4. **Communications Interfaces**

- **Network Requirements:**
  - HTTP/HTTPS protocols for secure data transfer.
  - WebSocket for real-time updates.
- **Security Standards:**
  - End-to-end encryption for sensitive data.
  - OAuth2 for user authentication with third-party services.

3.4. System Features

1. **Task and Chore Management**

- **Description and Priority**
  - Allow users to create, assign, and track household tasks and chores.
  - Set priorities, deadlines, and reminders for each task.
  - Mark tasks as completed and track the history of completed chores.
  - **Priority**: High
- **Stimulus/Response Sequences**
  - **Stimulus**: The user assigns a task.
    Response: The system records and schedules the task.
  - **Stimulus**: Deadline approaches

    Response: The system sends reminders.
  - **Stimulus**: The user sets a priority level for a task (e.g., high, medium, low).

    Response: The system highlights the task based on priority and sends reminders according to the urgency.
  - **Stimulus**: The user marks a task as in progress.

    Response: The system updates the task status and sends a notification to the user or relevant family members.
  - **Stimulus**: The user marks a task as completed.

    Response: The system updates the task status to "Completed," logs completion time, and notifies family members if needed.
  - **Stimulus**: The user cancels or deletes a task.

    Response: The system removes the task from the schedule and updates the task list.

- **Stimulus**: The user wants to delegate a task to another family member.

  Response: The system allows the task to be reassigned and sends a notification to the new assignee.

- **Stimulus**: The Task status is overdue.

  Response: The system sends an overdue reminder to the assigned user and updates the task list.

- **Stimulus**: The user assigns a recurring task (e.g., daily, weekly).

  Response: The system schedules the task to repeat according to the set frequency and sends regular reminders.

- **Functional Requirements**
  - **REQ-1:** The Users shall be able to create and assign household tasks.
  - **REQ-2:** The Users shall be able to set deadlines and reminders.
  - **REQ-3:** The users shall be able to mark tasks as in progress, completed, or canceled.
  - **REQ-4:** The system shall be able to send reminders to users when deadlines approach or are overdue.
  - **REQ-5:** The users shall be able to set priority levels for tasks (e.g., high, medium, low).
  - **REQ-6:** The users shall be able to view a list of assigned tasks, including status and due dates.
  - **REQ-7:** The users shall be able to track the progress of tasks through status updates.
  - **REQ-8:** The system shall be able to support recurring tasks with customizable frequencies (e.g., daily, weekly).
  - **REQ-9:** The users shall be able to delegate tasks to other family members.

2. Household Budgeting Tool

- **Description and Priority**
  - Enable users to set monthly or weekly budgets for various household categories.
  - Track spending and provide visual insights (charts, summaries) for budget adherence.
  - Notify users when spending nears the set limits in each category.
  - **Priority**: High
- **Stimulus/Response Sequences**

- **Stimulus**: The user sets a monthly or weekly budget for a household category.

  Response: The system begins tracking spending within the set limits for that category.

- **Stimulus**: Spending nears the set budget limit (e.g., 80%).

  Response: The system sends a notification to the user informing them of the nearing limit.

- **Stimulus**: The user spends money.

  Response: The system updates the spending in real time and adjusts the remaining budget.

- **Stimulus**: The user exceeds the set budget.

  Response: The system sends an alert to the user indicating the budget has been exceeded.

- **Stimulus**: The user adjusts the budget mid-month or week.

  Response: The system recalculates the remaining budget based on the new limit and updates the tracking.

- **Stimulus**: The user views the spending report.

  Response: The system generates a visual summary (e.g., chart or graph) showing current spending vs. the set budget.

- **Stimulus**: The user requests a summary of overall spending for the month or week.

  Response: The system provides a report on total spending and the remaining budget.

- **Stimulus**: The user sets a new notification threshold (e.g., 50% of the budget).

  Response: The system adjusts the notification settings and alerts the user when the new threshold is reached.

- **Stimulus**: Spending in a category nears the set budget limit (e.g., 80%).

  Response: The system sends a notification to the user informing them of the nearing limit.

- **Functional Requirements**
  - **REQ-1:** The users shall be able to set monthly or weekly budgets.
  - **REQ-2:** The system shall be able to track spending in real-time.
  - **REQ-3:** The system shall be able to provide visual representations

(e.g., charts, graphs) for budget adherence.

- **REQ-4:** The system shall be able to send notifications when spending reaches predefined thresholds (e.g., 80% of the budget).
- **REQ-5:** The system shall be able to allow users to adjust the budget mid-month or week and update the tracking accordingly.
- **REQ-6:** The users shall be able to view detailed spending reports and overall budget adherence.
- **REQ-7:** The system shall be able to provide insights and recommendations based on past spending patterns to help users manage their budgets.
- **REQ-8:** The system shall be able to ensure that users are alerted when the budget limit is exceeded.
- **REQ-9:** The system shall be able to allow users to set specific time intervals for receiving notifications (e.g., weekly, monthly).
- **REQ-10:** The users shall be able to set up recurring budgets and adjust them as needed.
- **REQ-11:** The system shall be able to provide a feature to track shared expenses between multiple users within the household.

3. Meal Planning and Recipe Suggestions

- **Description and Priority**
  - Automates meal planning by suggesting recipes based on available ingredients and user dietary preferences.
  - Adjusts recipes according to the quantities of ingredients on hand and offers modification options.
  - Tracks inventory in real-time to ensure recipe suggestions are always up to date.
  - Suggest alternative recipes when the ingredients are insufficient.
  - Allows users to save and organize their favorite recipes for easy access to personalized meals.
  - Minimize food waste by utilizing available ingredients efficiently.
  - Streamlines the meal planning process for convenience.
  - **Priority**: High.
- **Stimulus/Response Sequences**
  - **Stimulus**: Available ingredients are updated in the inventory.

    Response: The system automatically suggests recipes based on the available ingredients.
  - **Stimulus**: User inputs available ingredients.

    Response: The system suggests recipes based on the ingredients provided and dietary preferences.

- **Stimulus**: New ingredients are added to the inventory.

  Response: The system updates recipe suggestions accordingly.

- **Stimulus**: Ingredients in inventory are used.

  Response: The system updates inventory and adjusts future recipe suggestions.

- **Stimulus**: A recipe is selected by the user.

  Response: The system offers modification options for adjusting the recipe based on the quantity of ingredients on hand.

- **Stimulus**: A recipe modification is needed (based on available ingredients).

  Response: The system automatically adjusts the recipe based on the ingredient quantities available.

- **Stimulus**: Ingredients in inventory are running low.

  Response: The system notifies the user and suggests reordering the missing ingredients.

- **Stimulus**: New ingredients are added to the inventory.

  Response: The system updates recipe suggestions accordingly.

- **Stimulus**: The recipe is not feasible due to missing ingredients.

  Response: The system suggests alternative recipes using available ingredients.

- **Stimulus**: Meal plan is modified or meals are removed.

  Response: The system automatically adjusts the meal plan based on available ingredients.

- **Stimulus**: The user asks to save a recipe.

  Response: The system allows users to save and organize their favorite recipes.

- **Stimulus**: Dietary restrictions or preferences are updated.

  Response: The system adjusts the meal plan and recipe suggestions to align with the new preferences.

- **Stimulus**: The user requests a weekly meal plan.

  Response: The system generates an AI-based meal plan, considering past preferences and available ingredients.

- **Stimulus**: Inventory is reviewed.

  Response: The system suggests new recipes based on the current inventory and tracks any missing ingredients.

- **Functional Requirements**
    - **REQ-1:** The system shall be able to suggest recipes automatically based on the available ingredients and user dietary preferences.
    - **REQ-2:** The system shall be able to allow recipes to be modified based on the quantity of ingredients on hand, adjusting ingredient amounts as needed.
    - **REQ-3:** The system shall be able to allow users to save and organize their favorite recipes for future use.
    - **REQ-4:** The system shall be able to track inventory in real time and update recipe suggestions accordingly.
    - **REQ-5:** The system shall be able to provide alternative recipe suggestions when an ingredient is missing or insufficient.
    - **REQ-6:** The system shall be able to automatically adjust meal plans when ingredients are unavailable, ensuring feasible meals are suggested.
    - **REQ-7:** The system shall be able to suggest recipes based on the user's dietary restrictions and preferences, customizing the meal plan accordingly.
    - **REQ-8:** The system shall be able to generate a weekly meal plan automatically, considering past preferences and available ingredients.
    - **REQ-9:** The system shall be able to provide users with modification options for adjusting recipes according to ingredient quantities on hand.
    - **REQ-10:** The system shall be able to notify users when an ingredient in their inventory is running low and suggest reordering it.
    - **REQ-11 :**The system shall be able to suggest alternative recipes when a selected recipe is not feasible due to missing ingredients.
    - **REQ-12:** The system shall be able to update recipe suggestions when new ingredients are added to the inventory.
    - **REQ-13:** The system shall be able to update recipe suggestions when ingredients in inventory are used.

4. Whole-House Inventory Management

- **Description and Priority**
    - Enable inventory tracking for various household items (e.g., food, cleaning supplies).
    - Send alerts when items are low in stock or expired.
    - Support tagging items with purchase dates and estimated replacement dates.
    - **Priority**: High

- **Stimulus/Response Sequences**
  - **Stimulus**: Inventory is updated (e.g., new items are added, quantities adjusted).

    Response: The system updates the inventory and tracks all changes, ensuring proper tagging of items with purchase dates and estimated replacement dates.
  - **Stimulus**: Item quantity falls below the predefined threshold set by the user (e.g., stock of a specific household item is running low).

    Response: The system sends an automatic alert to the user, notifying them that the item is low in stock and needs to be replenished.
  - **Stimulus**: Item reaches its expiration date (e.g., perishable goods or products with expiration dates).

    Response: The system generates an alert, informing the user that the item has expired and needs to be disposed of or replaced.

  - **Stimulus**: New items are added to the inventory by the user or through an automatic update (e.g., scanning or manual entry).

    Response: The system adds the item to the inventory and assigns a purchase date, along with an estimated replacement or use-by date for efficient tracking.
  - **Stimulus**: Item's replacement date or use-by date approaches (e.g., a cleaning product or non-perishable food item nearing its useful life).

    Response: The system sends a proactive reminder to the user that the item needs to be replaced or disposed of to maintain household efficiency.
  - **Stimulus**: The user manually adjusts item details (e.g., changing the quantity or purchase date).

    Response: The system updates the inventory and ensures all corresponding alerts are recalculated based on the new data.
  - **Stimulus**: Inventory is reviewed (e.g., periodic check by the user).

    Response: The system generates a summary of the current inventory status, highlighting any items that are nearing their expiry or replacement dates and providing a suggestion for reordering.
  - **Stimulus**: The user removes an item from inventory (e.g., product is used or discarded).

Response: The system updates the inventory, adjusts the stock count, and sends notifications if the item is part of a regularly restocked category.

- **Stimulus**: The user requests a detailed inventory report (e.g., a list of all items, categorized by type, status, or purchase date).

  Response: The system generates and displays a comprehensive inventory report, allowing the user to review and organize items based on specific parameters like expiry or quantity.

- **Stimulus**: Item details are automatically updated through integration with third-party systems (e.g., purchase data from grocery store apps).

  Response: The system synchronizes and updates the inventory with the latest item information, ensuring all changes are reflected.

- **Stimulus**: A product is identified as having a shelf life approaching its end (e.g., nearing its last use date).

  Response: The system suggests actions, such as replacing the item or consuming it before the expiration, and updates the alert system accordingly.

- **Stimulus**: The user uploads an image of a receipt.

  Response: The system uses OCR (Optical Character Recognition) to extract item information (e.g., name, quantity, price) from the receipt and automatically updates the inventory with the new items and details.

- **Functional Requirements**
  - **REQ-1:** Users shall be able to manually add items to the inventory through input or by scanning product barcodes.
  - **REQ-2:** The system shall be able to automatically track and organize household items with details such as name, quantity, purchase date, expiration date, and replacement date.
  - **REQ-3:** The system shall be able to use Optical Character Recognition (OCR) to read images of receipts and automatically extract item details (e.g., product names, quantities, and prices) to update the inventory.
  - REQ-**4:** The system shall be able to send notifications or alerts when an item's expiration date is approaching.
  - **REQ-5:** Users shall be able to manually update or remove items from the inventory, such as after use or disposal.
  - **REQ-6:** The system shall be able to automatically update inventory quantities when new purchases are added or when items are removed.
  - **REQ-7:** Users shall be able to set threshold levels for items, and the system shall alert them when the stock falls below these levels.
  - **REQ-8:** The system shall be able to allow users to review the status of their inventory, displaying items that are low in stock, expired, or

approaching expiration.

- **REQ-9:** The system shall be able to support the automatic updating of inventory based on linked third-party APIs for external product stock or pricing data (e.g., grocery stores).
- **REQ-10:** Users shall be able to set automatic replenishment rules for specific items, prompting the system to notify them when it's time to reorder.
- **REQ-11:** The system shall be able to generate a real-time inventory report, summarizing items by category, quantity, expiration status, and more.

5. Family Calendar and Event Scheduler

- **Description and Priority**
  - Create and share family events, reminders, and appointments.
  - Sync with users' calendars for easy accessibility.
  - Send reminders and updates on upcoming events.
  - **Priority**: Medium
- **Stimulus/Response Sequences**
  - **Stimulus**: The user schedules a family event or appointment (e.g., birthday, doctor's appointment).

    Response: The system adds the event to the shared family calendar and assigns a date, time, and location.
  - **Stimulus**: The user invites family members to an event.

    Response: The system sends invitations to the selected family members and adds the event to their personal calendars (if synced).
  - **Stimulus**: User updates or modifies an event (e.g., changing the time, location, or details).

    Response: The system updates the shared family calendar and sends notifications to affected family members about the changes.
  - **Stimulus**: Event time approaches (e.g., a family gathering, doctor's visit).

    Response: The system sends a reminder to all involved family members about the upcoming event.
  - **Stimulus**: The user sets a recurring event (e.g., weekly family meeting, monthly chores).

    Response: The system creates the event and sets the recurrence pattern (e.g., weekly, monthly), automatically adding it to the calendar.

- **Stimulus**: The user marks an event as completed or canceled.

  Response: The system updates the family calendar to reflect the completion or cancellation of the event and sends notifications to family members.

- **Stimulus**: The user checks the family calendar for upcoming events.

  Response: The system displays a list of upcoming family events, highlighting any changes, upcoming reminders, or overlapping events.

- **Stimulus**: The event's scheduled time arrives.

  Response: The system sends an alert or reminder to family members involved in the event.

- **Stimulus**: User adds a note or attachment (e.g., directions, agenda) to an event.

  Response: The system stores and displays the attached note or file alongside the event on the family calendar.

- **Stimulus**: The user syncs personal calendar with the family calendar.

  Response: The system synchronizes events from the personal calendar with the family calendar, ensuring consistency across both calendars.

- **Functional Requirements**
  - **REQ-1:** The users should create, modify, and delete family events (e.g., birthdays, appointments).
  - **REQ-2:** The system should allow users to invite family members to events and send event invitations.
  - **REQ-3:** The system should support syncing the family calendar with users' personal calendars for easy access and consistency.
  - **REQ-4:** The users should set recurring events, such as weekly or monthly meetings, appointments, or reminders.
  - **REQ-5:** The system should send reminders to family members prior to an event, based on customizable settings (e.g., one day before, one hour before).
  - **REQ-6:** The system should allow users to attach notes or files (e.g., event agendas, directions) to events.
  - **REQ-7:** The system should update the family calendar in real-time to reflect changes in event details (e.g., time, location).
  - **REQ-8:** The users should mark events as completed or canceled, and the system will notify affected family members.
  - **REQ-9:** The system should display upcoming events, highlighting any changes, reminders, or conflicts with other events.
  - **REQ-10:** The system should allow users to add additional details to

events, such as locations, descriptions, or special instructions.

- **REQ-11:** The system should provide an overview of all family events, with filtering options for daily, weekly, or monthly views.
- **REQ-12:** The system should automatically adjust event notifications based on user preferences (e.g., frequency, timing).
- **REQ-13:** The system should ensure secure access to the calendar, allowing only authorized family members to view or modify events.

6. Grocery Price Comparison Tool

- **Description and Priority**
  - Compare prices of grocery items across multiple stores and suggest the cheapest options.
  - Allow users to set up a preferred store and prioritize items based on price or proximity.
  - Support integration with store apps for live price updates (if available).
  - **Priority**: Medium.
- **Stimulus/Response Sequences**
  - **Stimulus**: The user inputs a grocery list.

    Response: The system compares prices across stores and suggests the cheapest options.
  - **Stimulus**: Preferred store is selected.

    Response: The system prioritizes items from the selected store in recommendations.
  - **Stimulus**: A price drop is detected for a frequently purchased item.

    Response: The system sends an alert to the user with updated pricing information.
  - **Stimulus**: The user enables bulk-buy suggestions.

    Response: The system highlights cost-saving deals and bulk discounts.
  - **Stimulus**: The user sets a budget for grocery shopping.

    Response: The system recommends the best selection of items within the specified budget.
  - **Stimulus**: The user enables historical price tracking.

Response: The system provides price trends and suggests the best time to purchase.

- **Stimulus**: A store updates its price list through integration.

  Response: The system synchronizes and updates the price comparison data in real time.

- **Stimulus**: The user selects an item and requests alternative store options.

  Response: The system lists stores where the item is available, ranked by price and location.

- **Stimulus**: Store promotions or discounts are detected.

  Response: The system notifies users of available deals based on their grocery list.

- **Functional Requirements**
  - **REQ-1:** The system should allow users to compare grocery prices across multiple stores.
  - **REQ-2:** The system should enable users to set a preferred store and prioritize items accordingly.
  - **REQ-3:** The system should support integration with store apps for real-time price updates.
  - **REQ-4:** The system should notify users when a price drop occurs for frequently purchased items.
  - **REQ-5:** The system should provide cost-saving recommendations, including bulk-buy options.
  - **REQ-6:** The system should allow users to set a grocery shopping budget and suggest items accordingly.
  - **REQ-7:** The system should track historical price trends and suggest the best time to buy items.
  - **REQ-8:** The system should update price comparison data when a store updates its price list.
  - **REQ-9:** The system should allow users to request alternative store options for selected items.
  - **REQ-10:** The system should notify users of available store promotions and discounts.

7. Emergency Notifier

- **Description and Priority**
  - Provide a quick alert feature for family members to notify others in case of emergencies.
  - Send instant notifications with the sender's location and message.

- **Priority**: High

- **Stimulus/Response Sequences**
  - **Stimulus**: The user triggers an emergency alert.

    Response: The system notifies all family members of the sender's location and emergency message.
  - **Stimulus**: The user selects “Medical Emergency.”

    Response: The system shares critical medical information with emergency contacts.
  - **Stimulus**: The user enables an SOS shortcut (e.g., double-pressing power button).

    Response: The system sends an emergency alert instantly.
  - **Stimulus**: Emergency contact acknowledges the alert.

    Response: The system updates other family members with status confirmation.
  - **Stimulus:** The user configures emergency contacts and alert preferences.

    Response: The system saves and applies settings for faster emergency response.

- **Functional Requirements**
  - **REQ-1:** The system shall be able to allow users to trigger an emergency alert instantly.
  - **REQ-2:** The system shall be able to include the sender's location and emergency message in notifications.
  - **REQ-3:** The system shall be able to allow users to select predefined emergency types (e.g., medical, fire, security).
  - **REQ-4:** The system shall be able to provide an SOS shortcut (e.g., double-pressing the power button) for quick activation.
  - **REQ-5:** The system shall be able to allow emergency contacts to acknowledge alerts and update family members.
  - **REQ-6**: The system shall be able to allow users to configure emergency contacts and alert preferences.

8. Home Workout Planner

- **Description and Priority**
  - Suggest home workout routines based on user fitness levels and goals.
  - Track exercise completion and offer personalized progression tips.
  - Allow users to schedule workout sessions and receive reminders.

- **Priority**: Medium

- **Stimulus/Response Sequences**
  - **Stimulus**: The user inputs fitness goals (e.g., weight loss, muscle gain, flexibility).

    Response: The system suggests personalized workout routines.
  - **Stimulus**: The user selects available workout equipment (e.g., dumbbells, resistance bands).

    Response: The system customizes workout plans based on available equipment.
  - **Stimulus**: Workout session is completed.

    Response: The system tracks progress, updates history, and offers personalized tips.
  - **Stimulus**: The user schedules a workout session.

    Response: The system sets reminders and sends notifications before the session starts.
  - **Stimulus**: The user skips or modifies a planned workout.

    Response: The system adjusts the workout schedule and progression plan accordingly.
  - **Stimulus**: The user requests a difficulty adjustment (e.g., easier, harder).

    Response: The system recalibrates workout intensity and progression.

- **Functional Requirements**
  - **REQ-1:** The system should suggest workouts based on user fitness goals and available equipment.
  - **REQ-2:** The system should track progress and provide personalized tips for improvement.
  - **REQ-3:** The system should allow users to schedule workout sessions and receive reminders.
  - **REQ-4:** The system should enable users to modify workout intensity based on feedback.
  - **REQ-5:** The system should provide visual and audio guidance for each workout routine.
  - **REQ-6:** The system should track skipped workouts and suggest catch-up sessions

9. **Child Activity Tracker**

- **Description and Priority**
  - Track children's activities, progress, and milestones.
  - Set daily activity goals, including educational and recreational tasks.
  - Notify family members about their children's progress.
  - **Priority**: Medium
- **Stimulus/Response Sequences**
  - **Stimulus**: The user sets activity goals for a child (e.g., reading time, outdoor play).

    Response: The system tracks and monitors progress.
  - **Stimulus**: The child completes an activity.

    Response: The system records completion, updates progress, and awards achievements.
  - **Stimulus**: Activity completion threshold is reached (e.g., 5 completed reading sessions).

    Response: The system sends notifications and provides progress summaries.
  - **Stimulus**: Parent or guardian requests a report.

    Response: The system generates a summary of completed activities, milestones, and trends.
  - **Stimulus**: The user schedules a recurring activity (e.g., daily bedtime story).

    Response: The system sets reminders for the activity.
  - **Stimulus**: The system detects inactivity or missed goals.

    Response: The system sends alerts to parents and suggests alternative activities.

- **Functional Requirements**
  - **REQ-1:** The system should allow users to set and track activity goals for children.
  - **REQ-2:** The system should record activity completion and track progress over time.
  - **REQ-3:** The system should notify family members about children's progress and milestones.
  - **REQ-4:** The system should generate detailed reports on children's

activity engagement.

- **REQ-5:** The system should suggest alternative activities if goals are not met.
- **REQ-6:** The system should provide achievement badges or rewards for completed activities.

10. Feedback and Ratings

- **Description and Priority**
  - Enable users to rate recipes and leave feedback.
  - Display average ratings and reviews for each recipe.
  - Allow users to sort recipes based on ratings and popularity.
  - **Priority**: Medium
- **Stimulus/Response Sequences**
  - **Stimulus**: The user rates a recipe.

    Response: The system updates the recipe's average rating.
  - **Stimulus**: The user leaves feedback on a recipe.

    Response: The system stores the review and displays it.
  - **Stimulus**: The user submits general feedback about the app.

    Response: The system stores the feedback and notifies the development team.
  - **Stimulus**: The user rates the app overall (e.g., 1-5 stars).

    Response: The system updates the app's average rating.
  - **Stimulus**: The user submits an issue report or bug feedback.

    Response: The system logs the issue and notifies the support team.
  - **Stimulus**: The user submits feedback on app features or usability.

    Response: The system collects feedback for review and future improvements.
  - **Stimulus**: The user accesses the app’s rating page.

    Response: The system displays the app’s overall rating and user feedback.
  - **Stimulus**: The user requests to leave feedback for the app.

    Response: The system prompts the user to provide a rating and additional comments.

- **Stimulus**: The user selects a feature to rate within the app (e.g., Recipe Suggestions, Inventory Management).

  Response: The system prompts the user to rate that specific feature and provide feedback.

- **Stimulus**: The user views the app's rating trends.

  Response: The system shows historical rating trends (e.g., increase or decrease in ratings over time).

- **Stimulus**: The user requests to view feedback from other users about the app.

  Response: The system displays the average rating, recent reviews, and comments from other users.

- **Functional Requirements**
  - **REQ-1:** Users should rate recipes on a scale (e.g., 1-5 stars).
  - **REQ-2:** Users should leave written reviews for recipes.
  - **REQ-3:** Recipe ratings should be displayed to users, showing an average rating and individual reviews.
  - **REQ-4:** Users should submit general feedback about the app's overall performance and features.
  - **REQ-5:** The app should collect and store feedback on issues, bugs, or feature requests from users.
  - **REQ-6:** App ratings should be displayed publicly, showing an average rating based on user submissions.
  - **REQ-7:** User feedback should be categorized (e.g., general feedback, app features, bug reports) for better organization.
  - **REQ-8:** The app should allow users to rate specific features (e.g., Recipe Suggestions, Inventory Management).
  - **REQ-9:** The system should automatically notify the development or support team when an issue is reported via user feedback.
  - **REQ-10:** The app should display historical trends of its ratings (e.g., changes in average ratings over time).
  - **REQ-11:** The app should allow users to view feedback and ratings submitted by other users.

11. Health Information Display

- **Description and Priority**
  - Provide detailed nutritional information for each recipe, including calories, macronutrients, and potential allergens.
  - Display daily/weekly nutritional summaries to help users track their health goals.

- Allow users to filter recipes based on dietary restrictions or health goals.
- **Priority**: Medium

- **Stimulus/Response Sequences**
  - **Stimulus**: The user selects a recipe

    Response: The system displays nutritional information.
  - **Stimulus**: The user filters recipes based on dietary restrictions

    Response: The system suggests recipes accordingly.
  - **Stimulus**: The user selects a recipe

    Response: The system displays detailed nutritional information for the recipe, including calories, macronutrients, and allergens.
  - **Stimulus**: The user filters recipes based on dietary restrictions

    Response: The system suggests recipes that meet the dietary restrictions.
  - **Stimulus**: The user requests daily nutritional summary

    Response: The system displays a summary of the user's daily nutritional intake, including calories and macronutrients.
  - **Stimulus**: The user requests weekly nutritional summary

    Response: The system displays a summary of the user's weekly nutritional intake, including calories and macronutrients.
  - **Stimulus**: The user adjusts health goal (e.g., calorie target or macronutrient distribution)

    Response: The system updates the suggestions based on the new health goal.
  - **Stimulus**: The user reviews a recipe's nutritional information

    Response: The system highlights potential allergens and provides guidance based on the user's health goals.
  - **Stimulus**: The user adds a recipe to their meal plan

    Response: The system updates the nutritional summary based on the selected recipe.
  - **Stimulus**: The user changes dietary preference (e.g., low-carb, gluten-free)

    Response: The system updates recipe suggestions and nutritional summaries according to the new preferences.
  - **Stimulus**: The user logs a recipe as eaten

Response: The system updates the user's daily nutritional intake.

- **Stimulus**: The user marks a food item as a favorite

  Response: The system tracks and includes the item in the user's daily nutritional summary.

- **Functional Requirements**
  - **REQ-1:** Nutritional information should be provided for each recipe, including calories, macronutrients, and potential allergens.
  - **REQ-2:** The user should view daily/weekly nutritional summaries, including total calories and macronutrient breakdown.
  - **REQ-3:** Recipes should be filtered based on dietary restrictions or health goals (e.g., low-carb, gluten-free, high-protein).
  - **REQ-4:** The system should automatically update nutritional information when a recipe is added to the meal plan.
  - **REQ-5:** The user should adjust their health goals (e.g., calorie target, macronutrient distribution), and the system will update the recipe suggestions accordingly.
  - **REQ-6:** Nutritional information for each recipe should be displayed in an easy-to-read format.
  - **REQ-7:** The system should highlight potential allergens based on user health information or dietary preferences.
  - **REQ-8:** User's health goals (e.g., weight loss, muscle gain) should influence recipe suggestions and nutritional summaries.
  - **REQ-9:** The user should track their nutritional intake by logging meals and viewing updated summaries.
  - **REQ-10:** The system should allow users to save and refer to recipes with specific health benefits (e.g., heart-healthy, low-sugar).
  - **REQ-11:** The system should enable users to track progress towards their health goals, updating based on recipe consumption.

12. Shopping List Creation

- **Description and Priority**
  - Help users create and manage shopping lists based on meal plans and pantry inventory.
  - Enable users to add custom items to the shopping list.
  - Synchronize shopping lists with available inventory and recipe requirements.
  - **Priority:** High
- **Stimulus/Response Sequences**
  - **Stimulus**: The user creates a meal plan

Response: The system generates a shopping list.

- **Stimulus**: The user adds a custom item

  Response: The item is added to the shopping list.

- **Stimulus**: The user adds a custom item to the shopping list.

  Response: An item is added to the shopping list, and it can be categorized by type (e.g., produce, dairy).

- **Stimulus**: The system identifies an ingredient already in the pantry inventory.

  Response: The system removes that item from the shopping list and notifies the user.

- **Stimulus**: The user adjusts the quantity of an item on the shopping list (e.g., increase number of tomatoes).

  Response: The system updates the shopping list and adjusts the total quantity needed.

- **Stimulus**: The user selects an item to purchase from the shopping list.

  Response: The system marks the item as purchased and updates the list accordingly.

- **Stimulus**: The user requests a shopping list for a specific recipe.

  Response: The system generates a shopping list that includes only the ingredients for that recipe.

- **Stimulus**: The user selects an item on the shopping list and links it to a specific store.

  Response: The system suggests the best store for purchasing the item based on price or proximity.

- **Stimulus**: The user's pantry inventory is updated (e.g., item used up or added).

  Response: The system automatically synchronizes the shopping list with the updated inventory to avoid duplicating items.

- **Functional Requirements**
  - **REQ-1:** The user shall generate shopping lists based on meal plans.

- **REQ-2:** The user shall add custom items to shopping lists.
- **REQ-3:** The Shopping lists shall be synchronized with inventory and recipe requirements.
- **REQ-4:** The user shall be able to add custom items to shopping lists.
- **REQ-5:** The Shopping lists shall be synchronized with inventory and recipe requirements to avoid duplicate items.
- **REQ-6:** The system shall allow users to adjust the quantity of items on the shopping list.
- **REQ-7:** The user shall be able to mark items as purchased, and the list is updated accordingly.
- **REQ-8:** The Shopping lists shall be organized by categories (e.g., fruits, dairy).
- **REQ-9:** The user shall view a shopping list specific to a selected recipe.
- **REQ-10:** The system shall suggest the best store to purchase items based on price or proximity.
- **REQ-11:** Pantry inventory shall update automatically and synchronize with the shopping list, removing items already available.

3.5. Nonfunctional Requirements

1. **Performance & security Requirements**

- **Requirement**: Ensure app responsiveness under load while maintaining secure data handling.
- **Measurement**:
  - System response time: ≤ 3 seconds under normal load.
  - Support for concurrent users: 1000.
  - Stress testing tools: JMeter or LoadRunner.
  - **Data encryption:** Ensure that sensitive data, such as user credentials and personal information, is encrypted during transmission (e.g., using HTTPS/TLS).
  - **Authentication:** Ensure that authentication mechanisms (e.g., multi-factor authentication) are not compromised under load and that session management is secure.
  - **DDoS Protection:** The system should implement protection