# COMP3030J Software Engineering Project 2 2024-2025

Beijing-Dublin International College

## Project Proposal - Group 14 SKT

**Group Members:**

Haoxiang Han<br>
Jiawei Xing<br>
Ruiyang Liu<br>
Shaojun Wang<br>
Xiaopeng Zhang<br>
Xuning Wei

**Professor:**

Catherine Mooney

# 1 Introduction of our company

SKT Inc. is a premier software engineering company dedicated to delivering cutting-edge solutions that drive digital transformation while promoting health and well-being (SDG 3). Our mission is to support sustainable well-being through innovative technology and tailored software solutions. With over 8 years of experience in software development, we have successfully delivered 35+ projects across diverse industries. Our dynamic and experienced team collaborates to design and implement high-quality solutions that align with our clients' business goals while fostering a healthier and more sustainable future.

# 2 Introduction of client company

Harvest AgriFoods Innovations Ltd. has evolved from a supplier of basic agricultural products into a specialized provider of processed foods, driven by the need to adapt to an increasingly competitive market. The company is committed to delivering high-quality, healthy, and delicious foods, catering to the growing demand for nutritious and flavorful options. As part of its growth strategy, the company is seeking an innovative digital solution that highlights its product advantages while streamlining the online purchasing process.

# 3 Customer problem statement

1. **Understanding of Healthy Food by Users**: Despite their desire to follow a healthy diet, many people are reluctant to buy health food online since they do not know enough about it. However, there isn't a website that assists customers with customized meal planning. Incorporating the company's products into customers' everyday diets is also necessary. Therefore, a tool that guides customers in individualized meal pairings is needed.

2. **Inventory Management System for the Business**: To track goods and raw materials in real time, an inventory management system is required. Managing inventory allocations and rapidly updating stock changes should be made easier for administrators via the website. Additionally, it need to facilitate the importation of new goods and raw materials while keeping track of the amount connections between them for simpler administration. Customers should be shown alternatives when a product is unavailable.

3. **Social Media Development**: Mainstream social media platforms are flooded with advertisements. To ensure authenticity and high-quality products, the company needs a clean, dedicated social media platform that aligns with the sustainability and well-being ethos. This platform should allow users to review and rate products, and easily search for the company's products. The website's style should closely align with the SDG 3 theme, categorizing posts according to different topics.

# 4 Solution

With the rapid expansion of its business, Harvest AgriFoods Innovations Ltd. has identified a critical need to establish a robust online sales and marketing platform. This platform will not only showcase the company's product offerings effectively but also enhance customer engagement and enable them to reach a broader target audience.

**Function Module 1 - Optimal Nutrition Ratio**

1. **User Profile Management**: Users enter personal details and set health goals (e.g., fat loss, muscle gain), with profiles securely stored in SQLite for future customization.

2. **Nutrition Calculator Engine**: Computes Basal Metabolic Rate and daily energy needs, adjusting macronutrient and micronutrient distribution based on user goals and lifestyle.

3. **AI-Driven Food Recommendation**: Uses AI to match nutritional needs with available products, prioritizing company offerings and refining suggestions based on feedback.

4. **Interactive Dashboard**: Provides real-time nutritional summaries, progress tracking, and visual analytics to help users monitor and optimize their diet. The user interface will be designed using HTML, CSS combined with Bootstrap.

**Function Module 2 - Inventory Management & Product Synthesis**

1. **Raw Material Tracking**: The system manages raw material information through Flask. It calculates required material quantities based on requirements to generate finished products. All data is stored in an SQLite database supporting rapid retrieval and updates.

2. **Real-time Inventory Management**: The platform automatically adjusts stock levels after production or sales. Flask interacts with the front-end to display accurate inventory status. The system flags out-of-stock products and recommends similar alternatives.

3. **Automatic Inventory Updates**: Upon order placement, the system automatically deducts inventory and updates stock status. Post-order fulfillment, production plans are dynamically adjusted to align with inventory demands.

4. **Expired Product Handling**: The system calculates and monitors product expiration dates. When products approach expiration, the platform alerts administrators for disposal. Expired items are automatically removed from sales listings.

**Function Module 3 - Forum**

1. **Discussion Threads**: Organize discussions by creating categories (e.g., Healthy Eating, Fitness, Lifestyle Tips). Users can start new threads or reply to existing ones, keeping topics structured and easy to navigate.

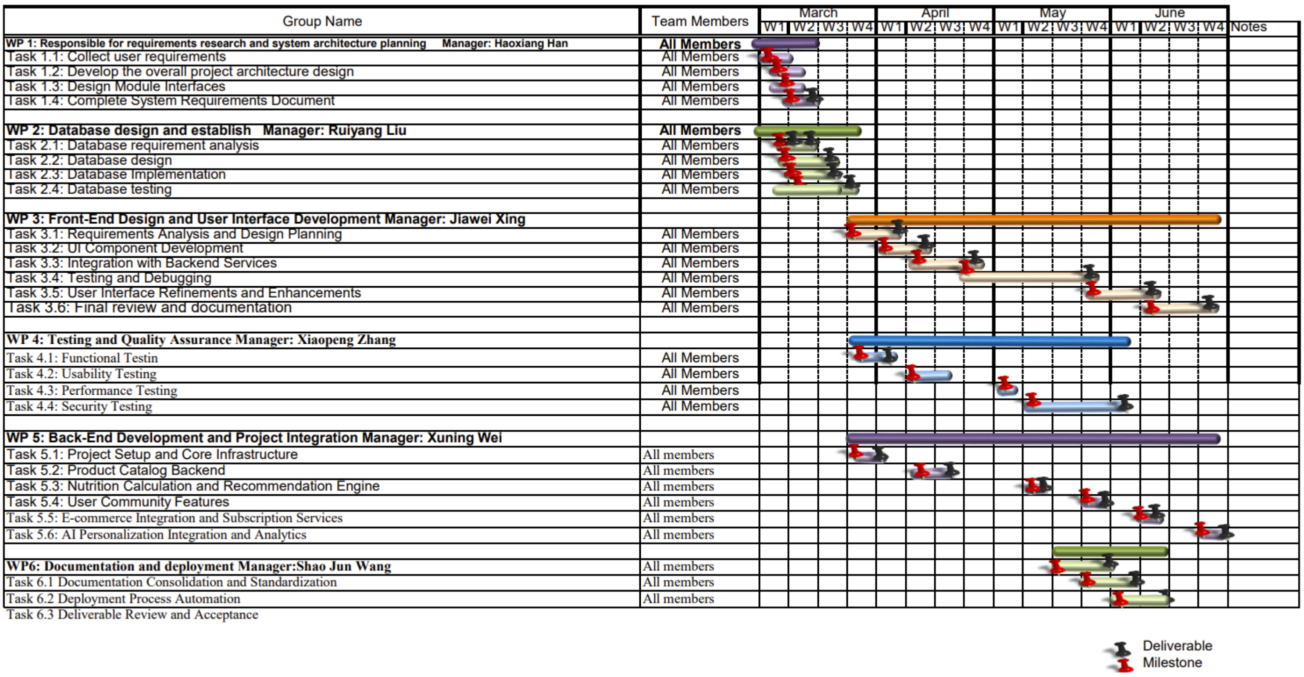
2. **Comment System**: Enable rich text posts with multimedia support. Users can comment on threads, reply to specific posts, and edit their contributions.

3. **Search and Navigation**: Implement a robust search engine within the forum for keywords and tags stored in SQLite database. Tagging posts by topic and popularity helps users quickly locate relevant discussions and content.

4. **Notification**: Provide real-time notifications for replies, mentions, and updates on followed threads. Social sharing options allow users to spread interesting discussions beyond the forum.

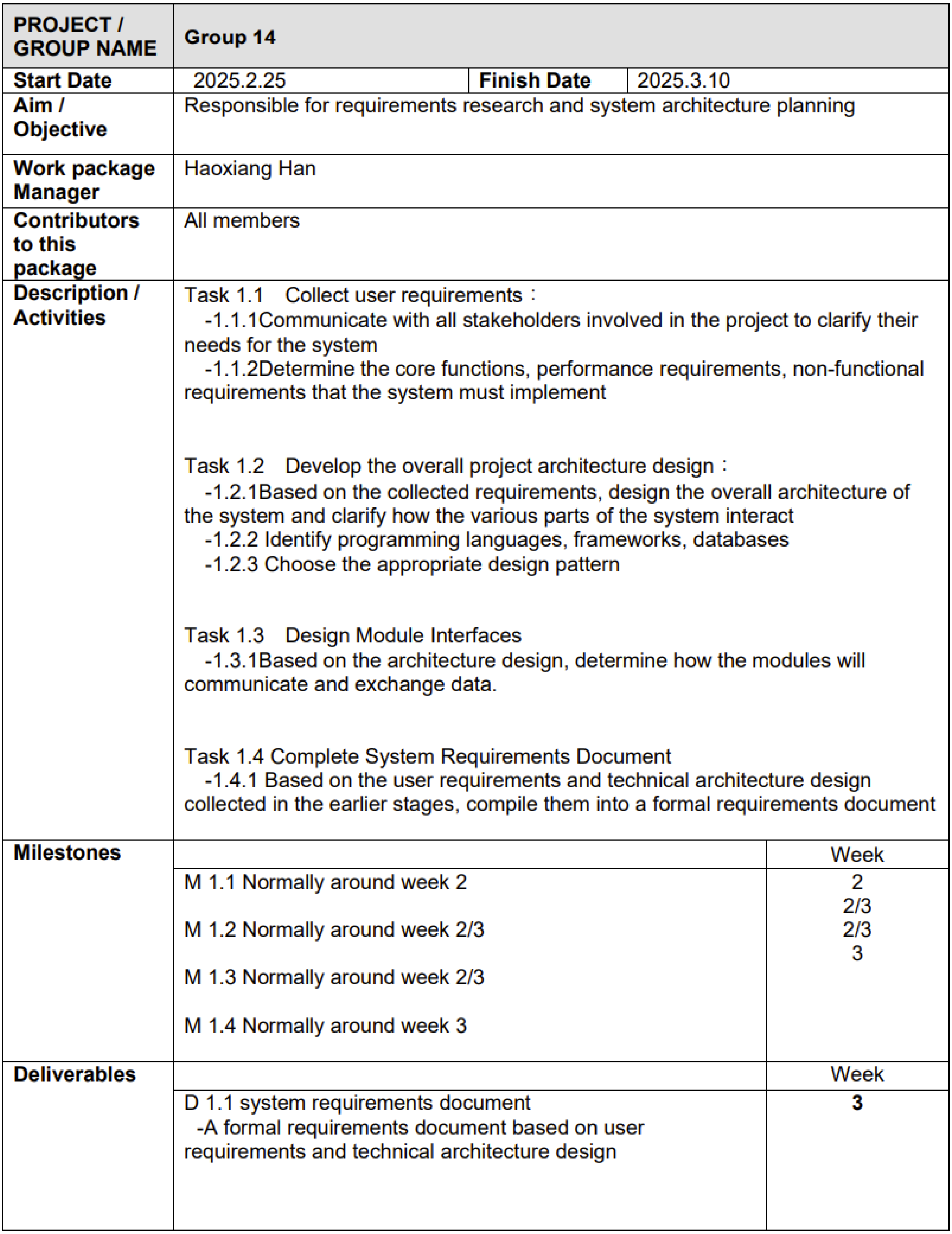
# 5 Appendix

## A. Gantt Chart



| Group Name | Team Members |
|---|---|
| **WP 1: Responsible for requirements research and system architecture planning Manager: Haoxiang Han** | **All Members** |
| Task 1.1: Collect user requirements | All Members |
| Task 1.2: Develop the overall project architecture design | All Members |
| Task 1.3: Design Module Interfaces | All Members |
| Task 1.4: Complete System Requirements Document | All Members |
| | |
| **WP 2: Database design and establish Manager: Ruiyang Liu** | **All Members** |
| Task 2.1: Database requirement analysis | All Members |
| Task 2.2: Database design | All Members |
| Task 2.3: Database Implementation | All Members |
| Task 2.4: Database testing | All Members |
| | |
| **WP 3: Front-End Design and User Interface Development Manager: Jiawei Xing** | |
| Task 3.1: Requirements Analysis and Design Planning | All Members |
| Task 3.2: UI Component Development | All Members |
| Task 3.3: Integration with Backend Services | All Members |
| Task 3.4: Testing and Debugging | All Members |
| Task 3.5: User Interface Refinements and Enhancements | All Members |
| Task 3.6: Final review and documentation | All Members |
| | |
| **WP 4: Testing and Quality Assurance Manager: Xiaopeng Zhang** | |
| Task 4.1: Functional Testin | All Members |
| Task 4.2: Usability Testing | All Members |
| Task 4.3: Performance Testing | All Members |
| Task 4.4: Security Testing | All Members |
| | |
| **WP 5: Back-End Development and Project Integration Manager: Xuning Wei** | |
| Task 5.1: Project Setup and Core Infrastructure | All members |
| Task 5.2: Product Catalog Backend | All members |
| Task 5.3: Nutrition Calculation and Recommendation Engine | All members |
| Task 5.4: User Community Features | All members |
| Task 5.5: E-commerce Integration and Subscription Services | All members |
| Task 5.6: AI Personalization Integration and Analytics | All members |
| | |
| **WP6: Documentation and deployment Manager:Shao Jun Wang** | All members |
| Task 6.1 Documentation Consolidation and Standardization | All members |
| Task 6.2 Deployment Process Automation | All members |
| Task 6.3 Deliverable Review and Acceptance | |

# B. Work Packages



| | | | |
|---|---|---|---|
| **PROJECT / GROUP NAME** | **Group 14** | | |
| **Start Date** | 2025.2.25 | **Finish Date** | 2025.3.10 |
| **Aim / Objective** | Responsible for requirements research and system architecture planning | | |
| **Work package Manager** | Haoxiang Han | | |
| **Contributors to this package** | All members | | |
| **Description / Activities** | Task 1.1 Collect user requirements：<br>-1.1.1Communicate with all stakeholders involved in the project to clarify their needs for the system<br>-1.1.2Determine the core functions, performance requirements, non-functional requirements that the system must implement<br>Task 1.2 Develop the overall project architecture design：<br>-1.2.1Based on the collected requirements, design the overall architecture of the system and clarify how the various parts of the system interact<br>-1.2.2 Identify programming languages, frameworks, databases<br>-1.2.3 Choose the appropriate design pattern<br>Task 1.3 Design Module Interfaces<br>-1.3.1Based on the architecture design, determine how the modules will communicate and exchange data.<br>Task 1.4 Complete System Requirements Document<br>-1.4.1 Based on the user requirements and technical architecture design collected in the earlier stages, compile them into a formal requirements document | | |
| **Milestones** | | Week | |
| | M 1.1 Normally around week 2 | 2 | |
| | M 1.2 Normally around week 2/3 | 2/3 | |
| | M 1.3 Normally around week 2/3 | 2/3 | |
| | M 1.4 Normally around week 3 | 3 | |
| **Deliverables** | | Week | |
| | D 1.1 system requirements document<br>-A formal requirements document based on user requirements and technical architecture design | **3** | |

| | | | |
|---|---|---|---|
| **PROJECT / GROUP NAME** | **Group 14** | | |
| **Start Date** | 25/2/2025 | **Finish Date** | 10/6/2025 |
| **Aim / Objective** | Database design and establish | | |
| **Work package Manager** | Ruiyang Liu | | |
| **Contributors to this package** | All members | | |
| **Description / Activities** | Task 2.1 Database requirement analysis<br>-2.1.1 Review exist documents, understand the database requirement.<br>-2.1.2 Have a team meeting to gather detailed database requirement.<br>-2.1.3 Determine the entity and the attribute of the database.<br>-2.1.4 Define the quantitative relation between entities.<br><br>Task 2.2 Database design<br>-2.2.1 Draw the E-R diagram<br>-2.2.2 Define appropriate data types, constraints (such as primary keys, foreign keys, and unique constraints) for each column in the database tables.<br><br>Task 2.3 Database Implementation<br>-2.3.1 Choose the correct database management system base on the database requirement.<br>-2.3.2 Create the database (entity, attribute,relationship).<br>-2.3.3 Add some date to test.<br><br>Task 2.4 Database testing<br>-2.4.1 Conduct unit testing on individual database objects<br>-2.4.2 Perform integration testing to verify the interaction between different database components | | |
| **Milestones** | | | Week |
| | M 2.1 & M 2.2 Normally around week 1<br>M 2.3 & M 2.4 Normally around week 2/3 | | 1<br>2/3 |
| **Deliverables** | | | Week |
| | D2.1 Database Requirement Specification Document<br>D2.2 ER Diagram<br>D2.3 SQL Scripts for Database Creation and Sample Data Insertion<br>D2.4 Database Testing Report | | D2.1 Database Requirement Specification Document<br>D2.2 ER Diagram<br>D2.3 SQL Scripts for Database Creation and Sample Data Insertion<br>D2.4 Database Testing Report |

| | | | |
|---|---|---|---|
| **PROJECT / GROUP NAME** | **Group 14** | | |
| **Start Date** | 2025.6.11 | **Finish Date** | 2025.7.10 |
| **Aim / Objective** | Documentation and deployment | | |
| **Work package Manager** | Shao Jun Wang | | |
| **Contributors to this package** | All members | | |
| **Description / Activities** | **Task 6.1 Documentation Consolidation and Standardization**<br>• **6.1.1 Document Collection and Alignment**: Aggregate deliverables from frontend, backend, database, security, and testing modules, ensuring consistency in technical terminology and version numbers.<br>○ **6.1.1.1** Verify API documentation matches backend code interfaces (e.g., Swagger vs. actual routes).<br>○ **6.1.1.2** Check consistency between database ER diagrams and SQL script table structures.<br>• **6.1.2 Format Standardization**: Refactor all documentation using Markdown, integrate into a Confluence knowledge base, and generate PDF archival versions.<br>○ **6.1.2.1** Define document templates (heading hierarchy, code block formatting, diagram labeling rules).<br>○ **6.1.2.2** Automatically generate table of contents and version changelogs (e.g., via GitHub Actions).<br>• **6.1.3 Overview Documentation Authoring**: Create system architecture diagrams and technology stack descriptions, highlighting module dependencies.<br>**Task 6.2 Deployment Process Automation**<br>• **6.2.1 Deployment Script Development**: Write Ansible Playbooks and Terraform configurations for one-click deployment across environments (dev/test/prod).<br>○ **6.2.1.1** Automate frontend static resource deployment to CDN (e.g., AWS S3 + CloudFront).<br>○ **6.2.1.2** Configure database initialization scripts (including table creation and baseline data insertion).<br>• **6.2.2 CI/CD Integration**: Configure Jenkins/GitLab CI pipelines to trigger documentation generation and version releases.<br>○ **6.2.2.1** Integrate Swagger UI for automatic API documentation updates.<br>○ **6.2.2.2** Containerized deployment: Build Docker images and push to a private registry (e.g., Harbor).<br>• **6.2.3 Rollback Mechanism Design**: Develop a 5-minute rollback plan using Docker image versions and database backups.<br>**Task 6.3 Deliverable Review and Acceptance**<br>• **6.3.1 Internal Audit**: Conduct cross-team meetings to validate consistency between documentation, code, and deployment configurations.<br>○ **6.3.1.1** Verify API document fields match backend models (e.g., data types, mandatory fields). | | |

| | | | |
|---|---|---|---|
| **PROJECT / GROUP NAME** | **Group 14 / SKT** | | |
| **Start Date** | 2025.3.10 | **Finish Date** | 2025.5.5 |
| **Aim / Objective** | Back-end development and project integration | | |
| **Work package Manager** | Xuning Wei | | |
| **Contributors to this package** | All members | | |
| **Description / Activities** | **Task 5.1: Project Setup and Core Infrastructure**<br>**Task 5.2: Product Catalog Backend**<br>5.2.1: Implement API endpoints for retrieving product information (name, description, nutritional information, images, etc.).<br>5.2.2: Develop the functionality for displaying product categories and subcategories.<br>5.2.3: Implement search functionality for the product catalog.<br>5.2.4: Develop API endpoints for handling user-uploaded food information (validation, storage). Consider moderation workflows.<br>**Task 5.3: Nutrition Calculation and Recommendation Engine**<br>5.3.1: Implement the nutrition calculation logic based on user input (age, height, weight, goals).<br>5.3.2: Design and implement algorithms for calculating required nutrient intake based on user goals (weight loss, muscle gain, general health).<br>5.3.3: Develop the recommendation engine to suggest suitable food items (prioritizing the company's products) based on the calculated nutrient needs.<br>5.3.4: API integration with product catalog to retrieve nutritional data.<br>**Task 5.4: User Community Features**<br>5.4.1: Implement API endpoints for posting, retrieving, updating, and deleting user posts.<br>5.4.2: Implement user commenting and rating functionalities.<br>5.4.3: Develop forum and discussion board functionalities.<br>5.4.4: Implement features to moderate content and flag inappropriate posts.<br>5.4.5: Implement reporting mechanisms for flagging inappropriate content<br>**Task 5.5: E-commerce Integration and Subscription Services**<br>5.5.1: Integrate with a payment gateway (e.g., Stripe, PayPal).<br>5.5.2: Implement order processing and management functionalities.<br>5.5.3: Develop the subscription service functionality (recurring billing, delivery scheduling).<br>5.5.4: Create APIs for managing user orders and subscriptions.<br>5.5.5: Implement security measures to protect payment information.<br>**Task 5.6: AI Personalization Integration and Analytics**<br>5.6.1: Design and implement the health profile data storage and retrieval system.<br>5.6.2: Integrate the AI/ML algorithms for personalized food recommendations.<br>5.6.3: Develop the smart shopping list generation functionality.<br>5.6.4: Implement data analytics tracking to monitor platform usage and identify areas for improvement. | | |
| **Milestones** | | | Week |
| | **M 5.1: Product Catalog MVP** | | **4/5** |
| | **M 5.2: Nutrition Calculation Engine Ready** | | **6/7** |
| | **M 5.3: Interim Report & Functional Core** | | **9** |

| | | | |
|---|---|---|---|
| **PROJECT / GROUP NAME** | **Group 14** | | |
| **Start Date** | 25/2/2025 | **Finish Date** | 10/6/2025 |
| **Aim / Objective** | Responsible for front-end design and user interface. | | |
| **Work package Manager** | Jiawei Xing | | |
| **Contributors to this package** | All members | | |
| **Description / Activities** | **Task 3.1: Requirements Analysis and Design Planning**<br>• Gather and analyze the user requirements for the front end and user interface (UI).<br>• Research design patterns and technologies (e.g., HTML, CSS, JavaScript).<br>• Create initial wireframes and design prototypes for the application.<br>• Define the layout and navigation flow of the application.<br>**Task 3.2: UI Component Development**<br>• Develop the main UI components: login page, dashboard, navigation bar, and content display areas.<br>• Implement responsive design for various screen sizes (desktop, tablet, mobile).<br>• Ensure user-friendly and intuitive navigation.<br>• Start integrating visual elements like icons, buttons, and text inputs.<br>**Task 3.3: Integration with Backend Services**<br>• Work with backend developers to integrate frontend UI with API endpoints.<br>• Implement frontend validation for user input.<br>• Ensure real-time data fetching and interaction between UI components and the backend.<br>**Task 3.4: Testing and Debugging**<br>• Perform unit testing for individual UI components.<br>• Conduct usability testing with the team to ensure a smooth user experience.<br>• Debug any issues related to UI rendering, layout, or interactions.<br>• Optimize the UI for performance (e.g., fast loading times, smooth animations).<br>**Task 3.5: User Interface Refinements and Enhancements**<br>• Refine UI based on feedback from internal testing.<br>• Improve visual elements, animations, and transitions to enhance user experience.<br>• Ensure accessibility (color contrast, screen reader support, etc.).<br>• Implement final adjustments for the look and feel of the interface.<br>**Task 3.6: Final Review and Documentation**<br>• Finalize all front-end design features.<br>• Review and document the entire front-end design and development process.<br>• Prepare for integration with the overall project and provide the final deliverables. | | |
| **Milestones** | | Week | |
| | • **M 3.1 (Week 4):** Completion of requirements gathering, wireframes, and design prototypes. | **4** | |
| | • **M 3.2 (Week 6):** First iteration of UI components (login, dashboard, navigation) completed. | **6** | |
| | • **M 3.3 (Week 9):** Integration with backend services for API interactions and real-time data fetching. | **9** | |
| | • **M 3.4 (Week 11):** UI component testing and debugging phase. | **11** | |

| | | | |
|---|---|---|---|
| **PROJECT / GROUP NAME** | **Group 14** | | |
| **Start Date** | 2025.3.14 | **Finish Date** | 2025.6.10 |
| **Aim / Objective** | Testing and Quality Assurance | | |
| **Work package Manager** | Xiaopeng Zhang | | |
| **Contributors to this package** | All | | |
| **Description / Activities** | **Task 4.1 Functional Testing**: Functional testing will ensure that each feature of the website works as expected.<br>**Task 4.2 Usability Testing**: Usability testing will assess how user-friendly and intuitive the website is for users of different backgrounds<br>▪ 4.2.1 **Navigation Testing:** Verify that the site's structure is easy to navigate, with clearly defined menus and categories.<br>▪ 4.2.2 **User Interface (UI) Testing:** Assess the visual appeal and user interaction with different elements, ensuring they are visually consistent with the company's branding<br>**Task 4.3 Performance Testing**: Performance testing will ensure that the website performs optimally under various conditions.<br>▪ 4.3.1 **Load Testing:** Simulate high traffic to evaluate how the website handles increased user interactions.<br>▪ 4.3.2 **Speed Testing:** Check that the website loads quickly, even with high-resolution images and videos.<br>▪ 4.3.3 **Stress Testing:** Test the limits of the website's infrastructure to handle unexpected spikes in user load.<br>**Task 4.4 Security Testing**: Security testing will ensure that user data is protected, particularly for e-commerce transactions and personal health information.<br>▪ 4.4.1 **Data Encryption**: Test the encryption of sensitive data, especially in payment systems and health-related records.<br>▪ 4.4.2 **Vulnerability Scanning:** Conduct tests to identify any security vulnerabilities within the website's code.<br>▪ 4.4.3 **Access Control:** Verify that user roles and permissions are correctly implemented to prevent unauthorized access. | | |
| **Milestones** | | Week | |
| | - **M 4.1**: Initial functional testing and UI/UX review. | **4/5** | |
| | - **M 4.2**: Performance testing, including load and speed tests. | **6/7** | |
| | - **M 4.3**: Interim report – focus on integrated features and SDG compliance. | **9** | |
| | - **M 4.4**: Final functional and security testing. | **11** | |
| **Deliverables** | | Week | |
| | **D 4.1**: Test cases for functional and usability testing. | **4/5** | |
| | **D 4.2**: Interim report with identified bugs and resolved issues. | **9** | |
| | **D 4.3**: Final test results, including performance and security testing outcomes. | **13/final week** | |