**Project Task:**

**Client - Bazaar Ceramics (attached file)**

Part 1: List out all regular liaison which suitable for the client.

|  |  |  |
| --- | --- | --- |
| Method | Frequency | Topics |
| Phone calls | Weekly  Urgencies | Update on project status, review of work done, planning of upcoming tasks.  Information critical to the project’s success. E.g. Insufficient budget, change in the requirements, project closure. |
| Emails | Follow up to phone calls  Regular communication channel | Sumarise and record the relevant points discussed by phone.  Any non urgent communication. E.g. bug report, UI adjustments, clarifications. |
| Instant messaging | Before phone calls  After email  Before meetings | Channel to ease other form of communication.  Can be used to ask if the other person is available for a phone call, to receive confirmation that an email has been received or to confirm an upcoming meeting. |
| Meetings | Scheduled | Sign-offs, testing, product demo and walk-throughs, hand-offs, interviews, or any topic that would benefit from an in-presence meeting. |

Part 2: Complete the Detailed Functional Requirements Report using the report template attached.

The report should include the following items:

Functional Requirements

Specific requirements

Work Breakdown Structure

Prototypes

Non-Functional Requirements

* Usability
* Legal or regulatory requirements
* Reliability
* Performance

Requirements Report

For the project

Bazaar Ceramics

Author: Alessandro Ferro

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Revised by** | **Date** | **Notes** |
| 1.0.0 | Alessandro Ferro | 03/01/2020 | First version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Functional Requirements

## Purpose

The purpose of this document is to provide a detailed description of the Bazaar Ceramics website project. It will describe the goals and the features of the website, it will details tasks and deadlines and it will present a prototype of the main pages and functionalities.

The following sections will take in analysis the quality attributes of the website, explaining ease of use, boundaries and constraints and expected performance.

## Overall description

The Bazaar Ceramics website will showcase the company’s products online, allowing customers to browse the products and place orders online.

The products will be divided in two different categories: houseware and art pieces.

The website will be accessible and usable by all the most popular browser.

The data will be stored in an in-house database.

Two different categories of user will act on the system:

* + Customers can:
    - navigate the website
    - browse the company’s products
    - search for a specific product
    - see a prodcut’s details
    - register
    - login
    - add to cart
    - checout
    - send messages to the companny.
  + System admin can:
    - login
    - create a new product
    - retrieve product details
    - update product details
    - delete products.

# Specific requirements

This section details the epected behaviour of the website in relation to the users interactions with the system.

**Customer use cases**

* Navigate website



A customer access the website homepage and by clicking the links in the navigation bar can navigate to the following pages:

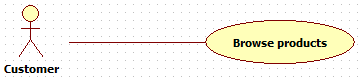
* + About us
  + Contact us
  + Products (divided by category)
  + Shopping cart.

The system redirect the user to the intended web page.

The checkout page will be accessible only through the shopping cart page.

Prerequisites: none

* Browse Company’s Products



A customer clicks on the “Categories” link in the navigation bar.

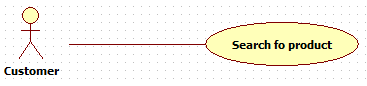
The system opens a contextual drop down menu.

The customer clicks on one of the available categories.

The system redirects the customer to the relative products page.

Prerequisites: The database is up and running.

* Search for product



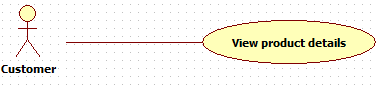
A customer types something in the search box in the navigation bar and press enter.

The system looks for products where the product details contain the keyword typed by the user.

The website populate the page with the products retrieved. If no products are found, a message is output to the user.

Prerequisites: The database is up and running.

* View product details



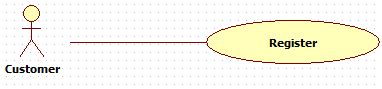
A customer can click on a specific product to see the product details.

The system redirects the customer to the prod

uct detail page.

Prerequisites: the customer is in a page that displays the products (Home page, Products page). The database is up and running.

* Register



A customer needs to register in order to be able to purchase products.

A customer click on the register button in the navigation bar.

The system opens a registration form.

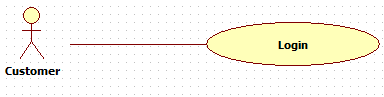
The customer enters the required information, then clicks the register button.

The system check if the infomration enter are valid:

* + Valid 🡪 The system add the new customer to the ddatabase
  + Invalid: The system output an error message to the user.

Prerequisites: the email used for the registration has not already be used for registration. The database is up and running.

* Login



A customer clicks on the login button in the navigation bar.

The system opens the login form.

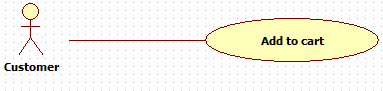
The customer enters email and password, then clicks the login button.

The system checks if the email exists.

* + Email doesn’t exist 🡪 The system output an error message to the user.
  + Email exists 🡪 The system checks is the password entered matches the email:
    - Password doesn’t match 🡪 The system output an error message to the user.
    - Password matches 🡪 The system starts a session for the user.

Prerequisites: the customer is already registered. The database is up and running.

* Add to cart



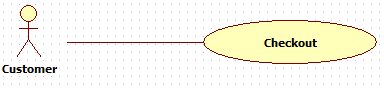
A customer click the add to cart button in the product details page of a specific product.

The system add the product the customer shopping cart.

Prerequisites: the customer is logged in. The customer is on the detail page of the product.

The product is available. The database is up and running.

* Checkout

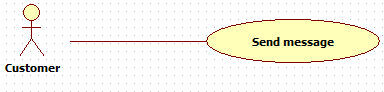


A customer clicks the checkout button in the Shopping Cart page.

The ystem redirects the customer to the payment page.

Prerequisite: the customer is logged in. The customer is on the Shopping Cart page. There are products in the shopping cart. The database is up and running.

* Send message



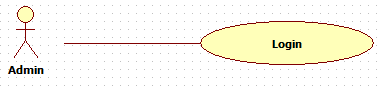
A customer fills the contact form in the Contact Us page and clicks submit.

The system sends themessage to the company’s email.

Prerequisites: The database is up and running.

ADMIN use cases

* Login



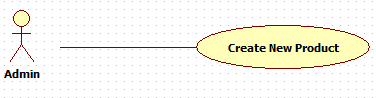
An admin clicks on the login button in the navigation bar, enter email and password and clicks login.

The system checks if the email exists.

* + Email doesn’t exist 🡪 The system output an error message to the admin.
  + Email exists 🡪 The system checks is the password entered matches the email:
    - Password doesn’t match 🡪 The system output an error message to the admin.
    - Password matches 🡪 The system starts a session for the admin and opens the admin dashboard.

Prerequisites: the admin credentials are already in the database. The database is up and running.

* Create a new product



An admin clicks on the New button in the admin dashboard.

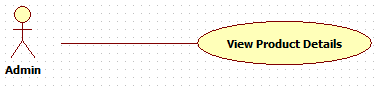
The system opens the Create New Product form.

The admin enters the fields requested and clicks on the Create button.

The system create a new product and add it to the database.

Prerequisites: The admin is logged in. The database is up and running.

* View Product Details

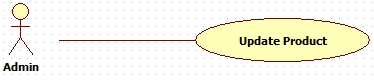


An admin clicks on the View button next to a product.

The system opens a view with the products details.

Prerequisites: the admin is logged in. The database is up and running.

Update Product



An admin clicks the update button next to a product.

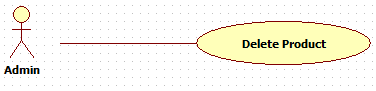
The system opens an edit product form.

The admin edits the fields and clicks the Update button.

The system update the data in the databse and confirm tha changes to the admin.

Prerequisites: the admin is logged in. The database is up and running.

* Delete Products



An admin clicks the Delete button next to a product.

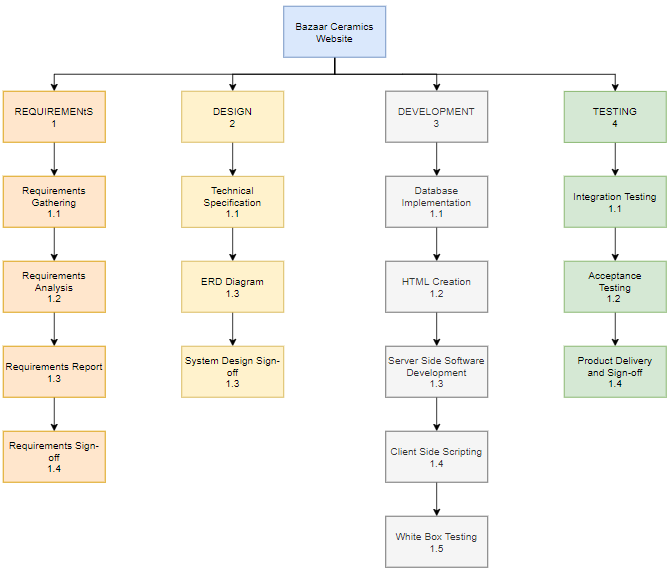
The system opens a confirmation box.

The admin clicks Confirm.

Prerequisites: the admin is logged in. The database is up and running.



# Work Breakdown Structure



# Prototypes

Please refer to the attached files ‘bazaarceramics\_prototype’

The prototyping strategy adopted for this project is that of the evolutionary prototype.

A partially working version of the website has been built to give an idea of the look and feel of the product.

This version has not to be considered definitive and its purpose is to serve as an “hands on” tool to stimulate discussion with the stakeholders. It is a point of reference that should help to better understand the shape, characteristics and functionalities of the final product and therefore, better define the client’s requirements.

The prototype consists of two different parts:

1. Customer facing: the pages that will be accessible by the customers of Bazaar Ceramics. The pages have been created to simulate the functionalities defined so far.

A customer is allowed to:

* + Register
  + Login
  + Navigate the web pages
  + Browse products
  + View products details
  + View the shopping cart
  + Add products to the shopping cart
  + Send a message to the company
  + Log out

Checkout functionalities and payment forms have not been implmeneted because they have not been defined yet.

1. Admin facing: An admin is allowed to access the admin dashboard. At the current stage only the UI prototype is available and it consists of a single page on which the required information are loaded.

The admin can view:

* + List of users
  + Products catalogue
  + Sale statistics

There have been also provided graphic elements to update, delete and add products and customers, but their functionalities have not been implemented yet.

The admin can also log out from the system.

When accessing the prototype as a customer, it is possible to register a new user with the registration form. The credentials created will be valid until the end of the session.

However the admin cannot register and the appropriate credentials must be used.

The credentials are available in the ‘\_CREDENTIALS.txt’ file inside the root of the project itself, or in the ‘users.php’ file inside the ‘includes’ folder. There is one Admin and three customers credentials available.

To properly discuss the prototype, a meeting with the client should be organized to allow the developer to walk-through the various aspects of the product and receive, discuss and record any feedback from the relevnt stakeholders.

# Non-Functional Requirements

## Usability

The goal of the project in terms of uability is to guarantee that the website works efficiently, in an ituitive manner and with the least possible workload on the customer part.

Efficieny is represented by the workflow to accomplish tasks within the website, and is measured in terms of time required for each task, nimber of tasks a user can complete independently and by the number of successful transactions.

Intuitiveness is reflected by how easy the system is to use by a someone approaching it for the first time.

The workload refers to the effort a user has to put in in order to accomplish a task, and it should always be reduced to the minimum possible.

To guarantee that the above characteristics are implemented, the website should take advantage of a minimalistic design philosophy. Navigation should be kept simple, with as many parts of the website in reach of the user at any moment, without requiring the user to follow articulated paths through the web pages. The only exception would be the checkout, reachable only from the shopping cart.

The web pages should also be specifically themed, with each one dedicated to serve a well defined purpose, in a clear and evident way.

A similar approach should be applied to the implementation of the application functionalities. The interface presented to the user will need to be as simple and barebone as possible. The user shouldn’t be required to put any relevant effort in learning how to use the product and any possible interaction with the system sould be made obvious by the interface.

The design should adhere to modern and well established principles, to ensure both an ‘up with the times’ look and feel, and a UI composition that is familiar to the user.

## Legal or regulatory requirements

To enable the necessary ecommerce functionalities, such as payment and delivery, the website will need to collect sensitive user data. This data come in the form of user credentials created to access the website, and personal financial information to make payments.

Bazaar Ceramics customers should be informed and agree about the collection, the modalities and the use that will be made of their data. A privacy policy should be disiplayed on the screen and should require mandatory confirmation in order to proceed and submit the information to the website.

Another aspect related to the handling of sensitive information, is security. Authentication and authorization mechanics will be implemented in the form of user credentials: a username (or email) and a password that will be stored in a database in encrypted form.

Session management will rely only on what are currently considered the best practices and both cookies and session will be designed to expire under certain circumstances (e.g. idle time).

The documentation that will be procided to the customer at the product delivery and sign-off, will clearly state the different levels of authentication and authorization and will identify who detains them.

## Reliability

To minimise the amount of maintenance required to keep the website operational, thorough testing will be executed.

The code will be tested to avoid that any major bug will be delivered in the final product. The presence of bugs is to be expected in every software product, at any stage of development, but by applying effective testing techniques early in the development life cycle, their number and gravity would be reduced considerably. It has been proven that the cost of solving a bug related issue in a software product, grows exponentially the further the bug is encountered during development. In the context of this project, testing wil start at the requirements gathering stage and will be iterated at every following phase.

When a MVP (Minimum Viable Product) will be available, stress testing will also be executed to observe the behaviour of the product in a simulated yet realistic environment.

It exceeds the scope of this project to identify and evaluate risks of physical nature that may compromise the reliability of the product.

## Performance

* **Uptime**

Uptimeshould be as close to 99.999% as possible. The strategy adopted from a development perspective is to reduce code maintenance to the minimum by adopting scalabale solutions in every aspect of development, from database design and implementation to software architecture and code.

* **TTFB (Time to first byte)**

To achive the quickest TTFB (Time to first byte) possible and keep the leading time in an acceptable frame, the following strategies will be adopted:

* + Code optimization – The efficiency of the back-end code plays a role in how fast a resource is retrieved and delivered over the network. The code delivered will not only be written with concern for its efficacy, but the development team will consider and test different approaches and different data structures to achieve maximum efficiency.
  + File optimization – the resources will be organised to reduces the load on the server during each http transaction. A single css file will be used for styling and minimal client side scripting will be included. It will also be taken in consideration the use of css sprites for the use of persistent images such as the company’s logo, backgrounds or any other image the client may wish to include. Products and users images do not fall into this category due to their dynamic nature.
  + Images optimization – images size will be reduced without compromising the quality and where possible and appropriate, svg will be preferred and html embedded.
  + CDN (content delivery network) – considering the interest of Bazaar Ceramics in expanding on international markets, the use of cdn to retrieve and deliver content from multiple locations over the world should be kept in consideration

Part 3: Prepare a client sign-off document using the report template attached.

The report should include the following items: Project Summary, Project Goal, Project Deliverables, Comments (If Any), Document Signatures

Project Name: Bazaarr Ceramics

Project Sponsor: Bazaar Ceramics

Service Owner: Kym Hayward

Project Manager: Alessandro Ferro

Document Date: 03/11/2020

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Project Summary | | | |
| **Start Date** | 25/10/2020 | **Finish Date** | 20/10/2020 |
| **Project Duration** | 15 Days | | |

|  |
| --- |
| 2. Project Goal |

* The project has been completed within the agreed deadlines.
* The product looks, performs and functions as per the agreed requirements.
* The product has been tested and has been found conform to the client expectations.
* The product is accompanied by all the required and agreed documentation.

|  |
| --- |
| 3. Project Deliverables |

* Bazaar ceramics website, consisting of all the files that constitute the complete software product. This include scripts, images and any other assets needed for the correct rendering, performing and functioning of the product.
* Test document completed with all the tests effectuated, user manual, software architecture document.

|  |
| --- |
| 4. Comments (If Any) |

|  |  |  |  |
| --- | --- | --- | --- |
| 5. Document Signatures | | | |
| Name | Description | Signature | Date mm/dd/yyyy |
| Service Owner Name | By signing this document, I acknowledge that I have received all the stated deliverables at the agreed to quality levels. |  |  |
| Project Manger Name | By signing this document, I acknowledge that I have delivered all the stated deliverables at the agreed to quality levels. |  |  |