DP1 2022-2023

Documento de Requisitos y Análisis del Sistema

Proyecto <End Of Line>

## Repositorio: <<https://github.com/gii-is-DP1/dp1-2022-2023-ling-2>>

Vídeo: <<https://www.youtube.com/watch?v=bj8AAYRFOh4>>

# Miembros <en orden alfabético por apellidos>:

* Amstler Nikolaus
* Bonciu Alin
* Gálvez López, Juan Pedro
* Moreno Contreras, Victoria Eugenia
* Mutter, Maren
* Rodríguez Ruiz, Antonio

Tutor: García Rodríguez, Jose María

# GRUPO G<Ling>-<02>

## Versión <01>

<28/09/2022>

## Historial de versiones

*Estos son ejemplo del contenido que debería tener el historial de cambios del documento a entregar a lo largo de los sprints del proyecto*

| Fecha | Versión | Descripción de los cambios | Sprint |
| --- | --- | --- | --- |
| 28/09/2020 | V1 | * Document creation | 1 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

*Esta es una plantilla que sirve como guía para realizar este entregable. Por favor, mantén las mismas secciones y los contenidos que se indican para poder hacer su revisión más ágil.*

# Descripción general del proyecto

*End Of Line is a solo or multiplayer card based board game. The game’s premise is to connect your cards to make a line, while evading your oponent from cutting it. The games last for around 5 to 10 minutes, and it ends when one of the players places down all of their cards, in which case they win, or they cannot place any more cards on the board, in which case they lose.*

*The objective of the project is to implement this game, and some of the gamemodes, including online playing, puzzle mode (single-player). And the possibility to register as a user.*

# Tipos de Usuario / Roles

**Administrator**: An administrator will have control over the platform, and be able to see all games, and moderate the platform.

**Player:** A player will be able to create and play games with other players.

**Spectator:** A player that has joined a game but only to see other people play it as an spectator.

# Historias de Usuario

…

**Mockup de Interfaz de usuario de la historia**

**…**

S<01> - <Multiplayer game creation>

<As a player, I want to be able to create games, so that I can play with other people>

**Positive Scenarios:**

S<01>+E<01> - <Logged in>

**“Given that an user is logged in, when they try to create a game, the game is then created successfully** ”

**Negative Scenarios:**

S<01>-E<01> - <Not Logged in>

**“Given that an user is not logged in, when they try to create a game, they then get a error message”**

S<02> - <Single game creation>

**<**As a player, I want to be able to create single player games, so that I can rest from multiplayer**>**

**Positive Scenarios:**

S<01>+E<01> - <Logged in>

**“Given that an user is logged in, when they try to create a game, the game is then created successfully** ”

**Negative Scenarios:**

S<01>-E<01> - <Not Logged in>

**“Given that an user is not logged in, when they try to create a game, they then get a error message”**

S<03> - <Gameplay>

<As a player I want to be able to play games, so that I can enjoy end of line>

S<04> - <View the created games>

<As a player I want to be able to see all my created games, so that I know how much I have played>

S<05> - <User basic functionality>

<As a player, I want to be able to Log-In, out, or sign up, so that I can have my own user with which I can always play>

**Positive Scenarios:**

S<05>+E<01> - <Logged in>

**“Given that an user is logged in, when they enter a game, then they can play**”

**Negative Scenarios:**

S<05>-E<01> - <Logged in>

**“Given that an user is logged in, when they try to log in, they then get a error message”**

S<06> - <Profile editing>

<As a player, I want to be able to change my profile data, so that I can keep my info updated if anything changes>

**Positive Scenarios:**

S<06>+E<01> - <Valid information>

**“Given that an user has entered a valid value for a specific field or fields, when they try to apply the changes, they then get a success message and are redirected to their user page”**

**Negative Scenarios:**

S<06>-E<01> - <Invalid information>

**“Given that an user has entered a non-valid value for a specific field or fields, when they try to apply the changes, they then get a error message saying what field or fields are invalid”**

S<07> - <Current games control>

<As an admin, I want to be able to see all ongoing games, so that I have information about the number of active players>

S<08> - <Overview of games’ popularity>

<As an admin, I want to be able to see all games ever, so that I can have an idea of the popularity of the platform>

**Positive Scenarios:**

S<08>+E<01> - <Administrator logged in>

**“Given that the administrator can see all the games, when they try to see the games popularity, then they can see the most popular game**”

**Negative Scenarios: -**

S<09> - <Current games control (1)>

<As an admin, I want to be able to make CRUD operations over users, so that I can moderate the platform>

**Positive Scenarios:**

S<11>+E<01> - <Ban user>

**“Given that an an admin is doing moderation tasks, when they try to remove a player, then the player is removed**”

**Negative Scenarios:**

S<09>-E<01> - <Admin deletion>

**“Given that an admin is doing moderation tasks, when they try to apply a CRUD operation another admin, they then get a error message saying they do not have enough permissions”**

S<10> - <Current games control (2)>

<As an admin, I want to be able to see all player data, so that I can moderate the platform>

S<11> - <User statistics visualization>

<As a player, I want to be able to see stats from my games (Number of games, duration of games), so that I can keep track of my improvement>

**Positive Scenarios:**

S<11>+E<01> - <Logged in>

**“Given that an user is logged in, when they try to see the game stats, then they can track their improvement**”

S<12> - <User ranking>

<As a player, I want to be able to see and participate on a ranking system, so that I can compare my skills to others>

S<13> - <User achievements>

<As a player, I want to be able to have profile achievements, so that I can be proud of my skill>

S<14> - <Friend Requests>

<As a player, I want to be able to send friend requests, so that I can send them game invites easier>

**Positive Scenarios:**

S<14>+E<01> - <Friend request to a non-friend user>

**“Given that an user is logged in, when they send a friend request to another user that is not yet their friend, then the request is added to the list of pending friends request of that user** ”

**Negative Scenarios:**

S<14>-E<01> - <Friend request to a friend user>

**“Given that an user is logged in, when they send a friend request to another user that is already their friend, then the request is not send and a error message saying *“This user is already your friend”* appears**”

S<14>-E<02> - <Pending request>

**“Given that an user is logged in, when they send a friend request to another user that has a pending request from the first user, then the request is not send and a error message saying *“There is already a pending friend request to this user”* appears**”

S<15> - <Game invitations>

**<**As a player, I want to be able to send game invites to friend so that they can play with me or spectate my games>

**Negative Scenarios:**

S<15>-E<01> - <Invitation to a non-friend>

**“Given that an user has created a game, when they send a game invitation to a non-friend user, then the invitation is not send and a error message saying *“This user is not your friend”* appears**”

S<15>-E<02> - <Invitation to an already present user>

**“Given that an user has created a game, when they send a game invitation to a friend user that is already present either as spectator or player, then the invitation is not send and a error message saying *“This user is already present in the game”* appears**”

S<16> - <Game spectation>

<As a player, I want to be able to spectate my friends game whenever I want, so that I can take a break from playing myself>

**Positive Scenarios:**

S<16>+E<01> - <Spectate a friend>

**“Given that an user is friend with another user, when they try to spectate them, then they can watch them playing**”

**Negative Scenarios:**

S<16>-E<01> - <Spectate a player who is not friend with>

**“Given that an user is not friend with another user, when they try to spectate them, then they can’t watch them playing**”

S<17> - <Game chat>

<As a player, I want to be able to chat during games, so that I have an extra layer of interaction with my opponents>

S<18> - <Achievements editing>

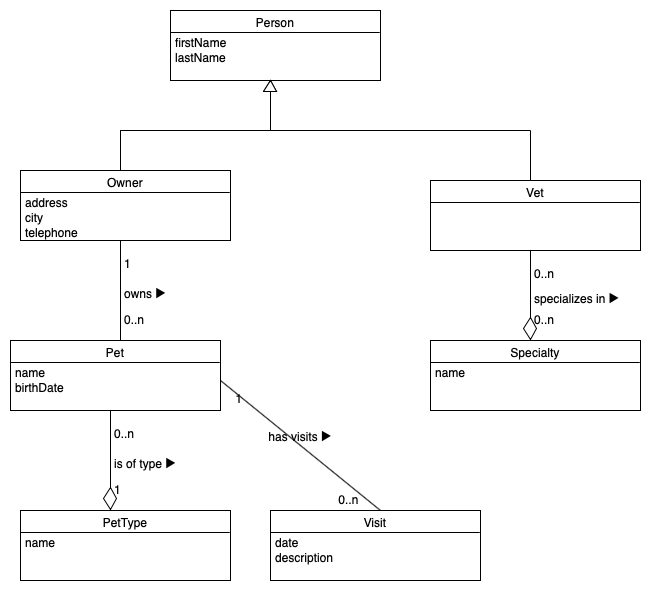
<As an admin, I want to be able to modify criteria for achievements, so that I can adapt them to whatever changes arise>

# Modelo de Datos

*En esta sección debe proporcionar un diagrama UML de clases que describa el modelo de datos a implementar en la aplicación. Este diagrama estará anotado con las restricciones simples (de formato/patrón, unicidad, obligatoriedad, o valores máximos y mínimos) de los datos a gestionar por la aplicación.*

*Si la aplicación es original y se contextualiza en un dominio de aplicación distinto al de la clínica de mascotas, puede formatear el diagrama UML como desee siempre y cuando cumpla las directrices especificadas en el párrafo anterior.*

Ej:



# Reglas de Negocio

R-<X> <Nombre Regla de negocio>

<Descripción de la restricción a imponer>

Ej: *R1 – Diagnósticos imposibles*

*El diagnóstico debe estar asociado a una enfermedad que es compatible con el tipo de mascota de su visita relacionada. Por ejemplo, no podemos establecer como enfermedad diagnosticada una otitis cuando la visita está asociada a una mascota que es un pez, porque éstos no tienen orejas ni oídos (y por tanto no será uno de los tipos de mascota asociados a la enfermedad otitis en el vademecum).*

…

# Planificación

Aquí debe proporcionar una tabla con la asignación de elementos a implementar por pareja y sprint, el primer elemento que aparezca para una pareja será el primero que será abordado durante el sprint correspondiente.

A la hora de realizar la asignación de tareas tenga en cuenta que:

1. En el Sprint 1 debería asignar primero las tareas más simples (creación de entidades aisladas por ejemplo), para facilitar la familiarización del equipo con las tecnologías a usar.
2. Las tareas no pueden abordarse en cualquier orden, sino que existen dependencias entre las mismas. Por ejemplo, no podemos crear las relaciones entre las entidades si las entidades no han sido implementadas, o no podremos implementar las historias de usuario sin las entidades y relaciones que requieren.
3. Es una buena práctica asignar la creación de las pruebas de una funcionalidad a miembros distintos de los que la implementaron, incluso a otra pareja. Sin embargo, al ser pruebas unitarias también es admisible que lo hagan las mismas personas.
4. Es conveniente modificar la composición de las parejas entre un sprint y otro para facilitar la colaboración entre todos los miembros del equipo.

Ej:

| ***Tipo*** | ***Elemento*** | ***Asignación*** | ***Sprint*** |
| --- | --- | --- | --- |
| *Entidad* | *Disease* | *<Satoru Iwata>*  *<*Eiji Aonuma*>* | *1* |
| *Relación 1:N* | *Disease-Diagnose* | *<Satoru Iwata>*  *<*Eiji Aonuma*>* | *1* |
| *Historia de Usuario* | *H1-1* | *<Satoru Iwata>*  *<*Eiji Aonuma*>* | *1* |
| *Regla de Negocio* | *R1* | *<Satoru Iwata>*  *<*Eiji Aonuma*>* | *2* |
| *Prueba* | *H1+E1* | <Hidemaro Fujibayashi>  <Koji Igarashi> | *2* |
| *Entidad* | *Diagnose* | <Hidemaro Fujibayashi>  <Koji Igarashi> | *1* |
| *…* | *…* | *…* | *…* |

# Métricas del proyecto

| **Métrica** | **Valor** |
| --- | --- |
| **Nº de entidades (excluyendo actores)** | *<X>* |
| **Relaciones** | *<R>* |
| **Relaciones N:N** | *<Z>* |
| **Restricciones Simples** | *<V>* |
| **Reglas de Negocio** | *<W>* |
| **Historias de Usuario totales** | *<U>* |
| **Historias de usuario involucrando 2 o más entidades** | *<P>* |
| **Historias de usuario involucrando 3 o más entidades** | *<Q>* |
| **Actores** | *<Y>* |

Módulos extra que se pretenden desarrollar: <X>. Ej: Módulo de estadísticas.