|  |
| --- |
| **PROJECT PLANNING & MANAGEMENT FORM**  **CMSE 201**  **GROUP NO: 12**  **PROJECT NAME: Stray Animals Tracking Mobile App**  **PROJECT START DATE: 11/10/2023.**  **PROJECT END DATE: 24/12/2023**  **SUPERVISOR: Duygu Celik Ertugrul**  **SEMESTER TERM: Fall 2023/2024** |

A.1. Preliminary Project Information

# A.1.1

|  |  |
| --- | --- |
| **Project No** | 2 |
| **Project Name** | Stray Animals tracking mobile app |
| **Start Date** | 11/10/2023 |
| **End Date** | 24/12/2023 |
| **Time** | 2 months and 13 days |

# A.1.2

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Manager** | | | |
| **Name Surname** | OMAR AHMED | **ID No** | 20801041 |
| **Title/Role** | Project Manager | | |
| **Address** | Famagusta | | |
| **Phone** | +905338399207 | | |
| **Email** | omar.alyousufi10@gmail.com | | |

A.2 Group Information

# A.2.1

|  |  |  |  |
| --- | --- | --- | --- |
| **Student 1** | | | |
| **Name Surname** | AHMED SALMI | **ID No** | 22703233 |
| **Title/Role** | Database developer | | |
| **Address** | Famagusta | | |
| **Phone** | +9718568311359 | | |
| **Email** | Ahmedsu2004@gmail.com | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student 2** | | | |
| **Name Surname** | AHMED HABBADI | **ID No** | 21009123 |
| **Title/Role** | UI Designer | | |
| **Address** | Famagusta | | |
| **Phone** | +213672253826 | | |
| **Email** | Aminehabbadi7@gmail.com | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student 3** | | | |
| **Name Surname** | Zeynep Pelin Çolak | **ID No** | 17300009 |
| **Title/Role** | Tester | | |
| **Address** | Famagusta | | |
| **Phone** | +905338721202 | | |
| **Email** | colakpelin8@gmail.com | | |

# A.2.2

|  |
| --- |
| **List of Completed / Ongoing Projects of Team** |
| Cmse201- Introduction to Software Engineering |

B.1 Introduction to Project

# B.1.1

|  |
| --- |
| **Summary of Project** |
| To improve the welfare of stray dogs and cats, the “Rescue and Rehabilitation of Stray Animals” project works to remove them from the streets, give them necessary veterinary care, encourage responsible ownership through adoption, increase public awareness, and promote humane treatment. The project aims to address the various difficulties faced by stray animals and foster a culture of compassion and care, with a particular emphasis on community involvement and sustainable finance. In the end, the initiative hopes to provide these vulnerable animals with a chance at a happier and healthier life. |

# B.1.2

|  |
| --- |
| **Key Words** |
| Pets, Adopted, Humanity, Stray Animals, Application, Website, HTML, pHp, CSS, Financial support, Food Aid, Opportunities, Street, Abandoned, Programming, illness. |

# B.1.3

|  |
| --- |
| **Aim of Project** |
| The goal of this project is to develop a mobile/web application to provide foster care/sponsorship for stray animals (dogs and cats). The software will link stray animals to appropriate authorities and organizations. In other words, the purpose of this project is to help stray animals that do not have a home or owner. |

# B.1.4

|  |
| --- |
| **Innovative Aspects/Contributions of Project** |
| * Providing a mobile/web application management of a foster care/sponsorship for stray animals * Offering an administrative panel for effective operation of the foster care/sponsorship program * Managing the connection of authorities through the application and identifying the needs of stray animals * Including donations makes it easier to financially support the program. * Making scrum management available to enable efficient and successful project management. * Making advantage of modern design trends to provide users with an easy-to-use engaging experience. * Overall, this project provides an innovative an easy approach to help and enhance lives of stray animals |

# B.1.5

|  |
| --- |
| **Methods to be Applied** |
| * Design stage: the application’s design will be based on current design trends and user-centered design principles, ensuring that users have an easy-to-use and interesting experience. * Agile development: agile development concepts will be applied to guarantee that the application is created iteratively and in a flexible way, allowing us to respond to changing requirements and use demands. * Continuous testing: testing will be carried out continuously during the development cycle to guarantee that the application has been thoroughly examined and is free of errors and other problems prior to release.. * To guarantee that new features and upgrades are provided promptly and efficiently, continuous integration and deployment process will be used. |

# B.1.6

|  |
| --- |
| **Economic and National Outcomes** |
| The economic advantages of assisting stary animals are numerous. While there are costs connected with reaccusing and caring for these animals the potential advantages go beyond animal welfare. Communities that engage in the welfare of stray animals may benefit economically, from grater tourism to lower public health costs and higher social capital. Policymakers' communities, and individuals must evaluate the economic ramifications of their efforts to aid stray animals, since this understanding may help assure the long-term viability of such projects. finally, assisting stray animals is not only a moral duty but also a means of strengthening and prospering communities. |

B.2 Reason of Starting the Project, Methods and R&D Stages

# B.2.1

|  |
| --- |
| **1- Explain the reason of starting this project. (Max 500 character)** |
| Concern about the terrible circumstances stray dogs and cats suffer from led to the creation of the "Stray Animals Tracking Mobile App" project. These animals suffer from malnourishment, illness, and famine, among other things. To put an end to their suffering and build a society that is empathetic is the project's main goal. In addition to promoting responsible adoption, it seeks to use technology for effective monitoring and rescue operations. Through the promotion of an empathetic and respectful culture, public awareness campaigns aim to alter social perceptions around stray animals. The project aims to create a future in which animals and humans live in harmony, giving these defenseless species a chance at a happier and healthier existence. |

|  |
| --- |
| **2- Explain the purpose of this project.** |
| The "Stray Animals Tracking Mobile App" project's main purpose is to lessen the extreme pain that stray dogs and cats go through since they are forced to deal with untreated illnesses, malnourishment, and the harsh reality of life on the streets. The aim is to initiate a shift in our society in the direction of one that is empathetic and compassionate toward all sentient beings, including stray animals. Through encouraging responsible adoption, enhancing rescue and monitoring operations with technology, and running public awareness campaigns to transform people's perceptions of stray animals, this project acts as a catalyst for change. Its main goal is to bring about a time when people and animals can live in harmony, giving these defenseless animals the chance to live happier and healthier lives in a society characterized by compassion and decency. |

|  |
| --- |
| **3- Explain**   * **output of project** * **national / international standards if exist.** * **the specific objectives of the project** * **success criteria’s** * **realistic constraints** |
| Output: A smartphone app for safely tracking and reporting stray animals.    Standards: Complying with international ethical guidelines.    Objectives: Create an app that is easy to use, interacts with communities, and encourages moral behavior.    Success criteria: Ethical behavior, community involvement, and effective tracking.    Realistic Constraints: Resources available, budget, behavioral shift, and technical difficulties. |
| **4- Explain**   * **the methods to be applied during R&D activities.** * **applications** * **technics and tools to be used.** * **standards to be followed under the workflow** |
| **Which SOFTWARE PROCESS MODEL below will you apply? Why? How? Explain.**  **\* The waterfall models.**  **\*V-model of software process?**  **\*Evolutionary development?**  **\*Component-based software engineering? Etc.**  The best software process model for projects with well-defined, consistent requirements and high accuracy is the waterfall model. By using a methodical, sequential process that starts with requirements analysis and moves through design, development, testing, deployment, and maintenance, it guarantees high reliability. This approach may not be the best choice if needs are constantly changing or if quick adaptation is needed during development. However, its heavy focus on documentation and thorough testing is in line with the necessity to produce a precise product in a competitive market.    **Explain, Project Workflow:**   1. **Feasibility and Pre-research:** To better understand the problems associated with tracking stray animals, you will perform a thorough investigation of the market and domain during this first phase. You will interact with users of current systems to get useful feedback and recommendations. Furthermore, resource analysis and cost estimation utilizing models like COCOMO will aid in determining the project’s profitability. The project’s viability will be built during this period, MS project and other similar tools will be helpful in managing these initial stages. 2. **System Design:** Prioritizing the requirements will let you start the design, develop, test, and deliver cycle with an incremental approach. The goal is to progressively construct a system that fulfills the needs of the development team and the end users. Creating effective algorithms that maximize performance, memory performance, memory utilization, and user-friendliness will be the main goal of the design phase. Software engineering tools like draw.io will be used to help develop the documentation, which will include use cases and ER diagrams. The method also encourages creative participation and feedback from consumers. 3. **Software development:** The real code for a mobile tracking app is done in this phase, which makes use of HTML, CSS, JavaScript, and database languages. The goal of the development process will be to provide an application that is both aesthetically pleasing and easy to use. Data management and storage will be made possible by the SQL database, allowing for statistical and storage activities that improve user experience. 4. **Prototype implementation and testing work:** A system prototype that first implements a portion of the anticipated functionalities will be created at the end of each milestone. The preceding prototype will be improved upon in subsequent cycles until the ultimate product, which includes all the intended services, is achieved. To minimize errors, testing will take place concurrently with the coding step, eliminating the need for a separate testing phase. Several software engineering tools will be used in the testing process to guarantee a high-quality result. 5. **Maintenance:** Regular performance analysis and user feedback gathering will be carried out in order to remain competitive and current with changing technologies and user expectations. This data will direct error rectification, older version updates, and continuous software product maintenance. Every modification will be recorded for future reference, guaranteeing the app's continued relevance and usefulness.With the help of an incremental development model-based project workflow, you will be able to maintain a flexible and dynamic development process that guarantees correctness, responds to user feedback, and keeps your mobile app for tracking stray animals competitive in an industry that is always changing. |
| **5- Explain**   * **the contribution of national/international technological development if exist.** * **starting a new research and development projects within or outside the team** * **launch new applications or research studies in different technology areas.**   **With whom can we cooperate?**  **Expectations:**  **Published work:**  **Can your output be an input for other similar national/international projects?** |
| Regarding the welfare of stray animals, this mobile tracking app meets a variety of purposes. It serves individuals, small businesses, and local government agencies looking for efficient ways to locate and assist stray animals. The app provides a comprehensive platform for in-the-moment animal help and recovery, fostering coordination among authorities, volunteers, and shelters. Communities and animal activists find it to be a useful tool because of its capacity to enhance animal welfare, link stakeholders, and expedite rescue operations. |

B.3 Innovative and Unique Aspects

# B.3.1

|  |
| --- |
| **1- Describe**   * **differences** * **advantages** * **superiority**   **compared to other similar projects.** |
| * This app is made to help with problems pertaining to stray animals, but the other project is made expressly with healthcare purposes and saving people’s lives. * Animal-related projects frequently receive greater support and involvement from the community. * Working on a stray animal’s project can be a better fit for you to have a strong emotional connection to the topic and for the other project its strict regulatory regulations may help healthcare apps by giving users a sense of security and trust. |

# B.4.1

|  |
| --- |
| **2- Who can contribute to this project in your team?** |
| * Project Manager * Lead programmer * System Designer * Database Developer * Interface Developer * Tester |

C.1 Gantt Chart and Work Packages

C.1.1 Gantt Chart

A close-up of a graph

Description automatically generated

# C.1.2 List of Work Packages

|  |  |
| --- | --- |
| **Work Package No** | 1 |
| **Work Package Name** | **Project Feasibility and Pre-Research (Feasibility Analysis)** |
| **Start-End Date and Time** | **11.10.2023-20.11.2023** |
| **Related Organizations** |  |

|  |
| --- |
| **1- List the activities of work packages.** |
| **1.1 Project Process and Economic Feasibility:** Project initiation: Project initiation: This stage entails determining the goals of the endeavor, the interested parties, and its general parameters. Planning: This stage entails drafting a thorough project plan that addresses scheduling, budgeting, resource allocation, and the creation of a work breakdown structure (WBS). Execution: This stage entails carrying out the project plan, keeping an eye on developments, and adjusting as needed to stay on course. Monitoring and control: During this stage, the project's progress is compared to the plan, and any necessary adjustments are made to maintain the project's course. Closure: This stage entails completing the project and making sure that all goals have been achieved. Cost-benefit analysis: This method establishes if a project is economically feasible by calculating the project's expenses and weighing them against any potential gains. Return on investment (ROI): against ascertain whether a project is financially feasible, the predicted return on investment must be computed and compared against the cost of capital.Net present value (NPV): To ascertain whether a project is economically feasible, the present value of the project's future cash flows is computed and compared to the initial investment. Payback period: This is figuring out how long it will take the project to start making enough money to pay for the original expenditure. |
| **2- Describe the methods and parameters that will be used for work package.** |
| * - Work breakdown structure (WBS): A WBS is a hierarchical project breakdown that divides the project into more manageable, smaller components. Further dissecting each component allows for the creation of work packages that may be assigned to particular individuals or teams. * - Scope of work: A work package's deliverables, tasks, and activities are specified in the scope of work. This comprises a thorough explanation of the tasks that need to be completed, the materials needed, the timetable, and the financial estimate. * - Responsibility matrix: For every work package, a responsibility matrix outlines the duties and responsibilities of each team member or team. This makes it easier to make sure that everyone is aware of expectations and who is in charge of each task. * - Timeline: A timeline gives each work package's timetable, including start and end dates and any dependencies on other work packages, a visual representation. * - Performance metrics: Progress is monitored and each work package's timely and cost-effective completion is guaranteed through the use of performance metrics. This covers metrics like cost variance, schedule variance, and earned value. * - Quality standards: These specify the requirements that must be fulfilled by every work package in order for it to be deemed finished. This covers guidelines for procedures, documentation, and deliverables. * - Risk management: This entails determining possible risks and creating plans to lessen them. This covers hazards like delays, cost overruns, or quality problems that arise from completing work packages. |
| **3- List the experiments, tests and analysis in the work package.** |
| * Usability testing: Holding user testing sessions to assess the application's usability, effectiveness, and user happiness. * Security testing: Penetration testing is done to assess how secure an application is against possible threats. * Performance testing: To make sure the program runs smoothly and satisfies the necessary performance standards, test it under various loads and usage scenarios. * Functionality testing: Checking that the application satisfies all of the functional specifications stated in the project scope. * User acceptance testing: To make sure the application satisfies the needs and specifications of the intended users, user acceptance testing is carried out. * Compatibility testing: Verifying that the program works with a range of hardware, operating systems, and web browsers. * Data analysis: Examining the user information that the program has gathered in order to better understand user behavior and preferences and to enhance the usability and operation of the application. * Localization testing: Verifying that the program is appropriately translated and localized across linguistic and cultural contexts. |
| **4- List the output of work package and its success criteria’s.** |
| * User Requirements Analysis Work Package * **Output**: Documented list of user requirements, needs, and preferences for the mobile/web application. * **Success Criteria**: The user requirements should be comprehensive, clear, and measurable, and must be approved by the project sponsor or stakeholders. * System Design Work Package * **Output**: Detailed system architecture and design documents that will be used for software development. * **Success Criteria:** The system design should be complete, coherent, and technically feasible. The design should meet the user requirements and address any potential issues or risks. * Software Development Work Package * **Output**: Fully functional mobile/web application that meets the user requirements and system design specifications. * **Success Criteria:** The software should be completed on time, within budget, and meet the quality standards set by the project sponsor or stakeholders. The software should be tested for functionality and user experience. * Testing and Validation Work Package * **Output**: Test cases and validation reports for the mobile/web application. * **Success Criteria**: The application should be tested for functionality, usability, and security. All identified issues or bugs should be resolved, and the application should meet the quality standards set by the project sponsor or stakeholders. * Deployment Work Package * **Output**: Released and deployed mobile/web application. * **Success Criteria**: The application should be deployed on time, with minimal downtime or disruptions. The application should meet the quality standards set by the project sponsor or stakeholders. * Maintenance and Support Work Package * **Output**: Maintenance and support plan for the mobile/web application. * **Success Criteria**: The maintenance and support plan should be approved by the project sponsor or stakeholders. The plan should outline how the application will be updated, monitored, and maintained to ensure its continued functionality and usability. Technical support should be available to users. |
| **5- Explain the relation of output with other work packages** |
| The output of the "User Requirements Analysis" work package will provide input to the "System Design" work package, as it will define the needs and preferences of the users.  The output of the "System Design" work package will provide input to the "Software Development" work package, as it will specify the software requirements and architecture.  The output of the "Software Development" work package will provide input to the "Testing and Validation" work package, as it will be tested for functionality and user experience.  The output of the "Testing and Validation" work package will provide input to the "Deployment" work package, as it will ensure that the application meets the quality standards and is ready for release.  The output of the "Deployment" work package will provide input to the "Maintenance and Support" work package, as it will be responsible for fixing bugs, updating features, and providing technical support to the users. |

|  |  |
| --- | --- |
| **Work Package No** | 2 |
| **Work Package Name** | **Based System Design Technology (Analysis & Design stage)** |
| **Start-End Date and Time** | **21.10.2023-09.11.2023** |
| **Related Organizations** |  |

|  |
| --- |
| **1- List the activities of work packages.** |
| * User requirement analysis * System requirements specification * Feasibility analysis * System architecture design * Interface design * Database design * Module design * Algorithm design * GUI design * Security design * Prototype development * Testing and debugging * Performance optimization * Documentation |
| **2- Describe the methods and parameters that will be used for work package.** |
| * User requirement analysis: Surveys, interviews, focus groups, and user personas. * System requirements specification: Functional and non-functional requirements, use cases, and requirement traceability matrix. * Technical, economic, and operational feasibility are all considered in the feasibility analysis. * Component diagrams, high-level designs, and low-level designs make up system architecture design. * Interface design: Mockups, wireframes, and usability testing. * Database design: Entity-relationship diagram, normalization, and data dictionary. * Module design: Object-oriented design, class diagram, and dependency diagram. * Algorithm design: Pseudocode, flowchart, and complexity analysis. * GUI design: Visual design, layout design, and accessibility testing. * Security design: Threat modeling, risk assessment, and security controls selection. * Prototype development: Rapid prototyping and user testing. * Testing and debugging: Unit testing, integration testing, system testing, and debugging tools. * Performance optimization: Code profiling, benchmarking, and caching. * Documentation: User manual, technical manual, and release notes. |
| **3- List the experiments, tests and analysis in the work package.** |
| * User acceptance testing: Verify if the system meets user requirements. * Usability testing: Evaluate the system's ease of use and user experience. * Functionality testing: Verify the system's functional requirements. * Performance testing: Measure the system's response time, throughput, and resource utilization. * Security testing: Assess the system's vulnerability to attacks and data breaches. * Code review: Evaluate the code's readability, maintainability, and scalability. * System integration testing: Verify the system's compatibility with other systems and services. * Stress testing: Evaluate the system's performance under extreme load conditions. * Regression testing: Verify that new changes do not introduce defects in existing functionality. |
| **4- List the output of work package and its success criteria’s.** |
| * User requirement analysis: User requirement specification document * System requirements specification: System requirement specification document * Feasibility analysis: Feasibility report * System architecture design: High-level design document, low-level design document, and component diagram * Interface design: Mockups, wireframes, and usability testing report * Database design: Entity-relationship diagram, database schema, and data dictionary * Module design: Class diagram, dependency diagram, and module interface specification * Algorithm design: Pseudocode, flowchart, and complexity analysis report * GUI design: Visual design, layout design, and accessibility testing report * Security design: Threat modeling report, risk assessment report, and security controls specification document * Prototype development: Functional prototype and user testing report * Testing and debugging: Test cases, test results, and bug report. * Performance optimization: Performance benchmark report and optimization report * Documentation: User manual, technical manual, and release notes. |
| **5- Explain the relation of output with other work packages** |
| The output of each work package is an input to the next work package. For example, the user requirement specification document is an input to the system requirement specification work package. The system requirement specification document is an input to the system architecture design work package. The high-level design document and component diagram are inputs to the module design work package. The class diagram and module interface specification are inputs to the algorithm design work package. The functional prototype is an input to the testing and debugging work package. |

|  |  |
| --- | --- |
| **Work Package No** | 3 |
| **Work Package Name** | **Development of System Software (Development Stage)** |
| **Start-End Date and Time** | **10.11.2023-04.12.2023** |
| **Related Organizations** |  |

|  |
| --- |
| **1- List the activities of work packages.** |
| * Define software requirements, * Design software architecture * Develop software modules and components. * Integrate software components and modules. * Test software functionality and performance * Fix defects and bugs * Deploy software to production environment. * Provide post-deployment support |
| **2- Describe the methods and parameters that will be used for work package.** |
| * For defining software requirements: interview stakeholders, gather user stories, analyze existing systems, and document requirements using tools such as use cases, user stories, and requirements documents. * For designing software architecture: use software design patterns, follow design principles such as SOLID, and use design tools such as UML diagrams. * For developing software modules and components: follow coding standards, use version control, and use appropriate programming languages and frameworks. * For integrating software components and modules: use integration testing, follow integration design patterns, and use integration frameworks. * For testing software functionality and performance: use automated testing, perform functional testing, load testing, and stress testing. * For fixing defects and bugs: use a bug tracking system, prioritize and triage bugs, and perform regression testing. * For deploying software to production environment: use deployment automation tools, follow deployment checklists, and monitor deployment progress. * For providing post-deployment support: provide helpdesk support, monitor system performance, and fix defects and bugs reported by users. |
| **3- List the experiments, tests and analysis in the work package.** |
| * Unit testing * Integration testing * System testing * Acceptance testing * Performance testing * Load testing * Stress testing * Security testing * Usability testing |
| **4- List the output of work package and its success criteria’s.** |
| * Software modules and components * Integrated software system * Tested and validated software system * Deployed software system to production environment * Bug-free and reliable software system * User-friendly and efficient software system * Meeting stakeholder requirements and expectations * Meeting project schedule and budget constraints |
| **5- Explain the relation of output with other work packages** |
| The output of this work package (developed and tested software system) is related to the output of the previous work packages (software requirements and software architecture) as it builds on the requirements and design to develop a working software system. The output is also related to the next work package (software maintenance and support) as it provides a reliable and maintainable software system for ongoing support and improvements. |

|  |  |
| --- | --- |
| **Work Package No** | 4 |
| **Work Package Name** | **Prototype Implementation and Test Study and Maintenance (Test & Maintenance stage)** |
| **Start-End Date and Time** | **05.12.2023-24.12.2023** |
| **Related Organizations** |  |

|  |
| --- |
| **1- List the activities of work packages.** |
| Conducting system testing  Conducting user acceptance testing  Conducting performance testing  Conducting security testing  Conducting maintenance and support activities  Monitoring system performance and user feedback  Identifying and addressing defects or bugs  Updating system documentation and user manuals |
| **2- Describe the methods and parameters that will be used for work package.** |
| Developing test plans and test cases  Executing tests and recording results  Analyzing test results and identifying defects  Prioritizing and addressing defects |
| **3- List the experiments, tests and analysis in the work package.** |
| - Testing the system to make sure that every part functions as a whole.  -Carrying out user acceptability testing to make sure the program satisfies user requirements.  -Performance testing to make sure the program satisfies performance standards.  -Performing security tests to make sure the program is safe. |
| **4- List the output of work package and its success criterias.** |
| **Outputs:**  Completed prototype implementation.  Test plans and test result reports  Bug reports and solutions  Updated system documentation and user guides  Maintenance and support procedures  **Success Criteria’s:**  Availability of the prototype application at a certain level  Tests reach and pass a certain coverage area.  Reducing the error rate below a certain level  Updated documentation to meet user needs.  Maintenance and support procedures have a certain quality standard. |
| **5- Explain the relation of output with other work packages** |
| Requirement Analysis: The outputs of the Prototype Application and Testing and Maintenance work packages help to verify the applicability and effectiveness of the requirements identified in the requirement analysis work package. Therefore, the prototype work and test results can assist in validating the outputs of the requirement analysis work package.  Design: The Prototype Application and Testing and Maintenance work packages test the accuracy of the design goals identified in the design work package. The prototype work and test results can confirm the accuracy of the design decisions made in the design work package or suggest the need for a re-evaluation of the design.  Coding: The Prototype Application and Testing and Maintenance work packages test the accuracy and functionality of the code written in the coding work package. Therefore, the prototype work and test results can help to detect and correct any errors made in the coding work package.  Testing and Verification: The Prototype Application and Testing and Maintenance work packages are directly related to the Testing and Verification work package. The prototype work and test results can help to verify the test scenarios identified in the Testing and Verification work package.  Maintenance: The Prototype Application and Testing and Maintenance work package is directly related to the Maintenance work package. The prototype work and test results can help to identify the maintenance requirements of the application and assist in planning the corrections to be made in the Maintenance work package. |

# C.1.3 List of Milestones (should be matched in the Grantt chart)

|  |  |  |
| --- | --- | --- |
|  | **Description of Output** | **Expected Time Interval** |
| 1 | **Project Feasibility and Pre-Research (Feasibility Analysis)** | **11.10.2023-20.10.2023** |
| 2 | **Based System Design Technology (Analysis & Design stage)** | **21.10.2023-09.11.2023** |
| 3 | **Development of System Software (Development Stage)** | **10.11.2023-04.12.2023** |
| 4 | **Prototype Implementation and Test Study and Maintenance (Test & Maintenance stage)** | **05.12.2023-24.12.2023** |
| 5 | **Project Closure** | **09.12.2023** |

# C.1.4 List of Risks *(see following example, write possible risks for your project!)*

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Probability | Effects | Your Strategy |
| The time required to develop the software is underestimated. | High | Serious | Break the project down into smaller Manageable tasks. estimating fewer components is frequently more accurate than estimating the complete project at once. |
| Software tools cannot work together in an integrated way. | High | Tolerable | Rigorously test your integration to guarantee data consistency and dependability. Implement monitoring and alerting systems to detect and resolve any integration issues as soon as possible. |
| Customers fail to understand the impact of requirements changes. | Moderate | Tolerable | Educate the client on the software development process. Including the effects of requirement modifications. Assist them in comprehending how adjustment might impact the project timeline and budget. |
| The rate of defect repair is underestimated. | Moderate | Tolerable | Replace potentially defective components with more reliable bought-in components. |
| The size of the software is underestimated. | High | Serious | Investigate buying SW components.  Investigate use of a program generator. |
| Code generated by code generation tools is inefficient. | Moderate | Insignificant | Selecta code generation tool that is known for creating efficient code. |
| Key staff are ill at critical times in the project. | Moderate | Serious | Reorganize teams so that there are more overlap of work and people therefore understand each other’s jobs. |
| The database used in the system cannot process as many transactions per second as expected. | Moderate | Serious | Investigate the possibility of buying a higher-performance database. |

C.2 Project Management and Organization

# C.2.1 Project Team

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Personnel Name** | **Title** | **ID** | **Education Status** | **Graduation Date** | **Date of Starting Work** | **Idea Owner** |
| Omar Ahmed | Project Manager | 20801041 | Undergraduate | 2025 | 11/10/23 | Yes |
| Ahmed Habbadi | UI Designer | 21009123 | Undergraduate | 2026 | 11/10/23 | Yes |
| Ahmed Salmi | Database Developer | 22703233 | Undergraduate | 2026 | 11/10/23 | Yes |
| Zeynep Colak | Tester | 17300009 | Undergraduate | 2023 | 11/10/23 | Yes |

# C.2.2 Organization Scheme (an example is given below!)

A diagram of a company

Description automatically generated

D.1 Economic Forecasts

|  |
| --- |
| **1- Evaluate the commercialization potential of project outcomes. List possible risks here?** |
| Funding: Finding sufficient funding to commercialize the project can be challenging. Various financial needs, such as business development, marketing, software development, and user support, must be addressed.  Market Research: Stray Animals Tracking Mobile App requires comprehensive market research to understand whether there is demand in a specific market. It is essential to assess market size and competitors.  Competition: There may already be similar applications and services in the market. Dealing with this competition and differentiation can be a formidable task.  Legal and Ethical Issues: Stray animals tracking applications may be subject to legal and ethical concerns, such as privacy, data protection, and animal rights. It is crucial to navigate these issues carefully.  Technological Challenges: The technical requirements of the application and the need for continuous updates can pose technological challenges. It must also work seamlessly on various devices.  User Adoption: Will users embrace your application? Gathering user feedback and making the application user-friendly is essential.  Collaboration and Partnerships: Collaborating with partners can enhance the success of your application, but finding the right partners and managing relationships is crucial.  Data Security: Ensuring the security of users' personal data, animal tracking data, and other sensitive information is a significant concern. There is a risk of data breaches or leaks.  Legal Matters and Permissions: Compliance with relevant regulations, obtaining permissions, and securing the necessary authorizations for animal tracking are essential. |

|  |  |
| --- | --- |
| **2- List your expectations to your team which are come by your project** | |
| Time-to-market (month): | 9-12 month |
| The expected increase in sales revenue (%): | %30 |
| The expected increase in market share (%): | %20 |
| Time to start to gain: | 3-6 month |

D.2 National Outcomes

|  |
| --- |
| **1- Specify the output that may be subject to patent, utility model and industrial design registration in the project.** |
| Algorithm and Data Processing Methods: Any unique algorithms or methods developed for tracking and managing stray animals, analyzing data, or improving the app's functionality could potentially be patented.  Design of the User Interface (UI) and User Experience (UX): If an application's user interface or any of its creative UX aspects have a unique and eye-catching look, they may be eligible for industrial design registration.  Hardware Components: If the project involves the development of any unique hardware components, such as GPS trackers or sensors, these may be candidates for utility model registration if they offer a novel and practical technical solution.  Mobile App Features: Specific features within the mobile app, such as real-time tracking, data visualization, or innovative notification systems, could potentially be patented if they represent inventive and non-obvious solutions.  Data Collection and Analytics Techniques: Methods for collecting and analyzing data related to stray animals' movements, behavior, or health could be considered for patent protection, especially if they offer a unique and valuable approach.  Innovative Animal Tracking Devices: If the project involves the development of new tracking devices or collars for animals, these may be eligible for patent or utility model protection, depending on their technical innovations. |
| **2- Explain the potential of project and its outputs that may influence social life, education, health etc.** |
| * Social Life:   Improved Animal Welfare: The project can contribute to better animal welfare by enabling the tracking and care of stray animals. This may lead to a reduction in animal suffering, promoting a more compassionate and responsible society.  Community Engagement: The app can foster community engagement by encouraging individuals to get involved in animal rescue efforts, building a sense of shared responsibility and collaboration.   * Education:   Awareness and Education: The app can serve as an educational tool, helping people understand the challenges faced by stray animals and the importance of responsible pet ownership. It can also provide resources for schools and educators to teach students about animal welfare and tracking technology.  Practical Learning: Students pursuing fields like veterinary science, animal behavior, or environmental studies can gain practical experience through involvement with the app, contributing to their education and career development.   * Health:   Zoonotic Disease Prevention: By tracking and managing stray animals, the app can indirectly contribute to public health by reducing the risk of zoonotic diseases transmitted from animals to humans.  Mental Health: Interacting with and helping animals can have a positive impact on people's mental health. The app can provide opportunities for users to engage with animals, which can be therapeutic.   * Environment:   Ecosystem Balance: Stray animals can sometimes disrupt local ecosystems. By managing and tracking them, the app can help maintain a more balanced environment, benefiting native wildlife and vegetation.   * Economic Impact:   Job Creation: The project can create job opportunities related to animal care, app development, data analysis, and more, contributing to local economies.   * Safety:   Human Safety: The app can contribute to public safety by helping to locate and manage potentially aggressive or sick stray animals, reducing the risk of harm to humans.   * Community Building:   Volunteer and Community Building: The app can encourage the formation of volunteer groups and communities dedicated to animal welfare, promoting social cohesion. |
| **3- Explain the positive and negative effects of project outputs for environment and human being.** |
| **Positive Effects on the Environment and Human Beings:**  Animal Welfare: Positive: The project can significantly improve animal welfare by helping track and rescue stray animals, providing them with necessary care and shelter. Positive: Reducing the suffering of stray animals contributes to a more compassionate and humane society.  Environmental Impact: Positive: Managing stray animals can help protect local ecosystems from potential disruptions caused by these animals. Positive: By addressing issues related to stray animals, the project can indirectly support the conservation of native wildlife and plant species.  Public Health: Positive: Managing stray animals can reduce the risk of zoonotic diseases, enhancing public health and safety. Positive: The project can create awareness about the importance of responsible pet ownership and proper vaccination, contributing to a healthier community.  Community Building: Positive: The project can foster a sense of community by encouraging individuals to get involved in animal rescue efforts, promoting collaboration and shared responsibility. Positive: Volunteer groups and communities dedicated to animal welfare can form, promoting social cohesion and a sense of purpose among participants.  **Negative Effects on the Environment and Human Beings:**  Environmental Impact: Negative: In some cases, tracking and capturing stray animals may disrupt local ecosystems and potentially affect native species. Negative: The project may inadvertently encourage the removal of certain species of stray animals, which can have ecological consequences.  Economic Impact: Negative: The project may require funding for operations, which could divert resources from other environmental or community projects. Negative: In some cases, the project could create competition with existing animal-related businesses.  Community Impact: Negative: There may be resistance to or opposition to the project from some members of the community, particularly if it affects existing cultural practices or traditional views on stray animals. Negative: Stray animal management efforts can sometimes face challenges related to animal rights and ethics, leading to public disagreements. |

(M013) Instrument / Equipment / Software / RELEASE PURCHASES

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Name** | |  | | | | | | | | | |
| **Line no** | **Instrument / Equipment / Software / Publication Name** | | **No. of Item** | **Capacity** | **Technical specification** | **Purpose of Project Activities** | **Post-Project Place of Use / Purpose** | | **Unit Price (USD)** | **Unit Price (TL)** | **Total Amount (TL)** |
| **R & D** | **Production** |
| **1** | **Microsoft Office 365** | | **4** |  | **Collection of office-related applications** | **Management** | **yes** |  | **100** | **2830** |  |
| **2** | **Internet Connection** | | **1** |  | **Min.20 Mbps** |  | **yes** |  | **90** | **2550** |  |
| **3** | **Computer/Laptop** | | **4** |  | **Core i7 quadcore 2.2 GHz or faster-32 Gb, 100 Gb unused space** | **Involved in every task of the project** | **yes** | **Yes** | **1500** | **42400** |  |
| **4** | **MS Project** | | **4** |  | **Project Management Software** | **Planning** |  | **yes** | **120** | 3400 |  |
| **5** | **Modelio** | | **2** |  | **Tools for drawing diagrams** | **Uml diagrams** | **yes** |  |  |  |  |
| **6** | **HTML** | | **1** |  | **Creating electronic documents** | **Coding Purposes** | **yes** |  |  |  |  |
| **7** | **Python** | | **2** |  | **Language for developing the web application** | **Coding purposes** | **yes** |  | **350** | **9900** |  |
| **8** | **MySQL** | | **1** |  | **Tool for creating database** | **Database Operations** |  | **yes** | **200** | **5650-5750** |  |
| **9** | **Jest** | | **1** |  | **Test Automation** |  |  | **yes** |  |  |  |
|  |  | |  |  |  |  |  |  |  | **TOTAL** | **66780** |

(M030) Quarterly Estimated Cost Form (TL)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name:** | | | | |
| **Cost Item** | **2023** | | **TOTAL**  **(TL)** | **TOTAL COST RATE OF CONTENTS (%)** |
| **I** | **II** |
| **Personnel** | 20,000 | 20,000 | 40,000 | 30 |
| **Travel** | 500 | 500 | 1,000 | 10 |
| **Instrument / Equipment / Software / Publications** | 5,000 | 5,000 | 10,000 | 30 |
| **Domestic Works Made by R & D and Testing Institutions** | 10,000 | 10,000 | 20,000 | 5 |
| **International Works Made by R & D and Testing Institutions** |  |  |  |  |
| **Domestic Services Procurement** | 10,000 | 10,000 | 20,000 | 5 |
| **Overseas Service Procurement** |  |  |  |  |
| **Material** | 15,000 | 15,000 | 30,000 | 20 |
| **TOTAL COST** | 61,000 | 61,000 | 122,000 | 100 |
| **CUMULATIVE COST** |  |  |  | 100 |
| **IN THE PROJECT TOTAL MAN-MONTH** | | | 121,500 | |

|  |  |
| --- | --- |
| Activity | Task |
| A | Project Feasibility and Pre-research |
| B | Based System Design Technology (Design stage) |
| C | Development of System Software |
| D | Web System concept for software development |
| E | Software integration |
| F | Coding |
| G | Prototype implementation and test study and maintenance |

APPENDIX

**A1. CPM and Network Diagram**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Activity | Predicates | Duration | Optimistic Time | Most Likely | Pessimistic Time | Expected Duration |
| A | - | 12 | 10 | 12 | 14 | 12 |
| B | A | 15 | 13 | 15 | 17 | 15 |
| C | A | 19 | 16 | 19 | 22 | 19 |
| D | C | 7 | 5 | 7 | 9 | 7 |
| E | B, C, D | 5 | 2 | 5 | 7 | 4.83 |
| F | E | 10 | 8 | 10 | 12 | 10 |
| G | F, E | 4 | 3 | 4 | 6 | 4.16 |

NETWORK DIAGRAM:



|  |  |
| --- | --- |
| Path | Expected Duration |
| ABEFG | 45.99 |
| ABEG | 35.99 |
| ACDEG | 46.99 |
| ACEG | 39.99 |
| ACDEFG | 56.99 |
| ACEFG | 49.99 |

**The critical path is “ACDEFG” because is the longest one with 56.99 of expected duration.**

**A2.1 Effort and Duration estimates with COCOMO**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Business Functions** | **Simple** | **Simple Weight** | **Average** | **Average Weight** | **Complex** | **Complex Weight** | **UFPs** |
| **User Input Functions (IT)** | **6** | **3** | **3** | **4** | **1** | **6** | **36** |
| **User Output Functions (OT)** | **4** | **5** | **2** | **5** | **10** | **7** | **100** |
| **User Inquiries (QT)** | **4** | **3** | **6** | **4** | **4** | **6** | **60** |
| **Internal Files (FT)** | **3** | **7** | **5** | **10** | **7** | **15** | **176** |
| **External Interfaces (ET)** | **2** | **6** | **3** | **7** | **3** | **10** | **63** |
|  |  |  |  |  |  | **UFPs** | **435** |

**A2.2 Technical Complexity Factor (TCF) Table**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Factors** | **Complexity** | **Complexity Value** |
| **1** | **Data Communication** | **Significant** | **4** |
| **2** | **Distributed Data Processing** | **Moderate** | **2** |
| **3** | **Performance Criteria** | **Significant** | **4** |
| **4** | **Heavily utilized Hardware** | **Incidental** | **1** |
| **5** | **High Transaction Rates** | **Significant** | **4** |
| **6** | **Online Data Entry** | **Average** | **3** |
| **7** | **Online Updating** | **Average** | **3** |
| **8** | **End-user Efficiency** | **Significant** | **4** |
| **9** | **Complex Computations** | **Incidental** | **1** |
| **10** | **Reusability** | **Average** | **3** |
| **11** | **Ease of Installation** | **Significant** | **4** |
| **12** | **Ease of Operation** | **Essential** | **5** |
| **13** | **Portability** | **Significant** | **4** |
| **14** | **Maintainability** | **Essential** | **5** |
|  |  | **DI** | **47** |
|  |  | **UFP** | **435** |
|  |  | **FP= UFP\*[0.65+0.01\*DI]** | **487.2** |

**A2.3 Final Effort and Duration estimates**

We will be estimating our effort and the project’s length using COCOM’s Basic Organic mode because we are working on a small project with a small team and no technical innovation. The project’s front end and back end will be implemented using Python frameworks; the average SLOC/FP for Python is 53 lines of code, according to “namcook.com” case example 6.

The KLOC for our project is equal to SLOC\*FP/1000 = 25.82

53 \* 487.2/1000 = 25.82

Now, given that:

E = a (KLOC)^b

D = c(E)^d

Where Effort = E & Duration = D. Our corresponding coefficients for this project are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project | a | B | c | D |
| Organic mode | 2.4 | 1.05 | 2.5 | 0.38 |

Therefore,

E = 2.4 \* 25.82^1.05 = 72.9 PM

D = 2.5 \* 72.9^0.38 = 12 months

Required number of team members = E/D = 72.9/12 = 6, according to COCOMO estimations.

Software Requirements Specification

for

**Stray Animals Tracking Mobile App**

Version 1.0

Prepared by

Group #12

10 11 2023

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The Stray Animals Tracking Mobile App was developed to protect stray animals and facilitate support from compassionate individuals. This application is designed to identify lost or distressed animals on the streets, notify users of their locations, and streamline the process of requesting assistance in emergency situations. Additionally, it seeks to enhance communication between organizations and volunteers dedicated to caring for stray animals, providing a user-friendly platform to strengthen collaboration and effectively manage resources. The Stray Animals Tracking Mobile App aspires to bring together animal enthusiasts, local communities, and animal welfare organizations to collectively contribute to the well-being of stray animals.

## Document Conventions

## Intended Audience and Reading Suggestions

This document caters to a diverse audience interested in contributing to the Stray Animals Tracking Mobile App project. It is specifically made for mobile app developers seeking insights into technical aspects, animal welfare associations and volunteers dedicated to helping stray animals, individual enthusiasts wishing to support through the application, local government bodies concerned with stray animal issues, and donor organizations interested in providing financial support. To navigate the document effectively, developers can focus on technical details and API documentation, users on scenarios and interface design, and all stakeholders on sections outlining project objectives, benefits, support channels, and contribution paths.

## Product Scope

The Stray Animals Tracking Mobile App is designed to address the protection of stray animals and facilitate the swift location of lost animals through a mobile application. The app focuses on notifying users about the location of lost or distressed stray animals, creating emergency assistance calls, and fostering effective communication between animal welfare organizations and volunteers. Prioritizing a user-friendly interface with an intuitive design, the application aims to be accessible to users of all levels. Additionally, it enhances the user experience by incorporating features such as mobile notifications, map integration, and animal profiles. The Stray Animals Tracking Mobile App seeks to improve support and aid processes for stray animals by enriching user experiences and fostering community engagement. The goal is to increase societal awareness about stray animals and empower animal enthusiasts to take effective action.

## References

Smith, J., & Doe, A. (Yıl). "Stray Animals: A Comprehensive Study on the Challenges and Solutions." Journal of Animal Welfare, 10(2), 123-145.

Animal Rescue Foundation. (2022). "Stray Animals Tracking: Best Practices in Mobile App Development for Animal Welfare."

World Health Organization. (2021). "Guidelines for Ethical Treatment of Stray Animals in Urban Environments."

# Overall Description

## Product Perspective

The Stray Animals Tracking Mobile App is a comprehensive mobile application designed to address the protection of stray animals and the swift location of lost ones. Equipped with a user-friendly interface, the application enables users to easily locate stray animals, aid, and communicate with animal welfare organizations. The app enriches the user experience through features such as mobile notifications, interactive map integration, and detailed animal profiles. Moreover, it facilitates effective collaboration among various stakeholders, bringing together individual animal enthusiasts, local authorities, and animal welfare associations. This not only benefits individuals but also contributes significantly to the broader community and organizations dedicated to helping stray animals. The Stray Animals Tracking Mobile App merges technology and compassion, offering a solution-oriented perspective to address issues related to stray animals.

## Product Functions

Animal Location Tracking:

Determine the locations of lost or distressed stray animals for users.

Provide real-time location information to users through map integration.

Emergency Assistance Calls:

Enable users to quickly create emergency assistance calls.

Send emergency notifications to animal welfare organizations and volunteers.

Animal Profile Management:

Create and manage detailed profiles for stray animals.

Include information such as the animal's health status, photos, and other relevant details in profiles.

Notifications and Updates:

Send users important updates and notifications through the application.

Inform users about local events, campaigns, and emergency situations.

User-Friendly Interface:

Provide an easily understandable and user-friendly interface.

Design an intuitive layout for users, including those using the application for the first time.

Communication and Community:

Enhance communication among users and foster community building through interactive features.

Allow users to share their experiences and engage through comments and sharing options.

Security and Data Privacy:

Safely store users' personal information.

Process user data in compliance with data privacy policies.

Multilingual Support:

Offer multilingual support for users from different regions and language groups.

## User Classes and Characteristics

1. Regular User:

Characteristics:

Description: Regular users are those who use the app to report, track, and assist stray animals.

Features:

Ability to create and manage a user profile.

Access to animal location tracking functionalities.

Capability to create emergency assistance calls.

Receive notifications about lost animals and emergency situations.

Participate in community engagement features (comments, sharing, etc.).

Needs:

Intuitive user interface for easy navigation.

Clear instructions on using location tracking and emergency assistance features.

Timely notifications and updates about local animal-related events.

2. Animal Welfare Organization Representative:

Characteristics:

Description: Representatives from animal welfare organizations who use the app to coordinate and respond to assistance requests.

Features:

Access to a dedicated organization profile.

Receive and respond to emergency assistance calls.

View detailed animal profiles and their locations.

Collaborate with other organizations and volunteers.

Receive notifications about local events and collaborations.

Needs:

Efficient notification system for emergency assistance calls.

Tools for managing and updating organization details.

Integration with organizational communication channels.

3. Volunteer:

Characteristics:

Description: Individuals willing to volunteer their time and efforts to assist in animal welfare activities.

Features:

Ability to create and manage a volunteer profile.

Access to animal location tracking functionalities.

Receive notifications about lost animals and emergency situations.

Collaborate with animal welfare organizations and other volunteers.

Participate in community engagement features (comments, sharing, etc.).

Needs:

Clear information on available volunteering opportunities.

Seamless communication with organizations and other volunteers.

Recognition for contributions and achievements.

4. Administrator/Mod:

Characteristics:

Description: App administrators or moderators responsible for overseeing and managing the community.

Features:

Monitoring and moderating user-generated content.

Enforcing community guidelines and policies.

Handling reported issues and disputes.

Managing app settings and configurations.

Providing support to users and organizations.

Needs:

Robust moderation tools and dashboards.

Efficient communication channels with users and organizations.

Regular training and updates on community management policies.

## Operating Environment

Mobile App Platforms:

Supported Platforms: Mobile devices running iOS and Android operating systems.

App Stores: App Store (iOS) and Google Play Store (Android).

Internet Connectivity:

Required Connection: Internet access is required for the full functionality of the application.

Data Communication: Secure and fast data communication using the HTTPS protocol.

Map Integration:

Map Services: Appropriate map services for real-time map integration within the application (e.g., Google Maps, Map box).

Location Services: Access to device location services to determine user and animal locations.

Database System:

Type: Relevant, secure, and scalable database system (e.g., SQL or NoSQL-based).

Security Measures:

Data Security: Encryption and security measures to protect user information.

Authorization and Authentication: Authentication processes for users and organizations.

Application Updates: Regular application updates to address security vulnerabilities and introduce new features.

Various Devices and Screen Sizes:

Responsive Design: Ensuring a responsive design for compatibility with different mobile devices and screen sizes.

Language Support:

Multilingual Support: Providing multilingual support for users from different language groups.

Performance Optimization:

Speed and Performance: Performance optimization for the application to operate efficiently and swiftly.

Backup and Recovery:

Data Backup: Regular backup processes for user information and system data.

Recovery Plans: Plans for recovery in case of system crashes or data loss.

## Design and Implementation Constraints

Data Privacy and Security:

Constraint: Full compliance with data privacy and security laws is required.

Solution: Implement security measures such as encryption, authorization, and authentication to protect user data.

Map Services:

Constraint: Access to map services requires the provision of appropriate API keys.

Solution: Collaborate with a reliable map service provider and ensure the security of API keys for map integration.

Multilingual Support:

Constraint: Enabling multilingual support requires language translation and localization processes.

Solution: Utilize language translation services targeting various language groups and implement a multilingual support infrastructure.

Internet Connectivity:

Constraint: Full functionality of the application relies on internet connectivity.

Solution: Implement data compression and caching on the application side to make it more tolerant to low internet speeds.

Application Updates:

Constraint: Regular application updates are necessary to fix security vulnerabilities and introduce new features.

Solution: Keep users informed through automatic update notifications and user-friendly update processes.

Database Constraints:

Constraint: Optimize database queries and use a scalable database solution.

Solution: Implement database indexing, query optimization, and use a scalable database structure to enhance performance.

Screen Sizes and Resolutions:

Constraint: The application should work consistently across different screen sizes and resolutions.

Solution: Design a responsive user interface based on responsive design principles to adapt to various screen sizes.

User Engagement and Feedback:

Constraint: Responding promptly to user feedback and complaints is necessary.

Solution: Establish user support channels, feedback forms, and ensure regular user interaction.

## User Documentation

User Manual:

A comprehensive guide that outlines the installation process, system requirements, and step-by-step instructions on using the Stray Animals Tracking Mobile App. This manual will cover both basic and advanced features, ensuring users can navigate and utilize the app effectively.

On-line Help:

In-app assistance that provides context-specific guidance to users. The on-line help will be easily accessible from within the application, offering instant support on various functionalities, settings, and troubleshooting.

Tutorials:

A series of tutorials designed to help users understand specific features of the app. These tutorials will include detailed explanations, visuals, and practical examples to ensure users can make the most out of the Stray Animals Tracking Mobile App.

FAQ Section:

A frequently asked questions section to address common queries and concerns. This will serve as a quick reference for users looking for immediate solutions or clarification on specific aspects of the application.

Delivery Formats:

The user documentation will be provided in digital formats, including PDFs and online web pages. This ensures easy accessibility for users across various devices and platforms.

Standards:

The documentation will adhere to industry-standard writing and formatting conventions to enhance clarity and user-friendliness. It will also follow best practices for technical documentation, ensuring that users can quickly find the information they need.

Feedback Mechanism:

A system for users to provide feedback on the documentation will be implemented. This will help identify areas of improvement, address any user confusion, and continuously enhance the quality of the user documentation.

Version Control:

The documentation will be versioned to align with software releases. Each version will clearly indicate the changes and updates, allowing users to stay informed about new features or modifications.

## Assumptions and Dependencies

The system's compatibility with hardware and web browsers is critical to the project's success.

# External Interface Requirements

## User Interfaces

For the convenience of users interested in a website about stray animals, the user interface is made to be straightforward and easy to use. Basic and broad explanations of the user interface are provided below:  
  
**-**Users who are interested in stray animals will be required to register or log in when they visit the webpage.   
  
-If there are any problems with a user's login or registration details, the system will **guide them**.  
   
 -Every user will get access to a customized webpage with content about stray animals on it.  
   
 -A guide will be made available to new users who are interested inadopting stray animals **to** make **their** journey easier.  
  
-All around the website, information regarding stray animals will be prominently presented.  
  
-There will be a help button on every interface, guaranteeing assistance with topics pertaining to stray animals.

## Hardware Interfaces

Strong compatibility with Android and iOS devices is provided by the system, which requires a minimum of 2GB of hard disk space, 512MB of memory, and an i3 or comparable processor. Its adaptable interface ensures a seamless and optimum user experience by customizing interactions to each device's basic capabilities. Wi-Fi or 3G connectivity is created to enable effective communication with the remote server; the addition of a Global Positioning System (GPS) allows for accurate location tracking of neighboring restaurants, hence improving delivery efficiency. The IP communication protocol is used to establish network connections, demonstrating a dedication to adaptability and performance across a wide range of mobile devices, even those with the most basic characteristics.

## Software Interfaces

Compatibility with several operating systems is guaranteed by the Stray Animal Information System, which includes iOS 15 and above, Android 10.0 and above, and Windows 7 and above. The system includes a Relational Database Management System (RDBMS) for smooth data management. This system offers an easy-to-use database update, control, and usability platform. This guarantees effective tracking and management of data about stray animals, adding to a thorough and simple experience for everyone who uses the system.

## Communications Interfaces

The web server and client browser of the Stray Animal Information System communicate with each other seamlessly thanks to the Hypertext Transfer Protocol (HTTP). The TCP/IP protocol facilitates communication between servers and web browsers. The system sends notifications, verification codes, and other pertinent information to clients via email using the Simple Mail Transfer Protocol (SMTP) to improve user communication. As a result, a strong security protocol will be put in place to ensure financial transactions' security. A secure internet connection is also set up to connect users to the database, underscoring the dedication to protecting personal data about stray animals.

# System Features

## User Features

**4.1.1 Account Management**

### 4.1.1.1 Description and Priority

With the use of this function, users can get fast notifications when new stray animal sightings occur or when there are changes about animals they have reported. By giving timely information, it improves the app's responsiveness and user engagement.

### 4.1.1.2 Stimulus/Response Sequences

Certain functions within this category are initiated automatically by the system's default settings. Others are in reaction to a user's directive.

### 4.1.1.3 Functional Requirements

REQ-1: New users shall be able to register.

REQ-2: Old users can login

REQ-3: Logged in users shall be able to logout.

REQ-4: Users shall be able to change the language setting.

REQ-5: Users shall be able to renew their password if they forget it.

REQ-6: System shall notify users in case of invalid entry.

REQ-7: Users can preview pets without logging in.

REQ-8: Users can live track animals without logging in.

REQ-9: Users can sign up as volunteers.

## 4.1.2 Customer Menu Integration

**4.1.2.1 Description & Priority:**

The range of what customers can do with the animals on their menu is covered by these features.

**4.1.2.2 Stimulus/Responses Sequences:**

The system will disable certain options and filter the customer's choices based on their selected preferences.

**4.1.2.3 Functional Requirements:**  
REQ-1: Animals details should be shown to the user.

REQ-2: The user must be able to see the available animals.

REQ-3: The user must be able to see the location of the animals.

REQ-4: The user will be able to see the animals’ breeds, age of the animal.

REQ-5: Animals on the list of choices may be deselected by the customer.

### 4.1.3 Animal Tracking:

#### 4.1.3.1 Description and Priority:

The system will provide customers with the privilege to view more accurate locations once they apply for adoption. It has further information as well.

#### 4.1.3.2 Stimulus/Response Sequences:

The adoption procedure and the instant an application is submitted will be handled by the system. The user can choose whether to see some of the details.

#### 4.1.3.3 Functional Requirements:

REQ-1: The client must be able to apply for adoption.

REQ-2: The client must be able to use GPS to follow the progress of their delivery on a map if they allow map access.

REQ-3: Real-time updates are required for all.

REQ-4: If the scheduled delivery time is not available by the volunteer, the system will alert the customer.

**4.2 Administrator Features:**

**4.2.1 Account Management:**

**4.2.1.1 Description and Priority:**

Those in charge of managing the Stray Animals website need personal accounts to gain access to the framework specifically assigned to their tasks. This is essential for maintaining security because requiring a password strengthens the system's defenses and ensures that personal data on stray animals is kept safe.

**4.2.1.2 Stimulus/Response Sequences:**

Anyone utilizing the Stray Animals website must have an account; by default, the system will prompt users to create an account. Users can access and unlock a variety of control panels that improve their capacity to deal with and handle data pertaining to stray animals.

**4.2.1.3 Functional Requirements:**

REQ-1: Administrators, often known as "admin," are required to register with the system.

REQ-2: The ability to log in and log out should be available to administrators.

REQ-3: The ability for administrators to modify contact details for stray animals is required.

REQ-4: Stray animal service regions must be chosen by administrators.

REQ-5: Delivery schedules for stray animals ought to be customizable by administrators and volunteers.

REQ-6: The ability to upload the Stray Animals website logo is necessary for administrators.

REQ-7: Admins shall be able to accept adoption applications.

REQ-8: Admins can approve or deny volunteer applications.

**4.2.2 Options Management:**

**4.2.2.1 Description and Priority:**

This is vital: Because the Stray Animals web application has a significant impact on the business, administrators need to be able to control it. Administrators play a crucial role in maintaining and modifying the Stray Animals website to ensure optimal operation and user experience, as seen by their authority to make changes that directly affect customers.

**4.2.2.2 Stimulus/Response Sequences:**

This category's actions are all started by user commands. All modifications will be implemented.

**4.2.2.3 Functional Requirements:**

REQ-1: The ability for administrators to modify and update the available animals.

REQ-2: The ability to add new pets for adoption must be granted to administrators.

REQ-3: The authority to alter or update status should be granted to administrators.

REQ-4: All stray Animals must be visible to administrators.

**4.2.3 Adoption Management:**

**4.2.3.1 Description and Priority:**

On the Stray Animals website, the administrator will be able to examine and control the specifics and statistics of applications for adoption that they come across.

**4.2.3.2 Stimulus/Response Sequences:**

On the Stray Animals website, a specific area devoted to the history of all adoption applications will be accessible.

**4.2.3.3Functional Requirements:**

REQ-1: Access to the current outstanding applications by administrators is required.

REQ-2: The ability to approve or reject applications is a requirement for administrators.

REQ-3: The ability to remove applications should belong to administrators.

REQ-4: The application history of every customer must be visible to administrators.

REQ-5: Applications that have been completed must be visible to administrators.

REQ-6: Cancelled applications must be visible to administrators.

REQ-7: The ability to show the availability of animals for adoption must be possessed by administrators.

**4.2.4 Customer Management:**

**4.2.4.1 Description and Priority:**

On the Stray Animals website, administrators must be able to connect and communicate with users. Administrators can find areas for improvement by using the reviews and feedback that are gathered because of this interaction. Additionally, it guarantees the provision of essential user care and assistance, improving the platform's overall user experience.

**4.2.4.2 Stimulus/Response Sequences:**

The Stray Animals website will have a specific area for administrative evaluations and comments from customers. Requests for customer care will also be sent to the administrators' inbox, which will expedite correspondence and guarantee that customers' requirements are met quickly.

.

**4.2.4.3 Functional Requirements:**

REQ-1: Reviews and comments from customers must be visible to administrators.

REQ-2: Customers should be able to send direct messages to administrators with concerns and help requests.

REQ-3: The ability to view consumer queries is necessary for administrators.

REQ-4: The capacity to reply to client inquiries is a requirement for administrators.

REQ-5: Customer service inquiries must be handled by administrators.

# Other Nonfunctional Requirements

## Performance Requirements

* The system should have a fast response time.
* The load time for the user interface should take less than 2 seconds.
* Log in should be verified in five seconds.
* In five seconds or less, queries must deliver responses.
* Log out should take less than three seconds.
* System be able multiple numbers at the same time.
* System should be exact and accurate.
* There should be little inaccuracy in the system.
* A high workload should not be able to break the system.

## Safety Requirements

-All outgoing communications between clients and the data server of the system need to be encrypted.

-Access to different subsystems will be restricted by a user login screen that requests personal data, including an email address, name, and password, to maintain a secure environment.

-The system data is backed up every 24 hours as part of routine backups that are crucial to preventing data loss in the event of a system breakdown. Backup copies are kept offsite and in a secure location using files, CDs, flash drives, or hard disks as storage media.

-In order to guarantee the protection of the system's database and allow for speedy restoration in the event of an emergency or disaster, it is imperative to maintain a strong backup system.

## Security Requirements

Authentication and authorization: authentication and authorization are required for all users. Users’ access to features should be determined by their responsibilities.

Data security: users’ data must be kept secure, and precautions must be taken to prevent malevolent intruders from accessing it. Users’ data must be encrypted to safeguard data privacy. And database security must be assured.

Update and maintenance: the system must be updated and maintained on a regular basis. This reduces security weaknesses.

Dos protection: the application servers must be protected from Dos assaults. the application may become inoperable in the case of such an assault.

Preventing unauthorizes access: appropriate controls and blocking must be implemented to prevent unauthorized system access. This can keep malicious attackers from wreaking havoc on the app.

## Software Quality Attributes

Usability: the program should have an easy-to-use interface and be simple to utilize. The language and icons used must be easily understood by the applications intended audience.

Maintainability: the applications code should be modular and scalable, making it simple to maintain and update.

Readability: the application must be online 24 hours a day. Seven days a week with a minimum downtime pf 99%, it must be capable of recovering from unanticipated failures in a reasonable time.

Scalability: the program must be able to manage a high number of users and scale up or down depending on load.

Portability: the program shall be able to run on several platforms. Including android, IOS and web browsers

Testability: the application shall be built to be testable, with automated testing tools and procedures utilized throughout the development process to assure quality and eliminate faults.

Accessibility: the application must be usable by people with disabilities, such as those with vision or hearing problems, the application must adhere to accessibility guidelines such as WCAG 2.0 AA.

Interoperability: the application shall be able to interface with other systems and applications, allowing for safe and efficient data transmission*.*

## Business Rules

Animals' information registered in the system can only be updated by authorized personnel.

A sponsor donation must be deposited directly to the account of the stray animals that need help.

The system should be able to monitor sponsorship activities to keep track of and report on sponsors' assistance with stray animals.

Sponsors should be able to amend their profile information and cancel sponsorship for a specific stray animal through the system.

The system must guarantee that any payments supplied to stray animals' caregivers are correctly documented.

The system should get frequent input on the care services offered to stray animals by caregivers.

The system must guarantee that caregivers have the resources to address stray animals' health feeding, and other requirements.

The system must ensure that all information provided to the caregivers about stray animals must be private.

# Other Requirements

**Legal requirements**

The application must adhere to all applicable local, state, and federal laws and regulations.

**Security requirements**

To safeguard user data and prevent unwanted access, the application must have proper security features. This comprises user data storage security sensitive data encryption, and user authentication and permission procedures.

**Performance requirements**

The program must be built to be fast and efficient in responding to user queries. Response time must be rabid enough to avoid user irritation and application abandonment.

**Combability requirements**

To provide optimum accessibility and usefulness, the program must be compatible with a wide range of devices and operating systems. It should also be built to function with a variety of screen sizes, resolutions, and orientations.

**Readability requirements**

The program must always be dependable and accessible to users. It must be capable of handling heavy traffic volumes while avoiding crashes or outages. Additionally, the program must be structured to recover fast from ant faults and problems.

**Maintenance requirements**

The application must be developed in such a way that it can be readily maintained and upgraded. This involves creating an easy-to-use interface for managing content and user accounts.

Appendix A: Glossary

**OS-**Operating System

**API-**Application Program Interface

**GUI-**Graphical USER Interface

**HTML-**Hyper-Text Markup Language

**UI-**User Interface

**SRS-**Software Requirement Specification

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

The application’s technological infrastructure will be determined.

The application’s server and database capacity will be assessed.

The budget needed to complete the implementation will be established.

The languages that will be supported by the application will be determined.

The application user interface and design will be decided.

The application’s test plan and test cases will be determined.

The application’s user base and traffic capacity will be established.

The platform which the application will be made available will be selected.

The application update schedule will be decided.

The list will be updated as the project progresses.

**Use Case Tabular Descriptions:**

|  |  |
| --- | --- |
| **Use case-1** | Login |
| **Actor** | Admin |
| **Goal** | To access to the system. |
| **Preconditions** | 1-Has a valid username and password  2- Is not already logged in |
| **Stimulus** | The Admin requests access to the system. |
| **Scenario** | 1- The Admin enters his username and password  2: The system grants access to the user |
| **Exceptions** | * No internet * İnvalid username or password |

|  |  |
| --- | --- |
| **Use case-2** | Preview Pets |
| **Actor** | User |
| **Goal** | -To choose a pet for adopt  - Be a volunteer |
| **Preconditions** | 1-Internet  2- Click on preview pets from login page without login |
| **Stimulus** | The user get an access to the pets |
| **Scenario** | 1. The user click on preview pet from login page 2. Choose a pet to adopt or Be a volunteer |
| **Exceptions** | * No internet * Login as an admin . |

|  |  |
| --- | --- |
| **Use case-3** | Choose a cat |
| **Actor** | User |
| **Goal** | To adopt a pet |
| **Preconditions** | -Click on the photo of the cat in the preview pets page |
| **Stimulus** | The user wants to adopt a cat or just have informations |
| **Scenario** | 1. Click on preview pets from login page 2. Click on the photo of a cat in the preview pets page |
| **Exceptions** | * No internet * You logged in as an admin from login page * Click on Be a volunteer in preview pets page |

|  |  |
| --- | --- |
| **Use case-4** | Choose a dog |
| **Actor** | User |
| **Goal** | To adopt a pet |
| **Preconditions** | Click on the photo of the dog in the preview pets page |
| **Stimulus** | The user wants to adopt a dog or just have informations |
| **Scenario** | 1. Click on preview pets from login page 2. Click on the photo of a dog in the preview pets page |
| **Exceptions** | * No internet * You logged in as an admin from login page * Click on Be a volunteer in preview pets page |

|  |  |
| --- | --- |
| **Use case-5** | Live tracking |
| **Actor** | user |
| **Goal** | Locating the pet |
| **Preconditions** | -Click on Live tracking from login page |
| **Stimulus** | The user wants to check if the pet is near to him |
| **Scenario** | 1. Click on Live tracking from login page |
| **Exceptions** | - No internet  - The device doesnt have access to the GPS |

|  |  |
| --- | --- |
| **Use case-6** | Sign in page |
| **Actor** | User |
| **Goal** | To adopt the pet |
| **Preconditions** | -Click on the column of the pet that the user choose |
| **Stimulus** | The user wants to adopt after checking the pet and its live location |
| **Scenario** | 1-Click on preview pets from login page  2-Choose the pet ( dog or cats )  3-Check the pets information  4-Check the live location  5-Click on the column of the pet that u want to adopt |
| **Exceptions** | * No internet * Logged as an Admin * Clicked on Be a volunteer in preview pets page |

|  |  |
| --- | --- |
| **Use case-7** | Choose a cat |
| **Actor** | User |
| **Goal** | To adopt a pet |
| **Preconditions** | -Click on the photo of the cat in the preview pets page |
| **Stimulus** | The user wants to adopt a cat or just have informations |
| **Scenario** | 1. Click on preview pets from login page 2. Click on the photo of a cat in the preview pets page |
| **Exceptions** | * No internet * You logged in as an admin from login page * Click on Be a volunteer in preview pets page |

|  |  |
| --- | --- |
| **Use case-8** | Adoption application |
| **Actor** | User |
| **Goal** | To adopt a pet |
| **Preconditions** | -Sign in or login if you are an old user |
| **Stimulus** | The user wants to apply for adoption |
| **Scenario** | 1-Click on the pet that you choosed  2-Sign in as a user |
| **Exceptions** | * No internet * You logged in as an admin from login page * Click on Be a volunteer in preview pets page * You didnt sign in |

|  |  |
| --- | --- |
| **Use case-9** | Waiting for approval |
| **Actor** | User |
| **Goal** | To receive the pet that you want to adopt |
| **Preconditions** | -Sign in as a user  -Adopt a pet |
| **Stimulus** | The user adopt a pet |
| **Scenario** | 1-Click on adopt in the adoption application page |
| **Exceptions** | * No internet * You didnt adopt any pet * You didnt sign in |

|  |  |
| --- | --- |
| **Use case-10** | Profile |
| **Actor** | User |
| **Goal** | To modify information |
| **Preconditions** | -Click on profile from addoption application |
| **Stimulus** | The user wants to check or modify his own information |
| **Scenario** | Click on profile from Adoption application page |
| **Exceptions** | * No internet * You didnt sign in |

|  |  |
| --- | --- |
| **Use case-11** | Edit/remove the pet |
| **Actor** | Admin |
| **Goal** | To make changes about the pets |
| **Preconditions** | * Logged as an Admin * Choose the kind of pets you want to make the changes |
| **Stimulus** | The admin wants to remove or edit informations |
| **Scenario** | 1-Logged as an Admin  2-Choose the category of the pet  3-Click on edit or remove |
| **Exceptions** | * No internet * You couldn’t log in as an Admin |

|  |  |
| --- | --- |
| **Use case-12** | Add a pet |
| **Actor** | Admin |
| **Goal** | To Add a pet |
| **Preconditions** | * Logged as an Admin * Choose the kind of pets you want to make the changes * Click on Add in Edit/remove page |
| **Stimulus** | The Admin wants to add a pet |
| **Scenario** | 1-Logged as an Admin  2-Choose the category of the pet  3-Click on Add |
| **Exceptions** | * No internet * You couldn’t log in as an Admin |

|  |  |
| --- | --- |
| **Use case-13** | Adoption Application |
| **Actor** | Admin |
| **Goal** | To check adoption orders |
| **Preconditions** | * Logged as an Admin * Receive a notification from the user wants to adopt |
| **Stimulus** | The admin has two options decline or accept the adoption application |
| **Scenario** | 1-Logged as an Admin  2-Receive a notification from the user for adopting a pet  3-Admin can decline or accept the adoption application |
| **Exceptions** | * No internet * You couldn’t log in as an Admin |

|  |  |
| --- | --- |
| **Use case-14** | Online Volunteer |
| **Actor** | Admin |
| **Goal** | To see the online volunteer |
| **Preconditions** | * Logged as an Admin * Accept an adoption application |
| **Stimulus** | The admin wants to send the adoption order to the volunteer |
| **Scenario** | 1-Logged as an Admin  2-Accept an addoption application |
| **Exceptions** | * No internet * You couldn’t log in as an Admin * Admin declin an adoption application |

|  |  |
| --- | --- |
| **Use case-15** | Sign in as a volunteer |
| **Actor** | Volunteer |
| **Goal** | To register or login ( old volunteer) |
| **Preconditions** | * Click on Be volunteer in the categories page * Register or sign in directly if you are an old volunteer |
| **Stimulus** |  |
| **Scenario** | 1-Click on Be volunteer in the category page  2-Register or sign in |
| **Exceptions** | * No internet * You logged as an Admin or user |

|  |  |
| --- | --- |
| **Use case-16** | Waiting for volunteering |
| **Actor** | Volunteer |
| **Goal** | To take the pet to the user |
| **Preconditions** | * Logged as an volunteer * Get a notification from the admin |
| **Stimulus** |  |
| **Scenario** | 1-Logged as an volunteer  2-Can decline or accept the adoption application |
| **Exceptions** | * No internet * You couldn’t log in as an volunteer |

|  |  |
| --- | --- |
| **Use case-17** | Profile |
| **Actor** | Volunteer |
| **Goal** | To modify information |
| **Preconditions** | * Logged as an Admin * Click on profile from waiting for volunteering page |
| **Stimulus** | The user wants to check or modify his own information |
| **Scenario** | Click on profile from Adoption application page |
| **Exceptions** | * No internet * You didnt log in as a volunteer |

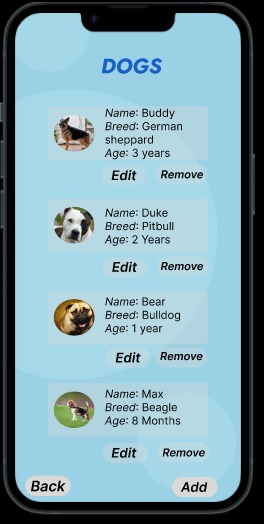
**User-Interface:**

**For admin:**

**A phone with a logo on it

Description automatically generatedA black cell phone with a cat and a blue background

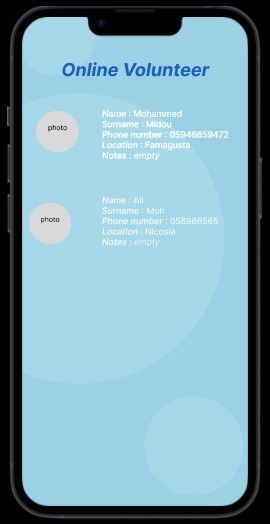
Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screen shot of a cell phone

Description automatically generatedA screenshot of a phone

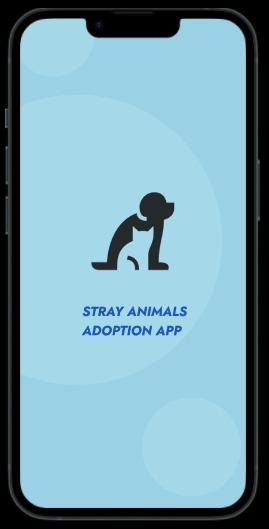
Description automatically generatedA screen shot of a phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated**

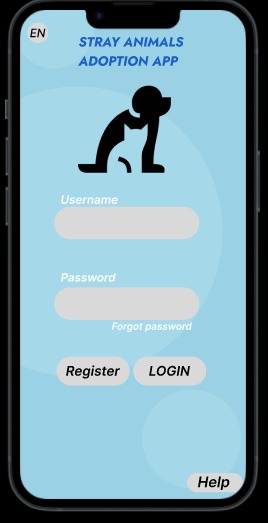
**For User:**

**A black cell phone with a cat and a blue background

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screen shot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screen shot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated**

**For Volunteers:**

**A phone with a logo on it

Description automatically generatedA black cell phone with a cat logo on it

Description automatically generatedA screen shot of a cell phone

Description automatically generatedA cell phone with a map on it

Description automatically generatedA screenshot of a phone

Description automatically generated**

**UML Diagrams:**

**1. Activity diagram:**

A screenshot of a computer

Description automatically generated

**Use case Diagram:**

A diagram of a computer network

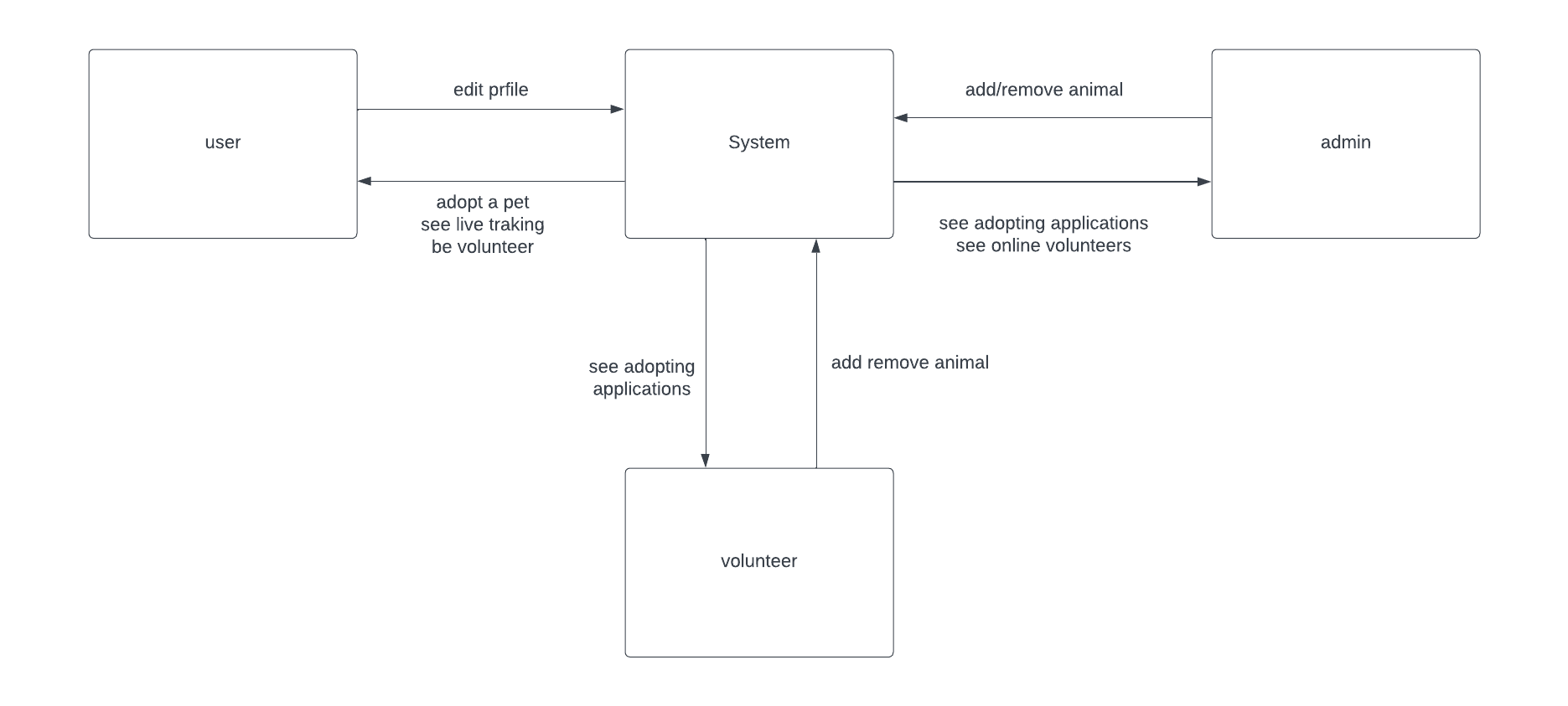
Description automatically generated

**Sequence diagram:**

A diagram of a software application

Description automatically generated with medium confidence

**Context Diagram:**



**Class diagram:**

A diagram of a computer

Description automatically generated

**BPMN:**

A diagram of a software system

Description automatically generated

Unclear above actors:

A diagram of a login successful

Description automatically generated

A diagram of a person

Description automatically generated