Software Requirements Specification

for

PetMatchMaking

/social network for pets - web application/

Prepared by:
Georgi Marinov

Edited by:
Yasen Ivanchev
Juliy Geshev
Budin Romanski
Atanas Katsarov
Miroslav Mitrev
Milen Stankov
Ilia Petrov
Dilyana Dangova

DevNetTeam

Version 3 /*approved*/
Version date: 11.03.2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
SRS-DNT-V1.0	28.02.2020	Additional specification of functional requirements	Version 1
SRS-DNT-V2.0	06.03.2020	Adding actualized user interfaces	Version 2
SRS-DNT-V3.0_Approved	10.03.2020		Version 3

1.   Introduction

1.1   Purpose

The purpose of this document is to give a detailed description of the requirements for the "**PetMatchmaking**" web application ("**web app**") Version #3, a part of the "**PetMatchmaking**" application "**app**". The latter is to consist of the above mentioned **web app** as well as of a mobile application ("**mobile app**"), which is going to be released at a later stage of the overall project and together with the web app will constitute the final long-term strategic product vision. The current document will illustrate the purpose and complete declaration for the development of the web app. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to the customer for his final approval and a reference for further developing of the current #3 version of the system for the development team.

1.2   Document Conventions

For a reference of any non-defined currently terms, acronyms, or abbreviations, please see Appendix A: Glossary.

Wherever in the document a blue colored font is placed, the respective requirement / functionality specifies a high level of significance of the requirement / functionality, or to the future thorough vision for the product. Wherever in the document a **blue bolded** colored font is placed, the respective requirement / functionality specifies a highest level of significance of the requirement / functionality, or to the future thorough vision for the product.

This is initial prioritization of the app's functionalities stressing more on the perspective further development and improvement of the product. This means that the colored requirements are more likely to form the future final product's core and to be recognized as the differentiation opportunities to be positioned towards potential and current users and against the competitors from a marketing point of view. This does not mean that the non-colored requirements are of low priority for the current product development stage.

1.3   Intended Audience and Reading Suggestions

The current document is intended for different types of reader, such as developers, project managers, marketing staff, users, testers, and documentation writers. Therefore, we would suggest that the reader scroll through the document contents first and proceed through the sections that are most pertinent from the perspective of his/her specific functions and role.

We would suggest that testers take into account the system features and the corresponding functional requirements stated in section 3 and section 5 when building the test plans and defining the test cases.

The user interfaces are scrutinized after the functional requirements and suggest in our opinion an acceptable possible user-friendly experience, representing in the same time a well organized and thorough user manual, which can be of course additionally improved.

1.4   Project Scope

The scope of the project is the "**PetMatchmaking**" web app, representing a social network for pets and their owners. Its main focus is to help people establish communication on the basis of their love and affection towards pet-animals. There are 3 main directions to which the app is targeting:

1)   **matchmaking**, or finding the best pet-partner for users' pets based on the users' residence – permanent or temporary location and other criteria like breed, age of their pet and more;
2)   **meeting** with other people, who have relevant pets' affection and/or similar interests connected with animals; this could be common walks, common gatherings, chats, etc.;
3)   looking for **help** from other people or offering help to the latter. This could be help with temporary walking the pet, hosting the pet, buying special food for pet, taking care for pet (e.g. asking for help by elderly people being not able to take the full necessary care for their pet);

The criteria on which the users[1] could potentially meet, match-make or help each other is defined in the app by the definition of the user' profile characteristics as well as on his/her pets' description.

The app could also potentially help people to find the best pet to adopt, abiding by the same criteria.

The web app is free-accessed via internet, (to be) based in a datacenter.

Logged-in users can provide their personal information and information about their pets. This information will act as the bases for the search results displayed to the user.

An administrator uses the app in order to administer the system and keep the information accurate. The administrator can, for instance, verify pet owners and manage user information.

The app needs internet connection to fetch and display results. The app does not need GPS connection, since the user information is not real time but predefined generally.

All system information is maintained in a database, which should be located on a web-server. The application also has the capability of representing both summary and detailed information about the pets, their owners and the perspective pet sitters / adopters.

The current version of the app is only in English language.

1.5   References

This document refers to:

- *Maintenance and Evolution for Pet MatchMaking Application document*
- *Planguage definitions, see:*
  https://seilevel.com/requirements/specifying-quality-requirements-with-planguage
- *Certified Tester Foundation Level Syllabus Version 2018 V3.1*
- *see also Appendix A: Glossary in this document*

There are terms and abbreviations that are explicitly defined in this document and others, whose meaning is predefined in the above listed sources.

If there are discrepancies between the usual and/or general definition of a terms explicitly defined in the current document, the definition of the term in the current document should be applicable.

2.   Overall Description

2.1   Product Perspective

The app is a new, self-contained product. The system will consist of web app and a database.

The **web app** will be used to input, extract and manage information, including user information, information about the pets, their owner(s), user-rights, as well as system features.

Since this is a data-centric product it will need somewhere to store the data. For that, a **database** will be used. The web app will communicate with the database. All of the database communication will go over the Internet.

2.2   Product Features

With the app, the users will be **able to** search for pets for mating purposes (matchmaking), look for event connected with pets, organize events connected with pets, ask for help connected with pets or offer help connected with pets.

The **result** will be based on the criteria the user inputs. There are several search criteria and it will be possible for the administrator of the system to manage the options for those criteria.

The result of the search will be viewed either in a **list view** or in a **map view** and will be accompanied by **text** as well as by a **picture**, if such is stored.

---

[1] About definition of the term "**users**" and the different types of users in the current document, please see **Appendix A: Glossary**

2.3   User Classes and Characteristics

There are three herewith defined types of users that interact with the system: Non-registered user, logged-in users and administrators[2]. Each of these three types of users has different use of the system so each of them has their own requirements.

The **Non-registered users** can only use the application to see the Home page, see the Info page, read articles if such are placed there, use the Contact us form.

The **Logged-in users** will be able to use the functionalities accessed by Non-registered users, also to use the search features, to manage their account by placing and editing user data, see published requests, read the information in the requests including seeing the uploaded photos in the requests. In order for the users to get a relevant search result there are multiple criteria the users can specify.

The **administrators** are managing the overall system so there is no incorrect information within it. An administrator can manage the information for each request/announcement as well as to specify the drop-down menus by adding items there.

2.4   Operating Environment

The app operates via any kind of web browser. The app could be accessed on any kind of devices, incl. mobile ones, which can use web browsers.

A mobile app is to be released on a later stage of the thorough product development.

2.5   Design and Implementation Constraints

2.5.1   Design

The system design consists of Front end and Back end, including Database:

- The **Front end** includes HTML, Bootstrap and CSS;
- The **Back end** includes PHP and MySQL.
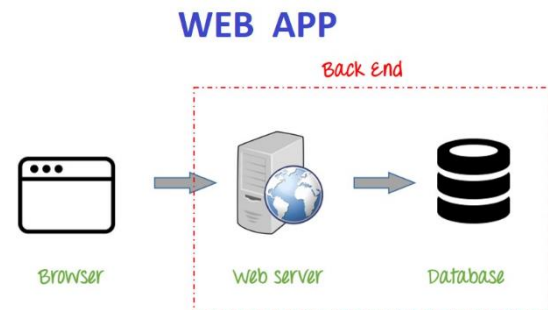
2.5.2   Database requirements and schema

The Database is relational and consists of tables with primary and foreign keys, defined with accordance to the relations between users', pets and requests data.

The Database is constructed by using MySQL. The sensitive user-data – like e-mail addresses and passwords is being encrypted.

Database minimum requirements:

MySQL 5.1.+, Server type: MariaDB,MyISAM or InnoDB - storage engine, PHP version: 7.2.+

optional phpMyAdmin 4.8.+

important: Server charset: UTF-8 Unicode (utf8) or UTF-8 Unicode (utf8mb4)

---

[2] see also **Appendix A: Glossary**

Database schema:

budaninj_petmatchdb users
- userID : int(11)
- userEmail : varchar(64)
- userPassword : varchar(64)
- userFName : varchar(64)
- userLName : varchar(64)
- userPhone : varchar(64)
- userBirthday : date
- userAbout : text
- userRegisterTime : datetime
- userStatus : varchar(64)

budaninj_petmatchdb rankings
- rankID : int(11)
- rankingUserID : int(11)
- rankedUserID : int(11)
- helpID : int(11)
- rankRank : int(11)
- rankAbout : text

budaninj_petmatchdb pets
- petID : int(11)
- userID : int(11)
- petName : varchar(64)
- petType : varchar(64)
- petBirthday : date
- petSex : varchar(64)
- petFood : text
- petTemper : text
- petAbout : text
- petStatus : int(11)
- petRegisterTime : datetime

budaninj_petmatchdb interested
- helpID : int(11)
- interestedUserID : int(11)
- status : int(11)

budaninj_petmatchdb helprequests
- helpID : int(11)
- userID : int(11)
- helpLocLat : float(10,6)
- helpLocLng : float(10,6)
- helpLocCity : varchar(64)
- helpLocStreet : varchar(64)
- helpLocNum : varchar(64)
- helpLocCountry : varchar(64)
- helpRegisterTime : datetime
- helpStartTime : datetime
- helpEndTime : datetime
- helpPayment : text
- helpAbout : text
- helpType : varchar(64)
- userPhone : varchar(64)
- helpStatus : int(11)

budaninj_petmatchdb pets_helprequests
- petID : int(11)
- helpID : int(11)
- userID : int(11)
- recordRegisterTime : datetime

2.5.3   Constraints

The app will be constrained by the **capacity of the database**. The database may be forced to queue lots of incoming requests and therefore increase the time it takes to fetch data.

The **Internet connection** is also a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function.

The **legal constraints** for using the app are connected with the exposure of personal data in the app and the respective user consent that should be ensured regarding the local personal data legislation, as well as General Data Protection Regulation (GDPR).

2.6   User Documentation

The user documentation consists of the current SRS document, the User manual, the Maintenance and Evolution document

3. System Features and Functional requirements

3.1 User Class 1 – Non-registered user functionalities

Overview: Non-registered users should be able to see the Home page, the Info section with all its pages and the Contact us page and to read the content of these sections and pages.

3.1.1 ID: FR1

DESC: Given the user has entered the app
Then the user should be able to see the Login form

3.1.2 ID: FR2

DESC: Given the user has entered the app
Then the user should be able to see a Sign-up link and the user should be able to enter a Sign-up form

3.1.3 ID: FR3

DESC: Given the user has entered the app
Then the user should be able to reach the Info section and the user should be able to read all the content of the Info section

3.1.4 ID: FR4. Feature: Contact us form

Overview: Non-registered users should be able to send messages using the Contact us form.

DESC: Given the user has entered the app and the user has entered the Contact us form
When the user wants to send a message
And the user has entered Message title and Message content and Name and an e-mail address
Then the user should be able to send a message and to see a text verifying the successful sending.

3.1.5 ID: FR5. Feature: Create an account

Scenario: Required information for registration
Given the user wants to create an account
And the user does not have an account for the given username / e-mail address
When the user selects the sign up link And the user provides a first name And an e-mail address and a password And successfully confirms the correct password,
Then the user should be able to create an account and receive a verification message;
Scenario: Confirmed registration
Given the user has receive a verification message
Then the user should be directed automatically to the Requests page of the app.

3.2 User Class 2 – Logged-in User

Overview: Logged-in Users should be able to see and use the above-described sections of the app which are visible and usable by non-registered users (*see art. 3.1*) except FR1, FR2 and FR5, as well as to see and use the below described functionalities applicable for Logged-in User.

3.2.1 ID: FR6. Feature: User log-in

Overview: In order to use all the functionalities of the app applicable for logged-in users, a user should be logged in to the app.
Scenario: Successful log-in
Given the user wants to log in
When the user enters the app and the user provides a username And a password
If the username and password are existing and corresponding and correct
Then the user should be logged in to his/her account and directed automatically to the Requests section.

3.2.2   ID: FR7. Feature: Profile page – edit account's information

DESC: In order to manage (add, edit or delete) information, a user Should be logged in to the app
Given the user is logged in and the information exists
When the user wants to edit his/her account information
Then the user should be able to edit the information in a form.
Scenario: Delete information in mandatory fields
When the user deletes the information from a mandatory field
And the user does not enter another information in the mandatory field
Then the user should NOT be able to submit the form.

3.2.3   ID: FR8. Feature: Search

DESC: Given that a user enters the app,
And the users selects the Requests section
And the users selects the All Requests sub-section
then the user should be able to perform a search, according to **2 or more search criteria**. The minimum search criteria are location and type of the request. There should also be a free-of-choice/Show all search option. A user should be able to select one or more search criteria in one search.

3.2.4   ID: FR9. TITLE: Search result in a list view

DESC: The default view of the search results is a list.
Search results can be viewed in a list.
Each element in the list represents a specific user request.
Each element in the list should include the type of request, date of request, pet species, pet name, request/pet town, name of the user who requested, short subject of the request, pet photo and an information link/**button**.

3.2.5   ID: FR10. TITLE: Search result in a map view

DESC: Search results can be viewed on a map. On the map, the location, according to the user's request, is shown.
A specific pin will represent the request's location. On each pin there should be a (short) information according to what is specified in the corresponding request.
The map view should have a default zoom.

3.2.6   ID: FR11. TITLE: Selecting the information link

DESC: A user should be able to select the information link/**button**, which is included on each of the result items in the list view.
When the user selects the information link/**button**
then the link/**button** should disclose to the user
1) the whole information of the request provided by the pet's owner, including additional information if added by the pet's owner
2) pets in the request section, including all the information of the pet, added by the pet's owner;
3) pet owner data section, including all the information of the pet's owner, added by the pet's owner;
4) a map view, showing the location of the address in the request, shown on a map.

3.2.7   ID: FR12. TITLE: Search by town

DESC: A user should be able to select a town from a drop-down menu.
When the user selects the town and the user clicks the search button,
then the result fetched is a list consisting of only the requests, for which (an address and/or neighborhood in) the selected town is specified.

3.2.8   ID: FR13. TITLE: Search by request's type

DESC: A user should be able to select a requests' type from a drop-down menu. When the user selects the requests' type and the user clicks the search button,
then the result fetched is a list consisting of only the requests for which the selected type is specified.

3.2.9   ID: FR14 – TITLE: Search by multiple criteria

DESC: A user should be able to select both a town and a request's type from the separate drop-down lists.
When the user selects the town and the user selects a request's type and the user clicks the search button,
then the result fetched is a list consisting of only the requests matching ALL of the selected criteria.

3.2.10   ID: FR15 free-of-choice search

DESC: A user should be able to conduct a search by not choosing neither filtering option.
When the user does not choose a filtering option and the user clicks the search button,
then the result fetched is a list of all the requests published in the app.

3.2.11   ID: FR16 TITLE: No match found

DESC: When the user selects the town and/or the user selects the breed and no match is found
then the user should see a message informing that the search finds no matching requests Then the user should be kept on the search page in order to get the possibility to conduct a new search right away.

3.2.12   ID: FR17

Feature: Add new request
DESC: The user should be able to add a new request.
In order to add new request
A user should fill the mandatory fields In a form.
Scenario: Filling in mandatory fields
Given the user wants to fill the mandatory fields
When the user selects one or more of his/her added pets And the user selects location And the user selects type of the request And the user selects Start date And the user enters About information And the user enters a phone number
Then the user has filled the mandatory fields of the form.
Scenario: Filling in optional fields
Given the user wants to fill in optional fields in the form
When the user provides End date And the user checks the Offering payment option
Then the user has filled in optional fields in the form
Scenario: Adding a new request with mandatory fields
Given the user has filled in the mandatory fields of the form
When the user submits the form
Then the new request containing the data filled in the mandatory fields should be added.
Scenario: Adding a new request with mandatory And optional fields
Given the user has filled in the mandatory fields of the form
And the user has filled in one or more optional fields of the form
When the user submits the form
Then the new request containing the data filled in the mandatory And in the optional fields should be added.

3.2.13   ID: FR18

Feature: Edit a request
DESC: The user should be able to edit each one of his request, one at a time.
Given the user wants to edit a request

When the user selects one of his/her requests
And the user edits, adds or deletes information
And the user submits the form
Then the request should be edited.
Scenario: Delete information in mandatory fields
If the user deletes the information from a mandatory field
Then the user should NOT be able to submit the form

3.2.14   ID: FR19

Feature: Delete a request
DESC: The user should be able to delete each one of his request.
Given the user wants to delete a request
When the user selects one of his/her requests And the user selects a delete option
Then the request should be deleted.

3.2.15   ID: FR20

Feature: Add a new Pet
DESC: The user should be able to add a new pet.
In order to add new pet
A user should fill the mandatory fields In a form.
Scenario: Filling in mandatory fields
Given the user wants to fill the mandatory fields
When the user enters a name And the user selects species And the user enters About info
Then the user has filled the mandatory fields of the form.
Scenario: Filling in optional fields
Given the user wants to fill in optional fields in the form
When the user provides pet's sex and birthday And the user enters Food habits And the user uploads a photo of the pet
Then the user has filled in optional fields in the form
Scenario: Adding a new pet with mandatory fields
Given the user has filled in the mandatory fields of the form
When the user submits the form
Then the new pet containing the data filled in the mandatory fields should be added.
Scenario: Adding a new pet with mandatory And optional fields
Given the user has filled in the mandatory fields of the form
And the user has filled in one or more optional fields of the form
When the user submits the form
Then the new pet containing the data filled in the mandatory And in the optional fields should be added.

3.2.16   ID: FR21

Feature: Edit a pet
DESC: The user should be able to edit each one of his pets, one at a time.
Given the user wants to edit a pet
When the user selects one of his/her pet
And the user edits, adds or deletes information
And the user submits the form
Then the pet should be edited.
Scenario: Delete information in mandatory fields
If the user deletes the information from a mandatory field
Then the user should NOT be able to submit the form

3.2.17   ID: FR22

Feature: Delete a pet
DESC: The user should be able to delete each one of his pets.
Given the user wants to delete a pet
When the user selects one of his/her pets And the user selects a delete option
Then the pet should be deleted.

3.2.18   ID: FR23

DESC:
Given the user is logged-in to his/her account
And the user wants to sign out
When the user select the sign out link
Then the user should be logged out of his/her account and directed automatically to the Logged-in page

3.3   User Class 3 – Administrator

3.3.1   ID: FR24. Feature: Administrator log in

In order to administer the system
An administrator Should be logged in to the app
Scenario: Successful log-in
Given the administrator wants to log in
When the administrator logs in with an administrator account
Then the administrator should be logged in as an administrator

3.3.2   ID: FR25 Feature: Manage animal species

In order to have a list of species
An administrator Should be able to manage the species
Scenario: Add a new species
Given the administrator is logged in
When the administrator add a new species
Then the new species should be added to the list of species
Scenario: Editing an existing species
Given the administrator is logged in
When the administrator edits an existing species
Then the species should be updated in the list of species
Scenario: Delete a species
Given the administrator is logged in
When the administrator deletes a species
Then the deleted species should be removed from the list of species

3.4   System Feature 2 (and so on)

4. External Interface Requirements

4.1 User Interfaces

4.1.1 Login screen

    A non-registered user of the app should see the home page with a log-in and sign-up options when he/she opens the app. If the user has not registered, he/she should be able to do that on the sign-up page.

    On the Login page there should be the following items:

1. Link to "Info" section on the Navigation bar
2. Link to "Contact us" form on the navigation bar
3. Title "Sign in",
4. input field "Email address"
5. input field "Password"
6. Button named "Sign in"
7. Link named "Sign up"

4.1.2 Sign-up screen

    When selecting the Sign up link, a user should be directed to a sign up form

    In the sign up form there should be the following items:

1. on the Navigation bar – Company's logo
2. on the Navigation bar - Link to "Info" section
3. on the Navigation bar - Link to "Contact us" form
4. Title "Sign in",
5. Input field "First name"
6. Input field "Last name"
7. Input field "Enter e-mail"
8. Text "We will never share your e-mail with anyone else"
9. Input field "password"
10. Input field "confirm password"
11. Submit Button named "Sign up"
12. Text "Already have an account?"
13. Link to "Sign in" form

4.1.3　Contact us screen – non-logged-in users view

1. on the Navigation bar – Company's logo
2. on the Navigation bar - Link to "Info" section
3. on the Navigation bar - Link to "Home" page
4. Title "Tell us what's on your mind",
5. Input field "Message title"
6. Input field "Message text"
7. Input field "Your name"
8. Input field "Your e-mail address"
9. Submit Button named "Message us"

4.1.4　Request Section → All requests screen

Navigation bar:
1. Link to "Info" page
2. Link to "Requests" page
3. Link to "My profile" page
4. Link to "My pets" page
5. Link to "Contact us" page
6. Link to "Sign out" option
7. Text "Signed in as *<First and Second name of the user>*"
   Below the Navigation bar – Filtering interface section
8. Text "Requests"
9. Drop down list Filter for location, named "City"
10. Drop down list Filter for type of requests, named "Type"
11. Check box named "Only future requests"
12. Button named "Filter"
13. Button named "Filter all on a map"
    Results section
14. RESULTS LIST showing
    - ON THE LEFT: the type of request, date of requested event, Location of request, Reward information (if any), Name of requested user, button named "View",
    - ON THE RIGHT: Picture of the pet in the request (if any);

4.1.5   Request Section → Add a new request screen

Navigation bar:
1. Link to "Info" page
2. Link to "Requests" page
3. Link to "My profile" page
4. Link to "My pets" page
5. Link to "Contact us" page
6. Link to "Sign out" option
7. Text "Signed in as *<First and Second name of the user>*"
Submit form section
8. Text "New Request:"
9. Check box named "Select one or more pets for this request"
10. Drop down list Filter for location, named "Search address, area or location"
11. Drop down list Filter for type of request, named "Type"
12. Date-type input field, named "Start date"
13. Date-type input field, named "End date"
14. Check box named "Offering reward"
15. Text Input field, named "Description"
16. Text Input field, named "Phone number for contact"
17. Submit button named "Publish the request"

4.1.6   Request Section → My requests screen

Navigation bar:
1. Link to "Info" page
2. Link to "Requests" page
3. Link to "My profile" page
4. Link to "My pets" page
5. Link to "Contact us" page
6. Link to "Sign out" option
7. *Text "Signed in as <First and Second name of the user>"*
Results section
8. RESULTS LIST showing for each result:
   - ON THE LEFT: the type of request, date of requested event, Location of request, Reward information (if any), button named "Delete", button named "Edit", button named "View",
   - ON THE RIGHT: Picture of the pet in the request (if any);

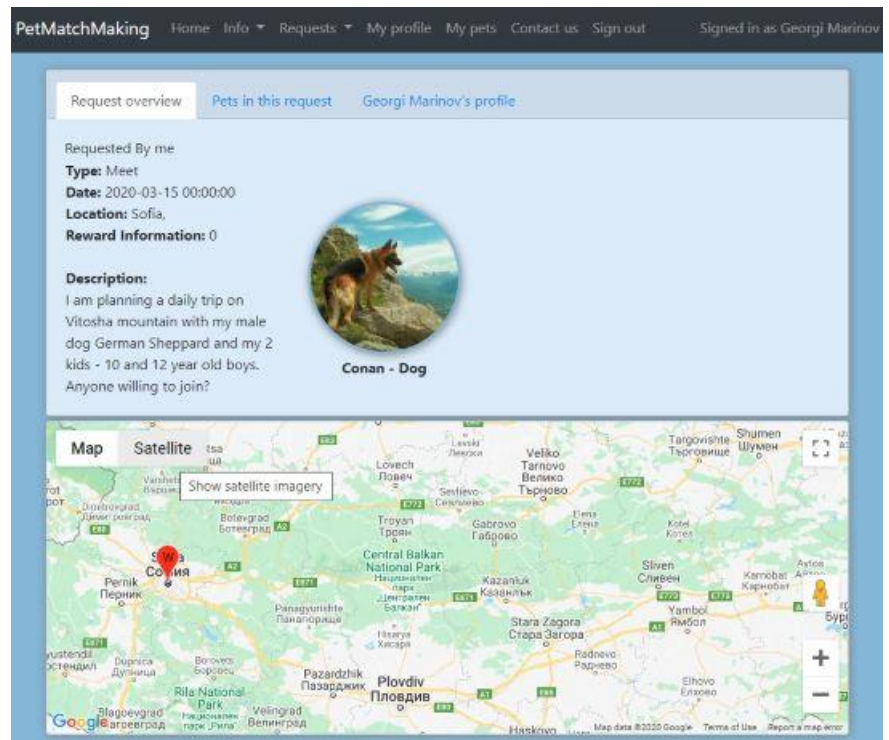### 4.1.7   Request Section → View a request Overview screen

Navigation bar:
1. Link to "Info" page
2. Link to "Requests" page
3. Link to "My profile" page
4. Link to "My pets" page
5. Link to "Contact us" page
6. Link to "Sign out" option
7. *Text "Signed in as <First and Second name of the user>"*
   View section /below navigation bar/:
   Button named Request Overview → when pressed exposes:
8. Text: "Requested by <Names of the user requested>"
9. Text: "Type of request: <Type of the request>"
10. Text: "Date of request: <Date of the request>"
11. Text: "Location: < Location of the request >"
12. Text: "Reward information: <Description of the terms with regard to the reward if any>"
13. Text: "Description <Description of the request if any>"
14. Photo of the pet if any
    Map /below View section/:
15. Pin on the map pointing to the location /town, neighborhood/ of the request
    Button named Pets in the request
    Button named <Names of the user requested>'s profile

4.1.8   Request Section → View a request – Pets in the request screen

Navigation bar:
1. Link to "Info" page
2. Link to "Requests" page
3. Link to "My profile" page
4. Link to "My pets" page
5. Link to "Contact us" page
   6. Link to "Sign out" option
7. *Text "Signed in as <First and Second name of the user>"*
   View section /below navigation bar/:
   Button named Request Overview
   Button named Pets in the request →
   when pressed exposes:
8. Text: "Sex: <Pet's sex>"
9. Text: "Birthday<Pet's birthday>" if any
10. Text: "Food habits: <Food habits if any>"
11. Text: "About: <Description of the request if any>"
   Button named <Names of the user requested>'s profile
   Map /below View section/: Pin on the map pointing to the location /town, neighborhood/ of the request

4.1.9   Request Section → View a request – Owner's profile screen

Navigation bar:
1. Link to "Info" page
2. Link to "Requests" page
3. Link to "My profile" page
4. Link to "My pets" page
5. Link to "Contact us" page
6. Link to "Sign out" option
7. *Text "Signed in as <First and Second name of the user>"*
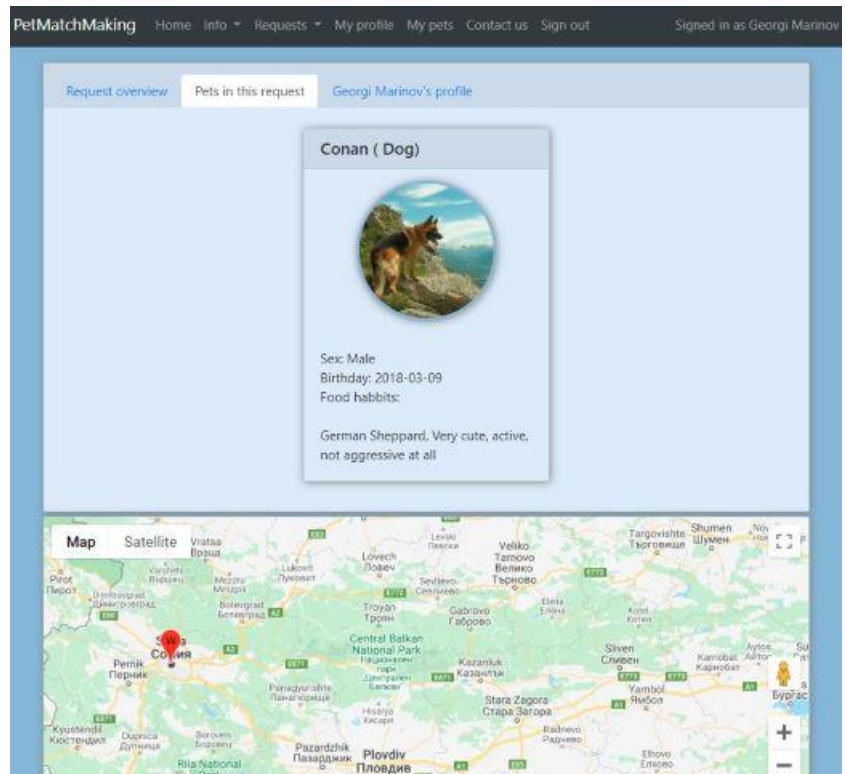   **View section** /below navigation bar/:
   Button named Request Overview
   Button named Pets in the request
   Button named <Names of the user requested>'s profile → when pressed exposes:
8. Text: "Name: <First and Last names of the user requested>"
9. Text: "Birthday<User's birthday>" if any
10. Text: "Rating: <User's rating >" if any
11. Text: "About: <Description of the user>" if any
12. Text "All <First and Last names of the user requested>'s pets"
13. Photos, resp. names of the user's pets if any

Map /below View section/: Pin on the map pointing to the location /town, neighborhood/ of the request

4.1.10   My Pets Section → Add a Pet screen

Navigation bar:
1. Link to "Info" page
2. Link to "Requests" page
3. Link to "My profile" page
4. Link to "My pets" page
5. Link to "Contact us" page
6. Link to "Sign out" option
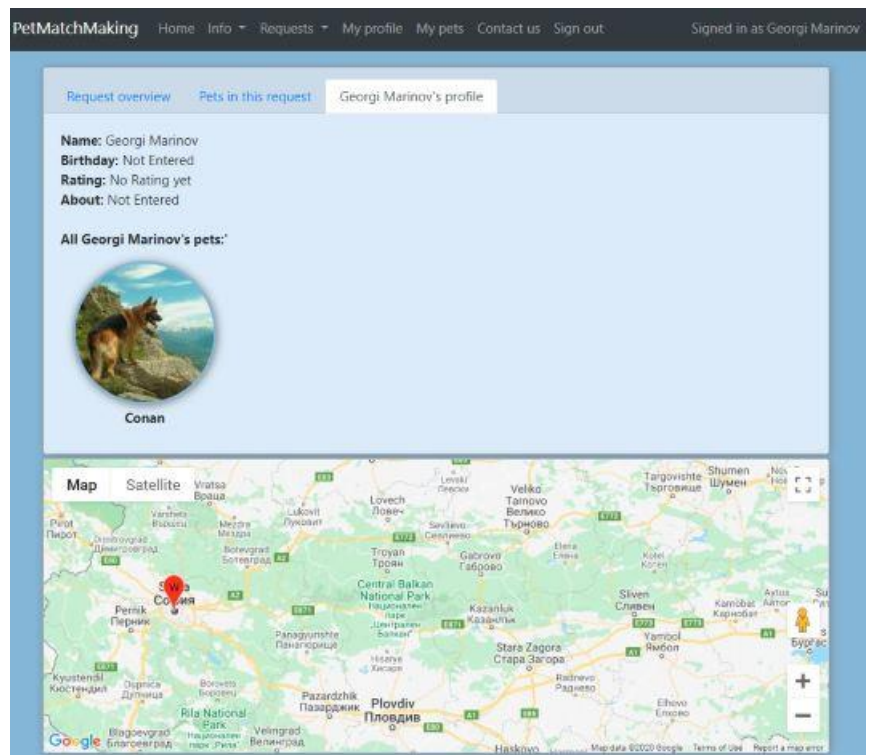7. Text "Signed in as <*First and Second name of the user*>"
   Adding new pet Form section
8. Text "Adding new pet"
9. Text "Name"
10. Drop-down list Filter named "Type"
11. Drop-down list Filter named "Sex"
12. Date-type input field, named "Birthday"
13. Text Input field, named "Food habits:"
14. Text Input field, named "About:"
15. FileUpload Button named "Choose File"
16. Submit button named "Save pet"

4.1.11   My Pets Section → Pets List screen

Navigation bar:
1. Link to "Info" page
2. Link to "Requests" page
3. Link to "My profile" page
4. Link to "My pets" page
5. Link to "Contact us" page
6. Link to "Sign out" option
7. Text "Signed in as <*First and Second name of the user*>"
   Below Navigation bar and above Result section:
   Button named "Add a pet"
   Results section
14. Text "My pets"
15. RESULTS LIST showing for each result:
    - UP: Name of the pet, Species of the pet
    - DOWN: Sex, Birthday, Food habits Description if any inputted in About section, button named "Delete", button named "Edit"

4.1.12   My Pets Section → Edit pet screen

The screen is the same as "My Pets Section → Add a Pet screen" except the following difference: The title name is: "Edit <name of the pet>" instead of "Adding new pet"

4.1.13   My Profile section → General screen

Navigation bar:
1. Link to "Info" page
2. Link to "Requests" page
3. Link to "My profile" page
4. Link to "My pets" page
5. Link to "Contact us" page
6. Link to "Sign out" option
7. Text "Signed in as *<First and Second name of the user>*"
   Below Navigation bar and above Pets added by the:
8. Page title with text "My Profile"
9. INFO ABOUT THE USER SECTION:
   - Name of the user
   - Birthday if entered
   - Rating if received
   - About if entered
10. Button named "Edit profile" directing to Edit profile form
16. A LIST SECTION, with pets added by the user containing:
    - Photo of the pet if added
    - Name of the pet
17. OTHER INFORMATION ABOUT THE USER SECTION:
    - e-mail address
    - Phone number if entered
    - Button named "Change password" directing to Change password form

4.1.14   My profile section → Edit Profile screen

Navigation bar:
1. Link to "Info" page
2. Link to "Requests" page
3. Link to "My profile" page
4. Link to "My pets" page
5. Link to "Contact us" page
6. Link to "Sign out" option
7. Text "Signed in as *<First and Second name of the user>*"
   Below Navigation bar:
8. Page title with text "Edit Profile"
9. Input field named "e-mail address" – inactive; The user will not be able to change the e-mail address
10. Text Input field named "First name"
11. Text Input field named "Last name"
12. Text Input field named "Phone number"
13. Date input field named "Birthday"

14. Text Input field named "About me"
15. Submit Button named "Update profile"

4.2   Hardware Interfaces

Since the app has no designated hardware, it does not have any direct hardware interfaces.

4.3   Software Interfaces

The communication between the database and the app consists of operation concerning both reading and modifying the data. There is no other connection between this product and other specific software components, including other databases, operating systems, tools, libraries, and integrated commercial components.

5.   Other Nonfunctional Requirements

5.1   Performance Requirements

5.1.1   ID: QR1 TITLE: Prominent search feature

DESC: The search feature should be prominent and easy to find for the user.

5.1.2   ID: QR2 TITLE: Usage of the search feature

DESC: The different search options should be evident, simple and easy to understand.

5.1.3   ID: QR3 TITLE: Usage of the result in the list view

DESC: The results displayed in the list view should be user friendly and easy to understand. Selecting an element in the result list should only take one click.

5.1.4   ID: QR4 TITLE: Usage of the result in the map view

DESC: The results displayed in the map view should be user friendly and easy to understand. Selecting a pin on the map should only take one click.

5.1.5   ID: QR5 TITLE: Usage of the information link /button

DESC: The information link /button should be prominent and it should be evident that it is a usable link /button. Selecting the information link /button should only take one click.

5.1.6   ID: QR6 TAG: ResponseTime

GIST: The fastness of the search
SCALE: The response time of a search
METER: Measurements obtained from 1000 searches during testing.
MUST: No more than 2 seconds 100% of the time.
WISH: No more than 1 second 100% of the time.

5.2   Safety Requirements

Overview: The app should assure user data confidentiality and that the users are informed how to abide by the legal requirements.

The app should inform the users of the legal matters with regard to storing and reading of personal data and other juridical aspects of users' activities in the context of the app and to suggest the necessary tools (e.g. declarations, consents, etc.) connected with personal data processing regulations as well as with the other juridical aspects of users' activities.

5.3   Security Requirements

Overview: The app should be protected from unauthorized access with security solutions.

5.3.1   ID: QR7. TAG: UserLoginAccountSecurity

GIST: Security of accounts.
SCALE: If a user tries to log in to the app with a non-existing account
then the user should not be logged in.
The user should be notified about log-in failure.
METER: 1000 attempts to log-in with a non-existing user account during testing.
MUST: 100% of the time.

5.3.2   ID: QR8. TAG: AdminLoginAccountSecurity

GIST: Security of accounts.
SCALE: If an admin tries to log in to the app with a non-existing account
then the admin should not be logged in.
The admin should be notified about log-in failure.
METER: 1000 attempts to log-in with a non-existing user account during testing.
MUST: 100% of the time.

5.4   Software Quality Attributes

Apart from the above mentioned quality requirements the app should have the following additional quality attributes: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. These are to be additionally specified and detailed in order to be verifiable.

The app should be user-friendly with high intuitive user interface.

The screen size should adapt to the different devices with which users access the app so that all the menus, links, buttons and text are visible and readable.

6.   Other Requirements

The app should bring about easy, effective and emotionally satisfactory communication between pet lovers and other users with its thorough vision, design and colors.

The system database capacity should be assessed and eventual overloading should be oversaw on regular basis in order that and adequate and timely measures could be taken towards its extending.

Appendix A: Glossary

Term	Definition
The app	The application specified in the current document
The system	All the components making possible the functioning of the app, incl. front end and back end
User	Someone who interacts with the app
Non-registered user	Someone who interacts with the app without having a user account
Registered user	Someone who interacts with the app and has a user account
Logged-in user	A Registered user who interacts with the app after he/she has logged-in with their username/e-mail and password
Non-Logged user	A Registered user who interacts with the app without having logged-in with their username/e-mail and password
Admin/Administrator	System administrator who is given specific permission for managing and controlling the system
Stakeholder	Any person who has interaction with the system who is not a developer.
DESC	Description
FR	Functional requirement
QR	Quality requirement
GIST	A short, simple description of the concept contained in a PLanguage statement
TAG /Planguage/	A unique, persistent identifier contained in a PLanguage statement Performance.Report.ResponseTime
SCALE /Planguage/	Seconds of elapsed time between pressing the Enter key or clicking OK to request a report and the beginning of the display of the report.
METER /Planguage/	Stopwatch testing performed on 30 test reports that represent a defined usage operational profile for a field office accountant
WISH /Planguage/	No more than 1.5 seconds for all reports