**Z-Tech Web-based eCommerce System**

**Proponent Names:**

**Keyan andy, delgado**

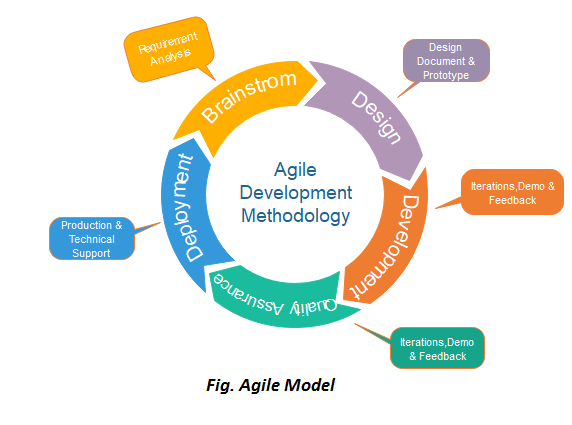
**Junggay, Juneil**

**Ogatis, Jener Kevin**

**jimbo, ulama**

**abdul, aziz marot**

Agile Development Methodology



*Figure 3.0* *Agile Development Methodology of Zyrus Tech Web App*

Agile development methodology aligns well with the dynamic and evolving nature of web-based development, enabling Z-Tech Web-based ecommerce to deliver a high-quality product that meets client expectations, adapts to changes, and provides value to end-users.

**Phases of Methodology**

**First phase of Methodology**

In the requirements analysis phase using Agile methodology, researchers follow a series of processes to gather, analyze, and document requirements. They work closely with the owner to capture requirements as user stories or items in a backlog, representing specific features or functionalities. The researchers prioritize these user stories and plan the content and duration of each sprint, which is a short development iteration. During sprints, the researchers collaborate closely with the owners, developing and delivering incremental features while gathering feedback. They hold daily stand-up meetings to provide updates, discuss challenges, and align efforts. Continuous feedback from the owner is sought to validate assumptions, make adjustments, and ensure that the developed features meet requirements. Regular integration and testing are conducted to ensure proper functionality, and at the end of each sprint, a review is held with stakeholders to demonstrate completed work and gather feedback. A retrospective meeting is also conducted to reflect on the sprint and make improvements to the development process. These processes allow researchers to effectively analyze requirements and develop a web application that meets stakeholder expectations in an iterative and collaborative manner.

**Second Phase of Methodology**

In the Design and Planning phase of Agile development, researchers focus on translating the gathered requirements into a detailed design and creating a plan for the development process. They begin by defining the overall architecture and high-level design of the web application, determining the system components and their interactions. Researchers then delve into the detailed design phase, creating specifications for each component or module, including database schema, user interfaces, APIs, and integrations. User interface (UI) design is also addressed, ensuring usability and visual aesthetics through wireframes or design mock-ups.

Technical and development planning is a crucial aspect of this phase. Researchers select appropriate technologies, frameworks, and tools while considering factors such as scalability, security, performance, and maintainability. They break down the design into smaller tasks, estimate the effort required for each, and prioritize them to create a development plan or schedule. Iteration planning further refines the plan by selecting user stories or tasks for each sprint, considering their priority, complexity, and dependencies.

Test planning is another key activity, ensuring a comprehensive strategy for testing the web application. Researchers define test scenarios, select appropriate testing techniques, and establish the necessary test environments and data. This planning phase sets the stage for subsequent development and testing phases, laying a solid foundation for efficient implementation. By focusing on detailed design, technical planning, development planning, and test planning, researchers ensure a well-structured and coordinated approach in moving forward with the Agile development process.

**Third Phase of Methodology**

In the Implementation phase of Agile development, researchers focus on the actual coding and development of the web application. They write code and develop the application based on the design specifications and user stories. Following coding standards and best practices, researchers utilize selected technologies and frameworks to implement the desired functionality. Task allocation is done to distribute work among the development team members, ensuring a balanced workload and efficient progress.

Collaboration and communication within the team are emphasized during this phase. Regular meetings and discussions help address questions, clarify requirements, and ensure a shared understanding of the tasks at hand. Continuous integration is employed, allowing the developed code to be frequently integrated into a shared repository or version control system. This practice helps identify and resolve conflicts or issues early on, promoting a smooth and cohesive development process.

Researchers follow an iterative approach, working in short development cycles or sprints. Incrementally developed features or functionalities are completed and delivered at the end of each sprint, facilitating continuous feedback and improvement. Quality assurance is integral, with thorough testing conducted during the implementation phase. Unit testing verifies the functionality of individual components, while integration testing ensures the proper interaction between different modules.

Bug fixing and refactoring are addressed promptly as identified during testing. Researchers resolve any bugs or issues and engage in refactoring to improve code structure, optimize performance, and enhance maintainability. Documentation is updated throughout the development process, capturing changes made, providing installation or deployment instructions, and updating user manuals or technical documentation accordingly.

By following these processes in the Implementation phase, researchers ensure the successful translation of design specifications into a functioning web application. Close collaboration, iterative development, and a focus on quality assurance contribute to the delivery of a high-quality and reliable end product.

**Next Phase of Methodology**

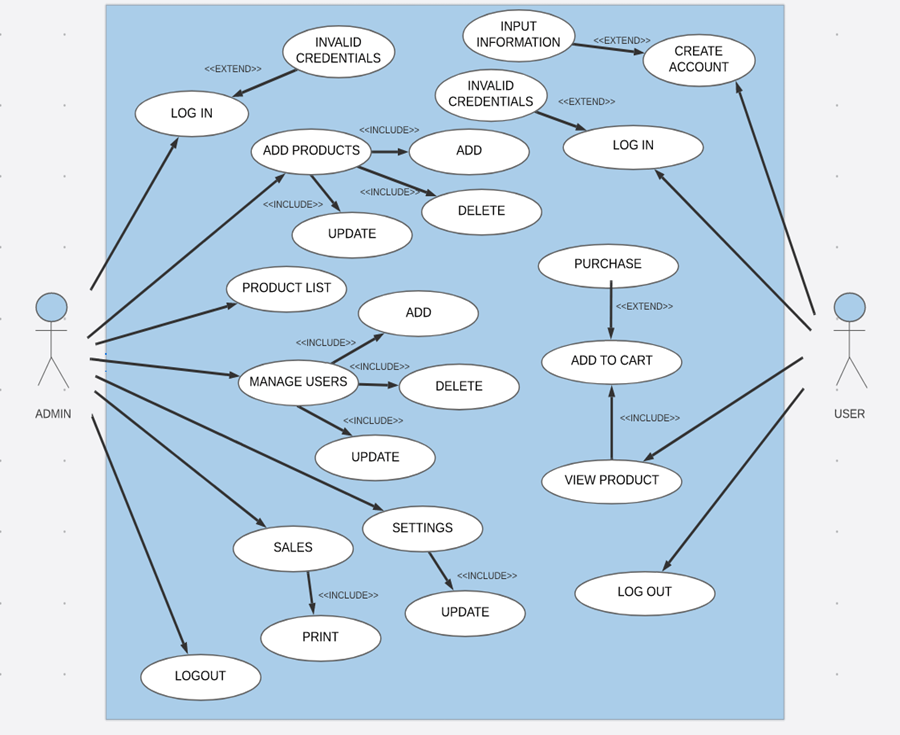
The Testing and Quality Assurance phase in Agile development focuses on thoroughly testing the developed web application to ensure its functionality, reliability, and adherence to the specified requirements. Researchers begin by creating a comprehensive test plan that outlines test objectives, scenarios, cases, and data. They execute planned tests, verifying the application's functionality, performance, and security. Defect management is crucial, with researchers tracking and prioritizing identified issues and collaborating with the development team to address and resolve them.

Regression testing is performed to ensure that fixes or changes made to address defects do not introduce new issues. User acceptance testing involves end-users or owners, validating the application's readiness for deployment through real-world usage simulations and gathering feedback. Continuous integration and continuous testing are emphasized, allowing for early issue identification and resolution.

Performance and security testing are conducted to assess scalability, responsiveness, resource usage, and overall application security. Throughout the testing phase, documentation is maintained, including test plans, cases, results, and defect records. Test reports are generated to communicate progress, coverage, and the application's quality to the owner.

By following these processes in the Testing and Quality Assurance phase, researchers ensure that the developed web application meets the required quality standards and performs as intended. Thorough testing, defect management, user feedback, and a focus on performance and security contribute to delivering a reliable and user-friendly final product.

**USE CASE DIAGRAMS**



*Figure 3.3 Use case diagram of Z-Tech web based Ecommerce System*

**SEQUENCE DIAGRAM**

View Product

Purchase

Stock Item

Add to cart

Prepare ()

Check()

Condition

Prepare ()

\*

Object

NeedsToReoder()

[check= ‘true’’]  
remove()

Iteration

Message

Self-Delegation

[needsToReoder =”true”  
new

return

A Reoder  
Item

Creation

A Delivery  
Item

[check= ‘true’’]  
new

**SEQUENCE DIAGRAM- SYSTEM LOGIN**

[Result]

Validate

Login

ValidateUser

CheckUserDatails

[User Datails]

Users

Security Manager

Login Screen

Costumer

**Types of Requirements**.

* **Functional requirements**
* **User Registration and Authentication:**

Customers should be able to create accounts with unique usernames and passwords.

The system should provide authentication mechanisms to ensure secure access.

* **Product Browsing**

The system should have a user-friendly interface with categories and filters for product browsing.

Each product should have a detailed description and high-quality images.

* **Shopping Cart**

Users should be able to add products to their shopping cart.

The system should allow users to view and modify the items in their cart.

* **Order Placement**

Customers should be able to place orders securely.

The system should generate order confirmation emails for customers**.**

* **Payment Processing:**

The system should support secure payment methods, including cash on delivery.

Integration with payment gateways for online transactions should be implemented.

* **Administrative Panel**

The system should provide an administrative panel for Z-Tech staff.

Staff should be able to manage products, view and process orders, and access customer information**.**

* **Sales Tracking**

The system should record daily, weekly, and monthly sales.

It should provide a dashboard for staff to easily monitor sales performance**.**

* **Reporting:**

The system should generate sales reports using visually appealing bar graphs for easy interpretation.

Reports should include information about products, sales, and orders.

* **Security Measures**

Implement secure data transmission and storage practices.

Protect sensitive customer information and payment details.

* **Localization**

The system should cater to customers within the specific local area as mentioned in the scope.

Implement localization features if necessary, such as language preferences.

* **Accessibility**

Ensure the web system is accessible to users with disabilities.

Follow best practices for web accessibility standards.

* **Search Functionality**

Implement a search feature to help customers quickly find products based on keywords.

* **Feedback and Reviews**

Allow customers to provide feedback and reviews for products.

Display average ratings for products based on customer reviews.

* **Non-functional requirements**
* **Performance**

The system should load product pages within 3 seconds to ensure a responsive user experience.

It should handle a minimum of 500 concurrent users without significant performance degradation.

* **Scalability**

The system should be scalable to accommodate an increasing number of products and users.

It should handle a 20% increase in products and users over the next year.

* **Reliability**

The system should have an uptime of at least 99.9% to ensure continuous availability.

It should have a reliable backup and recovery mechanism to prevent data loss.

* **Security**

Implement encryption for data transmission to ensure secure communication.

Regularly update and patch the system to address security vulnerabilities.

* **Usability**

The user interface should be intuitive and user-friendly.

The system should be accessible to users with different levels of technical expertise.

* **Compatibility**

The web system should be compatible with major web browsers such as Chrome, Firefox, Safari, and Edge.

It should be responsive and usable on various devices, including desktops, tablets, and smartphones.

* **Scalability**

The system should be able to handle an increasing number of users and transactions as the business grows.

It should scale horizontally by adding more servers if needed.

* **Data Privacy and Compliance**

The system should comply with data protection regulations, ensuring the privacy of customer information.

Implement mechanisms to handle user consent for data processing.

* **Documentation**

Provide comprehensive documentation for system administrators and end-users.

Include user manuals, system architecture documentation, and troubleshooting guides.

* **Response Time**

The system should respond to user interactions (e.g., clicks, form submissions) within 1 second.

Ensure that all user interactions provide feedback promptly.

* **Capacity Planning**

Regularly assess system capacity and plan for necessary upgrades or optimizations.

Conduct capacity planning exercises to anticipate future resource requirements.

* **Backup and Recovery**

Implement regular backup procedures for the system data.

Develop and test a robust recovery plan in case of system failures.

* **Audit Trail**

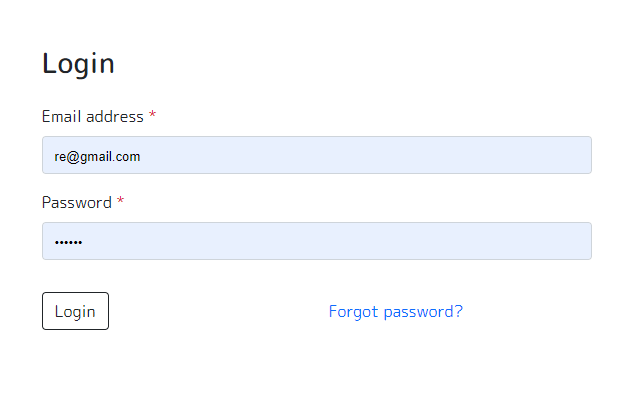
Maintain an audit trail for critical system activities, including user logins, product updates, and order processing.

Ensure that the audit trail is secure and accessible only to authorized personnel.

**User Interface**

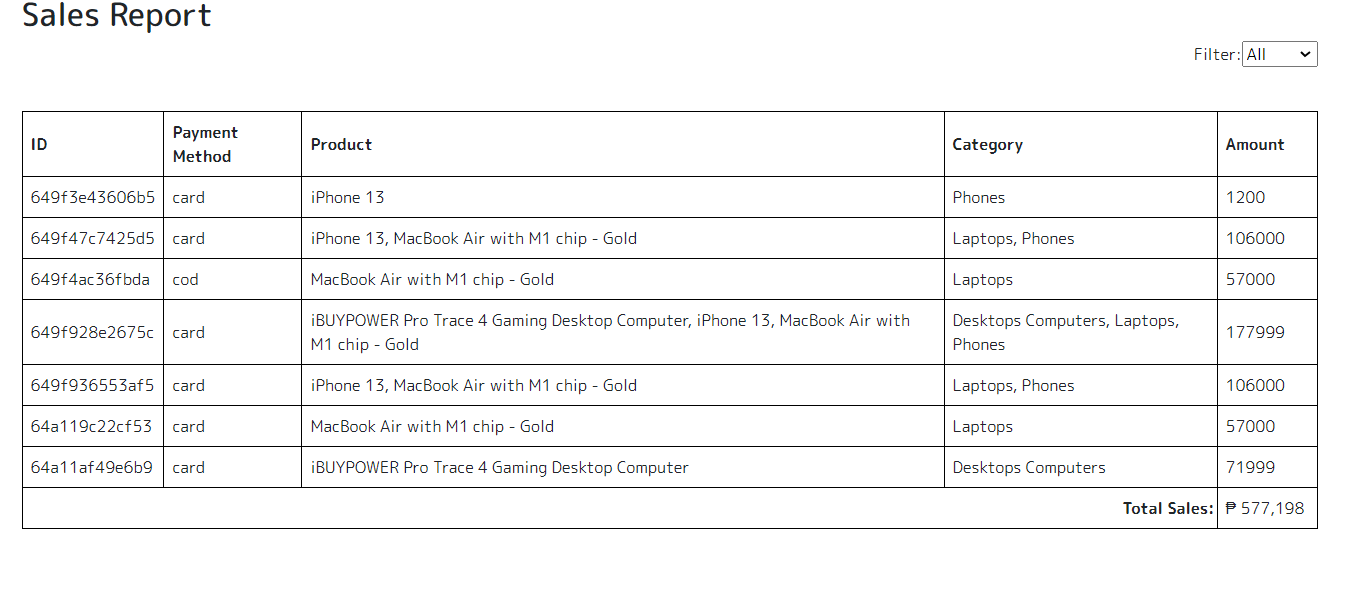
1. Think of your possible design and draw your desired interface based on the title selected in the first activity

**Admin/Customer Login Page**



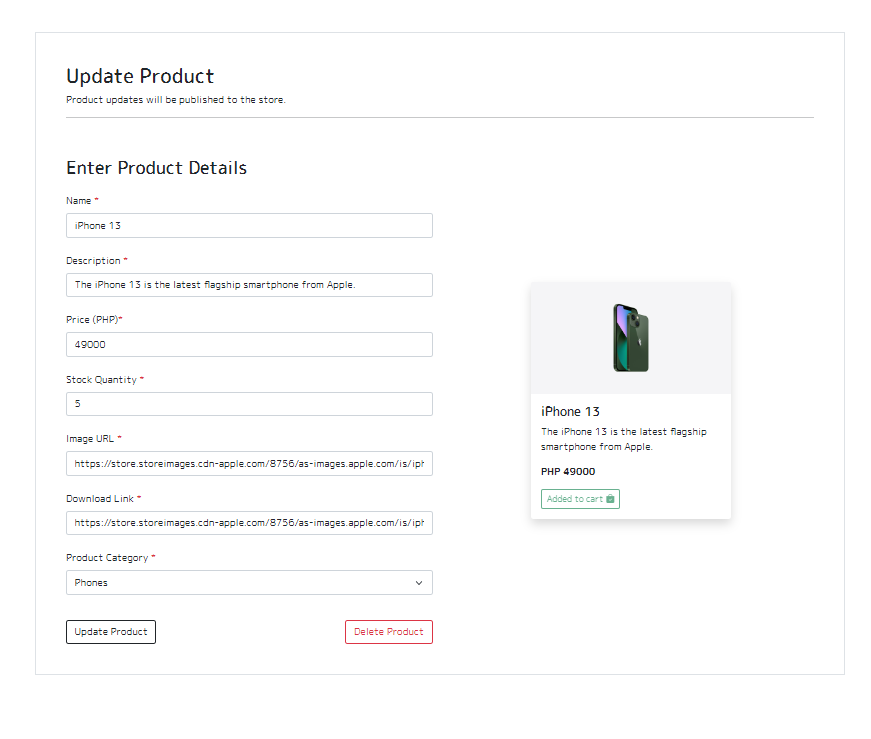
*Figure 3.4 Log in for Admin/Customer in Zyrus Tech Web App*

**Admin Sales Report**



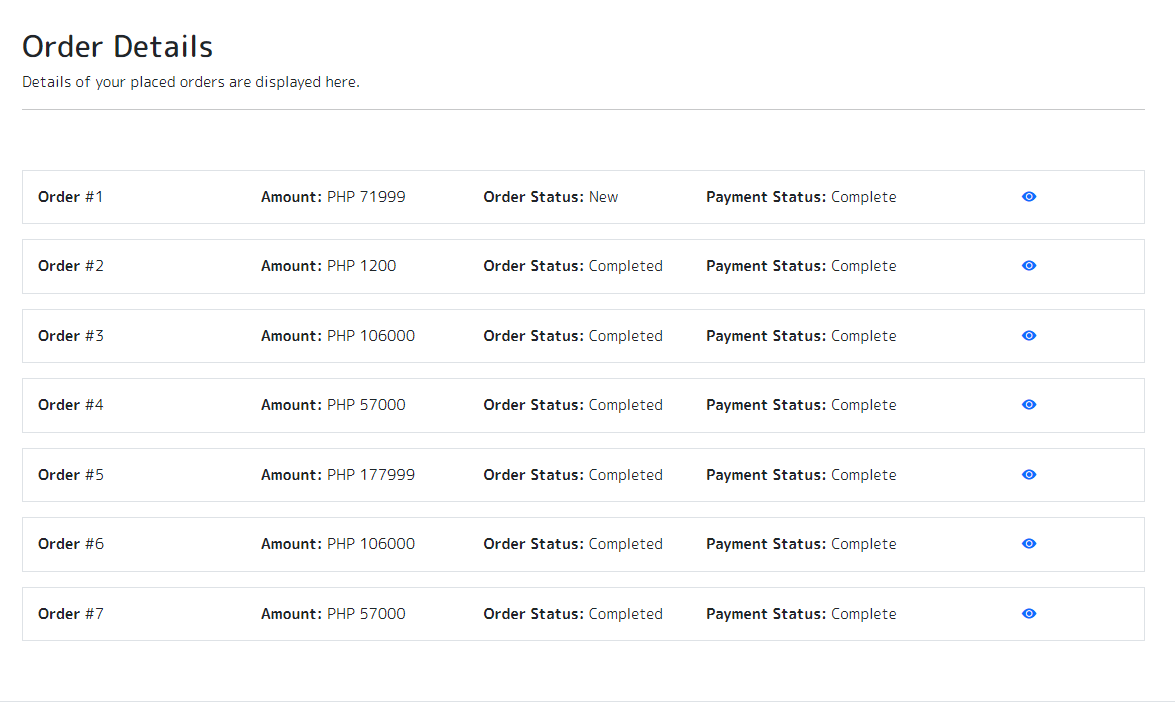
*Figure 3.5 Sales Report of Zyrus Tech Web App*

**Update Product**



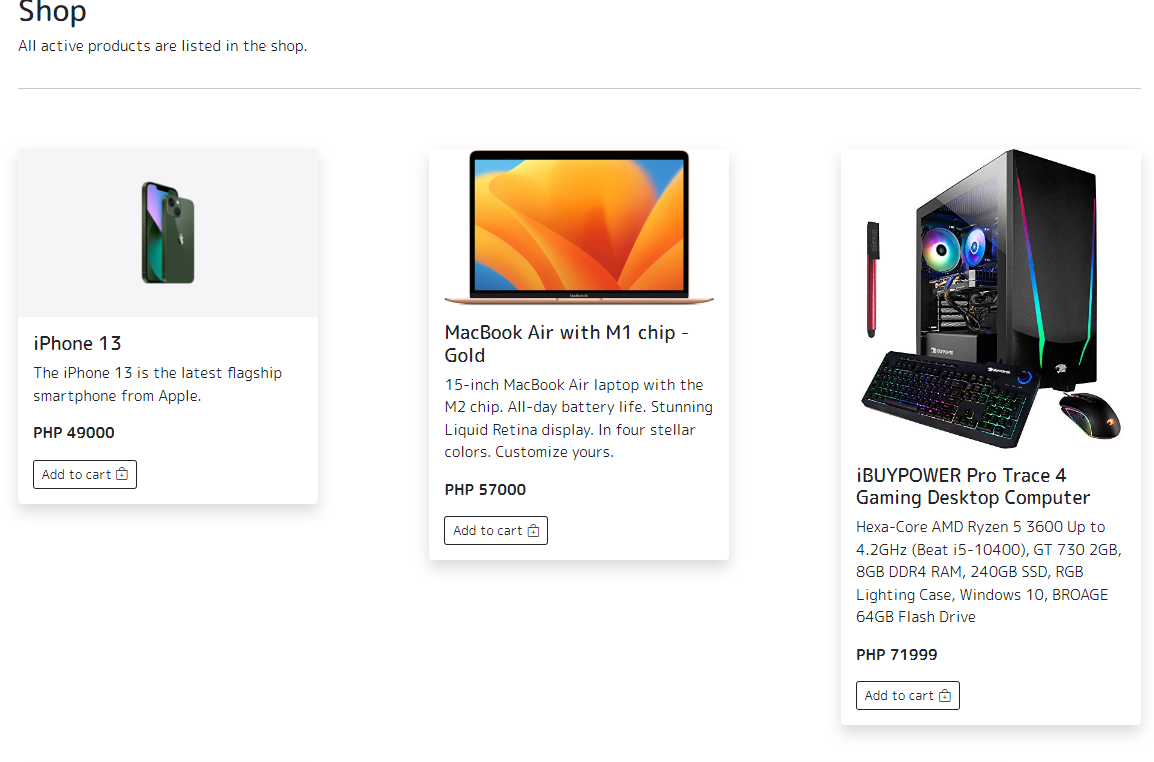
*Figure 3.6 Update Product of Zyrus Tech Web App*

**Orders**



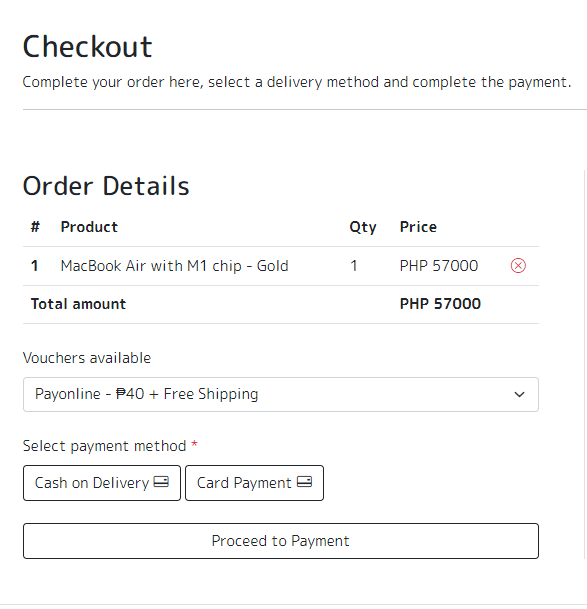
*Figure 3.7 Orders of the customer form Zyrus Tech Web App*

**Products**



*3.8 Product of Zyrus Tech Web App*

**Check Out**

****

*Figure 3.9 Check out of customers’ orders*