Text

Description automatically generated with medium confidence

PARTS4U– E-COMMERCE SITE

(WEB APPLICATION)

SOFTWARE ENGINEERING

(Semester 5, 2023)

**Students:** **Mentor:**

Lenart Kurtishi (129224) Prof. Besnik Selimi

Lum Aliji (129365)

**FACULTY OF CONTEMPORARY SCIENCES AND TECHNOLOGIES – TETOVO**

**February 2023, Tetovo**

Contents

[**1.** **ABSTRACT** 2](#_Toc133172141)

[**2.** **PROCESS MODEL** 3](#_Toc133172142)

[2.1 Extreme Programming (XP) techniques 4](#_Toc133172143)

[**3.** **SYSTEM STAKEHOLDERS** 4](#_Toc133172144)

[**4.** **REQUIREMENTS ENGINEERING** 6](#_Toc133172145)

[4.1 User requirements 6](#_Toc133172146)

[4.2 System requirements 7](#_Toc133172147)

[4.3 Functional requirements 8](#_Toc133172148)

[4.4 Non-functional requirements 9](#_Toc133172149)

[**5.** **SYSTEM MODELING** 11](#_Toc133172150)

[5.1. Use-Case Diagram 11](#_Toc133172151)

[5.2. Sequence Diagram 12](#_Toc133172152)

[5.3. Class Diagram 13](#_Toc133172153)

[**6.** **SOFTWARE DESIGN** 14](#_Toc133172154)

[6.1. Architectural Design Decisions 14](#_Toc133172155)

[6.2. Architectural Pattern 15](#_Toc133172156)

[**7.** **SOFTWARE TESTING** 16](#_Toc133172157)

[7.1 Unit Testing 17](#_Toc133172158)

[7.2 Feature testing 18](#_Toc133172159)

[7.4 System testing 19](#_Toc133172160)

[7.4 Acceptance Testing 21](#_Toc133172161)

[**8.** **GITHUB** 21](#_Toc133172162)

[**9.** **Conclusion** 22](#_Toc133172163)

# **ABSTRACT**

Parts4U is an e-commerce website designed to provide customers with a seamless experience in building their own PC based on their preferences. The website offers a variety of products such as CPU, cooler, motherboard, RAM, and other parts that can be added or removed from the customer's build. The website is developed by a team of four individuals, including two developers and two contributors, and utilizes a range of software engineering concepts and aiming to ensure a high-quality user experience.

The development process of Parts4U involved several stages, including requirements gathering, design, implementation, and testing. The system architecture was designed to ensure scalability, reliability, and maintainability. The database was designed using best practices to ensure data integrity and security. The user interface was designed to be intuitive and user-friendly, with a focus on ease of use and accessibility.

Due to the time frame constraints, the e-commerce website will be launched locally first. However, the website has been designed to be easily scalable and adaptable for future expansion. This documentation outlines the project's requirements, design, implementation, and testing, and serves as a comprehensive guide for both developers and end-users. It covers topics such as system architecture, database design, user interface, security, and performance. With Parts4U, customers can easily create their dream PC and, making the process of building a custom PC more accessible and convenient than ever before.

# **PROCESS MODEL**

The incremental Agile development process model is an iterative and incremental approach to software development that involves breaking down a large project into smaller, more manageable pieces called iterations. Each iteration involves a complete cycle of the software development process, including planning, design, implementation, testing, and delivery. The incremental Agile development process model emphasizes collaboration, flexibility, and customer satisfaction, with a focus on delivering working software quickly and continuously improving it through feedback.

For our development of Parts4U, we followed an incremental Agile development process, where we developed the website in small, incremental steps. Each iteration focused on implementing a specific set of features, such as the user registration and login functionality, the admin page, and the PC-building functionality. The incremental approach allowed our team to get feedback from stakeholders and end-users early on in the development process, allowing us to make changes and adjustments as necessary.

One of the key benefits of using the incremental Agile development process model is its ability to accommodate changes in requirements and priorities. As we received feedback from stakeholders and end-users, we were able to adjust our development plan and make changes to the website to better meet their needs. This approach ensured that the final product was tailored to the needs of our target audience.

Overall, the incremental Agile development process model was well-suited to our development of Parts4U, allowing us to deliver a high-quality, user-friendly website that met the needs of our target audience.

## 2.1 Extreme Programming (XP) techniques

In addition to the incremental Agile development process model, our team also incorporated extreme programming (XP) into the development process for Parts4U. XP is a software development methodology that emphasizes collaboration, communication, and flexibility, with a focus on delivering high-quality, working software quickly.

One of the key practices of XP is Test-Driven Development (TDD), where developers write automated tests before writing the actual code. This approach helps ensure that the code meets the requirements and is free of bugs, as well as promoting more efficient development practices. Our team used TDD extensively throughout the development of Parts4U, which helped us catch and address issues early on in the development process.

Another key practice of XP is pair programming, where two developers work together on the same computer to develop the code. This approach promotes collaboration, knowledge sharing, and code quality, as well as providing an opportunity for mentoring and skill development. Our team used pair programming throughout the development of Parts4U, which helped us work more efficiently and develop higher-quality code.

Incorporating XP practices into our development process for Parts4U helped us to deliver a high-quality, user-friendly website that met the needs of our target audience. By emphasizing collaboration, communication, and flexibility, as well as using Test-Driven Development and pair programming, we were able to work more efficiently and effectively as a team, resulting in a better end product.

# **SYSTEM STAKEHOLDERS**

A stakeholder is any individual or group that has an interest in the outcome of a project. For the Parts4U e-commerce website, there are several stakeholders who have a vested interest in the success of the website. Understanding the needs and expectations of each stakeholder group is crucial for ensuring that the website meets its goals and delivers value to its users. In this section, we will identify and discuss the various stakeholders for Parts4U, including customers, admins, developers, investors, suppliers, and competitors.

By understanding the perspectives of each stakeholder group, we can ensure that Parts4U is designed and developed to meet the needs of its users and stakeholders, while also achieving its business goals.

**1. Customers:** The primary stakeholders for Parts4U are the customers who use the website to purchase computer parts. The website's success depends on its ability to meet the needs and expectations of its customers, such as providing a user-friendly interface, a wide selection of parts, and competitive prices.

**2. Admins:** Another important stakeholder group for Parts4U are the administrators who manage the website. The success of the website depends on the ability of the admins to keep the website up-to-date, manage inventory, and ensure that orders are processed efficiently.

**3. Developers:** The development team who built the website are also stakeholders, as their work will determine the functionality, usability, and quality of the website.

**4. Investors:** Investors who have put money into the development of Parts4U are also stakeholders, as they have a financial interest in the success of the website.

**5. Suppliers:** The suppliers who provide the computer parts for sale on the website are also stakeholders, as the success of the website will impact their sales and reputation.

**6. Competitors:** Competitors in the computer parts market are also stakeholders, as the success of Parts4U will impact their market share and revenue.

Identifying and understanding the various stakeholders for Parts4U was crucial for ensuring the success of the website. By understanding the needs and expectations of each stakeholder group, the development team can build a website that meets the needs of its users, administrators, investors, suppliers, and competitors.

# **REQUIREMENTS ENGINEERING**

## 4.1 User requirements

User requirements are the high-level goals and objectives that the system or software is expected to meet from the perspective of the end user. They are typically expressed in non-technical language and describe what the user expects to be able to do with the system or software. User requirements define the context of the system and provide a clear understanding of the user's needs and expectations.

User requirements are the foundation for the development of software, and they should be elicited from the stakeholders who will use the system. The process of defining user requirements involves identifying the goals and objectives of the system, the tasks that the user wants to perform with the system, and the conditions under which the tasks will be performed. It also includes identifying any constraints or limitations that may affect the design and development of the system.

The development team must work closely with the stakeholders to ensure that the user requirements are clearly defined and understood by everyone involved in the project. This involves active communication, collaboration, and feedback to ensure that the user requirements are relevant, accurate, and complete.

User requirements can be prioritized based on their importance and the impact they have on the system's overall performance. This helps to ensure that the most critical needs of the user are addressed first, and that the system meets the user's needs in a timely manner. User requirements are critical to the success of the system or software, as they define the purpose and scope of the system and guide the development process from start to finish.

In other words, user requirements are the needs and expectations of the website's end-users. Some examples of user requirements for this website include:

* Easy-to-use interface that allows customers to navigate the website and build a custom PC with ease.
* Ability to create an account and save their preferences for future visits.
  + Access to a wide range of computer parts to choose from.

## 4.2 System requirements

System requirements are a set of specifications that define the technical details and capabilities of the system or software. These requirements describe how the system or software should behave in order to meet the user requirements and functional requirements. System requirements define the hardware, software, and network components needed to support the system or software and ensure that it performs optimally.

System requirements include both functional and non-functional requirements. Functional requirements describe the behaviour and features of the system or software, while non-functional requirements define its performance, reliability, usability, and security. System requirements also cover the scalability and compatibility of the system or software and its ability to interact with other systems and software.

System requirements are usually developed after the user requirements and functional requirements have been defined. They are developed through a process of analysing the functional requirements and determining the technical specifications needed to support them. System requirements are important for the development team as they provide the technical guidance for developing the system or software.

System requirements can be classified into two categories: mandatory requirements and optional requirements. Mandatory requirements are those that must be met in order for the system or software to function properly. Optional requirements are those that are desirable but not critical to the system's functionality. System requirements should be prioritized based on their importance and the impact they have on the system's overall performance.

System requirements must be clearly defined, verifiable, and testable. They must also be documented and communicated to all stakeholders involved in the development process. The development team must ensure that the system requirements are met during the design, development, and testing phases of the project.

System requirements are the technical specifications that are required for the website to function properly. Some examples of system requirements for Parts4U include:

* Server hardware and software requirements.
* Database requirements.
* Third-party software and library requirements.
* Security and data protection requirements.

## 4.3 Functional requirements

Functional requirements are specific and detailed descriptions of what the system or software should be able to do. They describe the behaviour of the system or software under specific conditions and the functions or features it should provide to the user. They can be categorized into different functional areas such as data entry, data processing, calculations, outputs, and system management.

Functional requirements serve as a basis for software design and testing. They are typically expressed in the form of use cases, scenarios, and user stories that detail how the system should respond to different inputs or events. Functional requirements help developers understand the user's needs and how the system should respond to those needs. They also help to ensure that the software is developed to meet specific requirements and that it provides the intended functionality.

The process of defining functional requirements involves identifying the features and functions required by the user and defining the inputs, processes, and outputs required for each function. The requirements must be clearly defined, testable, and verifiable. The development team must work closely with the stakeholders to ensure that the functional requirements are properly defined and understood by everyone involved in the project.

Functional requirements can be prioritized based on their importance and the impact they have on the system's overall performance. This helps to ensure that the most important features are developed first, and that the system meets the user's needs in a timely manner.

Functional requirements describe the features and capabilities that the website must have in order to meet the needs of its stakeholders. Some examples of functional requirements for Parts4U include:

* User registration and login functionality.
* Product catalog management functionality for admins.
* Build PC functionality to allow customers to select computer parts to build their own custom PC.
* Order management functionality for admins.

## 4.4 Non-functional requirements

Non-functional requirements are a set of specifications that define the performance, reliability, usability, and security aspects of the system or software. Unlike functional requirements, non-functional requirements do not describe the behaviour or features of the system or software, but rather the qualities or attributes that make the system or software usable and efficient.

Non-functional requirements are important because they ensure that the system or software meets the user's expectations for performance, reliability, usability, and security. They also help to ensure that the system or software is scalable, maintainable, and meets the regulatory or industry standards.

Non-functional requirements can be classified into several categories, including performance, reliability, availability, scalability, maintainability, security, usability, and compatibility. Each category defines a specific aspect of the system or software and the level of performance or quality required.

Non-functional requirements must be measurable, verifiable, and testable. They must also be clearly defined and communicated to all stakeholders involved in the development process. The development team must ensure that the non-functional requirements are met during the design, development, and testing phases of the project.

Non-functional requirements are essential to the success of the system or software, as they ensure that the system or software performs optimally and meets the user's expectations. They also help to reduce the risk of errors, security breaches, and system failures.

Non-functional requirements describe the performance, reliability, and usability aspects of the website. Some examples of non-functional requirements for Parts4U include:

* Response time and page load time requirements.
* Availability and uptime requirements.
* Usability and accessibility requirements.
* Scalability and performance requirements.

The requirements engineering process for Parts4U involved eliciting, analysing, and documenting these requirements, as well as validating and verifying them to ensure that they were complete, accurate, and verifiable. The development team worked closely with some stakeholders to review and refine the requirements, and to ensure that they were aligned with the project goals and objectives. By identifying and documenting clear and concise requirements, the development team was able to guide the development process and deliver a high-quality idea, user-friendly e-commerce website that met the needs and expectations of its stakeholders.

# **SYSTEM MODELING**

We need system modeling because it helps us to better understand and visualize the behavior, structure, and components of the system or software being developed. System modeling is an important step in the software development life cycle as it enables us to identify potential issues and areas of improvement early on in the development process.

## 5.1. Use-Case Diagram

Use case diagrams are a type of system model that describe the system's functionality from the user's perspective. They help to identify the actors (users, systems, or external entities) involved in the system and the tasks they perform. Use case diagrams are often used to define the system's requirements and to communicate with stakeholders about the system's functionality.

Diagram, schematic

Description automatically generated

## Sequence Diagram

Sequence diagrams are a type of system model that describe the interactions between objects in the system over time. They show the sequence of messages exchanged between objects in response to different inputs and events. Sequence diagrams are useful for modeling the dynamic behaviour of the system and for identifying potential issues related to performance, concurrency, and synchronization.

Diagram

Description automatically generated

Admin

Database

Website

## Class Diagram

Class diagrams are a type of system model that describe the static structure of the system, including the classes, attributes, methods, and relationships between them. They help to define the data structure of the system and to identify potential issues related to data management and storage. Class diagrams are often used to generate code and to communicate with developers about the system's structure and behaviour.

A picture containing graphical user interface

Description automatically generated

# **SOFTWARE DESIGN**

Software design is the process of defining the architecture, components, interfaces, and other characteristics of a software system or application. It involves translating the requirements and specifications of the system into a design that can be implemented by developers. The goal of software design is to create a system that is efficient, reliable, maintainable, and scalable.

During the software design phase, we also made important decisions about the back end, front end, and database architecture of Parts4U.

Logo

Description automatically generatedFor the back-end, we opted to use PHP, a popular programming language for web development, and Laravel, a widely used PHP framework. This allowed us to create a robust and scalable back-end architecture to manage the server-side logic and interact with the database seamlessly.

For the front-end, we decided to use HTML and CSS to create a visually appealing and responsive user interface that would allow customers to easily build their own PC.

Finally, for the database, we decided to use MySQL, a popular open-source relational database, to store product information and customer orders.

## 6.1. Architectural Design Decisions

Architectural design decisions involve selecting the appropriate architecture for the software system. It is important to make these decisions early on in the software development process to ensure that the system is designed in a way that meets the requirements and objectives of the stakeholders.

For this project, the following are some of the architectural design decisions:

* Separating the application into three tiers: the presentation layer (front-end), the business logic layer (back-end), and the data layer (database).
* Using PHP and Laravel as the programming language and framework for the back-end development.
* Logo

  Description automatically generatedUtilizing HTML and CSS for the front-end development to create an interactive and user-friendly interface.
* Choosing MySQL as the database management system to store and manage the data for the application.
* Implementing a Model-View-Controller (MVC) architectural pattern to ensure separation of concerns between the different layers of the application.

These architectural design decisions were made to ensure that the "Parts4U" system is scalable, maintainable, and meets the needs of the stakeholders. The separation of concerns between the front-end, back-end, and database layers ensures that each layer can be developed and maintained independently, while the use of Laravel and MySQL provides a stable and efficient platform for the back-end. The implementation of MVC architecture also ensures that the code is modular, reusable, and easy to maintain.

## Architectural Pattern

In software engineering, an architectural pattern is a general, reusable solution to a commonly occurring problem in software architecture. The architectural pattern provides a set of predefined subsystems,

their responsibilities, and the rules and guidelines for organizing the relationships between them.

A picture containing icon

Description automatically generatedFor our Parts4U e-commerce website, we have decided to use the Model-View-Controller (MVC) architectural pattern. This pattern separates the application into three interconnected components, each with a specific role:

Model: responsible for managing the data and business logic of the application

View: responsible for the presentation layer of the application, which handles the user interface and user input

Controller: responsible for handling user requests, coordinating the interactions between the model and view, and controlling the flow of data

Using the MVC pattern in our software design ensures a clear separation of concerns and modularity, making it easier to maintain, test, and scale our application.

# **SOFTWARE TESTING**

For our project, we are aiming to use a combination of manual and automated testing to ensure the quality of our software. We are planning to created test cases based on user requirements and functional specifications and execute them manually to identify any issues or bugs.

In addition, we will use PHPUnit, a popular PHP testing framework, to automate our unit tests for the back-end code. This will allow us to catch bugs early in the development process and ensure that the code is functioning as expected.

We also will conduct integration testing to ensure that all the components of the system are working together as intended. This involve testing the communication between the front-end and back-end.

Finally, we will perform acceptance testing to ensure that the system meet the expectations of our stakeholders and end-users. We will create test cases based on user stories and scenarios and execute them to verify that the system fulfilled its intended purpose.

## 7.1 Unit Testing

Unit testing is was performed to test the individual components of the system. In this type of testing, each unit of code will be tested independently to ensure that it performs as expected. For example, the functionality of a specific function or class was tested using unit testing.

The code provided is a PHP unit test case class that contains three test methods:

1. A function “test\_login\_form()” - that sends a GET request to `/login` endpoint and asserts that the response status code is 200.

2. A function “test\_user\_duplication()” -that creates two `User` objects with different names and asserts that their names are not equal.

3. A function “test\_delete\_user()” - creates a `User` object using factory and then deletes it. This method always passes the test by asserting true.

Text

Description automatically generated

## 7.2 Feature testing

This is a PHP code for testing the registration functionality of a Laravel-based web application. It includes two test functions that use the Laravel testing framework.

1. Function "test\_show\_registration\_screen," sends a GET request to the /register endpoint and expects a 200 status code in response. This is to ensure that the registration page is accessible.

2. Function "test\_new\_user\_can\_register," sends a POST request to the /register endpoint with some data, including a name, email, password, password confirmation, and acceptance of terms. It then asserts that the user is authenticated and that the response redirects to the home page.

Text

Description automatically generated

## 7.4 System testing

The purpose of this project is to allow users to build their own custom PC based on the products(parts) that the administrators provide. Users are allowed to build their PC and also calculate the prices of their custom made PC through our Website, meanwhile the administrators have access to pushing products/parts in the website.

- This test will include the functionality of the website, including user interface, admin interface, the ability to build a pc, edit a pc and delete components.

- Hardware and Software Environment Used for Testing:

* The E2E testing for our PC part picker website was performed on a Windows 11 desktop computer with a Ryzen 7 processor, 16GB of RAM, and a 512GB SSD. The testing was performed on the latest versions of the web browser Google Chrome. The website was tested using a responsive design to ensure compatibility with different screen sizes and resolutions.
* We used XAMPP to host the website on a local server during testing. The website was built using the Laravel framework , PHP, and MySQL as the database management system. The website was tested for functionality, user interface, and admin interface using manual testing and automated testing tools, such as PHPUnit for unit testing and Cypress (12.10.0) for end-to-end testing.
* This hardware and software environment was selected to ensure that the website was tested in a realistic and comprehensive manner and to provide the best possible user experience.

**Test Scenarios and Cases**

* User interface: testing the user interface for usability and functionality.
* Admin interface: testing the admin interface for usability and functionality.
* Component selection: testing the ability to select different components and build a PC.
* select different components into building a custom PC
* **Executed tests:**

- E2E(End to end testing of the web on Google Chrome)

Graphical user interface, text, application, website

Description automatically generated

In addition to end-to-end testing using Cypress and automated unit testing with PHPUnit, we also performed manual testing to thoroughly check the functionality of our PC part picker website. This manual testing included a range of test scenarios and cases, such as testing the user interface for usability and functionality, testing the admin interface for usability and functionality, testing the ability to select different components and build a PC, and testing the ability to edit and delete components.

## 7.4 Acceptance Testing

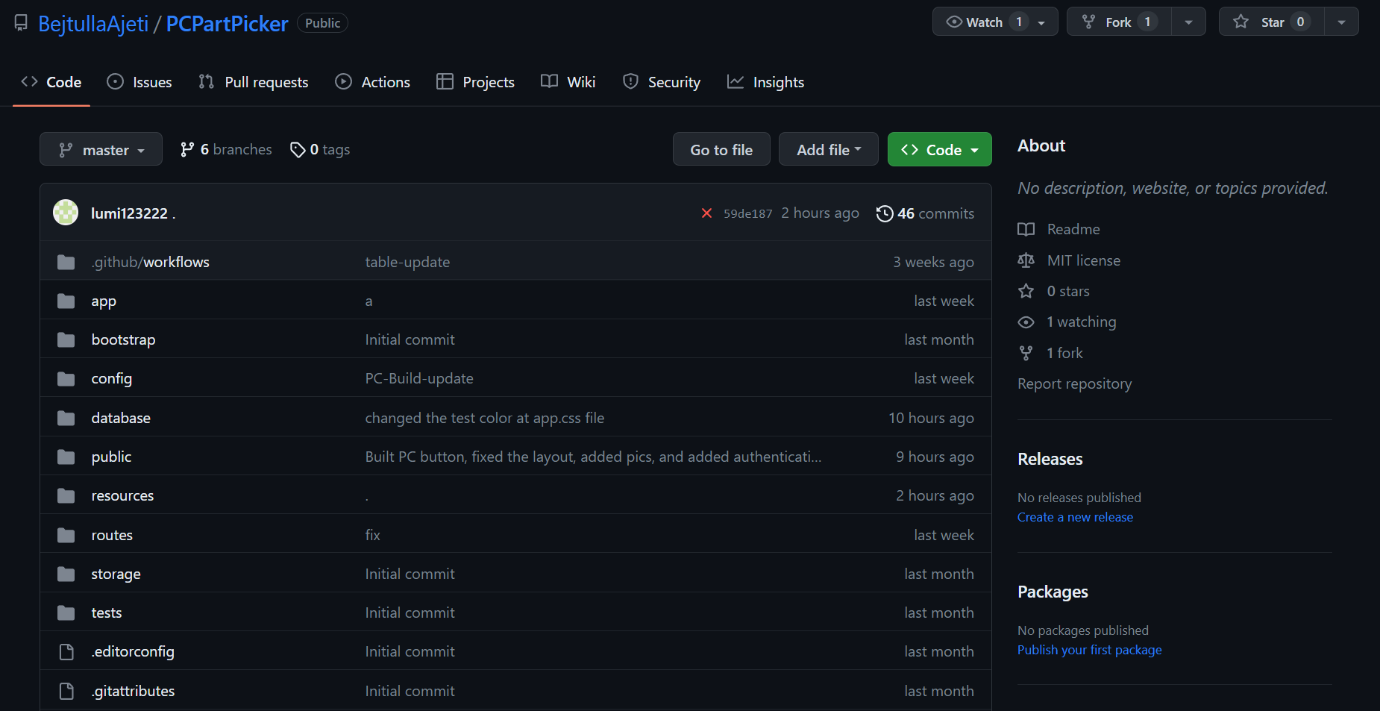
Acceptance testing is usually performed by the stakeholders or end-users of the system to ensure that it meets their requirements and expectations. In our project, acceptance testing will be carried out by a group of beta users who will be chosen to test the system before its final release. The beta users were given a set of tasks to perform on the system and their feedback will be used to identify any issues or bugs that needed to be fixed. The acceptance testing phase will be critical in ensuring that the final product will meet the needs of the end-users and will be able to deliver a satisfactory user experience.

# **GITHUB**

Github played a crucial role in our project workflow. We used it as a code repository, version control system, and collaboration platform.

Here's how we used Github in our project:

* Code Repository: We created a new Github repository for our project and pushed our code to it. This allowed us to have a central location to store our code and easily collaborate with each other.
* Version Control: Github provides powerful version control features that allowed us to track changes to our code over time. We used Git branches to work on new features or fixes without affecting the main codebase. Once we were happy with the changes, we merged them back into the main branch.
* Collaboration: Github's collaboration features made it easy for us to work together as a team. We used issues to track tasks and bugs and assigned them to team members. We also used pull requests to review and approve changes before merging them into the main branch.
* Continuous Integration: We used Github Actions to set up a continuous integration pipeline. This allowed us to run automated tests on every code change to catch any issues early on.
* Project Management: Github has built-in project management features that we used to track the progress of our project. We created a Kanban board to manage our tasks and move them through different stages of development.
* The GitHub Repository link is: <https://github.com/BejtullaAjeti/PCPartPicker>



# **Conclusion**

In conclusion, our project Parts4U is an e-commerce platform designed to provide a user-friendly experience for buying and selling car parts. We have used an incremental agile development process to ensure that our team is able to make progress on the project in a timely manner. Our requirements engineering process has helped us define the user, system, functional, and non-functional requirements necessary for a successful e-commerce platform. Our system modeling has allowed us to better understand the system architecture and user interactions through the use of use case diagrams, sequence diagrams, and class diagrams. In our software design phase, we have utilized Laravel and PHP for the back-end, HTML and CSS for the front-end, and MySQL for the database.

We have also utilized Github as our primary version control tool, which enabled us to collaborate effectively as a team, track changes, and manage our codebase efficiently.

Although the project is not yet finished, we are committed to continuing our work to refine and improve the system further. We believe that with the right effort and resources, Parts4U can become a successful and highly useful platform for both customers and vendors alike.