
Pet Pack

**Pet Supplies Delivery
Software Architecture Document**

Version 1.0



PET PACK

WE LIKE PETS

Pet Pack

Revision History

Date	Version	Description	Author
29/4/2019	1.1	First Version	Grupo1

Table of Contents

Introduction	4
Purpose	4
Scope	4
Definitions, Acronyms, and Abbreviations	4
References	4
Overview	4
Architectural Representation	4
Architectural Goals and Constraints	5
Use-Case View	5
Logical View	5
Overview	5
Architecturally Significant Design Packages	6
Use-Case Realizations	6
Process View	6
Deployment View	6
Implementation View	6
Overview	6
Layers	7
Data View (optional)	7
Size and Performance	7
Quality	7

Software Architecture Document

1. Introduction

1.1 Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

1.2 Scope

The Software Architecture Document outlines the decisions that lead to the principal structures used. Furthermore, it gives a description of each structure, including the fundamental specifications that allow the comprehension of the system in its environment.

1.3 Definitions, Acronyms, and Abbreviations

- AWS - Amazon Web Server

1.4 References

Pet Pack project: <https://github.com/shipupi/IngeSoft>

1.5 Overview

The Software Architecture Document specifies the goals, constraints, aspects of architecture, and expectations of the software. The document is divided into 11 sections: Introduction, Architectural Representation, Architectural Goals and Constraints, Use-Case View, Logical View, Process View, Deployment View, Implementation View, Data View, Size and Performance and Quality.

2. Architectural Representation

The views that are necessary to describe the software architecture and its representation are:

- Home Screen which shows the best-selling products of the website and leads to the different categories of items and services that can be purchased
- Search page which shows the result the user has searched for, for example all items belonging to one category
- Cart page which shows the shopping cart including all items the user has saved there during his shopping process and has an option to go directly to the purchase of all items in the shopping cart
- Purchase page where the user has to put in all relevant information for the order he wants to place
- User profile which gives an overview of all orders placed by the user and his account information

3. Architectural Goals and Constraints

The main objectives that the architecture needs to accomplish are the complete implementation of the basic use cases given to the team. Once this goal is achieved the next objective will be the implementation of different HCI techniques in order to create a smooth user experience on the website. One of the mayor constraints is that the team will need to work with a completely new set of development tool which requires

some time to get used to as well as the time limit which will heavily impact the the amount of hours per day needed in order to complete the project.

4. Use-Case View

Central functionality of the final system can be seen in the following use-cases:

Use_Case_1: Add and remove products to the site

Use_Case_3: Add a user

Use_Case_5: Browse shop

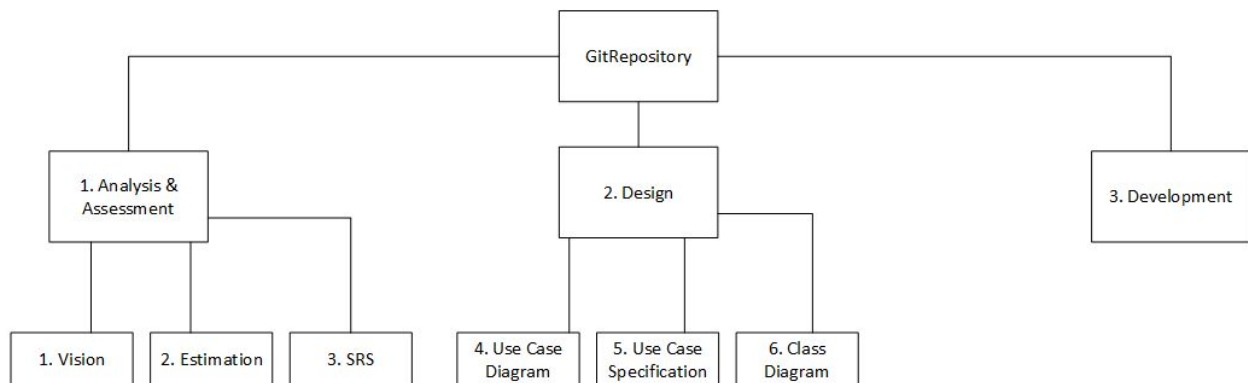
Use_Case_13: Place order

5. Logical View

5.1 Overview

The software is architecturally structured into a GitHub directory and as such it has a tree-like structure. The first level is comprised of three folders named after the step in Software Development belonging to their files (Analysis & Assessment, Design and Development). Furthermore, it precedes its name the order in which it should be taken into account. Analysis & Assessment and Design also have subfolder with a progressive order but Development does not there is no concrete order of implementation of the different packages.

A graphical interpretation of the repository is as follows:



5.2 Architecturally Significant Design Packages

As it can be seen in the above diagram the directory is broken into two distinct categories: Analysis & Assessment and Design and Development. The first two have an order that must be followed as they are the plan to follow in the development stage. Meanwhile, the Development stage every component works independently from the other and does not follow a strict order in to be built in.

Analysis & Assessment has what can be summed up as the contract. What the client wants (Vision), how much will it cost (Estimation) and what will be built (SRS).

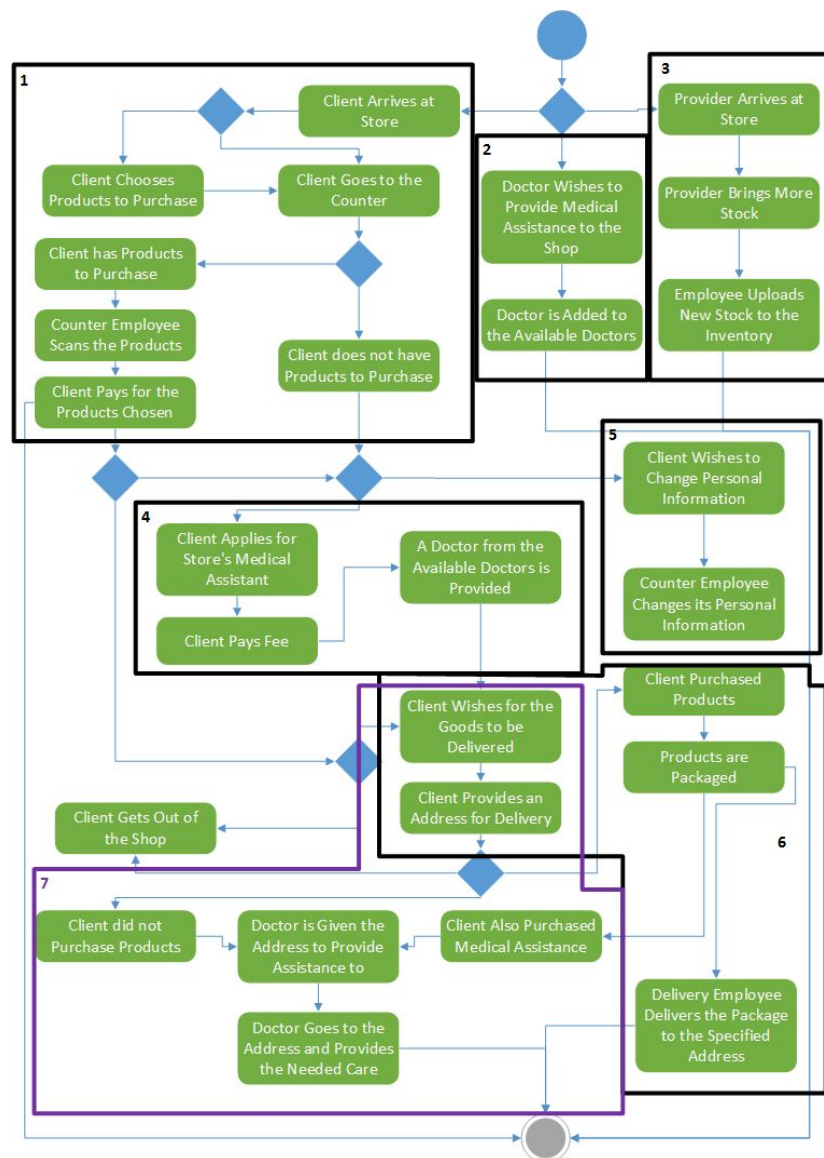
Design has an indepth look at how the software will be built and how it will operate. This show the inner workings of the program and gives an accurate picture of the look and feel of the software before seeing it in action. It is divided into: every distinct action someone can take in the software (Use Case Diagram); what does each action perform and how it performs it (Use Case Specification); how each action is going to be built (Class Diagram).

5.3 Use-Case Realizations

All the 19 use cases are specified and explained with its respective diagram in the git repository of the project. Go to the main Git folder and enter the Design folder, from there get to the folder named Use Case Specification.

6. Process View

The following diagram displays the processes which comprise the Business Model PetVet uses in a day-to-day basis:



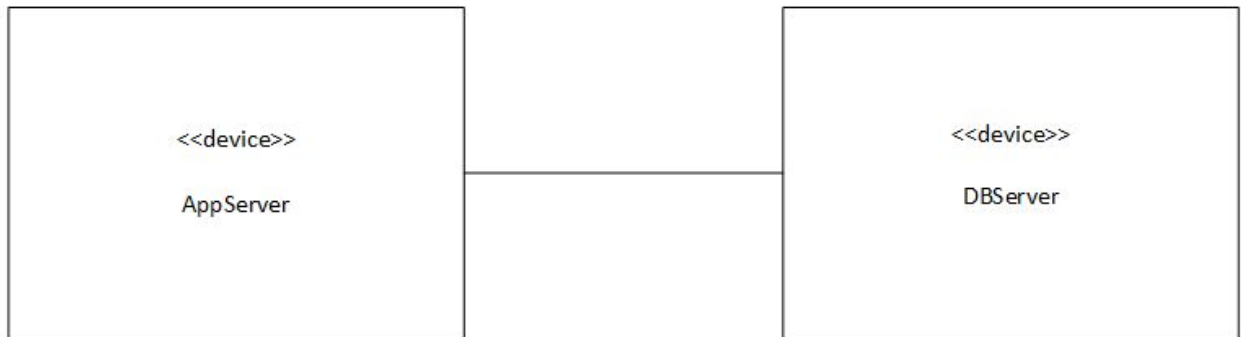
1. Sale Process

2. Medical Caregiver Contracting Process

3. Purchase Process
4. Medical Care Appointment Process
5. Client Personal Information Update Process
6. Sale Delivery Process
7. Medical Care Service Process

7. Deployment View

The software will be deployed on a AWS server, with debian 9,9 (stable), which will be assigned to a static external IP address. A different server will host the database, also AWS with debian 9,9 (stable).



8. Implementation View

8.1 Overview

- Application: Django framework structure
- GUI:
- Database: PostgreSQL

8.2 Layers

Application:

- Models:
 - Person
 - a. User
 - b. Admin
 - c. Packer
 - d. Vet
 - e. DeliveryEmployee
 - ShoppingCart
 - PackageHistory
 - Order
 - Address
 - Consult
 - Product

9. Data View (optional)

As the spatial complexity will not be prioritized, data structures for the storage of the users, administrators, veterinarians, and products will be chosen depending on its convenience and facility to access them (either for data writing or data finding).

10. Size and Performance

The website will allow the customer to search for or upload products as fast as possible. In order to make that possible the software is designed to be highly efficient. It must support a large quantity of products and an even larger quantity of clients.

11. Quality

The most important quality aspect of the Pet Pack website is usability. That is why the during the development there will be a strong focus on making the website very simple and straightforward to use. Privacy and security of information is really important concerning the treatment of user information like delivery addresses..