

T1A3 - TERMINAL APPLICATION PET SITTER APP

Student ID: CAM012112

Student name: Vanessa Denardin

TABLE OF CONTENTS

1. TERMINAL APP – FEATURES AND HOW TO USE
2. LOGIC
3. DEVELOPMENT PROCESS
4. CHALLENGES
5. FAVOURITE PARTS
6. CODE OVERVIEW

1. TERMINAL APP

- The app is a database where a pet sitter can store various information, such as the name and contact of the pet owner, pet's special needs, dates and to-do lists for the service.
- Login screen - user and password
- Menu - pet sitter profile, client's info, job (date and tasks to do), exit
- Clients' info screen allows to add, edit and delete pet's info
- Job screen allows to add, edit and delete date and task's list

1. TERMINAL APP – FEATURES

- Welcome login
- Caregiver and clients (pet's owners) information
- Pet's details
- Job order
- Task list for each job order
- Log out

EXTRAS

- Contact's list
- Mark tasks as completed
- Financial management
- Availability calendar
- Pictures of the pet
- Email notifications
- Message chat

1. TERMINAL APP – TARGET AUDIENCE

- App developed to help a single pet sitter to manage their clients' pets and tasks data
- First version intended to only replace manual data entry from pet sitters, not to be a multi-user system and/or having any financial management
- The app will help the professional caregiver to keep records of all the clients with important information and pet's needs including tasks related to support the pet well-being while the owners are away.

2. LOGIC

- User is prompted with a welcome screen and login page for accessing the data
- A menu is then shown for selection of actions that could be:
 - *Pet sitter profile*
 - *Clients*
 - *Jobs*
 - *Exit*
- On the profile page, the user can setup its own information to show on the Job page for future printing out to help pet sitter
- Client's page will show a list of all existing clients with possibility of add, edit or delete entries. Further down on client's details, pet's information can also be added, edited or deleted
- Job's page will initially list all upcoming gigs for the next 7 days or can also look into all records. Jobs are made up of pet sitter/clients/pets' details and a list of tasks that are required to complete for that order.
- All menus always offer the option to go back one menu, return to home or exit application.

3. DEVELOPMENT PROCESS

- Prepare GitHub project with Kanban board to keep track of all work required and identified during brainstorms
- App flow brainstorm on paper including how screens will look like
- Classes including data structure that would be required to save the information that the user needs
- CRUD for all classes including a dedicated Database class to convert Hashes into JSON and follow the DRY principle (avoid duplication of code)
- App interface to implement all CRUD methods

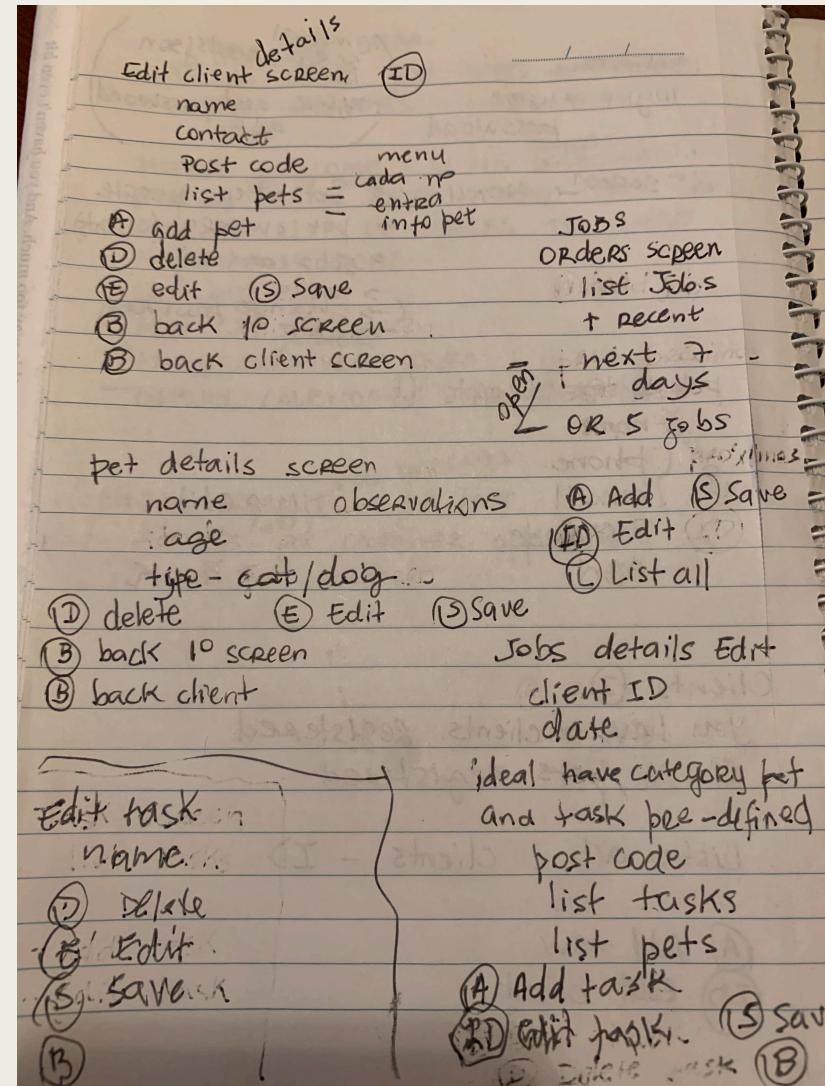
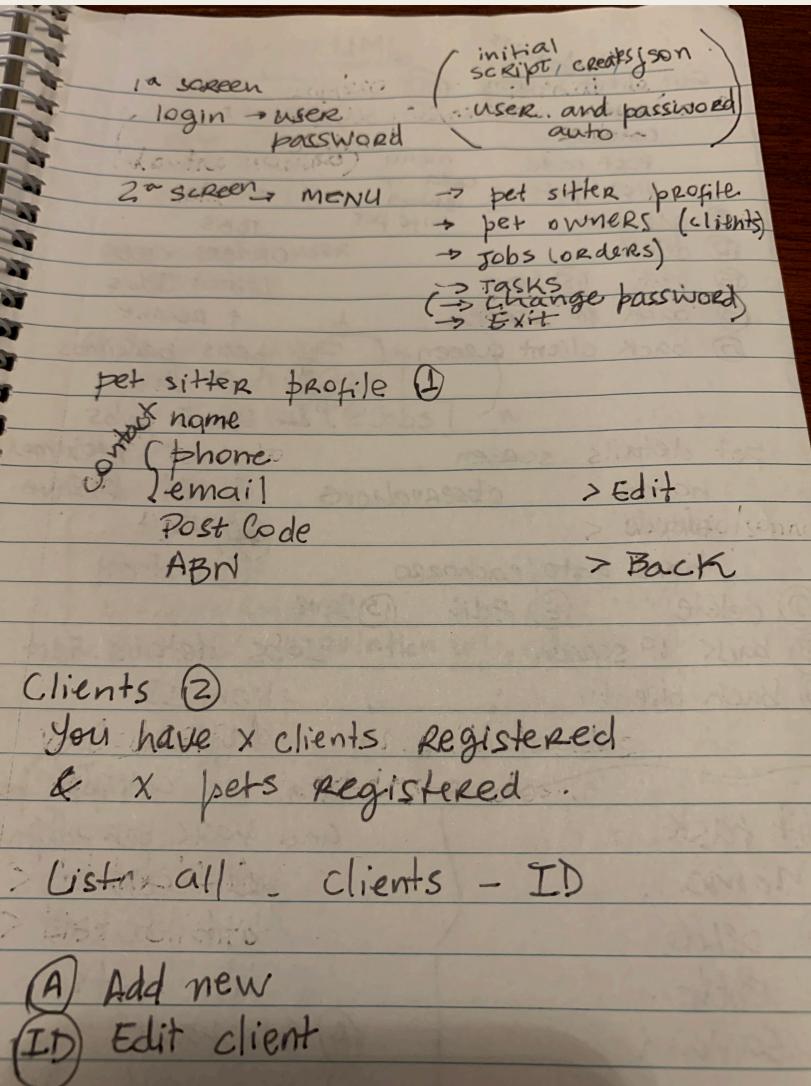
3. DEVELOPMENT PROCESS

The screenshot shows a GitHub project board for the repository `vanessadenardin / pet-sitter-app`. The board is titled "App development" and was updated 3 hours ago. It features four columns: "To do", "In progress", "Review/Testing", and "Done".

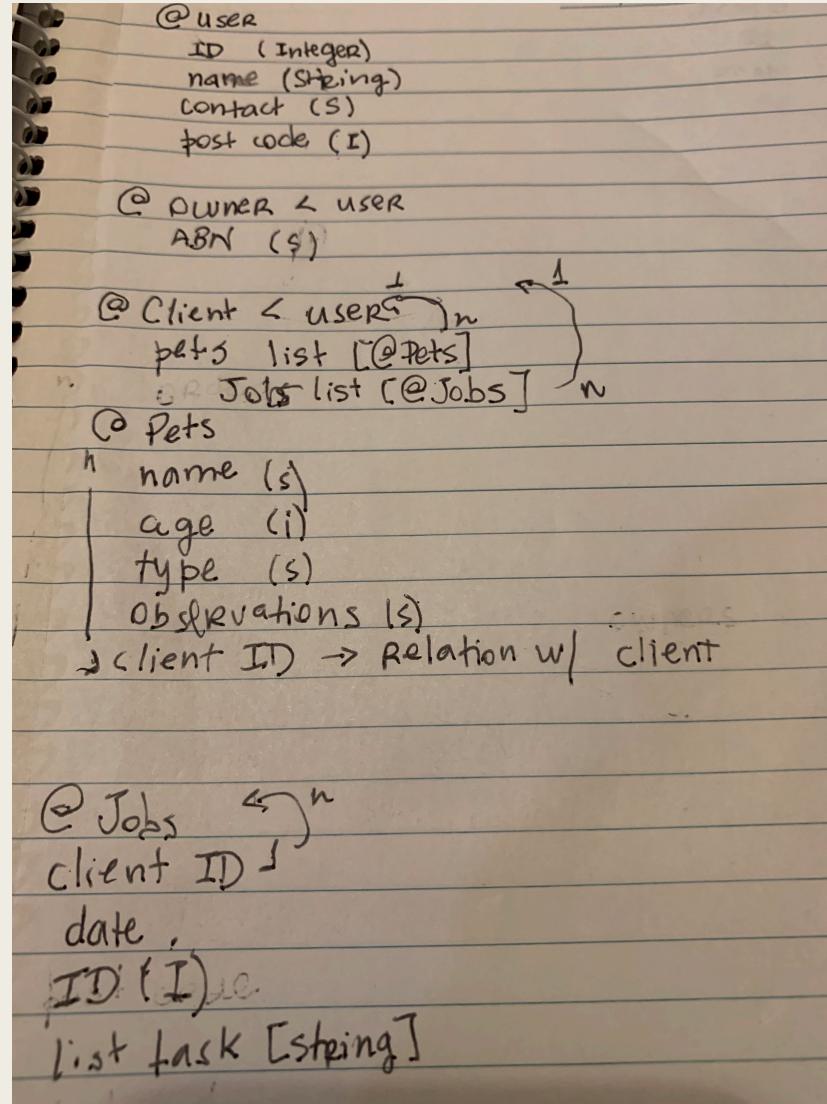
- To do:**
 - Create
 - uml diagram
 - control flow
 - flowchart
 - Extras
 - Financial control
 - Contact's list
 - Mark tasks as completed
 - Availability calendar
 - Pictures of the pet
 - Email notifications
 - Message chat
 - shell script
- In progress:**
 - gemoji (<https://github.com/github/gemoji>)
 - asciart (<https://rubygems.org/gems/asciart/versions/0.0.9>)
 - artii (<https://rubygems.org/gems/artii/versions/2.1.2>; <https://github.com/miketierney/artii>)
 - rmagick (<https://radek.io/2015/06/29/catpix/>)
 - colorize (<https://rubygems.org/gems/colorize>; <https://github.com/fazibear/colorize>)
 - rainbow (<https://rubygems.org/gems/rainbow>)
 - emoji (<https://github.com/wpeterson/emoji>)
- Review/Testing:**
 - CRUD test cases
 - rspec tests
 - Define how many classes are required
 - Petsitter
 - Pets
 - Clients - Pets owners
 - Job order - Tasks list
 - Database
- Done:**
 - Develop CRUD methods
 - add (create)
 - get - get_all, get_by_id - (read)
 - edit (update)
 - delete
 - Define at least 3 features for app
 - Welcome login
 - Caregiver and pets owners info
 - Pets info
 - Jobs - Tasks list
 - Log out
 - Define how screens look like
 - Added by vanessadenardin

<https://github.com/vanessadenardin/pet-sitter-app/projects/1>

3. DEVELOPMENT PROCESS



3. DEVELOPMENT PROCESS



3. DEVELOPMENT PROCESS

■ TODO:

- *Implement all menu functionality*
- *Improve all unit tests*
- *Create script to setup the application with things like install gems and generate initial json file*
- *Document everything*

4. CHALLENGES



Defining classes



Handling errors



Interaction with an
external file – data
file



DRY code

5. FAVOURITE PARTS



Choosing what to do
(brainstorming idea)



Planning app
(GitHub Projects)



Designing screens
layout



Ruby gems



Overcoming
challenges

6. CODE OVERVIEW



THANK
YOU!