(max\_digits=5, decimal\_places=2, default=0.00)
     + notes: TextField(null=True, blank=True)
5. **Meeting**
   * **Description**: Represents a scheduled meeting for a project or team.
   * **Fields**:
     + project: ForeignKey(Project, on\_delete=CASCADE, null=True, blank=True, related\_name='meetings')
     + team: ForeignKey(Team, on\_delete=CASCADE, null=True, blank=True, related\_name='meetings')
     + title: CharField(max\_length=200)
     + start\_time: DateTimeField()
     + end\_time: DateTimeField()
     + location: CharField(max\_length=200, null=True, blank=True)
     + created\_at: DateTimeField(auto\_now\_add=True)
   * **Relationships**:
     + attendees = ManyToManyField(customer.User, related\_name='meetings', blank=True)
6. **MeetingNote**
   * **Description**: Notes or minutes from a meeting.
   * **Fields**:
     + meeting: OneToOneField(Meeting, on\_delete=CASCADE, related\_name='notes')
     + content: TextField()
     + created\_at: DateTimeField(auto\_now\_add=True)
     + updated\_at: DateTimeField(auto\_now=True)

**7. api App**

**Purpose**: Handles external integrations, tokens, microservice references, etc.

1. **ApiToken**
   * **Description**: Stores tokens for external APIs (bank feeds, stock market data, etc.).
   * **Fields**:
     + user: ForeignKey(customer.User, on\_delete=CASCADE, related\_name='api\_tokens', null=True, blank=True)
     + token\_name: CharField(max\_length=100) (e.g., “Plaid Access Token”)
     + token\_value: TextField() (encrypted at rest)
     + created\_at: DateTimeField(auto\_now\_add=True)
     + expires\_at: DateTimeField(null=True, blank=True)
2. **MicroserviceReference**
   * **Description**: Describes an external or internal microservice used by the platform.
   * **Fields**:
     + name: CharField(max\_length=100)
     + base\_url: URLField()
     + description: TextField(null=True, blank=True)
     + api\_key: TextField(null=True, blank=True) (encrypted if storing secrets)
     + created\_at: DateTimeField(auto\_now\_add=True)
3. **IntegrationLog**
   * **Description**: Logging of events or data exchanges with external services.
   * **Fields**:
     + service: ForeignKey(MicroserviceReference, on\_delete=CASCADE, related\_name='logs')
     + request\_payload: JSONField(null=True, blank=True)
     + response\_payload: JSONField(null=True, blank=True)
     + status\_code: IntegerField()
     + timestamp: DateTimeField(auto\_now\_add=True)

**8. (Proposed) notifications App**

**Justification**: Central place to handle cross-module user notifications. This prevents scattering notification logic across multiple apps.

1. **Notification**
   * **Description**: A user-facing notification (finance alert, chore reminder, etc.).
   * **Fields**:
     + recipient: ForeignKey(customer.User, on\_delete=CASCADE, related\_name='notifications')
     + title: CharField(max\_length=200)
     + message: TextField()
     + created\_at: DateTimeField(auto\_now\_add=True)
     + read: BooleanField(default=False)
     + notification\_type: CharField(max\_length=50, choices=[('FINANCE','Finance'),('HOMELIFE','Homelife'),('WORK','Work'),('SCHOOL','School'), ...])
2. **NotificationPreference**
   * **Description**: Stores user preferences about how and when they want to receive notifications.
   * **Fields**:
     + user: OneToOneField(customer.User, on\_delete=CASCADE)
     + email\_notifications: BooleanField(default=True)
     + push\_notifications: BooleanField(default=True)
     + sms\_notifications: BooleanField(default=False)