Natural Online Store

Analysis and Design Document

Student: Roxana-Ioana Aldea

**Group:30431**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 18/03/2020 | 1.0 | Project Deliverable 1. Project Specification, Elaboration Iteration I, Construction and Transition sections added. | Roxana-Ioana Aldea |
| 02/04/2020 | 1.1 | **Project Deliverable 2:** Domain Model, Architectural Design, Component and Deployment diagrams | Roxana-Ioana Aldea |
| 29/04/2020 | 1.2 | **Project Deliverable 3:** Design Model, Data Model | Roxana-Ioana Aldea |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

# Natural Online Store is a web application designed for clients who want to find a natural cure for their illnesses or needs.

# The application allows the users to find the proper products either by selecting their illnesses or by selecting a category (e.g. teas, essential oils). When the user selects a product, he/she can find out more information about it, for example ingredients, administration method or storage mode. If the user accidentally selects a product which contains an ingredient at which he/she is allergic, a notification message will pop up.

# The products can be added to a cart and after finishing the command, it can be easily sent. In case of a product not being available, the user will be notified when the stock is refilled.

# The application has two types of users, an administrator which can add, edit or delete products and a user which is the client and can buy products.

# Elaboration – Iteration 1.1

# Domain Model

*The domain model of this application can be seen as a visual representation of the conceptual classes in the domain of interest of this application. There is a class for each type of user: administrator, teacher, student. Also, there are classes for courses and for teacher evaluation. Other classes will be added along the way.*

*O imagine care conține hartă, text

Descriere generată automat*

# Architectural Design

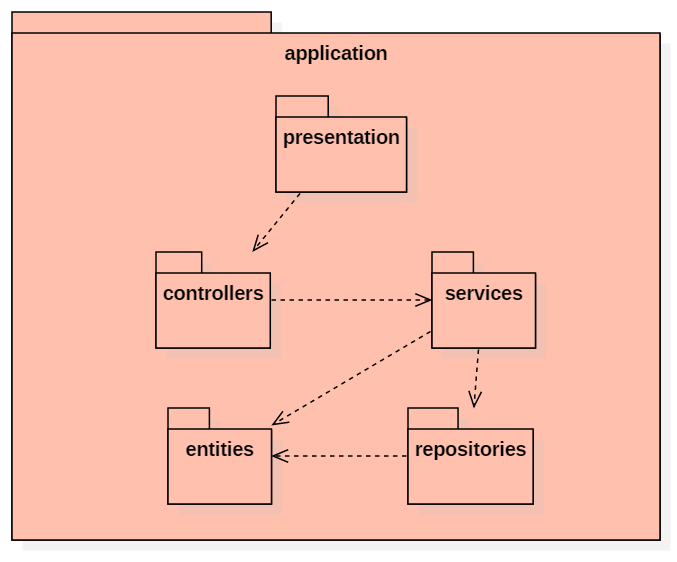
## Conceptual Architecture

*For this project the best choice would be a layered architectural style because it is a web application. The project is based on a client-server model, so each request has to be fulfilled, and there are several steps be accomplished. This way, each layer can perform a specific role. The architecture is a 3-tire architecture, split in a Data Access Layer, a Business Layer and a Presentation Layer.*

*O imagine care conține text

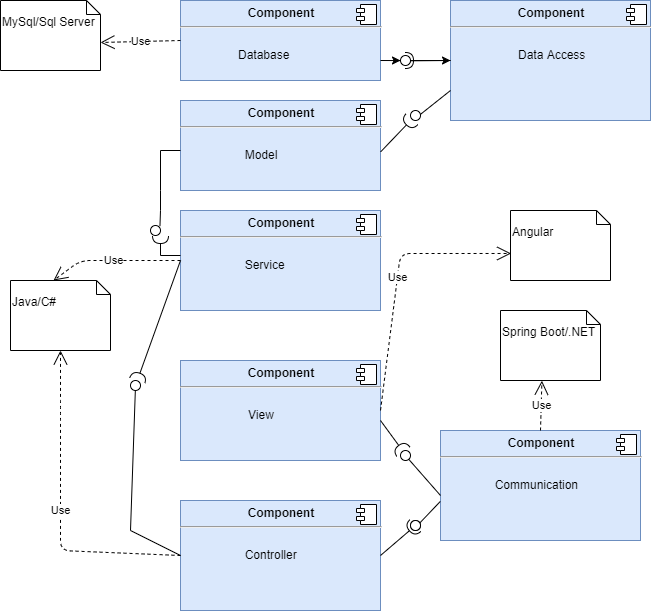
Descriere generată automat*

## Package Design



## Component and Deployment Diagrams

Component diagrams are used to present the physical artefacts of the system and also the relationships between them. The artefact includes files, executables, libraries. It is build during the implementation phase of the application.



# O imagine care conține captură de ecran Descriere generată automat

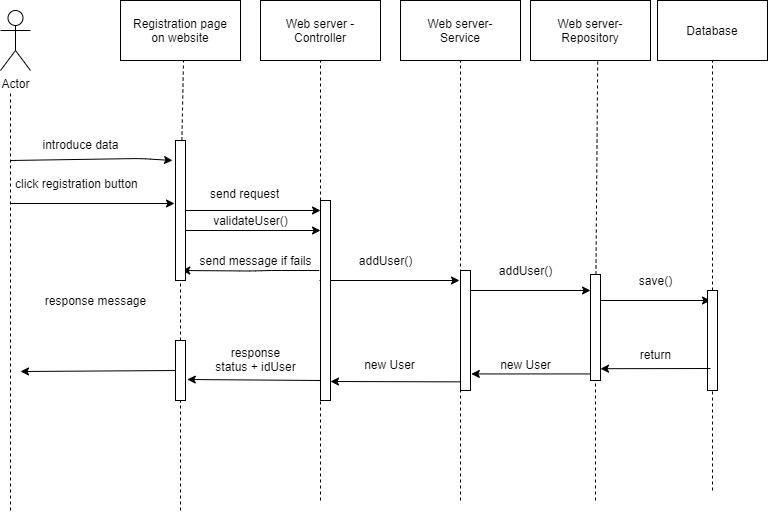
# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

*[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]*

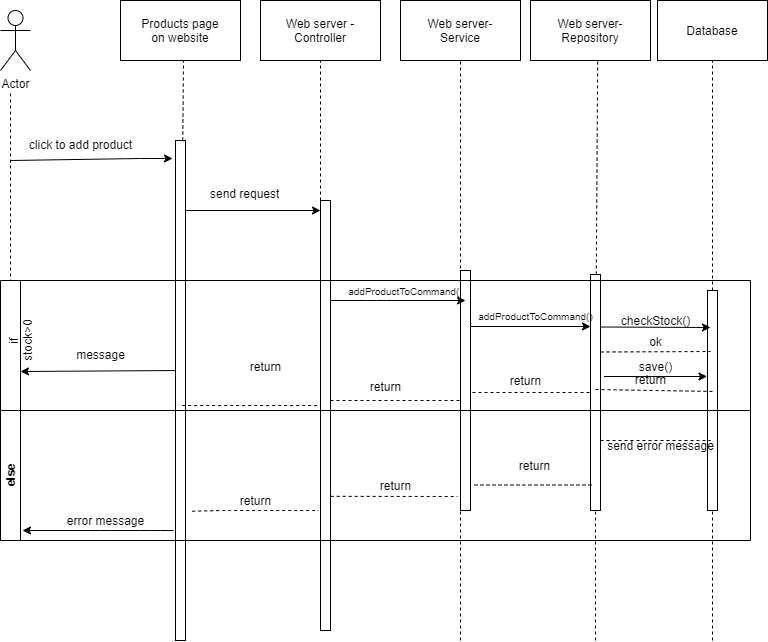
1. Register

**

O imagine care conține captură de ecran

Descriere generată automat

1. Add product

**

*O imagine care conține captură de ecran

Descriere generată automat*

## Class Design

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

Below is a part of classes and for controller and service packages I have added 2 classes, but there will be many more. I have exemplified this by drawing the packages. This will not be the final version of the class diagram.

*O imagine care conține text, hartă

Descriere generată automat*

In this project, the Observer Design Pattern will be used in order to implement the system of notifications, when some stock products are refilled by the administrator. It may appear more design patterns on future implementation.

# Data Model

*[Create the data model for the system.]*

**

# Unit Testing

# Elaboration – Iteration 2

# Architectural Design Refinement

I have decided to use a client-server architecture instead of a layered one in order to reduce the coupling between the presentation and the business logic. This way, the application can have many clients in different f

# Design Model Refinement

# Construction and Transition

# System Testing

The integration tests are performed based on the use-case diagram and they will indicate that the system functions properly.

Test cases:

Register: The client user has to fill the required fields with valid data, valid phone number or email address, also has to introduce his/her allergies. Otherwise, a notification will appear.

Login: the user selects his/her profile: administrator or client, introduces his/her username and password and if they are found in the database, will be allowed to enter the application.

Edit product: The administrator can edit products, provided they exist in the database

Delete product: The administrator can delete products, provided they exist in the database

Buy product: The user can buy products and if the product contains none of the ingredients he/she is allergic, he/she can add it. Otherwise, a notification will pop up and if the user marks that he/she still want to add that product, the product will be added in the cart.

# Future improvements

The application can be improved in many ways. One improvement will be to present some recipes which use ingredients from the site and by clicking on the ingredient name, the user to be redirected to that ingredient’s page where it can be added to the cart.

Other improvement will be to generate a report of the command which the user can save.

Another scenario will be to offer the users vouchers based on the past command, which can be used for future commands or to assign each product an expiration date and to make some discounts for product which expire that month.

# Bibliography