2020

Project Proposal

A close up of some shoes

Description automatically generated

x19175108 Geneci Ui Fhatharta

x19176414 Katarzyna Casserly

x19141815 Ronan Behan

x19139306 Rory Cleary

Server Side Development

SportShop Application

Contents

[1 Project Overview 3](#_Toc53933386)

[1.1 Description of the SportShop application 3](#_Toc53933387)

[1.1.1 Major Goal 3](#_Toc53933388)

[1.1.2 Specifics Objectives 3](#_Toc53933389)

[1.2 Specific requirements of the SportShop application 4](#_Toc53933390)

[1.2.1 Functional Requirements 4](#_Toc53933391)

[1.2.2 Non-Functional Requirements 5](#_Toc53933392)

[1.3 Proposed Web Application Specification 6](#_Toc53933393)

[1.3.1 Features 6](#_Toc53933394)

[1.3.2 Navigation 7](#_Toc53933395)

[1.4 Mock-ups screens 8](#_Toc53933396)

[1.5 Attachments and Others 11](#_Toc53933397)

# Project Overview

## Description of the SportShop application

The scope of this project is to design and implement an e-commerce website for SportShop. Ruby On Rails will be the designated language used to create the application. Rails is suitable for large and small web apps, future proofed to expand easily. It is relatively quick to create a site due to its existing pre-defined configurations. There are already three main environments included: development, testing and production. Ruby On Rails is based on Model-View-Controller (MVC) architecture. It allows one to create a very responsive and flexible application.

The SportShop application will be responsive and it will automatically adjust to work on all screen resolutions and devices (desktop, laptop, tablet & phone). A relational database to store all data (products, customer, cart, order etc.) will need to be created.

SportShop will be an interactive application allowing response on request from the web browser. The primary purpose of the website will be selling products online. The application allows users to browse through the website in order to search, select and purchase products. If a user decides to purchase a product, they will first need to create an account. SportShop will store customer details (address and order details). It will facilitate safe payment processing using major credit cards, debit cards or PayPal.

SportShop exists to increase sales, work as a marketing tool in order to improve customer service and to obtain data regarding general product demand.

### Major Goal

To plan, analyse, design, develop and deploy a full web application that sell sport products for men, women and kids using Ruby on Rails as programming language and platform.

### Specifics Objectives

As a project team assigned to work on the development of a web shop project, we have many specifics objectives, under the SMART methodology. Our objectives are: Specifics, Measurable, Achievable, Realistic and Time-Bounded.

* To agree a group rule and sign a contract team.
* To list the requirements needed to design the system.
* To work well as a group and divide tasks evenly.
* To fulfil the project’s needs by being able to provide a working system at the end of the college term.
* To set out reachable deadlines for the necessary projects to be completed.
* To meet once a week to ensure each group member knows the tasks to be done.
* To set up a meeting plan, based on project spec, in order to define specifics dates needed.
* To understand all requirements stated for this project.

## Specific requirements of the SportShop application

### Functional Requirements

The SportShop application main objective is to allow a registered user to process online sport products shopping. In order to allow it the following functionalities must be developed.

* A fully functional SportShop application
* A top menu bar with home, products pages, contact us pages contained in it.
* A side navigation bar with the product pages by categories
* A database to hold information inputted into the website.
* An easy to use website.
* An easy to use menu bar.
* Non-registered user can open the website url.
* Non-registered user can navigate between products pages.
* Non-registered user cannot make a purchase.
* Registration page – The SportShop application must provide a registration page to define a username and a password to a user.
* Login page – To allow user to log into the web service at any time.
* The web service must allow user to log out.
* The web service provides a search functionality to the inventory
* Inventory – any user can visit different pages of the web application
  + Men AND Women AND Kids products pages
* Navigation can be done by:
  + Product name
  + Product model
  + Product brands
  + Product sizes
* Registered user can make purchase
* Registered user can add item to the cart
* Registered user should be able to remove item from the cart
* The web application must have a wish list
* Registered user can add item to the Wish list
* A link to the checkout should be available all the time at all products pages
* The web application must have a Payment page
* The payment page must provide a summary of what user has in their cart
* The payment page provides a total of the purchase being made
* The payment page collects user personal details
* The payment page collects user card details
* The payment page collects user deliver address OR / AND billing address details
* User can access the website in many different device (Desktop, Laptop, Mobile and Tablets)
* Registered users have access to his/her historical orders

### Non-Functional Requirements

* Ensure there is a security measure to only allow admin user access to change the content of the website pages.
* Ensure that only registered users can make purchase on the website
* Making changes to the website should be user friendly and easy to complete.
* The website should be compatible with all browsers and viewable on most technologies, for example desktops, laptops and mobile devices.
* The website should be easy for the user to use with easy navigation through the pages and consistent design and page layouts.
* The website needs to have a fast response time and processing time for example in accessing the sign-up forms to register.
* The website will be live for twenty-four hours a day, seven days a week and will be free of charge for users to use.

## Proposed Web Application Specification

Our Web Application will be implemented using the Ruby on Rails framework and will have the following pages available:

1. Home page - The homepage will be the initial focus and it will feature a navigation menu that will direct users to the part of the site they wish to visit. The purpose of this page is to introduce the visitor to the site and give an overview of the products on offer, and to display a variety of products with an objective to keep the user interested enough so that they will purchase from our application.
2. About Us page – Detailing a short history of our company and what our goals are.
3. Registration page – Enabling users to register so that they can purchase our available products.
4. Profile page - details on user, such as phone number and email address, which can be edited by user. Purchase history of user will also be provided on this page.
5. Product page - displaying inventory which can be browsed in a variety of ways and can also be searched. The user can click on some of the sample products displayed on homepage or can choose from major product categories on the main menu. User also can use search function to search site for specific item(s). The ability to choose between Men’s, Women’s and Children products will be available.
6. Contact Us page – a facility that will enable users to contact our support team through filling out a form and will have the ability to contact us with queries.
7. Check out page – displaying what the user has placed in their cart, how many items they are purchasing and the total amount due. They can also edit their shopping cart from here by adding or deleting items. When they are ready to finalise their order they will be redirected to a payment page to complete the process.
8. Thank you, page, – after a purchase is made a page displaying a thank you message and order number will be visible on the user’s screen.

### Features

We want people to find out about our web design, search engine optimisation, and accessibility services by searching the web.

The website will be accessible to search engine spiders and be coded with good on-page search engine optimisation. A website with good SEO will appear higher on a search engine's results.

Features you might find on a site with good SEO include error-free copy, clean code, and many inbound links to the page from other reputable sites. These links are usually to social media websites.

There will be a Sign-up/Login button for users to become registered displayed on every page.

This e-commerce site will allow users to log onto the site and browse the inventory by various attributes, e.g. product name, manufacturer, model, colour etc.

A user can register with the site to allow for purchases on the site. Once registered, the user can add items to his or her cart. Items can be added and removed by the user as he or she browses the inventory.

Once the user is ready to make a purchase the user is brought to a payment page where details on what is in their cart is displayed and a total price is also provided. The user can then make a payment for the purchases.

Responsive Web Design will be included in our specifications, in order to provide optimal viewing across a wide range of devices i.e. desktop computers to mobile devices.

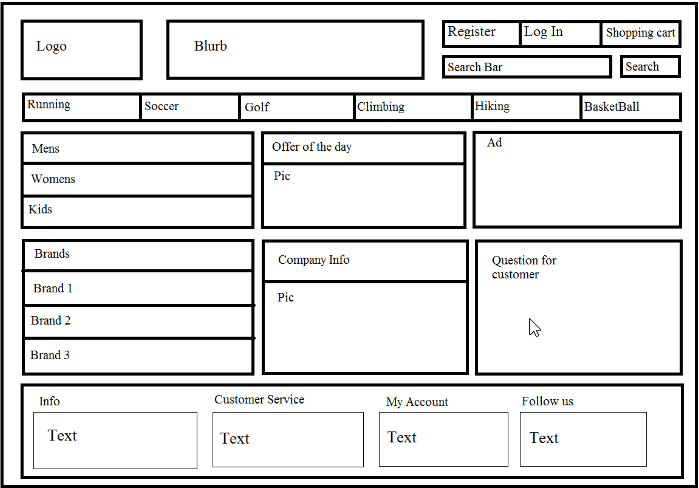
### Navigation

The navigation system will be intuitive and allow the visitor to quickly navigate the site starting from any page.

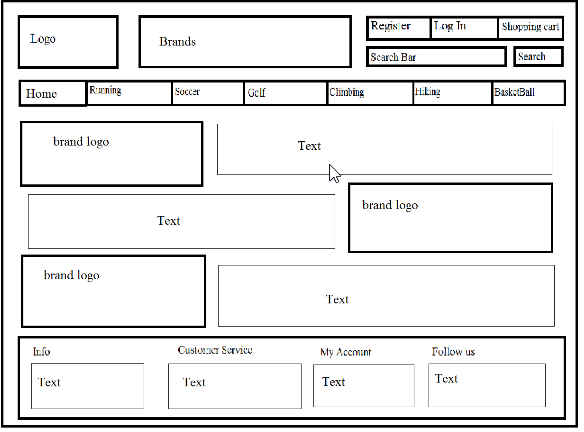
A fixed menu bar will be offered to the user with the choice to redirect to another part of our site on demand and without the need for scrolling through pages.

## Mock-ups screens

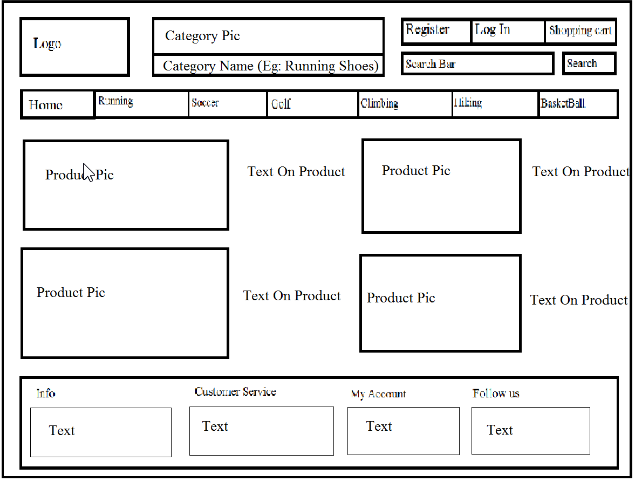
Home page



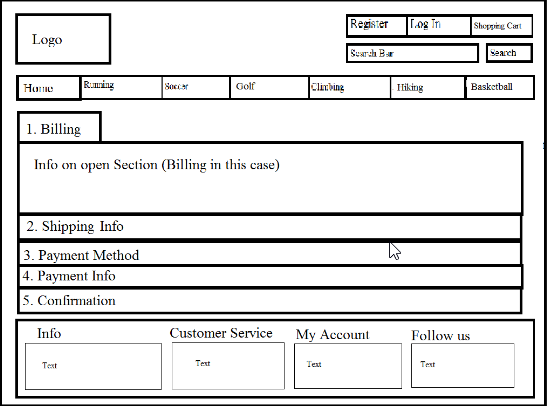
Brands Page



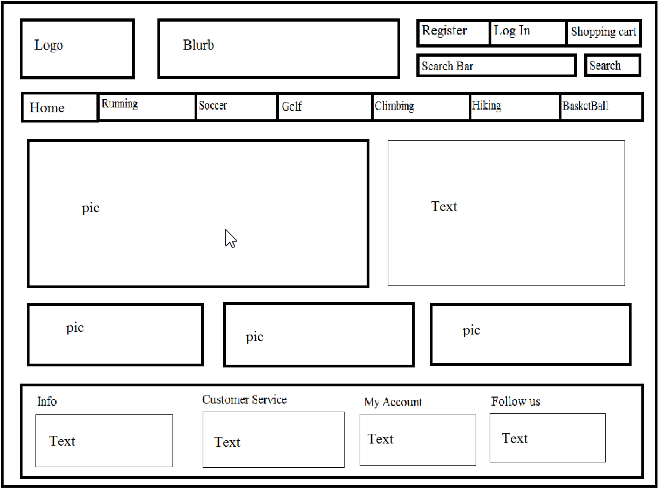
Certain Category Page



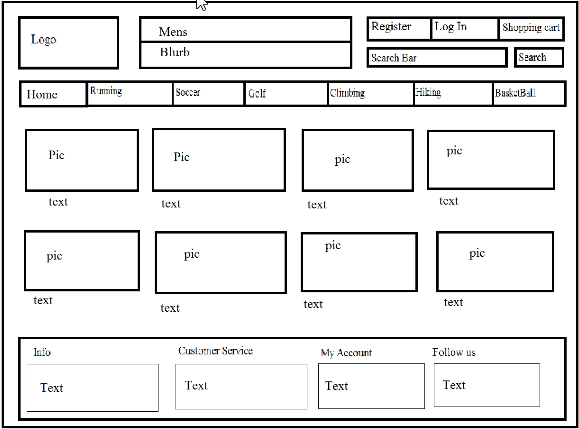
Check Out Page



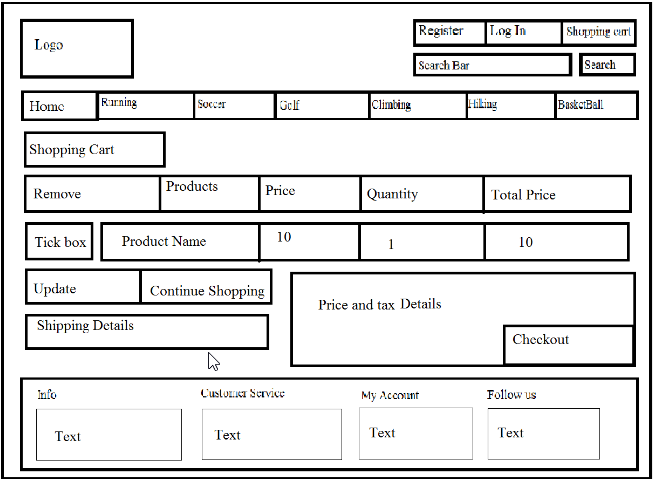
Company Page



Product Layout Page



Shopping Cart Page



## Attachments and Others

Proposed project timeline

