Engenharia de Requisitos

Hugo Coelho 1162086
Ilídio Magalhães 1191577
Paulo Abreu 1240481
Pedro Oliveira 1240482

January 12, 2025

1 Overview

In Sprint 2, our main focus was on rectifying the work we had done in the previous sprint based on the feedback provided by the professor. We also worked on the implementation of composite products, as outlined in the following user story:

"As a Producer, I want to be able to create a basket composed of simple products, which can also be offered individually."

This involved adjusting our approach to product offerings, ensuring that composite products could be created from simple products while maintaining the option to offer these individual products separately. The adjustments were made to meet the requirements and align with the professor's feedback, ensuring a more refined and functional implementation.

2 Division of Work

Scrum Master: Hugo Coelho
-Guaranteed compliance with agile practices.
-Ensured the team followed agile principles.
-Monitored and maintained agile methodology adherence.

Development Team:

- Hugo Coelho
- Ilídio Magalhães
- Paulo Abreu
- Pedro Oliveira

3 Deliverables

3.1 1. Component and Domain Model Diagrams

The **Component Diagram** represents the overall architecture of the system. It includes the following components:

- **Database**: Hosted on the DEI servers, the database serves as the central data storage for the application.
- **AMAP System**: This is the core system responsible for processing the data and integrating different parts of the application. The AMAP System connects to various components, including the backend and the frontend.
  - **AMAP Backend**: The backend component of the AMAP system processes business logic and handles requests from the frontend and external systems.
  - **AMAP Web**: This component refers to the web interface that interacts with the backend. It contains the implementation of the frontend and provides the necessary functionality for users to interact with the system.
  - **User Interface (UI)**: This is the user-facing component, allowing users to interact with the application, view data, and manage processes.

Below is a visual representation of the component diagram:

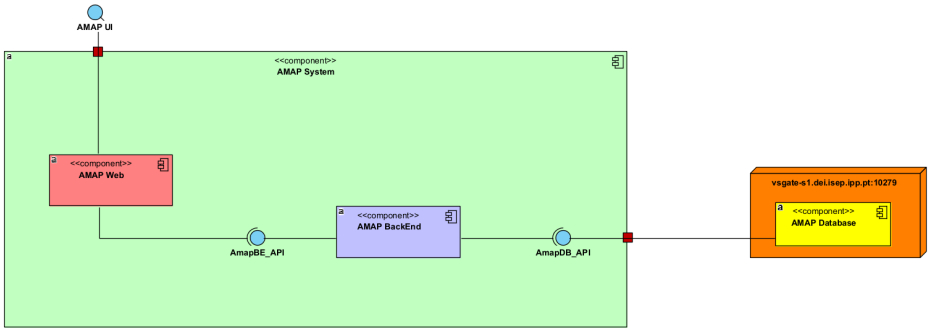

Figure 1: Component Diagram of the System

After explaining the component diagram, we proceed to the **Domain Model Diagram**, which represents the entities and their relationships within the system.

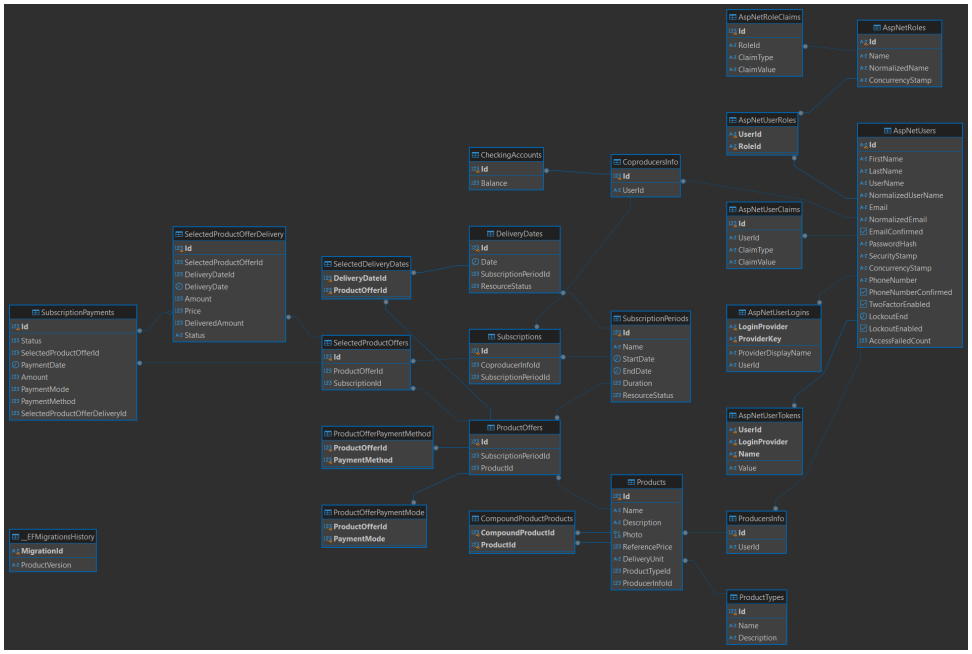

Figure 2: Domain Model Diagram

4 Technology Stack

4.1 Frontend:

- **HTML**: Markup language used to structure the content of web pages. It is the fundamental base for frontend development.
- **CSS**: Cascading style sheets used to control the layout and visual style of web pages, ensuring consistent presentation.
- **JavaScript**: Programming language used to create interactivity and dynamic content on web pages. It allows users to interact with the interface efficiently.

4.2 Backend:

- C#: Primary programming language for backend development with .NET 8.0.

4.3 Database and Hosting:

- **PostgreSQL**: Open-source relational database management system used to store and manage the application's data. Chosen for its reliability, scalability, and support for complex queries.
- **DEI Servers**: The PostgreSQL database is hosted on the DEI servers, providing a secure and reliable environment for the application's data storage.

5 User Story Analysis

User Story 1: *As a Producer, I want to define the product catalog I produce, with name, brief description, photo, reference price, and delivery units for each product. This product can be simple (e.g., cured goat cheese) or composite (e.g., basket of assorted vegetables);*

Responsible Group Members:

- Hugo Coelho
- Ilídio Magalhães
- Paulo Abreu
- Pedro Oliveira

5.1 1. Define Product Catalog

5.1.1 1.1 Create a New Product

MVP:

- US1-TASK01: Define a name
- US1-TASK02: Add a brief description
- US1-TASK03: Upload a photo
- US1-TASK04: Set a reference price
- US1-TASK05: Define delivery units
- US1-TASK06: Specify the product type (Simple or Basket)

MVI1:

- US1-TASK07: Include a dropdown or selection mechanism to specify product type (simple or composite).

MVI2:

- US1-TASK08: Add support for composite products, allowing producers to select and group simple products.
- US1-TASK09: Allow multiple photos per product.
- US1-TASK10: Add tags to categorize products.

Backlog:

- US1-TASK11: Reuse existing products as components when adding basket products.

5.1.2 1.2 View Product List

MVP:

- US1-TASK12: List Photo.
- US1-TASK13: List Name.
- US1-TASK14: List Price.
- US1-TASK15: List Delivery Units.

MVI1:

- US1-TASK16: Add functionality to display product by clicking on a product.
- US1-TASK17: Display a grid with larger photos for easier visual identification.
- US1-TASK18: Create a category management menu to add, edit, or delete categories.

MVI2:

- US1-TASK19: Add filters.
- US1-TASK20: Include sorting options for columns like price or name.

Backlog:

- US1-TASK21: Add a search bar to quickly find products by name or keywords.

5.1.3 1.3 Edit Product Details

MVP:

- US1-TASK22: Add an "Edit" button for each product in the catalog to access a pre-filled form for updating details.
- US1-TASK23: Allow updates to all fields, including photos and product types.

MVI1:

- US1-TASK24: Show change history.

MVI2:

- US1-TASK25: Notify customers about significant updates.
- US1-TASK26: Enable inline editing for simple fields like name, price, or delivery units.

Backlog:

- US1-TASK27: Notify stakeholders (e.g., consumers or AMAP admins) when key product details are updated.

5.1.4 1.4 Deactivate Product

MVI1:

- US1-TASK28: Ensure the product has no dependencies (e.g., in orders or baskets).
- US1-TASK29: Archive the product to make it inactive but keep it in the system for historical purposes.

Backlog:

- US1-TASK30: Automate notifications to customers and partners when a product is deactivated.

5.2 2. Advance Product Management

5.2.1 2.1 Categories and Tags

MVI2:

- US1-TASK31: Create a category management menu to add, edit, or delete categories.
- US1-TASK32: Allow producers to assign specific categories to products (e.g., "vegetables").

Backlog:

- US1-TASK33: Implement filters based on tags to simplify catalog searches.

5.2.2 2.2 Catalog Customization

MVI2:

- US1-TASK34: Allow producers to toggle between list and grid views.
- US1-TASK35: Offer sorting options, such as by price, creation date, or popularity.

5.3 3. Reports and Analytics

5.3.1 3.1 Catalog Analytics

MVI1:

- US1-TASK36: Provide sales reports for each product, including units sold and revenue generated.

Backlog:

- US1-TASK37: Offer insights on the most viewed or most added-to-composite products.

5.3.2 3.2 Sustainability and Certification

MVI1:

- US1-TASK38: Allow producers to include certifications (e.g., "Organic Production") and cultivation methods.

Backlog:

- US1-TASK39: Create a section to highlight the environmental impact or sustainable practices of products.

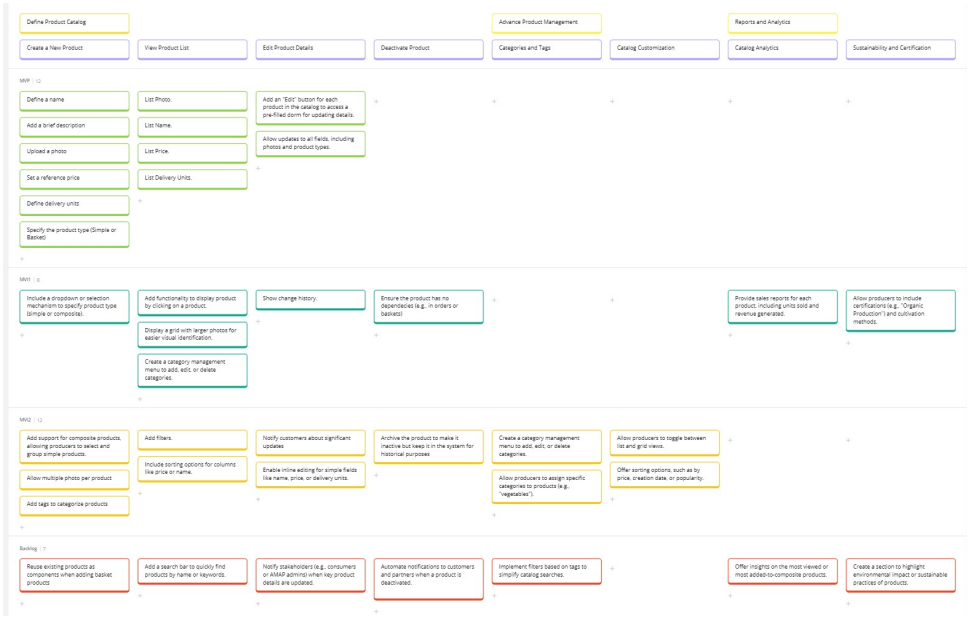

Figure 3: Story Mapping - First User Story

Activities	Sprint 1	Sprint 2
Define Product Catalog	TASK01, TASK02, TASK03, TASK04, TASK05, TASK06, TASK07, TASK12, TASK13, TASK14, TASK15, TASK16	TASK17, TASK22, TASK23
Advance Product Management	X	X
Reports and Analytics	X	X

Table 1: Activity Distribution Across Sprints

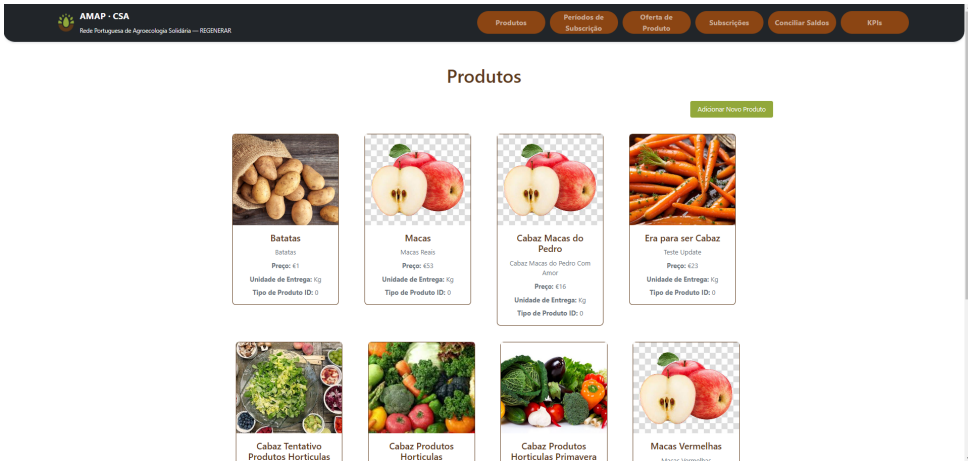

Figure 4: Products View

5.3.3 MVP (Minimum Viable Product)

- **Tasks:** TASK01, TASK02, TASK03, TASK04, TASK05, TASK06, TASK12, TASK13, TASK14, TASK15, TASK22, TASK23, TASK28, TASK29, TASK30

Reasoning: The MVP focuses on enabling producers to define and manage their product catalog effectively. It includes core functionalities like adding basic product details (name, description, photo, price, and delivery units), listing products, and editing product information. Deactivating products is also part of the MVP to ensure producers can manage their product catalog dynamically.

5.4 MVI1 (Minimum Viable Increment 1)

- **Tasks:** TASK07, TASK08, TASK09, TASK10, TASK11, TASK16, TASK17, TASK18, TASK19, TASK20, TASK21, TASK24, TASK25, TASK31, TASK32

Reasoning: After the MVP, this increment enhances usability and product management features. This includes the ability to specify product types, support multiple photos per product, and categorize products. The product list is enhanced with more user-friendly features such as sorting, filtering, and viewing product details upon click.

5.4.1 MVI2 (Minimum Viable Increment 2)

- **Tasks:** TASK34, TASK35, TASK36, TASK38, TASK39, TASK40

Reasoning: This increment focuses on advanced features for catalog customization and reporting. Producers can customize the catalog views, sort products based on different parameters, and gain insights into product performance. Additionally, sustainability and certification features are introduced, providing transparency and aligning with environmental standards.

5.4.2 Justification for Prioritization

1. **MVP**: The MVP focuses on establishing a functional product catalog system, which is the foundation for managing product offerings. These are the basic requirements to enable producers to list, manage, and deactivate products.

2. **MVI1**: Enhances the product catalog's usability, including categorization, photo management, and product details viewing, improving the overall user experience for both producers and consumers.

3. **MVI2**: Adds flexibility and advanced features like catalog customization, sorting options, and reporting, offering deeper insights and customization for producers.

5.4.3 Conclusion

This prioritization plan enables the delivery of a functional and flexible product catalog system in stages. The MVP ensures the basic catalog management features are ready for use, while the subsequent increments enhance the system's usability, customization, and reporting capabilities. The backlog items are non-critical but add value in the long term, ensuring the system can evolve based on user needs and business goals.

5.5 User Story 2

Description: *"As an AMAP Manager, I want to be able to define a subscription period with a specific duration (e.g., semester), setting start and end dates, as well as the dates when deliveries will be made within the period, automatically notifying the producers at the start of the process."*

Responsible Group Members:

- Hugo Coelho
- Ilídio Magalhães
- Paulo Abreu

5.5.1 1. Find Subscription Periods

- **Open Subscription Period Listing Page**
  - **MVP:**
    - US2-TASK01: Display subscription period name.
    - US2-TASK02: Display subscription period periodicity.
    - US2-TASK03: Display subscription period start date.
    - US2-TASK04: Display subscription period end date.
- **Pagination**
  - **MVI2:**
    - US2-TASK05: Go to first/last page.
    - US2-TASK06: Navigate between pages.
  - **Backlog:**
    - US2-TASK07: Select subscription periods per page.
- **Refine List**
  - **Backlog:**
    - US2-TASK08: Search by free text.
    - US2-TASK09: Refine by Start/End Date Interval Range.
    - US2-TASK10: Refine by current date.
- **Sort List**
  - **MVI2:**
    - US2-TASK11: Sort by Periodicity.
    - US2-TASK12: Sort by Start/End Date.
    - US2-TASK13: Sort by Subscription Period Name.

5.5.2 2. Create Subscription Period

- **Open the Subscription Period creation page**
  - **MVP:**
    - US2-TASK14: Display a form with empty subscription period details.
- **Name the Subscription Period**
  - **MVP:**
    - US2-TASK15: Name the subscription period.
  - **MVI1:**
    - US2-TASK16: Use a unique subscription period name.
- **Select Subscription Period Periodicity**
  - **MVP:**
    - US2-TASK17: Select subscription period periodicity.
- **Choose the Start Date**
  - **MVP:**
    - US2-TASK18: Choose the Start Date.
- **Choose the End Date**
  - **MVP:**
    - US2-TASK19: Choose the End Date.
- **Add the Delivery Dates**
  - **MVP:**
    - US2-TASK20: Choose a delivery date.
    - US2-TASK21: Add/Remove delivery date.
- **Create the new Subscription Period**
  - **MVP:**
    - US2-TASK22: Create the subscription period.
    - US2-TASK23: Send an email notification to the producers with the name of the new subscription period.
    - US2-TASK24: Return to subscription period listing page.
  - **Backlog:**
    - US2-TASK25: Send email notifications to producers with the subscription period details.

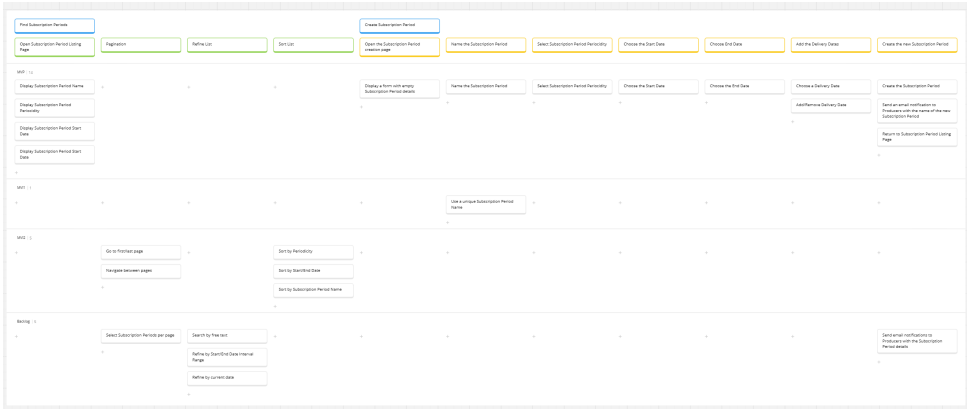

Figure 5: Story Mapping - User Story 2

Activities	Sprint 1	Sprint 2
Find Subscription Periods	TASK01, TASK02, TASK04	TASK03
Create Subscription Period	TASK14, TASK15, TASK18, TASK19, TASK22, TASK23, TASK24	TASK09, TASK17, TASK20, TASK21

Table 2: Sprint Activities

Figure 6: Subscription Period View

5.5.3 MVP (Minimum Viable Product)

(TASK01-TASK04, TASK14, TASK15, TASK17-TASK24)

- **Reasoning**: The MVP focuses on delivering the essential features required for the basic operation of the AMAP system. This includes the creation and management of subscription periods (TASK01-TASK04, TASK14, TASK15, TASK17-TASK24) and auxiliary functions like automated email notifications. These core functionalities ensure the system addresses immediate business needs while establishing a solid foundation for future enhancements.

—

5.5.4 MVI1 (Minimum Viable Increment 1)

(TASK16)

- **Reasoning**: This increment addresses the critical need for clear and reliable identification of subscription periods by enforcing unique names. As the primary reference in the

app and email notifications, unique names prevent confusion and mismanagement. This enhancement ensures accurate communication, supports efficient subscription tracking, and maintains the integrity of user interactions.

—

5.5.5 MVI2 (Minimum Viable Increment 2)

(TASK05, TASK06, TASK11, TASK12, TASK13)

- **Reasoning**: This increment is designed to improve the functionality and usability of the subscription period management system by addressing key user needs for navigation and organization. Pagination tasks (TASK05 and US2-TASK06) ensure users can efficiently navigate large datasets without performance degradation. Sorting tasks (TASK11 to TASK13) empower users to organize data in meaningful ways, making it easier to locate specific subscription periods. These enhancements strike a balance between functionality and simplicity, ensuring the system remains intuitive while accommodating growing data volumes.

—

—

5.5.6 Justification for Prioritization

1. **MVP**: These are the essential features required to deliver a functional subscription period management system, addressing the core needs of subscription creation, management, and basic email notifications.

2. **MVI1**: Builds on the MVP by ensuring clear identification of subscription periods through unique names, addressing a critical usability issue that directly impacts user operations.

3. **MVI2**: Focuses on improving system usability and efficiency with navigation and sorting enhancements, ensuring users can manage larger datasets effectively and organize information intuitively.

5.5.7 Conclusion

This prioritization ensures the subscription period management system is delivered incrementally, starting with the MVP to establish core functionality for creating and managing periods. MVI1 addresses critical usability by enforcing unique names, while MVI2 improves navigation and sorting for efficient data handling. The backlog captures advanced features like search and filtering for future enhancements. This approach balances immediate functionality with continuous improvement to meet evolving user needs.

5.6 User Story 3

Description: *"As a Producer, I want to be able to create the product offer during the defined subscription period by AMAP management. To do this, I should be able to choose the dates I want to deliver products, select from my catalog which products I want to deliver and on which dates, as well as define the available payment methods and their breakdown (e.g., full or installment payment)."*

Responsible Group Members:

- Hugo Coelho
- Pedro Oliveira

5.6.1 1. Select Products to Product Offer

- **Choose Product from Catalog**
  - **MVP:**
    - US3-TASK01: Display a list of products from the producer's catalog.
    - US3-TASK02: Easy access by pressing the first letter of the product.
  - **MVI 1:**
    - US3-TASK03: Create a dropdown menu with the existing products.
  - **MVI 2:**
    - US3-TASK04: Allow bundling of products (e.g., combining multiple items into a package).
    - US3-TASK05: Allow selection of multiple products for a single delivery date.
  - **Backlog:**
    - US3-TASK06: Implement filters to quickly search for products by name, type, or category.

5.6.2 2. Manage Subscription Periods

- **Allow producers to select the subscription period from available options.**
  - **MVP:**
    - US3-TASK07: Create a dropdown menu with subscription periods.
    - US3-TASK08: Display all subscription periods available.
  - **MVI 1:**
    - US3-TASK09: Display the name and season of the chosen subscription period.

5.6.3 3. Set Delivery Schedule

- **Choose Delivery Dates**
  - **MVP:**
    - US3-TASK10: Choose a date between the start and finish of the Subscription Period.
    - US3-TASK11: Provide a calendar to select delivery dates within the subscription period.
  - **MVI 1:**
    - US3-TASK12: Provide the delivery dates defined for the Subscription Period.

        * US3-TASK13: Create a dropdown menu with only the dates chosen in the Subscription Period.
    - **MVI 2:**
        * US3-TASK14: Display a warning if selected dates fall outside the allowed subscription period.
        * US3-TASK15: Suggest optimal delivery dates based on historical data or demand trends.
    - **Backlog:**
        * US3-TASK16: Integrate with external calendar systems for seamless scheduling.

- **Update Delivery Dates**
    - **MVP:**
        * US3-TASK17: Update or choose a more convenient delivery date for the product's delivery.
        * US3-TASK18: Create a dropdown menu with delivery dates.
    - **MVI 1:**
        * US3-TASK19: Remove previously chosen delivery date.

5.6.4 4. Define Payment Details

- **Select Payment Methods**
    - **MVP:**
        * US3-TASK20: Provide options for payment methods (e.g., MBWay, credit card).
        * US3-TASK21: Create a dropdown menu with options.
    - **MVI 1:**
        * US3-TASK22: Allow selection of multiple payment methods for flexibility.
        * US3-TASK23: Create a checkbox to select various types of payment methods.
    - **Backlog:**
        * US3-TASK24: Integrate payment gateway for secure online transactions.

- **Update Payment Methods**
    - **MVP:**
        * US3-TASK25: Allow producers to update payment methods.
    - **MVI 1:**
        * US3-TASK26: Create a checkbox to select various types of payment methods.
        * US3-TASK27: Check or uncheck payment method.
    - **Backlog:**

        * US3-TASK28: Notify consumers of changes in payment methods.

- **Define Payment Breakdown**
  - **MVP:**
    * US3-TASK29: Offer options for full or installment payments.
    * US3-TASK30: Create a dropdown menu with payment breakdown options.
  - **MVI 1:**
    * US3-TASK31: Add multiple payment breakdown methods.
    * US3-TASK32: Create a checkbox to select various types of payment breakdowns.
  - **Backlog:**
    * US3-TASK33: Enable notifications/reminders for upcoming installment payments.

- **Update Payment Breakdown**
  - **MVP:**
    * US3-TASK34: Allow producers to update payment methods.
  - **MVI 1:**
    * US3-TASK35: Create a checkbox to select various types of payment breakdowns.
    * US3-TASK36: Check or uncheck payment method.
  - **Backlog:**
    * US3-TASK37: Notify consumers of changes in payment breakdowns.

5.6.5 5. Confirm Product Offering

- **Final Review and Submission**
  - **MVP:**
    * US3-TASK38: Display a summary of selected dates, products, and payment options for review.
    * US3-TASK39: Allow submission of the product offering to the AMAP system.
    * US3-TASK40: Create submission button.
  - **MVI 2:**
    * US3-TASK41: Generate a downloadable summary report of the product offering.
  - **Backlog:**
    * US3-TASK42: Send a confirmation email upon successful submission.

5.6.6 6. Display and Manage Product Offerings

- **List All Product Offerings**
  - **MVP:**
    - US3-TASK43: Create a table to display all product offerings.
    - US3-TASK44: Include columns for key details (e.g., subscription period, delivery dates, products, payment methods).
  - **MVI 1:**
    - US3-TASK45: Add filters (e.g., by subscription period, product type) to refine the displayed offerings.
  - **MVI 2:**
    - US3-TASK46: Include a search bar for quick lookup of specific offerings.
  - **Backlog:**
    - US3-TASK47: Add pagination for better handling of large data sets.
- **Display Actions for Each Offering**
  - **MVP:**
    - US3-TASK48: Include an "Edit" button for each row in the table to update product offerings.
  - **MVI 1:**
    - US3-TASK49: Include a "Delete" button for each row to remove the product offering.
  - **Backlog:**
    - US3-TASK50: Add a "View Details" button to review product offering details.

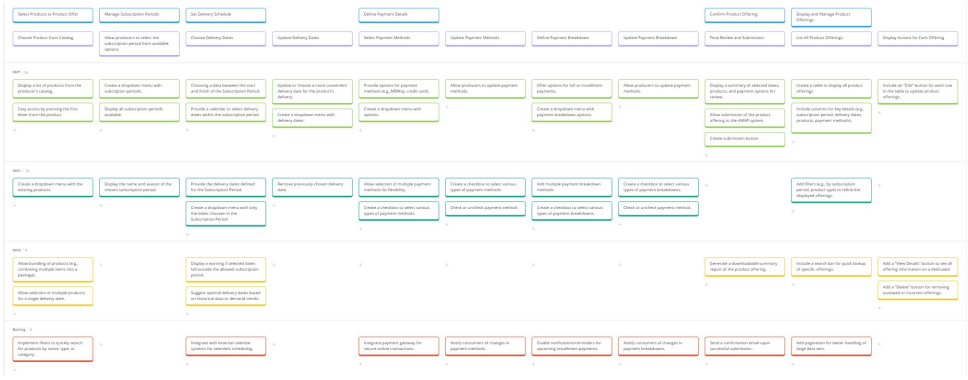

Figure 7: Story Mapping - User Story 3

Activities	Sprint 1	Sprint 2
Select Products to Product Offer	TASK01, TASK02, TASK03	X
Manage Subscription Periods	TASK07, TASK08	TASK09
Set Delivery Schedule	TASK10, TASK11	TASK12, TASK13, TASK17, TASK18
Define Payment Details	TASK20, TASK21, TASK29, TASK30	TASK22, TASK23, TASK25, TASK26, TASK27, TASK31, TASK32, TASK34, TASK35, TASK36
Confirm Product Offering	TASK38, TASK39, TASK40	X
Display and Manage Product Offerings	TASK43, TASK44	TASK48

Table 3: Activity Distribution Across Sprints

Figure 8: Product Offer View

5.6.7 MVP (Minimum Viable Product)

- **Core Features:**
  - The MVP should focus on ensuring that the basic functionalities necessary for AMAP to operate are in place.
  - **Tasks:** TASK01, TASK02, TASK07, TASK08, TASK10, TASK11, TASK17, TASK18, TASK20, TASK21, TASK25, TASK29, TASK30, TASK34, TASK38, TASK39, TASK40, TASK43, TASK44, TASK48.
- **Reasoning:** These tasks ensure that the AMAP system can manage product offerings, allow users to select products, submit orders, and handle the essential operations of the service. These features are critical for the initial launch and usability of the system.

5.6.8 MVI1 (Minimum Viable Increment 1)

- **Enhancing Flexibility and Usability:**
  - Once the MVP is achieved, the focus will shift towards enhancing the system's usability, flexibility, and addressing some operational challenges.
  - **Tasks:** TASK03, TASK09, TASK12, TASK13, TASK19, TASK22, TASK23, TASK26, TASK27, TASK31, TASK32, TASK35, TASK36, TASK45.
- **Reasoning:** These tasks involve refining critical areas such as better delivery date options, user-friendly subscription management, and additional features that improve the overall user experience. These tasks also deal with operational elements like payment flexibility and optimized delivery management.

5.6.9 MVI2 (Minimum Viable Increment 2)

- **Advanced Functionalities:**
  - This increment introduces advanced features that optimize the user experience and address edge cases like invalid date selection and more complex bundling options.
  - **Tasks:** TASK04, TASK05, TASK14, TASK15, TASK41, TASK46, TASK49, TASK50.
- **Reasoning:** This increment adds advanced functionalities such as bundling options and more sophisticated delivery schedule management, addressing potential edge cases for the system's optimization.

5.6.10 Justification for Prioritization

- **MVP:** These tasks are critical for launching a functional product offering system that handles core requirements, such as managing subscriptions, product offerings, and payments.
- **MVI1:** Builds upon the MVP to enhance flexibility, usability, and operational efficiency, based on user feedback and further requirements.
- **MVI2:** Focuses on adding advanced features to optimize the user experience and system performance.

5.6.11 Conclusion

- This structured approach ensures the core functionalities of AMAP are addressed first with the MVP. The system will evolve in increments, improving usability and flexibility while addressing more advanced features in later stages. The backlog contains improvements for future development to ensure continuous growth and enhancement of the system.

User Story 4

As a Co-Producer, I want to be able to subscribe to the offer created by Producers, selecting the products and quantities I want to receive on each date, as well as choosing the payment method and installment options available.

Responsible Group Members:

- Hugo Coelho
- Paulo Abreu

Story Mapping

- **Find Product Offer**
  - **View Product Offer List**

        * **MVP:**
            · US3-TASK01: Display Product Offer List.
            · US3-TASK02: Display Description.
        * **MVI 1:**
            · US3-TASK03: Display Image.
            · US3-TASK04: Display Price.
        * **Backlog:**
            · US3-TASK05: Show Promotions.
    - **Pagination**
        * **MVI 2:**
            · US3-TASK06: Go to First and Last Page.
            · US3-TASK07: Navigation between Pages.
        * **Backlog:**
            · US3-TASK08: Select Products per Page.
    - **Refine List**
        * **MVI 2:**
            · US3-TASK09: Refine by Category.
        * **Backlog:**
            · US3-TASK10: Search by free text.
            · US3-TASK11: Refine by Price Range.
            · US3-TASK12: Refine by Promotions.
    - **Sort List**
        * **MVI 2:**
            · US3-TASK13: Sort by Price.
        * **Backlog:**
            · US3-TASK14: Sort by Stock.
            · US3-TASK15: Sort by Discount.
            · US3-TASK16: Sort by Recommended.

- **Select Product Offer**
    - **Open Detail Page for Product Offer**
        * **MVP:**
            · US4-TASK17: Display Image.
            · US4-TASK18: Display Details.
        * **MVI 2:**
            · US3-TASK19: Display Image Carousel.
        * **Backlog:**
            · US3-TASK20: Display Similar Products.
    - **Select Dates**
        * **MVP:**

        - US4-TASK21: Show Available Dates.
    - **Backlog:**
        - US3-TASK22: Suggest New Dates.
  - **Select Quantities**
    - **MVP:**
        - US4-TASK23: Show Input for Quantities per Date.

- **Add to Shopping Cart**
  - **Add to Shopping Cart**
    - **MVP:**
        - US4-TASK24: Add Product to Shopping Cart.
  - **View Selected Product Offer List**
    - **MVP:**
        - US4-TASK25: Display Total Amount.
    - **MVI 1:**
        - US4-TASK26: Delete Products from Shopping Cart.
    - **MVI 2:**
        - US4-TASK27: Change Quantity per Date.
    - **Backlog:**
        - US3-TASK28: Show Recommended Products based on Shopping Cart.

- **Checkout**
  - **Select Payment Method**
    - **MVP:**
        - US4-TASK29: Show Input for Payment Method.
    - **Backlog:**
        - US3-TASK30: Support MB.
        - US3-TASK31: Support MB Way.
        - US3-TASK32: Support for Credit Card.
  - **Select Payment Model**
    - **MVP:**
        - US4-TASK33: Show Input for Payment Mode.
    - **Backlog:**
        - US3-TASK34: Support Klarna.
  - **Confirm Information**
    - **MVI 1:**
        - US3-TASK35: Show Information About Subscriptions.
    - **Backlog:**
        - US3-TASK36: Email Notification.

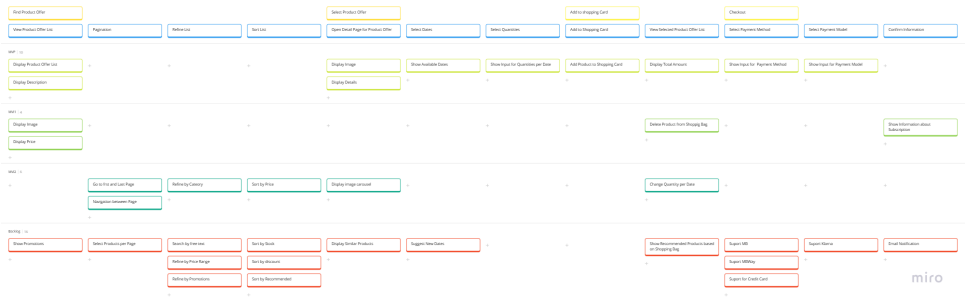

Figure 9: Story Mapping - User Story 4

Activities	Sprint 1	Sprint 2
Select Products Offer to Subscription	TASK01, TASK02	X
Define Details About Subscription	X	TASK21, TASK23
Define Details About Payment	X	TASK29, TASK33
Confirm Subscription	X	TASK35

Table 4: Sprint Activities for Subscription Management

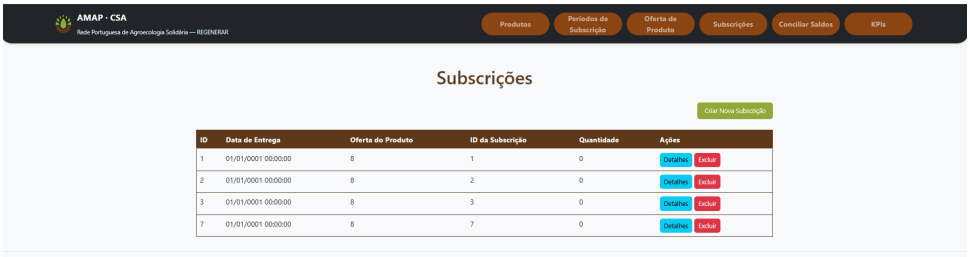

Figure 10: Subscriptions View

MVP (Minimum Viable Product)

- **Tasks:** TASK01, TASK02, TASK17, TASK18, TASK21, TASK23, TASK24, TASK25, TASK29, TASK33
- **Reasoning:** The MVP should focus on core functionalities required to enable the basic operations of selecting, managing, and submitting product offerings within the AMAP system. These are critical to establish a working solution and meet immediate business needs.

MVI1 (Minimum Viable Increment 1)

- **Tasks:** TASK03, TASK04, TASK26, TASK35

- **Reasoning:** After achieving the MVP, the focus shifts to improving flexibility and usability, such as refining delivery date selection, supporting multiple options, and enhancing payment flexibility.

MVI2 (Minimum Viable Increment 2)

- **Tasks:** TASK06, TASK07, TASK09, TASK13, TASK19
- **Reasoning:** This increment introduces advanced features, such as bundling and optimizing delivery schedules, while addressing edge cases like invalid date selection.

User Story 5

Como Gestor da AMAP quero poder despoletar o cálculo dos valores a pagar pelos Coprodutores e dos valores a receber pelos Produtores, tendo em conta as subscrições realizadas e os pagamentos realizados;

Story Mapping

1. **Calculate Payments for Co-Producer**
   (a) **View Co-Producer List**
      i. **MVP:**
         - US3-TASK01: Display Co-Producers List.
      ii. **MVI 1:**
      iii. **MVI 2:**
         - US3-TASK02: Display Some Details about Co-Producers.
      iv. **Backlog:**
   (b) **Pagination**
      i. **MVP:**
      ii. **MVI 1:**
      iii. **MVI 2:**
         - US3-TASK03: Go to First and Last Page.
         - US3-TASK04: Navigation between Page.
      iv. **Backlog:**
         - US3-TASK05: Select Co-Producers per Page.
   (c) **Refine List**
      i. **MVP:**
      ii. **MVI 1:**
      iii. **MVI 2:**
      iv. **Backlog:**
         - US3-TASK06: Search by free text.
         - US3-TASK07: Filters.

(d) **Sort List**

i. **MVP:**

ii. **MVI 1:**

iii. **MVI 2:**

iv. **Backlog:**

- US3-TASK08: Sort by Oldest Co-Producer.

(e) **Select Co-Producer**

i. **MVP:**

ii. **MVI 1:**

iii. **MVI 2:**

iv. **Backlog:**

- US3-TASK09: Show Co-Producer Details.

(f) **Calculate Amounts to be Paid**

i. **MVP:**

- US3-TASK10: Calculate Total Amount to be Paid.

ii. **MVI 1:**

- US3-TASK11: Calculate Amount to be Paid for Each Co-Producer.

iii. **MVI 2:**

- US3-TASK12: Export List As Report.

iv. **Backlog:**

- US3-TASK13: Show Every Subscriptions to be Paid for Each Co-Producer.
- US3-TASK14: Send Email with Report.

2. **Calculate Payments for Producers**

(a) **View Producers List**

i. **MVP:**

- US3-TASK15: Display Producers List.

ii. **MVI 1:**

iii. **MVI 2:**

- US3-TASK16: Display Some Details about Co-Producers.

iv. **Backlog:**

- US3-TASK17: Display Similar Products.

(b) **Pagination**

i. **MVP:**

ii. **MVI 1:**

iii. **MVI 2:**

- US3-TASK18: Go to First and Last Page.
- US3-TASK19: Navigation between Page.

iv. **Backlog:**

- US3-TASK20: Select Co-Producers per Page.

(c) **Refine List**

i. **MVP:**

ii. **MVI 1:**

iii. **MVI 2:**

iv. **Backlog:**

- US3-TASK21: Search by free text.

(d) **Sort List**

i. **MVP:**

ii. **MVI 1:**

iii. **MVI 2:**

iv. **Backlog:**

- US3-TASK22: Sort by Oldest Producer.

(e) **Select Co-Producer**

i. **MVP:**

ii. **MVI 1:**

iii. **MVI 2:**

iv. **Backlog:**

- US3-TASK23: Show Producer Details.

(f) **Calculate Amounts to Receive**

i. **MVP:**

- US3-TASK24: Calculate Total Amount to Receive.

ii. **MVI 1:**

- US3-TASK25: Calculate Amount to Receive for Each Producer.

iii. **MVI 2:**

- US3-TASK26: Export List As Report.

iv. **Backlog:**

- US3-TASK27: Show Every Subscriptions to Receive for Each Producer.
- US3-TASK28: Send Email with Report.

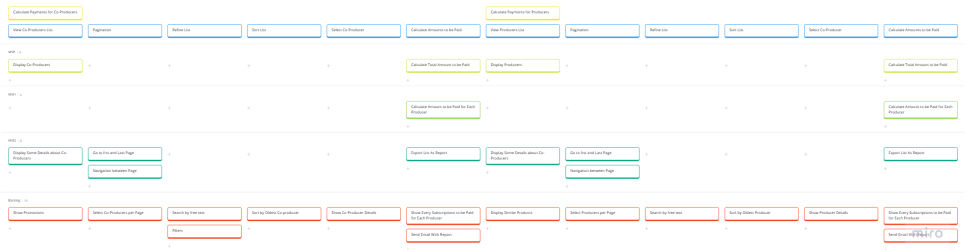

Figure 11: User Story 5- Story Mapping

User Story 6:

As the AMAP Manager, I want to be able to trigger the calculation of the amounts to be paid by the Co-Producers and the amounts to be received by the Producers, taking into account the subscriptions made and the payments processed.

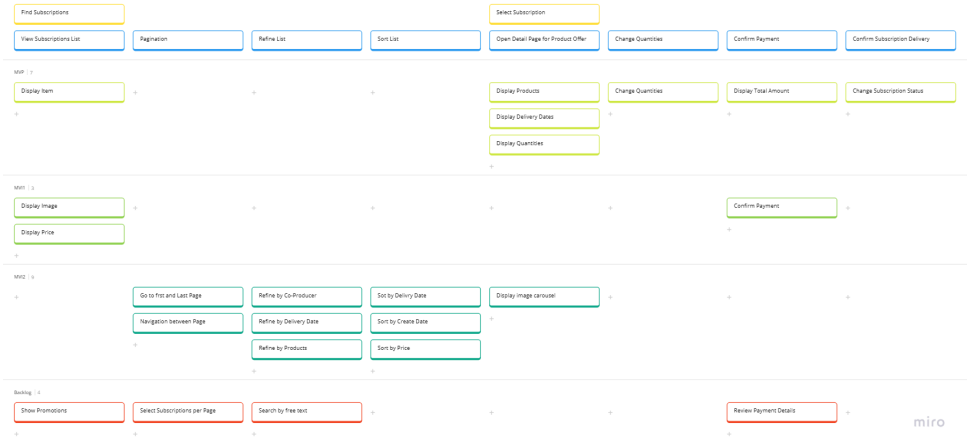

Figure 12: User Story 6- Story Mapping

User Story 7: KPI Consultation

As a Manager, I want to consult critical KPIs for management, such as the value delivered by producers per delivery and period, and the average value subscribed by co-producers per delivery and period.

Responsible Group Members:

- Hugo Coelho
- Pedro Oliveira

5.6.12 1. View Critical KPIs

1.1 Select KPI Type

MVP:

- US7-TASK01: Provide a dropdown menu to allow the manager to select Value Delivered by Producer or Average Value Subscribed.

MVI1:

- US7-TASK02: Display a brief description of each KPI type for clarification (e.g., Total value delivered per producer during the period).

Backlog:

- US7-TASK03: Allow multi-selection of KPIs for simultaneous display.

5.6.13 1.2 Load KPI Data

MVP:

- US7-TASK04: Provide a Load button that, when clicked, loads the data related to the selected KPI into a table format.
- US7-TASK05: Display the data in a clear, tabular format (columns for date, producer/co-producer, and value).

MVI2:

- US7-TASK06: Provide an option to download the table data as CSV or Excel file.

5.6.14 1.3 View Data Table

MVP:

- US7-TASK07: Display the table with the data showing the Value Delivered by Producer or Average Value Subscribed, based on the manager's selection from the dropdown.

MVI1:

- US7-TASK08: Sort List by Subscription Period.
- US7-TASK09: Sort List by Producer - (Value Delivered by Producer).

MVI2:

- US7-TASK10: Enable pagination for large sets of data to ensure the table remains usable.

5.6.15 1.4 Monitor Data Load Status

MVI2:

- US7-TASK11: Show a status message indicating the success or failure of loading the data after clicking the Load button.

Backlog:

- US7-TASK12: Allow filtering the data by period (e.g., current month, last 3 months, etc.).

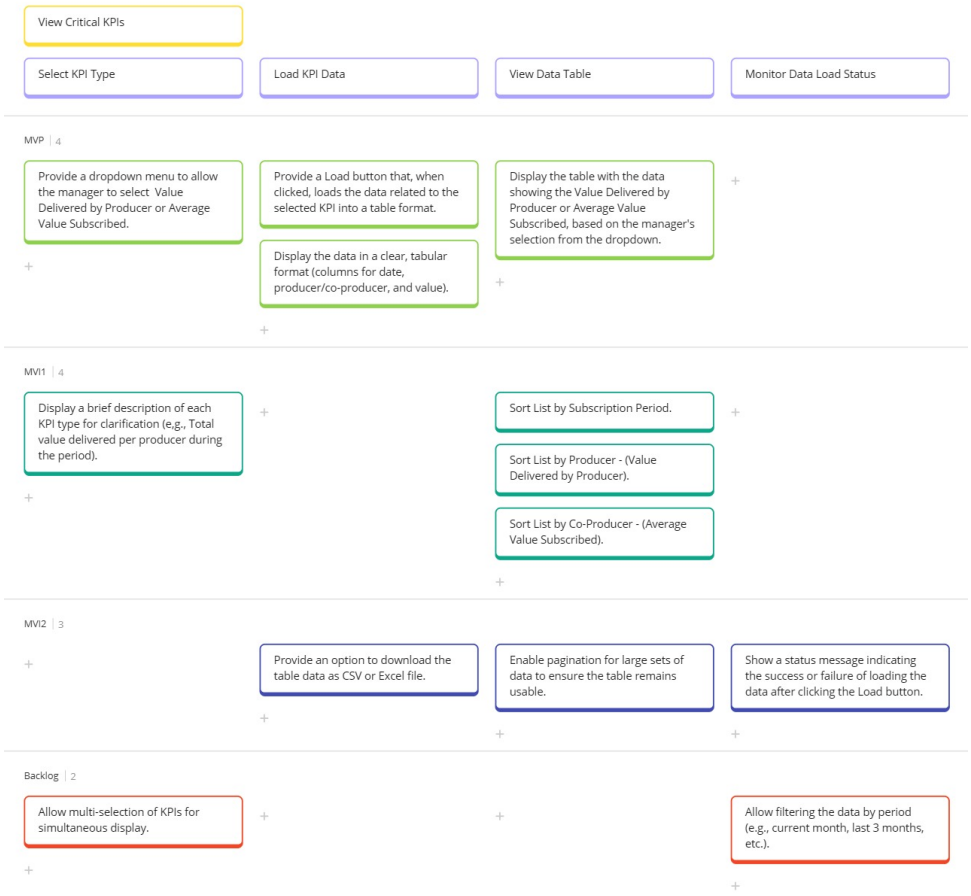

Figure 13: Story Mapping - User Story 7

Activities	**Sprint 1**	**Sprint 2**
View Critical KPIs	TASK1, TASK2, TASK4, TASK5, TASK6	X

Table 5: Activity Distribution Across Sprints for KPI Management

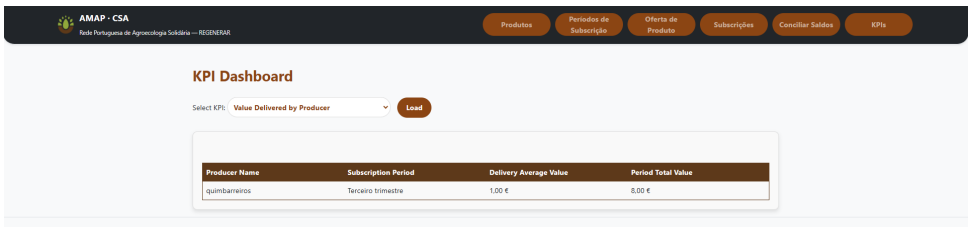

Figure 14: KPIs View

5.6.16 MVP (Minimum Viable Product)

- **Core Features:**
  - The MVP focuses on delivering the core functionalities needed for KPI selection, data loading, and basic visualization.

- **Tasks:** TASK01, TASK04, TASK05, TASK07.

- **Reasoning:** These tasks are essential for enabling the manager to view critical KPIs and make informed decisions, forming the foundation of the system.

5.6.17 MVI1 (Minimum Viable Increment 1)

- **Enhancing Usability and Functionality:**
  - The first increment focuses on enhancing the system by adding descriptions and sorting options.
  - **Tasks:** TASK02, TASK08, TASK09, TASK10.

- **Reasoning:** This will make the interface more intuitive for the manager and improve user experience by providing better navigation and data presentation.

5.6.18 MVI2 (Minimum Viable Increment 2)

- **Advanced Features:**
  - The second increment introduces advanced functionalities for better data handling and user experience.
  - **Tasks:** TASK06, TASK11, TASK12.

- **Reasoning:** These tasks focus on adding convenience features and improving system performance for more efficient management and decision-making.

5.6.19 Conclusion

- The story mapping organizes development into structured increments, ensuring delivery of core functionalities first while progressively enhancing the system. This approach guarantees a balance between immediate utility and future improvements, enabling efficient management of critical KPIs.

User Story 8: Calculate and Fetch Balances

As an AMAP Manager, I want to trigger the calculation of the differences between the amounts paid and the delivery amounts, updating the co-producers' account balances accordingly.

Responsible Group Members:

- Hugo Coelho

5.6.20 1. Calculate and Fetch Balances

1.1 Open Co-Producer Account Balances Page

MVP:

- US8-TASK01: Display button to calculate and fetch balances.

5.6.21 1.2 Calculate and Fetch Balances

MVP:

- US8-TASK03: Display Co-Producer.
- US8-TASK04: Display Co-Producer Balance.

MVI1:

- US8-TASK05: Add Pagination, Sorting and List Refinement..

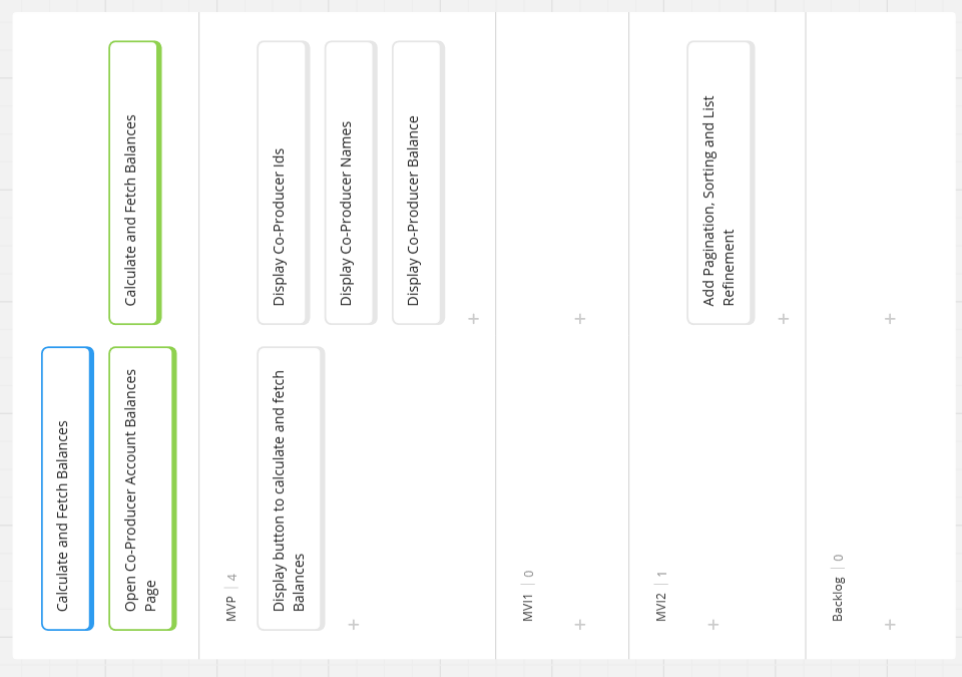

Figure 15: Story Mapping - Eight User Story

Activities	Sprint 1	Sprint 2
Calculate and Fetch Balances	TASK01	X
Calculate and Fetch Balances	TASK02, TASK03, TASK04	X

Table 6: Task Distribution Across Sprints for Calculate and Fetch Balances

5.7 User Story 9: Create a Product Basket

As a Producer, I want to create a basket composed of simple products, which can also be offered individually.

Responsible Group Members:

- Ilídio Magalhães
- Pedro Oliveira

5.7.1 1. Compose Basket Creation

1.1 API for Basket Creation

MVP:

- US9-TASK01: Create an endpoint to allow producers to create a new basket with basic information (e.g., name, description).
- US9-TASK02: Associate products with a basket.

MVI1:

- US9-TASK03: Add validation in the endpoint to ensure the basket has at least one product.
- US9-TASK04: Support custom pricing with an optional field.

5.7.2 1.2 API for Basket Management

MVP:

- US9-TASK05: Create an endpoint to retrieve a list of all baskets.
- US9-TASK06: Provide an endpoint to fetch details of a specific basket.

MVI1:

- US9-TASK07: Implement an endpoint to allow editing the basket's name, description, or pricing.
- US9-TASK08: Create a DELETE endpoint to remove a product from the basket.

Backlog:

- US9-TASK09: Add search and filtering parameters to the endpoint (e.g., by name, creation date, etc.).

5.7.3 1.3 Monitoring and Validation

MVP:

- US9-TASK10: Implement error handling and validation for all endpoints, returning meaningful HTTP status codes (e.g., 400 Bad Request for invalid inputs).

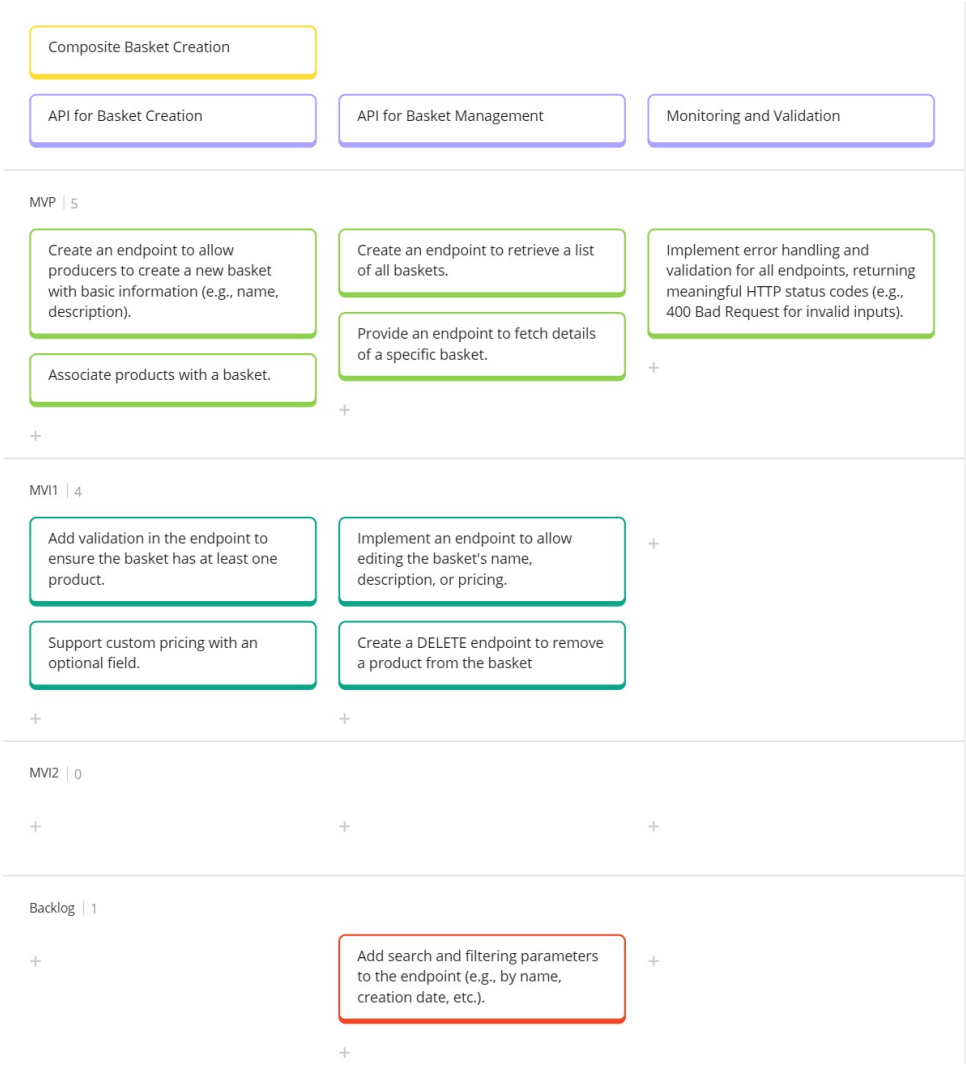

Figure 16: Story Mapping - Ninth User Story

Activities	**Sprint 1**	**Sprint 2**
Composite Basket Creation	X	TASK01, TASK02, TASK03, TASK05, TASK08, TASK10

Table 7: Activity Distribution Across Sprints for Composite Basket Creation

6 Non-Functional Requirements

Non-Functional Requirement 1: Multi-Tenancy The application is designed to support multi-tenancy, enabling multiple users to operate within the system independently. Each user will have their own isolated data, and they will not be able to access or modify data belonging to other users. This ensures proper data segregation and privacy between users within the system.

Non-Functional Requirement 2: Security The application incorporates security features such as authentication and authorization mechanisms. This ensures that only authorized users can access the system, and their access is restricted based on their roles. The security protocols safeguard user data and system integrity through controlled and secure access to the application.

Non-Functional Requirement 3: Cross-Platform Compatibility The application is built to run across a variety of devices and platforms. The system is compatible with multiple operating systems, including Android, iOS, Windows, Linux, and macOS, providing a consistent user experience across various platforms.

Non-Functional Requirement 4: Usability and User Experience The application prioritizes usability, focusing on creating an intuitive interface that is easy to navigate. The interface is designed to minimize complexity, allowing users to perform key tasks with few actions.

Conclusion

The foundation laid through this project represents the beginning of what could become a significant transformation for the AMAP initiative. By developing a structured and scalable system, we have created the core infrastructure that can support future enhancements and expansion. This marks the first step in enabling a more efficient, flexible, and user-friendly platform that has the potential to revolutionize the way producers and co-producers interact. As the project evolves, the groundwork established here will facilitate further innovation, paving the way for a more sustainable and impactful AMAP ecosystem. The project not only addresses current needs but also sets the stage for future growth and positive change within the initiative.