SE 216 SOFTWARE PROJECT MANAGEMENT

HOME AGAIN

A Mobile Application For Finding Lost Pets

SOFTWARE DEVELOPMENT PLAN

Beyza ÖZGÜR & Bora YÖRÜK & Birkan SARIBACAK & Kubilay KALKAN

6/13/2021

CONTENTS

1. OVERVIEW

1.1. Problem Definition

When pets get lost, the pet owners try to find them via social media apps such as Instagram and Twitter. In some cases (e.g., the earthquake occurred on October 30, 2020 in İzmir), this situation creates a complexity, and it takes a lot of time to find the pets since there is not a specific application which is used for this purpose. Since pets are not used to live outside their home, this can be a dangerous experience for them.

1.2. Background Information

Our project is being undertaken to help people who lost their pets to find them easily via our application "*Home Again*". Our mission is to create a platform for lost pets, so the owners can use our mobile application instead of trying to find them via other social media applications.

This platform will be used for everyday usage. If a pet got lost or found, people can report the situation on our application.

This platform will only help pet owners and the people who found pets to get each other's contact informations via direct message feature of our application. The communication between pet's owner and the people who found the pet is not our responsibility.

Firstly, users need to create an account to see or share posts. After they login, they will be able to see missing reports of the pets, or they can create one. Missing reports can contain a photograph of the pet, the city name and the time which it got lost and its breed. Users will be able to search reports by city name and see the results from this city rather than seeing all the reports. They can also filter the reports by breed. Additionally, users can share a contact address through direct message such as phone number, so that people can contact them. We are planning to record these missing report data to a database, so that other users can also access the data from same database.

If other people who use our application have seen the pet, they can contact the pet owners. This will give an opportunity to owners to find their pets.

Finally, we will show some of the shelters' and animal associations' donation links to increase awareness on stray animals. We are not responsible for payment transactions. Our aim is just to share reliable donation sources with our users.

For the implementation of this project, we are going to use Java programming language. As IDE, we are going to use Android Studio since Android is the most used mobile operating system. GUI’s will also be implemented using this IDE. MySQL will be used to store the necessary account data. GitHub is needed as the version & management control.

1.3. Objectives

- Finding lost pets.
- Creating a specialized platform for finding animals and make it easier to find animals.
- Sharing donation links of the shelters so that people can donate to stray animals.
- Creating an easy-to-use interface so that people of any age can use our application.
- Raising awareness on stray animals.

1.4. Scope

As a result of the project, an application for finding pets will be created. People will be able to reach their lost pets and extra awareness will be created for stray animals.

2. HIGH-LEVEL FUNCTIONALITY

2.1. Functional Requirements

1. Users can create an account to log in to the system using username and password.
2. Users can create missing reports via photos and informations of their pets.
3. The system allows users to see other reports created by other pet owners or pet finders.
4. Users can filter missing reports by city name so that they can see only the reports from a specific city.
5. Users can filter missing reports by animal breeds (such as cat and dog) so that they can see only the reports from a specific type.
6. Users can make modifications on missing reports and contact informations.
7. The system allows users to delete the missing reports they created.
8. Users can report unrelated or harmful posts.
9. Users can delete their accounts.
10. The system shows users the donation links of some animal associations and shelters.

2.2. Non-Functional Requirements

1. The data of the users should be protected with laws such as KVKK.
2. The system should shut down in case of cyber attack.
3. The processing of requests should take at most 10 seconds.
4. The system should be working for the mobile phones included Android 7.0 and the upper versions.
5. The system should support at most 150.000 users for a quality performance.
6. Screen refresh time should be at most 30 seconds.

3. STAKEHOLDERS

In our project, we have 5 types of stakeholders;

- Development Teams Within The Company
- Users
- Institutions
- Animals and Animal Related Associations
- Shareholders

3.1. Development Teams Within The Company

- **Developers of The Project**

  If the project succeeds, their reputation will increase and they will gain experience, become more recognized in the market. But if the project fails, they will lose time and reputation.

- **Competitors**

  If the project fails, they will have more budget than the developer team of the project. They will have more chance to be involved in the projects. If it succeeds, they will have fewer resources in future projects.

3.2. Users

- **Pet Owners**

If the project succeeds, they will have a chance to find their pets via our application. If it fails, they will lose a lot of time to find their pets.

- **People Who Found Pets**

If the project succeeds, they can easily find the pet owners. If the project fails, they will not be able to find a new home for pets easily.

- **Donators**

If the project succeeds, they can help animals to get better conditions to help. They can trust and donate to the links of reliable animal associations provided by our app. If it fails, they may not make certain of the donation info, so they might not donate. The trust for the app may decrease.

3.3. Institutions

- **Government**

If the project is successful, taxation will be imposed by the government. It will not be affected much, if project fails. It will not get the taxes.

- **Banks**

If the project succeeds, they can become more recognizable via advertisements and they can benefit from donations in terms of interest. If is fails, banks will not be affected much.

3.4. Animals and Animal Related Associations

- **Lost Pets**

  If our project becomes successful, lost pets will be found and returned to their owners. If the project fails, their lives will be in danger.

- **Stray Animals**

  If the project succeeds, they will have food, better health & life conditions via the donations. If it fails, they may not get health care or extra food.

- **Pet Shops**

  If the project succeeds, they will be able to sell more products such as animal food. It will be a positive impact for them. If it fails, they may not sell extra products to shelters.

- **Animal Shelters**

  If the project succeeds, they will have extra money to use for stray animals which is provided by donators (by accessing donation links from our app). If it fails, they will not have extra money.

3.5. Shareholders

If the project is successful, shareholders will gain money when the application is being used and they will be able to find sponsors. They will feel good for helping animals. Their self-confidence may increase as well as their recognition. If the project fails, they will lose time, money and reputation.

4. PROJECT STAFFING

In this project, we need six roles (software project manager, requirements engineer, lead designer, coder, tester and toolsmith) for our four team members.

1. **Software Project Manager:** Does the planning, scheduling, budgeting, and leads the development team. Enhances communication between members and follows the progression of the project from beginning to the end. Checks if the members do their tasks or not. (Beyza ÖZGÜR)

2. **Requirements Engineer:** Manages requirements by elicitation, elaboration, negotiation, specification, validation. Uses user story cards to specify functional and non-functional requirements. Documents all the specified requirements. (Birkan SARIBACAK)

3. **Lead Designer:** Designs interactive and easy to use graphical user interfaces. Creates prototypes and interacts with users/customers to see if the designes satisfy their requirements and needs. Documents, clarifies and maints design till the end of the project. (Kubilay KALKAN)

4. **Coder:** Designs necessary classes and databases. Implements the whole system and the graphical user interfaces. Debugs the program to fix errors. Refactors program for a better code quality and understandability. (Beyza ÖZGÜR)

5. **Tester:** Checks and tests the codes written by coder based on the requirements and requests of the users. Plans and reports the results of product reviews and testing. (Bora YÖRÜK)

6. **Toolsmith (Tools Expert):** Installs, troubleshoots, upgrades and maintains the software programs that will be used by the development team. (Kubilay KALKAN)

5. SOFTWARE PROCESS MODEL

5.1. Necessary Needs From The Organizational Process

1. Rapid and frequent feedbacks from customers are needed.
2. Easy to use software tools are needed to make team work easier.
3. Story cards written by customers/users are needed to specify requirements.
4. Pair programming groups are needed to check each other's work.
5. Collective ownership is needed so that all developers can work in every step of the software processes and anyone can contribute to every area.
6. Refactoring is needed to maintain simplicity.
7. Automated unit test framework is needed to write tests for a new functionality.
8. Incremental planning is needed based on new story cards.
9. A task is need to be integrated into the whole system as soon as it is completed.

5.2. Unnecessary Needs From The Organizational Process

1. No need for excessive design and documentation at the end of each software process.
2. No need for a large team consisting of many members.
3. No need to finish one process to start another process.
4. No need to plan all stages in advance.
5. No need for long working hours.
6. No need to go back to previous process to add new changes
7. No need for so many details on user stories.

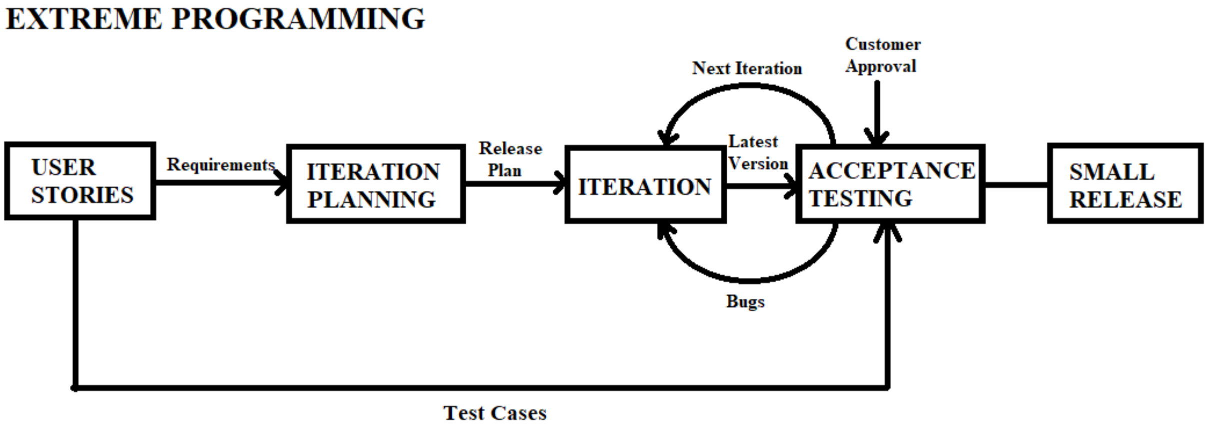
5.3. Software Process Name: Extreme Programming (XP)

Software Process Description:

Extreme programming (XP) is one of the most used agile development techniques. It focuses on the simplest version of the product that can work. Refactoring is important to maintain the simplicity and improve the understandability of the software. The planning is incremental and iterations occur within processes. XP is an extreme approach to iterative development. For example, new versions may be built several times a day and built is successful only if tests are successful. Therefore,test-first development is a must. Additionally, full-time customer involvement and feedback is necessary in XP. User requirements are written on user story cards.

These story cards should include role of the user, goal and reason. Development team breaks the story cards into tasks to be implemented. Pair programming and collective ownership are also principles of this method. The team is small and it consists of 2-10 people. They work in pairs and every member work on all areas of the system. All of the team members are responsible for all of the codes. It is a good approach for development of small releases.

5.4. Software Process Model:



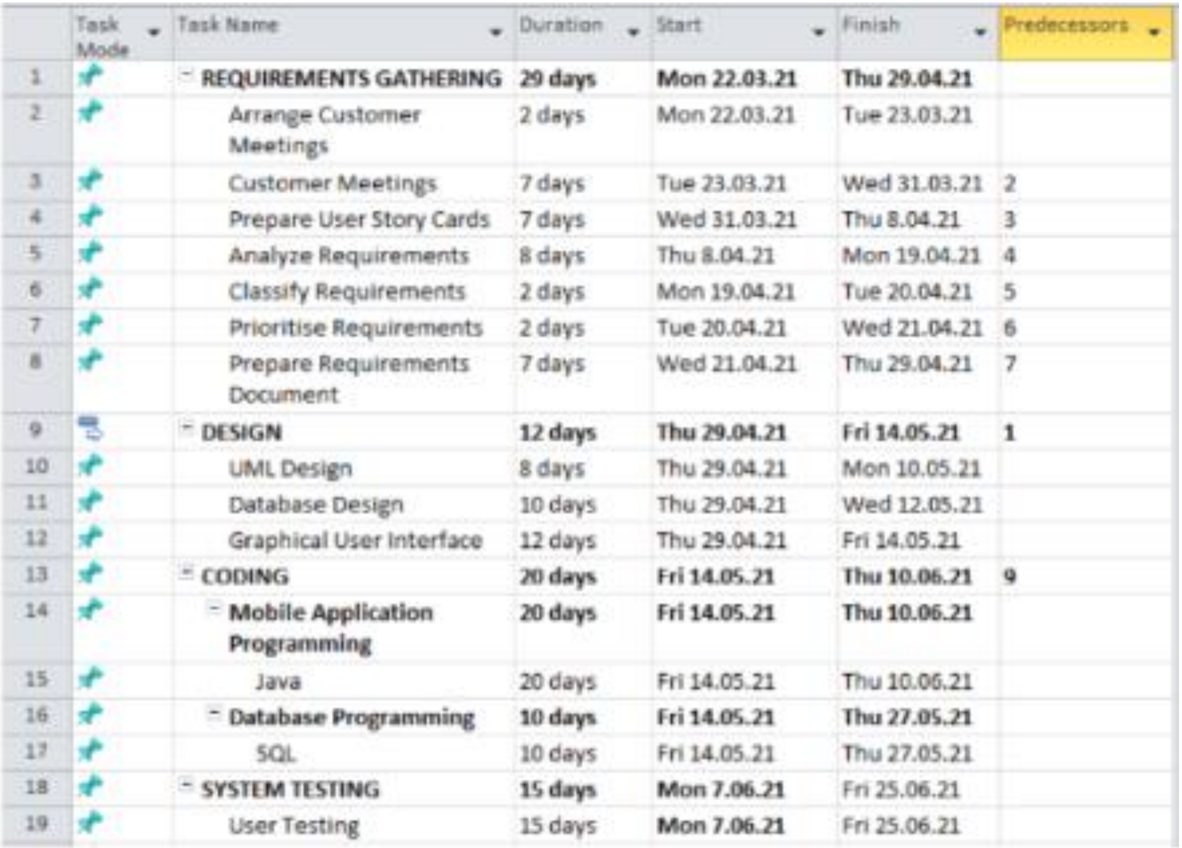
5.5. Reasons To Choose This Model:

- Our team consists of four members and XP approach needs a small team.
- It is suitable for small applications like ours.
- It saves time because there is no excessive documentation.
- We don’t need to finish one process to start another process.
- Its cost is much less than other incremental development models.
- Its speed to change is high.
- The time to market is short.
- Refactoring is inexpensive.

- Because of the constant customer involvement, it is more possible for user to be satisfied with the end product.
- Incremental development reduces the risk of marketing.
- All members will have equal responsibility of all the codes.
- Refactoring improves the understandability of the codes.
- Simple design that works is enough.
- It provides minimum useful set of functionalities.
- The probability of code failure is less because of test-first development.
- XP avoids long working hours because it may cause inefficient codes and unproductivity.

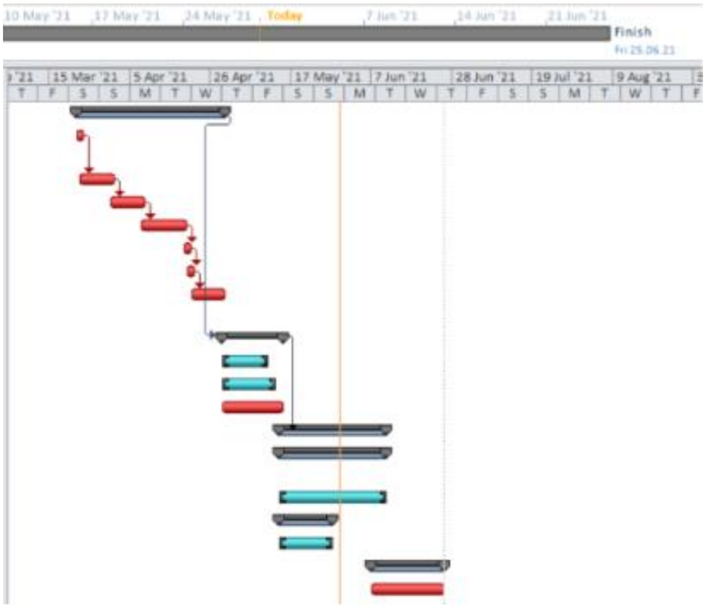
6. SCHEDULE AND EFFORT

Schedule:



	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		**REQUIREMENTS GATHERING**	**29 days**	**Mon 22.03.21**	**Thu 29.04.21**	
2		Arrange Customer Meetings	2 days	Mon 22.03.21	Tue 23.03.21	
3		Customer Meetings	7 days	Tue 23.03.21	Wed 31.03.21	2
4		Prepare User Story Cards	7 days	Wed 31.03.21	Thu 8.04.21	3
5		Analyze Requirements	8 days	Thu 8.04.21	Mon 19.04.21	4
6		Classify Requirements	2 days	Mon 19.04.21	Tue 20.04.21	5
7		Prioritise Requirements	2 days	Tue 20.04.21	Wed 21.04.21	6
8		Prepare Requirements Document	7 days	Wed 21.04.21	Thu 29.04.21	7
9		**DESIGN**	**12 days**	**Thu 29.04.21**	**Fri 14.05.21**	**1**
10		UML Design	8 days	Thu 29.04.21	Mon 10.05.21	
11		Database Design	10 days	Thu 29.04.21	Wed 12.05.21	
12		Graphical User Interface	12 days	Thu 29.04.21	Fri 14.05.21	
13		**CODING**	**20 days**	**Fri 14.05.21**	**Thu 10.06.21**	**9**
14		**Mobile Application Programming**	**20 days**	**Fri 14.05.21**	**Thu 10.06.21**	
15		Java	20 days	Fri 14.05.21	Thu 10.06.21	
16		**Database Programming**	**10 days**	**Fri 14.05.21**	**Thu 27.05.21**	
17		SQL	10 days	Fri 14.05.21	Thu 27.05.21	
18		**SYSTEM TESTING**	**15 days**	**Mon 7.06.21**	Fri 25.06.21	
19		User Testing	15 days	**Mon 7.06.21**	Fri 25.06.21	

Gantt Chart:



7. MEASUREMENTS

7.1. Questions to identify measurements:

1. Has everyone done their duty?
2. Did the project adhere to its schedule?
3. Does the project meet the desired requirements?
4. Did the project require extra cost?
5. What is the size of the project?
6. Were the training days enought to learn how to use the tools?

7.2. Identified Measurements:

- Individual time-logs (Total hours spent on tasks and meetings by each member) (1)
- Desired deadline and number of days to finish each tasks (2)
- Customer Feedbacks (3)
- Number of customer-involved meetings (3)
- Number of defects (3)
- Amount of budget allocated before and spent after the project (4)
- Number of executable, declarative, blank and comment lines (5)
- Number of training days (6)

7.3. Measurement Storage and Collection:

- Total hours spent on tasks and meetings by each member

  What: Person hours spent

  When: After each iteration

  Format: Real number data

  How: Recording to a pre-prepared individual time-log spreadsheets

  Whom: By Beyza Özgür

- Desired deadline and number of days

  What: Total number of days left to complete for each task until desired deadline.

  When: Each week on Thursday at 21.00

  Format: Real number data

  How: Comparing the deadline date to task finish date recorded on a spreadsheet.

  Whom: By Birkan Sarıbacak

- Customer feedbacks

  What: Number of positive and negative customer feedbacks

  When: After the project is gone into use

  Format: Written sentence data

  How: Based on recording Google Play Store comments and feedbacks via email.

  Whom: By Bora Yörük

- Customer-involved meetings

  What: Number of customer-involved meetings

  When: After each customer-involved meetings

  Format: Real number data

  How: By recording the meetings to a prespecified spreadsheet

  Whom: By Kubilay Kalkan

- Defect Count

  What: Number of defects found in codes to determine inaccurate requirements and incomplete design & implementation.

  When: After every written code

  Format: Real number data

  How: By counting and recording a prespecified spreadsheet

  Whom: By Beyza Özgür and Bora Yörük

- Number of executable, declarative, blank and comment lines

  What: Number of code lines to measure the size of the project

  When: After coding and testing processes are done

  Format: Real number data

  How: By recording the number of lines to a spreadsheet

  Whom: By Birkan Sarıbacak

- Amount of budget allocated before and spent after the project

  What: Amount of budget allocated before the project and total amount spent after the project is finished.

  When: After a new tool or need has been purchased.

  Format: Real number data

  How: By recording our predetermined budget and total costs spent on tools that used in our project to a spreadsheet.

  Whom: By Kubilay Kalkan

- Number of training days

  What: Training days of learning tools for each member

  When: Determined working days

  Format: Real number data

  How: Everyone will report their study time and these times will be recorded to a specified spreadsheet

  Whom: By Birkan Sarıbacak, Beyza Özgür, Bora Yörük, Kubilay Kalkan

7.4. Measurement Types and Descriptions

1. **Effort Measurements:** Total hours spent on tasks and meetings by the team
   Example: Number of individual working hours (50 hours)

2. **Defect Measurements (Testing):** Finding the defects of the codes
   Example: Bugs and mistakes found in the codes (15 lines of unnecessary codes)

3. **Cost Management:** Keeping track of tool expenses
   Example: Money amount ($50)

4. **Product Quality Measurements:** Keeping track of customer feedbacks and the number of customer-involved meetings
   Meetings Example: 15 meetings
   Feedback example: 10 positive feedbacks

5. **Time Management:** The number of days that spent to learn how to use the new tools
   Example: 5 days

6. **Product Size Measurements:** The number of code and comment lines
   Example: 1000 lines of code

8. PROJECT RISKS

We specified 12 important risks for this project. These risks are ranked according to their likelihood and the impact to the project. Descriptions of the risks and their combined ranks are given in a table in 8.3.

8.1. LIKELIHOOD RANK OF THE RISKS

1. **Coding –** The project may contain unnecessary and unefficient code blocks because the team does not have any experience with mobile application coding.

2. **Timing –** Project may not be finished in the desired time.

3. **Testing –** If testing process is not done properly, users' first impression on the project may cause a negative impact and this may increase the number of users.

4. **Training –** Training process may be difficult and may take so much time.

5. **Requirements Inadequacy –** The requirements specified by customers and end-users may not be enough. New requirements that were not identified before may be needed.

6. **Hardware Adaptation –** It may not work on versions above Android 7.0 as desired.

7. **Failure –** The application may not achieve the desired success, so it may have a small number of users.

8. **Complex GUI Design –** Due to the complex user interfaces, people of all ages and groups may have difficulty using the application.

9. **Tools –** Chosen tools may not be enough in the implementation process.

10. **Misuse –** The application may be misused.

11. **Budget –** Our budget may not be enough to finish the whole project.

12. **Team Member Turnover –** Any one of the members may leave the team before project is deployed because of conflicts within the team or personal problems.

8.2. IMPACT RANK OF THE RISKS

1. **Timing –** Project may not be finished in the desired time.

2. **Failure –** The application may not achieve the desired success so it may have a small number of users.

3. **Team Member Turnover –** Any one of the members may leave the team before project is deployed because of conflicts within the team or personal problems.

4. **Coding –** The project may contain unnecessary and unefficient code blocks because the team does not have any experience with mobile application coding.

5. **Testing –** If testing process is not done properly, users' first impression on the project may cause a negative impact and this may increase the number of users.

6. **Requirements Inadequacy –** The requirements specified by customers and end-users may not be enough. New requirements that were not identified before may be needed.

7. **Misuse –** The application may be misused.

8. **Complex GUI Design –** Due to the complex user interfaces, people of all ages and groups may have difficulty using the application.

9. **Tools –** Chosen tools may not be enough in the implementation process.

10. **Hardware Adaptation –** It may not work on versions above Android 7.0 as desired.

11. **Training –** Training process may be difficult and may take so much time.

12. **Budget –** Our budget may not be enough to finish the whole project.

8.3. COMBINED RANK

LIKELIHOOD RANK	IMPACT RANK	COMBINED RANK	RISK DESCRIPTION
2	**1**	**3**	**Timing** – Project may not be finished in the desired time.
1	**4**	**5**	**Coding** – The project may contain unnecessary and unefficient code blocks because the team does not have any experience with mobile application coding.
3	**5**	**8**	**Testing** – If testing process is not done properly, users' first impression on the project may cause a negative impact and this may increase the number of users.
7	**2**	**9**	**Failure** – The application may not achieve the desired success, so it may have a small number of users.
5	**6**	**11**	**Requirements Inadequacy** – The requirements specified by customers and end-users may not be enough. New requirements that were not identified before may be needed.
12	**3**	**15**	**Team Member Turnover** – Any one of the members may leave the team before project is deployed because of conflicts within the team or personal problems.
4	**11**	**15**	**Training** – Training process may be difficult and may take so much time.
6	**10**	**16**	**Hardware Adaptation** – It may not work on versions above Android 7.0 as desired.

8	**8**	**16**	**Complex GUI Design** – Due to the complex user interfaces, people of all ages and groups may have difficulty using the application.
10	**7**	**17**	**Misuse** – The application may be misused.
9	**9**	**18**	**Tools –** Chosen tools may not be enough in the implementation process.
11	**12**	**23**	**Budget** – Our budget may not be enough to finish the whole project.

9. SOFTWARE TOOLS

Project Tasks Which Require Software Tool Support

1. Mobile Application Implementation
2. Database Design
3. Version Control & Management

9.1. SOFTWARE TOOLS FOR TASK 1: IMPLEMENTATION

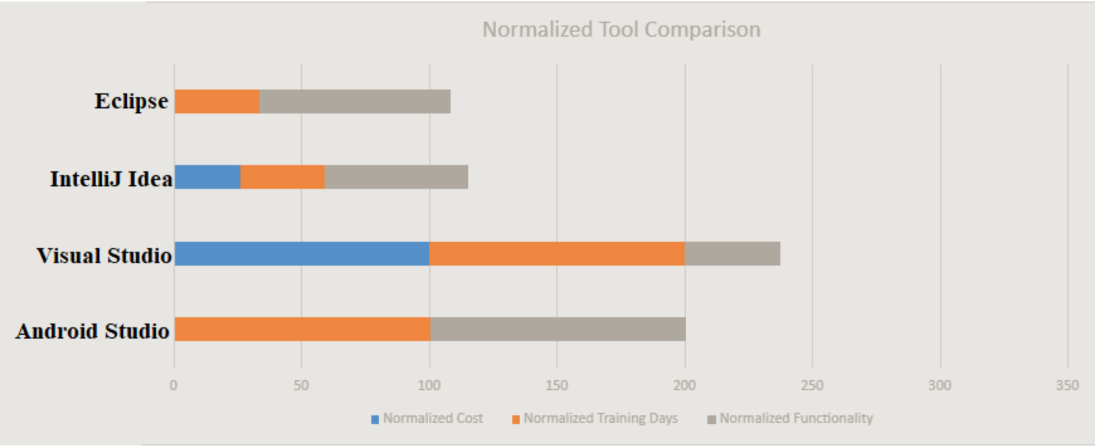
Tool Cost/Training/Functionality Data

Tool	Android Studio	Visual Studio	IntelliJ Idea	Eclipse
Cost	$1 (Free)	$250 (monthly) - $1 (Free)	$64.90 (monthly) - $1 (Free)	$1 (Free)
Training Days	15	15	5	5
Functionality	80	30	45	60

Normalized Cost/Training/Functionality Data

Tool	Android Studio	Visual Studio	IntelliJ Idea	Eclipse
Cost	0.4 (for $1)	100 (for $250)	26 (for $64.90)	0.4 (for $1)
Training Days	100	100	33.3	33.3
Functionality	100	37.5	56.25	75

Normalized Tool Graph



Which tool has been selected? Why?

Since we wanted to make an Android application, we chose Android Studio, thinking that it would be more useful to use the coding, testing and graphical user interface editing part of Android Studio. It is free and easy to install. It has good refactoring tools. Even though it will take longer to learn how to use this program, it has the best functionality. We will be able to do all the implementation, gui design and testing processes within only one tool. So, it actually saves time.

9.2. SOFTWARE TOOLS FOR TASK 2: DATABASE DESIGN

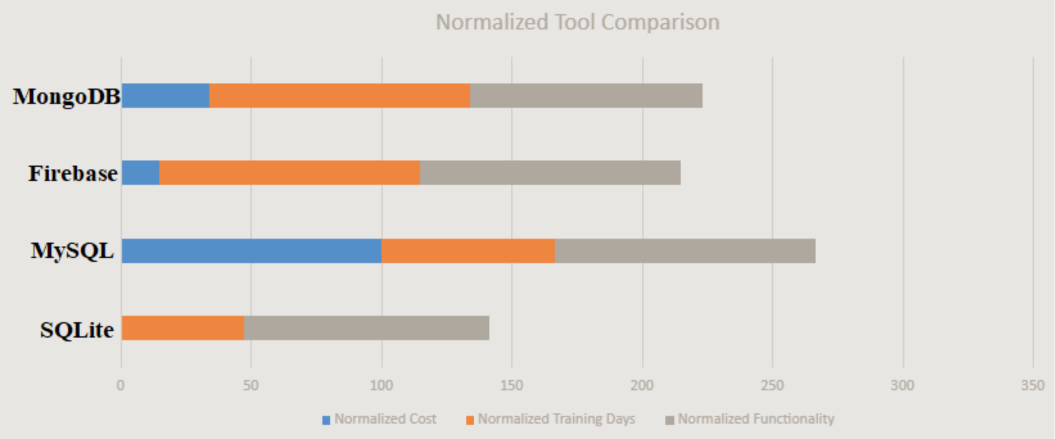
Tool Cost/Training/Functionality Data

Tool	SQLite	MySQL	Firebase	MongoDB
Cost	$2000 (one time fee) - $1 (Free)	$166.6 (monthly) - $1 (Free)	$24.99 (monthly) - $1 (Free)	$57 (monthly) - $1 (Free)
Training Days	7	10	15	15
Functionality	85	90	90	80

Normalized Cost/Training/Functionality Data

Tool	SQLite	MySQL	Firebase	MongoDB
Cost	0.6 (for $1)	100 (for $166.6)	15 (for $24.99)	34.2 (for $57)
Training Days	46.7	66.7	100	100
Functionality	94.4	100	100	88.9

Normalized Tool Graph



Which tool has been selected? Why?

We chose to use MySQL because it has the best functionality. For example, it is easy to use, it has a lot of database related features, its security feature is quite good, it is easily scalable and suitable for large databases, It provides good speed and performance, its user management is good, and it provides multiple access control. Its free version has all the features we need. Also, since we are currently learning MySQL at university, it will take less time to learn how to use it.

9.3. SOFTWARE TOOLS FOR TASK 3: VERSION CONTROL & MANAGEMENT

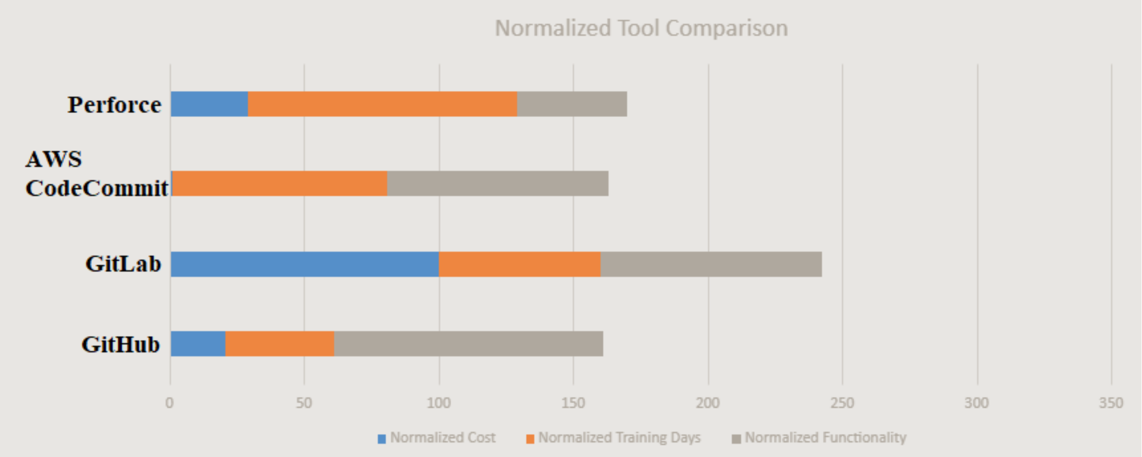
Tool Cost/Training/Functionality Data

Tool	GitHub	GitLab	AWS CodeCommit	Perforce
Cost	$21 (monthly) - $1 (Free)	$100 (monthly) - $1 (Free)	$1 (Free) - Pricing based on usage	$29 (monthly) - $1 (Free)
Training Days	10	15	20	25
Functionality	85	70	70	35

Normalized Cost/Training/Functionality Data

Tool	GitHub	GitLab	AWS CodeCommit	Perforce
Cost	21 (for $21)	100 (for $100)	1 (for $1)	29 (for $29)
Training Days	40	60	80	100
Functionality	100	82.3	82.3	41.1

Normalized Tool Graph



Which tool has been selected? Why?

GitHub is a website that allows programmers to work together on code. The first benefit is that it allows working together without damaging the project. Projects on GitHub are an example of open source software. It is easy to learn and its functionality is the best among others. Also, GitHub provides a student pack for free which has all the features of paid professional accounts. That's why we chose to use GitHub.

Note: *All of the prices were selected according to the monthly prices of the enterprice editions.*

10. PROJECT NEEDS

For this project we have three types of needs. These are software, hardware and support needs.

10.1. SOFTWARE NEEDS AND DESCRIPTIONS

1. **Windows 10 Operating System**

   Version: Windows 10-20H

   Installing & Upgrading by: Kubilay Kalkan (Tools Expert)

2. **Android Operating System**

   Versions: Android 7.0 and above

   Installing & Upgrading by: Kubilay Kalkan (Tools Expert)

3. **Android Studio**

   Version: Android Studio 4.1

   Installing & Upgrading by: Beyza Özgür (Coder), Kubilay Kalkan (Tools Expert)

4. **MySQL**

   Version: MySQL 8.0.24.0

   Installing & Upgrading by: Beyza Özgür (Coder), Kubilay Kalkan (Tools Expert)

5. **GitHub**

   Version: Latest version (2021)

   Since this is a web-based software, installation and upgrades are not required.

6. **Microsoft Office Programs**

   Version: Latest version (2019)

   Installing & Upgrading by: Kubilay Kalkan (Tools Expert)

7. **Discord**

   Version: Discord 11

   Installing & Upgrading by: Kubilay Kalkan (Tools Expert)

10.2. HARDWARE NEEDS AND DESCRIPTIONS

1. **Laptops**

   Functionality: Developing the mobile application, preparing all kinds of documentation and keeping in touch with team members.

   Necessary Softwares: Windows 10 OS, Android Studio 4.1, MySQL 8.0.24.0, Microsoft Office Programs, GitHub, Discord.

   Supporting Equipments: Charger, mouse, external monitor and keyboard.

   Will Be Used by: All of the team members (Beyza Özgür, Bora Yörük, Birkan Sarıbacak, Kubilay Kalkan)

   When: Every week from the beginning to the end of the project.

2. **Server**

   Functionality: Storing information of the users such as name, surname, email address and their posts etc.

   Supporting Equipments: Air conditioner (to cool down the server), necessary cables.

   Will Be Used by: Server Administrator

   When: Every week after the project is deployed and started being used.

3. **Android Mobil Devices**

   Functionality: Testing the program.

   Necessary Softwares: Android OS 7.0 and above.

   Supporting Equipments: Charger.

   Will Be Used by: Bora Yörük (Tester) and Beyza Özgür (Coder)

   When: During coding and testing processes, after each iteration has finished.

4. **Connection Cable**

Functionality: Connecting mobile devices to computers for testing

Will Be Used by: Bora Yörük (Tester) and Beyza Özgür (Coder)

When: During coding and testing processes, after each iteration has finished.

5. **Wireless Modem**

Functionality: Connecting to the Internet to use and install programs and holding team meetings remotely.

Will Be Used by: All of the team members (Beyza Özgür, Bora Yörük, Birkan Sarıbacak, Kubilay Kalkan)

When: Every week from the beginning to the end of the project.

6. **Printer**

Functionality: Printing speadsheets, meeting reports and required documentations after each sprint.

Supporting Equipments: Cartridge, necessary cables.

Will Be Used by: All of the team members (Beyza Özgür, Bora Yörük, Birkan Sarıbacak, Kubilay Kalkan)

When: Every week from the beginning to the end of the project.

10.3. SUPPORT NEEDS AND DESCRIPTIONS

1. **IT Team**

   When: Every week on Saturday 12.00.

   How: By checking hardware needs, fixing problems related hardware & software, and executing backup system.

2. **Server Administrator**

   When: Every week on Wednesday at 12.00 after the project is gone into use.

   How: By monitoring server activity, implementing new server structures.

3. **Cyber Security Expert**

   When: Every week on Friday 12.00 after the project is gone into use.

   How: By protecting our mobile application from cybersecurity risks, threats and vulnerabilities.

4. **Human Resources**

   When: Every week at working hours.

   How: By recruiting, interviewing hiring new team members and managers when needed.

5. **Accountant**

   When: Every week at working hours.

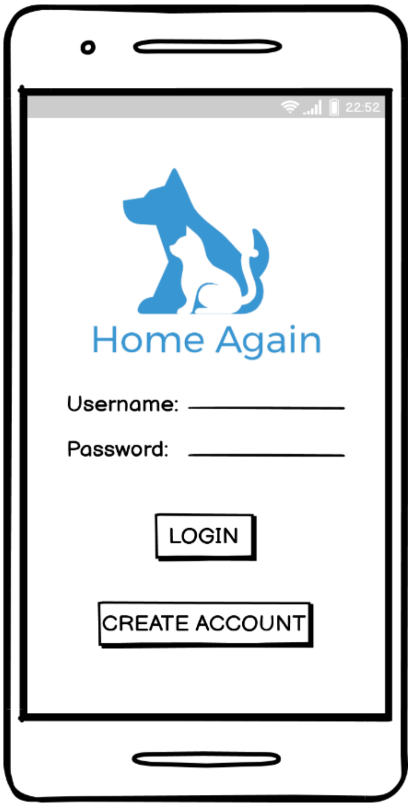
   How: By analyzing financial records and keeping track of the budget.

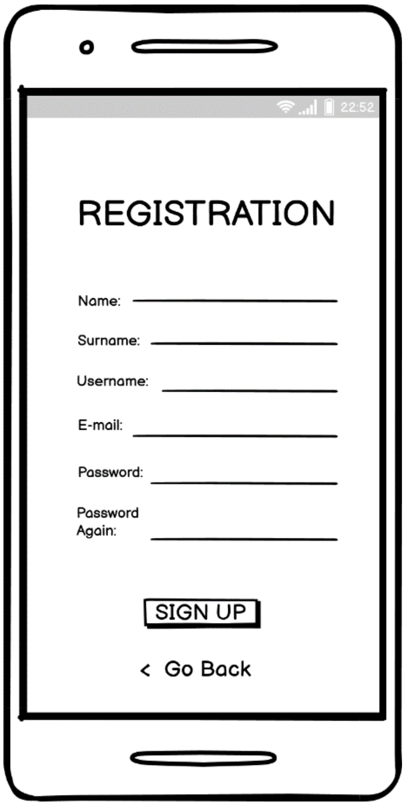
6. **Secretary**

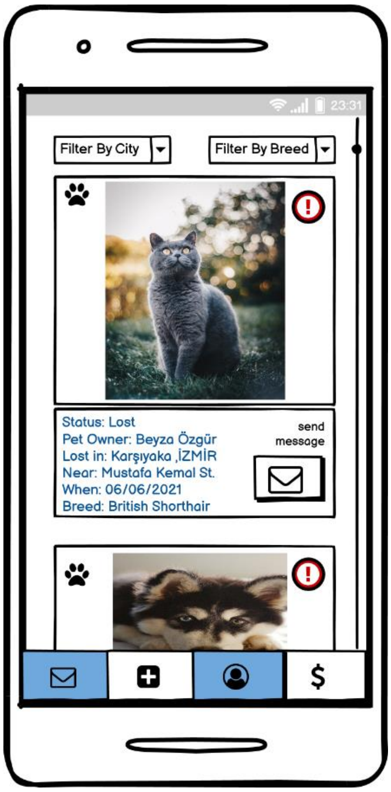
   When: Every week during meetings and at working hours.

   How: By keeping meeting reports and preparing necessary documents.

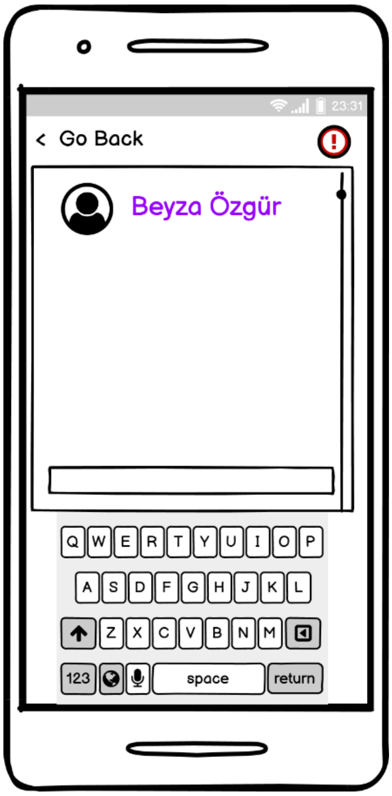
11. GRAPHICAL USER INTERFACES

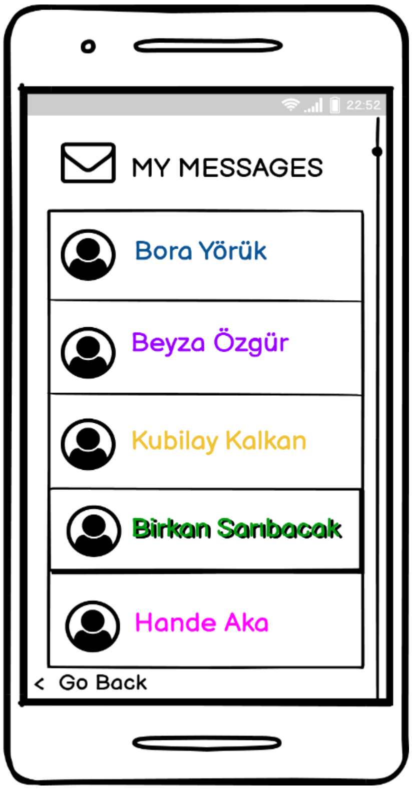






Status: Lost
Pet Owner: Beyza Özgür
Lost in: Karşıyaka ,İZMİR
Near: Mustafa Kemal St.
When: 06/06/2021
Breed: British Shorthair





MY MESSAGES

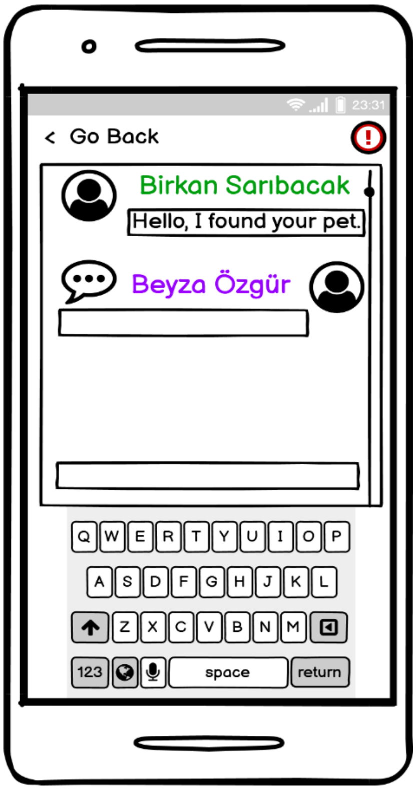
Bora Yörük

Beyza Özgür

Kubilay Kalkan

Birkan Sarıbacak

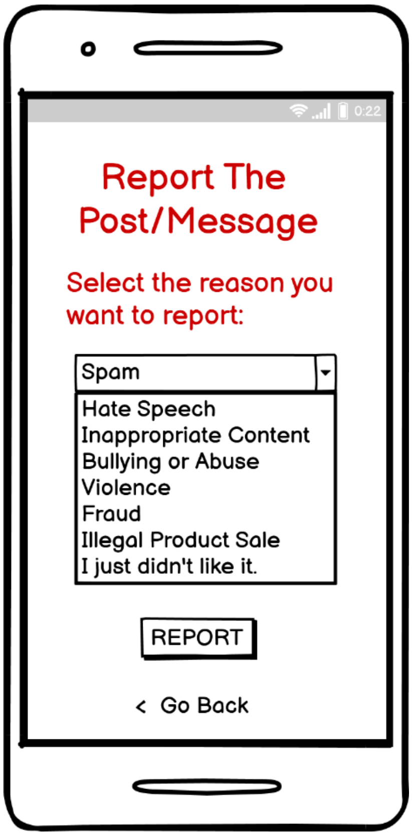
Hande Aka



Birkan Sarıbacak

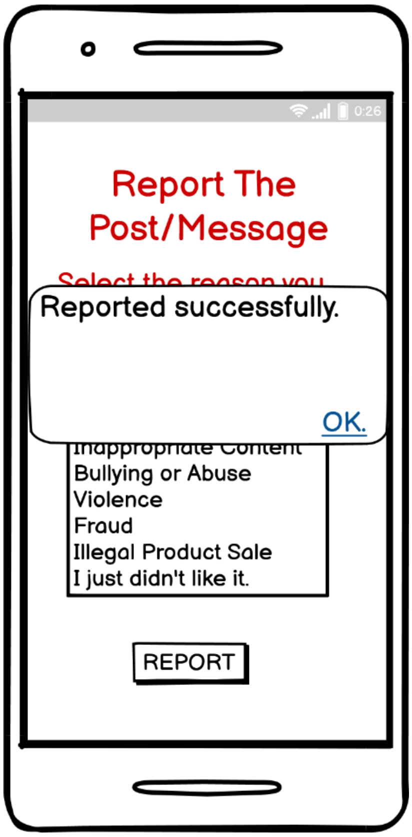
Hello, I found your pet.

Beyza Özgür



Report The Post/Message

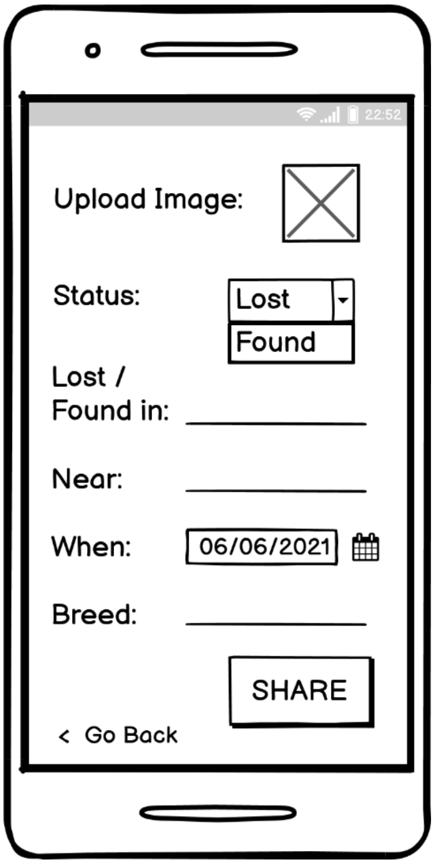
Select the reason you want to report:

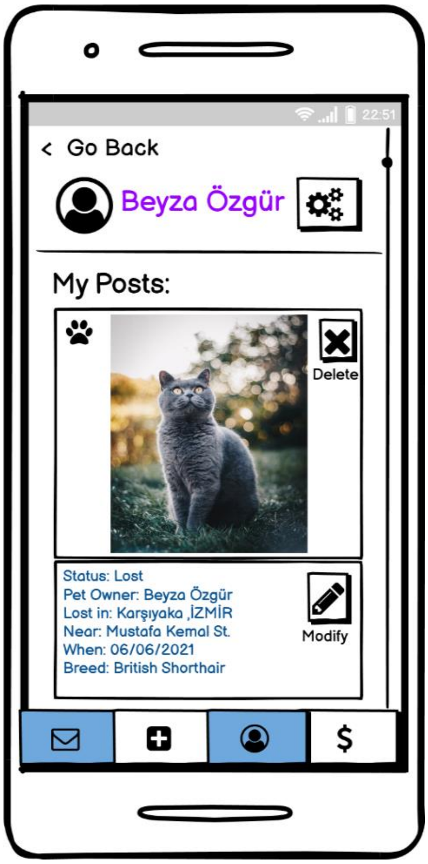


Report The Post/Message

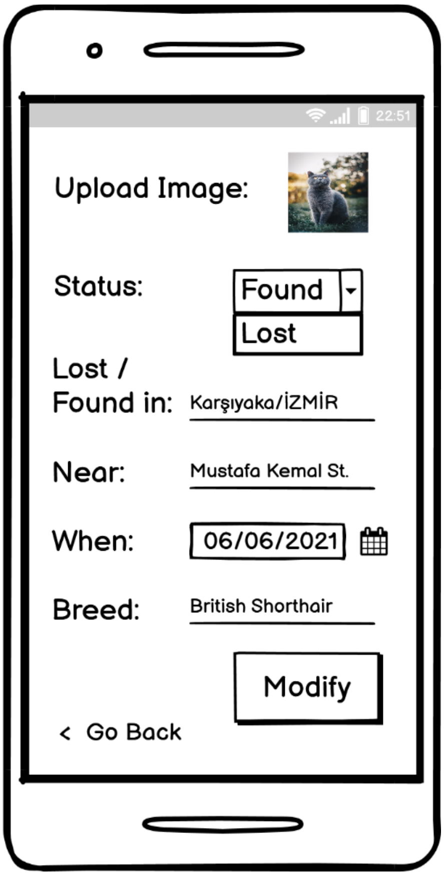
Select the reason you

Reported successfully.





Status: Lost
Pet Owner: Beyza Özgür
Lost in: Karşıyaka ,İZMİR
Near: Mustafa Kemal St.
When: 06/06/2021
Breed: British Shorthair



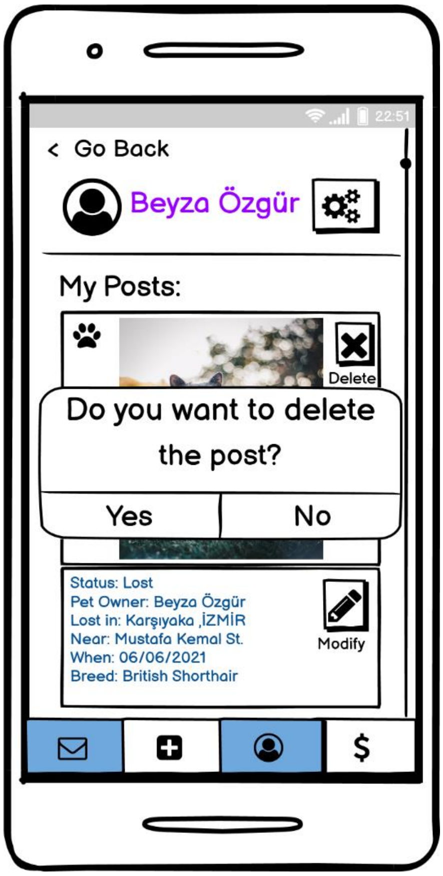
Status: Found

Lost / Found in: Karşıyaka/İZMİR

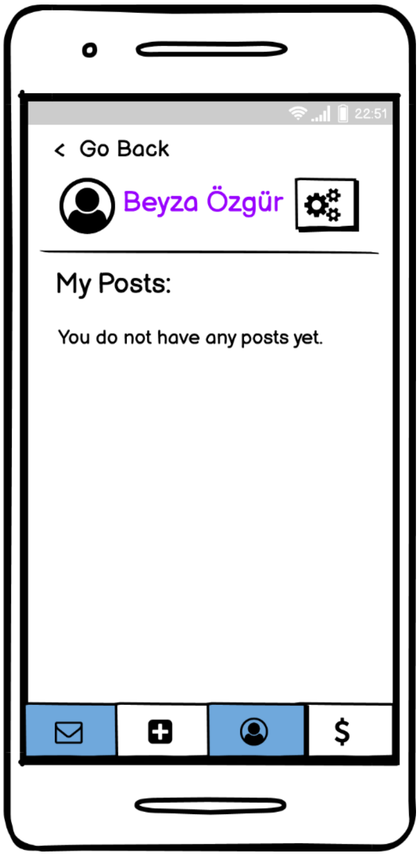
Near: Mustafa Kemal St.

When: 06/06/2021

Breed: British Shorthair



Status: Lost
Pet Owner: Beyza Özgür
Lost in: Karşıyaka ,İZMİR
Near: Mustafa Kemal St.
When: 06/06/2021
Breed: British Shorthair


22:51

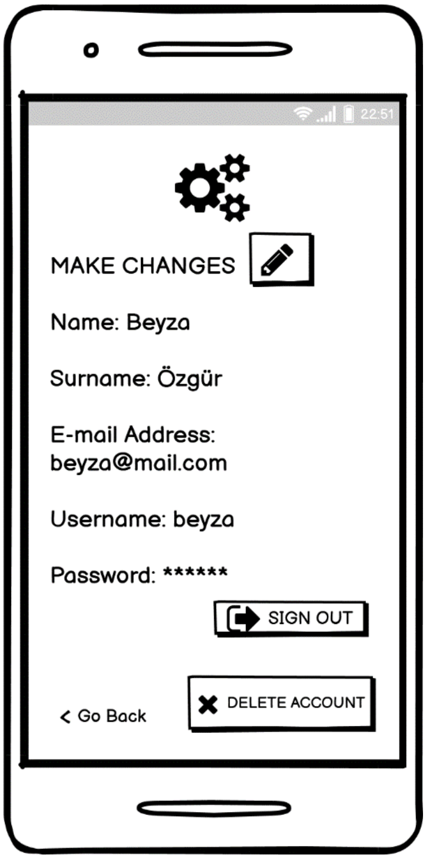
< Go Back

Beyza Özgür

My Posts:

You do not have any posts yet.

$


22:51

MAKE CHANGES

Name: Beyza

Surname: Özgür

E-mail Address: beyza@mail.com

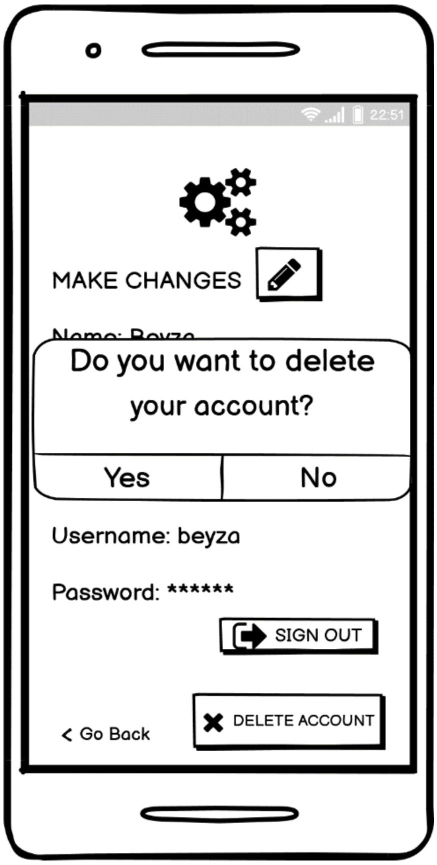
Username: beyza

Password: ******

SIGN OUT

< Go Back

DELETE ACCOUNT


22:51

MAKE CHANGES

Name: Beyza

Do you want to delete your account?

Yes | No

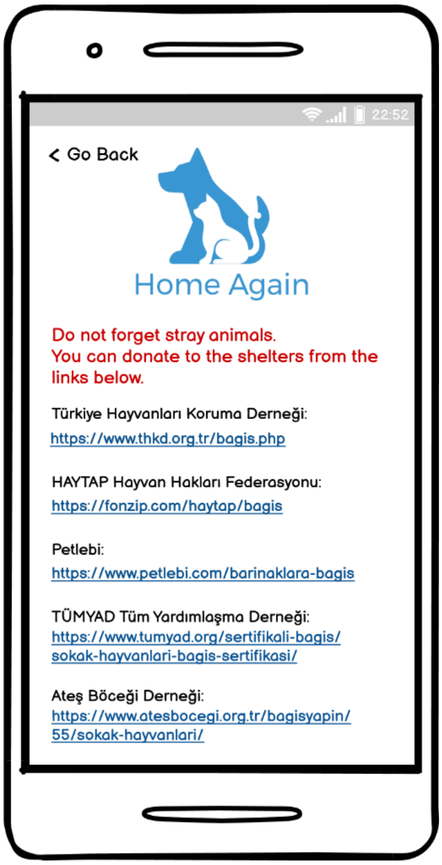
Username: beyza

Password: ******

SIGN OUT

< Go Back

DELETE ACCOUNT


22:52

< Go Back

Home Again

Do not forget stray animals.
You can donate to the shelters from the links below.

Türkiye Hayvanları Koruma Derneği:
https://www.thkd.org.tr/bagis.php

HAYTAP Hayvan Hakları Federasyonu:
https://fonzip.com/haytap/bagis

Petlebi:
https://www.petlebi.com/barinaklara-bagis

TÜMYAD Tüm Yardımlaşma Derneği:
https://www.tumyad.org/sertifikali-bagis/sokak-hayvanlari-bagis-sertifikasi/

Ateş Böceği Derneği:
https://www.atesbocegi.org.tr/bagisyapin/55/sokak-hayvanlari/

“Home Again” Graphical User Interfaces Designed By Group 6:

Beyza Özgür

Bora Yörük

Birkan Sarıbacak

Kubilay Kalkan

IMAGE CREDITS

Cat Photo by Kirsten Bühne from Pexels: https://www.pexels.com/photo/photo-of-british-shorthair-cat-sitting-on-grass-field-1521306/

Dog Image By:

Photo by Lucas Andrade from Pexels : https://www.pexels.com/photo/black-and-white-siberian-husky-4681107/

Home Again logo was designed by using Free Logo Maker: logomakr.com/2MzpBr

12. CONCLUSION

In conclusion, this document covers all of the necessary information to start the development of the project such as overview which is a summary of the plan, required high-level functionalities, stakeholders, staffing, software process model, schedule and effort shown using Gantt chart, measurements, risks, tools to be used during the project, needs, graphical user interfaces. If these are applied by the development team lead by the project manager, and the project manager regulary controls the tasks and if the team conforms to this plan, the goals of the development team will be accomplished in the most effective way.