LE QUERNEC Loévan / FERREIRA Louise / BORNET Joshua / LEMPEREUR Julien CDOF2

Software Engineering – project

We choose to design an e-commerce website software for our activity. It is a primeur (a place where we sell fruits and vegetables). On the website, the user should be able to buy directly on the website what he wants and be delivered to his home.

The size of each sprint will be 2 weeks. We think it would be enough, given the size of our group, the project, and the recommendations.

User story

As a customer, I want to:

- Browse a selection of high-quality fresh fruits and vegetables, select my items, and place an order for delivery or pickup.
- Find a well-organized list of products on the platform, with clear images, product names, descriptions, prices, and availability.
- Filter and search for products by category, variety, or origin.
- Access detailed product pages with information about nutrition, farming practices, and source.
- Specify the quantity, add items to my cart, and view a summary of my selections.
- Adjust item quantities or remove products from my cart.
- Initiate the payment process with options for home delivery or in-store pickup.
- Provide delivery details or choose a pickup location.
- Calculate the final cost, including taxes and applicable fees.
- Enter my contact information and select a payment method.
- Receive an order confirmation email.
- Allow the store to verify product availability and estimate delivery or pickup time.
- Receive order status updates, including preparation, dispatch, and estimated delivery or pickup time.

Epic story

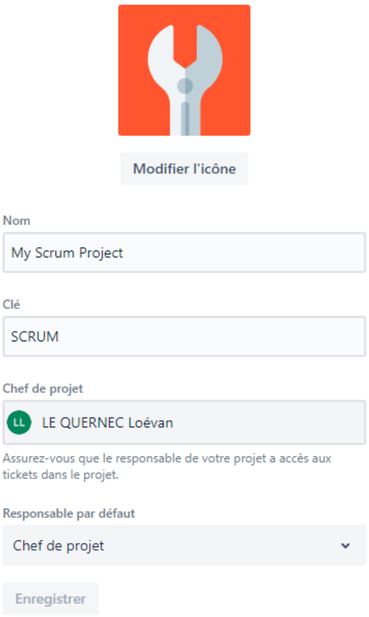
As a fresh produce shop owner, this involves:

- Platform setup: Choosing software and configuring it for diverse produce.
- Catalog management: Creating organized product listings with details and pricing.
- User registration: Implementing secure account creation and management.
- Search and navigation: Developing a search system and intuitive browsing.
- Product pages: Designing detailed product pages, including nutritional info and source details.
- Shopping cart and checkout: Building a smooth cart and checkout process with delivery and pickup options.
- Order management: Handling orders efficiently, with notifications for preparation.
- Payment processing: Ensuring secure payment methods and data encryption.
- Delivery and pickup: Managing delivery fees and establishing pickup locations.

- Order tracking and notifications: Providing order status updates and estimated times.
- Customer support: Integrating support channels and enabling customer feedback and reviews.

Jira

Here is the link to the Jira: https://cdof2grp2.atlassian.net.



Nom

My Scrum Project

Clé

SCRUM

Chef de projet

LL LE QUERNEC Loévan

Assurez-vous que le responsable de votre projet a accès aux tickets dans le projet.

Responsable par défaut

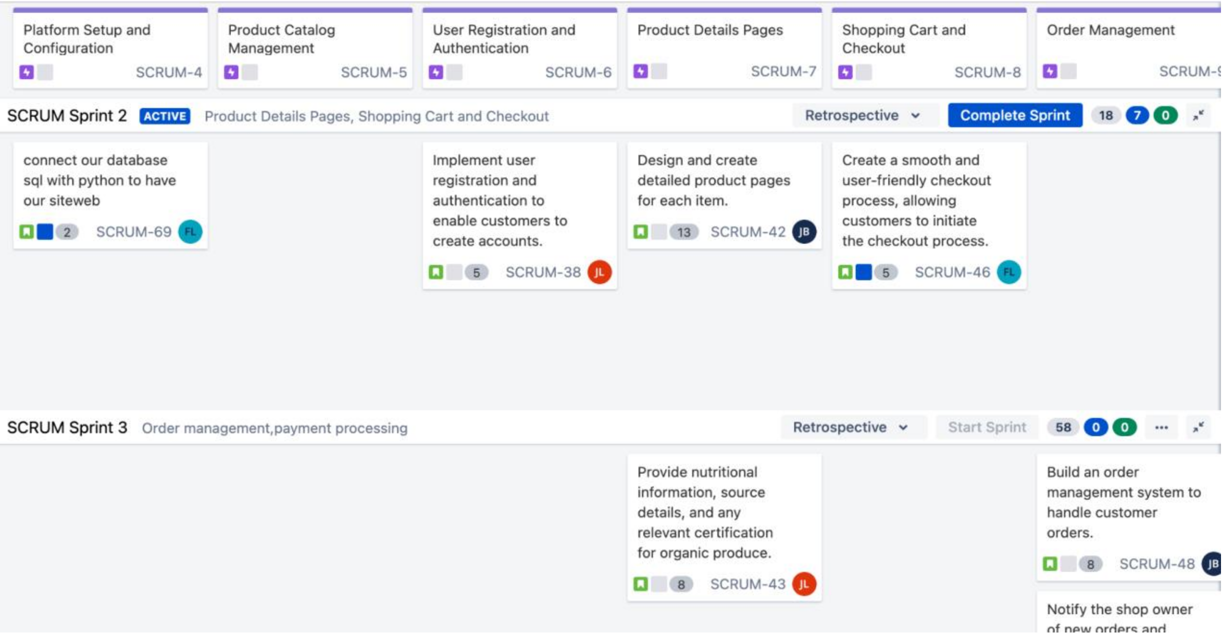
Chef de projet

Here is our plan:

- Define roles:
  - Product Owner (Louise): Ideally, the shop owner or a representative responsible for defining and prioritizing the product backlog.
  - Scrum Master (Julien): A dedicated Scrum Master who will facilitate the Scrum process and help the team.
  - Development Team (Loévan and other people): Cross-functional members responsible for implementing the features.
  - The Director of Operations (Joshua) oversees and optimizes daily activities, resource management, and strategic planning to ensure an organization runs efficiently and achieves its objectives.
- Create a product backlog:
  - The Product Owner should start by creating a product backlog in Jira, which should include a list of user stories and epics that need to be addressed for the e-commerce project.
  - User stories could include tasks like "User can search for fruits and vegetables," "User can add items to the shopping cart," "User can select delivery or pickup options," and so on.
  - Epics could represent larger themes like "User authentication and registration," "Product catalog management," "Order processing," and more.

- Define the duration of our sprints. Here is 2-week sprint duration.
- For each sprint, plan specific activities based on the priorities from the product backlog. Consider the features and functionalities that should be developed within each sprint.
- Assign user stories and epics to team members for each sprint.
- At the beginning of each sprint, we conduct sprint planning meetings with the Product Owner, Scrum Master, and Development Team.
- Hold daily standup meetings (daily scrum) to ensure team members are aligned, discuss progress, and identify any obstacles or blockers.
- After each sprint, we conduct a sprint review to demonstrate completed work to stakeholders.
- We continue the process, iterating through sprints, addressing user stories and epics, …

Here is an extract of our User Story Map of our project:



Platform Setup and Configuration SCRUM-4

Product Catalog Management SCRUM-5

User Registration and Authentication SCRUM-6

Product Details Pages SCRUM-7

Shopping Cart and Checkout SCRUM-8

Order Management SCRUM-9

SCRUM Sprint 2 ACTIVE Product Details Pages, Shopping Cart and Checkout

connect our database sql with python to have our siteweb 2 SCRUM-69 FL

Implement user registration and authentication to enable customers to create accounts. 5 SCRUM-38 JL

Design and create detailed product pages for each item. 13 SCRUM-42 JB

Create a smooth and user-friendly checkout process, allowing customers to initiate the checkout process. 5 SCRUM-46 FL

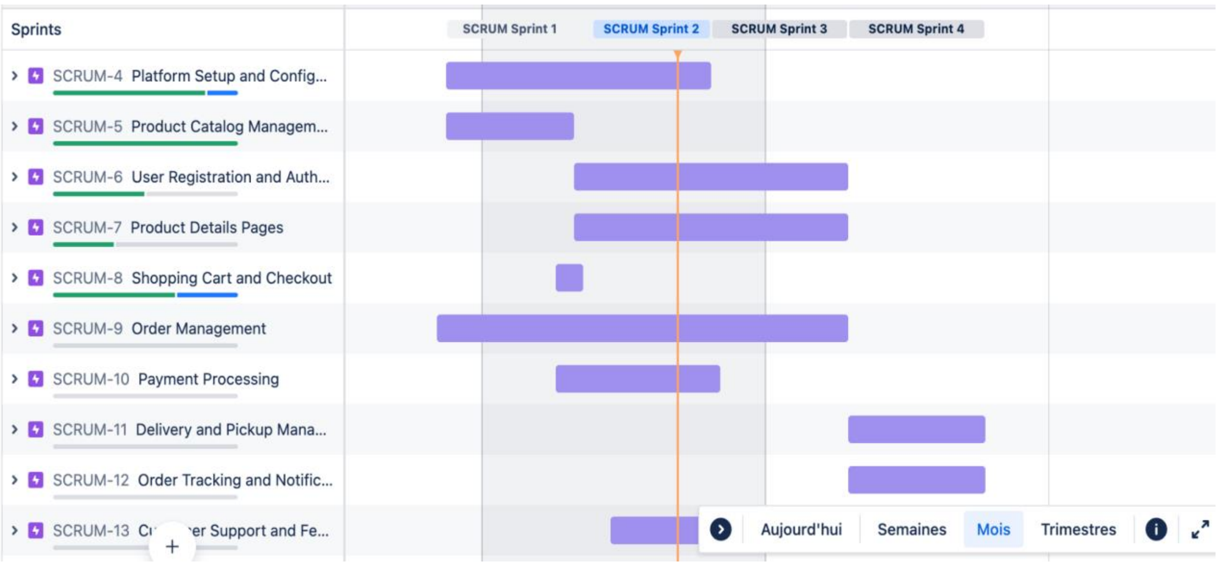
SCRUM Sprint 3 Order management,payment processing

Provide nutritional information, source details, and any relevant certification for organic produce. 8 SCRUM-43 JL

Build an order management system to handle customer orders. 8 SCRUM-48 JB

Notify the shop owner of new orders and

Here is our chronology:



Sprints

SCRUM-4 Platform Setup and Config...

SCRUM-5 Product Catalog Managem...

SCRUM-6 User Registration and Auth...

SCRUM-7 Product Details Pages

SCRUM-8 Shopping Cart and Checkout

SCRUM-9 Order Management

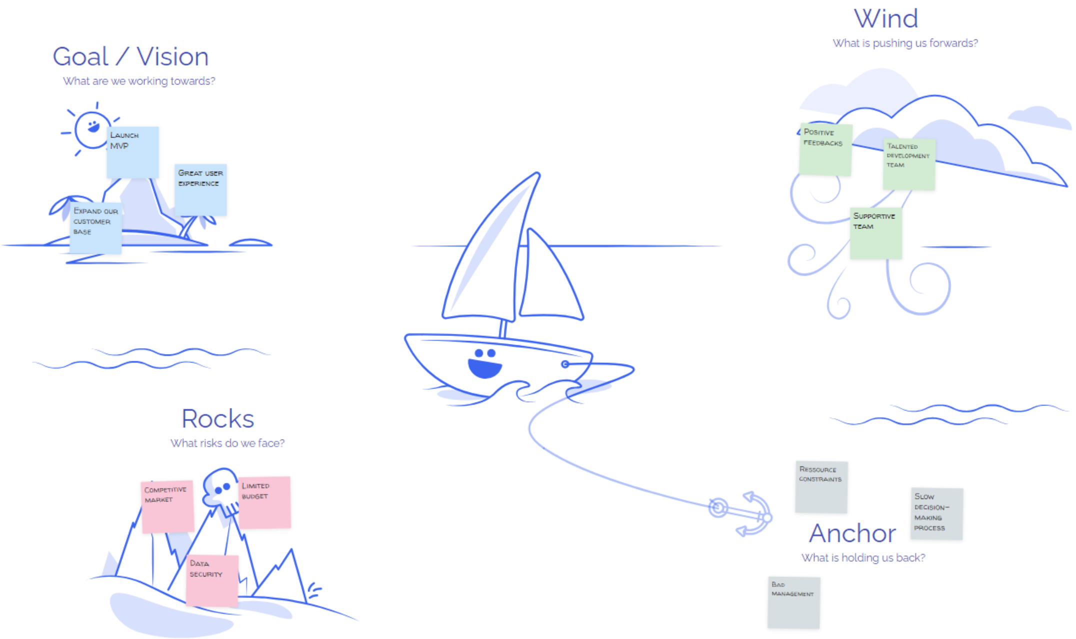
SCRUM-10 Payment Processing

SCRUM-11 Delivery and Pickup Mana...

SCRUM-12 Order Tracking and Notific...

SCRUM-13 Cu er Support and Fe...

An example of a speedboat:

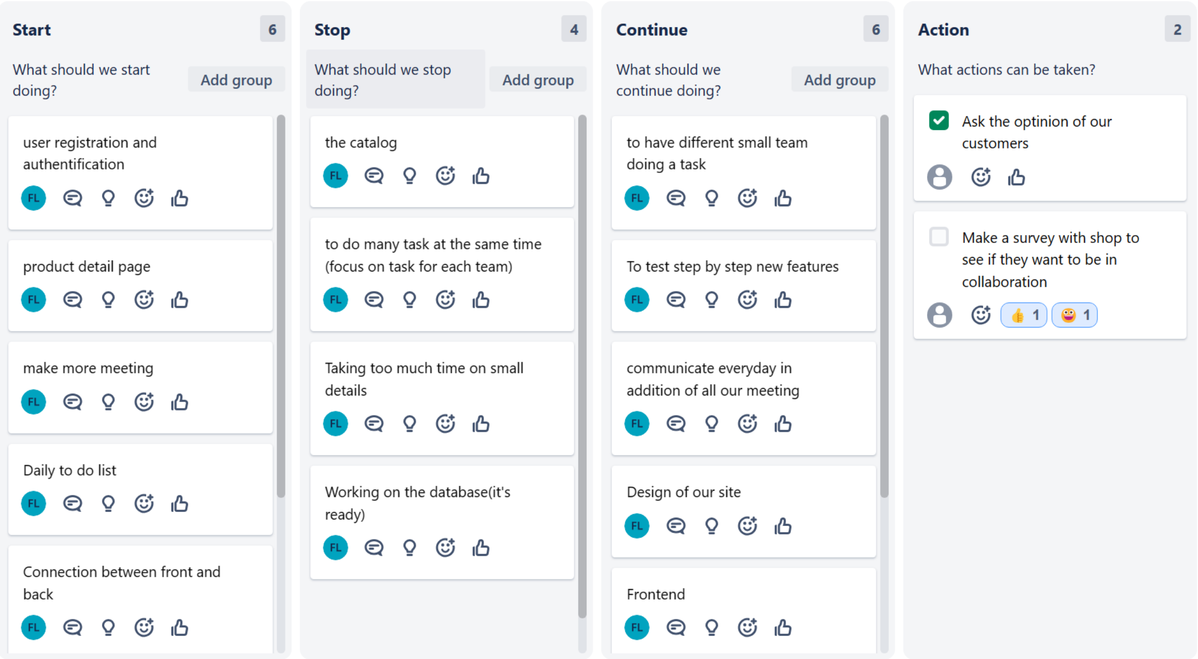


We did the user story estimation with the free online tool poker planning, using our user story. You can find our poker planning with all our three votes for each of us using this link: https://www.planitpoker.com/board/#/room/938ec05551de4946b2c07e0272dd58df

What is done?

For now, we have achieved 2 of our 4 sprints and we are in sprint 3 that stops on 9 of January, sprint 4 starts on the 10 of January and stops on 24 of January. So, as we have achieved 2 sprints, we have also done 2 retrospectives.

1st retrospective:



Start 6

What should we start doing? Add group

- user registration and authentification
- product detail page
- make more meeting
- Daily to do list
- Connection between front and back

Stop 4

What should we stop doing? Add group

- the catalog
- to do many task at the same time (focus on task for each team)
- Taking too much time on small details
- Working on the database(it's ready)

Continue 6

What should we continue doing? Add group

- to have different small team doing a task
- To test step by step new features
- communicate everyday in addition of all our meeting
- Design of our site
- Frontend

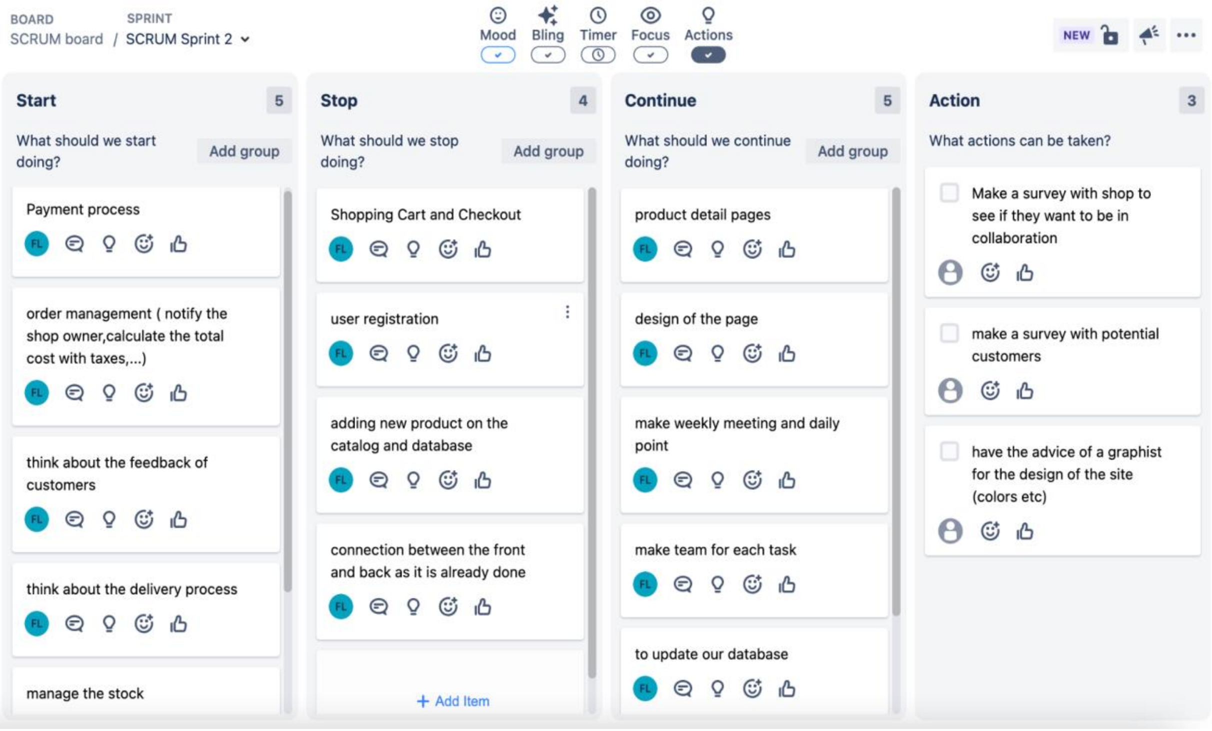
Action 2

What actions can be taken?

☑ Ask the optinion of our customers

□ Make a survey with shop to see if they want to be in collaboration 👍 1 😜 1

2nd retrospective:



SCRUM board / SCRUM Sprint 2

Start 5

What should we start doing?

- Payment process
- order management (notify the shop owner,calculate the total cost with taxes,...)
- think about the feedback of customers
- think about the delivery process
- manage the stock

Stop 4

What should we stop doing?

- Shopping Cart and Checkout
- user registration
- adding new product on the catalog and database
- connection between the front and back as it is already done

Continue 5

What should we continue doing?

- product detail pages
- design of the page
- make weekly meeting and daily point
- make team for each task
- to update our database

Action 3

What actions can be taken?

- □ Make a survey with shop to see if they want to be in collaboration
- □ make a survey with potential customers
- □ have the advice of a graphist for the design of the site (colors etc)

So far, we have achieved the following epic stories:

- Product catalog management (you can see more precisely it on our Jira) but to join this report we have our catalog.
- Develop a shopping cart system for customers to add and manage their selected items.
- Create a smooth and user-friendly checkout process.
- Incorporate delivery options.
- Set up the e-commerce platform, including selecting the appropriate software or development framework.
- Configure the platform to accommodate a diverse range of fresh fruits and vegetables.
- Implement user registration and authentication to enable customers to create accounts.
- Ensure secure and seamless login and account management functionality.

So, first we have done a catalog, made some research to find the origins of fruit and vegetables that we wanted to put on our site but also made some research for the price. We have made our SRS and BRD, our Jira etc.

Then we have created a database with SQL plus, in parallel we have created our frontend with html and Django.

We have chosen to stay in green colors to choose high quality image to stay in our graphic artistic direction. In the item page, he can see all our fruit and vegetable and choose the quantity and add to the chart then when he has all choose, he can click to go to the payment.

In the payment page the client can enter his personal information and information payment then he can also if he wants to remove an item of his chart and he can see the total price of his order and then click to confirm.

Then we have linked our database to our backend to make some API call with models on Django.

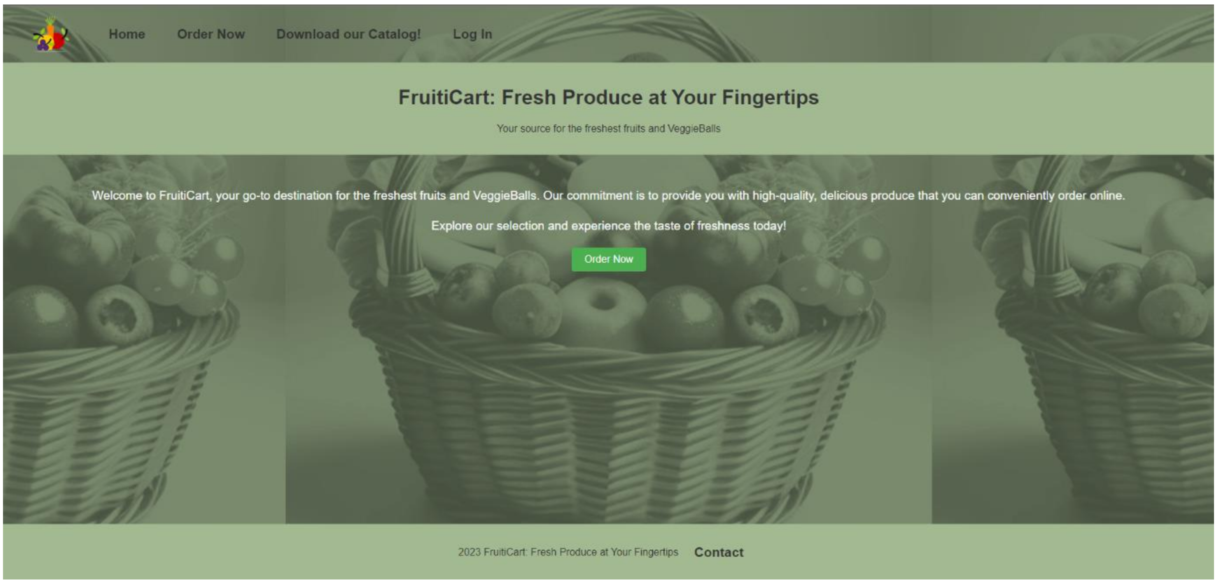
We have made a navbar where we can log in or create a new account or change information such as password and download our catalog. Customers can also contact us. Moreover, if we are logged in, our information is full on the payment page.

Quick preview

If you want to try the website by yourself, don't forget to read the README.md file to setup it.

Here's the GitHub link: https://github.com/Blxucreep/software-engineering-project.

Home page:



FruitiCart: Fresh Produce at Your Fingertips

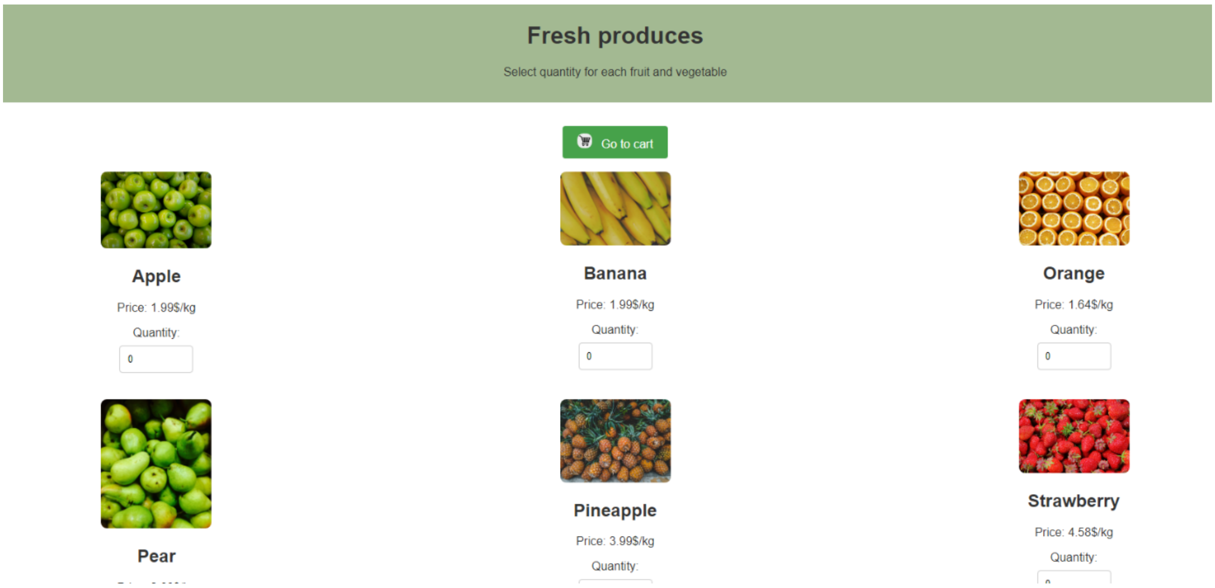
Your source for the freshest fruits and VeggieBalls

Welcome to FruitiCart, your go-to destination for the freshest fruits and VeggieBalls. Our commitment is to provide you with high-quality, delicious produce that you can conveniently order online.

Explore our selection and experience the taste of freshness today!

2023 FruitiCart: Fresh Produce at Your Fingertips

Produces page:



Fresh produces

Select quantity for each fruit and vegetable

Apple
Price: 1.99$/kg
Quantity: 0

Banana
Price: 1.99$/kg
Quantity: 0

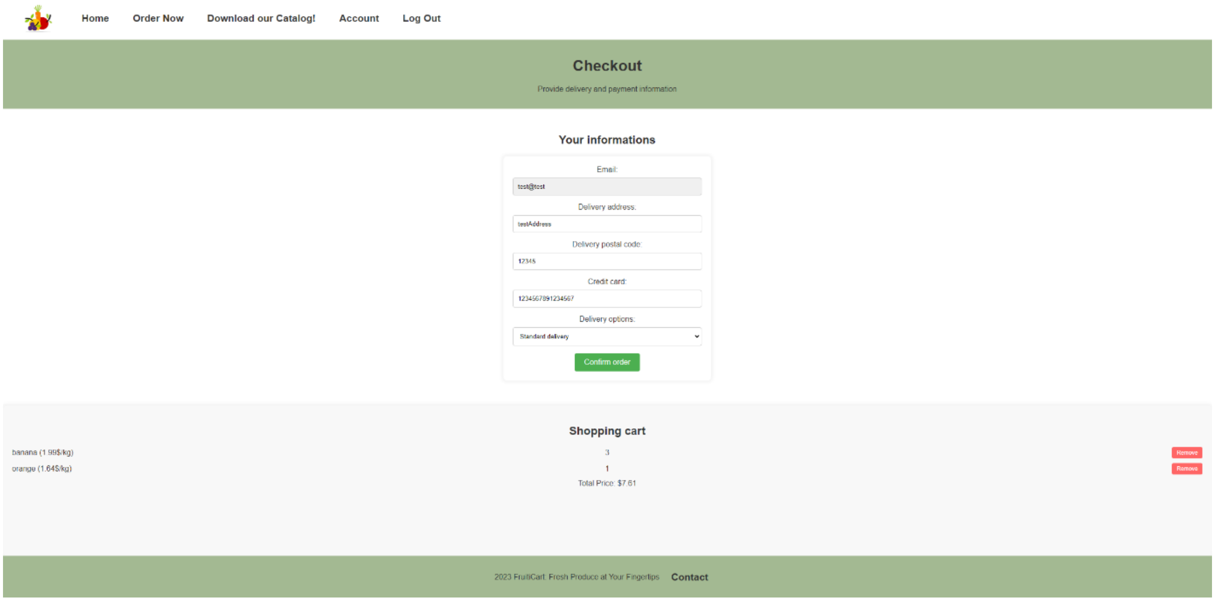
Orange
Price: 1.64$/kg
Quantity: 0

Pear

Pineapple
Price: 3.99$/kg
Quantity:

Strawberry
Price: 4.58$/kg
Quantity:

Order information page:



Home Order Now Download our Catalog! Account Log Out

Checkout

Provide delivery and payment information

Your informations

Email:
test@test

Delivery address:
testAddress

Delivery postal code:
12345

Credit card:
1234567891234567

Delivery options:
Standard delivery

Confirm order

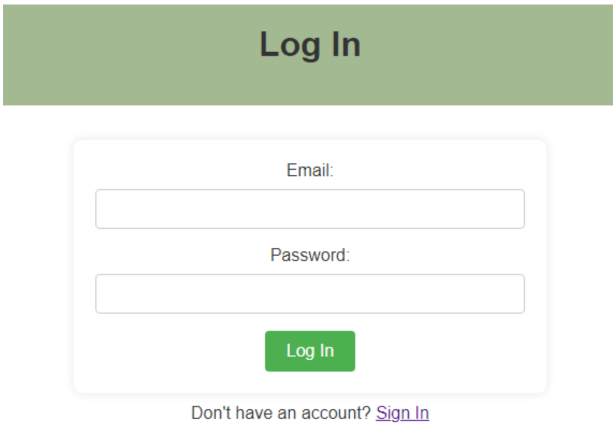
Shopping cart

banana (1.99$/kg)	3	Remove
orange (1.64$/kg)	1	Remove

Total Price: $7.61

2023 FruitCart: Fresh Produce at Your Fingertips Contact

Login/Signin page (with functionality that prevents inappropriate input):



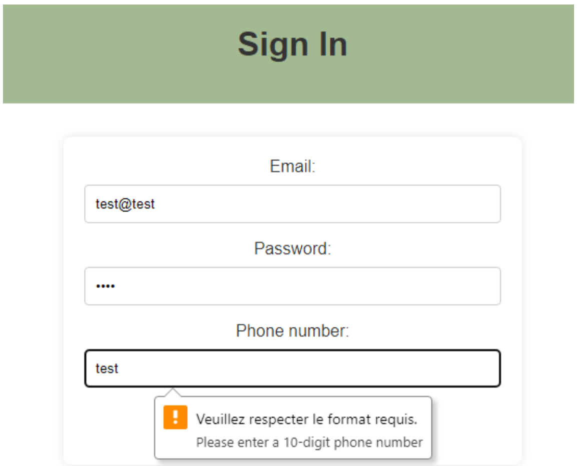
Log In

Email:

Password:

Log In

Don't have an account? Sign In



Sign In

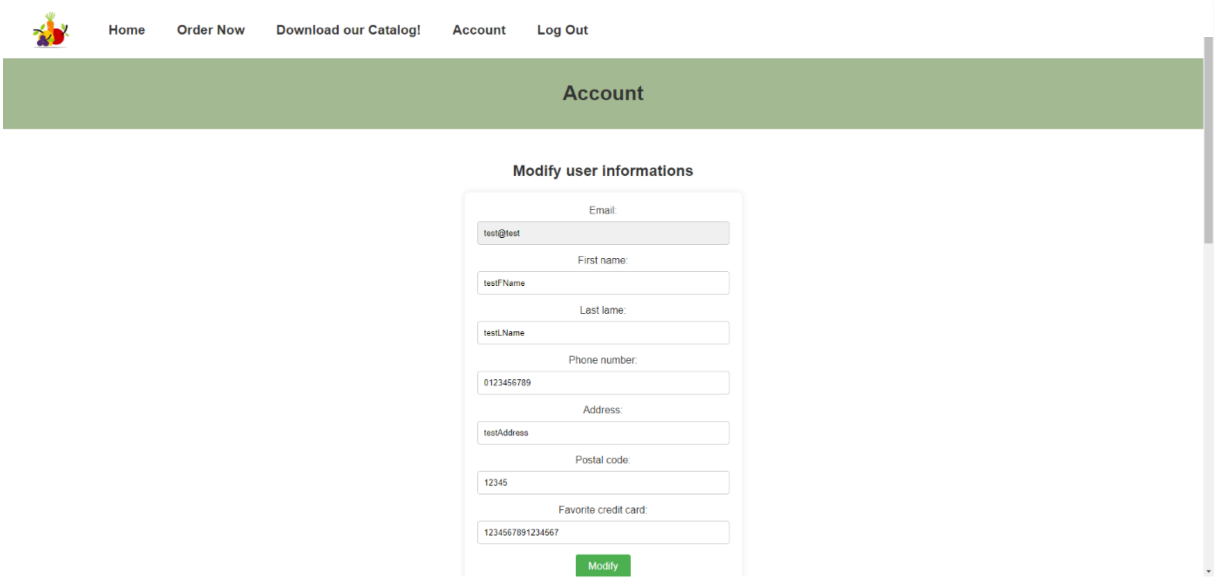
Email:
test@test

Password:
••••

Phone number:
test

Veuillez respecter le format requis.
Please enter a 10-digit phone number

Account page:



Home Order Now Download our Catalog! Account Log Out

Account

Modify user informations

Email:
test@test

First name:
testFName

Last lame:
testLName

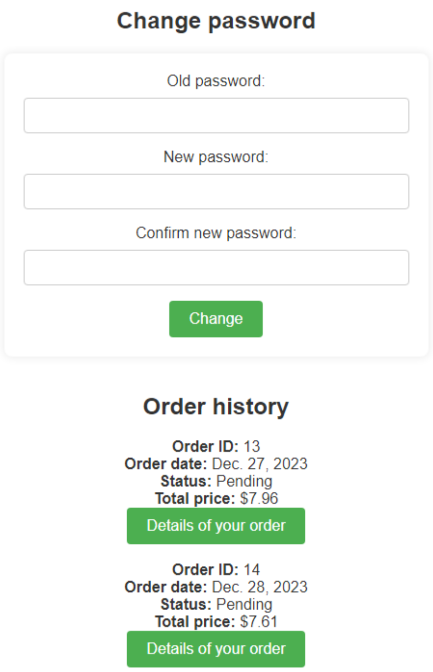
Phone number:
0123456789

Address:
testAddress

Postal code:
12345

Favorite credit card:
1234567891234567

Modify



Change password

Old password:

New password:

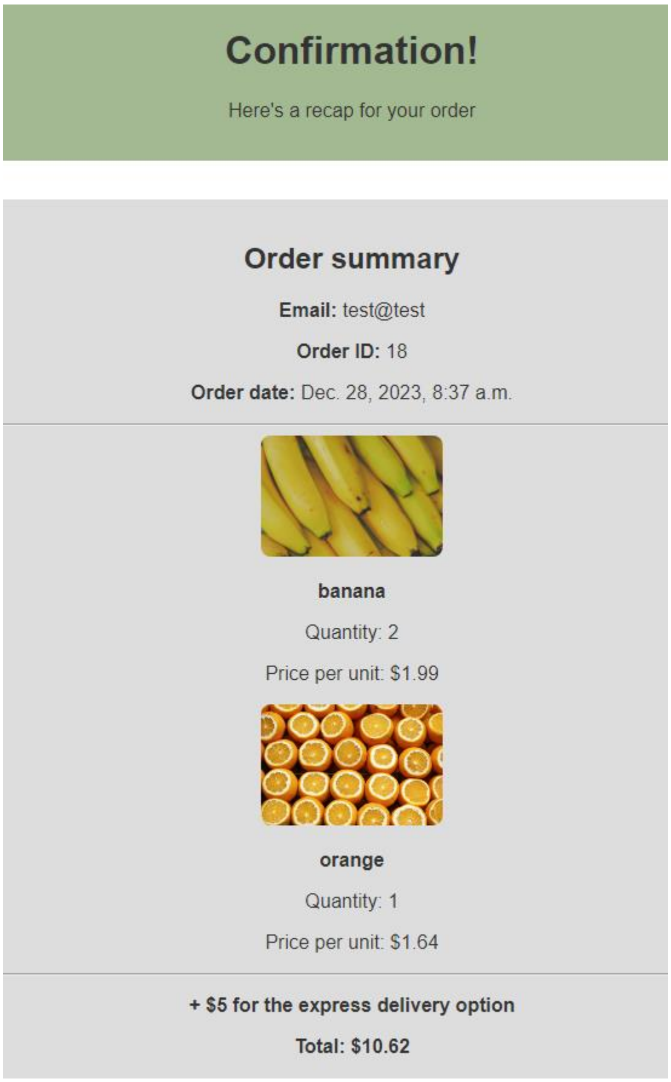
Confirm new password:

Order history

Order ID: 13
Order date: Dec. 27, 2023
Status: Pending
Total price: $7.96

Order ID: 14
Order date: Dec. 28, 2023
Status: Pending
Total price: $7.61

Confirmation page:



Confirmation!

Here's a recap for your order

Order summary

Email: test@test

Order ID: 18

Order date: Dec. 28, 2023, 8:37 a.m.

banana

Quantity: 2

Price per unit: $1.99

orange

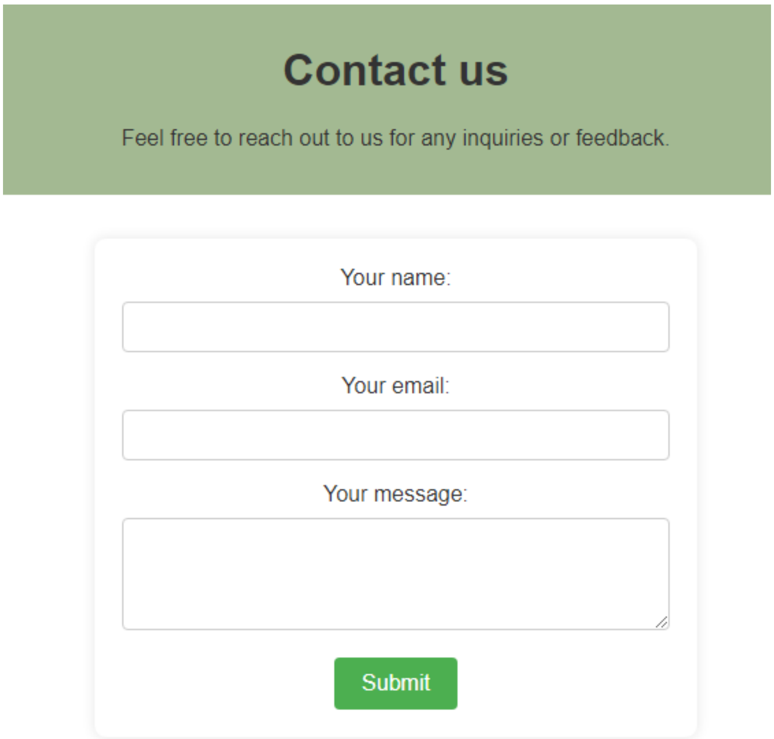
Quantity: 1

Price per unit: $1.64

+ $5 for the express delivery option

Total: $10.62

Contact page:



Contact us

Feel free to reach out to us for any inquiries or feedback.

What could have we done?

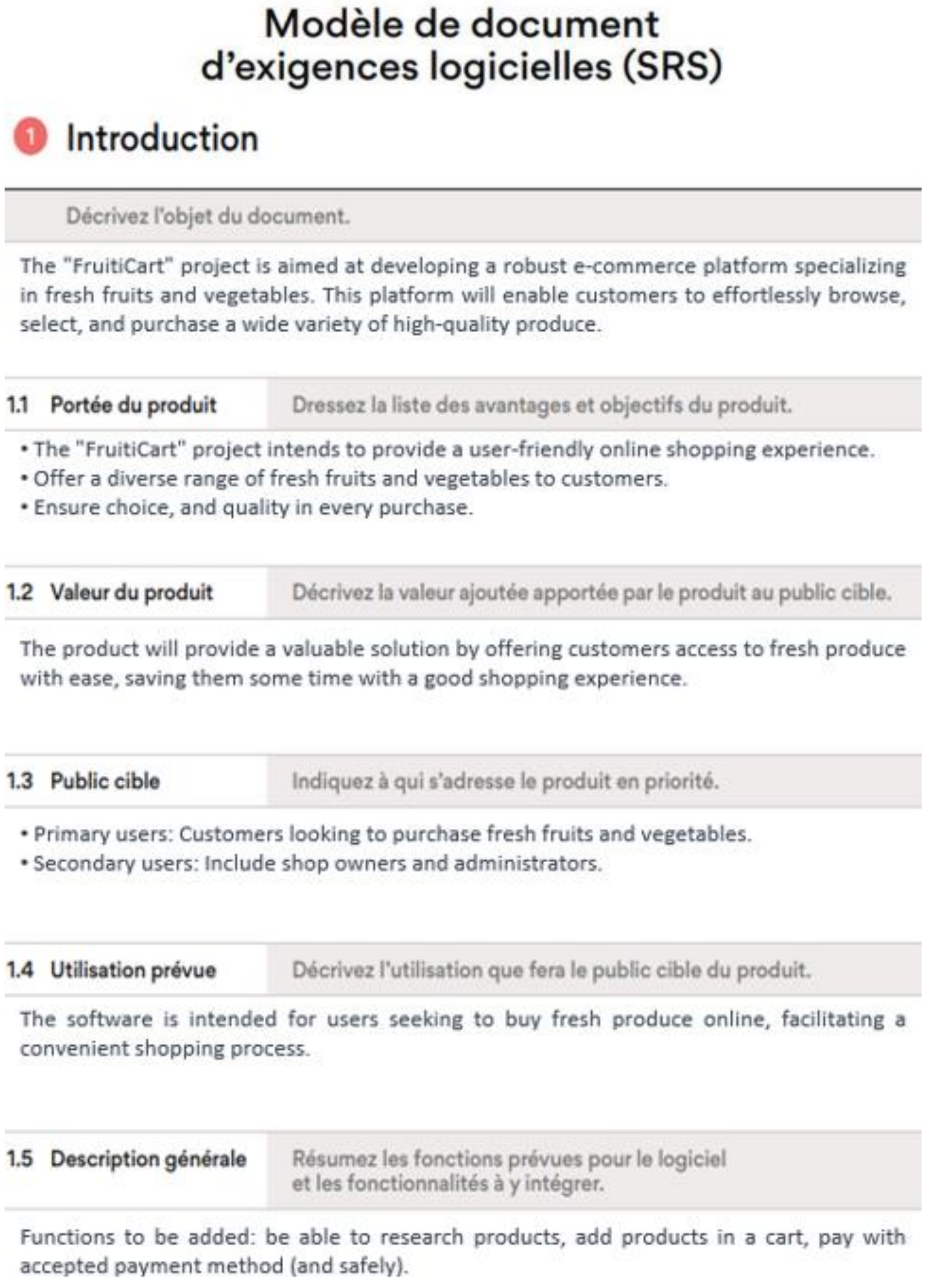
To continue this project, on sprint 3 we will do the order management, so we will build a system to handle customer order, notify the shop owner of new order and provide detail to prepare the order, calculate the final cost with all the taxes and include a management of the stock. Then we will do the payment processing, so include a secure payment method, include various payment method (CB, Paypal, etc.), ensure data encryption, collect, and process payment detail from customers. We will provide nutritional information and information about certification products.

In sprint 4, we will create the delivery and pickup part: set up a delivery system that calculates delivery fees based on location, establish a network of pickup locations with details on availability, allow customers to select their preferred delivery address or pickup point and provide any specific instructions. We will do also order tracking and notifications: provide customers with order status tracking and notifications throughout the order lifecycle, include estimated delivery or pickup times in notifications, collect customer contact details, including name, phone number, and email address for order confirmation and notifications.

Conclusion

In this project we have used our learning about software engineering (Jira, BRD, SRS, etc.) but also other learning about how to implement a user-friendly website linked to a database and how to work in team on a coding project. But we have learned also new things such as create a site with Django like to an SQL database. We have achieved 2 sprint and are very proud about what we have done for now because we have also taken advance in our expected work. Now to continue this project, we need to implement more things that we can’t do with our knowledges and resources such as secure payment method or delivery options.

SRS annex



Modèle de document d’exigences logicielles (SRS)

1 Introduction

Décrivez l’objet du document.

The "FruitiCart" project is aimed at developing a robust e-commerce platform specializing in fresh fruits and vegetables. This platform will enable customers to effortlessly browse, select, and purchase a wide variety of high-quality produce.

1.1 Portée du produit

Dressez la liste des avantages et objectifs du produit.

- The "FruitiCart" project intends to provide a user-friendly online shopping experience.
- Offer a diverse range of fresh fruits and vegetables to customers.
- Ensure choice, and quality in every purchase.

1.2 Valeur du produit

Décrivez la valeur ajoutée apportée par le produit au public cible.

The product will provide a valuable solution by offering customers access to fresh produce with ease, saving them some time with a good shopping experience.

1.3 Public cible

Indiquez à qui s’adresse le produit en priorité.

- Primary users: Customers looking to purchase fresh fruits and vegetables.
- Secondary users: Include shop owners and administrators.

1.4 Utilisation prévue

Décrivez l’utilisation que fera le public cible du produit.

The software is intended for users seeking to buy fresh produce online, facilitating a convenient shopping process.

1.5 Description générale

Résumez les fonctions prévues pour le logiciel et les fonctionnalités à y intégrer.

Functions to be added: be able to research products, add products in a cart, pay with accepted payment method (and safely).

2 Exigences fonctionnelles

Dressez la liste des exigences graphiques et de conception, des exigences pour le système d'exploitation et des contraintes du produit.

Functional requirements:

1. Product listings: The platform must display a well-organized list of fresh fruits and vegetables available for purchase, including images, product names, descriptions, prices, and availability.

2. Search and filtering: Users should be able to search and filter products quickly

3. Product details pages: Detailed product pages should be created for each item, offering additional information.

4. Shopping cart: Users should be able to add selected items to their virtual shopping carts, which will provide a total cost.

5. Checkout process: A user-friendly checkout process should allow users to choose between delivery to a specified address or pickup from a nearby location.

6. Order customization: Users should have the flexibility to adjust the quantity of each item in the shopping cart or remove products.

7. Delivery and pickup management: The system should support a delivery system and a network of pickup locations.

8. Order tracking and notifications: Users should receive order status updates and estimated delivery or pickup times.

9. Customer support: The platform should integrate communication channels (e.g., chat, email, phone) for customer support.

System constraints:

1. Operating system compatibility: The platform must be compatible with multiple operating systems, including Windows and macOS.

2. User interface design: The design of the user interface should follow modern design principles to ensure a good user experience (high-quality images and intuitive navigation menus).

3 Exigences d'interface externe

3.1 Exigences d'interface externes

Décrivez la logique qui régit les interactions entre les utilisateurs et le logiciel (dispositions d'écran, guides de style, etc.).

- **User interface requirements:**
  - The user interface should be user-friendly.
  - Good screen layouts and navigation menus.
- **Hardware interface requirements:**
  - The software should be compatible with a wide range of devices.
  - It should support both Windows and macOS operating systems.

3.2 Exigences d'interface utilisateur

Dressez la liste des appareils prévus et pris en charge par le logiciel, des exigences réseau et des protocoles de communication à utiliser.

- **Screen layout and design:**
  - The layout should be clean and organized to make it easy for users to browse products.
  - High-quality images of fruits and vegetables.
  - Product listings including name, description, price, and availability.
- **Navigation menus:**
  - The platform should offer user-friendly navigation with categories.
  - Search and filtering options should be available for users to find specific items quickly.

3.3 Exigences d'interface logicielle

Mentionnez les connexions entre votre produit et d'autres composants logiciels, notamment le framework frontend/backend, les bibliothèques, etc.

- **Frontend and backend frameworks:**
  - The software should be built using a modern and well-documented frontend framework like React for the user interface.
  - The backend should be developed using Node.js or similar.
- **Libraries and dependencies:**
  - Libraries and external dependencies should be carefully managed.
  - The software should adhere to best practices for modular and maintainable code.

3.4 Exigences d'interface de communication

Précisez les exigences relatives aux programmes de communication utilisés par votre produit, par exemple les e-mails ou les formulaires intégrés.

- **User notifications:**
  - Users should receive email notifications for order confirmations, updates on order status and feedback requests.
- **Customer support:**
  - The platform should provide communication channels such as chat, email, or phone support for users to seek assistance.

4 Exigences non fonctionnelles

4.1 Sécurité

Mentionnez les règles de confidentialité et de protection des données à respecter.

- Data privacy: The platform must adhere to strict data privacy regulations and protect customer information from unauthorized access.
- Secure payment: Payment processing must use encryption to ensure secure transactions.

4.2 Capacité

Décrivez les besoins de stockage actuels et futurs de votre logiciel.

- Storage needs: The system should be designed to meet current and future storage requirements efficiently.
- Scalability: The platform should handle increased user activity and product listings.

4.3 Compatibilité

Indiquez la configuration matérielle minimale requise pour faire tourner le logiciel.

- Multi-platform: Ensure compatibility with various operating systems and web browsers.
- Mobile responsiveness: The platform must be responsive and usable on mobile devices.

4.4 Fiabilité

Calculez la durée moyenne avant la panne dans des conditions normales d'utilisation.

- Uptime: Calculate the average time between failures under normal usage conditions to ensure system reliability.
- Fault tolerance: Implement mechanisms to handle errors and prevent system downtime.

4.5 Évolutivité

Calculez les charges maximales acceptées par votre système tout en fonctionnant normalement.

- Load handling: Define the maximum acceptable loads under normal operation.
- Continuous integration: Establish a continuous integration process for rapid feature deployment and bug fixes.

4.6 Maintenabilité

Indiquez le processus d'intégration en continu à mettre en place pour déployer des fonctionnalités et corriger les bugs rapidement.

- Code maintainability: Follow best practices for clean and maintainable code to facilitate future updates.
- Documentation: Maintain comprehensive system documentation to assist with troubleshooting.

4.7 Utilisabilité

Précisez avec quelle facilité les utilisateurs finaux devraient pouvoir utiliser votre logiciel.

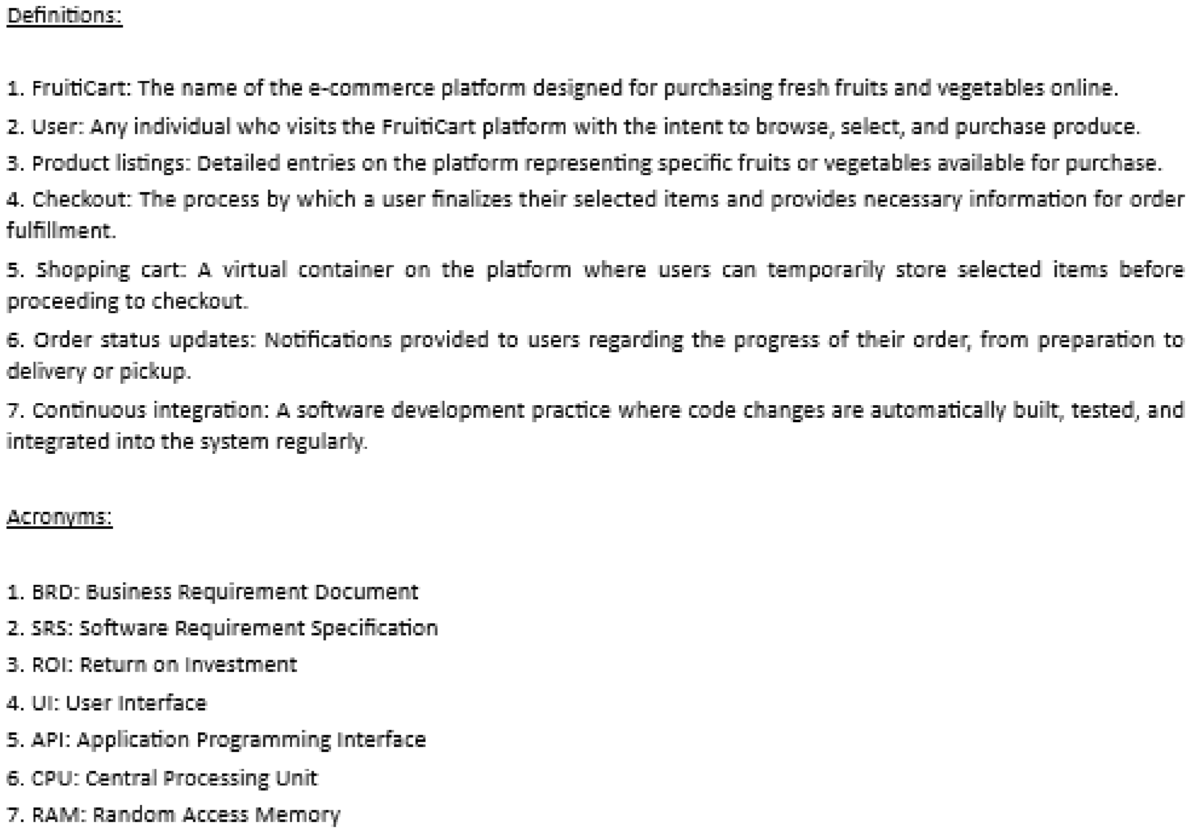
- User-friendly interface: The platform should have an intuitive user interface.
- User training: If necessary, provide training materials or onboarding assistance to help users.

4.8 Autre

Mentionnez ici toute autre exigence non fonctionnelle.

- Scalable infrastructure: Ensure that the underlying infrastructure can scale along with the platform's growth.
- Performance metrics: Define performance metrics and monitoring tools to evaluate the system's performance.

5 Définitions et acronymes



Definitions:

1. FruitiCart: The name of the e-commerce platform designed for purchasing fresh fruits and vegetables online.
2. User: Any individual who visits the FruitiCart platform with the intent to browse, select, and purchase produce.
3. Product listings: Detailed entries on the platform representing specific fruits or vegetables available for purchase.
4. Checkout: The process by which a user finalizes their selected items and provides necessary information for order fulfillment.
5. Shopping cart: A virtual container on the platform where users can temporarily store selected items before proceeding to checkout.
6. Order status updates: Notifications provided to users regarding the progress of their order, from preparation to delivery or pickup.
7. Continuous integration: A software development practice where code changes are automatically built, tested, and integrated into the system regularly.

Acronyms:

1. BRD: Business Requirement Document
2. SRS: Software Requirement Specification
3. ROI: Return on Investment
4. UI: User Interface
5. API: Application Programming Interface
6. CPU: Central Processing Unit
7. RAM: Random Access Memory