Design Document RUSNVC E-Shop

Cristin Rusnac

Fontys University of Applied Science

Abstract

The Design Document contains all the design related aspects of the project. The Technology Decision Matrix represents the choices made for specific technology based on certain criteria. The C4 Model describes the software architecture of the solution starting from the System Context Diagram and zoom in to the UML diagram to offer an in depth overview of the whole System. The Wireframes represents a visual mockup of the final product that is meant to be delivered. Product backlog consists of user stories, the prioritization of each user story and the evaluation of time it may take to implement each feature.

Technology Decision Matrix

The Technology Decision Matrix represents an analysis of the Technology options evaluated by specific criteria and the choices made for this project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Frontend Framework | | | | |
| Criteria | **Cost** | **Community** | **Reliability** | **Total** |
| **Vue** | 10 | 4 | 5 | 19 |
| **Angular** | 10 | 7 | 6 | 23 |
| **React** | 10 | 9 | 9 | 28 |

For the Frontend Framework React Js is the most optimal choice due to the fact that it has the largest community with out of box solutions, it is the most popular framework and most efficient one for this specific project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Backend | | | | |
| Criteria | **Cost** | **Community** | **Reliability** | **Total** |
| **Node JS** | 10 | 8 | 8 | 26 |
| **C#** | 10 | 7 | 7 | 24 |
| **Java** | 10 | 7 | 7 | 24 |

Node JS will be used for the backend, due to the fact that it has a large community, is easy to integrate with react and also JavaScript will be used for both frontend and backend.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Messaging | | | | |
| Criteria | **Learning Curve** | **Documentation** | **Reliability** | **Total** |
| **NATS** | 8 | 7 | 8 | 25 |
| **RabbitMQ** | 6 | 5 | 7 | 22 |
| **Kafka** | 6 | 6 | 7 | 23 |

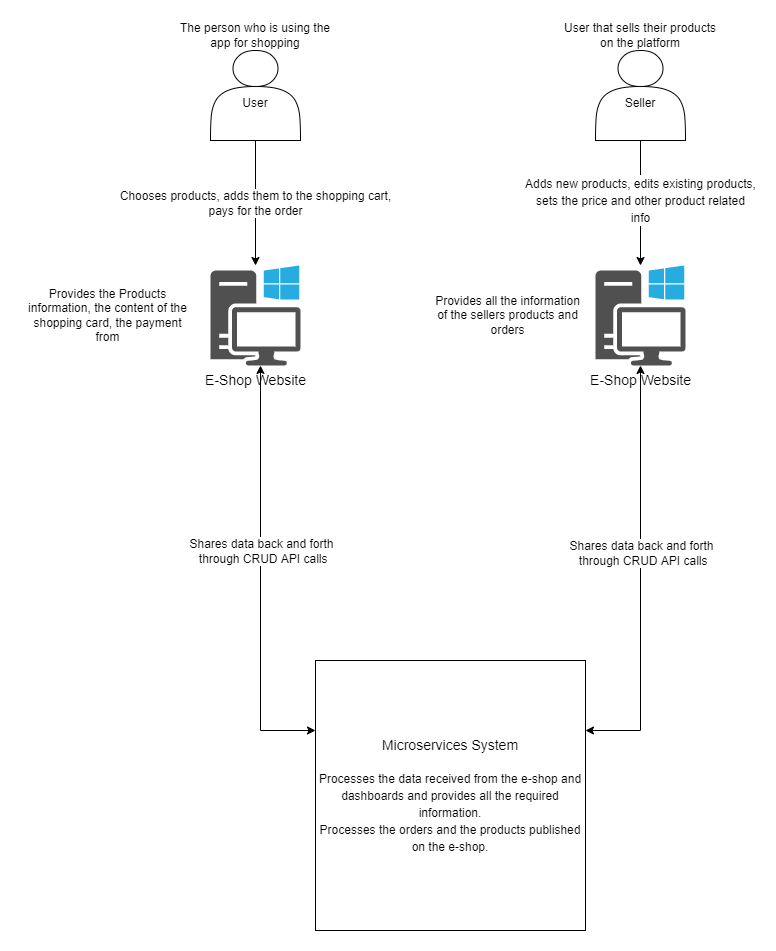
NATS is chosen as the most optimal Event Bus Library due to its ease of integration with any microservice system, a easier learning curve in comparison to RabbitMQ, the fact that it is easiest to setup and the comprehensive documentation.

C4 Model

The C4 Model is the most optimal way of visualizing software architecture since you start with modeling a System Context Diagram and through the development process zoom in up to the Code representation.

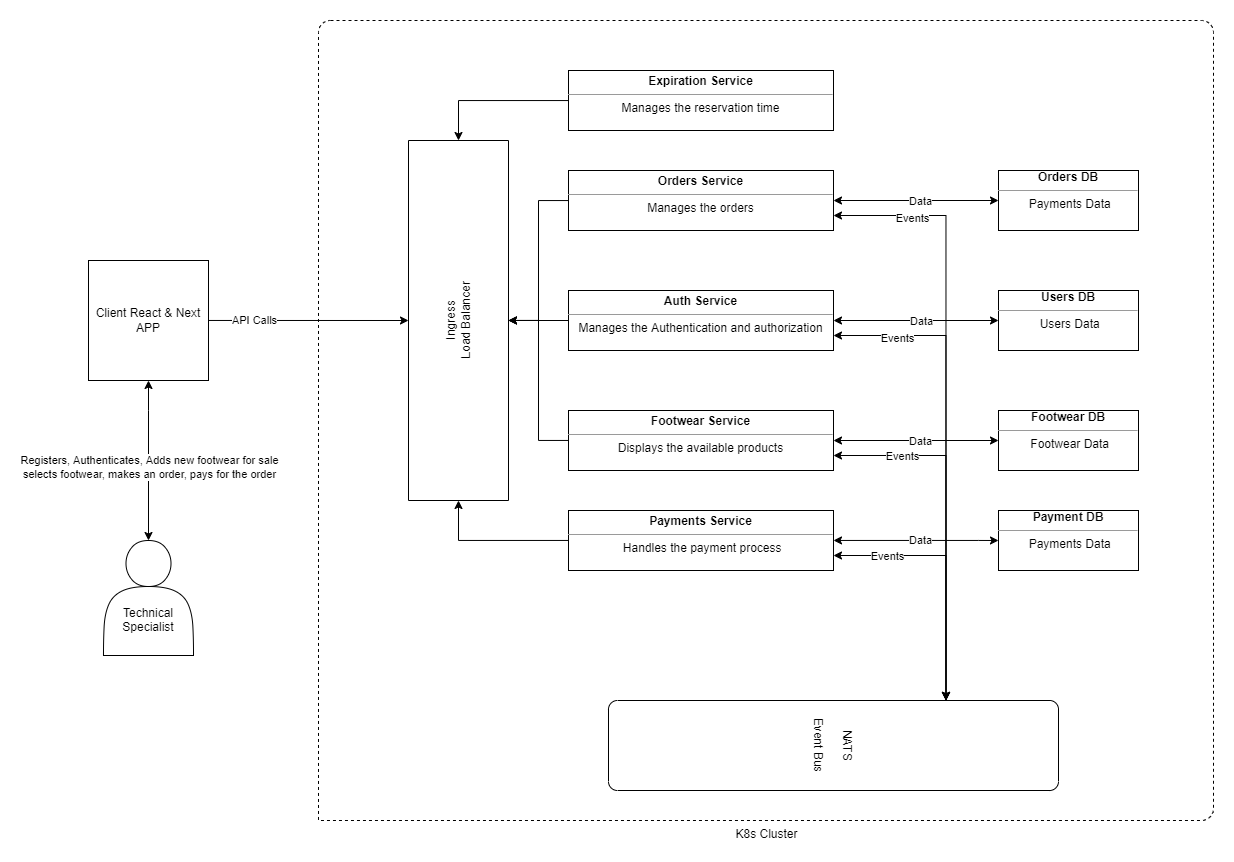
## C1: System Context Diagram

The Context Diagram represents the software system and its scope in a zoomed out format with all the stakeholders involved and how they are meant to interact with the system.



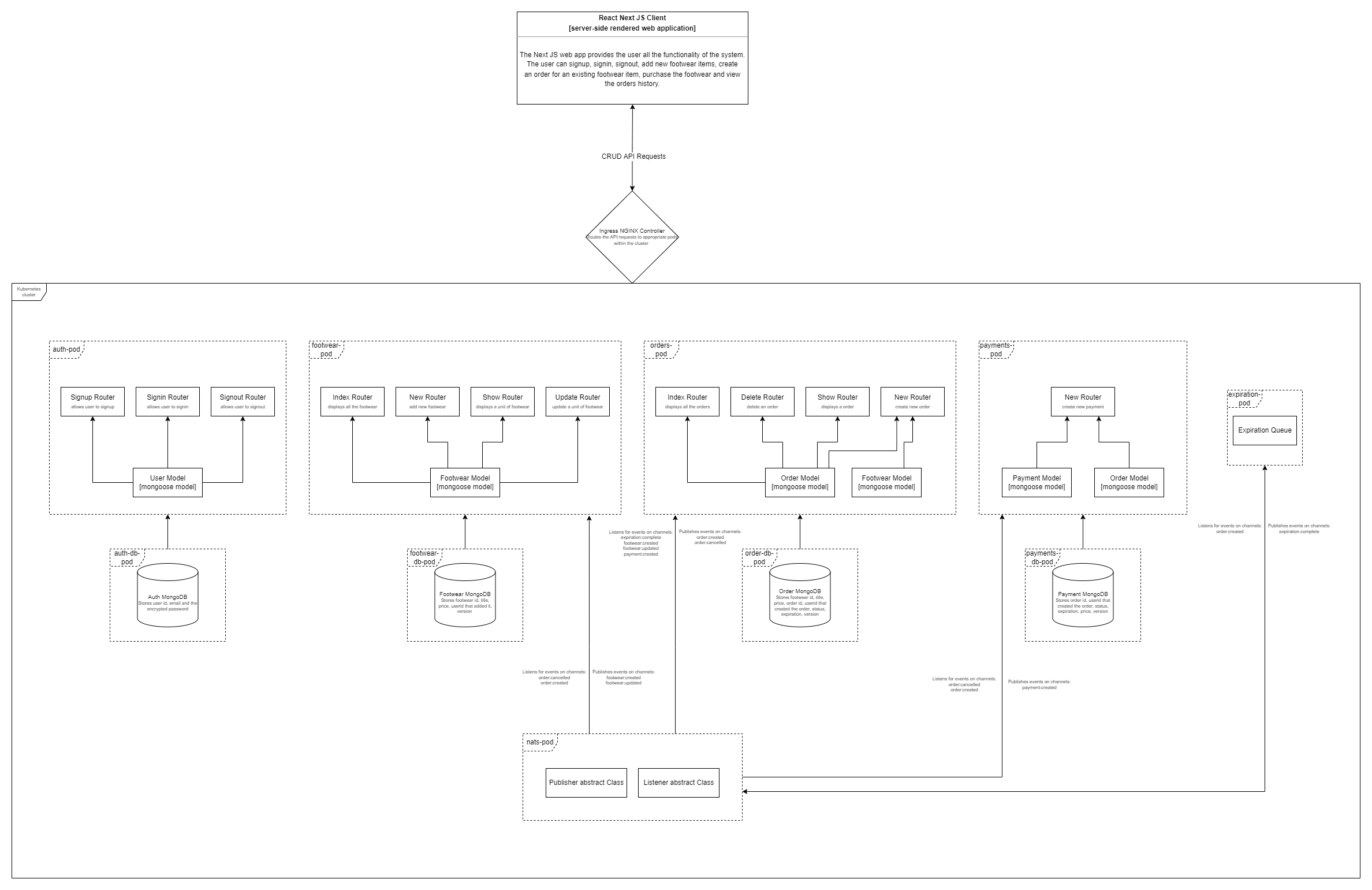
## C2: System Context Diagram

The Context Diagram represents the software system and its scope in a zoomed out format with all the stakeholders involved and how they are meant to interact with the system.



## C2: Component Diagram

The Component Diagram represents the each component of the system in this specific case the components of each microservice, the functionality of them, the responsibility and implementation details.



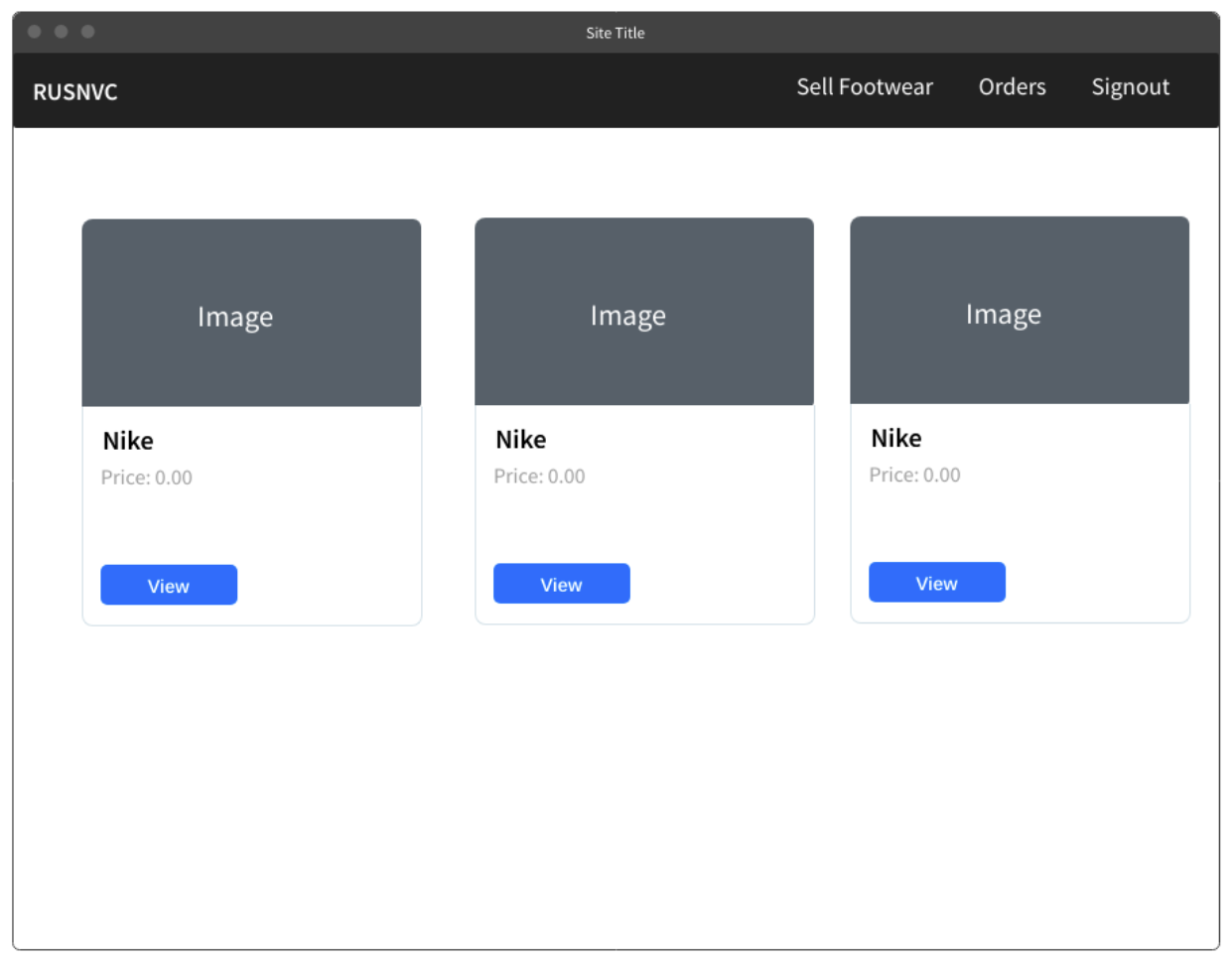
Wireframes

Authentication:

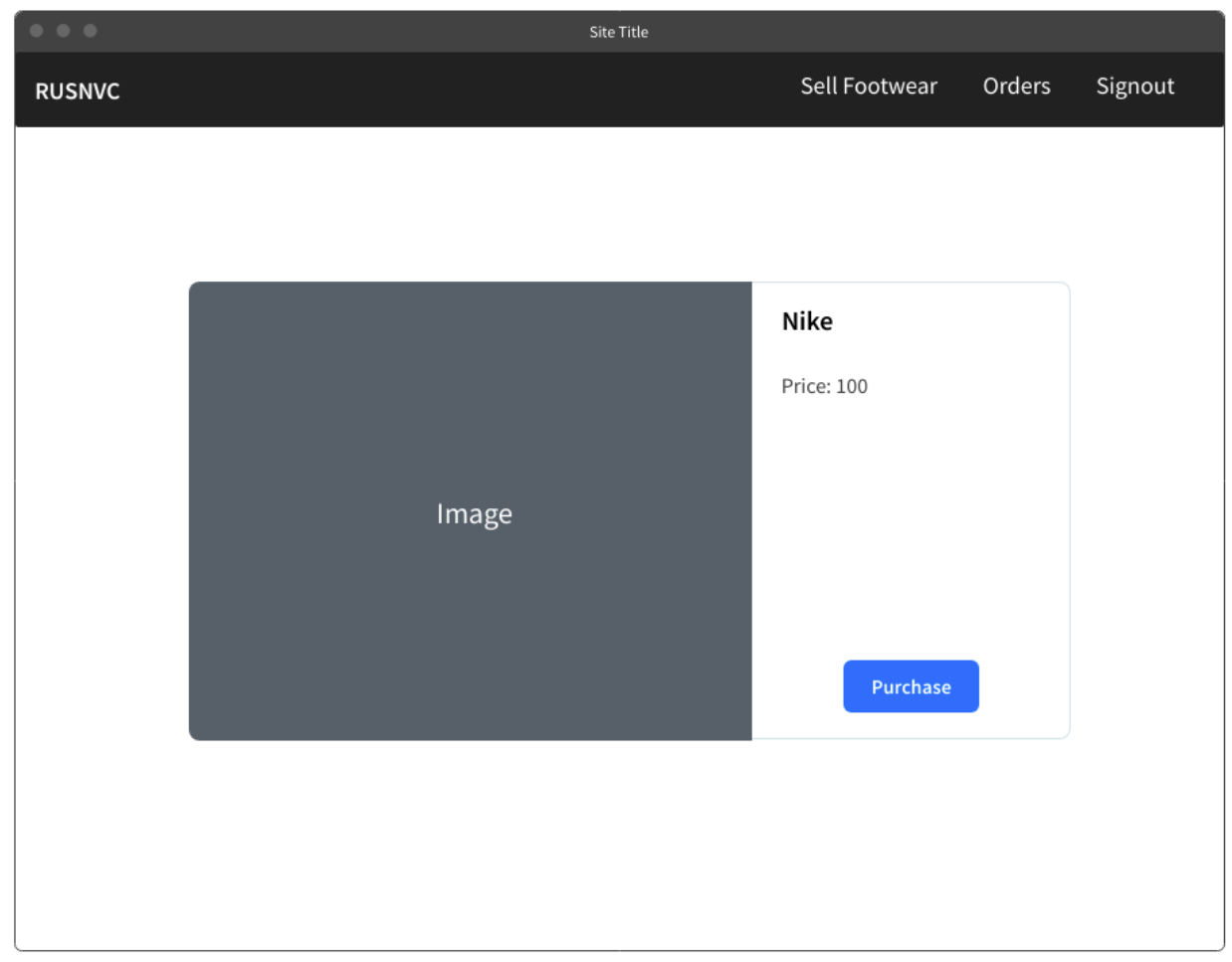
Graphical user interface, application

Description automatically generated

Homepage



Footwear Purchase



Payment Page

Graphical user interface, website

Description automatically generated

Orders

