# Base-line

This document presents the baseline of the EcoCaffe project and encompasses a set of key elements that reflect the project's status at a specific moment. These elements will include the project description, functional and non-functional requirements, design, implementation, and testing. The process will be updated in a GitHub repository.

The baseline will serve as a reference point for measuring and evaluating the project's performance over time.

This document contains the following sections of the project that may be considered as potential configuration items.

1. **Funtional Requirements (FR) and Non-Funtional Requirements(NFR):** This section is responsible for managing the functional and non-functional requirements of the project, which form the foundation for the success of a software project.
2. **Design:** This section includes the system architecture, component and module specifications, as well as class diagrams and user stories. It also encompasses the user interface and database design.
3. **Implementation:** This section describes the process of translating the design into executable code. It includes coding algorithms, creating classes, and functions. Additionally, it may address setting up the development environment, version control, and code documentation to facilitate its maintenance and understanding.
4. **Software testing:** The software testing section encompasses the planning, execution, and evaluation of tests to ensure the quality and reliability of the product. We describe and propose the environments for conducting the tests.

# Funtional Requirements (FR) and No-Funtional Requirements(NFR)

In this section, we will present the functional requirements according to the actors (client, administrator, employee) and the modules. Additionally, we will specify the non-functional requirements.

## Funtional Requirements

### User Profile Module

1. **Register user:** Allow users to register on the platform by providing basic information such as name, email and password.
2. **Login:** Provide secure login features.
3. **Logout:** Offer logout functions whenever they want.
4. **Recover password:** Include a mechanism for users to recover their passwords if they forget.
5. **Customize Profile:** Allow users to create and manage personalized profiles with information such as name, address, phone number, etc.
6. **Select language:** Allow you to select the site language.
7. **Manage email notifications:** Allow users whether or not they want email notifications
8. **Manage Password:** Allow users to securely change and update their passwords.
9. **Show activity history:** Record and display a history of user activities, such as orders placed, products purchased, etc.
10. **Manage payment methods:** Allow users to add, modify or delete payment methods associated with their account.

### Products Catalog Module

1. **Add Product:** Allow users to add new products to the system, entering information such as name, description, price, quantity in stock, etc.
2. **Edit Product:** Provide the ability to edit existing product information, such as updating prices, descriptions, or images.
3. **Show Products:** Shows the products registered by the seller in the database
4. **Delete Product:** Provide the option to remove products from inventory when necessary.
5. **Search and Filter:** Implement search and filter functions to allow users to quickly find products based on specific criteria, such as category, price, etc.
6. **Product Categorization:** Allow the classification of products into categories or labels to facilitate organization and search.
7. **Stock control:** Automatically update stock quantity when a sale is made or new inventory is received.
8. **Specify preparation ingredients:** Allows you to describe the ingredients that each product has on the menu.
9. **Low Stock Alerts:** Set up alerts or notifications when the quantity of a product in stock falls below a predefined threshold.
10. **Change History:** Maintain a history of product changes, recording who made the modification, when, and what was modified.

### Dashboard Module

1. **View total revenue:** Display total sales revenue.
2. **Display number of orders:** Show order number totals.
3. **Graph profits:** Show the profits from sales made each month.
4. **Graph best-selling products by category:** Shows the best-selling products according to their category.
5. **Graph least-selling products by category:** Shows the least-selling products according to their category.
6. **View Conversion Rate:** Show current conversion rate.
7. **Real-time monitoring of system performance:** Provide real-time information on current system performance and transactions.
8. **Graph Sales and Order Summary:** Present a summary of recent sales and pending orders.
9. **Show Customer Analysis:** Include information about new customers, repeat customers, and customer behavior trends.
10. **Show Product Performance:** Show product performance, highlighting best sellers, least sellers, and new products.

### Sales Module

1. **Add Products to Cart:** Allow users to add products to cart from the product details page or from the product list.
2. **Remove Products from Cart:** Provide the option to remove individual products or completely empty the cart.
3. **Modify Quantity of Products:** Allow users to adjust the quantity of products in the cart before proceeding to checkout.
4. **View Cart Summary:** Show a clear and detailed summary of the products in the cart, including names, images, prices and quantities.
5. **Calculate Total and Subtotal:** Automatically calculate the subtotal of the purchase and show the total with taxes and shipping costs, if applicable.
6. **Save Cart Status:** Allow users to save their cart so they can return and complete the purchase later.
7. **Validate availability in Stock:** Check the availability of the products in the cart to avoid purchasing out-of-stock products.
8. **Integrate Checkout Process:** Facilitate an easy transition from the cart to the checkout process and checkout.
9. **Confirm shipping information:** Request or confirm shipping information, such as address and delivery method, during the checkout process.
10. **Apply discounts and Promotion Codes:** Allow users to apply discount codes or special promotions to the cart.

### Orders and Billing Module

1. **Place Order:** Allow users to complete the purchasing process.
2. **Select Shipping Address:** Offer the option to select a shipping address or add a new one during the checkout process.
3. **Select Payment Method:** Allow users to choose between various payment methods, such as credit cards, PayPal, bank transfers, etc.
4. **Order confirmation:** Display a clear order confirmation, including product details, shipping address and payment method.
5. **Generate Invoice:** Automatically generate a detailed invoice for each confirmed order.
6. **Send Email Confirmation:** Send order confirmation emails to users, including order details and tracking links.
7. **Manage Order Status:** Provide different order statuses, such as "In Process", "Shipped", "Delivered", etc.
8. **Track order status:** Allow users to track the status of their order in real time through a tracking number.
9. **Generate Shipping Labels:** Facilitate the generation of shipping labels for easy logistical processing.
10. **Make Refunds:** Allow users to initiate return and refund processes.

## Non-Funtional Requirements

### Design

The main colors for UI must be:

Orange: #fca311

Black: #262626

Blue: #14213d

The background of the system must have an image that identifies the cafeteria.

The Font Family must be by tags the following:

* <H1> apple-system
* <H2> apple-system
* <p> Courgette
* <a> Cantarell

### Performance

The homepage load time should not exceed 3 seconds, ensuring a fast and efficient user experience.

The application must be able to handle a simultaneous load of 5,000 users completing a transaction within a time frame of 5 minutes during peak hours.

### Availability

The system must be available 99.9% of the time, allowing a maximum planned downtime of 43.2 minutes per month for maintenance.

### Scalability

The system should be able to handle a 100% increase in the number of concurrent users during high-traffic events, such as special sales or promotions.

The system architecture should be designed to scale horizontally to accommodate a 200% increase in the user base within a three-month period.

### Security

The application must use SSL/TLS encryption to protect user-sensitive information during transmission, complying with web security standards.

The website must implement HTTPS for data protection.

### Reliability

The system must have an error rate below 0.1%, ensuring reliable operation and minimal technical issues.

### Usability

The user interface must be intuitive, with an average learning time to make a purchase of less than 5 minutes for new users.

The application should provide clear and concise error messages, allowing users to easily understand and resolve problems without external assistance.

### Accessibility

The application must comply with WCAG 2.0 accessibility standards, allowing users with disabilities to access and use the platform effectively.

All images and media must have alt text.

### Browser Compatibility

The system must be compatible with the latest versions of Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

### Interoperability

The application must integrate with external shipping services to provide users with real-time tracking information.

The application must have API endpoints that adhere to RESTful principles, allowing seamless integration with third-party services and applications.

### Maintainability

The source code must follow best development practices and be modular to facilitate updates and bug fixes.

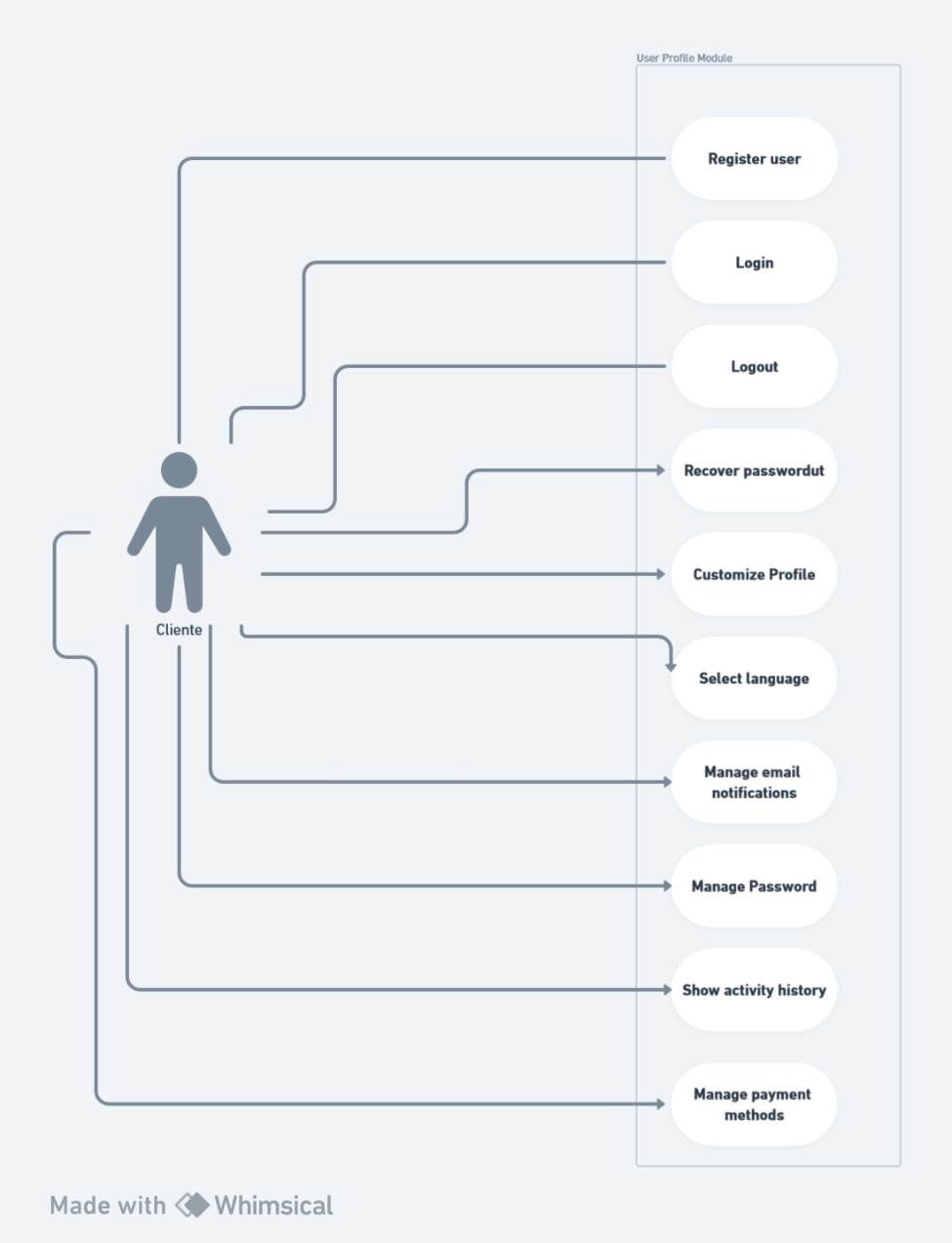
Code should be well-documented, and a version control system (e.g., Git) must be used to track changes, enabling efficient collaboration and future maintenance.

### Backups and recovery

Automatic backups of the database will be performed every night and retained for at least 30 days to facilitate data recovery.

# Design

In this section, we use UML language for system design. Detailed diagrams of components, use cases, entity-relationship diagrams, prototypes, and CRC cards will be provided.



**Figure 1 Use case diagram for the User Profile Module**

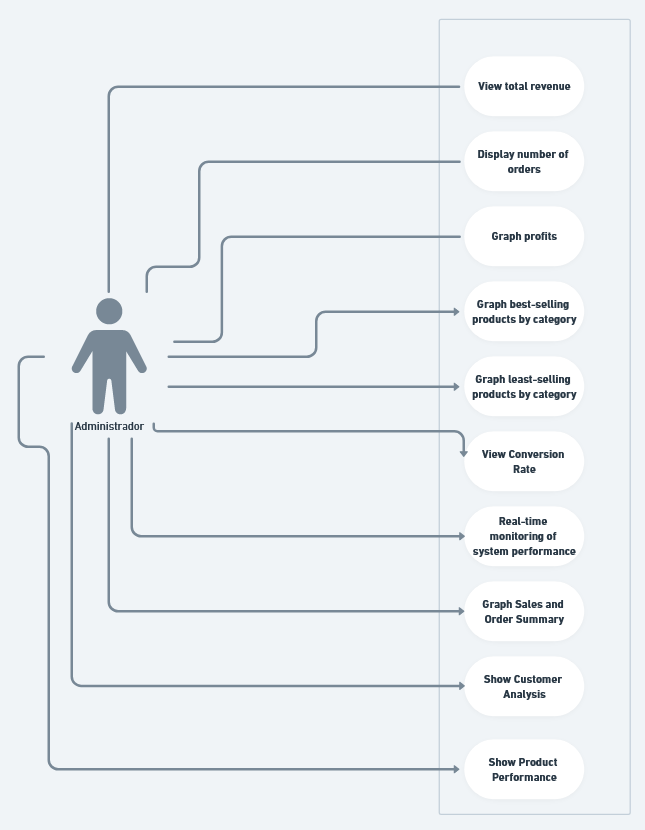


Figure 2 Use case diagram for the Administrator

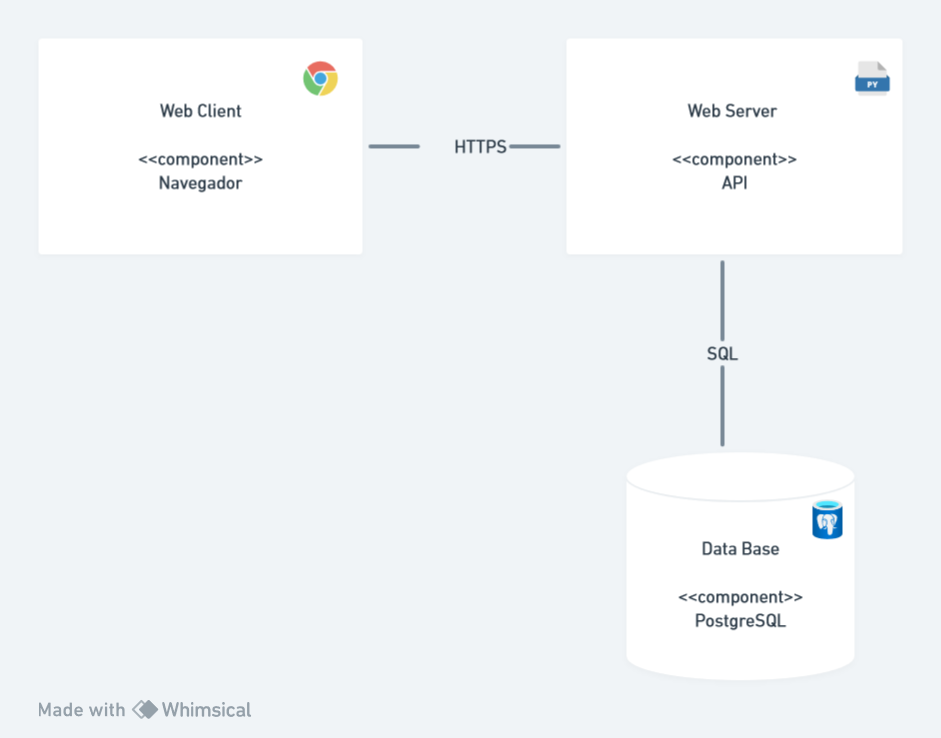


Figure 3 Component Diagram 1.0

# Changes Requests

**Change Request 1**

The clients is interesed in the platforms for collecting orders for clients (Didi, Uber Eat), so he is requesting that the System is compatible with such platforms.

**Configuration Items**

**Specific Requirements**

1. Receive the orders for the other platforms
2. Confirm the processing of the order
3. Send orders for processing in kitchen
4. Send confirmation notification to platforms
5. Send email confirmation to customer
6. Print orders for processing
7. Receive payment for the order
8. Confirm the payment (notification)
9. Display orders from other platforms on the dashboard.
10. Display sales profits from other platforms.
11. Display the quantity of products sold through partner platforms

**Orders and Billing Module:**

Is necessary add some requirements in the orders and billing module to receive customers orders from platforms as Uber Eat, Rappi and Didi and the Dashboard Module for display data of sales .

Example: Receive orders from Rappi.

Data Base: It’s necessary créate new table for orders of Uber Eats, Rappy, Didi .

Design : Is Affected, because this procces is in back-end from the platforms.

Implementation: We need a new class for the new requirements (order\_otherplatforms).

**Team group and responsability**

This change request number one has caused an impact on the duration, cost, and human resource management of the project.

The project will be extended by one more month and eight days, reaching a total of 6 months and 8 days and for its final delivery.

The project cost will increase as additional personnel will be integrated into the development team with a salary of 24780 mexican pesos.

For the analysis of a solution, it was decided to increase the salary of the most experienced analyst in the project and assign them the new change request. The salary was increased by 29500 pesos from the previous amount of 25000 pesos.

We need to add one senior developers to the development team because there are two module to modify (Orders and Billing and Dashboard).

The developer will need 30 days to complete the requirements with a salary of 24780 mexican pesos for month.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Funtional Requirements** | **Requirements specification** | **Design** | **Implementation** | **Software testing:** |
| Orders and Billing Module  (8 requirements) | 2 | 2 | 10 | 7 |
| Dashboard  (3 requirements) | 1 | 5 | 3 |
|  | 2 | 3 | 15 | 10 |
| Change request 1 | 30 days | | | |

The project cost, as estimated, increased by 28,280 pesos.

|  |  |  |
| --- | --- | --- |
| **Days** | **New Developer Senior** | **Analyst salary update** |
| Every day | 826 mx | 983.33 mx |
| 30 days | 24780 mx | Salary + 4500 mx |
| **Employee cost for the change request** | 29280 | |

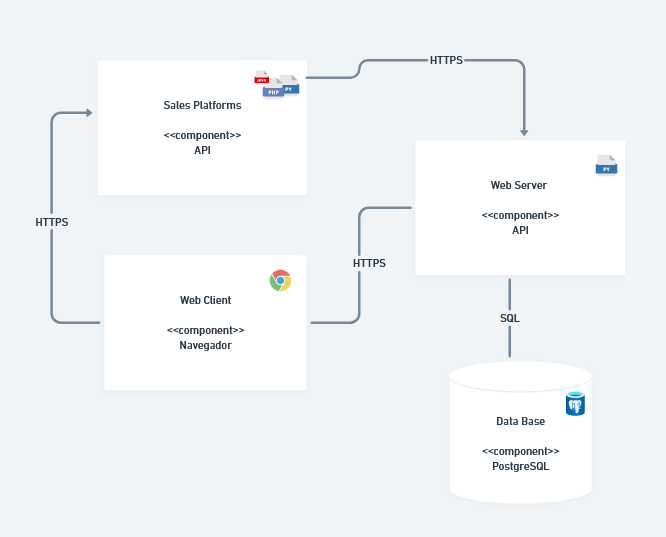


Figure 4 Component Diagram 1.1