

**DESIGN AND TESTING 2**

**SPRINT 1**

### 

|  |
| --- |
| **Components** |
| Guerrero Cuenca, Claudia |
| Macarro Klepsch, Miguel |
| Volante González, José Manuel |

|  |
| --- |
| **Content** |
| First the level chosen for our project will be mentioned.  Then show a summary of the full decription of this. Finally, this summary is specified with the different entities and features that will be implemented in our application. |

**Content**

1. Project description
   1. General description of the web application
   2. Entities
   3. User stories
   4. Features to modify in the existing system
   5. Specification with examples
2. Planning for the following sprints

**1.- PROJECT DECRIPTION**

**a.- General description of the web application**

Choice type and level:

Extension of the example provided, level 2 (min. 5 entities + 20 user stories).

Summary:

Our project consists in extending the project provided by the professors. The main features we are going to improve or implement are:

* The way visits are used: Instead of just having the owner or an administrator register the date of a visit, we will implement a fully automated scheduling system. An owner will be able to select a free timeslot with a veterinary of his choice, which will then be registered in the system.
* Payment information for a visit, including credit card data if applicable, will be stored in the system.
* In order to estimate the duration of a visit (for the scheduling) and its cost (for the payment system), we will categorize the visits by type (operation, revision, consultation)
* Veterinarians will be able to add a diagnosis to a visit, including prescriptions for medications.

**b.- Entities**

Entities to modify

**Visit**

Modified attributes:

* date: renamed to moment. Should represent a date with time

New relationships and attributes:

* Vet (many to one)
* VisitType (many to one)
* Diagnosis (one to one, optional)
* Paymnet (one to one, optional

New entities

These are the new entities that will be implemented with their attributes:

**VisitType** 🡪 Name, Price, Duration.

The default types are: consultation (20€, 30 min), revision (15€, 20 min) and operation (100€, 60 min).

**Diagnosis** 🡪 Moment, Description.

Relationships: Prescription (one to many)

**Prescription** 🡪 Frequency, Quantity.

Relationships: Medicine (many to one)

**Medicine** 🡪 Name, Business.

**Payment** 🡪 Date and time, Quantity), Method.

Relationships: CreditCard (one to one, optional), Secretary (many to one)

**CreditCard** 🡪 Holder, Brand, Number, ExpMonth, ExpYear, SecurityCode.

Relationships: None

**Secretary**

Extends from Person and has an associated user. This kind of user is in charge of registering payments.

Relationships: User (one to one)

The relationship between the classes can be seen in the following UML diagram. New classes that we will implement are highlighted in yellow.

../../../../../../../Desktop/model-extended.pdf

**c.- User stories**

User stories that we will implement in our application:

Visit scheduling

(Visit, VisitType, Vet, Pet, Owner)

**Feature**: Schedule an appointment online

**As a** pet owner

**So that** I don’t have to call

**I want** to be able to schedule appointments online.

(Owner, Pet, Visit)

**Feature**: Request a visit with a specific veterinarian

**As a** owner

**So that** I can receive service from a veterinarian that had treated my pet before

**I want** to be able to request a visit with a specific veterinarian.

(Owner, Pet, Vet)

**Feature**: Manage appointments automatically

**As a** clinic owner

**So that** I can save money on scheduling

**I want** the system to manage appointments for visits automatically, including making sure that no appointment is made outside of working hours (8:00 am – 8:00 pm, Monday through Friday) and that no vet has to visits scheduled at the same time

(Visit, Vet)

**Feature**: Select a type of visit

**As a** owner of the clinic

**So that** time and visits are better utilized

**I want** that the owner of a pet can select a type of visit, which has an approximate duration.

(Visit, Owner, VisitType)

**Feature**: Upcoming visits view

**As a** pet owner

**So that** I won’t forget an appointment

**I want** a view that shows the visits I have scheduled I the future

(Visit, Owner)

Payment registration

(Visit, Payment, CreditCard, Secretary)

**Feature**: Register payments of visit

**As a** owner of the clinic

**So that** I can reap all the benefits of a digital payment database

**I want** all payments to be registered in the system.

(Visit, Payment)

**Feature**: Register a payment with credit card or cash

**As a** pet owner

**So that** I can have the freedom of choosing how to pay for my visit

**I want** to be able to pay with credit card or cash

(Visit, Payment, CreditCard)

**Feature**: Store who registered a payment

**As a** clinic owner

**So that** I can make sure no fraudulent payments are registered

**I want** that every payment includes the secretary that registered it

(Visit, Payment, CreditCard, Secretary)

**Feature**: Store prices for visits by type

**As a** clinic owner

**So that** the secretaries know how much to charge a client

**I want** that the price of a visit is calculated based on its type.

(Visit, VisitType)

**Feature**: Suggest payment quantity

**As a** secretary

**So that** I can safe time

**I want** that when registering a payment, the quantity is already filled in in the form with the value of the VisitType.

(Visit, Payment, VisitType)

Diagnosis registration

(Visit, Diagnosis, Prescription, Medicine)

**Feature**: Add diagnosis to a visit

**As a** vet

**So that** I can later print it out

**I want** to be able to add a diagnosis to a visit

(Visit, Diagnosis)

**Feature**: Add prescriptions to a diagnosis

**As a** vet

**So that** I can tell the owner what medicine to give their pets

**I want** to be able to add prescriptions to a diagnosis.

(Diagnosis, Prescription)

**Feature**: Select medicine from database

**As a** vet

**So that** I can make sure I don’t prescribe a medicine that doesn’t exist or is spelled differently

**I want** to select a medicine from a list of medicines stored in the system when registering a prescription

(Diagnosis, Prescription, Medicine)

Medicine registration

(Medicine)

**Feature**: Edit the medicines in the system

**As a** vet or admin

**So that** vets can prescribe new medicines

**I want** add new medicines to the system, or delete ones (as long as they haven’t been prescribed yet)

(Medicine, Prescription)

Admin dashboard

(Visit, Payment, Diagnosis)

**Feature**: See all the characteristics of visits already made

**As a** administrator

**So that** I can check the correct functioning of the clinic

**I want** a view with all the visits already made that includes the features of that visits and a link for their diagnosis and payment.

(Visit, Payment, Diagnosis)

**d.- Features to modify in the existing system**

* When login as Vet, in Owner tab, an exception is raised.
* There are many owners for the same user. Limit to 1. In the list of owners put his personal data.
* An user not authenticated can’t do anything.
* Change logo and add something on the home page.

**e.- Specification with examples**

**Feature**: Schedule an appointment online

**Positive scenario**:

1. An owner is logged in the platform correctly
2. The owner selects one of his pets, a vet, and a type of visit
3. There should be a view that shows a weekly calendar in form of a table with all free slot for the given vet
4. The owner selects a free slot

Result: The visit is stored in the system. It should be visible to the owner in a upcoming visits view. If another or the same owner want to schedule a visit, the slot should be taken.

**Negative scenario**:

1. Log in as owner 1
2. Go to the visit scheduling form and select a pet, visit type and vet.
3. In a different window, log in as owner 2
4. Go to the visit scheduling form and select a pet, visit type and the same vet as in step 2.
5. Select a free timeslot and confirm.
6. As owner 1, select the same timeslot and confirm.

Result: A error message should be shown.

**2.- PLANNING FOR THE FOLLOWING SPRINTS**

(Asignar mas o menos 7 HU a cada persona sin parejas.)

Todos los integrantes de este grupo están de acuerdo a lo descrito:

Sevilla, Febrero 2020

