CS491: Senior Design Project I
Analysis and Requirement Report

Group: T2443

yodai

Yasemin AKIN 22101782	Deniz Can ÖZDEMİR 22003854	Oğulcan KARAKOLLUKÇU 21802642	Alphan TULUKCU 22003500	İlhami ULUĞTÜRKKAN 22102546

Table of Contents

1. Introduction....4
2. Current System....4
3. Proposed System....5
    3.1 Overview....5
        3.1.1 User Interface Layer....5
        3.1.2 Integration Layer....6
        3.1.3 Application Services Layer....6
            3.1.3.1 Authentication and User Management....6
            3.1.3.2 Animal Profile Management....6
            3.1.3.3 Adoption Process Management....6
            3.1.3.4 Notification Service....6
            3.1.3.5 Chatbot Service....6
        3.1.4 AI and Machine Learning Layer....6
            3.1.4.1 Chatbot AI Module....6
            3.1.4.2 Matching Algorithm Module....7
            3.1.4.3 Image Recognition Module....7
        3.1.5 Data Management Layer....7
            3.1.5.1 Relational Database....7
            3.1.5.2 File Storage....7
        3.1.6 Security and Compliance Layer....7
        3.1.7 Monitoring and Maintenance Layer....7
        3.1.8 Infrastructure Layer....8
    3.2 Functional Requirements....8
        3.2.1 User Registration and Authentication....8
        3.2.2 Deep Learning-Based Visual Breed Classification....8
        3.2.3 Recommendation System for Sustainable and Accurate Adoptions....9
        3.2.4 Quiz System....9
        3.2.5 Chatbot Support (for Q/A)....9
        3.2.7 Notifications and Alerts....9
    3.3. Non-Functional Requirements....9
        3.3.1 Usability....9
        3.3.2 Reliability....10
        3.3.3 Performance....10
        3.3.4 Supportability....10
        3.3.5 Scalability....10
        3.3.6 Sustainability....10
    3.4 Pseudo Requirements....10
    3.5 System Models....11
        3.5.1 Scenarios....11
        3.5.2 Use Case Model....16
        3.5.3 Object and Class Model....17
        3.5.4 Dynamic Models....17
        3.5.5 User Interface Navigational Paths and Screen Mock-ups....17
4. Other Analysis Elements....21
    4.1. Consideration of Various Factors in Engineering Design....21
        4.1.1 Constraints....21

4.1.1.1 Implementation Constraints..... 21
4.1.1.1.1 Data Collection..... 21
4.1.1.1.2 Data Privacy..... 21
4.1.1.1.3 Skill Gaps..... 21
4.1.1.1.4 Technology Limitations..... 21
4.1.1.1.5 Time Constraints..... 22
4.1.1.1.6 Scalability and Maintenance Constraints..... 22
4.1.1.2 Economic Constraints..... 22
4.1.1.2.1 Budget for Resources..... 22
4.1.1.2.2 Adoption Motives..... 22
4.1.1.3 Ethical Constraints..... 22
4.1.1.3.1 Bias in Algorithms..... 22
4.1.1.3.2 Animal Welfare..... 23
4.1.1.3.3 Adopter Privacy..... 23
4.1.1.3.4 Fair Use of AI..... 23
4.1.1.3.5 Cultural Sensitivities..... 23
4.1.2 Standards..... 23
4.2. Risks and Alternatives..... 24
4.3. Project Plan..... 26
4.4. Ensuring Proper Teamwork..... 26
4.5. Ethics and Professional Responsibilities..... 27
4.5.1 Prioritizing Animal Welfare..... 27
4.5.2 Transparency and Fairness..... 27
4.5.3 Data Privacy and Security..... 27
4.5.4 Promotion of Ethical Awareness..... 27
4.6. Planning for New Knowledge and Learning Strategies..... 27
5. Glossary..... 28
6. References..... 30

1. Introduction

After the recent legislation in Turkey that mandates the removal of stray animals from public places and their placement in shelters, it is urgent to enhance the adoption process to avoid overcrowding and euthanasia risks for these animals. In July 2024, this law was passed to address public safety concerns, but it has proved to be a source of tremendous difficulty for animal welfare organizations [1]. Our goal is to build a new platform to simplify adoption. Our goal is to strengthen the bond between potential adopters and the animals in need by providing comprehensive animal profiles, including health records and behavioral traits, personalized care recommendations, and most importantly, a sophisticated personalized stray animal recommendation system. The platform will also provide adoption guidance and post-adoption support. Beyond providing a solution to the urgent problems created by the new law, this initiative also serves to support the long-term health of stray animals in Turkey through the promotion of sustainable adoptions and strained shelter resources. This report aims to provide more detailed information about the project's vision, mission, scope, and content by including detailed information about the project's specifications.

Assisted by Kurtaran Ev, a well-known Istanbul, Türkiye rescue shelter, this project aims to completely transform how animals are adopted by creating a revolutionary platform. The platform aims to solve long-standing inefficiencies in current systems by leveraging advanced technologies such as image recognition, artificial intelligence (AI), and automated matching algorithms. Challenges also include no streamlined matching processes and incomplete or inaccessible animal profiles. This platform is looking to introduce a more technological and automated approach to offer an effective and user-friendly solution for prospective owners and shelter animals alike.

The platform’s core is to simplify and optimize the process of animals matching potential adopters. It does this by creating in-depth and comprehensive animal profiles, including health records, vaccination histories, behavioral assessments, and personalized care recommendations. These features give the adopters a better understanding of their potential pets’ needs so they can make informed decisions and be better prepared to adopt a pet. A matching algorithm adds to this by ensuring adopters are matched to animals that fit their lifestyles, preferences, and capabilities, making for a more compatible and successful adoption.

Once past the initial adoption of the platform, the support features are robust enough to ensure the long-term success of pet ownership. This initiative is anchored around a sophisticated AI-driven chatbot that assists at all stages of adoption, pre-adoption, adoption, and post-adoption. This chatbot is able to answer queries, help guide adoption procedures, and provide ongoing support for training and veterinary care. By bringing together these pieces, the platform builds a supportive ecosystem for animals and their new families, facilitating a smooth transition and intense bonds.

The overall goal here is to drive meaningful change in adoption that has the potential to be sustainable and energizing. The platform reduces the risk of returned adoptions. It addresses gaps in existing systems, not only reducing the burden on overcrowded shelters but also advancing the larger mission of animal welfare. This initiative aims to change that future by taking an innovative approach to create a future where animals in rescue shelters can find loving, permanent homes and their new adopters are equipped with the tools and knowledge to properly care for them properly [2].

2. Current System

We will use ChatGPT, a cutting-edge Large Language Model (LLM), as the basis for our chatbot agent, PatiGPT, in order to improve user experience and accelerate adoption processes. OpenAI's ChatGPT is a highly popular and powerful conversational AI that is utilized in many different fields. Its strong natural language processing capabilities will

be used to tailor and enhance it for Kurtaran Ev. Through this integration, PatiGPT will be able to help users in a number of ways, including answering inquiries about shelter policies, adoption procedures, and animal care; offering post-adoption support with customized guidance, and guiding users through the platform and assisting them in making wise decisions throughout their adoption journey. With ChatGPT as the core, our PatiGPT agent will bring a personalized, efficient, and interactive experience to the platform, making the adoption process more accessible for users while reducing the workload for shelter staff. While "PatiGPT" relies on ChatGPT for its foundation, all other components of the platform, such as the recommendation system, breed classification, and user interface, are being developed entirely by the team without reliance on pre-existing systems.

3. Proposed System

Figure 1: High-level system design

3.1 Overview

The proposed system architecture for "Kurtaran-AI" integrates several key components to deliver a seamless, user-friendly, and efficient platform for adoption processes. The high-level architecture comprises the following layers and modules:

3.1.1 User Interface Layer

The client side will be a cross-platform application that supports web, IOS, and Android systems. React Native will be used as a framework to develop a cross-platform system. Moreover, Expo will be integrated with React Native to simplify the development process by providing tools, services, and libraries that enhance the app's performance and compatibility. Expo's pre-configured environment will reduce setup time and allow easier device testing.

3.1.2 Integration Layer

The Integration Layer handles the communication of the User Interface with Backend Services by using an API Gateway. This keeps data flowing safely and efficiently, enabling third-party integrations like Google Maps API for shelter location services.

3.1.3 Application Services Layer

This layer represents the core business logic of the system, coordinating functionalities stated below.

3.1.3.1 Authentication and User Management

This module provides a secure and efficient way of user registration, authentication, and access control. Advanced encryption techniques are instigated to protect sensitive user information and maintain trust. The component comes with role-based permissions for granting specific access to system resources based on user roles.

3.1.3.2 Animal Profile Management

This component provides detailed profiles of animals, including important information such as health records, behavioral traits, vaccination history, and breed details. The module supports live updating, allowing shelter staff to update profiles easily and thus ensuring adopters always have the latest and accurate information.

3.1.3.3 Adoption Process Management

With this module, the adoption process workflow will be automated: application submission, eligibility checks, approvals, and status tracking. Designed with the least possible manual intervention to ensure transparency and efficiency, this process will provide real-time notifications to both applicants and staff.

3.1.3.4 Notification Service

Delivers real-time alerts and reminders to users regarding their adoption milestones and post-adoption responsibilities.

3.1.3.5 Chatbot Service

Provides AI-powered conversational support, guiding users through processes and addressing their queries efficiently. The application services layer integrates directly with the data management and AI layers, enabling real-time processing and decision-making.

3.1.4 AI and Machine Learning Layer

This layer includes advanced modules that are mentioned below.

3.1.4.1 Chatbot AI Module

This module will help users throughout the adoption lifecycle, from pre-adoption to adoption and post-adoption. The chatbot will be able to answer quick queries, guide users through adoption processes, and provide continued support for training and veterinary care. It will use frameworks like LangChain and LangGraph, and will connect to OpenAI's large language models (LLMs) and interact with the application via an API gateway at the backend. Self-refinement and Retrieval Augmented Generation (RAG) techniques will be leveraged for the chatbot to ensure security and correctness, dynamically access and provide accurate information while maintaining data integrity and upholding privacy regulations.

3.1.4.2 Matching Algorithm Module

The user-animal matching system will be based on traditional ranking methods. We will rank animals according to user parameters, which are determined by Kurtaran Ev and other possible rescue shelter clients. Our training dataset for this model will contain previous adoption successes and failures. We will use libraries such as numpy and pandas for data manipulation and computing and sklearn or rank_bm25 to calculate various matching scores between animals and users. We are planning to calculate various scores with different weights and calculate the final matching scores.

3.1.4.3 Image Recognition Module

To identify and classify breeds of stray cats and dogs from their images, this module uses a pre-trained CNN deep learning model. It analyzes the uploaded photos, and based on that, it recognizes the breed information and uses that information to update animal profiles on the platform while also leveraging web-scraped data. By automating the breed identification process, the module increases efficiency for shelter staff and helps potential adopters know more about the breed.

3.1.5 Data Management Layer

The data management layer is responsible for storing, organizing, and retrieving all system data reliably. It consists of:

3.1.5.1 Relational Database

The database will be powered by PostgreSQL hosted on a cloud relational database system (RDS) of choice, it stores structured data such as user profiles, adoption records, and animal profiles, ensuring consistency and integrity.

3.1.5.2 File Storage

This component represents a cloud-based multimedia-content solution, like photographs and documents, that is supposed to be easily accessible for huge datasets of animal pictures. At this point in time, two alternatives—Firebase and AWS S3—are considered to be the storage solutions; the final decision about this will be taken concerning project requirements and trade-offs made between scalability, costs, and integration capabilities in the future. Data management uses optimized query designs in order to provide real-time retrieval for user-facing features like breed classification and generation of recommendations.

3.1.6 Security and Compliance Layer

The security and compliance layer will provide better protection for the data of users and systems in all operations. The management of security with advanced end-to-end encryption, secure key storage, and intrusion detection systems is designed to protect against unauthorized access and reduce the occurrence of breaches. Role-based access control (RBAC) limits sensitive operations to authorized personnel and ensures accountability to reduce security risks. There is also a compliance module that follows international standards such as GDPR, hence the transparency of data usage, while embedding audit trails of data access and modification. Integrated within the infrastructure and application services, this layer runs a continuous monitoring of data flows in order to enforce regulatory standards without impacting performance or user experience.

3.1.7 Monitoring and Maintenance Layer

This layer ensures platform reliability and health by setting up a resilient functionality in place. System monitoring shall be performed by Prometheus, covering performance metrics, anomaly detection, and peak system behavior in real time. The detailed records

of all activities within the system are stored with a full-service logging capability for easier debugging, auditing, and verification of compliance. Integrated with automated backup systems, scheduled and on-demand backups assure data integrity and support the whole process of disaster recovery. All these tools and processes mean the platform is scalable, resilient, and responsive to rising demand and unexpected challenges.

3.1.8 Infrastructure Layer

The infrastructure level might be the core of the system, created to offer sustainable and efficient resources. It may include cloud services such as AWS or a similar provider, which will provide high availability and elasticity, so that no performance is lost during periods of traffic and it can easily scale to include more data if needed. Load balancing may be used to balance the incoming user requests with the servers so as to avoid situations where one server is overloaded or to give the user the best experience. In addition, a content delivery network (CDN) is also suggested to speed up the delivery of content, especially by caching static resources such as images and videos; this should decrease the latency and enhance the performance for users from distinct geographical locations. This layer is envisaged to be used in maintaining the system in high performance and responsiveness; however, certain decisions made here may change with future conditions and restrictions.

The user interacts with the interface layer, where they have the possibility to view profiles of animals, apply for adoptions, or even use the AI assistant. The actions are then forwarded to the integration layer via the API Gateway, which will route the requests to the appropriate application services securely. The application services layer handles the requests from users by interacting with the data management layer. It retrieves or stores any important data, like animal profiles or user details, in the relational database or file storage used.

The data moves back and forth between the Application Services Layer and the AI and Machine Learning Layer to offer better features. For example, when a user uploads a picture of an animal, the image recognition module looks at it and updates the animal profile through the application services layer. Meanwhile, the matching algorithm module calculates the compatibility scores using both the information of users and animals by using saved data from the Data Management Layer. This information is also accessed by the chatbot AI module to provide real-time, context-aware responses.

Security and compliance systems ensure that data being sent between layers is encrypted and concurs with privacy rules. The monitoring and maintenance layer checks how the system is performing and keeps records of important activities.

The architecture and the connected flow of data creates a smooth experience for users, ensuring operations are efficient, accurate, and safe throughout the system.

3.2 Functional Requirements

3.2.1 User Registration and Authentication

Only adoption candidates will register for the system by themselves. They are asked to give some personal information such as their name, surname, age, address, etc., some of which will be crucial for adoption eligibility. Therefore, we want each user to verify the information they give. Also, we want them to authenticate their account with some methods.

3.2.2 Deep Learning-Based Visual Breed Classification

The aim is to implement a model that classifies the breed of dogs and cats from the given images. To achieve this, 3 different models will be trained on a specific dataset and compare their evaluation metrics to choose the one with the highest standards: one simple Convolutional Neural Network (CNN), one MobileNet CNN architecture will be

trained from scratch, and one pre-trained CNN-based (ResNet50 or VGG16) model leveraging transfer learning will be trained from scratch.

3.2.3 Recommendation System for Sustainable and Accurate Adoptions

This functionality is the central part of the KurtaranAI. We did market research about similar applications, and we realized that no system uses a recommendation system for available animals. Therefore, we will focus on developing an efficient recommendation algorithm so that adoption candidates will see the animals that are convenient for their and the animals' lives. To develop this algorithm, we will use the metrics that have been already used by rescue shelters manually. Moreover, we want to make this system dynamic so that the previous decisions of the adoption candidates will also affect the next recommendations. For this purpose, we plan to use both collaborative and content-based filtering to build a hybrid system. Leveraging user data such as preferences, behavioral data, and lifestyle information gathered from pre-requested forms and user activities and animal data such as breed, age, size, temperament, and special needs (like medical conditions), we plan to practice feature extraction, similarity measuring, matrix factorization, etc.

3.2.4 Quiz System

The aim is to understand which candidates are more eligible to adopt an animal. To understand this, we will give some mandatory quizzes prepared by professionals. Having solved all the quizzes, candidates will see the adoption portal if they are eligible for adoption. If not eligible, the plan is to direct them to a temporary adoption domain. We may use additional quizzes before the adoption process is done.

3.2.5 Chatbot Support (for Q/A)

The other crucial part of the KurtaranAI* is the chat assistant. Kurtaran Ev -our pilot shelter observed that there are numerous questions about animal care, health, training, etc. on their Instagram DM. To inform people and decrease the workload about messages, we will develop an LLM (Large Language Model) agent for question/answer for everyone. Moreover, some approved chats will be visible to everyone so that they can be used as an external information source. By feeding the LLM with user data, the end plan is to develop personalized veterinary assistance for users to accompany them through their adoption journey.

3.2.7 Notifications and Alerts

The aim is to notify users about the progress of their adoption processes. By nature of the process, some steps require manual approvals from the rescue shelter attendants, we plan to update users on these advancements as the process continues. Additionally, for the post-adoption process, since the new adopters will be under the surveillance of the rescue shelter attendants, they will be required to upload or send some evidence to show that the process is being carried out well. During this process, alerts can be set for users to remind them of these checkpoints.

3.3. Non-Functional Requirements

3.3.1 Usability

The platform should be easy to navigate for users with minimal technical skills since the target groups' ages of rescue shelters vary on a large scale. Also, both adopting a pet and using a mobile app can be complicated for most of the users. The app should inform and navigate users. Ensure clear visual cues (e.g., buttons, links, tooltips) to guide users through the adoption process.

3.3.2 Reliability

The most crucial parts of the project are ChatBot and the matching algorithm between animals and users. These features must be reliable and dynamic to satisfy both animals' and users' needs. The chatbot feature of the application must have a 90% or higher correct response rate based on our frequent questions and answers dataset obtained by Kurtaran Ev's forum. Also, we are planning to have an approximately 80% precision rate for retrieving relevant animals for users concerning our dataset which was obtained by Kurtaran Ev's adoption archive. Also, we attach great importance to both features' robustness and dynamicity.

3.3.3 Performance

The app will include a lot of videos and images. While retrieving them from the database, we must wait less than 3 seconds. Use lazy loading for images and animations to improve page speed.

3.3.4 Supportability

The app will be cross-platform since the target users' platform usage is distributed nearly equally. Also, since this project is a senior design project, detailed documentation of the implementation and the design of the application will be documented clearly. Finally, with the feedback system of the application, bugs and issues can be prioritized and solved.

3.3.5 Scalability

The project has the potential to be used by thousands of people, necessitating a platform that can scale seamlessly to support growing demands. It must efficiently handle simultaneous access from multiple users without performance degradation, ensuring smooth interactions such as concurrent searches, chats, and profile views.

3.3.6 Sustainability

In order to be cost-effective in terms of sustainability, we will choose servers and databases that will meet our needs but are as free or low-priced as possible. As time progresses, we plan to continue in this way until we need a server that can manage a larger network, and then transfer it to paid servers.

3.4 Pseudo Requirements

1. Text within the application should be concise and clear to minimize user confusion and ensure a seamless experience. Developers should ensure labels and instructions follow this rule consistently.

2. Internal test data for training and validating the recommendation algorithm should mimic real-world conditions as closely as possible, even if it is not part of the final deployment.

3. Error messages displayed to the user should be informative but non-technical, enabling users to understand and resolve common issues independently.

4. The system's architecture should allow for easy debugging and modification, prioritizing modularity and clear documentation for internal team use.

5. Navigation flows in the prototype should adhere to a logical sequence for quick testing and adjustments during development, even if these flows may change in the final design.

3.5 System Models

3.5.1 Scenarios

Scenario 1	**Register System**
Actors	New Adoption Candidates
Entry Conditions	New adoption candidates clicks the sign up button after the initial presentation
Exit Conditions	Successfully sign up Fails to sign up Exit to the sign up screen Flow of Events
Flow of Events	1. In the first page, the user enters the general information name, surname, age and gender; and clicks on the "next" button. 2. In the second page, the user enters the account information as phone, email, and username. 3. In the third page, the user enters the information about the home animal will live as location, home size, room amount and kind. 4. In the last page, the user enters some information about their jobs and income, and accepts terms and conditions and clicks the register button. 5. An authentication mail is sent to the user. User enters the received code to the related field of the authentication page. 6. If the entered code is correct, the user is directed to the home page. 7. If the entered code is wrong, the user tries again.

Table 1: Scenario 1-Register

Scenario 2	**Login System**
Actors	Users who have an account in the Kurtaran-AI application.
Entry Conditions	New adoption candidates clicks the login button after the initial presentation
Exit Conditions	User successfully signs in. User signing-in fails. User closes the application. User clicks the sign-up button.
Flow of Events	1. User enters his/her email and password 2. Based on the results: - **If the account information is valid**: They are directed to the home page. - **If the account information is invalid:** They are redirected to the login system.

Table 2: Scenario 2-Login

Scenario 3	**Quiz System**
Actors	Registered users of the Kurtaran-AI application.
Entry Conditions	User has completed registration.

Exit Conditions	User passes the quiz and gains access to the main portal. User fails the quiz and is redirected to the “geçici yuva” portal.
Flow of Events	1. After registration, the system prompts the user to take a mandatory quiz. 2. User answers the quiz questions. 3. Based on the results: - **If the user passes**: They are granted access to the main adoption portal. - **If the user fails**: They are redirected to the temporary adoption portal. 4. The system logs the user’s quiz status (passed/failed).

Table 3: Scenario 3-Quiz System

Scenario 4	**Adoption Portal Interaction**
Actors	Registered users of the Kurtaran-AI application who are eligible for adoption.
Entry Conditions	- User logs into the application and accesses the main portal. - Users navigate to the portal.
Exit Conditions	Users like an animal and add it to their "İstekler" list. User rejects an animal and moves to the next profile. User exits the portal.
Flow of Events	1. User views an animal profile with details (e.g., name, age, health condition). 2. User takes one of the following actions: - **Swipe Right**: Likes the animal and adds it to their "İstekler" list. - **Swipe Left**: Rejects the animal and moves to the next profile. - **Scroll Down**: Views additional information about the animal (e.g., activity level, adoption requirements). 3. The system loads the next profile. 4. User repeats the process or exits the portal.

Table 4: Scenario 4-Adoption Portal

Scenario 5	**Temporarily Adoption Portal Interaction**
Actors	Users of the Kurtaran-AI application who are not eligible for adoption.
Entry Conditions	- User logs into the application and accesses the main portal. - Users navigate to the portal.
Exit Conditions	- Users like an animal and add it to their "İstekler" list. - User rejects an animal and moves to the next profile. - User exits the portal.

Flow of Events	1. User views an animal profile with details (e.g., name, age, health status). 2. User takes one of the following actions: - **Swipe Right**: Likes the animal and adds it to their "İstekler" list. - **Swipe Left**: Rejects the animal and moves to the next profile. - **Scroll Down**: Views additional information about the animal (e.g., activity level, adoption requirements). 3. The system loads the next profile. 4. User repeats the process or exits the portal.

Table 5: Scenario 5-Temporarily Adoption Portal

Scenario 6	**"İstekler" System**
Actors	Registered users of the Kurtaran-AI application.
Entry Conditions	- User clicked the "İstekler" button. - The user has reached their selection limit (e.g., 5 animals).
Exit Conditions	- User exits the review page.
Flow of Events	1. User navigates to the review page to see animals they have liked. 2. User views the list of animals and their details. 3. User performs one of the following actions: - **Remove an animal**: Frees up a spot in the "İstekler" list. - **Submit the application**: Sends a request to adopt the selected animals. 4. If the application is submitted, the user waits for a response from the shelters. 5. User exits the page when done.

Table 6: Scenario 6-Review System

Scenario 7	**PatiGPT Chat System**
Actors	Registered users of the Kurtaran-AI application.
Entry Conditions	- User navigates to the PatiGPT chat section.
Exit Conditions	- User exits the chat.
Flow of Events	1. User opens the PatiGPT chat interface. 2. User types a question about animal care, health, training, or adoption processes. 3. The PatiGPT system responds with helpful information: - **If the system has the answer**: Provides detailed guidance. - **If the system cannot answer**: Displays a message like "Our LLM agent is still under development. Please try again later." 4. User continues the conversation or exits the chat section when done.

Table 7: Scenario 7-PatiGPT Chat System

Scenario 8	**Profile Management System**
Actors	Registered users of the Kurtaran-AI application.
Entry Conditions	- User navigates to the profile page from the navigation menu.
Exit Conditions	- Users update their profile information. - Users change their profile image. - User reviews their account status and exits the page.
Flow of Events	1. User accesses the profile page. 2. User sees the following sections: - **Profile Information**: Name, email, address, and other details. - **Profile Image**: Current profile picture with the option to upload a new one. - **Account Status**: Indicates account completion progress (e.g., "Completed," "2/3 Completed"). 3. User performs one or more actions: - **Edit Information**: Updates name, email, or other personal details. - **Change Profile Image**: Uploads a new image to replace the current one. - **Review Account Status**: Checks completion progress and identifies missing steps. 4. User saves changes or exits the page when done.

Table 8: Scenario 8-Profile Management System

Scenario 9	**Animals Adding System**
Actors	Shelter Volunteer, Shelter Manager
Entry Conditions	- Shelter volunteer/manager logs into the system and navigates to the "Add New Animal" section.
Exit Conditions	- Animals are successfully added to the system. - User cancels the process without adding the animal.
Flow of Events	1. Shelter volunteer/manager selects the "Add New Animal" option. 2. The user uploads one or more images of the animal. 3. The system's **breed classification model** processes the uploaded images and detects the breed. 4. The system auto-fills some animal details (e.g., breed, size, and possible age range) based on the classification and fetches relevant data. 5. User reviews the auto-filled information and performs the following actions: - Edit Information: Adjusts or adds details such as name, health status, temperament, location, and adoption requirements. 6. User submits the form to post the animal. 7. The system validates the submission and confirms the successful addition of the animal to the portal. 8. If the user cancels at any step, the process terminates without saving.

Table 9: Scenario 9-Animals Adding System

Scenario 10	**Managing Adoption Applications**
Actors	Shelter Volunteer, Shelter Manager
Entry Conditions	- Shelter volunteer/manager logs into the system and navigates to the "Applications" section.
Exit Conditions	- Application is approved, rejected, or marked for follow-up. - User exits the application management page.
Flow of Events	1. Shelter volunteer/manager navigates to the "Applications" section. 2. The system displays a list of current applications for animals, including the following details: - Animal name and profile. - User details (name, address, profile information, and quiz score). - Matching point: A calculated compatibility score between the user and the animal. 3. User selects an application to view additional details: - Compatibility metrics (e.g., lifestyle match, living environment, etc.). - Additional user information submitted during the application process. 4. User performs one of the following actions: - **Approve Application**: Marks the application as approved and notifies the user. - **Reject Application**: Marks the application as rejected and notifies the user. - **Follow-Up Needed**: Flags the application for further review or additional information. 5. System updates the status of the application and reflects it in the applicant's portal. 6. The user exits the page when done managing applications.

Table 10: Scenario 10-Managing Adoption Applications

Scenario 11	**Application Acceptance**
Actors	User (Adoption Candidate), Shelter Volunteer, Shelter Manager
Entry Conditions	- User's application is reviewed and approved by the shelter volunteer or manager.
Exit Conditions	- User receives the notification and views the acceptance details. - The user schedules a meeting with the shelter. - The user exits without immediately scheduling a meeting.

Flow of Events	1. The shelter volunteer/manager approves the application. 2. The system sends a notification to the user about the application status. 3. User logs into the portal and sees a congratulations message with the following details: - Animal details (name, breed, health status, location, etc.). - Shelter requirements (e.g., necessary documents, transportation arrangements). - Available meeting dates and times for visiting the shelter. 4. User performs one of the following actions: - Schedule a Meeting: Selects an available date and time to meet the animal at the shelter. - Review Requirements: Reads the shelter's guidelines and prepares accordingly. - Exit: Leaves the page and decides to schedule a meeting later. 5. If the user schedules a meeting, the system confirms the booking and sends a reminder notification. 6. User exits the portal when done.

Table 11: Scenario 11-Application Acceptance

3.5.2 Use Case Model

Figure 3: Use Case Diagram of the KurtaranAI

3.5.3 Object and Class Model

Figure 3: Object and Class Diagram of the KurtaranAI

3.5.4 Dynamic Models

In order to ensure the stability of the application, we decided to prepare our dynamic models in the later milestone. Therefore, the dynamic models of the KurtaranAI are not provided.

3.5.5 User Interface Navigational Paths and Screen Mock-ups

We designed and implemented several pages of the KurtaranAI. When the application is opened and no user has been authenticated yet, the initial pages in Figure 4 are shown which include some information about the application. On the last page, the user can click the login and register buttons.

Figure 4: Initial Screens of the Project

The user can click the “Aramıza Katıl” button to register. After filling out the registration parts of the registration process and verifying the account through email authentication, the user can start to use the application. The user can fill in the email and password and start using the application directly if they are already registered. There will be two more registration pages, which require the home and income information. On the last page, there will be a user agreement form. When the user accepts this and clicks the “Tamamla” button, a verification email will be sent to the user to verify the email. After that, the user can log in and see the profile page.

Figure 5: Login and Register Pages

Figure 6: Initial Profile, Quiz Initial, and Example Quiz Question Pages

After login, the user will see only the profile page if his/her profile is not completed. In order to complete the profile and access the other parts, the user must solve an assigned quiz that analyzes whether the user is eligible for adoption or not. Otherwise, the user can only view the profile page.

Figure 7: Congratulation Screen, Profile Page, and Setting Page

After passing the quiz, the user will be a candidate and have access to use Portal and PatiGPT pages. The user reaches each tab by clicking any button fixed at the bottom of the screen. Also, users can set the notifications and language in the setting page.

Figure 8: PatiGPT, Adoption Portal, and Review Page of the Project

In the Adoption Portal tab, the user swipes the animal profile to the right or to the left. If an animal profile is swiped to the left the user does not see the same animal again. If you swipe to the right, the animal is added to the "İstekler". Lastly, in the "İstekler" tab, the user can see the swiped right animals, and if click the "Başvuruyu Tamamla" button, the application forms are set and sent to the shelters. If a user swipes an animal on the list, the animal is removed from the list.

Figure 9: Application Forms, Approved Application, and PatiGPT Pages of the Project

Users can see the applications after approving the list in the "İstekler" tab. If the application is approved, there is a "Detayları Gör" button to see the details as a pop-up. In the details part, users can see the address, phone and email of the shelter, and there is a meeting set component to arrange a meeting. The details page will be improved

according to feedback by the shelters. In the PatiGPT tab, the user types messages and gets a response from the AI agent powered by ChatGPT.

These mockups have been developed on React Native, and each component has functionalities. These will be improved according to feedback by the UI/UX experts and the shelters. These are just the initial design of the project.

4. Other Analysis Elements

4.1 Consideration of Various Factors in Engineering Design

4.1.1 Constraints

4.1.1.1 Implementation Constraints

4.1.1.1.1 Data Collection

Data collection is one of our project's most difficult tasks. It is challenging to collect comprehensive and reliable data because many rescue shelters store animal information using outdated or disorganized systems. For instance, some shelters may categorize animals using completely different systems, while others may lack behavior assessments or health data. This implies that before our platform can use the data efficiently, we will need to spend time standardizing it.

4.1.1.1.2 Data Privacy

Even if there are digital records present, legal agreements with the establishments may be necessary to share data with the platform in order to guarantee compliance with national and international privacy rules. The implementation of secure protocols and encryption techniques is necessary to protect sensitive data during transmission and storage, which adds even more complexity.

4.1.1.1.3 Skill Gaps

Our team has experience in programming but creating a platform like this puts us in territories we are still learning and testing. Although we have a clear idea about what artificial intelligence (AI) and machine learning (ML) are, it is a very big step to move to creating and optimizing convolutional neural network (CNN) models for practical applications. Furthermore, creating an AI-supported chatbot based on large language models (LLMs) (which are comparatively unknown to us) entails additional challenges. Designing a cross-platform mobile application that will be scalable and fault-tolerant, designing a microservice architecture, and creating a machine learning-based recommendation system are challenging tasks. All of these elements combined accentuate the lack of specific knowledge and constantly encourage us to extend and develop our knowledge for the sake of the project. This steep learning curve could slow down our development process because it will take quite some time before we master the use of the software.

4.1.1.1.4 Technology Limitations

While it is our intention to use complex models such as CNNs for image recognition and machine/deep learning algorithms for matching, we are constrained by the computational resources which are available to us. These models are often complex to compute and demand large resources, which can hinder the performance of the platform, or even freeze it during operations in real-time. Balancing the complexity of these algorithms with the need for a smooth user experience is another major hurdle. On top of that, we are working within the limitations of our existing knowledge and tools, which means some compromises may have to be made to meet our project deadlines.

4.1.1.1.5 Time Constraints

We have limited time to implement this project. Approximately one academic year. We need to bring the project from the planning and research phase to the implementation, testing, and launching phases within this year. Every phase of the project is time-intensive, especially training and refining ML models. Since we have limited time, if any of these phases are disrupted, it will seriously affect the next phase and reduce its efficiency, so it is very important for us to try to do everything within the planned time.

4.1.1.1.6 Scalability and Maintenance Constraints

We are currently developing the platform to serve a single shelter or a small user base, but if it becomes successful, it will need to grow. This involves handling far more datasets, more adopters, and more shelters. A major difficulty is designing a system that meets our present requirements while allowing for future scalability. On top of that, the platform will require ongoing updates and maintenance, like adding features or integrating new technologies, which goes beyond what we can realistically achieve within the scope of this project.

4.1.1.2 Economic Constraints

4.1.1.2.1 Budget for Resources

Since we are operating under extremely limited resources, cost is an overriding consideration that has to inform all our choices. While hosting on cloud platforms like AWS and Google Cloud has good capabilities for hosting and processing data it can be expensive especially when dealing with large datasets or performing computational intense processes. We want to employ a great number of open-source and free tools to reduce costs. Although these tools are cost-saving, their capabilities and support can be quite reduced, so we should always take time and decide on the most efficient. Another important consideration is that training most machine learning models requires having access to specialized hardware like GPUs which we don’t possess. All these resources can be acquired through cloud services, but in cases where the training period is extended, the expenses concerning the related services may increase significantly. Further, real-time processes like image recognition or dynamic matching need to compute fast and accurately, which may prove to be expensive and add to our cost. Optimizing the costs and performance is going to be a strategic issue that will need a proper approach towards resource management.

4.1.1.2.2 Adoption Motives

Our platform would be more attractive if we offered adoption-promoting incentives, such as first veterinarian examinations or discounted pet supplies, but they are expensive. We probably will not be able to incorporate such features in our initial edition without collaborations or sponsorships. We will need to look for innovative, low-cost solutions to make the adoption process rewarding for users.

4.1.1.3 Ethical Constraints

4.1.1.3.1 Bias in Algorithms

Potential bias in our AI models is one issue that really worries us. Our matching technology may unfairly prioritize certain animals, making it more difficult for other animals to find homes, if the training data we utilize is skewed, for instance, favoring particular breeds or appearances. For animals that are already at a disadvantage due to disabilities or less "popular" characteristics, this is particularly troublesome. In order to prevent these biases from being reinforced, we will need to carefully assess and modify our algorithms.

4.1.1.3.2 Animal Welfare

Ensuring animal welfare is central to our project, but it's also one of the hardest things to guarantee. If the platform makes it too easy to adopt an animal without proper vetting, it could lead to impulsive decisions and, eventually, returned adoptions. This is not only stressful for the animals but also undermines our goal of creating lasting bonds between adopters and their pets. We will need to include features that educate adopters about the responsibilities of pet ownership and encourage thoughtful decisions.

4.1.1.3.3 Adopter Privacy

The other important ethical issue involves the privacy of the adopter. We will be gathering personal information, and data preferences, which makes the platform vulnerable to data breaches. Implementing strong encryption and secure storage methods is essential, but it also adds complexity to our development process. We also need to ensure transparency by giving users control over their data, like the ability to review or delete it.

4.1.1.3.4 Fair Use of AI

Our chatbot is designed to provide ongoing support to adopters, but relying too much on automation can lead to problems. For example, if the chatbot gives generic or incorrect advice about training or healthcare, it could frustrate users or even harm the animals. We will need to ensure the chatbot is well-trained and capable of escalating complex queries to real experts.

4.1.1.3.5 Cultural Sensitivities

Adoption practices and attitudes toward animals vary widely across cultures, and we need to account for this in our design. For instance, in some communities, animals might be valued more for their utility (like guarding) than companionship. While our platform should promote responsible adoption, it also needs to be flexible enough to accommodate these cultural differences without alienating users. This will require thoughtful language choices and inclusive design.

4.1.2 Standards

In this project, it is important to follow certain standards in order to attain the goals of reliability, scalability and ethical and technical acceptability. The implementation of our solution will be done in accordance with the international and regional standards of software development, data protection and artificial intelligence to make sure that the platform complies with the highest quality standards. In particular, we will try to implement an information security management system based on the ISO/IEC 27001 standard and guarantee the confidentiality of adopter and shelter's data during storage and transferring. Also, we want to follow IEEE 12207 standard for software lifecycle processes so that there is consistency within the processes that will be used throughout the development of the project. Regarding data protection, GDPR for the users' data will be followed so that the personal data is not misused and the process is more transparent.

One of the key distinctive features of our work will be the establishment of the Pet Adoption Benchmark, which will serve as a new evaluation and improvement model of animal adoption procedures. This benchmark will then be useful for the shelters and organizations around the world in the assessment of their operations. By adhering to the above standards, we will aim to create a creative and responsible platform to meet the challenges in animal adoption.

4.2 Risks and Alternatives

Risk Category	Political Risk
The Risk	Legisl regulations might impose stricter regulations on the use of AI in animal adoption processes.
Impact	Medium
Mitigation Plan	Monitoring legal regulations closely and collaborating with legal advisors to ensure compliance with policies.

Table 12: Risk 1

Risk Category	Political Risk
The Risk	Government-imposed restrictions or delays in the approval of adoption applications due to evolving policies regarding animal welfare or cross-regional adoptions.
Impact	Medium
Mitigation Plan	Engage with local authorities and policymakers to stay updated on regulatory changes. Build a flexible adoption workflow that can adapt to varying requirements across regions. Establish partnerships with animal welfare organizations to ensure alignment with current regulations and streamline approvals.

Table 13: Risk 2

Risk Category	Economic Risk
The Risk	High costs associated with LLM usage and microservices infrastructure.
Impact	High
Mitigation Plan	Use of open-source alternatives and optimization of resource usage to reduce operational costs. Establishing sponsorships or partnerships to offset expenses.

Table 14: Risk 3

Risk Category	Insufficient Data Risk
The Risk	Lack of sufficient and reliable data for training the recommendation system. The data that institutions say they will share with us may be non-existent or

	incomplete.
Impact	High
Mitigation Plan	We have partnered with multiple shelters for diverse datasets and will apply data augmentation techniques to enhance training data. Also, web-scraped data will be utilized.

Table 15: Risk 4

Risk Category	Performance Risk
The Risk	System latency or failures under high user loads.
Impact	Medium
Mitigation Plan	Use of scalable infrastructure like AWS or Firebase for adaptive resource allocation.

Table 16: Risk 5

Risk Category	Security Risk
The Risk	Breach of user data, including personal and adoption-related information.
Impact	High
Mitigation Plan	We will implement robust encryption, secure APIs, and multi-factor authentication to safeguard sensitive data.

Table 17: Risk 6

Risk Category	AI Bias Risks
The Risk	Recommendation algorithms favoring certain breeds or characteristics, leading to unfair adoption practices.
Impact	Medium
Mitigation Plan	We will regularly evaluate and update AI models to minimize bias, ensuring equitable recommendations.

Table 18: Risk 7

Risk Category	Integration Risks
The Risk	Compatibility issues between different modules (chatbot, recommendation system, etc.).
Impact	Medium

Mitigation Plan	We will define clear integration protocols and conduct end-to-end testing during development.

Table 19: Risk 8

Risk Category	Adoption Rate Risk
The Risk	Low user engagement due to insufficient platform awareness or trust.
Impact	Medium
Mitigation Plan	Planning to launch targeted marketing campaigns and build credibility through partnerships with reputable shelters.

Table 20: Risk 9

4.3 Project Plan

Figure 10: Project Gantt Chart

4.4 Ensuring Proper Teamwork

In order to ensure proper teamwork, our team first decided on each feature's importance and complexity. After considering both technical and non-technical responsibilities, we decided to split the technical part immediately and assign a group member as a project manager to split the non-technical responsibilities among the group members. Non-technical responsibilities such as searching for sponsors or arranging meetings with potential partners are crucial for our project's success. For that, our assigned project manager Alphan Tulukcu, created a Notion Project Management page and as a group, we shared all the work as a team. From a technical perspective, we shared all the features depending on the internships that we made in the past and the courses we took. Since Deniz Can Özdemir and Yasemin Akın worked with LLMs and AI during their internships, they are responsible for our chatbot feature. İlhami Uluğtürkkan and Oğulcan Karakollukçu are taking information retrieval systems and machine learning classes so they are responsible for user-animal matching algorithms lastly, because of

Alphan Tulukcu's past internship and project experiences he is the project manager of the group and responsible for the architecture of the application. We also use GitHub in an organized manner to be able to work systematically, dynamically, and in parallel to each other in a well-arranged way.

4.5 Ethics and Professional Responsibilities

4.5.1 Prioritizing Animal Welfare

The welfare of the animals to be adopted from the application is the top priority. In order to save as many animals as possible, the needs of the animals will be taken into consideration and homes where they can be happy will be matched. This matching system will show users who want to adopt animals the animals that can live happily and peacefully with them. They will not be able to see animals that are not suitable for adoption. The user interface is specifically designed to ensure that as many animals as possible can find a happy home. For this purpose, consultancy is received from Berkay Taşkın and Dilay Sarpay who are UI/UX experts.

4.5.2 Transparency and Fairness

It is clearly stated to users that this application has been developed on a completely voluntary basis and that no fee will be charged to users under any circumstances. It will be clearly explained why the application wants users' data and for what purposes they will be used.

4.5.3 Data Privacy and Security

In order to establish as good a matching algorithm as possible, users will be asked to provide detailed information about themselves. For this reason, users' personal data must be protected. Back-end and database systems to be used for this purpose are being investigated.

4.5.4 Promotion of Ethical Awareness

Dogs and cats trying to survive on the street on their own cause many problems for both themselves and the people around them. It is very difficult for them to find food, drink and shelter on their own. For this reason, efforts should be made to adopt as many stray animals as possible. By sharing the ease of use and importance of adoption on the social media accounts of the shelters that will adopt animals through the application, as many people as possible will be informed about this project. It is aimed to solve the stray animal problem as much as possible by ensuring that animals on the streets find a happy home.

4.6 Planning for New Knowledge and Learning Strategies

Right now, our team is in touch with Etimesgut Municipality and Kurtaran Ev to learn the current status of the rescue shelters, the adoption process, and the target audience. Since the target audience of Kurtaran Ev and Etimesgut Municipality are very different, our team is in touch with both sides to understand and satisfy a large range of users and reach as many people as we can. For this, we are also in touch with Berkay Taşkın and Dilay Sarpay who are working as UI/UX experts. Our team takes their opinion based on the requests of partners and thinks of a solution for every complexity. In technical perspective, our team is in touch with Yağız Küçükkambak and Deniz Dalkılıç, experienced devops consultant and senior software engineer, in order to discuss our tech-stack and possible implementation methods. Our team also takes their opinion on areas such as maintenance, performance, stability of the application since our team has limited experience in these areas.

5. Glossary

API Gateway: A server that acts as an API front-end, handling client requests, routing them to appropriate backend services, and managing tasks such as authentication, rate limiting, and analytics [3].

Artificial Intelligence (AI): The simulation of human intelligence processes by machines, especially computer systems, enabling them to perform tasks that typically require human intelligence, such as visual perception, speech recognition, decision-making, and language translation [3].

Chatbot: A software application designed to simulate human conversation, allowing users to interact with digital devices as if they were communicating with a real person [4].

Convolutional Neural Network (CNN): A class of deep neural networks, most commonly applied to analyzing visual imagery, that uses a mathematical operation called convolution to process data in a hierarchical manner [5].

Cross-Platform Application: Software designed to operate on multiple computing platforms, such as Windows, macOS, Linux, iOS, and Android, without requiring separate codebases for each [3].

Data Privacy: The practice of handling and processing data in a manner that ensures the confidentiality and protection of personal information from unauthorized access or disclosure [3].

Deep Learning: A subset of machine learning involving neural networks with many layers (deep neural networks) that can learn and make intelligent decisions on their own by analyzing large amounts of data [3].

Firebase: A platform developed by Google for creating mobile and web applications, offering services such as real-time databases, authentication, and hosting [4].

Front-End Application: The part of a software application that interacts directly with the user, encompassing the user interface and user experience aspects [3].

LangChain: A framework for developing applications powered by language models, facilitating the integration of large language models with external data sources and APIs [4].

LangGraph: A tool or framework designed to enhance the capabilities of language models by structuring and managing their interactions with various data sources and processes [4].

Large Language Model (LLM): A type of artificial intelligence model trained on vast amounts of text data to understand and generate human-like language [6].

Machine Learning (ML): A branch of artificial intelligence that involves the development of algorithms and statistical models enabling computers to perform tasks without explicit instructions, relying on patterns and inference instead [3].

PostgreSQL: An open-source, object-relational database system known for its robustness, extensibility, and standards compliance [3].

Prometheus: An open-source systems monitoring and alerting toolkit, particularly suited for monitoring dynamic cloud environments [4].

React Native: An open-source framework developed by Facebook for building native mobile applications using JavaScript and React [4].

Recommendation System: Software that provides suggestions to users based on preferences, behavior, or past interactions. It employs collaborative or content-based filtering, or a combination of both, to generate personalized recommendations [7].

Retrieval Augmented Generation (RAG): A technique that combines retrieval-based and generation-based methods in natural language processing to produce more accurate and contextually relevant responses [8].

Role-Based Access Control (RBAC): A security method where access to system resources is granted based on the user's roles, ensuring users can only perform actions relevant to their responsibilities [9].

Transfer Learning: A machine learning technique where a model developed for a particular task is reused as the starting point for a model on a second task, leveraging prior knowledge to improve learning efficiency [3].

6. References

[1] “Turkey Approves Law to Remove Stray Dogs from Streets. Opposition Vows to Fight the ‘Massacre Law’,” AP News. [Online]. Available: https://apnews.com/article/turkish-parliament-approves-law-stray-dogs-a5870dbff7066f0d11f34758b39bcf5f?utm_source=chatgpt.com. [Accessed: November 20, 2024].

[2] "YOD-AI Documentation," YOD-AI. [Online]. Available: https://yod-ai.github.io/yodai/#. [Accessed: November 19, 2024].

[3] IBM, [Online]. Available: https://www.ibm.com. [Accessed: November 21, 2024].

[4] freeCodeCamp, “Learn to Code — For Free.” [Online]. Available: https://www.freecodecamp.org/. [Accessed: November 19, 2024].

[5] LeCun et al., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[6] T. B. Brown et al., "Language Models are Few-Shot Learners," in *NeurIPS*, 2020.

[7] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," *Advances in Artificial Intelligence*, vol. 2009, pp. 1–19, 2009.

[8] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *NeurIPS*, 2020.

[9] R. S. Sandhu et al., "Role-Based Access Control Models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, 1996.