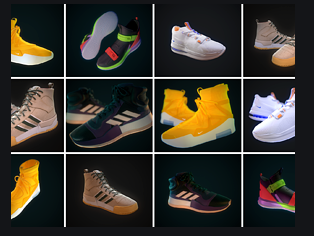
SSD 2020

Project Proposal Technical Specification



x19175108 Geneci Ui Fhatharta

x19176414 Katarzyna Casserly

x19141815 Ronan Behan

x19139306 Rory Cleary

Server Side Development

shoes@web Application

Contents

[1 Session 1 - Project Overview 3](#_Toc57222226)

[1.1 Description of the shoes@web Application 3](#_Toc57222227)

[2 Session 2 – Technical Specification 3](#_Toc57222228)

[2.1 Model 3](#_Toc57222229)

[2.2 Key Users Flow 4](#_Toc57222230)

[2.3 shoes@web Flow from visitor to User/Customer 4](#_Toc57222231)

[2.4 Functional Requirements for Non-registered User 5](#_Toc57222232)

[2.5 Functional Requirements for a Registered User 5](#_Toc57222233)

[2.6 Functional Requirements for an Administrator Manager 5](#_Toc57222234)

[2.7 Functional Requirements for an Administrator User 6](#_Toc57222235)

[2.8 Non-Functional Requirements 6](#_Toc57222236)

[2.11 External interfaces 9](#_Toc57222237)

[2.12 Extra functionalities 9](#_Toc57222238)

[2.13 Bibliography 10](#_Toc57222239)

# Session 1 - Project Overview

## Description of the shoes@web Application

The scope of this project is to design and implement an e-commerce website for shoes@web. 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 scale 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 developers to create a very responsive and flexible application.

Our shoes@web application will be responsive, and it will 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 be created.

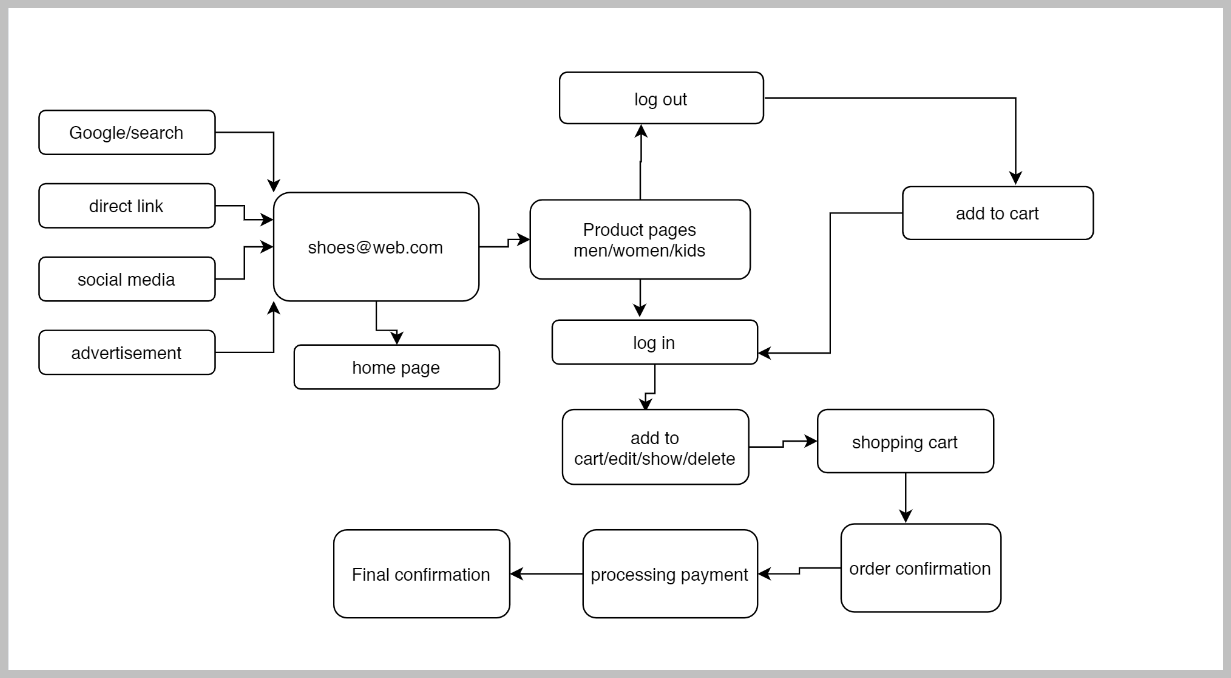
shoes@web will be an interactive application allowing response on request from the web browser. The primary purpose of the website will be selling products online. The application allows users to browse through the website in order to search, select and purchase products. If a user decides to purchase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

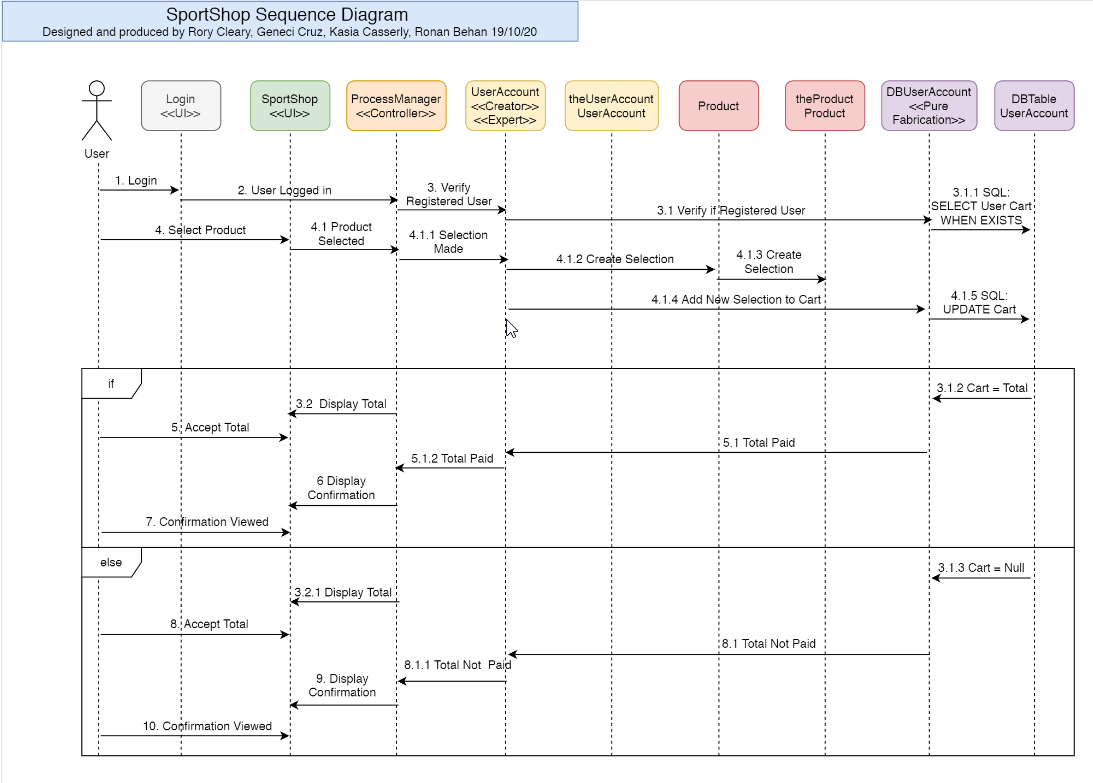
shoes@web exists to increase sales, work as a marketing tool to improve customer service and to obtain data regarding general product demand.

# Session 2 – Technical Specification

## Model

## Key Users Flow





## shoes@web Flow from visitor to User/Customer

The shoes@web applications main objective is to allow a registered user to purchase sport shoes products online. The following functionalities will be developed:

* A fully functional shoes@web application.
* Define access control to different users.
* A top navigation bar with buttons to access home, products pages, log in/log out, search and cart pages contained in it.
* A product section developed to ensure the user can access different products available by categories.
* A database to hold information inputted into the website.
* An easy to use website.
* An easy to use navigation bar.
* Flash message informing user: “Item successfully added to cart”.
* Flash message informing user: “Please agree with our terms and conditions”.
* Registration page – The shoes@web application must provide a registration page to define a username and a password to a user.
* The web service provides a search functionality to the inventory.
* Inventory – any user can visit different pages of the web application
  + Men’s, Women’s and Kids products pages
* Search functionality can be accessed by:
  + Product name
  + Product model
  + Product brands
  + Product sizes

## Functional Requirements for Non-registered User

* Non-registered user can open the website URL.
* Non-registered user can navigate between products pages.
* Non-registered user cannot add items to a Cart, therefore, cannot make a purchase.

## Functional Requirements for a Registered User

* Login page – To allow user to log into the web service at any time.
* The web service must allow user to log out.
* Registered user can make purchase
* Registered user can add item to the cart
* Registered user should be able to remove item from the cart
* 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
* Registered user can unregister from account

## Functional Requirements for an Administrator Manager

* Has Read/Write/Modify permissions
* Add, edit, and delete products
* Change prices of created products
* Deal with site security

## Functional Requirements for an Administrator User

* Has no write permission to the customer table
* Has write permission to edit images
* Has read permission to orders and addresses of customers

## 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
* 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 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.

***2.9*** ***Front End Technical Spec***

***Application***

Our web application will allow certain functions and aspects so that a registered user has the ability to log into their account and browse through the different product pages available and to purchase products through the cart facility. Registered users will also have the ability to delete their account and remove their data for our database.

These pages will include static and dynamic pages:

* Navigate to and from all pages to any other page on the application
* men’s shoes – browse through our online catalogue of men’s shoes
* women’s shoes - browse through our online catalogue of women’s shoes
* kids shoes - browse through our online catalogue of kids shoes
* sign up as a registered user
* log in/log out of user account
* search button – user can search through our database for a particular product

***Cart***

Registered users will also have access to the cart page, which will enable each user to:

* view cart
* add products
* remove products
* empty cart

***Favicon***

A favicon will be displayed when a user opens our application in certain browsers. The browsers that have the capability to display these favicons are Firefox, Edge and Chrome. At this current time Safari does not offer this function. Our favicon is a coloured shoe.

***Header***

Banner section will include the company logo and colours of navy, black and white. A slogan will also be visible to users.

After the user encounters our image header with the main image on the home page alone they will scroll down towards the card image section. This section includes 3 images displayed in a block and each will function as a direct link to the men’s, women’s and kids pages. Each card is coded with a link\_to image tag to the items\_url. When a user hovers over these images they will dynamically change their opacity and make the user aware of their function to redirect to another page on the application.

***Navigation***

Navigation area will be developed with buttons and icons and each will be designed with a different colour so as to make it a more dynamic experience for the user throughout the application.

* Home button (home icon) with a red colour
* Men’s button with a blue colour
* Women’s button with a pink colour
* Kids button with a green colour
* Log in button with a grey colour
* Log out button with a teal colour
* Search button with a purple colour
* Cart button with an orange colour

***Subscribe***

The next section will display a subscribe button, coloured in red, which enables a user to subscribe to our email service which will place each user on our mailing list and give them access to special offers on an advanced basis.

Side by side with this is our special offer section, alerting the user to current offers available.

***Footer***

At the bottom of our application is the footer section. This includes links to numerous other pages which increases the functionality and options available to the user. These pages are:

* about us – displaying the general information in relation to the company
* contact us – allowing the user to contact us using a form
* support - send details of a technical issue to our support team
* shipment – check out our shipping policies
* giftcard – allow a user to purchase a gift card
* returns/refunds – make user aware of the processes involved in returning a product

Links to various social media websites are displayed in this section also which will enable user to access these websites on request.

The Ruby language also provides the option of including a piece of code which increments the date. Our application has the functionality built into it which will change the year on an incremental basis when it is required. The code used is: <%= Time.now.year %>.

***Items/Products pages***

When the user redirects to the products pages they will be able to visualise the different product ranges that are on offer. Once they browse and select a product they like, they can view a description of the product, the price, available sizes and colours and if they wish, add the product to their cart. The functions available to the user will ensure that the chosen product is then sent to their cart where they can finalise their purchase or if they decide to change their mind can remove the product from their cart and begin the search for another product.

***Thank you page***

Once a purchase has been completed a page displaying a thank you message, and order number will be visible on the user’s screen.

***2.10*** ***Use of Third-Party Gems***

* Devise gem used to generate authenticated users.
* PayPal gem – To allow us to receive payment via PayPal system.

## External interfaces

* The shoes@web application will have a link to social media such as: Facebook, Instagram, Snapshot, Twitter, and LinkedIn.

## Extra functionalities

To make the website more users friendly and fast to access it some functionality can be added.

* Allow user to login to the shoes@web via Facebook or Google account.
* Provide users with order history.
* Add some useful JavaScript
* Make the application responsive which will resize the content on all devices.
* Allow the user to navigate throughout the application at different sections on each page.

***Project timeline***

# 

## Bibliography

Roth, R.M., Dennis, A. and Barbara Haley Wixom (2013). *System analysis and design.* Hoboken, N.J.: Wiley.

Lecturer McCabe, L (2020) ‘Introduction to Server-Side Development’ [PowerPoint presentation]. [HDSDEVSEPOL\_YR2](https://mymoodle.ncirl.ie/course/view.php?id=813)Server-Side Development*.* Available at: <https://mymoodle.ncirl.ie/course/view.php?id=813> (Accessed: 11 October 2020).

Lecturer McCabe, L (2020) ‘Ruby on Rails’ [PowerPoint presentation]. [HDSDEVSEPOL\_YR2](https://mymoodle.ncirl.ie/course/view.php?id=813)Server-Side Development*.* Available at: <https://mymoodle.ncirl.ie/course/view.php?id=813> (Accessed: 12 October 2020).

Lecturer McCabe, L (2020) ‘Routes, Relationships and Active Record’ [PowerPoint presentation]. [HDSDEVSEPOL\_YR2](https://mymoodle.ncirl.ie/course/view.php?id=813)Server-Side Development*.* Available at: <https://mymoodle.ncirl.ie/course/view.php?id=813> (Accessed: 12 October 2020).

‌Dr. Muhammad, I (2020) ‘SQL Programming DML (Part I)’ [PowerPoint presentation]. *HDCOMP\_877117964 Introduction to Databases.* Available at: <https://moodle.ncirl.ie/mod/resource/view.php?id=50820> (Accessed: 13 October 2020).

Dr. Muhammad, I (2020) ‘SQL Programming DML (Part II)’ [PowerPoint presentation]. *HDCOMP\_877117964 Introduction to Databases.* Available at: <https://moodle.ncirl.ie/mod/resource/view.php?id=50820> (Accessed: 13 October 2020).

MySQL:: MySQL Tutorial. 2020. Date Calculations. [ONLINE] Available at: https://dev.mysql.com/doc/mysql-tutorial-excerpt/5.7/en/date-calculations.html. [Accessed 13 October 2020].

Learn the Unified Modelling Language V2 – Up2UML within distinct Software Development Processes [ONLINE] Available at: <https://moodle.ncirl.ie/pluginfile.php/505443/mod_resource/content/1/alpha1191248764%20%281%29.pdf>. [Accessed 19 October 2020].

Schramel, M (2020) ‘POST Case study part II Analysis Phase’ [PowerPoint presentation]. [HDSDEVSEPOL\_YR2](https://mymoodle.ncirl.ie/course/view.php?id=813) *Object Oriented Software Engineering.* Available at: <https://moodle2019.ncirl.ie/course/view.php?id=420> (Accessed: 13 October 2020).