## Rent a Dog - Project Overview

## Introduction

The company is a project Group for the CSIS3275 – Software Engineering class. Founded in 2022, in the city of New Westminster by Andre, Alexandre, Armando and Mateus.

## 1. Problem Statement

Following a survey from GfK Global Study (2016)[[1]](#footnote-2), 33% of the Canadian population have dogs. If we consider the same amount for British Columbia, that is almost 2 million people or over 700 thousand households with dogs in their lives (BC Stats, 2021).

These people are mothers, fathers, grandparents, and children with their own responsibilities and tasks during the day and, therefore, may not always have time to give the proper attention to their dogs. It is common knowledge that dogs require attention during the day, especially the ones livings in condominiums or big breeds such as Border Collie, Australian Shepherd, Labrador Retriever, and Siberian Husky that are very highly energetic and frequently need to go for a walk or even a run (Ultimate Pet Nutrition Team, 2022).

Thinking about these people and their pets, our company produced a revolutionary idea, in which it is possible to rent your dogs for a few hours a day. So, when you are on a super busy day, feeling tired after a long work journey, or have a pre-scheduled appointment, you can submit in our system your dog availability. Another reason people would want to rent their dog is to get some extra cash at the end of the month, especially considering owning a dog in Canada costs, on average, $3,724 a year (PHI Direct, 2022), which can be a bit overpriced, to low-income families. Renting your pet can be a terrific way to help to pay those expenses and let your dog have a new experience!

On the other hand, there are millions of people in the province without pets due to several reasons, their tenancy agreement does not allow it, or they travel too frequently to have such a responsibility of taking care of another life in their hands. This way, we can remedy this lack of a companion in their life by renting a dog for a few hours to walk or play with.

So, our system can tackle both problems, allowing people to rent their own dogs for a fee of their choice and making extra money for their income, and allowing other people to have loyal friends for a few hours.s.

## 2. Functional Requirements

* User will need to register himself
* User can sign in to rent(offer) a dog or rent(get) a dog
* User can search for a dog
* Application will start and finish rent
* User can input data about the dog
* Dog owner can select available days to rent a dog
* User (renter) will be able to choose the amount of time to rent a chosen dog
* The same dog cannot be rented to a different people in a single day
* Only user above 19 years old will be allowed to use the system
* The application will track the dog location (version 2)
* The application will call emergency if the dog is out of the specified range (version 2)

## 3. Non-Functional Requirements

* Integrability: New features that will be added should not affect the current developed features
* Scalability: Backend will be scalable vertically and horizontally to support more users
* Stability: Service will not be out of service more than 5 minutes (Unless notified 48 hours in advance)
* Response time: System will not take more than 5 seconds to respond users’ request
* Accessibility: User interface will be intuitive and friendly for users

## 4. Rules

* We will charge 20% for each hour booked through our system and issue a monthly invoice
* We are not responsible for the dog in an emergency
* The user will accept the signing of the contract on the platform
* The first version will not track the dog
* Owner will be responsible for scheduling drop off and pick up, location will be renter's registration address saved in our system on sign in

## 5. Use Cases Diagram

Diagram

Description automatically generated

## 6. Revenue Models

**Transaction Revenue Model:** Considering the rental model, the most appropriate system revenue is revenue on each transaction. After brief research on the solutions that exist today, such as Uber, DoorDash, etc., the project team chose to set 20% of the total rental value as a service and support fee. This income will be enough to cover operational, sales and marketing expenses, as well as ensure the perpetuity of the business. The entire revenue projection is detailed in the Revenue Projection step.

## 7. Business Case

Introduction

Recently, a publication brought up an important topic: Dog Rental[[2]](#footnote-3). Emphasizing what has already been mentioned above, there are two needs that can be met with a software/service solution: On the one hand, dog owners without the time to take their canines for a walk or have fun, and on the other hand 67% of Canadians who do not have dogs, but who could have.

**Initial Investment**

Considering the development, the team estimated the costs below:

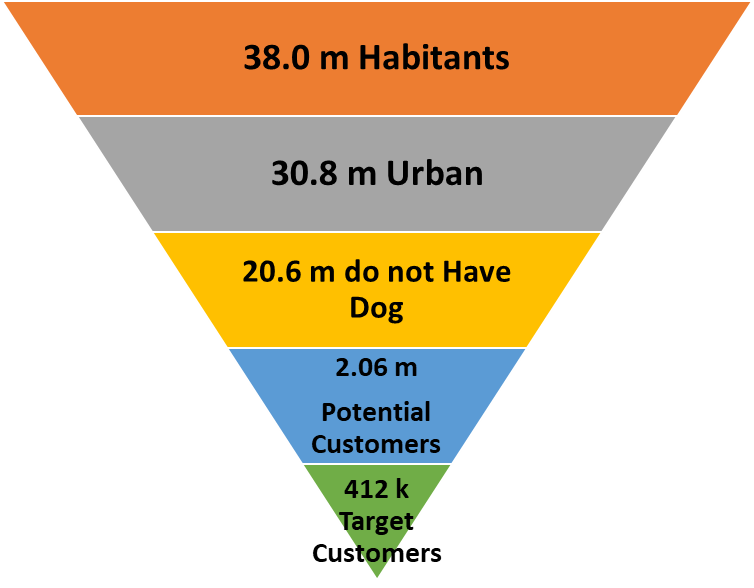
|  |  |  |  |
| --- | --- | --- | --- |
| **Investment** | **Hours** | **Hour Cost** | **Total CAD** |
| Developer | 1800 | 80.00 | $ 144,000.00 |
| Quality Check | 180 | 80.00 | $ 14,400.00 |
| Project Manager | 180 | 120.00 | $ 21,600.00 |
| **Total** | **2160** |  | **$ 180,000.00** |

With a group of 6 developers (1 SR, 2 Full and 3 JR), the project will have 2 months of duration. Furthermore, the complete team will have 2 testers and 1 PMO.

**Revenue Forecast**

The total population of Canada is estimated at 38 million, those 81% are living in urban areas. As mentioned before, 33% have dogs and 67% do not. The total number of urban people that do not have dogs is around 20.6 million, which is our population target. According to an internal estimative, about 10% of the population target could be our customer target (2 million).

To sum up, the team understands that it is possible to have 20% of the Market Share in the Customer Target. These customers will experience a gradual increase through the first 5 years of operation.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Year** | **1** | **2** | **3** | **4** | **5** |
| Avg. Customer | 10,000 | 25,000 | 40,000 | 55,000 | 80,000 |
| Ticket/Price | $ 25.00 | $ 25.00 | $ 25.00 | $ 25.00 | $ 25.00 |
| Uses by year | 24 | 24 | 24 | 24 | 24 |
| Company Fee | 20% | 20% | 20% | 20% | 20% |
| **Total Revenue** | **$ 1,200,000** | **$ 3,000,000** | **$ 4,800,000** | **$ 6,600,000** | **$ 9,600,000** |

Based on the 412 thousand customer target, in year 5 the company will have 20% of this market.

**Costs and Expenses**

Each line of the Forecast has its rational and will be described below.

Maintenance: Was estimated around 40 hours a month for system maintenance.

Cloud Services: For having the system running, the clous services include the operation of the system and was estimated in CAD 6.00 a year by each Customer.

Operational/Support Team: For support, each member of this operational group will manage 10 rentals a day, and the increase in headcount is related to the number of customers.

Administrative Team: Will be responsible for managing the operational team, sales, Marketing and Finances.

Marketing: Usually one of the main expenses, in this case, the company estimate the Cost of acquiring a new customer at CAD 25 each of them.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Year** | **1** | **2** | **3** | **4** | **5** |
| Maintenance | 38,400.00 | 38,400.00 | 38,400.00 | 38,400.00 | 38,400.00 |
| Cloud Services | 60,000 | 150,000 | 240,000 | 330,000 | 480,000 |
| Operational / Support Team | 320,000 | 800,000 | 1,280,000 | 1,760,000 | 2,560,000 |
| **Total Costs** | **418,400** | **988,400** | **1,558,400** | **2,128,400** | **3,078,400** |
| Administrative Team | 240,000 | 600,000 | 960,000 | 1,200,000 | 1,200,000 |
| Marketing | 250,000 | 625,000 | 1,000,000 | 1,375,000 | 2,000,000 |
| Taxes | 150,000 | 375,000 | 600,000 | 825,000 | 1,200,000 |
| **Total Expenses** | **640,000** | **1,600,000** | **2,560,000** | **3,400,000** | **4,400,000** |
| **Total Cost + Expenses** | **1,058,400** | **2,588,400** | **4,118,400** | **5,528,400** | **7,478,400** |

**Executive Summary - Financial Statements**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Year** | **1** | **2** | **3** | **4** | **5** |
| Total Revenue | $ 1,200,000 | $ 3,000,000 | $ 4,800,000 | $ 6,600,000 | $ 9,600,000 |
| Total Cost + Expenses | $ 1,058,400 | $ 2,588,400 | $ 4,118,400 | $ 5,528,400 | $ 7,478,400 |
| PROFIT | $ 141,600 | $ 411,600 | $ 681,600 | $ 1,071,600 | $ 2,121,600 |

The company will be profitable from year 1. Based on all assumptions described before, the team project calculates the NPV (Net Profit Value) for the first 5 years of operation.

To be more conservator, the Risk tax of 10%-year was added to a normal 5%-year tax to calculate the present value of the cashflow.

To sum up, the **NPV is CAD $2,060,890.**

**7. Design Goals**

* Delivery a good service throughout our platform to allow both renter and rentee to use the system whenever they need
* Allow renters to register themselves and their dogs and define the hourly price that the dog will be rented
* Allow rentees to register themselves, search for a dog and sign a contract to rent a dog
* Control the start and end for each contract
* Offer a stable and reliable application to our customers.

**8. Architectural Design**

Our application will have a first version to attend customers in early stages and a final version to accommodate all the requirements.

**For the Version 1, the following architecture will be implemented:**

Diagram

Description automatically generated

This architecture follows MVC model architecture.

A client implementation is based on HTML, CSS and Bootstrap technologies. This web client will communicate with View layer in server side. View communicates with Controller layer that will choose the correct path based on action required.

A security component was added in Controller layer to handle password and login confirmation process. Security component uses **pbkdf2 with hmac--sha256** password storage schema, with password and salt cryptographic to ensure that user data will be secured.

Controller layer communicates with Model that will handle and get information from MySQL database. In Model two bigger components were created to store information specific for each component, for this reason we will have two databases related to Register functionalities and another for Contract and payment process.

All the system is implemented using SpringBoot framework for the backend with MySql Database. For the frontend Thymeleaf was used.

**For the Version 2 this will be our target architecture:**

Diagram

Description automatically generated

For the version 2, we will follow the Microservice Architecture and Clean Architecture Model. The entry point of our system will be an API gateway that will expose our APIs to the clients, in this way our clients can run in a web page, in a mobile device, or in any other device. This can increase our client range in different platforms, resulting in a market expansion with less effort.

API gateway will receive the request through an APIRest request. After receiving this request, Gateway will redirect the credentials to a security microservice that will check if user is allowed or not to access the system. If not allowed, Gateway will reject the access to the user. The same password schema storage from version 1 will be used. After confirmation that user can access the system, Gateway will redirect to the correct microservice to process the request. Microservices are divided to guarantee that scope is in the right place. In this case we will have 2 microservices, one for registration and another for contract and payment. With different microservices, we will be able to escalate only the service that is more required in that moment, saving resources and money. Inside of each microservice we will apply Clean Architecture that allow us to keep system modularized and facilitate maintainability.

A communication with an external payment system will be implemented to keep payment process automated. Stripe in under analysis for payment API. (https://stripe.com/en-ca/payments)

For implementation we will follow S.O.L.I.D. and Clean Code principles.

Diagram

Description automatically generated Diagram

Description automatically generated

**9. Hardware and Software Configuration**

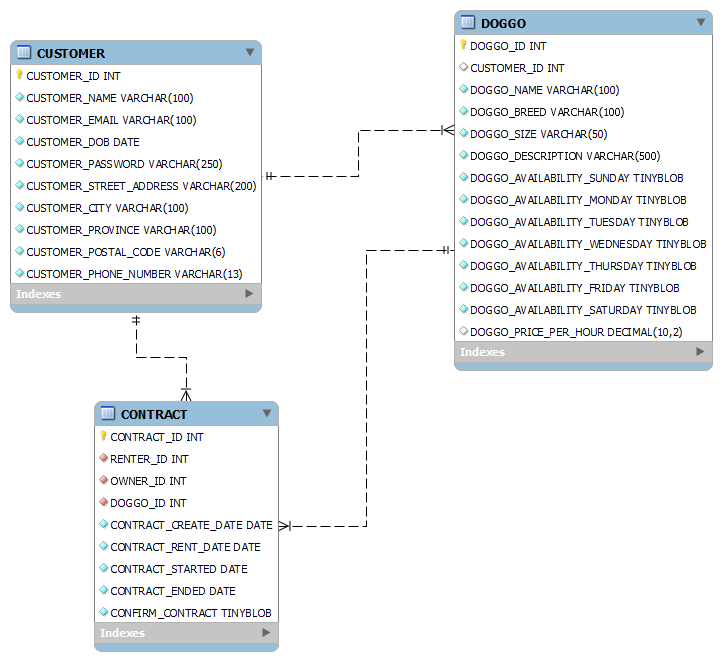
### Software

* The application will be web based, so the only software requirement is a web browser

### Hardware

* Any computer or smartphone/tablet connected to the Internet.

**10. Database Design**

****

**11. Interface Design**

Diagram

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, website

Description automatically generated

Graphical user interface, application, website

Description automatically generated

Graphical user interface, application

Description automatically generated

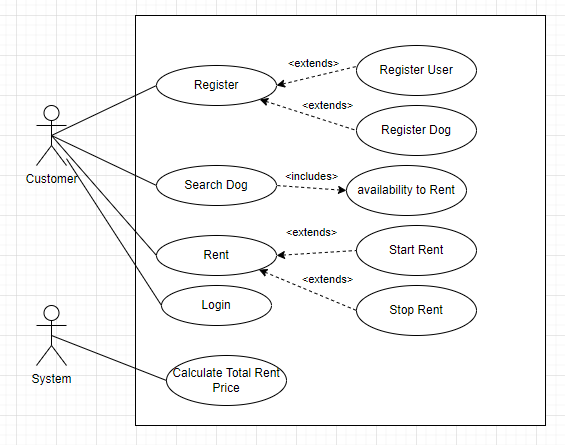
Graphical user interface, text, application

Description automatically generated

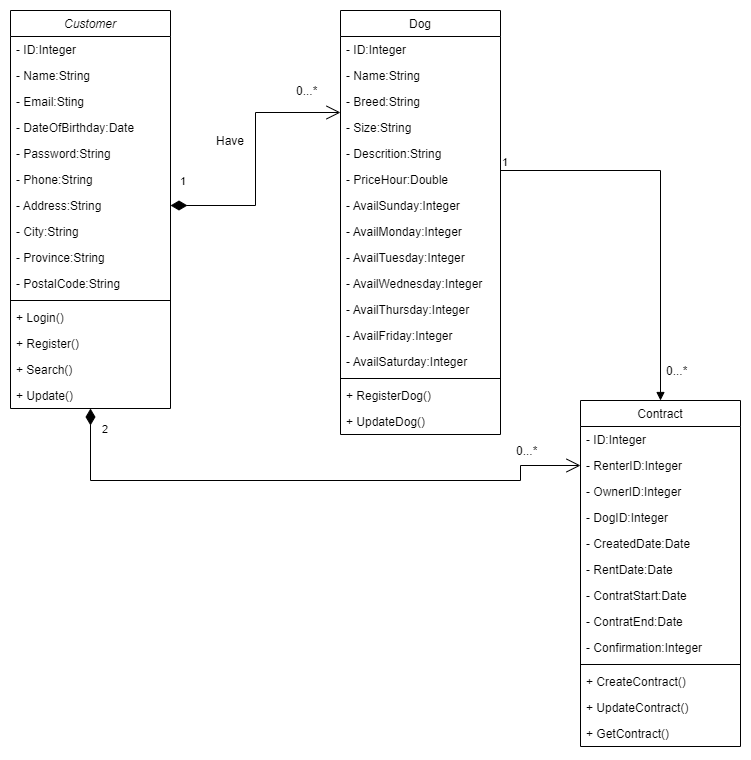
**12. Achievement of Non-Functional Requirements**

* **Integrability**
  + New features that will be added should not affect the current developed features
  + With creation of modules and microservices, new implementation will not affect the system running.
  + Regression tests will be applied in any new added implementation.
* **Scalability**
  + Backend will be scalable vertically and horizontally to support more users
  + With microservices, system will scalability only necessary microservices, saving resources and money.
* **Stability**
  + Service will not be out of service more than 5 minutes (Unless notified 48 hours in advance)
  + System can be replicated in the cloud to running more than 1 instance. This guarantees that at least one service will be running all the time.
* **Response time**
  + System will not take more than 5 seconds to respond users’ request
  + System will not take more than 5 seconds to respond users’ request. To achieve this, all the use cases are implemented straightforward, with low external dependencies.
* **Accessibility**
  + User interface will be intuitive and friendly for users using principles of UI development.

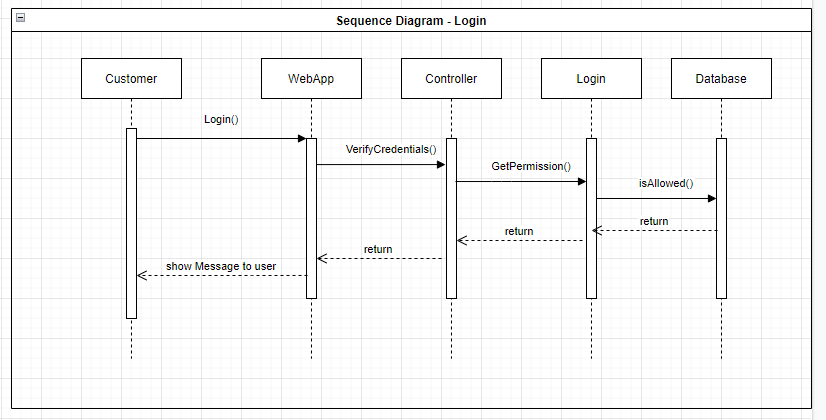
**13. Revised Use Case Diagram**



**14. Class Diagram**

****

**15. Sequence Diagram**



A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

A picture containing text, indoor

Description automatically generated

Diagram

Description automatically generated

1. GfK Global Study, 2016 | BC Stats, 2021 | Ultimate Pet Nutrition Team, 2022 | PHI Direct, 2022​ [↑](#footnote-ref-2)
2. https://dailyhive.com/vancouver/vancouver-public-library-canine-poetry-in-parks-dogs-revealed [↑](#footnote-ref-3)