

---

# Software Requirements Specification (SRS)

for



Prepared by

## Group Members:

Ghaida Alkhudhair	441926684	441926684@student.ksu.edu.sa
Aljwhra Almakhdoub	441201225	441201225@student.ksu.edu.sa
Raghad Aljuhaimi	441201212	441201212@student.ksu.edu.sa

Instructor:	Sarah Alotaibi – Farah Al-sheikh
Course:	CSC343 – Systems Analysis & Design
Section:	58274
Date:	11-2-2022

# Contents

<b>Introduction.....</b>	<b>4</b>
<b>1 Project Proposal .....</b>	<b>5</b>
1.1 Introduction.....	5
1.2 Statement of Problem.....	5
1.3 Project's Scope .....	5
1.4 Project's Objectives .....	6
1.5 Related Works .....	6
<b>2 Software Requirements Specification (SRS).....</b>	<b>7</b>
2.1 Customer Statement of Requirements.....	7-8
2.2 Glossary of Terms.....	9
2.3 User Requirements.....	10
2.3.1 Enumerated Functional Requirements.....	10-11
2.3.2 Enumerated Non-Functional Requirements.....	11
2.3.3 User Interface Mock-up .....	12-14
2.4 Functional Requirements Specifications.....	15
2.4.1 Stakeholders.....	15
2.4.2 Actors and Goals .....	15
2.4.3 Use Case Casual Description .....	15-17
2.4.4 Use Case Diagram.....	18
2.4.5 Use Case Fully-Dressed Description.....	19-21
<b>3 Software Design Document (SDD).....</b>	<b>22</b>
3.1 Introduction.....	22
3.2 Glossary of Terms.....	23
3.3 Interaction Diagrams.....	24
3.3.1 Sequence Diagrams.....	24-26
3.4 The System Structural Diagram.....	27
3.4.1 The Detailed Class Diagram “Software Artifact” .....	27
3.4.2 The Attributes and Methods Description for Each Class.....	28-31
3.5 The System Architecture and Design .....	32
3.5.1 The System Architectural Style.....	32
3.5.1.1 The System Organization.....	32
3.5.1.2 The Data Flow Model.....	32
3.5.1.3 The Control Model.....	32-33
3.5.1.4 Identifying the Subsystems.....	33

---

3.6 The Object Diagrams .....	34
3.6.1 The Object Diagrams.....	34
3.7 The System Behavioral Diagrams.....	35
3.7.1 The State Diagram.....	35
3.8 The User Interface Design.....	36-38
3.9 The Design of Tests.....	39
3.9.1 Unit Testing.....	39-41
3.9.2 Integration Testing.....	42
3.9.3 Acceptance Testing.....	42
3.9.4 Path Testing.....	43
3.9.5 Architecture Diagram .....	44
 <b>4 Individuals Contributions Breakdown.....</b>	 <b>44-47</b>
 <b>5 References.....</b>	 <b>48</b>

---

# 1-Project proposal

## 1.introduction

In our day to day lives we encounter many stray animals in neighborhoods, streets, and many places, that might get hurt from cars or careless people or die from hunger and sickness. Helping them and giving them a better home might be sometimes difficult or impossible. That is why we wish we could have an easy, efficient, and fast way to save these poor souls. Stray dangerous animals are a problem as well they might hurt people and spread diseases that is why we also need an easy way to prevent their danger.

## 2.Statement of Problem

Stray animals are a big problem that we witness in our neighborhoods and streets, these poor souls get hurt and die from hunger or a sickness daily.

But we must be careful not all stray animals are considered pets, there are some very dangerous animals out there that people might find that could possibly cause harm.

Our application will help save stray animals as well as help prevent people from getting hurt by any dangerous animals. By filing a report in our application about the animals; their location, kind and state our application will recommend organizations/clinics/shelters that could help these strays. Any reported animal either stray or dangerous will be shown in a map that our application provides so you could possibly help the strays or look out for the dangerous ones.

## 3.Project's Scope

We want to create a platform to make the adoption, protection from dangerous animals, treating the stray pets, and finding your lost pet much easier. The target users of our application are android users between the age of 14-60, for both genders. Our goal is to make the connection between the animals and humans much easier. You need to have an active internet access to use the app, it will use a GPS navigation system, to help the user locate the animal position in the map and a push notification function.

### Features:

- Every user can access the app without a username or a password.
- The user can go to the map to see the nearby animals or report any problem with an animal that we have mentioned previously.

- If the user wants to report a lost pet, or they want to report a dangerous animal, or a stray animal that needs treatment, we will provide for them a list of organizations/clinics/shelters that can nurse the type of animal they're reporting.
- What is the users' view of the map? If the animal is dangerous the user can only view its' location in the map. But if the animal is a stray pet the user can view its' location in the map and its' type. We chose this view to make it easier for the user to help the animals if they need it. Finally, the user can't view owned pets since they have owners.

**Constraints:**

Since it's a course project we might face some problems with time management, and when it come to the users experience of the app, we can't provide it to the public to view the reviews and improve on it.

## 4.Project's Objectives

Objectives:

- Develop an application that helps in protecting stray animals.
- Protect people from dangerous animals that could harm humans.
- Facilitate the reporting process by providing communication to the responsible authorities.

Services:

- Reporting stray or dangerous animals.
- Users can see the dangerous animals in the map to protect themselves.
- Providing ways to communicate with governmental authorities who are responsible for protecting animals so that they help the users to report animals.

## 5.Related Works

The "Balady" "Amana 940" are applications that provide many services. They have some common services as well, one of the services is reporting stray and dangerous animals and other types of animals. The main deference between our application and Balady and Amana 940 is that our application can show the dangerous animals in the map so people can stay away of that area until the responsible authorities take action.[7][8]

## 2-Software Requirements Specification (SRS)

### 1. Customer Statement of Requirements

According to Next City, “When humans and animals share space, conflict often follows in the form of animal-vehicle collisions, damage to property, the spread of disease and attacks on pets and people. [1] Unfortunately, people forget that animals need to be protected too. Several questions come in mind when we mention that, for example, how many stray animals do people see on the streets daily? What is the state of these poor strays? Injured, starved, infected, or even alive? These animals don’t deserve this treatment, they share the same space with us on this world and deserve proper care.

After interviewing our client "The Environmental Research and Wildlife Development Agency", they stated that they are focusing on ways that ensure high level of protection for both people and animals, we gathered some information by discussing the problem with the client, and what is the desired form of application they want, and how they are willing to make it functional, and their requirement stated a full stack application that provides the ability to inform nearby people who can help stray animals by reporting them by listing specific organizations communication tools to the reporter, then the organization should handle the situation of any animal appropriately. Moreover, the application can notify the users of the existence of dangerous animals to protect people. Lastly, the problem that we might face is some users may not report the animals correctly, so it could possibly cause conflict or a misunderstanding to other users who are being notified by the app.

#### 1.1. Report an Animal

People in their daily lives encounter animals all the time, some of the animals are stray, owned, or unfortunately dangerous. Animals could cause serious problems and injuries if they attempt to hurt humans or other animals. On the other hand, some animals are harmless, and they do need help, care and not to face hunger, or injuries. Our application is trying to solve these problems by offering a reporting process which provides people with contact tools of the responsible institutions to report any animal in a correct way if it needs help, can cause damage, or if it is a source of disturbance to inhabitants in neighborhoods regardless of the animal’s type or state.

#### 1.2. Ability to specify the location of animals

One of the remarkable problems is when inhabitants face dangerous animals, this could threaten the inhabitant's lives, spread fear over the neighborhood, and lead to a stressful lifestyle, therefore, the application provides the ability to specify in the map the place of the dangerous animal stray animals, it will appear to other users as a range -Since animals do not stay stable in an exact place- that shows the threatened area, so people can stay away from this area. Regarding the stray animals, it is possible to determine the state, the type and condition of the animal to make it possible to help that animal immediately if anyone else is near and willing to help if the responsible institution did not act yet.

### **1.3. Locating an owned pet**

One of the difficulties we had is reporting about an owned animal, since we do not want any unethical commercial use by using the application to locate animals to trade with others or abduct an animal rather than purchasing from a pet clinic, therefore, we wanted a way to make the most efficient use to fulfill the client's requirement. Our client wanted an application to locate a lost owned pet easily, thus, our application will provide the appropriate functionality to find them. Regarding the owned animals, the reporter can only report specifications of the pet, so when the user wants to search for their pet, they need to specify the pet's appearance, then a list of pets with those similar specifications will be provided for the user with pictures to compare the listed specifications with their lost pet, and they can contact the reporter by the chat functionality to possibly discuss the pet's last location or its state (Healthy, sick, injured... Etc.), or if the owner could send a picture of their pet to the reporter and more. In addition, the reporter might also send the report to an organization that we will provide tools to communicate with, so they could take care of the animal. If someone lost his pet, he/she can report that she/he lost his pet and the app will search for it from the previous reports when the app finds it, the app will provide the option to communicate with the founder.

### **1.4. When the user will be notified**

A main function in the application is sending notification to the users, yet each notification is different depending on the user's location and category of animal. If the user opens the map in the application wherever the user's location might be dangerous animal's precise location will appear to the user either close or far as well as the precise location of stray pets and their type. Since the application has access to the user's location it will notify the user with a warning if they are near any dangerous animals and will notify the user with a range of the dangerous animal's location.

### **1.5. Communication tool**

A big problem that the application 'Be a Rescuer' is trying to solve is the fate of any dangerous, stray, or lost owned animal that is reported in the application. That is why the application has the communication tool. The communication function is a function that depends on the type of animal and its state, if the animal is a healthy stray the communication function will list organizations/shelters that could care for it and provide a home for the animal, but if the animal is an unhealthy stray the function will list organizations/clinics/shelters that could nurse it, care for it, and provide a home for it. Additionally, If the animal is a lost pet and has a guardian, the function will list the possible guardians that registered in the system and reported their lost animal. The guardians are listed based on their report, if both reports match (the report of the guardian and founder) the guardian will be listed to the founder of the lost pet. Lastly if the animal is a dangerous animal the function will list possible organizations that could take care of it.

## 2. Glossary of Terms

TERM	DESCRIPTION
Stray animal	An animal that is homeless, abandoned, needs help, or not in the right place. [1]
Inhabitant	Person or animal that occupies a place and lives there. [1]
Notification	A message that is automatically sent to you on your mobile phone or computer.[2]
Function	A process which a computer or a software program uses to complete a task.[2]
Guardian	A person who's in charge of a pets' health and well-being for the rest of its life.[2]
Collisions	An instance of one moving object or person striking violently against another.[3]
Full stack	The entirety of a computer system or application, comprising both the front end and the back end.[3]



### 3. User Requirements

#### 3.1. Enumerated Functional Requirements

<b>REQ- ID</b>	<b>DESCRIPTION</b>	<b>PW</b>
REQ - 1	The system shall provide several numbers and contact tools of different institutions to the users.	5
REQ - 2	The user shall be able to take pictures.	3
REQ - 3	The user shall be able to upload their report into the map	3
REQ - 4	The system shall provide a map that show the range of the dangerous animals or the exact location, type, and condition of stray animals.	5
REQ - 5	The system shall have access to the user's location.	5
REQ - 6	The user shall give a location access to the system.	5
REQ - 7	The system shall provide a report for the users to fill during the reporting process.	3
REQ - 8	The user shall complete the provided report by the system in detail.	3
REQ - 9	The system shall ask if the user wants to be anonymous.	2
REQ - 10	The user shall be able to share photos through the chat between the reporter and the owner, founder...etc.	4
REQ - 11	The system shall provide the chat feature between users.	4
REQ - 12	the system shall alert the users when a dangerous animal is nearby.	5
REQ - 13	The system shall be able to register users.	5
REQ - 14	The user shall be able to access the app without registration.	3
REQ - 15	The system shall register the user when they want to report.	5
REQ - 16	The system shall register the user when they want to send or receive messages to other users.	5
REQ - 17	The system shall be able to store animal specifications in the database <ul style="list-style-type: none"> <li>• Owned or stray</li> <li>• Animal species</li> <li>• Age</li> <li>• Weight</li> <li>• Fur color 1,2....</li> <li>• Eye color</li> </ul>	5

REQ - 18	The system shall allow the user to update their profile's information when signed up.	4
REQ - 19	The user shall be able to update their profile's information when they're signed up.	4
REQ - 20	The system shall provide for the user a search bar with specifications to search for lost animals in the map.	4
REQ - 21	The user shall be able search for lost animals of any specifications with the search bar on the map.	4
REQ - 22	The system shall allow the user to view their information.	3
REQ - 23	The user shall be able to view their previous reports.	3
REQ - 24	The user shall be able to upload the found pets photos through the map.	3
REQ - 25	The user shall be able to delete/update their previous reports.	3
REQ - 26	The user shall be able to specify the type and condition of the stray animal.	5
REQ - 27	The user shall login to the system when they want to report.	5
REQ - 28	The user shall view stray animals and owned pets' information and their photos through the map.	3
REQ - 29	The user shall be able to specify the animal's type whether it is dangerous or not.	5

### 3.2. Enumerated Non-Functional Requirements

REQ- ID	DESCRIPTION	PW
REQ - 30	The system shall take pictures with less than 5 seconds.	5
REQ - 31	The system shall not take more than 1 second to show the communication tools of institutions while reporting.	5
REQ - 32	The system shall load the map in less than 2 seconds.	4
REQ - 33	The system shall upload the filed report in less than a minute.	5
REQ - 34	The system shall match the guardians with their lost pet in less than a minute.	3
REQ - 35	The system shall display a map specifying the dangerous and stray animal's location.	5
REQ - 36	The system shall be able to store user reports.	5
REQ - 37	The system shall provide user reports to institutions or admins.	5

### 3.3. User Interface Mock-up



Figure 1. Log in screen.



Figure 2. Sign in screen.



Figure 3. Username and Password screen.



Figure 4. Map screen.



Figure 5. Show animal information on map screen.



Figure 6. Edit profile screen.



Figure 7. Report screen.



Figure 8. Camera.

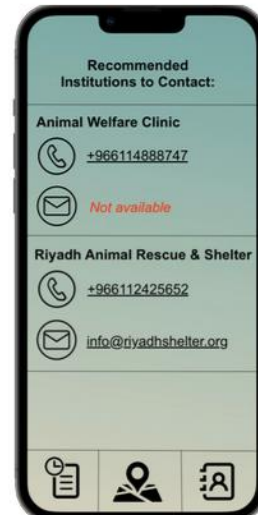


Figure 9. Contact tools screen.



Figure 10. Previous reports screen.



Figure 11. Messages screen.



Figure 12. Map screen as a guest.



Figure 13. Search bar screen.

## 4. Functional Requirements Specification

### 4.1. Stakeholders

*Citizens: Any citizen can be able to benefit from the application and use its services to protect themselves and others.*

*Saudi Wildlife Authority: Which is a national system that aims to preserve wildlife in Saudi Arabia, because the application helps to protect the environment and save animals. [5]*

*Information Technology staff: Who are responsible for maintaining the application if it faces errors or problems.*

### 4.2. Actors and Goals

<b>Actor</b>	<b>Type</b>	<b>Goal</b>
<i>User</i>	<i>Initiating</i>	<i>Reporting stray, owned, and dangerous animals that users may encounter anywhere to specified Institutions.</i>
<i>Institutions</i>	<i>Initiating</i>	<i>Institutions who want to check the map to look for the location of the reported animal to handle the situation.</i>
<i>Application Administrator</i>	<i>Initiating</i>	<i>They can update the status of the reports, check if the application has no errors, works correctly, and send notifications to nearby users.</i>
<i>Data Base</i>	<i>Participating</i>	<i>To store the information of users, animals, and communication tools that the application provides.</i>

### 4.3. Use Case Casual Description

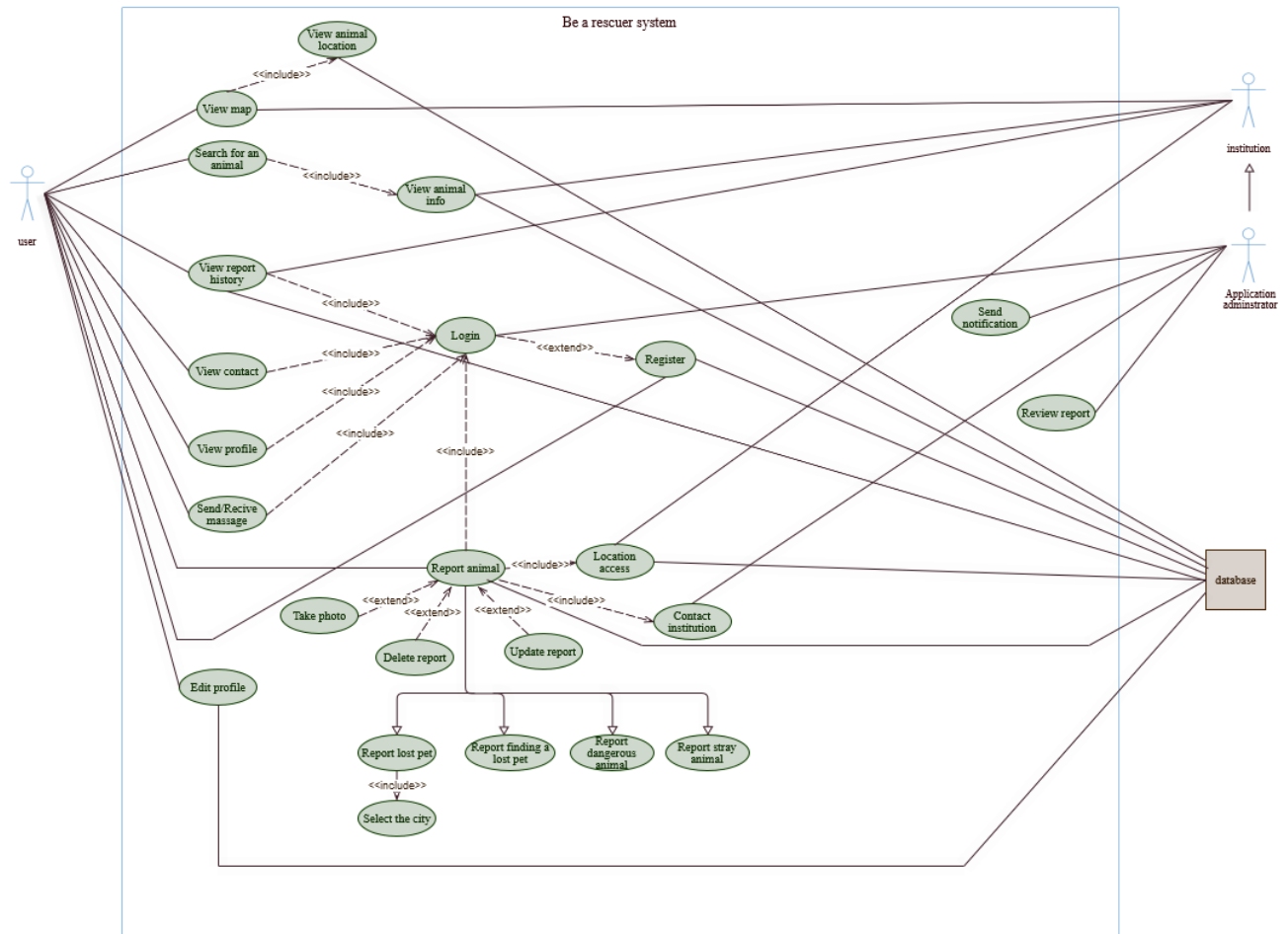
<b>Use Case ID</b>	<b>Name</b>	<b>Short Description</b>	<b>Corresponding REQ-id</b>
<i>UC1</i>	<i>View contact</i>	<i>Allow users to see the contact tools of specified institutions, either as a phone number or email.</i>	<i>REQ – 1.</i>
<i>UC2</i>	<i>Report animal</i>	<i>Allow users to report animals by providing a form to fill and providing the contact tools of institutions.</i>	<i>REQ – 1, REQ – 3, REQ – 7, REQ – 8, REQ – 9, REQ – 15, REQ – 33, REQ – 36.</i>

UC3	Send/Receive message	Allow the users to send or receive messages if the user is logged into the system.	REQ – 10, REQ – 11, REQ – 16.
UC4	Take photo	Allow the users to upload and share photos when they are logged in.	REQ – 2, REQ – 10, REQ – 24, REQ – 28, REQ – 30.
UC5	Delete report	Allow the users to delete the uploaded report (see UC3).	REQ – 7, REQ – 25, REQ – 27.
UC6	Update report	Allow the users to update the uploaded report (see UC3).	REQ – 7, REQ – 25, REQ – 27.
UC7	Contact institution	Allow the users to contact a provided institution (see UC3).	REQ – 1.
UC8	Report lost pet	Allow the users who lost their pets to report, and the database will match the reported specifications from the users who found a similar pet (Sub case for UC3).	REQ – 10, REQ – 17.
UC9	Select location	The user who has lost their pet needs to choose the location where the pet was last seen (Sub case for UC3/UC9).	REQ – 5, REQ – 6.
UC10	Report finding lost pet	Allow the users who find a lost pet to report it, so the pet guardian can contact them, the user will only report the location to the system administrator and institution (Sub case for UC3).	REQ – 8, REQ – 17, REQ – 26.
UC11	Report dangerous animal	Allow the users to report dangerous animals. No more further specifications are needed if the animal is dangerous (Sub case for UC3).	REQ – 29.
UC12	Report stray animal	Allow the users to report a stray animal (Sub case for UC3).	REQ – 26.
UC13/14	Log in / Sign up	Allow users to log into the system if they have previously signed up and created a profile or create a new profile by signing up.	REQ – 9, REQ – 13, REQ – 14, REQ – 15.

UC15/16	View profile / Edit Profile	Allow users to view and update their profile information (First name, Last name, ID number, Phone, Email and Date of Birth).	REQ – 18, REQ – 19.
UC17	View map	Allow users to view different types of animals on the map and each animal's location area.	REQ – 4, REQ – 5, REQ – 6, REQ – 20, REQ – 21, REQ – 32, REQ - 35.
UC18	View reports history	Allow users to view the previous reports they made and their information (Report ID, Type, Date, and Progress).	REQ – 17, REQ – 22, REQ – 23.
UC19	Search for an animal	Allow users to use the search bar to search for a stray/owned animal.	REQ – 14, REQ – 21.
UC20	View animal information	Allow users to view animals' information while they are viewing the map (See UC18).	REQ – 14, REQ – 28.
UC21	View all previous reports	Allow system administrator to view all previous reports that users have made.	REQ – 28.
UC22	Location access	Allow the users to specify the pet's location in a range (See UC3).	REQ – 5, REQ – 6.
UC23	Send notification	The system administrator will send notifications if there is a nearby dangerous animal.	REQ – 12.
UC24	Review Report	Institution and application administrators can review users' previous reports (See UC3).	REQ – 36, REQ – 37



## 4.4. Use Case Diagram



\* made by using gliffy tool. [6]

### 4.5. Use Case Fully-Dressed Description

<b>UC2</b>	Report animal
<b>Initiating Actor:</b>	User.
<b>Actor's Goal:</b>	To report an animal.
<b>Participating Actor:</b>	Institutions, Application Administration, Data base.
<b>Pre-Conditions:</b>	The user should be registered in the application to be able to report.
<b>Post-Condition:</b>	The location of the animal will appear in the map as a range, and the information of the animal will be stored in the data base after submitting the report.
<b>Flow of Events for Success Scenario:</b>	<ol style="list-style-type: none"> <li>1. User should be logged in the application.</li> <li>2. The user clicks on the report icon in the map.</li> <li>3. A form appears for the users to fill animal's information.</li> <li>4. The contact tools will appear to user.</li> <li>5. Animal's information will be stored in the data base.</li> </ol>
<b>Flow of Events for Extension (Alternate Scenario):</b>	<ol style="list-style-type: none"> <li>1.1 User does not sign in /log in.</li> <li>1.2 The report icon will not appear.</li> <li>1.3 The user cannot report.</li> </ol>

<b>UC1</b>	View contact
<b>Initiating Actor:</b>	User.
<b>Actor's Goal:</b>	To contact with an institution to report an animal.
<b>Participating Actor:</b>	Data base.
<b>Pre-Conditions:</b>	The user should be signed up in the application to be able to view the contact.
<b>Post-Condition:</b>	The user can contact specified institutions.
<b>Flow of Events for Success Scenario:</b>	<ol style="list-style-type: none"> <li>1.1 User continues as a guest into the application.</li> <li>2.1 After the user submits the report the application will immediately suggest the</li> </ol>

	contact tools to user according to the animal's type and state.
<b>Flow of Events for Extension (Alternate Scenario):</b>	<b>1.1</b> User does not sign up / log in. <b>1.2</b> The contact tools will not appear. <b>1.3</b> The user cannot contact any institution.

<b>UC17</b>	View map
<b>Initiating Actor:</b>	User.
<b>Actor's Goal:</b>	To view the map, view the animal's location and search for lost animals.
<b>Participating Actor:</b>	Institution.
<b>Pre-Conditions:</b>	The user can be signed up or viewing as a guest and allows access of location.
<b>Post-Condition:</b>	The user can be aware of surrounding animals of any type. If registered the user can search for lost animals of any specifications through the map.
<b>Flow of Events for Success Scenario:</b>	<b>1.1</b> User continues as a guest into the application and allows access to location. <b>1.2</b> User views range of location of surrounding animals. <b>2.1</b> User signs up / logs into the application. <b>2.2</b> User views range of location of surrounding animals and searches for any lost animal of any specifications.
<b>Flow of Events for Extension (Alternate Scenario):</b>	<b>1.1</b> User signs up / logs in or continues as a guest but not allow access to location. <b>1.2</b> Map view will not be accessible because it needs location access. <b>2.1</b> User continues as a guest and allows access to location. <b>2.2</b> User can view map but cannot search for a lost animal because they are not signed up / logged in.

<b>UC19</b>	Search for animal
<b>Initiating Actor:</b>	User.
<b>Actor's Goal:</b>	To search for an animal.

<b>Participating Actor:</b>	Institutions, Application Administration, Data base.
<b>Pre-Conditions:</b>	The user can search for an animal without registration.
<b>Post-Condition:</b>	The location of the animal will appear in the map as a range, and the information of the animal can be viewed by the participating actors.
<b>Flow of Events for Success Scenario:</b>	<p>1.1 User signs up / logs in or continues as a guest.</p> <p>1.2 The user clicks on the search bar on the map.</p> <p>1.3 An optional form appears for the users to fill animal's specifications, or they can directly search.</p> <p>1.4 Animal's location shall be retrieved from the database if there is match.</p> <p>1.5 Animal's location will appear as a range in the map.</p>
<b>Flow of Events for Extension (Alternate Scenario):</b>	<p>1.1 User can't access the map</p> <p>1.2 User can not find an animal.</p>

<b>UC23</b>	Send notification
<b>Initiating Actor:</b>	System administrator.
<b>Actor's Goal:</b>	To send notifications for other actors.
<b>Participating Actor:</b>	Application Administration, Data base.
<b>Pre-Conditions:</b>	The user should allow to receive notifications from the application.
<b>Post-Condition:</b>	The user will receive a notification about a nearby dangerous animal or when they receive a message.
<b>Flow of Events for Success Scenario:</b>	<p>1.1 User must be logged into the application, so they can receive a notification.</p> <p>1.2 The user receives a notification.</p>
<b>Flow of Events for Extension (Alternate Scenario):</b>	<p>1.1 User declines receiving notifications.</p> <p>1.2 User will not receive notifications.</p> <p>2.1 system fail to send notifications</p> <p>2.2 nothing</p>

---

## 3-Software Design Document (SDD)

### 1. Introduction

The design document is mainly about describing the targeted system product to the development team in order to provide the detailed system design architecture to be ready for the implementation phase. This document is a stable reference for all the team members from different backgrounds.

The first part of this document will be describing the interaction diagrams which are basically models that show how the system objects collaborate in some behavior. An interaction diagram comes in two types: the first one is the sequence diagram which we are going to include in this document. The sequence diagram shows the events from the actors to the system and the feedback returned to the actors. This kind of interaction diagram is depending on the use-case that was presented in the SRS document. The second interaction diagram is the collaborative model which is very similar to the concept of the sequence diagram that shows the objects as icons and the messages between the objects. Note that the collaboration diagram will not be presented in our document.

The second part of this document will be showing the system domain model which is the class diagram. The class diagram describes the static structure of the system by presenting it as classes, their attributes, methods/operations, and the relationships between these classes.

The third part of this document will be illustrating the system architectural style we chose, that reflects the basic strategy to build the system structure. It's generally come in three styles: shared data repository, or client-server, or the abstract machine/layered architectural style.

The fourth part of this document will be introducing the block diagram that is a high-level modularization that decomposes and identifies the subsystems in our system project. It helps the developers to understand the different inner layers that form the whole system.

The fifth part of this document will be presenting the state diagram, which describes the behaviour of the system objects and how it reacts to the events.

The last part of this document will be introducing the design of tests. We will be listing and describing the test cases that will be used for unit testing of our software. Also, we will discuss the test coverage, the integration testing strategy, and any plans related to testing the non-functional requirements that we mentioned earlier in this document.

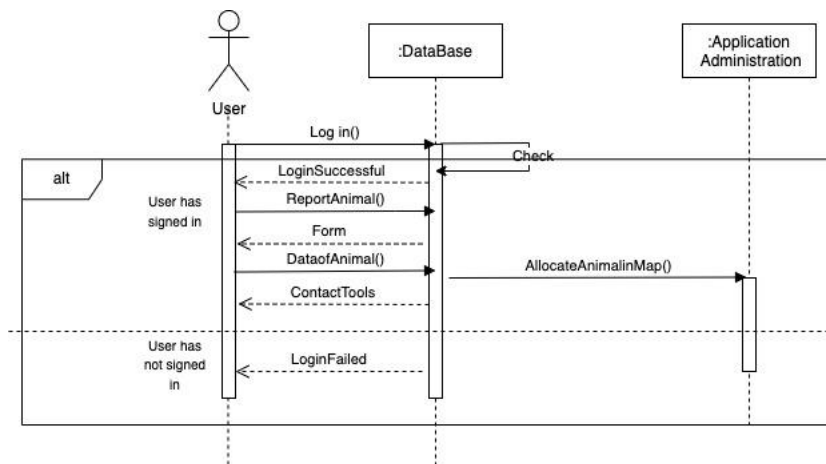
## 2.Glossary of Terms

TERM	DESCRIPTION
Stray animal	An animal that is homeless, abandoned, needs help, or not in the right place.
Notification	A message that is automatically sent to you on your mobile phone or computer.
Methodology	A body of methods, rules, and postulates employed by a discipline: a particular procedure or set of procedures.
Architecture	The manner in which the components of a computer or computer system are organized and integrated.
Structure	Something arranged in a definite pattern of organization.

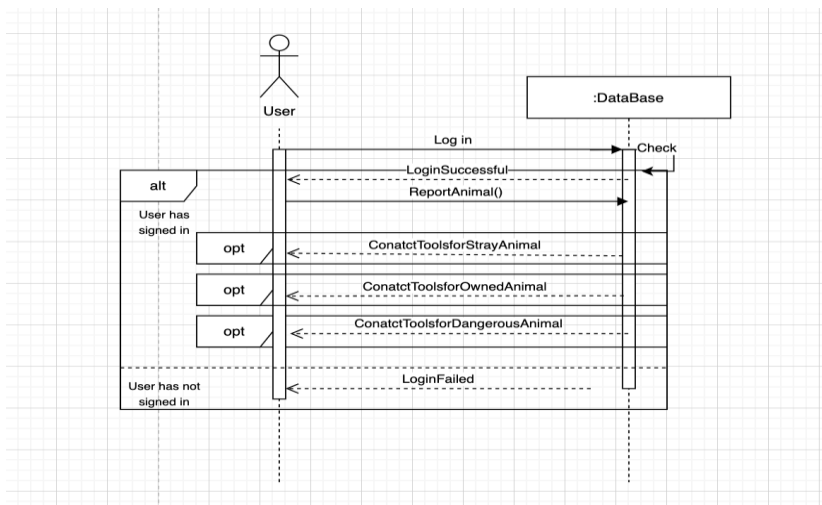
## 3. Interaction Diagrams

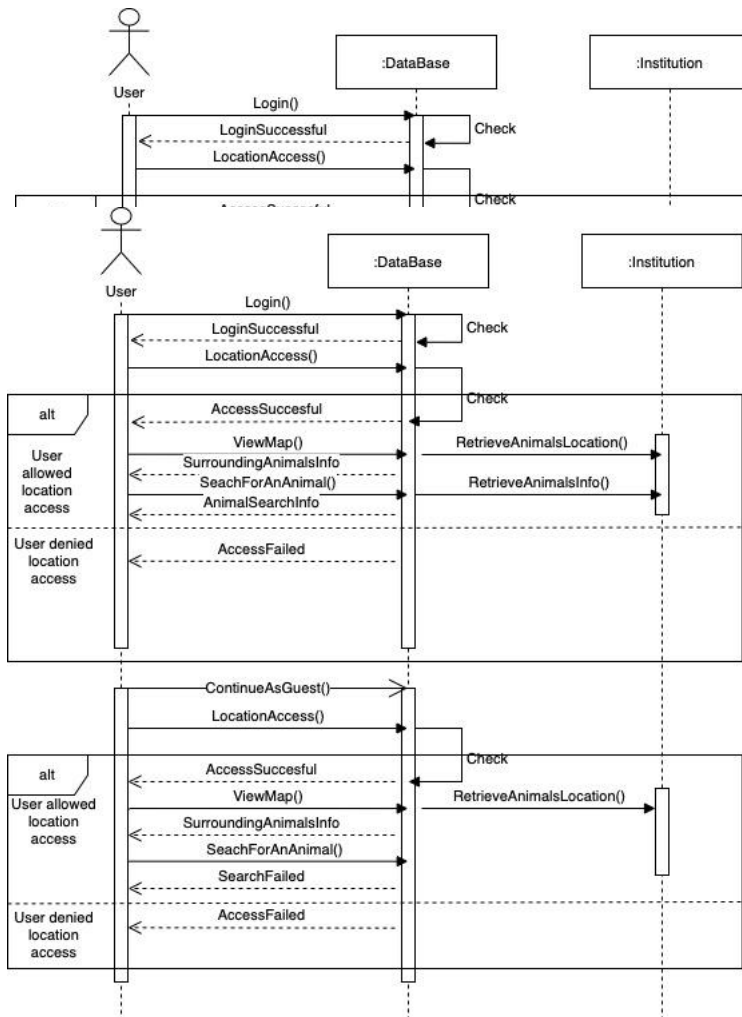
### 3.1 Sequence Diagrams

*Use Case: Report animal*

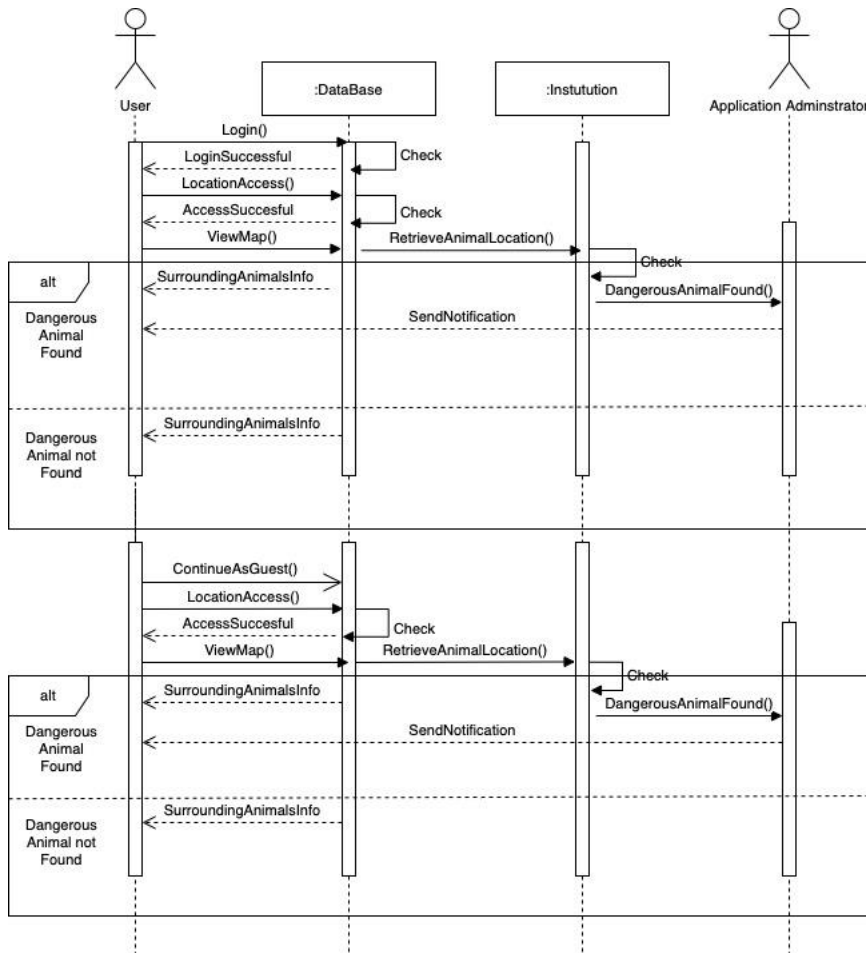


*Use Case: View contact*



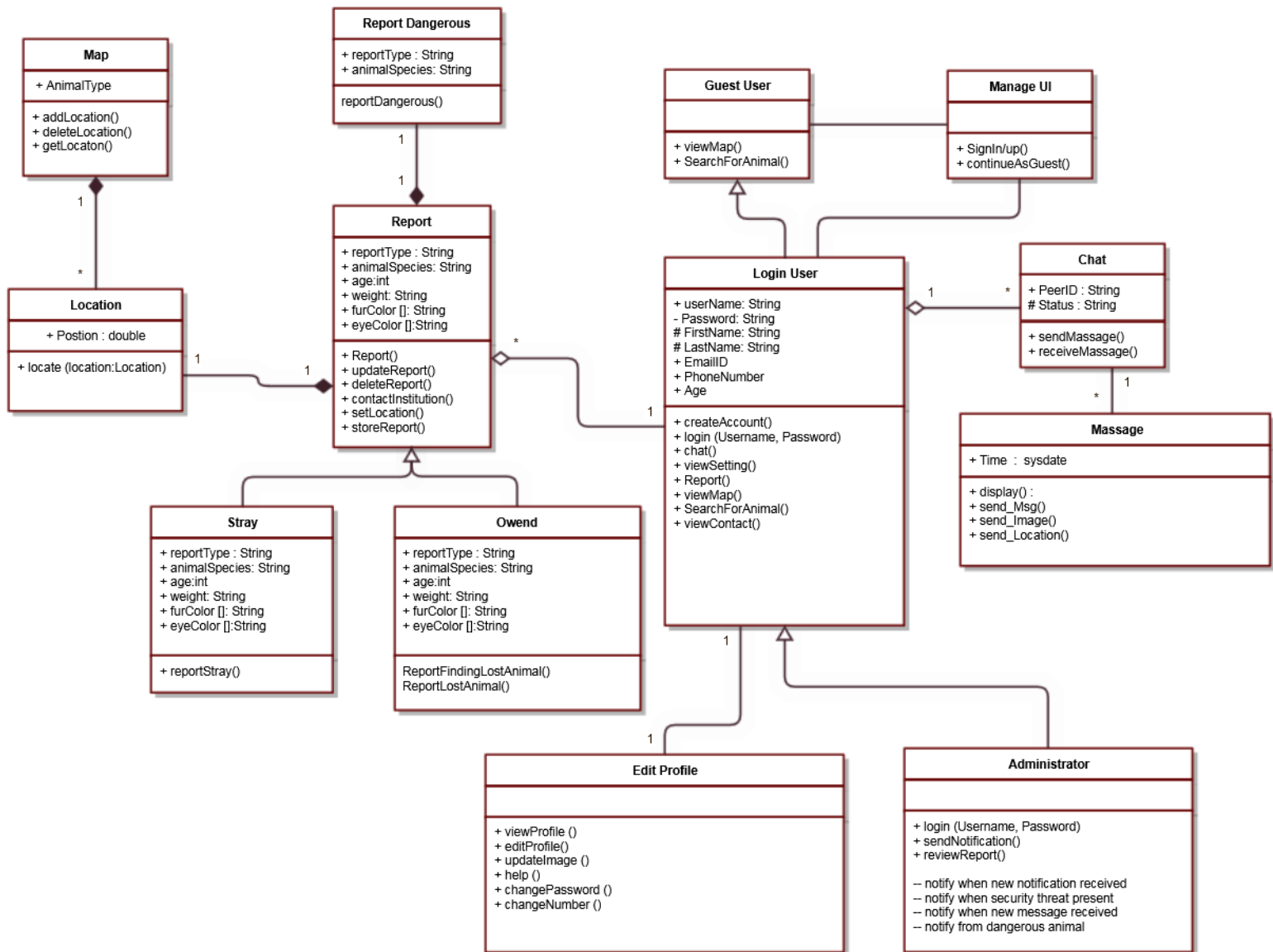
**Use Case: View map**



**Use Case: Send notification**

## 4.The System Structural Diagram

### 4.1 The Detailed Class Diagram “Software Artifact”



## 4.2 The Attributes and Methods description for each Class

### Class User:

#### Attributes:

- Username: represents the username of users.
- Password: represents the password of users.
- Firstname: represents the first name of users.
- Lastname: represents the last name of users.
- Phone: represents the phone number of users.
- Email: represents the email of users.
- IDnumber: represents the ID number of user.
- DateofBirth: represents the date of birth for users.

#### Methods:

- CreateAccount(): allow user to create an account in the application to get advantage of the applications services.
- Login(Username, Password): allow users to log in the application.
- Chat(): the method gives the ability to chat between other users to get information of animals.
- viewSitting(): view the sitting of the application.
- searchForAnimal(): the method gives the ability to search for animals.
- viewContact(): this method help users to view the contact tools.

### Class Chat:

#### Attributes:

- PeerID: represents an ID that helps to connect users together in chat room.
- Status: shows the status of the users, for example, Online or Offline.

#### Methods:

- sendMessage(): allow the user to send messages.
- receiveMassges():allow the user to receive messages.

### Class Massges:

#### Attributes:

- Time: shows the time of sending or receiving the message.

#### Methods:

- display(): print the message to users.
- send\_Msg(): apply the service of sending messages.
- send\_Image(): allow users to share images of animals.
- send\_Location(): allow users to share location of animals to get to them as fast as possible.

**Class Edit Profile:****Attributes:**

This class does not have any attributes.

**Methods:**

- viewProfile(): allow the users to view their profile.
- editProfile(): allow the users to edit their profile.
- updateImage(): allow the users to edit their image.
- help(): allow the user to ask for help.
- changePassword():allow the user to change their password.
- changeNumber():allow the user to change their number.

**Class Report:****Attributes:**

- reportType: shows the type of the report.
- animalSpecies: shows what is the species of the report animal.
- age: age of the reported animal.
- weight: weight of the reported animal.
- furColor: the colors of the animal's fur.
- eyeColor: the colors of the animal's eyes.

**Methods:**

- Report(): this is the method that allow to report animal, and can be inherited form the animal classes.
- updateReport(): updates the report of previously reported animals
- deleteRepoet(): deletes the report of previously reported animals
- contactInstitution():this method provides the contact tools of the institutions to the users that have reported an animal.
- setLocation(): set the location of the reported animal.

**Class Administration:****Attributes:**

*This class does not have attributes.*

**Methods:**

- login(Username, Password): since the administration is responsible for login in, this method accepts the username and password to check if the user is registered or not and give the permission to log in if the user is registered.
- sendNotification(): sending notification of the existence of a dangerous animal.

**Class GuestUser:**Attributes:

*This class does not have attributes.*

Methods:

- viewMap(): since guest users have limited services one of them is the ability to see the map, and this method help viewing the map.
- searchForAnimal():another services provided for guest users is searching for animals , this method give this ability.

**Class Manage UI:**Attributes:

*This class does not have attributes.*

Methods:

- signIn/up(): this method helps with signing in/up to the application.
- ContinueAsgeust(): this method allow user to have the experience of viewing the map or searching for an animal as a guest.

**Class Map:**Attributes:

- AnimalType: it stores the type of the animals either stray, dangerous, or owned.

Methods:

- addLocation(): this method adds a location of reported animals.
- deleteLocation(): this method used by the administrator to delete Location of previously reported animals.

**Class Location:**Attributes:

- Position: this attribute stores the coordinates of the animal in the map.

Methods:

- locate(location: Location): this method specifies the location of animals in the map.

**Class Stray Animal:**Attributes:

- reportType: shows the type of the report.
- animalSpecies: shows what is the species of the report animal.
- age: age of the reported animal.
- weight: weight of the reported animal.

- furColor: the colors of the animal's fur.
- eyeColor: the colors of the animal's eyes.

Methods:

- ReportStry(): when the user wants to report a stray animal this method will be used.

**Class Dangerous Animal:**

Attributes:

- reportType: shows the type of the report.
- animalSpecies: shows what is the species of the report animal.

Methods:

- ReportDangerous(): when the user wants to report a dangerous animal this method will be used.

**Class Owned Animal:**

Attributes:

- reportType: shows the type of the report.
- animalSpecies: shows what is the species of the report animal.
- age: age of the reported animal.
- weight: weight of the reported animal.
- furColor: the colors of the animal's fur.
- eyeColor: the colors of the animal's eyes.

Methods:

- ReportFindingLostAnimal():when the user find a lost owned animal and report it this method should be used.
- ReportLostAnimal(): when reporting a lost owned animal this method should be used.

## 5. The System Architecture and design

### 5.1. The System Architectural Style

#### 5.1.1 The System Organization

In our application we chose the Client-Server Architecture to structure our system because we provide specific services to different users, maintain information and access stored data which results in our system needing the permanent data storage. We plan to store the permanent information in our storage subsystem the Database that will be in the server part of the architecture.

We expect the server to handle the various requests and the access to services from different concurrent users and receive or send information. The Client-Server Architecture makes effective use of networked systems as well as it places the execution of software next to the user which helps boost efficiency.

#### 5.1.2 The Data Flow Model

Object-oriented programming (OOP) is a form of programming that heavily relies on classes and objects, which means that it is a methodology that provides the same program code but allows you to create instances of objects that perform different tasks. This method makes the writing and implementation of a program much simpler and easier. It also supports code reuse and enables objects to interact and complete tasks to form the complete system hence why we chose this approach for our system.

OOP integrates application and database creation into a single data model and language environment. It embraces data abstraction, encapsulation, and inheritance while allowing for object communication and recognition.

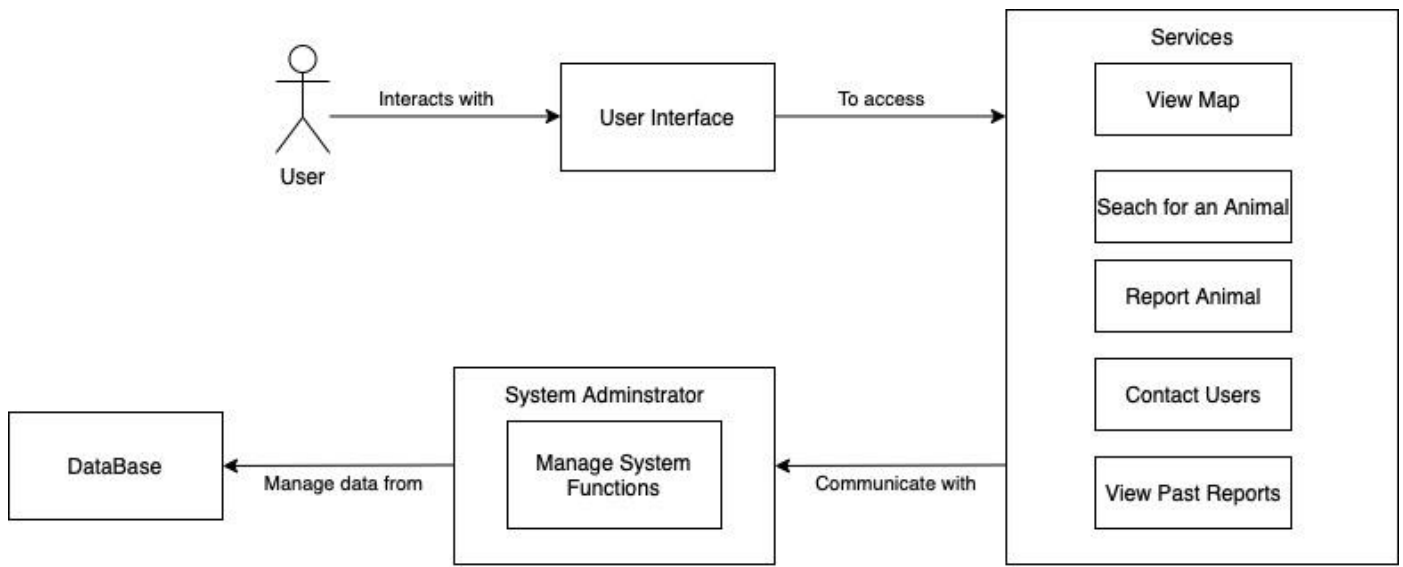
Why does object-oriented programming suit our system the best? Because our system's objects have similar attributes, are derived from common classes, and share some of the same methods that are implemented differently. Finally, our program is available on Android which is an OOP based platform.

#### 5.1.3 The Control Model

Our system access's location, displays a map, sends notifications to users if a dangerous animal is nearby, files reports, supports contacting different institutions and communicating with different users when needed. All these operations imply that it is a fast system, therefore the use of real-time system is a must. That is why we chose the Event-Driven Architecture model (EDA) which is a system that responds quickly to external and internal events, is very efficient to develop, gives the

user a better experience and is resilient; If a service fails, it can automatically restart and replay events from the event bus. After doing so, it can update other services by pushing events of its own to the stream.

## 5.2. Identifying the Subsystems

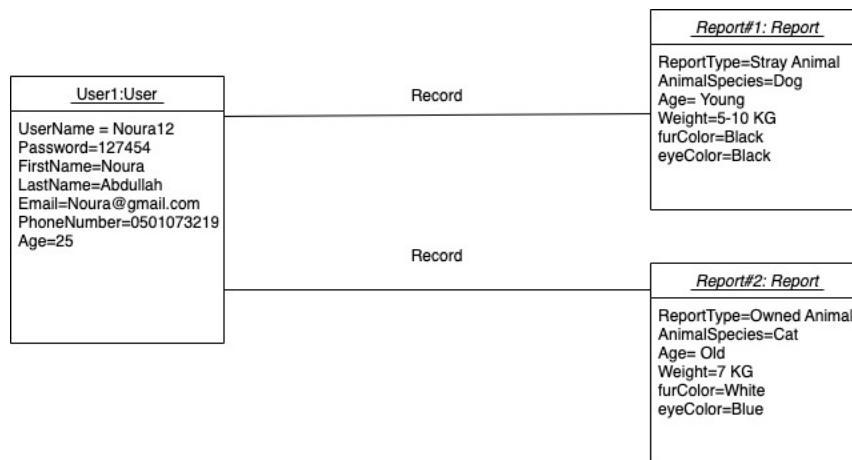




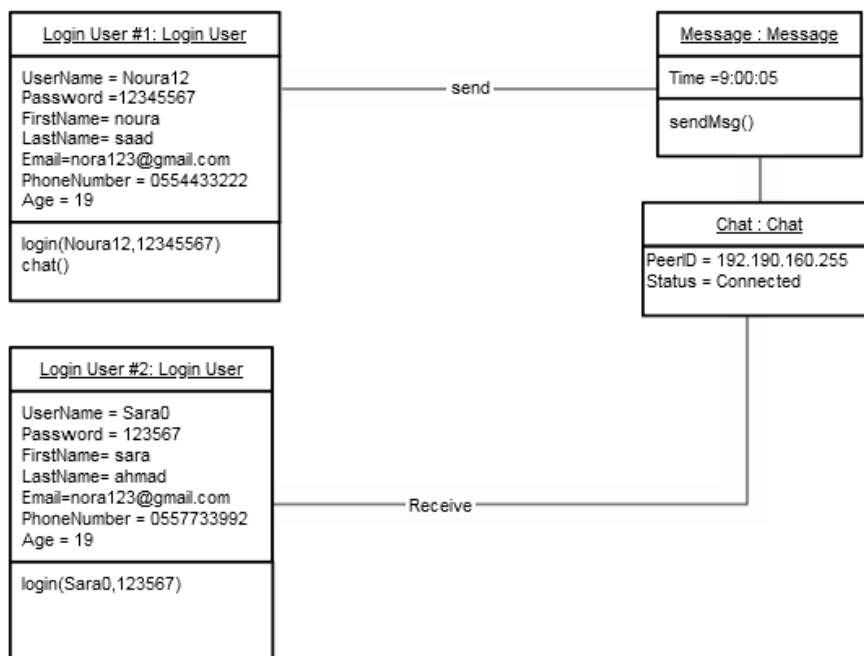
## 6. The Object Diagrams

### 6.1. The Object Diagrams

-Object diagram that shows the flow of the report process for a specific user that record a report of stray and owned animals.



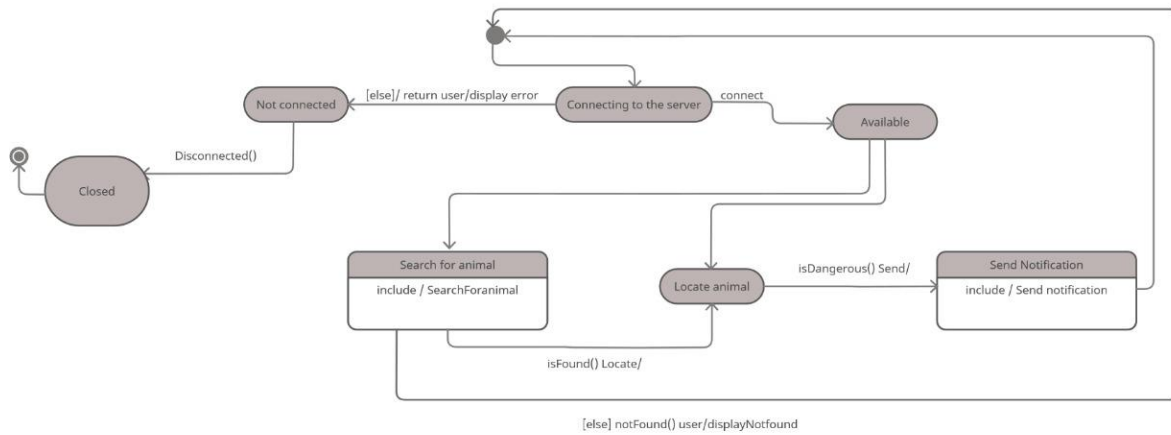
-Object diagram that shows the flow of the communication process between two different users



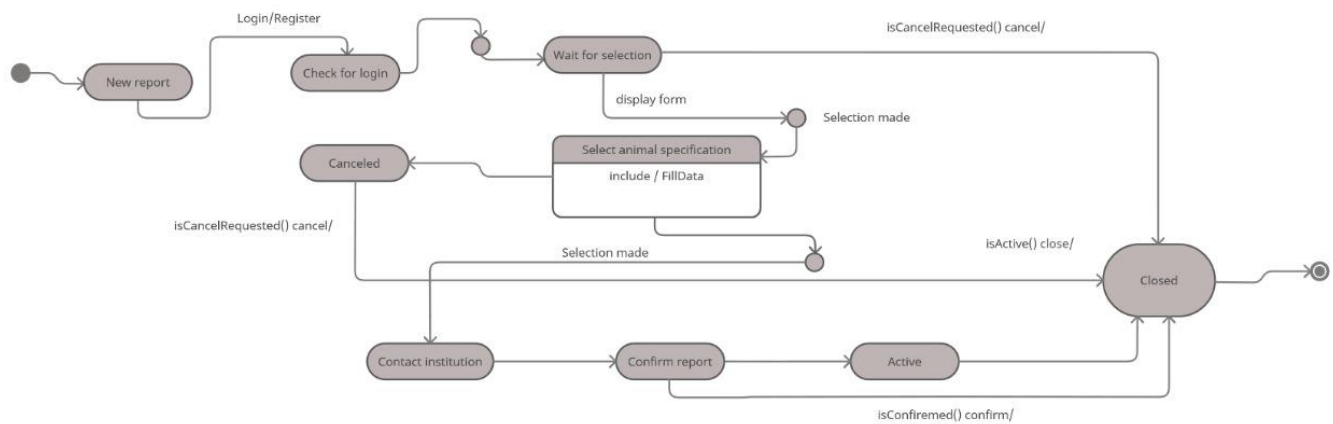
## 7. The System Behavioral Diagrams

### 7.1. The State Diagram

#### Be a Rescuer Locate State diagram



#### Be a Rescuer Report State diagram



## 8. The User Interface Design



Figure 1. Log in screen.



Figure 2. Sign in screen.



Figure 3. Username and Password screen.



Figure 4. Map screen.



Figure 5. Show animal information on map screen.



Figure 6. Edit profile screen.



Figure 7. Report screen.



Figure 8. Camera.

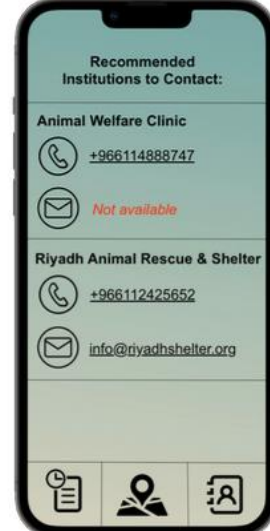


Figure 9. Contact tools screen.



Figure 10.  
Previous reports  
screen.



Figure 11. Messages  
screen.



Figure 12. Map  
screen as a  
guest.

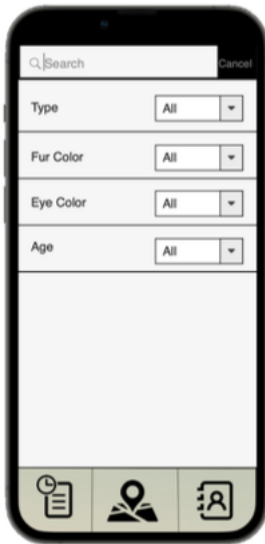


Figure 13. Search bar screen.



Figure 14. Receiving a notification.

## 9. The Design of Tests

### 8.1 Unit Testing

Test Case ID	Use case tested	Description	Test data/input	Expected Output	When it considered pass/fail	Actual Output	Any comments
TC1	UC13/14	Test if the system allow user to log in	Valid username and password	The system will show the home page to user	-Pass if the user is able to log in and view the home page screen -Fail if the user is not registered	As expected	none
TC2	UC2	Tests if the system allow user to report an animal	The user can press report button	The system will show the report button to user	-Pass if the user is able to press the report button -Fail if the user is not registered	As expected	Reporting any type of animals
TC3	UC1	Test if the system will show the contact tools properly	User can report animal and enter the information of the animal	The system can show the contact tools to user	-pass if the user is able to view the contact tools -Fail if the user did not complete the reporting process	As expected	The contact tools differs according to the type of the animal
TC4	UC3	Test if the system allow user to send and receive messages	If the user is logged in and can click on messages button	The system allows the user to start new chat or view chat	-Pass if the user is able to take the message button -Fail if the user is not registered	As expected	The messages are to connect between the reporters (reporter of lost animal or reporter for finding animal)
TC5	UC4	Test if the system allow user to take pictures	Press the photo button	The system allows the user to take pictures	-Pass if the user is able to take pictures -Fail if the picture button did not appear	As expected	none
TC6	UC5	Test if the system allow user to delete reports	Press deletes report button	The system allows the user to delete reports and apply the instructions	-Pass if there is a report with same information in the data base	As expected	none

					-Fail if the report is not met in data base		
TC7	UC6	The system will allow the user to update report	Press update report button	The system allows the user to update reports and apply the instructions	-Pass if there is a report with same information in the data base -Fail if the report is not met in data base	As expected	none
TC8	UC8	Test if the system allow user can specifically to lost(owned) pet	The user logged in and specify the type of the animals(owned)	The system shows the contact tools of the owned pet to user	-Pass if the user is able to press the report button and specify the animal type (owned) -Fail if the user is not able to press the report button and could not specify the animal type (owned)	As expected	While reporting the user can determine the type of the animal
TC9	UC9	Test if the system allows the user to select location on map	The user can select location from the location selection bar	The system allows the user to select the location to track their lost pet	-Pass if there is a location met the users' entered location -Fail if there is no lost pet	As expected	none
TC10	UC10	The system should allow user to report finding lost pet	The user can press report button	The system will show the report button to user	-Pass if the user is able to press the report button -Fail the user could not press the report button	As expected	none
TC11	UC11	Test if the system allow user can specifically to lost(dangerous) pet	The user logged in and specify the type of the animals(dangerous)	The system shows the contact tools of the dangerous pet to user	-Pass if the user is able to press the report button and specify the animal type (dangerous) -Fail if the user is not able to press the report button and could not specify the animal type (dangerous)	As expected	While reporting the user can determine the type of the animal
TC12	UC12	Test if the system allow user to specifically report stray animal	The user logged in and specify the type of the animals(stray)	The system shows the contact tools of the stray pet to user	-Pass if the user is able to press the report button and specify the animal type (stray) (dangerous) -Fail if the user is not able to press the report button and could not specify the animal type (stray)	As expected	While reporting the user can determine the type of the animal
TC13	UC15/16	Test is system allow user to view or edit their profile	The user should have an account and logged in	The system will show the user the profile and allow the user for edit	-Pass if the if the user can view/edit their profile -Fail if the user is not registered	As expected	none

TC14	UC17	Test if the system can view the map to user	The user either guest or logged in can click on the map	The system will show the map to user	-Pass if the user can view the map -Fail if the user cannot press on the map button	As expected	none
TC15	UC18	Test if the system can show the report history to user	The user press on history button	The system will display the previous reports	-Pass if the user can view the reports' history -Fail if the user is not logged in	As expected	none
TC16	UC19	Test if the system allows user to search for animal in map	The user can click on the map	The system will show the map to user and can search using search bar	-Pass if the map shown to user -Fail if the map does not appear	As expected	none
TC17	UC20	Test if system can show the animals' information to users while viewing the map	The user either guest or logged in can click on the map then press on animals' photo	The system shows the animals information	-Pass if user can read the information of the lost or owned animal -Fail if the user can not see any information	As expected	If the animal is dangerous, for the security of the animal limited information are displayed
TC18	UC21	Test if the system shows all previous reports to the administrator	The administrator can view the data through the data base	The system allow administrator to read information	-Pass if there are reports in data base -Fail if there are no previous reports	As expected	none
TC19	UC22	Test if the system allows the users to specify the location of the animals as a range	The user can press the map button and the specify the location	The system will view the map and allow to specify the location	-Pass if the user id logged in -Fail if the user is not logged in	As expected	During the reporting process
TC20	UC23	The system is sending notification to user	User should be registered and there is a dangerous animal nearby	The system will send notification successfully	Pass if the notification is sent	As expected	The notification is to alert people if any dangerous animal nearby



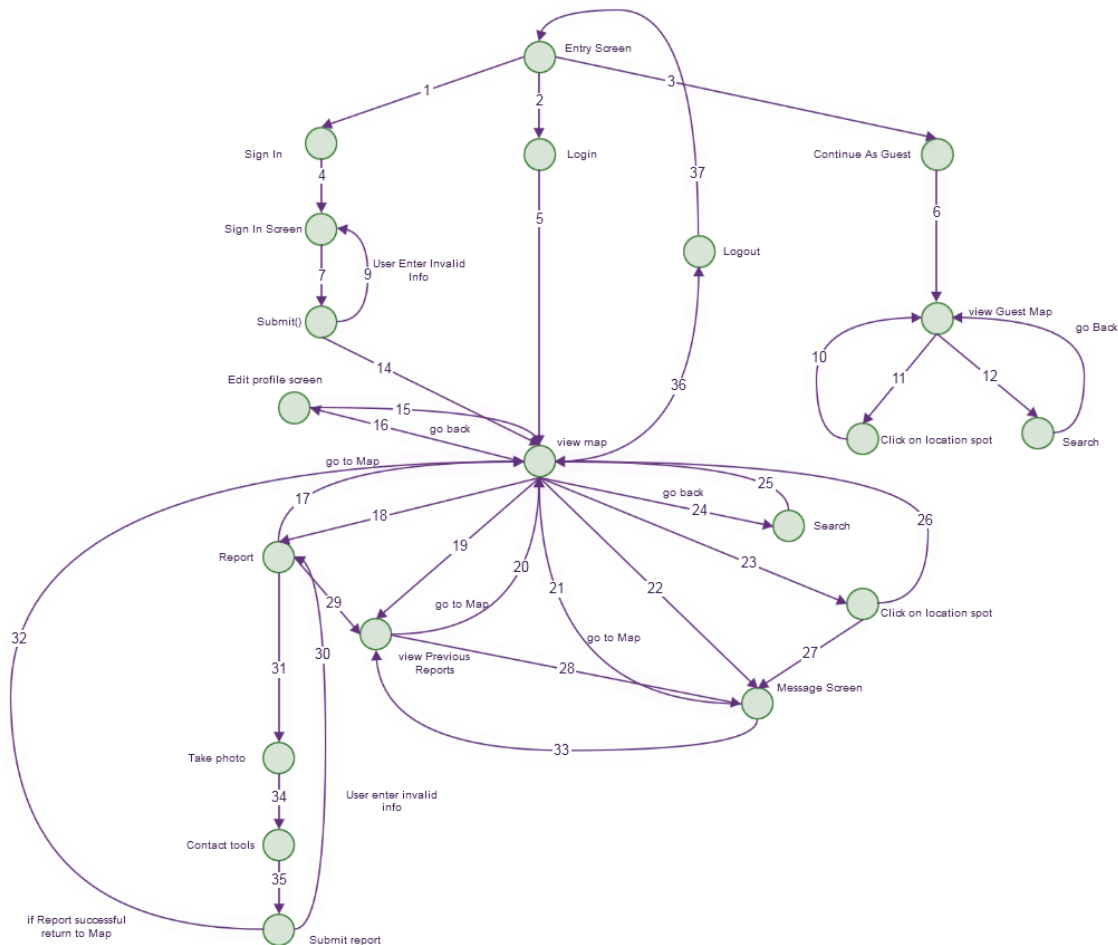
## 8.2 Integration Testing

Test Case ID	Description	Test data/input	Expected Output	When it considered pass/fail	Actual Output	Any comments
TC21	Check the interface link between the sign in / continue as guest page and the map	Enter sign in information (Username and password) and press the sign in button or choose to continue as a guest and allow location access	To be directed to the map page	Pass if the user can sign in or continue as guest and allows location access then views the map screen. Fail if the user is not able to sign in or continue as guest or if they deny location access then will not be able to view map screen.	As expected	None
TC22	Check the application link between the Database and the filed report information	Filed animal picture, type, state, eye color, fur color, age, and comments	The information gets stored in the Database and the report status can be tracked or deleted by the user	Pass if the report information is stored and can be retrieved or deleted. Fail if the report information is not stored or cannot be retrieved or deleted.	As expected	None

## 8.3 Acceptance Testing

A user who tested our application system thought that it was easy to use and understand and it helped them save a stray kitten in their neighborhood and check if their area is safe from any dangerous animals. They thought that the map idea was clever, but they suggested that it should show a more detailed location range of the animal, we will take that suggestion into consideration, but we did limit the animal's location range for a reason and that being that we don't want our map feature to be misused or cause any harm. The user also suggested that we should add a pet adoption feature, we will definitely think about that improvement in the future.

## 8.4 path testing

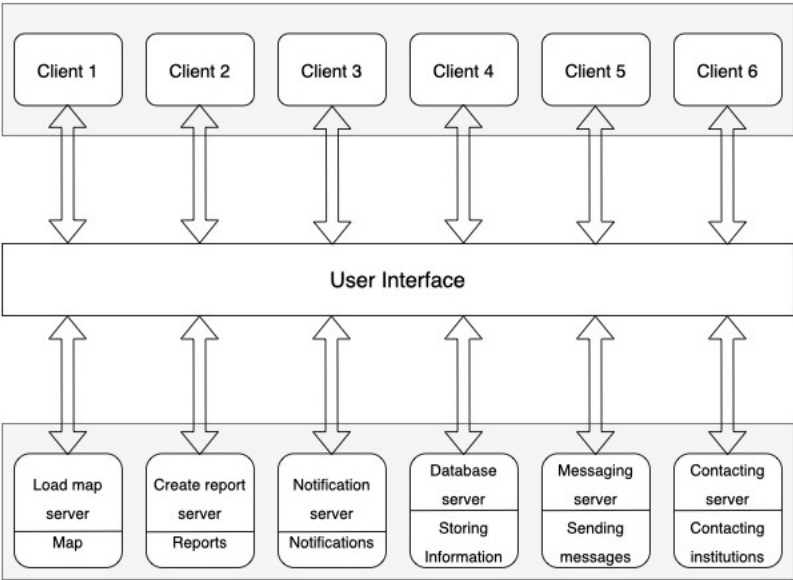


Number of paths = 8

Cyclomatic Complexity for each path:

- Edit profile = 11 edges - 8 Nodes +2 = 5
- Submit report = 16 edges - 11 nodes +2 = 7
- View previous reports = 14 edges - 8 nodes +2 = 8
- Message screen = 14 edges - 8 nodes +2 = 8
- Click on location spot= 12 edges - 8 nodes +2 = 6
- Search = 11 edges - 8 nodes + 2 = 5
- Guest clicks on location spot = 4 edges - 4 nodes + 2 = 2
- Guest search = 4 edges - 4 nodes +2 = 2

8.5 Architecture Diagram



10. Individuals Contributions Breakdown

Member name	Deliverable	Your Contribution (in details)
-------------	-------------	--------------------------------

**Aljwhra Almakhdoub**

Contribute with an idea (devices battery app)

Wrote the introduction and statement of problem

Naming the app

Customer Statement of Requirements:

1.4 and 1.5.

Glossary of Terms:  
Notification and Function.

User Requirements:  
13 User Requirements.

User Interface Mockup:  
9 Interfaces.

Use Case Casual Description:  
Log in / Sign up.  
View Profile.  
View Map.  
View Reports History.

Use Case Fully Dressed Description:  
UC18.

SDD-part1: Interaction diagrams: View map, Search for Animal and Send Notification.

SDD-part2: The System Architecture and Design and The User Interface Design.

SDD-part3: Glossary of Terms, Integration Testing and Acceptance Testing.

Final Report: Architecture Diagram

Implementation: Map Interface-Search interface.

**Ghaida Alkhudair**

Contribute with an idea (office hour)

Wrote the project objective and related work

Naming the app

Customer Statement of Requirements:

1.1 and 1.2.

Glossary of Terms:

Stray animal and Inhabitant.

User Requirements:

6 user Requirements.

User Interface Mockup:

4 interfaces.

Stakeholders.

Actors and Goals.

Use Case Casual Description:

View contact.

Send message.

Receive message.

Report animal.

Use Case Fully Dressed Description:

UC3.

UC1

SDD-part1: Interaction diagrams: Sequence diagram (Report animal and View contact).

SDD-part2: Attributes and Methods description for each Class and Object diagram that shows the flow of the report process.

SDD-part3: Unit Testing.

Implementation: Report Interface-Contact tools interface.

**Raghad Aljuhaimi**

Made the WhatsApp group

Contribute with an idea (animal map)

Wrote the project scope in the proposal

Customer Statement of Requirements:  
Introduction and 1.3.

Glossary of Terms:  
Inhabitant, Collisions, and full stack.

User Requirements:  
18 User Requirements.

Application logo.

Use case diagram

Use Case Casual Description:  
UC5 to UC12 and UC19 to UC24

Use Case Fully Dressed Description:  
UC19 and UC23.

SDD-part1: Behavioral diagram: state diagram (Report, Locate)

SDD-part2: Class diagram structure, Object diagram that shows the flow of the report process.

SDD-part3: Path Testing and Cyclomatic Complexity calculation.

Implementation: History interface.

## References

- [1] 15th O. 2014, "Wildlife causes problems in urban environments," *American City and County*, 15-Oct-2014. [Online]. Available: <https://www.americancityandcounty.com/2014/10/15/wildlife-causes-problems-in-urban-environments/> . [Accessed: 04-Mar-2022].
- [2] S. Hawker, *Colour oxford English dictionary*. Oxford: Oxford Univ. Press, 2011.
- [3] *Dictionary.cambridge.org*. 2022. *Cambridge Dictionary / English Dictionary, Translations & Thesaurus*. [online] Available at: <<https://dictionary.cambridge.org/>> [Accessed 12 February 2022].
- [4] Wikidata.org. 2022. *productive animal*. Available at: <https://www.wikidata.org/wiki/Q1797813>
- [5] *The Saudi Wildlife Authority*. [Online]. Available at : <https://mewa.gov.sa/en/Partners/Pages/%D8%A7%D9%84%D9%87%D9%8A%D8%A6%D8%A9-%D8%A7%D9%84%D8%B3%D8%B9%D9%88%D8%AF%D9%8A%D8%A9-%D9%84%D9%84%D8%AD%D9%8A%D8%A7%D8%A9-%D8%A7%D9%84%D9%81%D8%B7%D8%B1%D9%8A%D8%A9.aspx> . [Accessed: 03-Mar-2022].
- [6] "Diagramming Software & Team Collaboration Tools," Gliffy.com. [Online]. Available: <https://www.gliffy.com/>. [Accessed: 14-Apr-2021].
- [7] "خدمات بلدي" منصة بلدي [Online]. Available: <https://balady.gov.sa/Apps/App?id=1> . [Accessed: 29-Jan-2022].
- [8] "إضافة تعليقات" أمانة منطقة الرياض، 940 أمانة [Online]. Available: <https://www.alriyadh.gov.sa/ar/mobapps/Pages/Amanah940.aspx> . [Accessed: 29-Jan-2022].