SOFTWARE REQUIREMENTS SPECIFICATION

for

<Challenge>

Version 1.2 approved

Prepared by < *Joel Coelho* >

December 1, 2024

Contents

1 Introduction

This application provides the ability to show a list of products (plans, devices, mobile) and their information.

1.1 CHALLENGE CORE

The central core of the application is a web app that replicates a phone provider service. The application can be split into 4 subsections: Plans, Devices, Mobile, and Help. In the plans subsection, the user can see multiple phone plans and select one that he wants. In the devices subsection, the user will be able to see all the devices available to buy. The user will also see the multiple colours, the price, and the device name. In the mobile subsection, the user will be able to see all the mobile phones available to buy, this subsection will be mostly identical to the devices subsection. Finally, in the Help section, the user will be able to check the Frequently Asked Questions (FAQ), and also be able to contact support.

2 Documentation

2.1 Product Perspective

1. Device (Product) - A device has this information:
    a) Brand - The brand of the device
    b) Name - The name of the device
    c) Price - The price of the device
    d) Upfront - The upfront price of the device
    e) New status - New device status
    f) Discount - Discount of the device
2. Product Option - A Product Option has this information:
    a) Option Type - The type of the option, this can be p.e ”storage” or ”colour”
3. Product Option Value - A Product Option Value has this information:
    a) Value - The value of the option, this can be p.e ”white” or ”128GB”
4. Product Option Property - A product option property (variant of the product) has this information:
    a) Price - The price of the variant
    b) Upfront - The upfront price of the variant

2.2 Domain Model

2.3 Requirement Engineering

2.3.1 Functional Requirement

This section presents the functional requirements of the Challenge app where the authors and their interactions are described.

Authors

Non Registered User A non registered user
User A user that is logged on

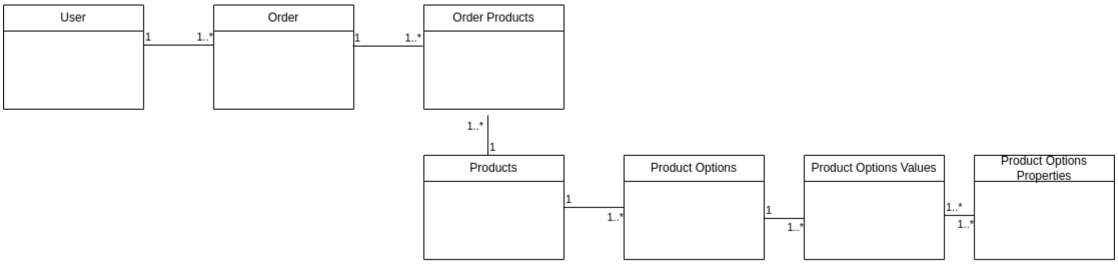

Figure 2.1: Domain Model

Use Cases

Non Registered User A non registered user

1. As a non registered user I want to register in the system so that I access the Challenge app
2. As a non registered user I want to log in to the system so that I access the Challenge app
3. As a non registered user I want to retrieve a lost password so that I can regain access to the Challenge app

User A user that is logged on **Non Registered User** A non registered user

1. As a user /non registered user I want to see all the available plans
2. As a user /non registered user I want to see all the available devices
3. As a user /non registered user I want to see all the available mobile devices
4. As a user /non registered user I want to have access to the Help section

User A user that is logged on

1. As a user I want to logout of the Challenge app

Coach A user that coaches athletes

2.3.2 Non Functional Requirements

1. Have good reliability
2. Multiple browser support
3. Follow GDPR guidelines
4. Have a good processing speed

2.4 Operating Constraints

2.4.1 Back-end

Ruby on Rails

Besides being a technology known to the user, Ruby on Rails, a framework for web applications, is crafted using the Ruby programming language. Below are the advantages of mastering Ruby on Rails compared to other coding languages:

1. Simplicity and Accessibility: Ruby is tailored for developers, boasting an uncomplicated syntax that's both easy to comprehend and write. Ruby on Rails adheres to the "convention over configuration" principle, simplifying initial setup with predefined defaults and assumptions, thus facilitating a smoother start for developers.

2. Swift Development: Renowned for its capability to expedite web application development, Ruby on Rails equips developers with built-in functionalities and conventions, such as automated database migrations, streamlining the process of building and tweaking applications.

3. Vast Community and Resources: With a thriving community of developers, Ruby on Rails offers abundant online resources, tutorials, and forums. This extensive support network makes it effortless for developers to seek assistance, learn from peers, and actively engage with the community.

4. Scalability and Versatility: Ruby on Rails is adaptable and scalable, empowering developers to construct expansive web applications capable of managing high-traffic volumes and intricate workflows. Furthermore, it provides extensive support for third-party libraries and tools, facilitating seamless extension and customization.

5. Lucrative Career Opportunities: Ruby on Rails enjoys widespread adoption by both startups and established enterprises, fostering a high demand for skilled developers proficient in this language and framework. Acquiring expertise in Ruby on Rails can unlock numerous career prospects within the tech industry.

2.4.2 Front-End

React

Besides being a technology known to the user, React, a library for building user interfaces or UI components has the following advantages:

1. Simplified Learning: React is praised for its ease of learning, backed by extensive documentation. It's favored by JavaScript developers for its simplicity, serving as the "V" in MVC architecture. Despite not being fully featured, it's lauded as an open-source UI library enhancing task execution.

2. Enhanced Reusability: React employs a component-based approach, fostering modular development. Components generate reusable HTML code fragments, simplifying application development and maintenance. The virtual DOM mechanism optimizes performance by selectively updating elements, enhancing scalability.

3. Useful Development Tools: React offers developer tools like React Developer Tools for efficient debugging and inspection. These extensions enable easy examination of component hierarchies within the virtual DOM, enhancing productivity and code optimization.

4. Rich JavaScript Library: React boasts a comprehensive JavaScript library, providing flexibility for diverse project requirements. Developers leverage this richness to adopt various approaches and methodologies, catering to individual project needs effectively.

5. Performance Boost: React enhances performance through virtual DOM implementation, alleviating traditional DOM manipulation bottlenecks. Operating on virtual components rather than directly manipulating the DOM results in smoother, faster performance, improving user experience.

2.5 Design

2.5.1 Component Diagram

- Client - Application running on user's device.
- Challenge Core - Responsible for Challenge Core functionalities.

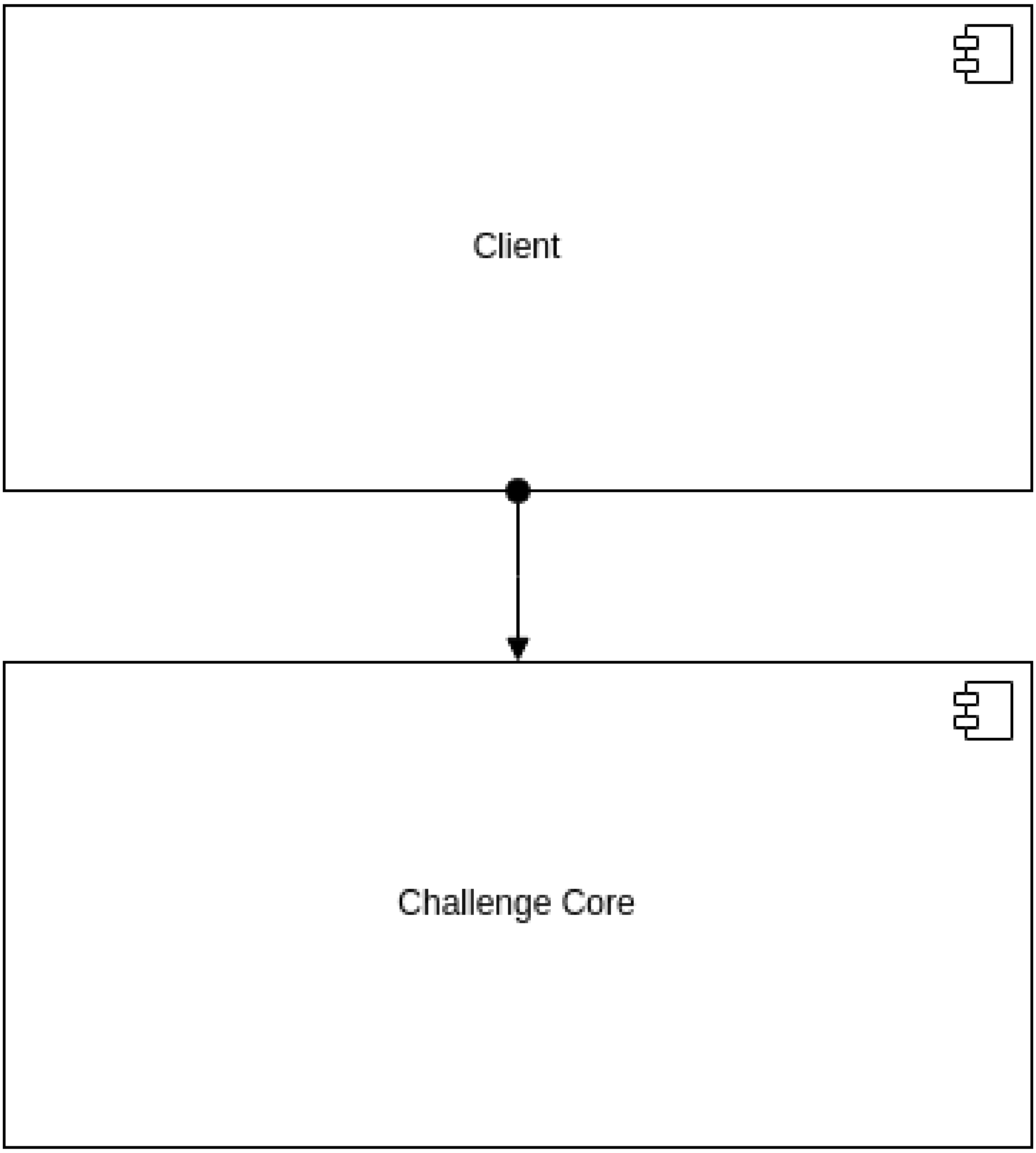

Figure 2.2: Component Diagram