**Project Team:**

- AHANYA

- CHANDY

- RUCHIT

**Problem Definition**

HomeHive aims to become a leading online marketplace for household products by offering unique features like cryptocurrency payments, shared carts, and group buying, while maintaining the familiar functionalities of established e-commerce platforms. However, several challenges need to be addressed to achieve this goal:

1.Competition: Standing out in the crowded e-commerce market requires differentiation from established platforms like Wayfair, Bed Bath & Beyond, and Amazon Home.

2.Technical complexity: Integrating innovative features like cryptocurrency payments, shared carts, and group buying involves significant development effort and expertise.

3.Security and regulations: Ensuring secure transactions with cryptocurrency and complying with evolving regulations pose legal and operational challenges.

4.Community building: Encouraging user engagement, reviews, and participation in social commerce features takes time and effort to establish.

**Requirement Specification**

The requirement specification should detail the functionalities and features of HomeHive, encompassing:

* User roles and permissions: Defining roles like admin, seller, buyer, and potentially moderator, with distinct functionalities for each.
* Product management: Specifying product listing, catalog management, search and filter options, and detailed product descriptions.
* Shopping and checkout: Outlining the shopping cart experience, checkout process with various payment options (including cryptocurrency), and order tracking functionalities.
* Communication and notifications: Implementing messaging between buyers and sellers, and notification systems for order status, promotions, and other activities.
* Additional features: Defining the specific functionalities of cryptocurrency payments, shared carts, and group buying features.

**Functional Requirements**

The functional requirements should list the specific actions and outcomes expected from the system, including:

* Users can register, login, and manage their profiles.
* Sellers can add, edit, and delete products, along with managing catalog details.
* Buyers can browse and search for products using advanced filters.
* Users can add and remove items from their shopping carts, including collaborating through shared carts.
* Users can complete orders using various payment options, including cryptocurrency.
* Users can track order status and access order history.
* Sellers and buyers can communicate through a messaging system.
* Users receive relevant notifications for order updates, promotions, and other events.

**System Requirements**

The system requirements should specify the hardware, software, and network infrastructure needed to support HomeHive, including:

* Programming languages and frameworks for front-end and back-end development (HTML, CSS, Javascript, Node.js, APIs)
* Database management system (MySQL)
* Security protocols for secure transactions and data protection
* Server and network infrastructure to handle traffic and processing

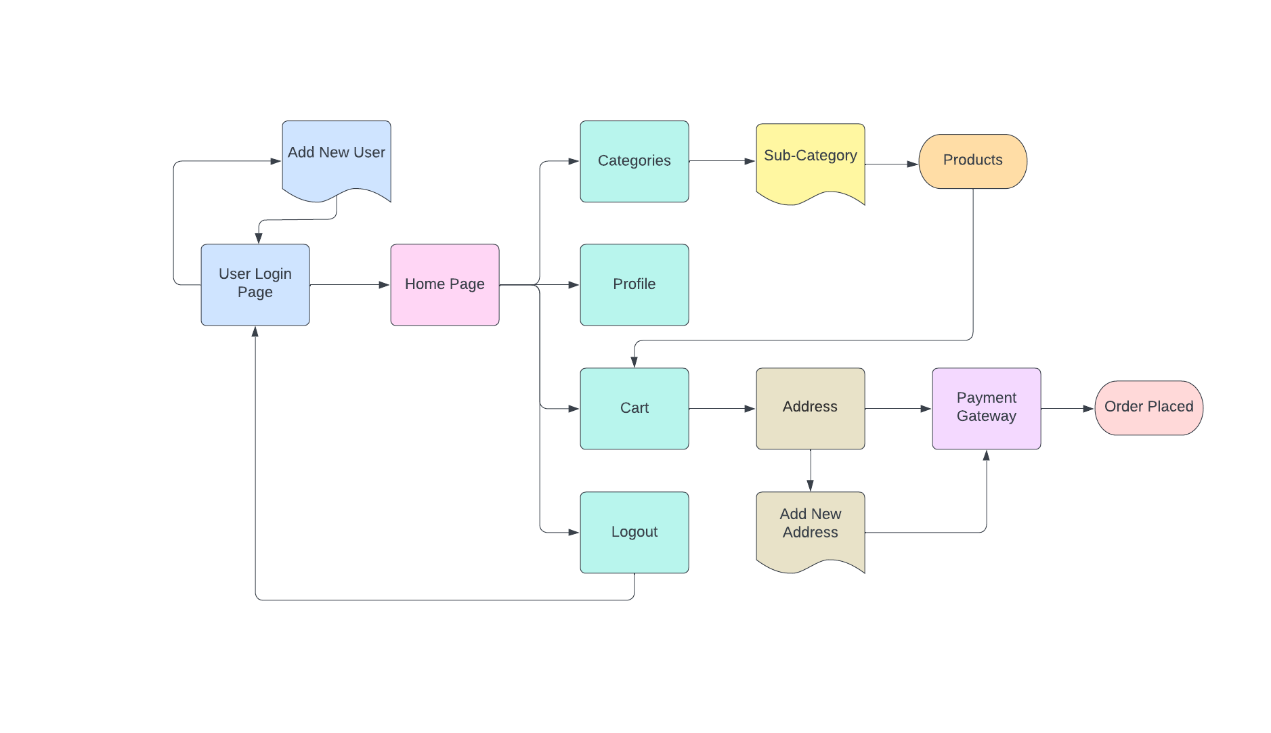
**Conceptual Model**

The conceptual model should provide a high-level overview of the system's architecture, including:

* Users (buyers, sellers, admins)
* Products and catalog data
* Shopping cart and checkout system
* Order management and tracking
* Communication and notification system
* Cryptocurrency payment integration

**Data Flow Diagram**

The data flow diagram should map the flow of data through the system, illustrating how data is captured, processed, and stored for each feature.



**ER Diagram**

The entity-relationship diagram should show the relationships between different entities in the system, such as users, products, orders, and payments.

**Planning and Scheduling**

Gantt charts and PERT charts can be used to visualize and plan the development process, outlining tasks, dependencies, and timelines for completing the project.

**Proposed Tools**

Based on the chosen technology stack, specific tools and libraries can be proposed for:

\* Front-end development (ReactJS, VueJS)

\* Back-end development (Express.js, Django)

\* API integration (Coinbase API, Stripe API)

\* Database management (MySQL Workbench)

\* Project management (Asana, Trello)

By addressing these areas and outlining the specific details, HomeHive can move towards a comprehensive and well-defined development plan.

Remember, this is a starting point, and you can refine and expand upon these sections based on further information and specific requirements.