A logo with text on it

Description automatically generated

Software Engineering and Quality Assurance Project for E-commerce Marketplace Website

Made by:

Thaís Martin Baramarchi

Luiz Felipe Kormann

Supervising Professor:

Dr. Mohammed Nazeh Al-Imam

**Verification**

**Feasibility Study**

In line with the client specifications this ecommerce website aims to enhance product sales of this company, the gathered goals are composed of three main subjects: functions, features, and budget. The goal of this project is to create an optimal online marketplace to digitalize its own physical marketplace.

The Level 1 Specifications are made of functions, features, and budget planning, this serves as a guideline for the client and the contractor.

|  |  |  |
| --- | --- | --- |
| **Specifications Level 1 – Functional Requirements**  Costumer Specifications | | |
| **Functions** | **Features** | **Budget** |
| 1. **User Authentication** | * User registration * Login/logout * User profile | * Integration * Security |
| 1. **Product Catalog** | * Product listing * Categories * Filters | * Integration * Management |
| 1. **Payment** | * Cart * Secure checkout * Order confirmation | * Security * Integration * Notifications |
| 1. **Search** | * Product search * Sorting | * Integration * Management |
| 1. **Authorization levels** | * Define and control access permissions | * Integration * Management |
| 1. **Error Handling** | * Specifying errors and defining error messages | * Management |

The Level 2 Specifications are made of Software, Hardware, and Quality Measures, they are used to define the systems specifications.

|  |  |
| --- | --- |
| **Specifications Level 2**  Software | |
| **Operating Systems** | * OS compatibility |
| **Programming Language** | * JavaScript |
| **Frameworks and Libraries** | * jQuery.js |
| **Database Management System (DBMS)** | * Using SQL as Database |
| **Third-Party Getaway Integrations** | * Integration with common payment getaways * Shipping services |

|  |  |
| --- | --- |
| **Specifications Level 2**  Hardware | |
| **CPU** | * Minimum CPU configurations |
| **Memory (RAM)** | * Minimum DDR4 GB |
| **Graphical Interface** | * Basic graphic user processor |
| **Internet connection** | * Stable 5mbps |

|  |  |
| --- | --- |
| **Specifications Level 2 – Non-functional Requirements**  Quality Measure | |
| **Functions** | **Features** |
| 1. **Usability** | * Efficient use * Intelligible interface |
| 1. **Security** | * User data privacy * Payment security |
| 1. **Performance** | * Page load speed * Scalability |
| 1. **Availability** | * Used to help maintenance |
| 1. **Reliability** | * System uptime * Data integrity |

* **Scope and Audience**

This project will create the ecommerce website for the “contractual company”. It includes all necessary algorithms and data to provide users with the best functionalities. This includes complex data regarding trends, budgets, and user types. Thus, this scope is in line with the requirements of the customer and the users.

To properly understand the scope and audience it is necessary to define the result of the output of qualitative and quantitative studies, a sample. The samples are the result of relating the end users’ preferences and the experts’ gathered data, both are acquired through questionnaires though the expert’s data needs more conversations and meetings to define its results.

|  |  |  |
| --- | --- | --- |
| **Study Type** | **Qualitative Study (Experts)** | **Quantitative Study (End Users)** |
| **Analysis** | In-depth insights | Expressed in numbers and graphs |
| **Characteristics** | Understanding, context, subjectivity | Testing, measurement, objectivity |
| **Focus** | Content analysis | Sampling |

* **Joint Report:**

The joint report is the validation of the qualitative and quantitative analysis.

|  |  |
| --- | --- |
| **Technical Requirements** | **Operational Requirements** |
| **Software Requirements:**   * OS – Windows Server; * DBMS – SQL; * Programming Language – JavaScript; * Third-party gateway Integrations. | **Personnel:**  The number of people necessary to do the project in their assigned positions. |
| **Hardware:**   * Server processing capabilities; * Internet connection capabilities; * Storage requirements. | **Compatibility:**   * The software and the OS need to be compatible with the Hardware.   **Scalability:**   * The Hardware must be able to handle an increased user and data load. |
| **Development Tools:**   * JavaScript for Back-end and Front-end; * jQuery.js as Framework * Common Payment gateway | **Version Control Integration:**   * To keep track the updates and versions a version control tool is used. |
| **Project Performance:**   * Interface load should be less than 5 seconds; * The Payment Transactions should not take more than 1 minute; * The Database Response Time should not be more than 5 seconds; * The website must be working 99% of the time - Reliability; * The server must be accessible to all users and provide the best experience – Availability. | **Data Runtime and Recovery:**   * Regular backup schedules * Defined guidelines for data recovery   **Monitoring and Reporting:**  Realtime reporting to monitor the systems performance. |

* **List of Keywords and Recommendations**
* **User Authentication** – A secure and user-friendly login and registration process interface.
* **User Profile** – Interface area where the users can manage their accounts and orders.
* **Search Catalog** – Implementation of a digital search function for the users to search and see items.
* **Product Catalog** – Transmigrate the existing physical product catalog to an online interface.
* **Cart** – Create a digital cart system for buyers to add and manage the desired items.
* **Checkout Process** – Efficient interface to process the order confirmation.
* **Order Management** – Order tracking, order history and return management for users.
* **Payment Processing** – A secure payment gateway for transactions

**Requirement Analysis**

* **Goal**

The objective of the project is to create the best e-commerce web-application that leads the user from homepage to the payment seamlessly to the best of the abilities considering the budget.

|  |  |
| --- | --- |
| **Specification Level 1** | **Specification Level 2** |
| Functions + Features + Budget  40% of budget for **Software**  25% of budget for **Hardware**  35% of budget for **Quality** | **Software:**  Buyer access + interfaces (2 main Functions: select, remove)  Seller access + interfaces (3 main functions: select, remove, add)  Administrator access + interfaces (4 Main functions: select, remove, add, modify)  App Management (Simulations and Mapping)  **Hardware:**  CPU or GPU Architecture, Processing power, Memory, and Storage  **LoQ:**  Quality characteristics(**Availability and reliability**),  Establish normal behavior:  Soft operation (SRTS) - (response time 60 sec per session)  Hard operation (0 - 5 sec threshold) -placement of indicators (metrics) on 10 interfaces |

**Input Table:**

**User 1: Seller access type**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Functions** | **Input Parameters** | **Interface Code** | **Operation Type Measurement** | **Operation on Database** |
| **User Registration** | Name  Surname  Email  Password  Address  Company Information | #01 / #02 | Soft | Write |
| **User Login** | Email  Password | #02 / #03 | Hard | Read |
| **Inventory Search** | Search query  Filters | #04 / #05 | Soft | Read |
| **Inventory Update** | Item  Quantity | #06 | Hard | Write |
| **Sales Report** | Item  Filters | #07 | Soft | Read |

**User 2: Buyer access type**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Functions** | **Input Parameters** | **Interface Code** | **Operation Type Measurement** | **Operation on Database** |
| **User Registration** | Name  Surname  Email  Password  Address | #01 / #02 | Soft | Write |
| **User Login** | Email  Password | #01 / #03 | Hard | Read |
| **Search** | Search query  Filters | #01 / #04 / #05 | Soft | Read |
| **Cart** | Item  Quantity | #08 | Soft | Write |
| **Payment Checkout** | Quantity  Payment information  Delivery address | #06 | Hard | Write |
| **Order List** | None | #07 | Soft | Read |
| **Feedback** | Customer review | #09 | Hard | Write |

**User 3: Administrator**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Functions** | **Input Parameters** | **Interface Code** | **Operation Type Measurement** | **Operation on Database** |
| **User Login** | Email  Password | #01 / #02 | Soft | Read |
| **Search**  **(Item/User/Order)** | Item  Email  OrderID | #03 | Soft | Read |
| **Create User** | Name  Surname  Company  Email  Password  Address  Role | #04 | Soft | Write |
| **Modify User** | Email | #05 | Soft | Write |
| **Modify Order** | OrderID | #07 | Hard | Write |
| **Inventory Update** | Item  Quantity | #06 | Soft | Write |

**Output Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Users** | **Activities** | **Behavioral Measurement** | **Technical Measurement** |
| **Buyer** | Browse through catalogue  Feedback | Login rate;  Viewed rate;  Abandoned purchases rate;  Completed orders rate; | Interface Performance;  Transaction success rate; |
| **Seller** | Manage Catalog  Add or Remove Items | Measure Purchase; Completion Time;  Customer-Seller interaction; | Product listing performance;  Order processing performance;  Interface performance; |
| **Administrator** | Management (orders, users, and items) | User configuration interface time performance;  Error handling performance;  Feedback administration and response; | System uptime;  Data Backup performance;  System reset performance;  Interface performance |

**Requirement Definition**

* **Function Metrics**

|  |  |  |
| --- | --- | --- |
| **Functions** | **Non-Functional Req.** | **Metrics** |
| Registration | Security | Response time Success rate |
| Login | Security  Scalability  Performance | Response time  Logins per day  Success rate |
| Search | Correctness  Scalability  Performance | Response time  Keywords  Searches per day |
| Add/Delete to/from Cart | Correctness | Response time  Use time  Uses per day  Conversion rate |
| Payment | Security  Performance  Authentication | Success rate  Authentication success rate |
| Sell Item | Security  Correctness | Items per day  Response time |
| Feedback | Scalability | Feedback per day |

\*Availability and Reliability are non-functional requirements for all functions

* **Data Structure**

A diagram of a website

Description automatically generated with medium confidence

* **List of Classes**
* User: the father class for Seller and Buyer, consisting of the common attributes and methods from both.

**-Attributes** are name, userId, address, password, and e-mail.

**-Methods** are Register, Login and Search.

* Seller: inherited User’s attributes and methods, it should allow to insert on the catalog items and modify their own items.

**-Attribute** is company.

**-Method** is sellItem.

* Buyer: also inherits attributes and methods from User class. Will be able to buy items.

**-Methods are** selectItem**,** view and removeItems

* Item: needs to have all the information about the items inserted by the seller.

**-Attributes** are name, itemId, price, description, stock, and discount.

* Order: this class should be able to store items to help them make the purchase and payment.

-**Attributes** are userId, orderId, shippingDate, paymentStatus, itemList.

-**Methods** are viewItems, removeItems, orderInfo, paymentInfo.

* **List of Interfaces**

**Seller Interfaces:**

* #01 Homepage
* #02 Register
* #03 Login
* #04 Inventory
* #05 Item List
* #06 Inventory Update
* #07 Sales Report

**Buyer Interfaces:**

* #01 Homepage
* #02 Register
* #03 Login
* #04 Search Result
* #05 Item
* #06 Checkout
* #07 Order List
* #08 Cart Interface
* #09 Feedback Interface

**Administrator Interfaces:**

* #01 Login
* #02 System Interface
* #03 Search Result
* #04 Create User
* #05 Modify User
* #06 Update inventory
* #07 Modify Order

**Design**

**Workflow**

* **Buyer Workflow**

**A diagram of a computer

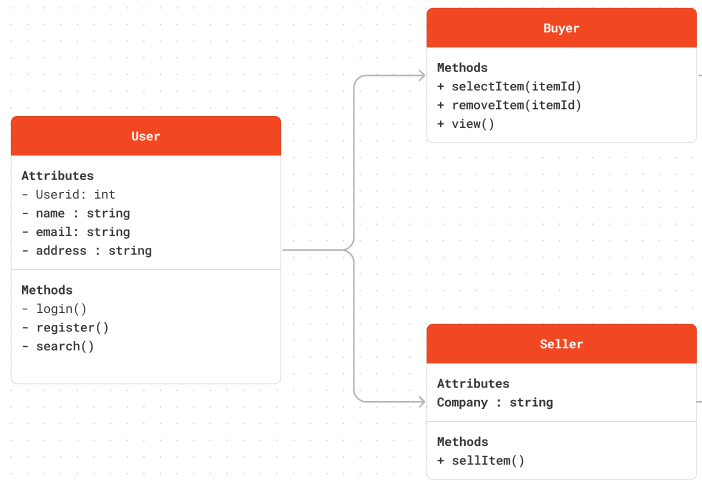
Description automatically generated**

* **Seller Workflow**

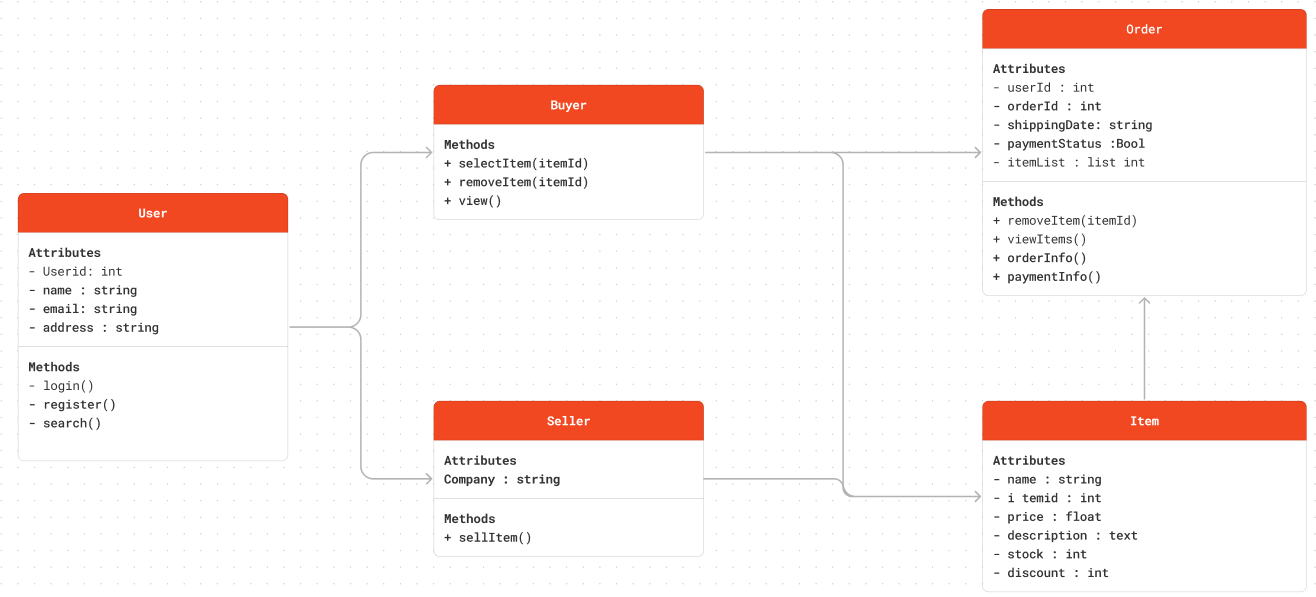
A diagram of a data flow

Description automatically generated

* **User Class:**

****

* **Class Diagram**

****

**Validation**

The process of confirming if the product satisfies the specified requirements and the user’s needs is denominated validation. It makes sure that the software works well, while following the criteria and showing the expected results. Validation is such an important part of the process because it guarantees that the product was evaluated and has achieved both quality and correctness.

**White Box** – Unit and Subsystem testing

Unit testing is a procedure that scrutinizes individual units or components within an application in isolation to verify their intended behavior. This process entails validating the correctness of small, independent sections of code, such as functions or specific segments within a function, with the aim of identifying and rectifying potential errors.

Conversely, subsystem testing is concerned with identifying errors in the code related to the interaction and mapping between functions and components. In contrast to unit testing, subsystem testing does not encompass the entire system but rather focuses on specific elements, such as features. To illustrate, in the context of a login system rewrite, subsystem testing would specifically assess the proper functioning of the login feature without conducting a comprehensive evaluation of the entire system.

**Black Box** – System integration testing

System integration testing entails the consolidation of all product features, subjecting the integrated system to comprehensive testing to identify potential errors and defects. This testing phase is imperative in ensuring the product's readiness for deployment, serving as a pivotal step prior to its introduction to end-users.

**Alpha Testing** – First Release

Neither black-box nor white-box testing incorporates real customer testing; instead, their focus is on identifying errors and deficiencies within the underlying code. Upon the successful completion of both testing methodologies, the integrated system represents the inaugural version of the product, paving the way for its potential introduction to customers in subsequent phases.

**Beta Testing** – Pre-Release (Sampling)

The beta test serves as the initial phase of product introduction to users. Despite potential customer interaction during this testing stage, the product remains incompletely released to the broader audience. This phase extends the testing process, systematically assessing the product for the emergence of new errors when subjected to user interaction. This includes evaluating aspects such as response time challenges that may arise as numerous users attempt to access specific features.

**Delta/ Gamma Testing** - Release candidate testing

Delta or Gamma testing represents the conclusive phase of validation and testing, signifying the product's official release to the entire user base. During this stage, the product is anticipated to operate seamlessly, devoid of any errors. The primary testing focus now is centered on continuous monitoring to ensure the sustained quality of service.