



Smart Home Pet Care

Software Requirement Specification

2022. 05. 01.

Introduction to Software Engineering 42
Team 2

| | |
|-------------|-----|
| Team Leader | 차현묵 |
| Team Member | 곽재원 |
| Team Member | 김준식 |
| Team Member | 양승찬 |
| Team Member | 유새하 |
| Team Member | 이경돈 |
| Team Member | 최하늘 |

| | |
|---|-----------|
| 1. Introduction | 8 |
| 1.1. Purpose | 8 |
| 1.2. Scope | 8 |
| 1.3. Definition, Acronyms, and Abbreviation | 9 |
| 1.4. References | 10 |
| 1.5. Overview | 10 |
| 2. Overall Description | 10 |
| 2.1. Product Perspective | 10 |
| 2.1.1. System Interface | 11 |
| 2.1.2. User Interface | 11 |
| 2.1.3. Hardware Interface | 11 |
| 2.1.4. Software Interface | 12 |
| 2.1.5. Communication Interface | 12 |
| 2.1.6. Memory Constraints | 12 |
| 2.1.7. Operations | 12 |
| 2.2 Product Functions | 14 |
| 2.2.1. 계정 관리 | 14 |
| 2.2.2. IoT 디바이스 등록 및 설정 | 14 |
| 2.2.3. 반려동물 이상행동 앱 푸시 알림 | 14 |
| 2.2.4. 반려동물 실시간 위치 파악 | 14 |
| 2.2.5. 반려동물 행동 로그 | 15 |
| 2.2.6. 동물의 건강상태 획득/분석 | 15 |
| 2.2.7. 반려동물 맞춤 케어 시스템 설정 | 15 |
| 2.3. User Characteristics | 15 |
| 2.4 Constraints | 15 |

| Smart Home Pet Care | Requirements Specification |
|--------------------------------------|-----------------------------------|
| 2.5. Assumptions and Dependencies | 16 |
| 3. Specific Requirements | 17 |
| 3.1. External Interface Requirements | 17 |
| 3.1.1. User Interfaces | 17 |
| 3.1.2. Hardware Interfaces | 28 |
| 3.1.3. Software Interfaces | 28 |
| 3.1.4. Communication Interfaces | 29 |
| 3.2. Functional Requirements | 35 |
| 3.2.1. Use Case | 35 |
| 3.2.2. Use Case Diagram | 41 |
| 3.2.3. Data Dictionary | 42 |
| 3.2.4. Data Flow Diagram | 45 |
| 3.3. Product Requirements | 46 |
| 3.3.1. Performance Requirements | 46 |
| 3.3.2. Space Requirements | 46 |
| 3.3.3. Security Requirements | 46 |
| 3.3.4. Usability Requirements | 47 |
| 3.3.5. Dependability Requirements | 47 |
| 3.4. Organizational Requirements | 47 |
| 3.4.1. Environmental Requirements | 47 |
| 3.4.2. Operational Requirements | 47 |
| 3.4.3. Development Requirements | 48 |
| 3.5. External Requirements | 48 |
| 3.5.1. Regulatory Requirements | 48 |
| 3.5.2. Ethical Requirements | 48 |
| 3.5.3. Accounting Requirements | 48 |

| Smart Home Pet Care | Requirements Specification |
|----------------------------|-----------------------------------|
|----------------------------|-----------------------------------|

| | |
|--|-----------|
| 3.5.4. Safety Security Requirements | 48 |
| 3.6. Organizing the Specific Requirements | 50 |
| 3.6.1. Context Model | 50 |
| 3.6.2. Process Model | 50 |
| 3.6.3. Interaction Model | 56 |
| 3.6.4. Behavior Model | 57 |
| 3.6.4.1. Data Flow Diagram | 57 |
| 3.6.4.2. Sequence Diagram | 58 |
| 3.7. System Architecture | 61 |
| 3.8. System Evolution | 61 |
| 3.8.1. Limitation and Assumptions | 61 |
| 3.8.2. Evolutions of Hardware and Software | 61 |
| 4. Supporting Information | 62 |
| 4.1. Software Requirement Information | 62 |
| 4.2. Document History | 62 |

표 목차

| | |
|--|----|
| 표 1. 프로젝트 약어 정의 | 9 |
| 표 2. 프로젝트 용어 정의 | 9 |
| 표 3. User Interface - 사용자 계정 관리 | 17 |
| 표 4. User Interface - IoT 디바이스 등록 및 설정 | 18 |
| 표 5. User Interface - 반려동물 이상행동 푸시알림 | 19 |
| 표 6. User Interface - 반려동물 실시간 위치 파악 | 20 |
| 표 7. User Interface - 반려동물 행동 로그 – 생성 | 21 |
| 표 8. User Interface - 반려동물 행동 로그 – 조회 | 22 |
| 표 9. User Interface - 반려동물 행동 로그 – 삭제 | 23 |
| 표 10. User Interface - 반려동물 건강체크 | 24 |
| 표 11. User Interface - 펫케어 시스템 - 메인 페이지 | 25 |
| 표 12. User Interface - 펫 케어 시스템 - 설정 페이지 | 26 |
| 표 13. Hardware Interfaces - 시스템에 사용 가능한 기기 | 28 |
| 표 14. Software Interfaces - MariaDB Server | 28 |
| 표 15. Communication Interfaces - 로그인 | 29 |
| 표 16. Communication Interfaces - 반려동물 로그 | 30 |
| 표 17. Communication Interfaces - 반려동물 촬영 영상 | 31 |
| 표 18. Communication Interfaces - IoT 작동/중지 | 32 |
| 표 19. Communication Interfaces - IoT 설정 변경 | 33 |
| 표 20. Communication Interfaces - 반려동물 건강정보 요청 및 업데이트 | 34 |
| 표 21. Use case - 사용자 계정 관리 | 35 |
| 표 22. Use case - 반려동물 이상행동 앱푸시 알림 | 36 |
| 표 23. Use case - 실시간 위치 파악 | 36 |
| 표 24. Use case - 반려동물 행동 로그 조회 | 37 |
| 표 25. Use case - 반려동물 건강 체크 | 37 |

| | |
|--|----|
| 표 26. Use case - IoT 작동 / 중지 | 38 |
| 표 27. Use case - IoT 설정 변경 | 39 |
| 표 28. Use case - IoT 디바이스 추가/제거 | 39 |
| 표 29. Data Dictionary - 사용자 계정 관리 | 42 |
| 표 30. Data Dictionary - 실시간 위치 모니터링 | 42 |
| 표 31. Data Dictionary - 반려동물 이상행동 감지 | 43 |
| 표 32. Data Dictionary - IoT 디바이스 등록 및 설정 | 43 |
| 표 33. Data Dictionary - IoT 디바이스 기능 제어 | 43 |
| 표 34. Data Dictionary - IoT 디바이스 작동 시간대 설정 | 44 |

그림 목차

| | |
|--|----|
| 그림 1. Use Case Diagram | 41 |
| 그림 2. Data Flow Diagram | 45 |
| 그림 3. Context Model | 50 |
| 그림 4. Process Model - 총괄 | 51 |
| 그림 5. Process Model - 계정관리 | 52 |
| 그림 6. Process Model - IoT 디바이스 관리 | 53 |
| 그림 7. Process Model - 반려동물 건강정보 수집/분석 | 54 |
| 그림 8. Process Model - 반려동물 행동 로그 기록 | 55 |
| 그림 9. Process Model - 반려동물 실시간 위치 파악 | 56 |
| 그림 10. Process Model - 반려동물 이상행동 감지 | 56 |
| 그림 11. Use Case Diagram | 57 |
| 그림 12. Data Flow Diagram | 58 |
| 그림 13. Sequence Diagram - 계정관리 | 59 |
| 그림 14. Sequence Diagram - 이상행동 Push 알림 | 59 |
| 그림 15. Sequence Diagram - 반려동물 위치 정보 | 60 |
| 그림 16. Sequence Diagram - 반려동물 로그 조회 | 60 |
| 그림 17. Sequence Diagram - IoT 디바이스 관리 | 61 |
| 그림 18. Sequence Diagram - 반려동물 건강상태 관리 | 61 |
| 그림 19. System Architecture | 62 |

1. Introduction

1.1. Purpose

이 문서는 반려동물 원격 케어 어플리케이션인 Smart Home Pet Care의 요구사항 명세서이다. Smart Home Pet Care는 1인 가구와 반려동물을 기르는 가구의 증가로 인해 집에 혼자 있게 되는 반려동물을 원격으로 돌보기 위한 어플리케이션이다. 이 어플리케이션은 성균관대학교 소프트웨어공학개론 수업의 2팀에 의해서 고안되었다. 본 명세서에서는 Smart Home Pet Care의 요구사항, 제약조건이 기술되어 있으며, Smart Home Pet Care의 디자인과 구현은 본 명세서의 내용을 따라 진행될 것이다. 본 명세서는 2팀의 프로젝트 진행과정에 활용될 것을 전제로 작성되었고 그에 따라 2팀을 주요한 독자로 상정하여 작성하였다. 또한 본 명세서는 소프트웨어 공학개론의 교수, 조교, 학생들이 열람한다는 전제하에 작성하였다. 기존의 어플리케이션과 달리, Smart Home Pet Care는 어플리케이션을 위해서만 만들어진 IoT 기기가 아닌 기존의 IoT 기기를 활용하여 반려동물을 돌볼 수 있고, 더 유기적인 돌봄 시스템을 제공하여 반려동물과 사용자의 만족감을 높이고자 한다.

1.2. Scope

Smart Home Pet Care는 사용자가 원격으로 반려동물을 돌보고 모니터링할 수 있는 시스템을 제공하고자 한다. 사용자는 반려동물을 돌보는 데 사용할 IoT 기기를 등록할 수 있다. 사용자는 반려동물에 부착된 IoT 기기를 통해 반려동물의 생체 신호를 모니터링 할 수 있다. 생체 신호에 문제가 발생한 경우 사용자에게 PUSH 알림을 제공하여 동물의 갑작스러운 건강 악화를 대비할 수 있다. 또한 반려동물의 행동을 모니터링하여 갑작스러운 짖음, 가구를 물어뜯는 행동 등의 이상 행동을 보이는지 확인할 수 있다. 이상 행동을 보이는 경우 이는 모두 로그를 남겨 DB에 저장된다. 사용자가 반려동물이 현재 집 안에서의 위치를 확인하고 싶은 경우 이를 실시간으로 확인할 수 있다. 사용자는 등록된 IoT 기기를 조작하여 먹이 제공, 집 안 청소, 온도 관리, 공기 관리를 통해 반려동물을 위한 환경을 지속적으로 제공할 수 있다. 이러한 시스템 제공을 통해 2팀은 기존의 IoT 기기를 활용하면서 집안의 환경을 더욱 유기적으로 조정하여 반려동물과 사용자에게 더욱 좋은 집안 환경을 제공하고자 한다.

1.3. Definition, Acronyms, and Abbreviation

표 1. 프로젝트 약어 정의

| 약어 | 설명 |
|------|-----------------------------------|
| RAM | Random Access Memory |
| DB | Database |
| UI | User Interface |
| HTTP | Hypertext Transfer Protocol |
| OS | Operating System |
| API | Application Programming Interface |
| IoT | Internet of Things |
| JSON | JavaScript Object Notation |
| AI | Artificial Intelligence |

표 2. 프로젝트 용어 정의

| 용어 | 정의 |
|------|---|
| 사용자 | 시스템을 사용하는 자 |
| 반려동물 | 사용자가 돌보는 동물 |
| 모니터링 | 반려 동물의 상태, 위치 등을 확인 하는 것을 의미하는 단어이다. |
| 네트워크 | 각각의 기기들을 연결하여 정보를 전송/수신할 수 있도록 하는 망. 이 문서에서는 인터넷을 의미한다. |
| 서버 | 시스템의 중앙에서 네트워크를 활용해 사용자에게 서비스를 제공하는 컴퓨터 또는 컴퓨터 프로그램을 의미하는 단어이다. |

1.4. References

- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, In IEEEXplore Digital Library. (https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc)
- Team 1. "Project Highlight Requirement". SKKU, Last Modified: April 25. 2021. (https://github.com/skkuse/2021spring_41class_team1/blob/main/doc/Project%20HighlightRequirement%202.0.4.pdf)
- Flutter API. (<https://github.com/PDFTron/pdftron-flutter>)

1.5. Overview

본 명세서에서는 Smart Home Pet Care의 전체적 기술 사항, 이해관계와 시스템 간의 인터페이스, 상세한 시스템 요구 사항의 3개의 챕터로 기술한다.

두 번째 챕터에서는 Smart Home Pet Care의 이해관계자가 가진 요구사항에 대해서 기술한다. 또한 이 장은 Smart Home Pet Care가 가진 시스템간의 인터페이스에 대해서 기술한다. 그리고 시스템이 제공하는 기능 요구사항과 비기능 요구사항에 대해 기술한다. 시스템을 구동하기 위한 사양과 시스템이 가지고 있는 제약조건에 대해서 서술한다.

세 번째 챕터에서는 Smart Home Pet Care의 요구 사항을 보다 상세히 기술한다. 각각의 서비스가 제공해야 하는 요구사항과 특징에 대해 자세히 기술한다. 시스템의 인터페이스에 대해서 보다 자세히 기술한다. 또한 시스템의 데이터베이스 아키텍쳐에 대해 기술하고 시스템의 개선이 필요한 부분에 대해서 기술한다.

부록에서는 본 명세서를 작성하는데 사용된 정보들의 출처와 본 문서의 수정 내역에 대해 기술한다.

2. Overall Description

2.1. Product Perspective

Smart Home Pet Care는 이용자들이 반려동물을 기르는 것에 있어서 편리성을 도모하고, 기존에 존재하던 불편함을 최소화할 수 있도록 도안 되었다. 본 시스템인 Smart Home Pet Care는

반려동물과 가정의 상태를 실시간으로 확인하고, 상황에 맞는 동작을 자동으로 수행하게 하거나 이용자가 직접 원격으로 명령을 지시함으로써 발생한 상황에 맞는 적절한 동작이 효과적으로 수행될 수 있도록 하는 것을 주요 목표로 하고 있다. 이러한 일련의 과정들을 효과적으로 작동시키기 위해서 기능을 제공하는 디바이스들은 IoT 기술을 이용해 연결되어 있다. 이렇게 연결된 있는 디바이스들은 상호 유기적으로 동작할 수 있게 됨으로써, 보다 효율적인 작업을 수행할 수 있게 될 것이다. 궁극적으로, 반려동물과 반려인 모두가 만족할 수 있는 최선의 환경을 제공하는 서비스를 구현하는 것이 목표이다.

2.1.1. System Interface

IoT 디바이스의 설정 정보 및 디바이스로부터 획득되는 모든 정보들은 기본적으로 서버에 저장된다. 이 정보들은 모두 암호화된 뒤 서버에 저장된다. IoT 디바이스의 경우 Wi-Fi에 연결되어 실시간으로 정보를 서버에 업로드하고, 이용자는 이 서버에 접근해 정보를 확인할 수 있다.

2.1.2. User Interface

UI는 모바일 기기의 화면을 통해 출력된다. 조작은 기기에 내장된 터치 기능과 일부 기능의 경우에는 키보드 입력이 필요할 수도 있으며 이는 가상 키보드 사용으로 대체가 가능하다.

모든 사용자는 계정을 이용해 로그인해야 시스템을 사용할 수 있게 된다. 사용자가 로그인을 수행하면 모니터링, 건강 체크, 펫 케어, 더 보기 등의 대분류를 통해 원하는 작업의 범주를 선택할 수 있다.

모니터링에서는 현재 반려동물과 가정의 상태를 확인할 수 있다. 이상 상황이 발생할 경우 발송되는 PUSH 로그를 확인하는 것뿐만 아니라, 사용자가 가정에 배치된 IoT 디바이스들을 이용해 실시간으로 관련된 정보를 요청해 제공받을 수도 있다.

건강 체크에서는 반려동물에게 장착된 웨어러블 디바이스로부터 획득한 정보를 이용해 표시하게 된다. 사용자는 외부에서도 자신 반려동물의 건강 상태를 실시간으로 확인할 수 있다.

펫 케어에서는 사용자가 지시할 수 있는 일종의 명령들이 나타나고, 이를 이용해 원격에서도 가정 및 반려동물의 상태를 제어할 수 있게 된다.

마지막으로 더 보기에서는 펫 케어와는 관련성이 적지만 필수적인 환경설정과 같은 기능을 배치함으로써, 전반적인 UI의 간결함과 확장성을 모두 달성할 수 있도록 했다.

2.1.3. Hardware Interface

Smart Home Pet Care는 Android 및 iOS 기반 모바일 기기를 대상으로 서비스를 제공하는 것을 목표로 하고 있다. 이에 해당하는 모바일 기기들은 최소 1.5GB 이상의 RAM 용량과 1.4GHz 이상의 AP를 탑재하고 있어야 한다.

2.1.4. Software Interface

Smart Home Pet Care는 현재 널리 사용되고 있는 Android 9 Pie 또는 iOS 14 이상을 요구한다.

메인 타깃은 Android 11 및 iOS 15 이상의 OS가 작동 중인 모바일 디바이스로, 해당 OS에서 어플리케이션 테스트가 이루어졌다.

2.1.5. Communication Interface

사용자의 기기와 서버는 HTTP 프로토콜을 이용해 통신을 수행한다.

이때, 수치 데이터와 같은 내용들은 JSON 포맷으로 통신이 이루어지며, 영상 데이터의 경우에는 HTTP Live Streaming 기술을 이용해 MPEG 포맷으로 통신이 이루어진다.

2.1.6. Memory Constraints

Smart Home Pet Care는 지속적으로 백그라운드에서 어플리케이션이 실행되며, 사용자에게 PUSH 알림을 전송하는 등의 작업을 수행하므로 최소 3GB 이상 RAM 용량을 요구한다.

별도로, 어플리케이션의 설치와 실행을 위해서 최소 500MB 이상의 내부 스토리지 용량을 요구한다.

2.1.7. Operations

- 로그인

사용자는 모바일 디바이스에 설치된 Smart Home Pet Care 어플리케이션을 이용해 서비스에 로그인할 수 있다.

- 계정 관리

사용자는 로그인 이후, 자신이 로그인한 계정에 대한 관리를 진행할 수 있다. 보다 로그인 편의를 위한 모바일 디바이스가 지원하는 보안 방식으로의 자동 로그인 진행 또는 Google과 같은 외부 계정 연동을 통한 쉬운 로그인 기능을 설정할 수 있다.

- IoT 디바이스

사용자는 IoT 디바이스의 설정과, 이에 따른 기능의 추가/제거를 수행할 수 있다.

IoT 디바이스의 연결을 어플리케이션 화면에 표시함으로써, 현재 사용 중인 IoT 디바이스의 목록을 확인하고 디바이스의 추가 연결 또는 현재 연결 중인 디바이스와의 연결 끊기를 설정할 수 있다.

IoT 디바이스의 연결 추가/감소에 따라서 사용할 수 있는 기능의 증가/감소가 발생할 수 있다. 디바이스의 연결 변화가 감지될 때마다 어플리케이션은 사용자가 사용 가능한 서비스의 목록을 제공함으로써 사용자는 필요한 서비스만을 취사선택할 수도 있다.

- PUSH 알림

어플리케이션이 백그라운드에서 상주하면서, 서버로부터 지속적으로 정보를 받는다. IoT 디바이스에서 정기적인 신호 혹은 긴급 신호가 감지되면 이는 서버를 거쳐 사용자에게 전달되는데, PUSH 알림을 통해 즉각적으로 확인할 수 있도록 한다.

- 펫 케어

건강 체크, 모니터링 등의 기능들을 구현하면서 수신되는 정보들을 종합적으로 이용해 반려동물을 돌보거나, 현재 가정의 상태를 제어하는 기능을 수행할 수 있다. 사용자는 이 기능을 수행하기 위해 루틴을 이용해 특정 시간에 맞추어 먹이를 제공하는 것과 같은 작업을 수행하도록 지시할 수도 있으며, 어플리케이션을 실행해 즉각적으로 임의의 작업을 수행하도록 지시할 수도 있다. 연결된 IoT 디바이스가 지원한다면 모든 작업은 예약 실행과 즉각 실행이 모두 가능하다.

- 건강 체크

반려동물에게 장착한 웨어러블 IoT 디바이스를 이용해 실시간으로 건강 정보를 확인할 수 있다. 정기적인 정보들은 실시간으로 서버로 전송되므로 사용자는 어플리케이션을 통해 원격에서도 건강 정보를 확인할 수 있다. 만약 사용자가 긴급하게 인지해야 하는 반려동물의 건강 문제가 발생할 경우에는 PUSH 알림을 병행해 알림을 빠르게 전달함으로써 더 신속한 대처가 이루어질 수 있도록 한다.

- 모니터링

반려동물에게 장착한 웨어러블 IoT 디바이스 및 집안에 배치된 타 IoT 디바이스들로 부터 실시간으로 받은 정보를 어플리케이션을 이용해 제공받음으로써, 현재 집에 없더라도 집안의 상황을

파악할 수 있도록 하는 기능을 사용자에게 제공한다. 만약 사용자가 긴급하게 인지해야 하는 상황이라고 판단될 경우에는, 모니터링과 PUSH 알림을 병행함으로서 보다 신속한 대처가 가능하여지도록 한다.

2.2 Product Functions

2.2.1. 계정 관리

펫케어 어플리케이션을 이용하고자 하는 사용자들이 앱 실행 후 맨 처음 보게 되는 화면이다. 기존 이용자의 경우 DB 서버에 등록되어있는 본인의 이메일 주소와 패스워드를 이용해 로그인할 수 있으며, ID/비밀번호 분실 시 본인확인을 통해 찾을 수 있는 기능이 있으며, 신규 사용자의 경우 몇 가지 정보들을 기입하여 본인의 계정을 생성할 수 있는 회원가입 기능이 구현되어 있다.

2.2.2. IoT 디바이스 등록 및 설정

펫케어 시스템에서 제공하는 기능들을 구현하기 위해, 시스템에 포함된 IoT 디바이스들을 관리하는 기능이다. 여기에서 언급되는 IoT 디바이스는 돌봄 로봇, 로봇 청소기, 반려동물을 위한 웨어러블 디바이스를 포함한다.

이용자는 자신이 원하는 기능을 사용하기 위해서 IoT 디바이스를 시스템에 등록하고, 이후에 해당 디바이스가 제공하는 기능을 사용할 수 있게 된다. 모든 IoT 디바이스를 등록하는 것이 아니라, 사용자는 자신이 원하는 기능을 제공하는 IoT 디바이스만을 취사선택해 시스템에 등록, 사용할 수 있다.

2.2.3. 반려동물 이상행동 앱 푸시 알림

돌봄 로봇이나 로봇 청소기 등 IoT 디바이스에 장착된 카메라로 촬영되는 반려동물의 실시간 영상에서 AI로 이상 행동이 감지되면 서버 요청에 따라 사용자에게 즉시 앱 푸시 알림이 전송된다. 여기서 이상 행동이란 반려동물이 일정 시간 이상 행동을 반복하거나, 혹은 움직임이 없을 경우 등을 말한다.

2.2.4. 반려동물 실시간 위치 파악

반려동물에 장착된 IoT 디바이스로부터 반려동물의 위치를 파악해서 어플리케이션 화면에 지도 형태로 실시간 위치를 표시한다. 사용자는 어플리케이션 위치 확인 메뉴를 통해 표시된 위치를 확인할 수 있으며 설정해 둔 위치를 벗어난다면 사용자에게 앱 푸시 알림을 전송하여 외출 시 집 내부에 반려동물이 있는지 확인할 수 있다.

2.2.5. 반려동물 행동 로그

사용자는 본인이 확인하지 못한 시간대의 반려동물 행동을 로그 기능을 통해 열람할 수 있다. 로그 기능은 반려동물 이상행동, 위치, 맞춤 케어 시스템의 장비들의 작동 여부 등의 사항을 발생 시간과 발생 위치를 함께 저장한다. 이상행동 등의 영상 정보가 같이 생성되는 로그의 경우 1분 이내의 영상을 함께 서버에 저장하여 사용자가 확인할 수 있도록 한다. 저장공간 확보를 위해 로그는 발생 시간으로부터 1개월만 보관한다.

2.2.6. 동물의 건강 상태 획득/분석

반려동물에게 부착되어 있는 스마트 디바이스가 맥박수, 산소포화도, 체온, 스트레스 수 등을 측정하여 사용자에게 이들의 수치화된 데이터와 이를 바탕으로 해석된 결과들을 제공받는다. 이러한 정보를 애완동물별 프로필 페이지로 묶어 제공함으로써 사용자에게 각 애완동물의 건강 상태를 직관적으로 확인할 수 있으며, 필요할 경우 사용자가 실시간으로 갱신된 건강 정보를 확인해 볼 수도 있다. 특정 지표가 비정상적으로 측정이 될 경우 필요한 조치사항을 사용자에게 제공한다.

2.2.7. 반려동물 맞춤 케어 시스템 설정

사용자는 등록된 IoT 디바이스를 조작하여 반려동물을 원격으로 케어할 수 있다. 사용자가 조작할 수 있는 IoT 디바이스로는 먹이 공급 시스템, 청소 로봇, 놀이 로봇, 온도 관리 시스템, 공기 관리 시스템이 있다. 사용자는 각각의 IoT 디바이스가 현재 어떤 상태인지 확인할 수 있다. 사용자는 각각의 IoT 기기의 작동 시간을 직접 설정하거나, IoT 디바이스가 자동으로 작동하게 하여 반려동물을 돌볼 수 있다.

2.3. User Characteristics

사용자는 반려동물을 기르고 있는 사람으로, 집을 비우는 경우가 많아 반려동물이 혼자 집에 있는 시간이 많다. 또한 휴대할 수 있는 전자 기기를 소지하고 있어야 한다.

2.4 Constraints

SRS에서 작성된 내용을 바탕으로 어플리케이션을 구현한다. 또한 어플리케이션 개발 시 다음 조건을 만족하도록 한다.

- Android 9.0 (API 26) 이상의 OS 버전에서 동작 가능
- IOS 14.0 이상의 OS 버전에서 동작 가능
- Android 11.0에서 어플리케이션 테스트
- IOS 15.0에서 어플리케이션 테스트
- 향후 관리를 위해 주석 사용 / 코드 최적화 작업

2.5. Assumptions and Dependencies

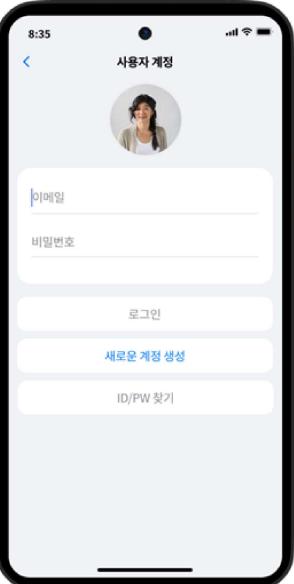
펫 케어 어플리케이션은 최소 Android 9.0 (API 26) 이상의 OS 버전, IOS 14.0 이상의 OS 버전을 사용하는 휴대 기기에서 사용되어야 한다. IOT 기기는 서버에 연결된 상태여야 한다.

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

표 3. User Interface - 사용자 계정 관리

| Name | 기능 제목 |
|---------------------------------------|---|
| Description of purpose | 사용자의 이메일 주소, 비밀번호를 이용하여 로그인. ID/비밀번호 찾기 기능과 신규 이용자의 경우 회원가입 기능 구현 |
| Input source / Output destination | 이메일 주소, 비밀번호 / 로그인 DB 서버 |
| Valid range / Accuracy / Tolerance | 사용자 정보가 담긴 DB 서버 |
| Unit of measure | 스크린 |
| Timing | N/A |
| Relationship with other input/outputs | N/A |
| Format and configuration of screen |  <p>사용자는 이메일과 주소를 이용하여 로그인 할 수 있다. 잘못된 정보 입력시, 오류 메시지가 팝업된다. 계정이 없는 경우, 회원가입 버튼을 통해 새로운 계정 생성이 가능하다. 만약 ID/PW가 기억나지 않을 경우에도 찾을 수 있는 기능이 구현되어 있다.</p> |
| Format and configuration of window | N/A |
| Data formats | Text, int |
| Command formats | Query |

| | |
|-------------|---------|
| End message | 해당사항 없음 |
|-------------|---------|

표 4. User Interface - IoT 디바이스 등록 및 설정

| Name | 기능 제목 |
|---------------------------------------|---|
| Description of purpose | IoT 디바이스를 시스템에 등록하고, 기능을 활성화/비활성화 하는 작업을 수행한다. |
| Input source / Output destination | Android OS 또는 iOS가 탑재된 사용자 장치를 이용한 사용자 요청 / 데이터베이스 |
| Valid range / Accuracy / Tolerance | IoT 디바이스 기준, 가정 내부에 설치된 Wi-Fi AP의 전파가 도달 가능하며, 안정적인 통신이 가능한 범위 |
| Unit of measure | 화면 |
| Timing | 사용자 요청 시 |
| Relationship with other input/outputs | 해당 기능을 이용해 구성된 IoT 디바이스의 연결과 기능 정보를 이용해, 시스템이 제공해야 할 펫케어 시스템의 기능이 결정됨 |
| Format and configuration of screen |  <p>현재 연결되어 있거나, 연결된 기기들이 화면에 표시된다. 연결 기록이 존재하는 디바이스들의 경우에는 “나의 기기”로 표시되어 손쉽게 연결/해제가 가능하며, 이 외에 “기타 기기”들로 추가적으로 연결이 가능한 디바이스들도 표시된다.</p> |
| Format and configuration of window | N/A |
| Data formats | Text, int |
| Command formats | Query |

| | |
|-------------|---------|
| End message | 해당사항 없음 |
|-------------|---------|

표 5. User Interface - 반려동물 이상행동 푸시알림

| Name | 반려동물 이상행동 푸시알림 |
|---------------------------------------|--|
| Description of purpose | 반려동물을 촬영한 실시간 영상을 AI로 분석하여 이상 행동이 감지되면 사용자에게 즉시 앱 푸시 알림이 전송된다. |
| Input source / Output destination | 서버 요청 / Android OS 또는 iOS가 탑재된 사용자 장치 |
| Valid range / Accuracy / Tolerance | 반려동물 이상 행동 판별 기준에 따른 범위 / 반려동물 행동 로그 데이터와 AI 성능에 따른 정확도 / AI 정확도 오차 한계 |
| Unit of measure | 앱 푸시 |
| Timing | 동기적 서버 요청 / 서버 요청의 즉각적인 실행 |
| Relationship with other input/outputs | AI의 이상 행동 탐지 결과에 따른 서버 요청을 입력으로 수신 |
| Format and configuration of screen |  <p>운영체제에 따른 앱 푸시 알림의 형태로, 텍스트 형태의 제목과 내용 및 알림 수신 시각, 이미지 형태의 배지 아이콘으로 구성된다. 알림의 삭제와 더보기 등의 구성은 운영체제의 기본 기능에 따른다.</p> |
| Format and configuration of window | N/A |
| Data formats | json, text |

| | |
|-----------------|-----|
| Command formats | N/A |
| End message | N/A |

표 6. User Interface - 반려동물 실시간 위치 파악

| Name | 실시간 위치 파악 |
|---------------------------------------|--|
| Description of purpose | 반려 동물의 위치를 실시간으로 파악하고, 사용자한테 지도 형태로 표시한다. |
| Input source / Output destination | 사용자 요청 & 서버 요청 / Android, IOS가 탑재된 사용자 기기 |
| Valid range / Accuracy / Tolerance | 사용자가 설정한 위치에 따른 범위 / GPS 송수신의 정확도 / GPS 송수신의 오차 |
| Unit of measure | 지도 화면 / 앱 푸시 |
| Timing | 위치 메뉴를 선택한 뒤 실시간(1s) / 서버 요청시 |
| Relationship with other input/outputs | 위치 메뉴를 선택 한 사용자 기기의 output이 서버의 input이 된다. / 서버의 output이 사용자 기기의 input이 된다. |
| Format and configuration of screen | <p>지도에서 위치를 선택을 선택하면 해당하는 위치로 카메라가 탑재된 디바이스가 이동하여 해당 위치의 모습을 보여준다.</p>  |
| Format and configuration of window | N/A |

| | |
|-----------------|------------------------------|
| Data formats | NMEA Sentence -> json string |
| Command formats | N/A |
| End message | N/A |

표 7. User Interface - 반려동물 행동 로그 – 생성

| Name | 반려동물 행동 로그 – 생성 |
|---------------------------------------|---|
| Description of purpose | 반려동물의 행동 로그를 수집하고 데이터베이스에 저장한다. |
| Input source / Output destination | IoT 디바이스 또는 반려동물 AI의 이벤트 발생 / 데이터베이스 |
| Valid range / Accuracy / Tolerance | N/A |
| Unit of measure | 이벤트 |
| Timing | IoT 디바이스 또는 이상행동 AI의 요청시 |
| Relationship with other input/outputs | AI의 이상행동 탐지 결과 또는 IoT 디바이스의 반려동물 감지 이벤트가 Input이 된다. |
| Format and configuration of screen | N/A |
| Format and configuration of window | N/A |
| Data formats | Packet(MP4, JSON string) |
| Command formats | Query |
| End message | N/A |

표 8. User Interface - 반려동물 행동 로그 – 조회

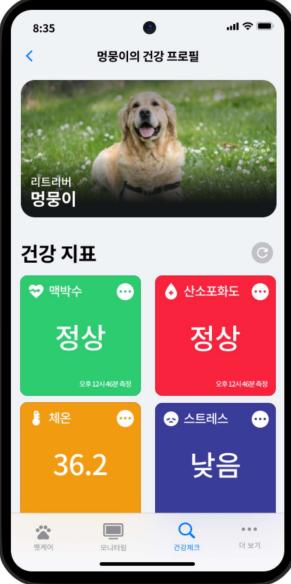
| Name | 반려동물 행동 로그 – 조회 |
|---------------------------------------|---|
| Description of purpose | 반려동물의 행동 로그를 조회기능을 통해 사용자가 영상 및 세부정보를 확인할 수 있다. |
| Input source / Output destination | 데이터베이스 / Android OS 또는 iOS가 탑재된 사용자 장치 |
| Valid range / Accuracy / Tolerance | 데이터베이스에 저장된 1달이내 로그 |
| Unit of measure | 화면 |
| Timing | 사용자 요청시 |
| Relationship with other input/outputs | AI의 이상행동 탐지 결과 또는 IoT 디바이스의 반려동물 감지 이벤트가 Input이 된다. |
| Format and configuration of screen |  <p>날짜 선택, 시간 선택, 로그 내용의 3부분으로 구성된다. 최상단은 최근 1달의 날짜를 표시한다. 각 날짜는 이벤트 발생 로그가 있을 경우 우상단에 표시한다. 시간 슬라이더의 로그 발생 시간의 하단에 로그 선택 버튼을 표시한다. 현재 선택된 버튼과 다른 로그는 색깔로 구분한다. 하단은 선택된 로그에 대해 발생 시간과 같이 자세한 세부 내용을 아래에 표시한다. 영상이 있는 경우 재생버튼을 추가로 표시하며, 재생버튼을 터치시 로그와 같이 녹화된 영상을 재생한다.</p> |
| Format and configuration of window | 반려동물 모니터링 탭에서 표시된다. |
| Data formats | Packet(MP4, JSON string) |
| Command formats | Query |

| | |
|-------------|-----|
| End message | N/A |
|-------------|-----|

표 9. User Interface - 반려동물 행동 로그 – 삭제

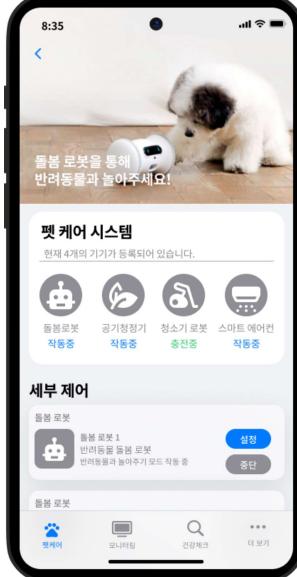
| Name | 반려동물 행동 로그 – 삭제 |
|---------------------------------------|---|
| Description of purpose | 서버의 저장공간을 위해 1달 이상 된 반려동물의 행동 로그를 삭제한다. |
| Input source / Output destination | N/A |
| Valid range / Accuracy / Tolerance | 데이터베이스에 저장된 1달 이상 로그 |
| Unit of measure | N/A |
| Timing | 일일 1회 자동 실행 |
| Relationship with other input/outputs | N/A |
| Format and configuration of screen | N/A |
| Format and configuration of window | N/A |
| Data formats | N/A |
| Command formats | Query |
| End message | N/A |

표 10. User Interface - 반려동물 건강체크

| Name | 건강 체크 |
|---------------------------------------|---|
| Description of purpose | 반려 동물의 건강정보를 측정하고 이에 대한 결과를 사용자에게 제공 한다. |
| Input source / Output destination | 사용자 / Android 혹은 IOS가 탑재된 사용자 기기 |
| Valid range / Accuracy / Tolerance | 사용자와 애완동물에게 부착된 디바이스 간의 유효거리 / 해당 없음 / 측정한 건강 지표별 오차 |
| Unit of measure | 화면 |
| Timing | 해당 애완동물의 건강체크 버튼 클릭시 |
| Relationship with other input/outputs | N/A |
| Format and configuration of screen |  <p>애완동물 별 건강 지표 및 상태를 알려주는 페이지가 존재하고 그 안에는 해당하는 애완동물의 건강 지표들의 수치나 상태들을 보여준다. 우측 중간에 위치한 리프레시 버튼을 통해 건강 지표들을 실시간으로 업데이트하여 확인해 볼 수 있다. 건강 지표아래 자세히 보기(...) 버튼을 누르면 해당 지표의 자세한 정보 및 기록들을 확인해 볼 수 있다.</p> |
| Format and configuration of window | N/A |
| Data formats | Button codes, JSON |
| Comm and formats | N/A |

| | |
|-------------|-----|
| End message | N/A |
|-------------|-----|

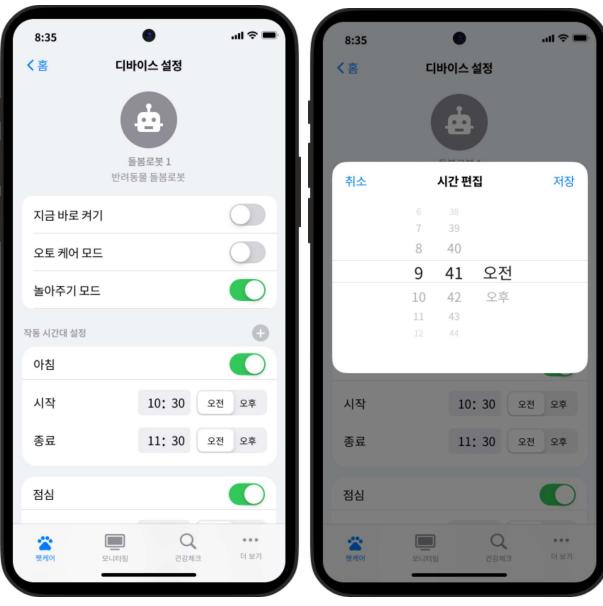
표 11. User Interface - 펫케어 시스템 - 메인 페이지

| Name | 펫 케어 시스템 - 메인 페이지 |
|---------------------------------------|---|
| Description of purpose | <p>사용자는 등록된 IoT 기기의 상태를 확인한다. 몇 개의 IoT기기가 등록되어 있는지 확인할 수 있으며, 어떤 기기가 작동 중 / 충전 중인지 확인할 수 있다.</p> <p>터치하여 등록된 기기에 대해서 설정 페이지로 이동하거나, 즉시 동작/중지하도록 할 수 있다</p> |
| Input source / Output destination | 사용자 / 안드로이드 OS 또는 iOS가 설치되어 있는 사용자의 핸드폰, 등록된 IoT기기 |
| Valid range / Accuracy / Tolerance | 화면의 버튼 수에 따른 범위 / 사용자의 터치 정확도에 따른 정확도 / 터치 감도 오차 |
| Unit of measure | 화면 |
| Timing | 비동기적 user input / user Instruction의 즉각적인 실행 |
| Relationship with other input/outputs | 모든 입력을 받은 뒤에, 사용자 기기는 입력 데이터 처리를 위해 입력을 서버로 보내고 원하는 출력 데이터를 요청한다. |
| Format and configuration of screen |  <p>1. 사용자는 등록된 IoT기기의 상태를 확인할 수 있다.</p> <p>2. 사용자는 설정 버튼을 터치하여 IoT 기기 설정 페이지로 이동해서 기기에 대한 설정을 진행할 수 있다.</p> <p>3. 사용자는 동작/중지 버튼을 터치하여 해당하는 IoT 기기를 동작/중지 시킬 수 있다.</p> <p>4. 시스템은 사용자의 요청에 즉각적으로 응답한다.</p> |
| Format and configuration of window | N/A |

| | |
|-----------------|-------------------------------|
| Data formats | (Int) Button codes |
| Command formats | Button code에 해당하는 Instruction |
| End message | N/A |

표 12. User Interface - 펫 케어 시스템 - 설정 페이지

| Name | 펫 케어 시스템 - 설정 페이지 |
|---------------------------------------|--|
| Description of purpose | 사용자가 등록된 IoT 기기의 설정을 조절한다. 사용자는 IoT기기의 작동 시간대를 설정하거나, 자동으로 작동하도록 두는 오토케어 모드, 지금 즉시 IoT기기를 동작/중지 할 수 있다. 작동 시간대를 설정하는 경우 시간대를 추가 또는 삭제할 수 있다. |
| Input source / Output destination | 사용자 / 안드로이드 OS 또는 iOS가 설치되어 있는 사용자의 핸드폰, 등록된 IoT기기 |
| Valid range / Accuracy / Tolerance | 화면의 버튼 수에 따른 범위 / 사용자의 터치 정확도에 따른 정확도 / 터치 감도 오차 |
| Unit of measure | 화면 |
| Timing | 비동기적 user input / user Instruction의 즉각적인 실행 |
| Relationship with other input/outputs | 모든 입력을 받은 뒤에, 사용자 기기는 입력 데이터 처리를 위해 입력을 서버로 보내고 원하는 출력 데이터를 요청한다. |

| | |
|---|---|
| <p>Format and configuration of screen</p> |  <p>1. 선택한 IoT기기의 상태를 확인할 수 있다.</p> <p>2. 사용자는 지금 바로 켜기/끄기 버튼을 통해 기기를 즉시 동작/중지 시킬 수 있다.</p> <p>3. 사용자는 오토 케어모드 버튼을 통해 기기가 자동으로 작동하도록 할 수 있다.</p> <p>4. 사용자는 작동 시간대 설정을 통해 작동 시간대에 기기가 동작하도록 할 수 있다. 버튼을 눌러 시간대를 추가하거나 기존의 시간대를 길게 눌러 삭제할 수 있다.</p> <p>5. 설정한 내용에 변화가 있을 경우 데이터베이스에 사용자의 변화된 현재 설정이 저장된다.</p> <p>6. 설정이 끝난 이후 사용자는 뒤로가기를 이용해서 펫 케어 시스템 – 메인 페이지로 돌아갈 수 있다.</p> |
| <p>Format and configuration of window</p> | N/A |
| <p>Data formats</p> | Int (button codes), Query |
| <p>Command formats</p> | Button code에 해당하는 instruction |
| <p>End message</p> | N/A |

3.1.2. Hardware Interfaces

표 13. Hardware Interfaces - 시스템에 사용 가능한 기기

| Name | 시스템에 사용 가능한 기기 |
|---------------------|--|
| Purpose/Description | 사용자들이 해당 조건을 만족하는 기기를 사용했을 때, 시스템이 제공하는 서비스를 받을 수 있도록 한다. / Android OS smartphone (Android 9.0 Pie 이상), iOS 기기 (iOS 12 이상) |

3.1.3. Software Interfaces

표 14. Software Interfaces - MariaDB Server

| Name | MariaDB Server |
|---------------------------------------|--|
| Purpose/Description | Multimedia/meta data를 관리하기 위한 Query input/output |
| Input source / Output destination | 사용자 / 서버 |
| Range/ Accuracy/ Margin of error | MariaDB Server의 성능에 따른다. |
| Unit | Query |
| Timing | 즉각적인 반응 |
| Relationship with other input/outputs | 서버의 모든 input/output과 관련 있다. |
| Format and configuration of screen | N/A |
| Format and configuration of window | N/A |
| Data Formats | Query |
| Command formats | Query statement |

| | |
|-------------|-----|
| End Message | N/A |
|-------------|-----|

3.1.4. Communication Interfaces

표 15. Communication Interfaces - 로그인

| Name | 로그인 |
|---------------------------------------|--|
| Purpose/Description | 사용자는 서버에 로그인하여 승인이 완료된 후부터 어플리케이션을 이용할 수 있다. |
| Input source / Output destination | Client / DB Server, DB Server / Client |
| Units of measure | Packet(JSON) |
| Timing | Instant Reaction |
| Relationship with other input/outputs | N/A |
| Format and configuration of screen | N/A |
| Format and configuration of window | N/A |
| Data Formats | Query, Text |
| Command formats | Query statement |
| End Message | N/A |

표 16. Communication Interfaces - 반려동물 로그

| Name | 반려동물 로그 |
|---------------------------------------|---|
| Purpose/Description | 반려동물의 이상행동/ 반려동물 위치 / 반려동물 건강상태 / IoT 디바이스 정보 로그 저장 |
| Input source / Output destination | Client / DB Server, DB Server / Client |
| Units of measure | Packet(JSON) |
| Timing | Instant Reaction |
| Relationship with other input/outputs | N/A |
| Format and configuration of screen | N/A |
| Format and configuration of window | N/A |
| Data Formats | Query, Text |
| Command formats | Query statement |
| End Message | N/A |

표 17. Communication Interfaces - 반려동물 촬영 영상

| Name | 반려동물 촬영 영상 |
|---------------------------------------|--|
| Purpose/Description | 반려동물의 이상행동 영상 저장 및 스트리밍 |
| Input source / Output destination | Client / DB Server, DB Server / Client |
| Units of measure | Packet(MP4) |
| Timing | 10Mbps (min: 2Mbps) |
| Relationship with other input/outputs | N/A |
| Format and configuration of screen | N/A |
| Format and configuration of window | N/A |
| Data Formats | Query, MP4 |
| Command formats | Query statement |
| End Message | N/A |

표 18. Communication Interfaces - IoT 작동/중지

| Use case name | IoT 작동/중지 |
|---------------------------------------|---|
| Purpose/Description | 사용자가 IoT 기기를 동작/중지 하고 기기의 상태를 DB에 저장한다. |
| Input source / Output destination | Client / DB Server, DB Server / Client |
| Units of measure | Packet(JSON) |
| Timing | Instant Reaction |
| Relationship with other input/outputs | N/A |
| Format and configuration of screen | N/A |
| Format and configuration of window | N/A |
| Data Formats | Query, Text |
| Command formats | Query statement |
| End Message | N/A |

표 19. Communication Interfaces - IoT 설정 변경

| Use case name | IoT 설정 변경 |
|---------------------------------------|--|
| Purpose/Description | 사용자가 IoT 기기의 설정을 변경하고 설정을 DB에 저장한다. |
| Input source / Output destination | Client / DB Server, DB Server / Client |
| Units of measure | Packet(JSON) |
| Timing | Instant Reaction |
| Relationship with other input/outputs | N/A |
| Format and configuration of screen | N/A |
| Format and configuration of window | N/A |
| Data Formats | Query, Text |
| Command formats | Query statement |
| End Message | N/A |

표 20. Communication Interfaces - 반려동물 건강정보 요청 및 업데이트

| Use case name | 반려동물 건강정보 요청 및 업데이트 |
|---------------------------------------|---|
| Purpose/Description | 사용자 요청시 저장된 애완동물의 건강 지표들을 반환하고 필요할 경우 실시간 건강 지표를 갱신하여 DB에 저장한다. |
| Input source / Output destination | Client / DB Server, DB Server / Client |
| Units of measure | Packet(JSON) |
| Timing | Instant Reaction |
| Relationship with other input/outputs | N/A |
| Format and configuration of screen | N/A |
| Format and configuration of window | N/A |
| Data Formats | Query, Text |
| Command formats | Query statement |
| End Message | N/A |

3.2. Functional Requirements

3.2.1. Use Case

표 21. Use case - 사용자 계정 관리

| Use case name | 사용자 계정 관리 |
|----------------|--|
| Actor | 어플리케이션 이용자, DB 서버 |
| Description | 이메일 주소, 비밀번호를 이용한 로그인/ 회원가입 / ID,PW찾기 |
| Normal Course | <p>기존 이용자의 경우</p> <ul style="list-style-type: none"> -이메일 주소/비밀번호 입력 -잘못 입력시 오류 메시지 팝업 ID/PW 분실시 하단의 ID/PW찾기 버튼 이용 -사용자의 주민등록번호와 실명 확인시 ID 공개 -핸드폰 번호/이메일 주소 인증 후 새로운 PW 생성 <p>신규 이용자의 경우</p> <ul style="list-style-type: none"> -회원가입 버튼을 눌러 계정 생성 -계정 생성시 ID로 사용할 이메일 주소, 비밀번호, 이름, 생년월일, 주민등록번호, 핸드폰 번호, 이메일 주소 기입 |
| Pre-condition | 어플리케이션이 실행된 직후 |
| Post-condition | <p>로그인 DB 서버에 사용자들의 이메일 주소/비밀번호들이 저장되어 있어야 한다.</p> <p>로그인 성공시-펫 케어 시스템(메인 페이지)로 넘어감</p> <p>로그인 실패시-팝업창 토스트</p> <p>신규 회원가입-회원가입용 정보 기입창으로 넘어감</p> <p>ID/PW 찾기-사용자 정보를 입력받아 ID/PW 찾는 창으로 넘어감</p> |
| Assumption | N/A |

표 22. Use case - 반려동물 이상행동 앱 푸시 알림

| Use case name | 반려동물 이상행동 앱 푸시 알림 |
|----------------|---|
| Actor | 로그인 상태의 사용자, 반려동물 |
| Description | 반려동물을 촬영한 실시간 영상을 AI로 분석하여 이상 행동이 감지되면 사용자에게 즉시 앱 푸시 알림이 전송된다. |
| Normal Course | 1. 사용자는 해당 기기에서 로그인한 적이 있다. 2. IoT 디바이스에 장착된 카메라로 촬영한 반려동물의 실시간 영상에 대해 AI의 이상 행동 탐지가 이루어지면 서버 요청으로 앱 푸시 알림을 수신한다. 이때, 앱은 미실행 상태일 수 있다. 여기서 이상 행동이란, 반려동물이 일정 시간 이상 짖거나 소리를 낼 경우, 혹은 움직임이 없을 경우 등을 말한다. 3. 푸시 알림을 확인한 사용자는 앱에 접속하여 반려동물 건강상태 체크 서비스 등을 이용해 반려동물에 대한 후속 조치를 진행할 수 있다. |
| Pre-condition | 사용자는 로그인 된 상태여야 한다. 기기가 네트워크에 연결되어 있어야 한다. |
| Post-condition | N/A |
| Assumption | 카메라가 장착된 IoT 디바이스가 앱의 맞춤 케어 시스템에 등록되어 있음을 가정한다. |

표 23. Use case - 실시간 위치 파악

| Use case name | 실시간 위치 파악 |
|---------------|--|
| Actor | 로그인 상태의 사용자, 반려동물 |
| Description | 앱을 켜고 위치 메뉴를 선택하면 실시간 위치를 보여준다 / 지정 범위를 벗어나면 앱 푸시로 알림을 준다. |
| Normal Course | [실시간 위치 파악] <ol style="list-style-type: none"> 사용자가 위치 메뉴를 선택하면 서버에 실시간 위치를 요청한다. 서버는 IoT 기기에서 수신받은 위치 정보를 사용자 기기에 전달한다. 사용자 기기는 지도 형태로 위치 정보를 표시한다. [지정 위치 벗어남 알림] <ol style="list-style-type: none"> 사용자가 어플리케이션에 로그인을 하고 지정 위치 범위를 등록한 상태이여야 한다. 서버에서 실시간으로 수신받은 위치 정보가 지정 위치 범위를 넘어섰을 때 사용자에게 알림을 송신한다. 알림을 수신받은 사용자는 어플리케이션에 접속하여 반려 동물의 |

| | |
|----------------|---|
| | 실시간 위치를 파악한다. |
| Pre-condition | 사용자가 앱을 켜고 위치 메뉴를 선택한 상태 / 로그인 한 상태, 위치를 등록한 상태 |
| Post-condition | N/A |
| Assumption | 반려 동물에게 위치 정보를 파악 할 수 있는 디바이스가 장착되어 있음을 가정한다. |

표 24. Use case - 반려동물 행동 로그 조회

| Use case name | 반려동물 행동 로그 조회 |
|----------------|---|
| Actor | 로그인 상태의 사용자 |
| Description | 서버 데이터베이스에 저장된 반려동물의 로그를 날짜와 시간에 따라 정렬하여 보여주며 각 로그를 선택하면 자세한 내용과 영상이 존재하면 영상을 보여준다. |
| Normal Course | <p>[로그 목록 확인]</p> <ol style="list-style-type: none"> 1. 사용자가 날짜를 선택한다. 2. 데이터베이스에서 선택한 날짜에 발생한 로그를 조회하여 사용자 기기로 전송한다. 3. 사용자 기기는 서버에서 받은 로그목록을 시간 바위에 표시한다. <p>[로그 세부 사항 확인]</p> <ol style="list-style-type: none"> 1. 사용자가 로그가 발생한 시간에 표시된 점을 터치한다. 2. 사용자 기기는 데이터베이스에 저장된 세부 내용과 영상을 아래에 표시한다. 3. 사용자가 영상을 터치하면 영상을 재생한다. |
| Pre-condition | 사용자가 앱을 켜고 위치 메뉴를 선택한 상태 / 로그인 한 상태, 위치를 등록한 상태 / 로그가 발생한 상태 |
| Post-condition | N/A |
| Assumption | N/A |

표 25. Use case - 반려동물 건강 체크

| Use case name | 반려동물 건강 체크 |
|---------------|---|
| Actor | 로그인 상태의 사용자 / 반려동물 |
| Description | 서버 데이터베이스에 저장된 반려동물의 건강 지표를 조회하여 사용자에게 관찰 대상인 반려동물의 건강 상태를 표시해준다. |
| Normal Course | [건강 지표 조회] |

| | |
|----------------|---|
| | <ol style="list-style-type: none"> 1. 사용자가 반려동물의 프로필 페이지에 진입한다. 2. 해당 반려동물의 각종 건강지표들에 대한 분석 결과를 제공한다. 3. 필요할 경우 해당하는 건강 지표에 대한 자세한 정보를 제공한다. <p>[실시간 건강 지표 갱신]</p> <ol style="list-style-type: none"> 1. 사용자가 건강 지표 업데이트 버튼을 누른다. 2. 백엔드 서버가 반려동물에게 부착된 건강 디바이스에 실시간 건강 지표의 갱신을 요청한다. 3. 갱신된 실시간 건강 지표정보를 가져와 DB에 저장한다. |
| Pre-condition | 사용자가 앱을 켜고 애완동물 프로필에 진입한 상태 / 애완동물의 프로필을 등록한 상태 |
| Post-condition | N/A |
| Assumption | 애완동물에게 건강 지표를 측정하는 디바이스를 부착한 상태라고 가정한다. |

표 26. Use case - IoT 작동 / 중지

| Use case name | IoT 작동 / 중지 |
|---------------|---|
| Actor | 사용자, 서버, IoT 기기 |
| Description | IoT 기기를 동작/중지 한다. |
| Normal Course | <p>[동작]</p> <ol style="list-style-type: none"> 1. 등록된 IoT 기기 중 사용자가 선택한 기기를 동작 상태로 바꾼다. 2. 전원이 계속 공급되는 IoT기기 (공기 정화 기기, 온도 관리 기기)는 기본적으로 중지 호출이 오기 전까지 동작한다. 전원이 계속 공급되지 않는 IoT기기(먹이 공급 장치, 청소 로봇, 놀이 로봇)은 동작 상태 이후 30분간 동작한다. 3. 사용자가 동작 시간대를 등록한 경우, 사용자가 등록한 시간대에서 가장 가까운 시간대의 중지 시각에 중지한다. <p>[중지]</p> <ol style="list-style-type: none"> 1. 등록된 IoT 기기 중 사용자가 선택한 기기를 중지 상태로 바꾼다. 2. 사용자가 동작 시간대를 등록한 경우, 사용자가 등록한 시간대에서 가장 가까운 시간대의 동작 시각에 동작한다. |
| Pre-condition | <p>[동작]</p> <p>IoT 기기의 중단 상태</p> |

| | |
|----------------|--|
| | [중지] IoT 기기의 동작 상태 |
| Post-condition | [동작] IoT 기기의 동작 상태 [중지] IoT 기기의 중단 상태 |
| Assumption | 해당 IoT기기는 충전 되어있어야 한다. |

표 27. Use case - IoT 설정 변경

| Use case name | IoT 설정 변경 |
|----------------|---|
| Actor | 사용자, 서버, 데이터베이스 |
| Description | IoT 기기의 설정을 저장한다. |
| Normal Course | <ol style="list-style-type: none"> 1. 사용자가 펫 케어 시스템 – 설정 페이지에 진입한다. 2. 사용자가 설정을 조작하여 설정의 변경 사항이 발생한다. 3. 설정의 현재 상태를 데이터 베이스에 저장한다. 4. 이 후 IoT 기기는 데이터 베이스에 저장된 설정을 바탕으로 작동한다. |
| Pre-condition | 사용자가 서버와 연결되어 있어야 한다. |
| Post-condition | IoT기기의 설정이 데이터 베이스에 저장되어 IoT기기가 이를 참고하여 동작을 할 수 있어야한다. |
| Assumption | N/A |

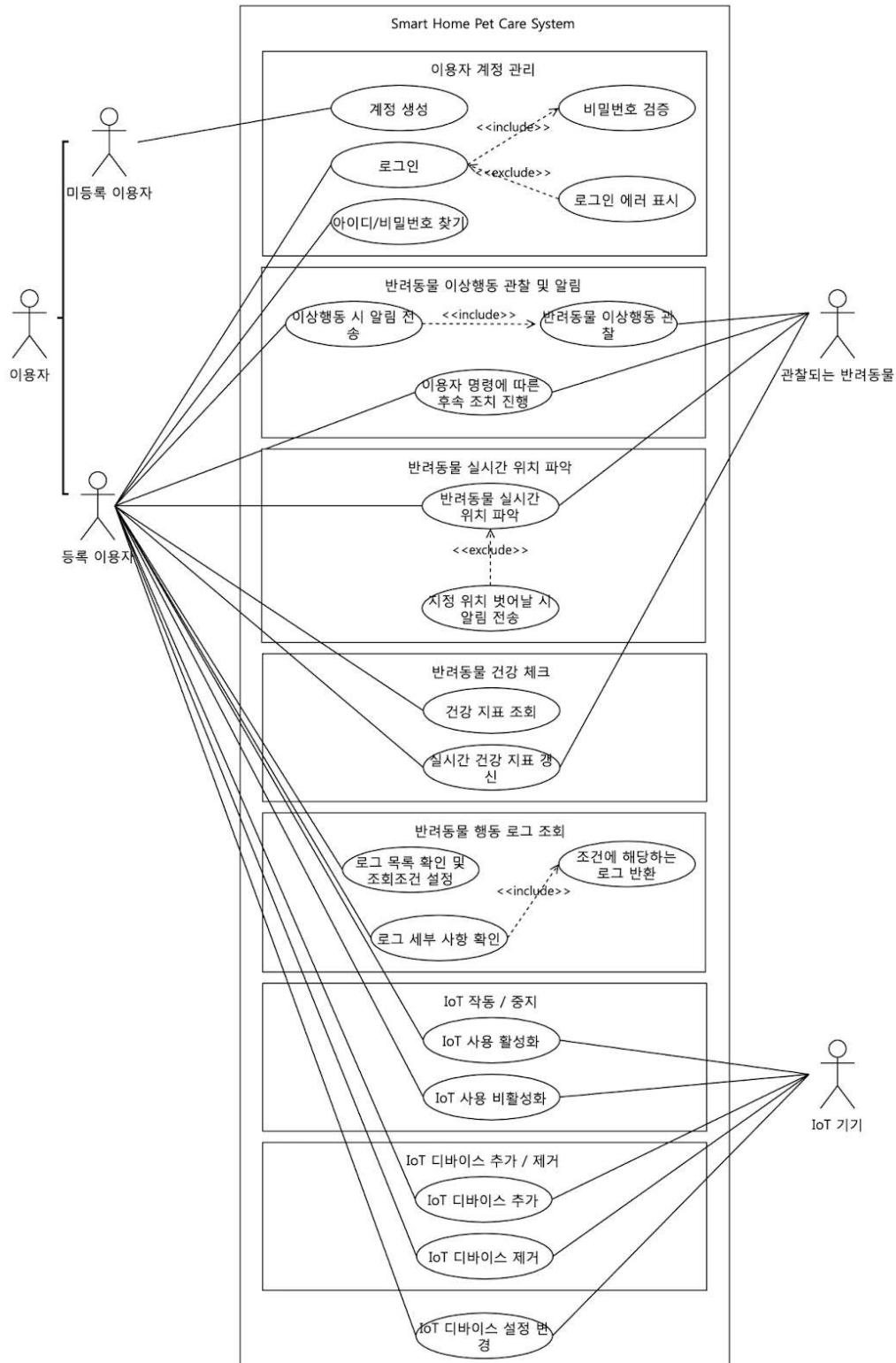
표 28. Use case - IoT 디바이스 추가/제거

| Use case name | IoT 디바이스 추가/제거 |
|---------------|-----------------|
| Actor | 사용자, 서버, 데이터베이스 |

| | |
|----------------|---|
| Description | IoT 기기의 연결을 추가/제거한다. |
| Normal Course | <p>[디바이스 관리 공통]</p> <ol style="list-style-type: none"> 1. 사용자가 펫 케어 시스템 어플리케이션을 실행시킨 후, [더보기]에 진입해 디바이스 추가/제거 페이지에 진입한다. 2. 사용자는 현재 등록된 디바이스의 정보를 확인하는 화면을 확인할 수 있으며, 추가적인 기기를 등록할 수 있는 기능 또한 제공받는다. <p>[디바이스 추가]</p> <ol style="list-style-type: none"> 1. 사용자가 추가적인 디바이스의 사용을 원할 경우, 화면에 있는 [기기 추가] 버튼을 누른다. 2. 현재 동일한 Wi-Fi에 연결되어 있는 기기의 정보들이 표시되고, 사용자는 이 기기들 중 연결하고자 원하는 기기를 선택한다. 3. 어플리케이션에 기기의 정보가 등록되고, 데이터베이스에 연동된 기기의 연결 정보가 저장된다. 4. 해당 연결 정보는 서버에 저장되어, 이용자가 외부에서도 IoT 디바이스를 이용할 수 있도록 한다. <p>[디바이스 제거]</p> <ol style="list-style-type: none"> 1. 사용자가 사용중인 디바이스의 제거를 원할 경우, 화면에 있는 기기 목록 중 하나를 선택해 제거한다. 이 때, 기기 목록의 오른쪽 상단에 위치한 더보기 메뉴를 이용한다. 2. 현재 어플리케이션과 연동된 디바이스의 정보가 데이터베이스에서 제거되고, 어플리케이션에서도 기기의 정보가 제거된다. 3. 서버에서도 해당 연결 정보가 제거되어, 이용자는 대상 디바이스와의 관계가 사라지게 된다. |
| Pre-condition | 사용자가 사용중인 어플리케이션이 작동중인 모바일 디바이스와 등록하고자 하는 IoT 디바이스가 동일한 네트워크에 연결되어 있으며, 여기서의 네트워크는 인터넷을 이용해 서버와 연결되어 있어야 한다. |
| Post-condition | IoT 디바이스와의 연결 정보가 올바르게 데이터베이스에 저장되어 있어야 서버에서 사용자가 사용중인 IoT 디바이스의 목록을 파악하고 이에 맞는 동작을 수행할 수 있다. |
| Assumption | IoT 디바이스가 충분히 충전되어 있어야 연결의 추가가 가능하다. 제거의 경우에는 해당되지 않는다. |

3.2.2. Use Case Diagram

그림 1. Use Case Diagram



3.2.3. Data Dictionary

표 29. Data Dictionary - 사용자 계정 관리

| Field | Primary | Foriegn | Constraint | Type | Description |
|----------------------------------|---------|---------|------------|---------|-------------|
| email(id) | o | | NOT NULL | String | 이메일 주소 |
| password | | | NOT NULL | String | 비밀번호 |
| name | | | NOT NULL | String | 이름 |
| birthday | | | NOT NULL | integer | 생년월일 |
| registration number | | | NOT NULL | integer | 주민등록 번호 |
| phone number | | | NOT NULL | integer | 전화번호 |
| email address(for certification) | | | NOT NULL | String | 이메일 주소(인증용) |

표 30. Data Dictionary - 실시간 위치 모니터링

| Field | Primary | Foriegn | Constraint | Type | Description |
|--------------|---------|---------|------------|--------------|--------------------|
| location | | | NOT NULL | char(100) | 위치 정보 |
| set_location | | | | char(100)[4] | 사용자가 등록한 위치 정보 |
| isOut | | | | boolean | 사용자가 등록한 위치 밖인지 여부 |

표 31. Data Dictionary - 반려동물 이상행동 감지

| Field | Primary | Foriegn | Constraint | Type | Description |
|-------------------|---------|---------|------------|-----------|--------------|
| Abnormal_behavior | | ○ | NOT NULL | Boolean | 반려동물 이상행동 여부 |
| behavior | | ○ | NOT NULL | char(100) | 반려동물 행동로그 |

표 32. Data Dictionary - IoT 디바이스 등록 및 설정

| Field | Primary | Foriegn | Constraint | Type | Description |
|-------------------|---------|---------|------------|-----------|---------------------|
| Device_SN | ○ | | NOT NULL | char(100) | IoT 디바이스의 일련번호 |
| Device_Type | | ○ | NOT NULL | char(100) | IoT 디바이스의 종류 |
| Connection_Status | | ○ | | Boolean | 디바이스와 어플리케이션의 연동 여부 |
| Connected_User | | ○ | | char(100) | 연동된 유저의 아이디 |

표 33. Data Dictionary - IoT 디바이스 기능 제어

| Field | Primary | Foriegn | Constraint | Type | Description |
|-------------|---------|---------|------------|-----------|----------------|
| Device_SN | ○ | | NOT NULL | char(100) | IoT 디바이스의 일련번호 |
| Device_Type | | ○ | NOT NULL | char(100) | IoT |

| | | | | | 디바이스의 종류 |
|--------------------|--|---|----------|---------|--------------------|
| Enabled_Location | | ○ | | Boolean | 위치 추적 사용 여부 설정 |
| Enabled_HealthInfo | | ○ | | Boolean | 건강 정보 제공 사용 여부 설정 |
| Enabled_Camera | | ○ | | Boolean | 카메라 사용 여부 설정 |
| Enabled_PushAlarm | | ○ | | Boolean | PUSH 알림 사용 여부 설정 |
| Device_Working | | | NOT NULL | Boolean | IoT 디바이스의 작동 여부 설정 |
| Device_WorkingMode | | | NOT NULL | int | 디바이스의 작동 모드 설정 |

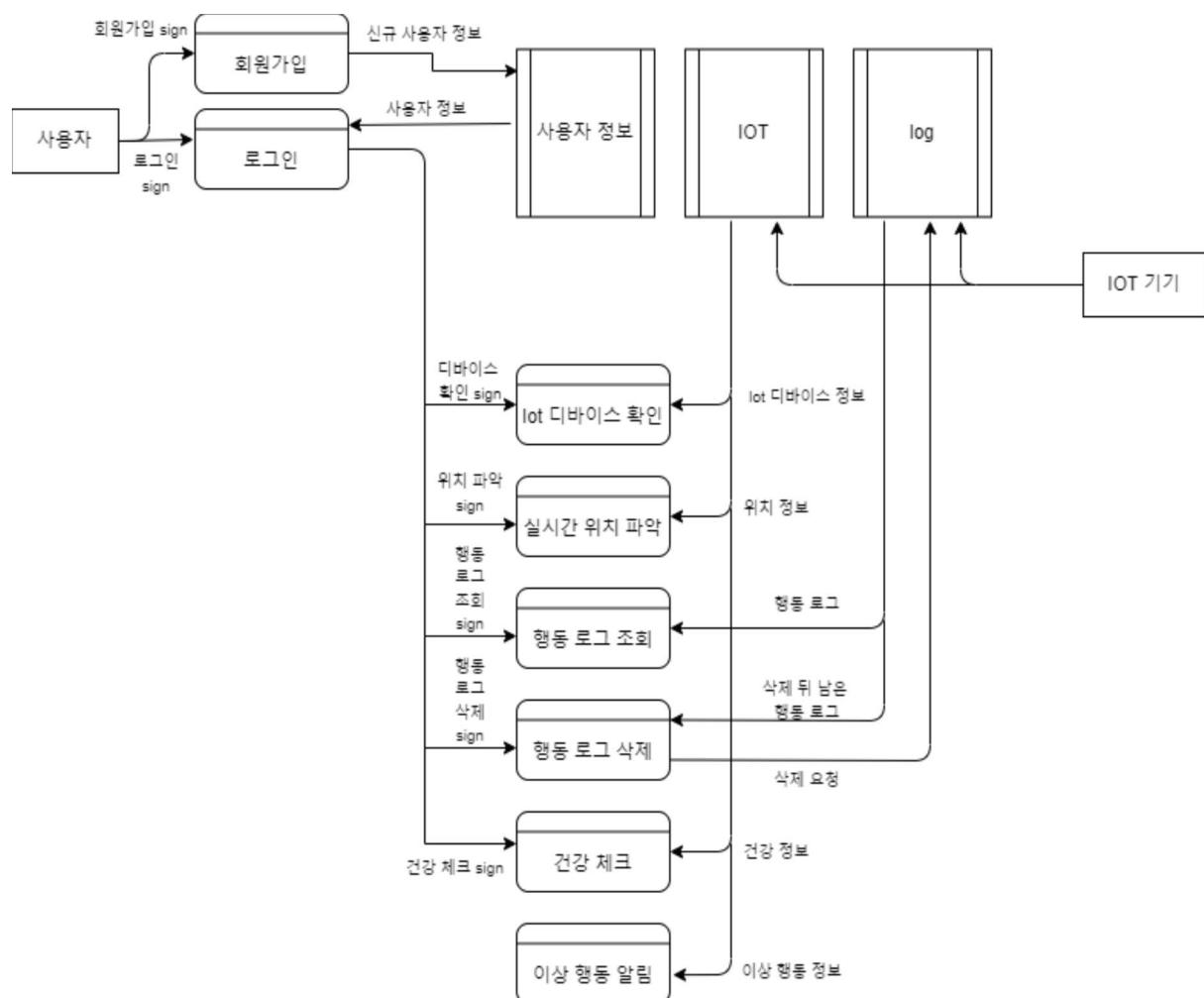
표 34. Data Dictionary - IoT 디바이스 작동 시간대 설정

| Field | Primary | Foriegn | Constraint | Type | Description |
|-------------|---------|---------|------------|-----------|----------------|
| Device_SN | ○ | ○ | NOT NULL | char(100) | IoT 디바이스의 일련번호 |
| Timeline_ID | ○ | | NOT_NULL | int | 디바이스의 작동 시간대의 |

| | | | | | ID |
|------------|--|--|----------|-----------|----------------|
| Start_Time | | | NOT_NULL | char(100) | 디바이스의 동작 시작 시간 |
| End_Time | | | NOT_NULL | char(100) | 디바이스의 동작 중지 시간 |

3.2.4 Data Flow Diagram

그림 2. Data Flow Diagram



3.3. Product Requirements

3.3.1 Performance Requirements

다음 성능 요구사항은 제품의 빠른 동작속도를 달성하고 쾌적한 사용이 가능한 최소한의 성능을 보장하기 위한 요구사항이다.

- 사용자 기기는 1.5GB 이상의 메모리와 1.4GHz이상의 CPU, Android 9.0 Pie 이상 또는 iOS 14 이상의 기기여야 한다.
- 로그인 프로세스는 2초 이내에 완료되어야 한다.
- 화면간 전환은 1초 이내에 완료되어야 한다.
- IoT 디바이스는 최대 10개까지 연결할 수 있다.
- 모니터링 시스템의 카메라 화면은 최소 480p, 2Mbps의 사양으로 스트리밍 해야한다.
- 이벤트 발생 시점으로부터 최대 1분 이내에 사용자 기기에서 푸시 알림을 확인할 수 있어야한다.
- 사용자 기기를 통해 IoT 디바이스를 조작을 명령하였을때 3초 이내에 IoT 디바이스에 명령이 전달되어야한다.

3.3.2. Space Requirements

다음 저장 요구사항은 제품이 얼마만큼의 저장공간과 RAM을 사용하는지를 정의하는 요구사항을 다루고 있다.

- 어플리케이션이 사용자 기기에서 차지하는 저장공간은 400MB를 넘지 않아야한다.
- 어플리케이션은 모든 상황에 있어서 200MB를 넘지 않아야한다.
- 모든 자료는 클라우드 서버에 저장되어야 한다.
- 데이터베이스에 저장되는 영상의 크기는 1개 영상당 20MB를 넘을 수 없다.

3.3.3. Security Requirements

다음 보안 요구사항은 제품이 작동하는 과정에서 개인정보를 보호하고 시스템에 대한 침입을 막기 위한 요구사항을 다루고 있다.

- 하나의 기기당 동시에 한 계정의 접근만 허용한다.
- 하나의 계정당 동시에 한 기기에서의 접근만 허용한다.
- 계정이 어플리케이션에서 로그아웃할 시 기기에 저장된 임시파일 등은 다른 계정에서 접근할 수 없어야 한다.

- 데이터베이스에 저장된 영상자료의 경우 각 계정 고유의 암호화 키를 이용해 암호화하여 저장한다.

3.3.4. Usability Requirements

다음 사용성 요구사항은 사용자들이 이용하는데 있어 오류를 발생시키지 않고 개발 의도대로 사용할 수 있도록 직관적인 사용성에 대해 다루고 있다.

- 모든 사용자는 별도의 설명서 없이 모든 기능을 쉽게 사용할 수 있어야 한다.
- 모든 기능은 일관된 인터페이스를 갖추어야 한다.
- 인터페이스는 직관적이고 간결해야 한다.

3.3.5. Dependability Requirements

다음 신뢰성 요구사항은 제품의 기능이 모두 의도대로 작동할 수 있도록 규정하기 위한 요구사항이다.

- 사용자 기기가 서버와 통신 도중 연결이 끊길 시 최소 3번 이상의 재시도를 진행한다.
- 서버와 IoT 디바이스의 지속적인 연결을 위하여 일정 시간 간격으로 연결 상태를 점검한다.
- IoT 디바이스는 서버와 연결이 단절 되었을 때는 작동을 정지한다.

3.4 Organizational Requirements

해당 파트에서는 비기능적 요구사항 중 하나인 조직적 요구사항에 대해 다룰 것이다. 조직적 요구사항은 소프트웨어의 사용자나 개발자의 조직내에서 사용되는 광범위한 시스템 요구사항들을 의미한다.

3.4.1. Environmental Requirements

환경적 요구사항은 시스템 운영 환경을 명시한 요구사항들로 그 요구사항들은 다음과 같다.

- 해당 시스템의 사용자는 애완동물을 기르는 사람들로 규정한다.
- 해당 시스템은 사용자의 가정의 원활한 유무선 네트워크 환경 아래에서만 작동한다고 규정한다
- 해당 시스템의 기능들은 전적으로 사용자에 의해 통제될 수 있음을 규정한다.

3.4.2. Operational Requirements

운영적 요구사항은 해당 시스템이 어떠한 방식으로 사용 될지를 정의하는 요구사항으로서 시스템 운영을 원활하게 하기 위해 필요한 요구사항 들이다. 사용자의 집안 환경에서 사용되는 시스템인

만큼 사용자의 사생활 보호가 최우선 순위로 고려된다. 해당되는 요구사항들은 다음과 같다.

- 사용자가 모니터링을 위해 촬영되는 화면이 사용자가 아닌 제 3자에게 노출되지 않도록 해야한다.
- 사용자가 펫케어 시스템을 이용하는 과정에서 사용자가 등록하지 않았거나 소유하지 않은 가전 제품은 통제할 수 없도록 해야한다.

3.4.3. Development Requirements

개발 요구사항은 시스템을 개발하는 개발자들 조직내에서 필요한 요구사항들을 의미한다. 해당되는 요구 사항들은 다음과 같다.

- 시스템의 어플리케이션은 Android와 iOS의 동시 호환성을 지원하기 위해 크로스 플랫폼을 지원하는 Flutter(v 2.10.5) 프레임워크를 사용한다.
- 시스템의 어플리케이션 개발 프로그래밍 언어는 Dart(v 2.14)를 이용한다.
- 시스템의 백엔드 시스템은 NodeJS(v 16.15.0) 환경에서 개발되어야 한다..
- 시스템의 데이터베이스 시스템은 MariaDB(v 10.9.0)을 이용한다.

3.5. External Requirements

해당 장에서는 외부에서 작용하는 요구사항들을 규정하며, 시스템 혹은 개발 과정에 대한 규제, 윤리, 회계, 안전과 보안에 관한 요구사항들을 다룬다.

3.5.1. Regulatory Requirements

규제 요구사항은 시스템 외부에서 작용하는 요구사항을 다룬다.

- 시스템이 사용하는 오픈소스 소프트웨어는 원작자의 사용 규정에 따라 접근 및 개발해야 한다.
- 시스템의 서비스와 해당 데이터는 국가개인정보보호 표준에 따라 개발 및 운영되어야 한다.

3.5.2. Ethical Requirements

윤리적 요구사항은 시스템의 개발과 운영에 있어서 비윤리적 행위의 개입을 규제한다.

- 시스템의 개발과 운영에는 악의적인 의도로 부당한 이득을 취하는 집단이 없어야 한다.
- 사용자의 의도에 따라 시스템을 범죄 수단으로 악용할 수 있는 행위는 차단되어야 한다.

3.5.3. Accounting Requirements

회계 요구사항은 시스템의 개발과 운영에 관련된 금전적인 사항을 규정한다.

- 시스템 개발과 운영에 사용되는 비용은 계약서에 명시된 규정을 따라야 한다.

3.5.4. Safety Security Requirements

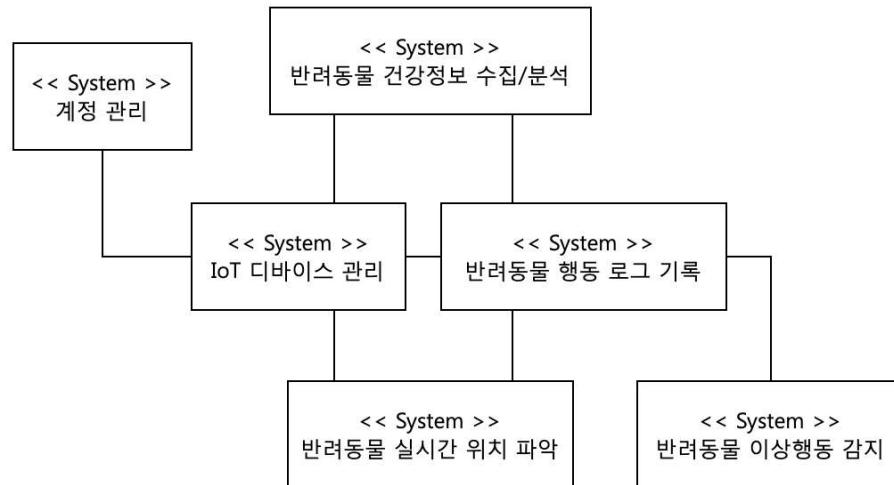
안전 보안 요구사항은 시스템의 개발과 운영에 있어서 안전과 보안이 지켜질 수 있도록 규정한다.

- 시스템 개발 과정에서 개발 자원과 데이터의 보안이 보장되어야 한다.
- 개인정보보호법에 의거하여 시스템에서 수집하는 모든 사용자 정보는 암호화되어 저장되며 사용자가 동의한 범위와 기간 내에서만 사용 가능해야 한다.
- 외부 시스템 사용자들의 접근이 불가능해야 하며, 외부의 충격으로 시스템 데이터가 손상되지 않도록 안전이 보장되어야 한다.

3.6. Organizing the Specific Requirements

3.6.1. Context Model

그림 3. Context Model



3.6.2. Process Model

프로세스 모델을 통해, 수행될 작업들의 프로세스 순서를 보여준다.

그림 4. Process Model - 총괄

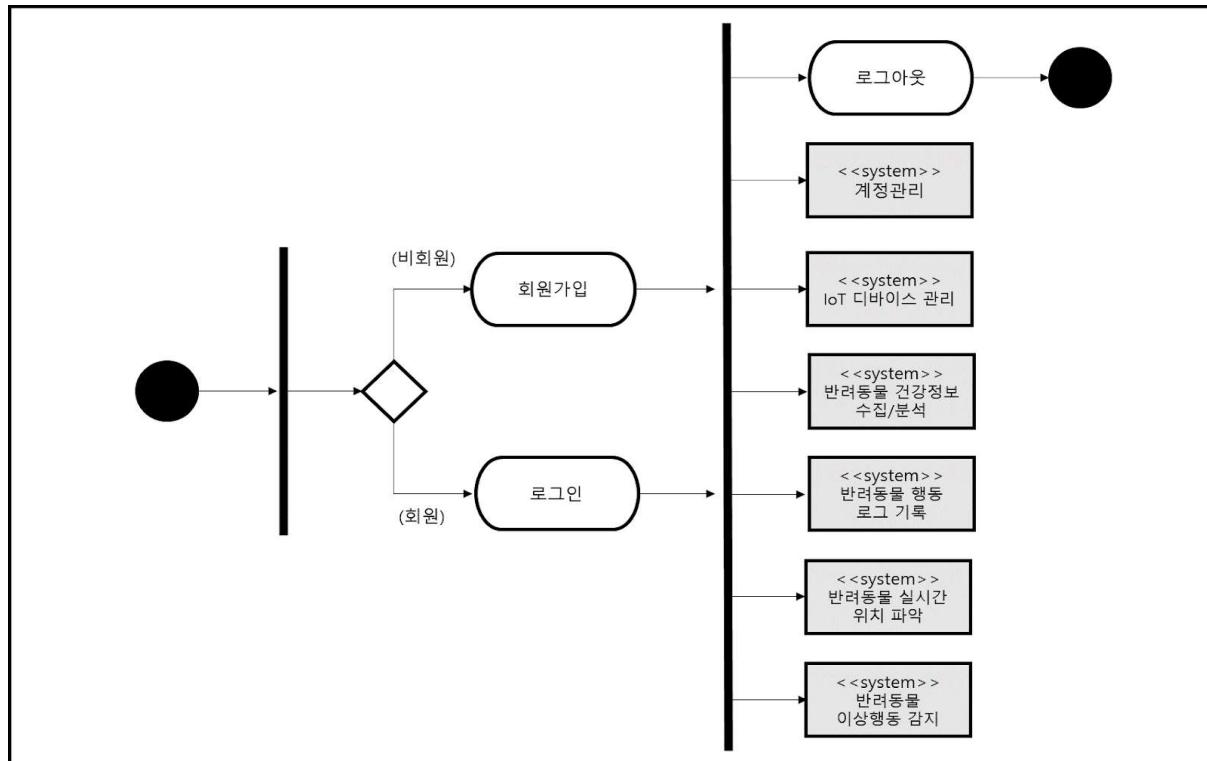


그림 5. Process Model - 계정관리

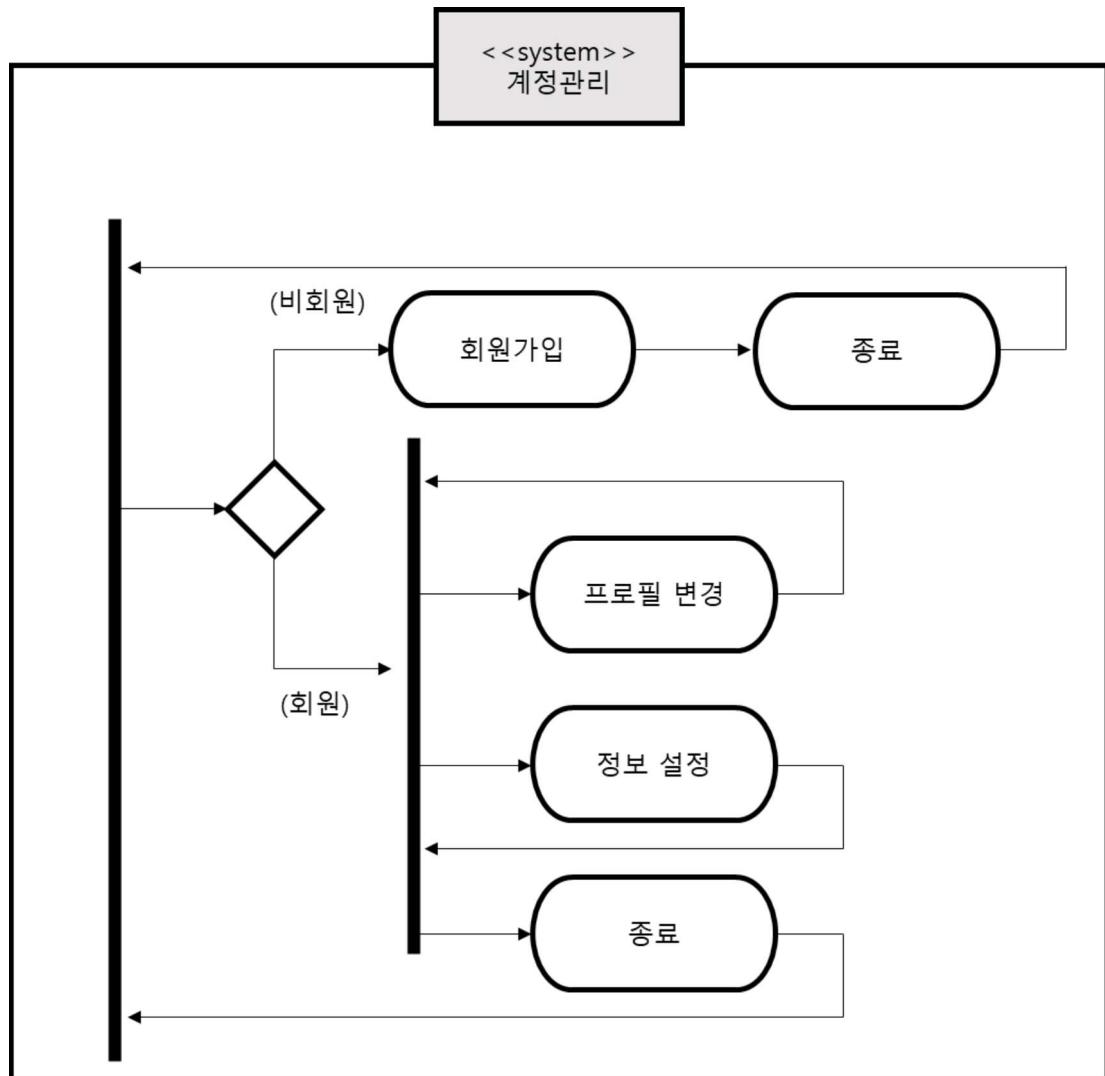


그림 6. Process Model - IoT 디바이스 관리

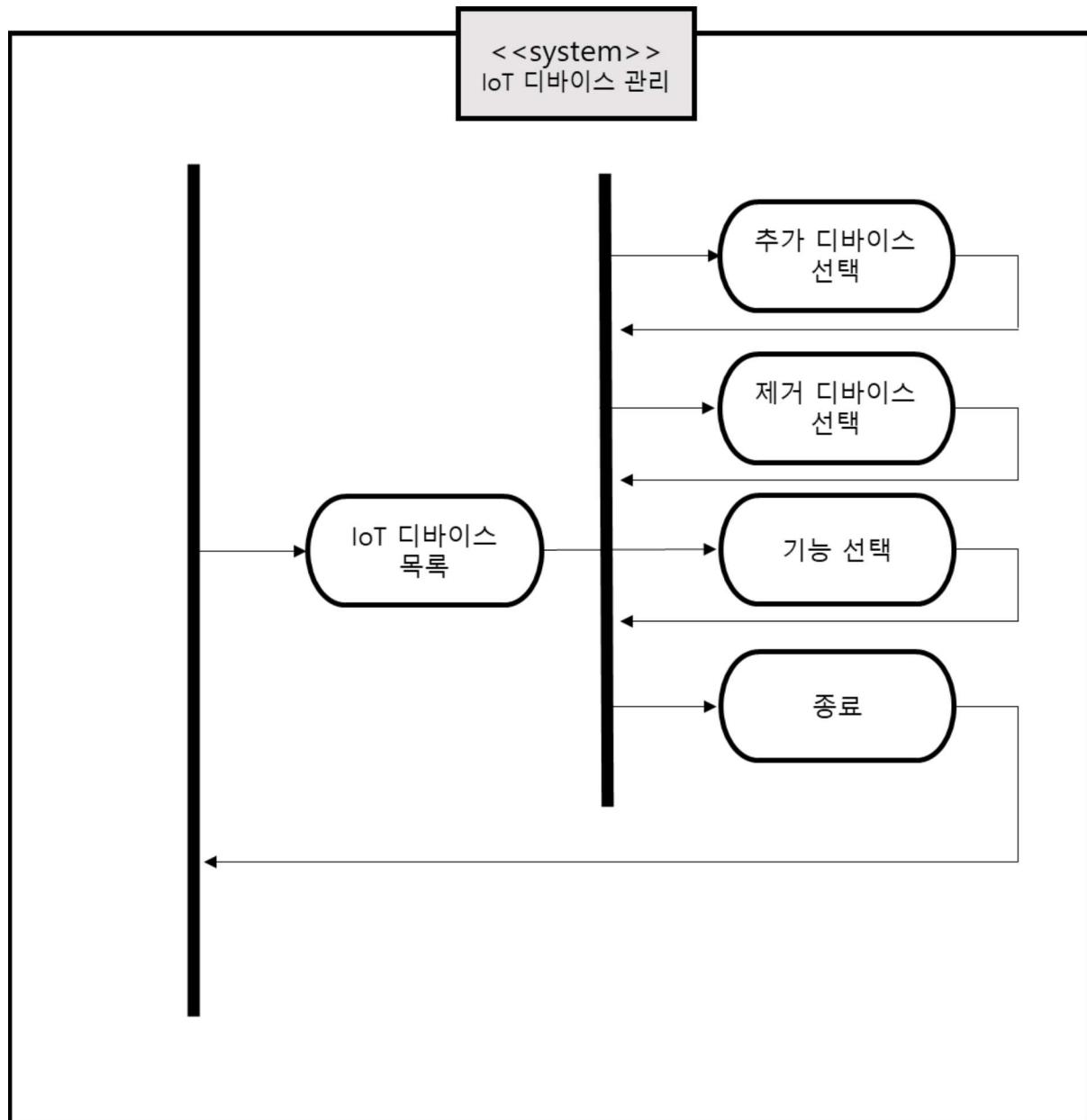


그림 7. Process Model - 반려동물 건강정보 수집/분석

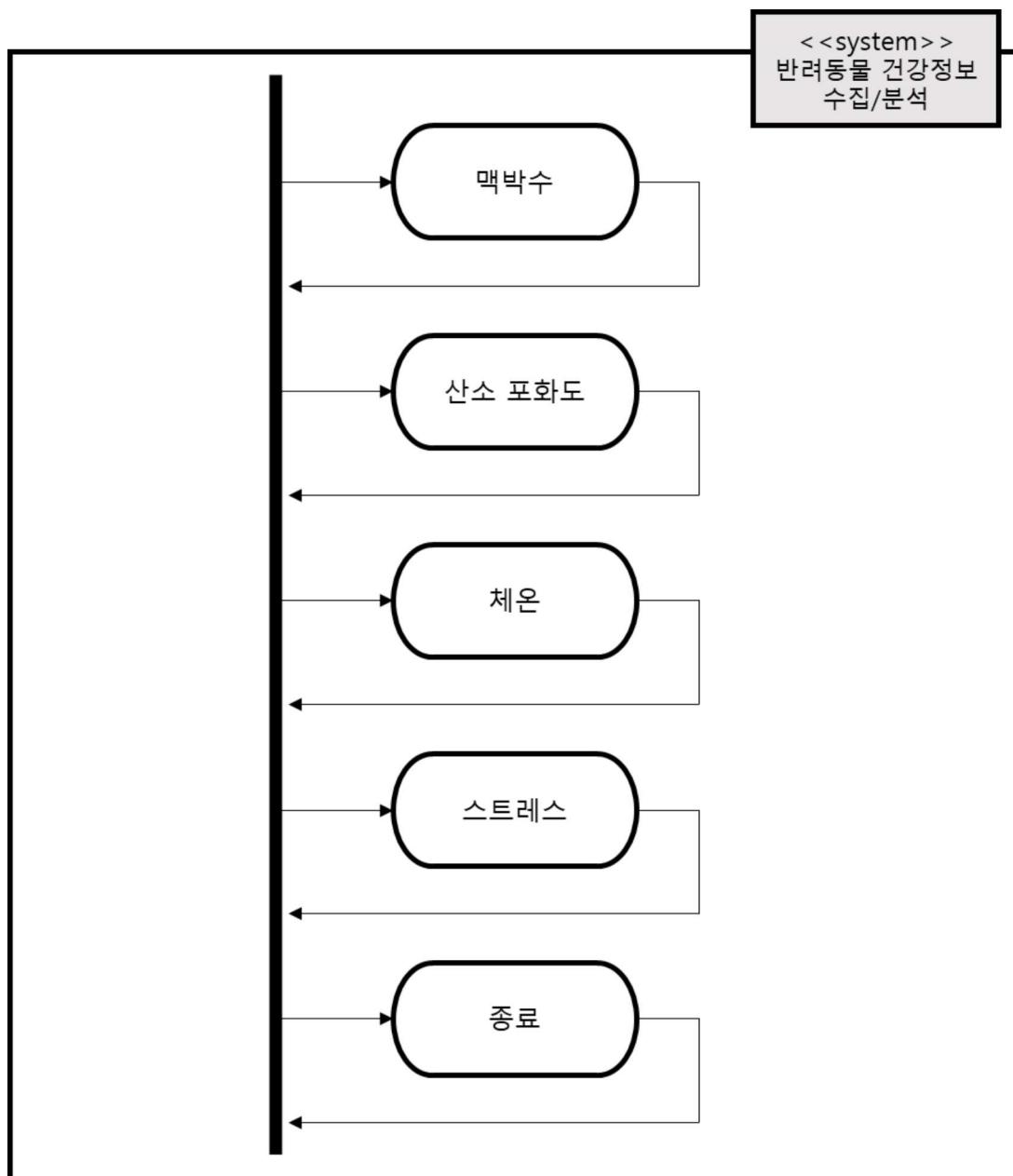


그림 8. Process Model - 반려동물 행동 로그 기록

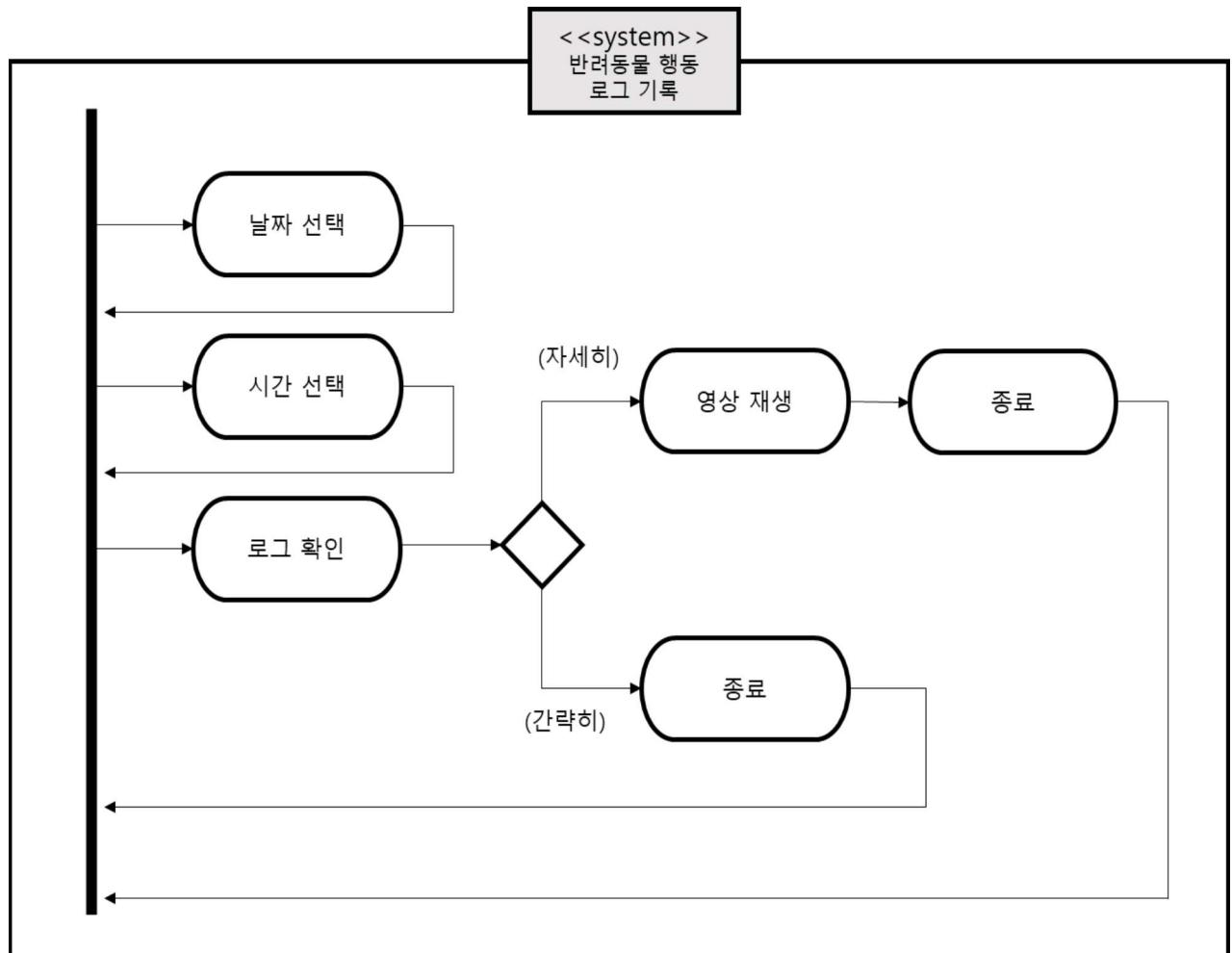


그림 9. Process Model - 반려동물 실시간 위치 파악

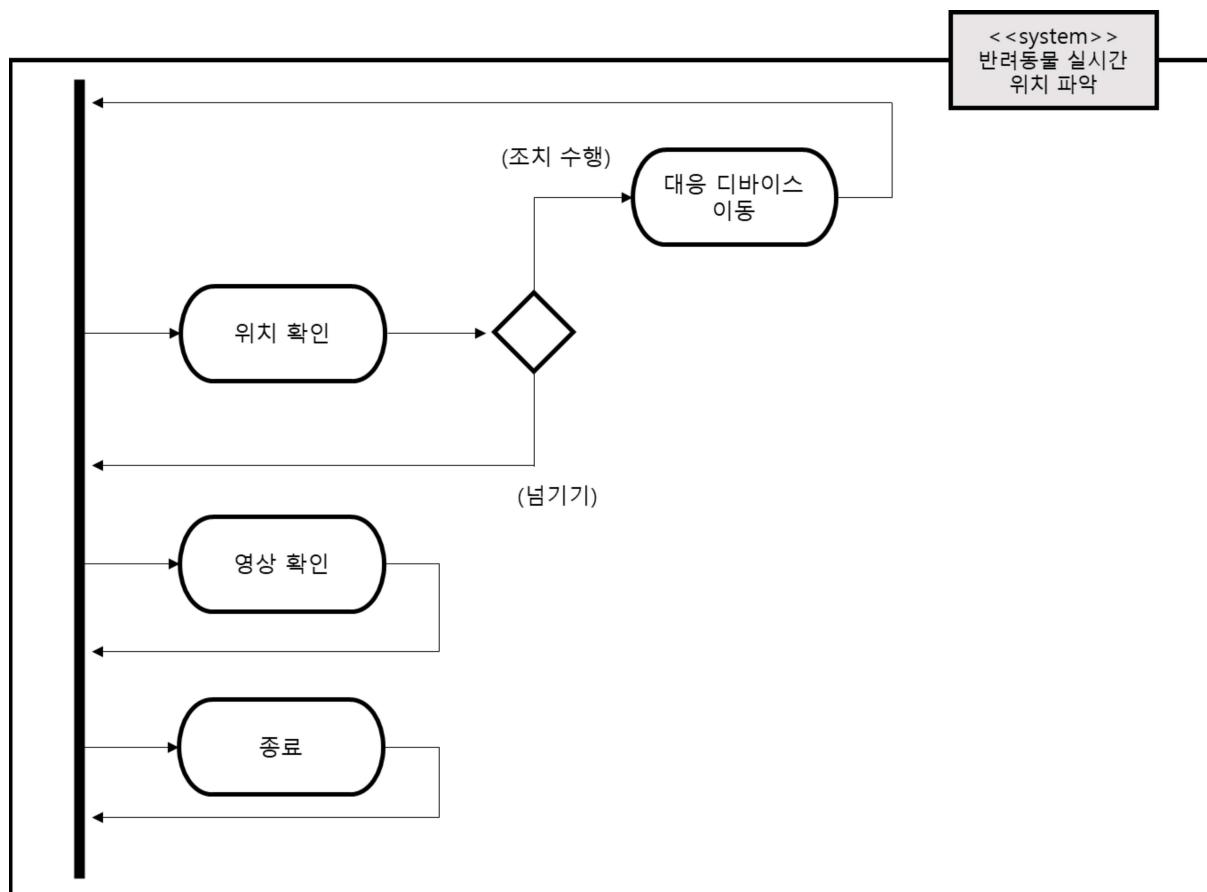
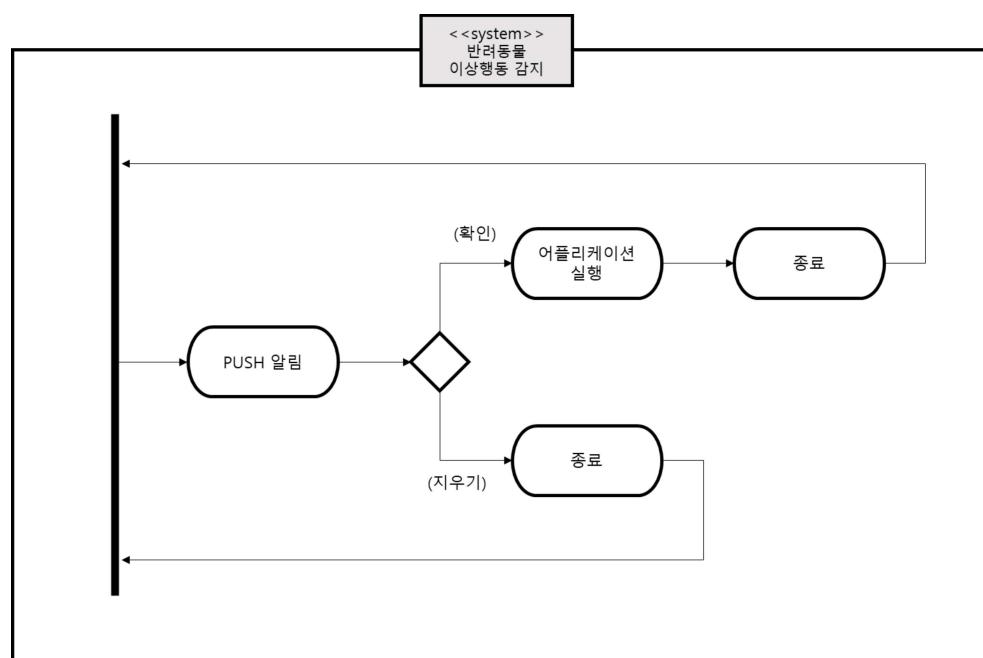


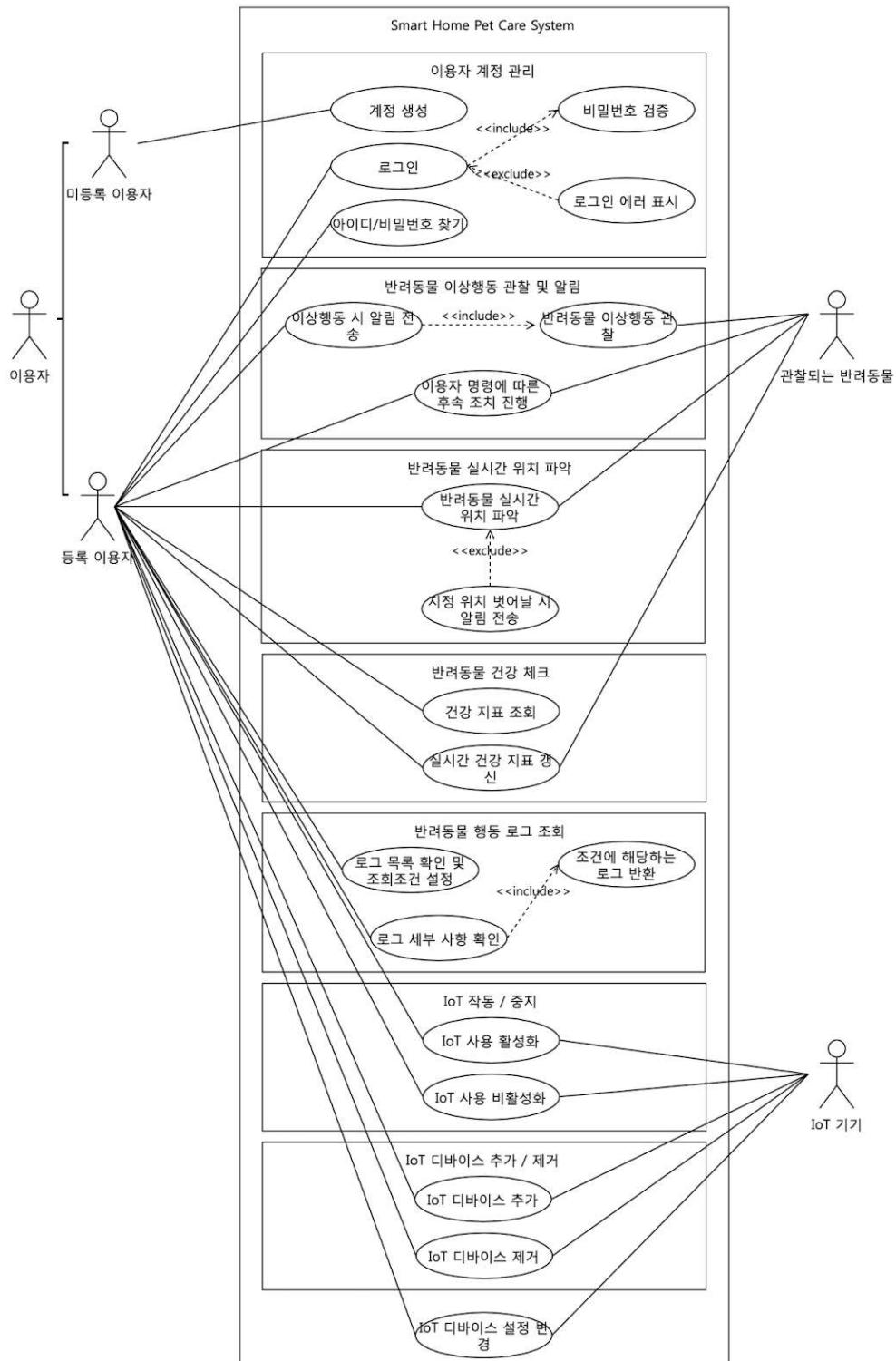
그림 10. Process Model - 반려동물 이상행동 감지



3.6.3. Interaction Model

Use case diagram을 통해 시스템에 참여하는 당사자 및 기기와 시스템 간의 관계를 확인할 수 있다.

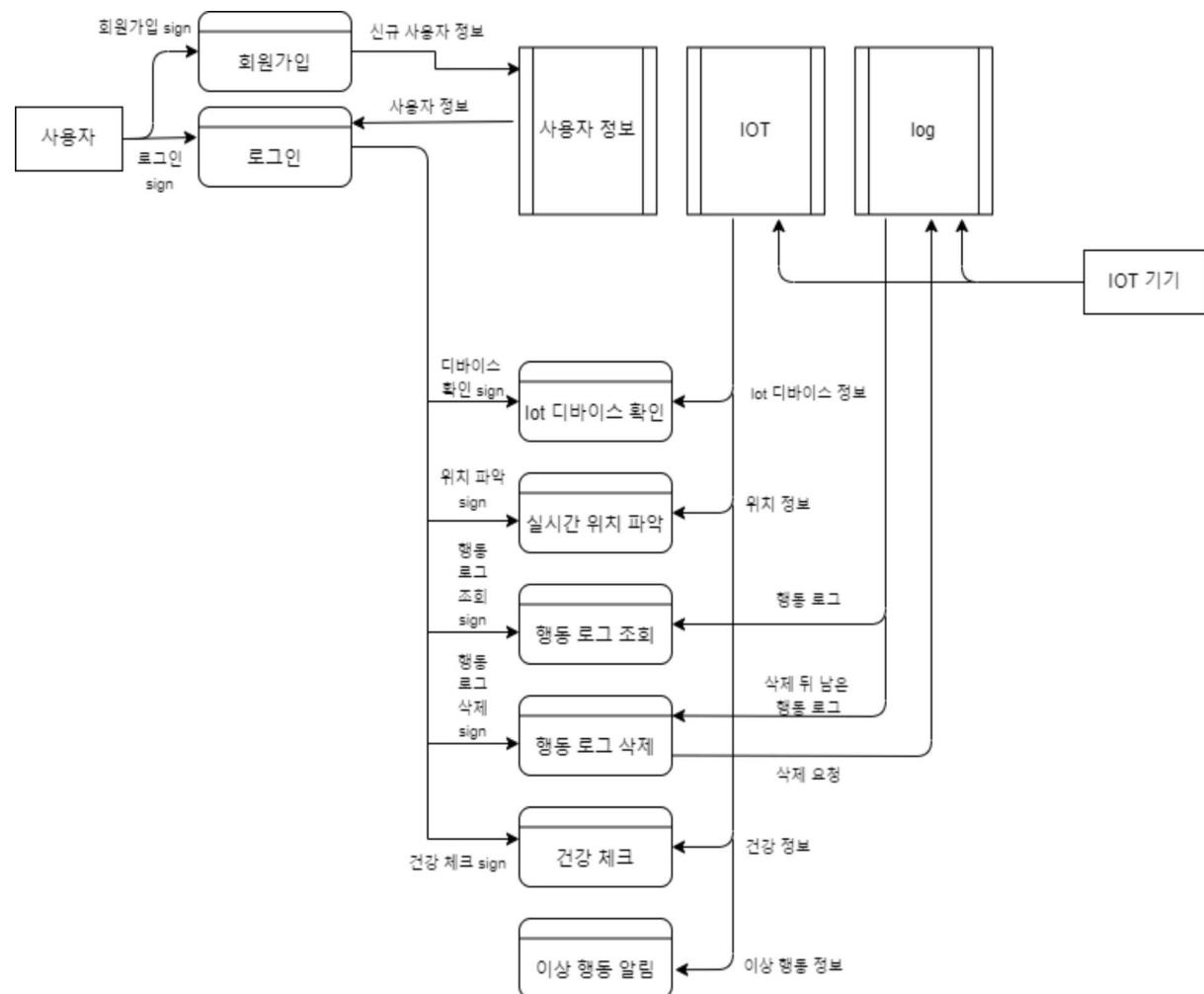
그림 11. Use Case Diagram



3.6.4. Behavior Model

3.6.4.1. Data Flow Diagram

그림 12. Data Flow Diagram



3.6.4.2. Sequence Diagram

그림 13. Sequence Diagram - 계정관리

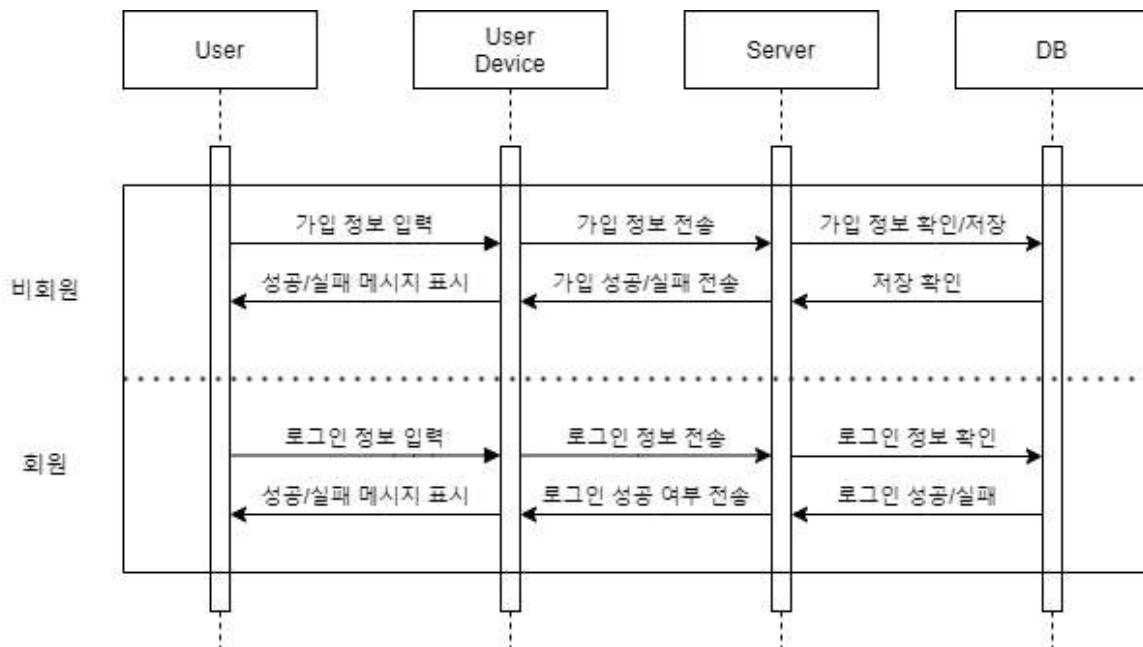


그림 14. Sequence Diagram - 이상행동 Push 알림

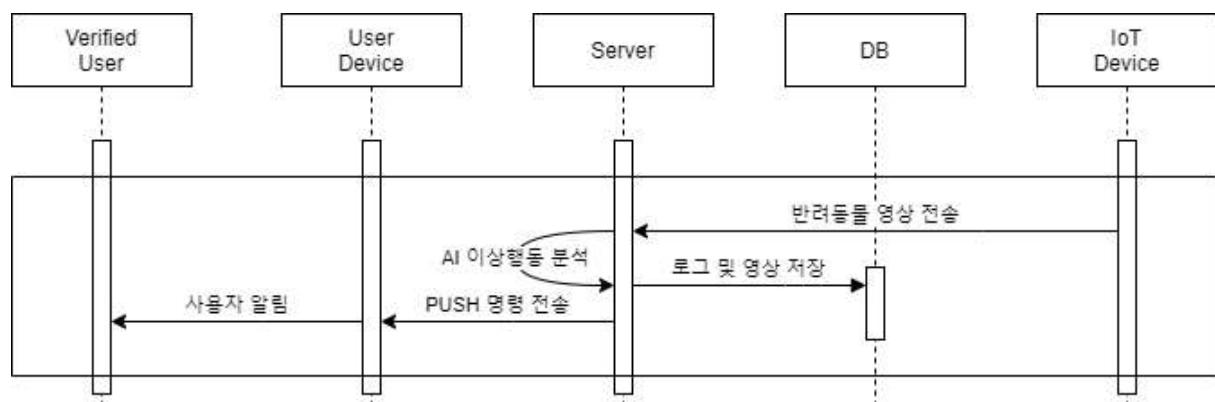


그림 15. Sequence Diagram - 반려동물 위치 정보

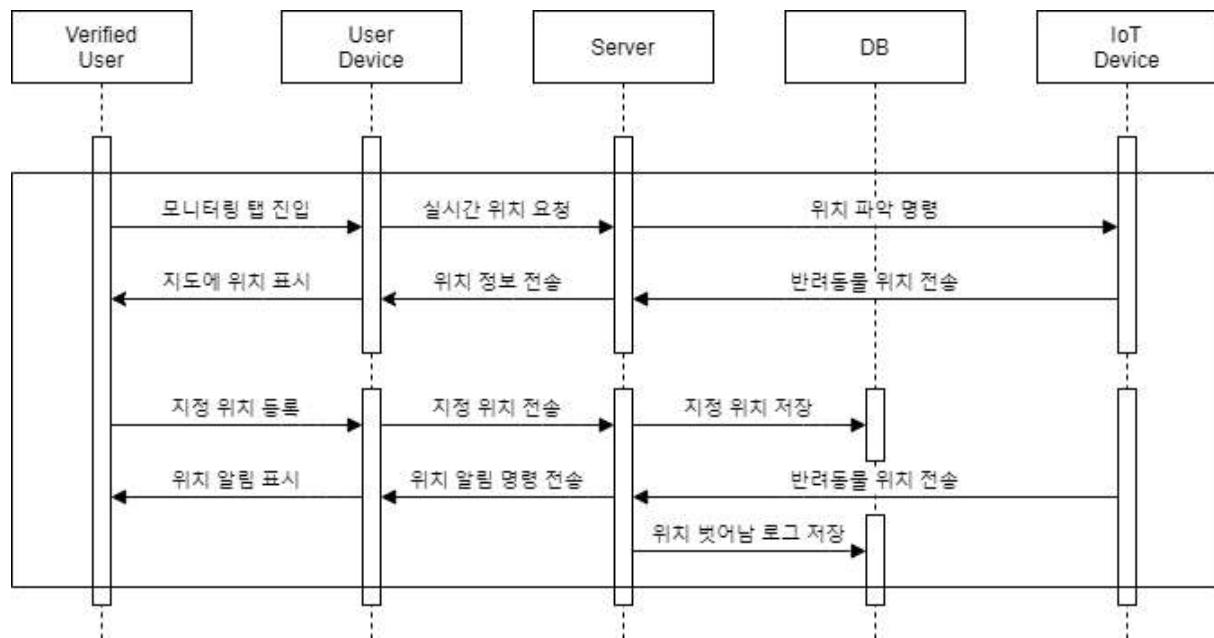


그림 16. Sequence Diagram - 반려동물 로그 조회

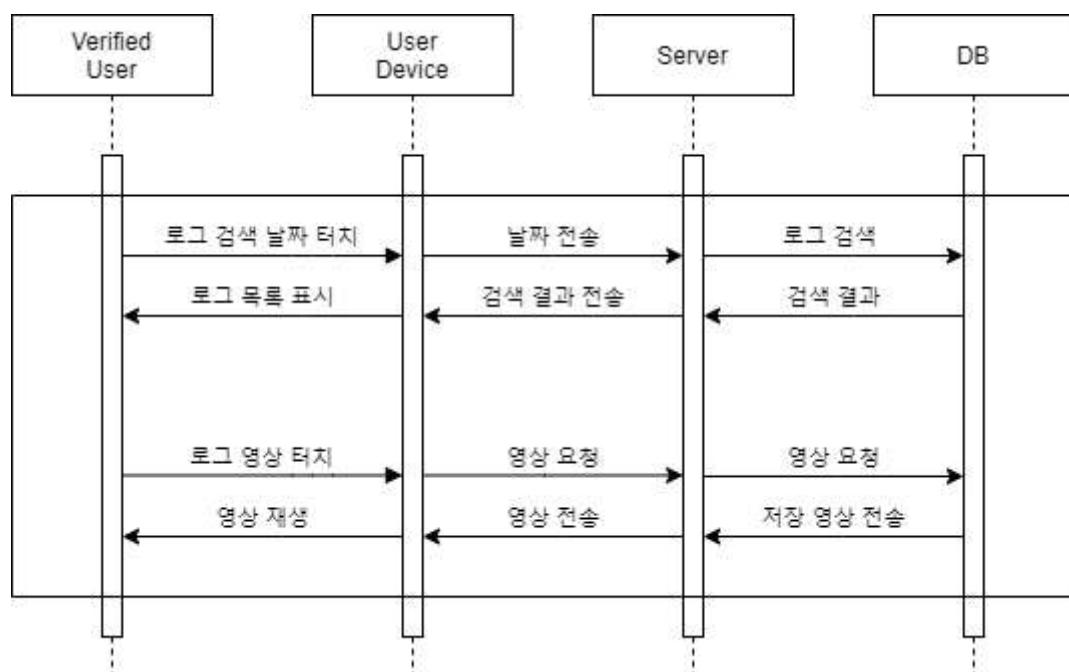


그림 17. Sequence Diagram - IoT 디바이스 관리

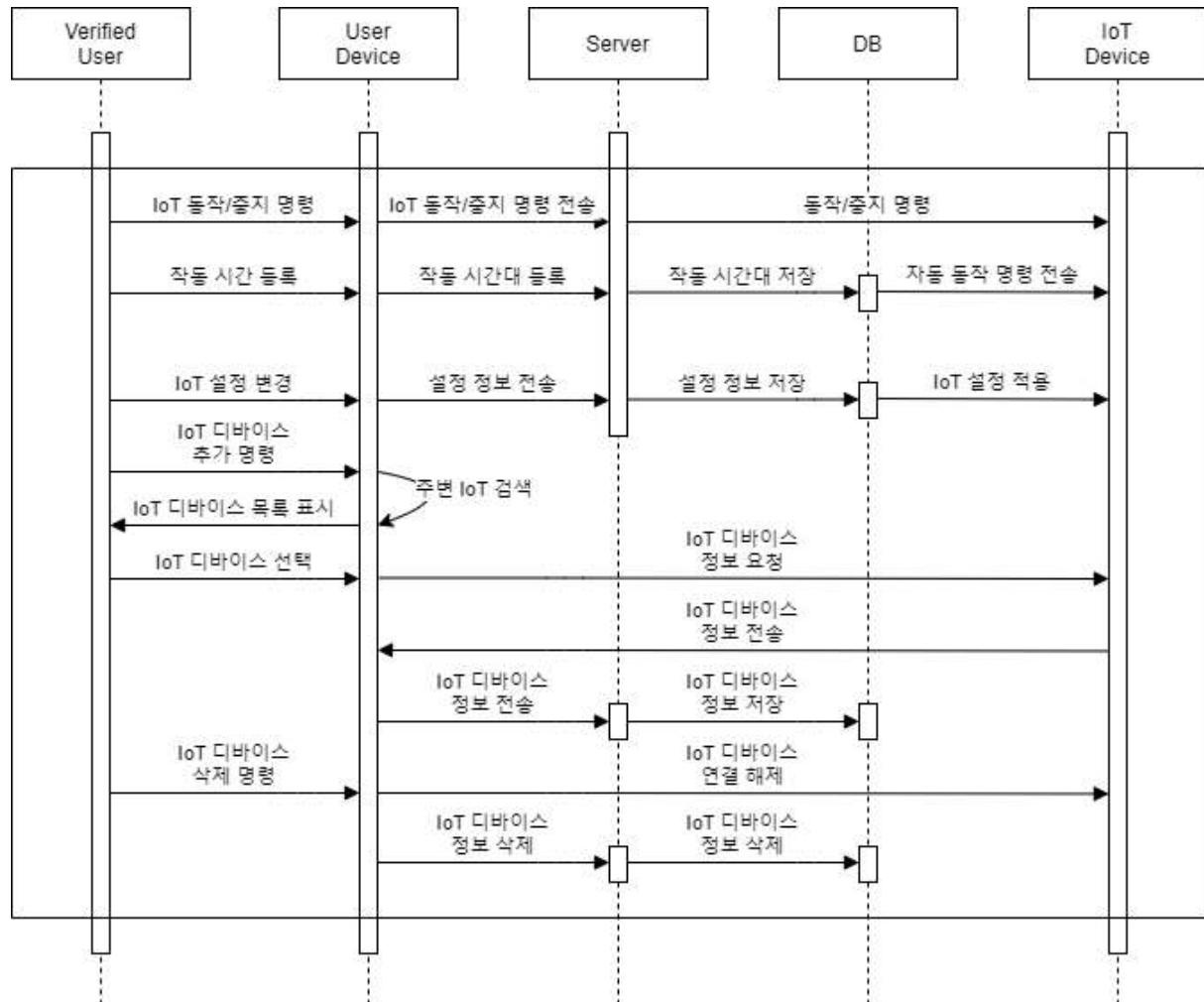
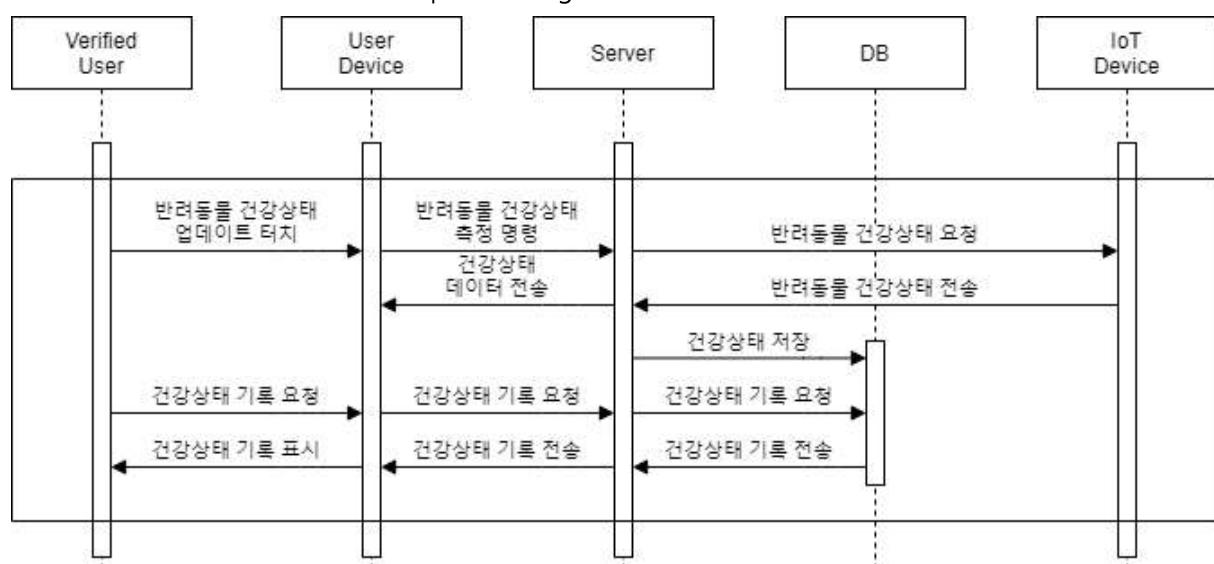
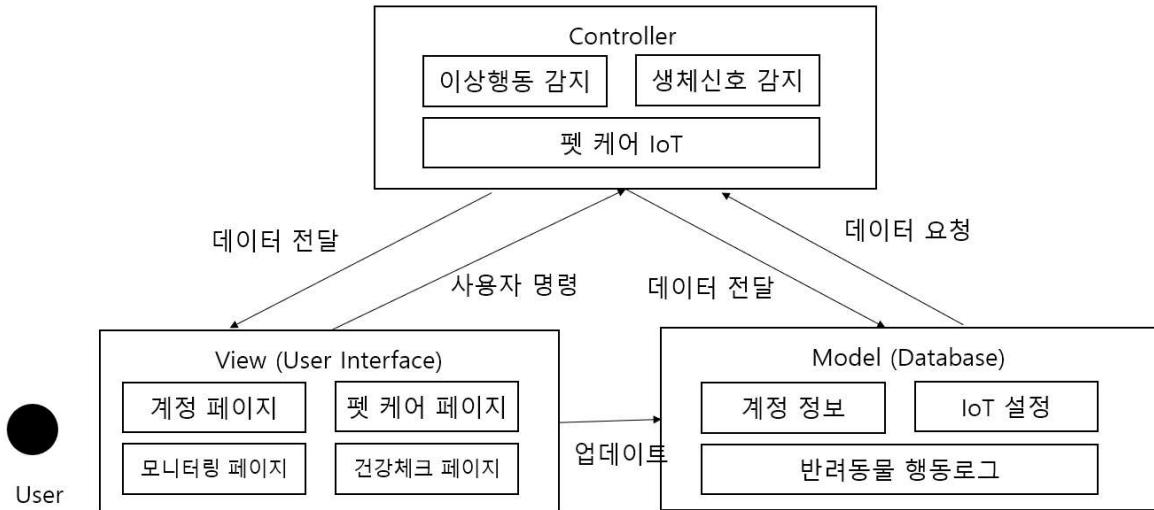


그림 18. Sequence Diagram - 반려동물 건강상태 관리



3.7. System Architecture

그림 19. System Architecture



3.8. System Evolution

해당 장에서는 시스템의 한도와 기본 가정, 그리고 발전 가능성에 대해서 다룬다.

3.8.1. Limitation and Assumptions

본 명세서에서 제안하는 시스템은 자체 기기뿐만 아니라 사용자가 사용하던 기존 IoT 디바이스들을 관리 및 통제할 수 있는 기능을 제공하기 때문에 기기 자체 결함 등 외부적인 영향을 받을 수 있으며, 이에 따른 한계를 가진다. 또한 시스템의 반려동물 맞춤 케어 시스템 설정 서비스는 사용하고자 하는 외부 IoT 디바이스가 시스템과 호환된다는 가정 하에 이용가능하다. 시스템은 실시간 영상 데이터의 송/수신이 원활한 수준의 네트워크 환경을 가정하며, 네트워크 기반 서비스를 제공하기 때문에 사용자의 서비스 이용 시 네트워크 환경의 변화에 취약하다는 한계를 가진다.

3.8.2. Evolutions of Hardware and Software

시스템에서 사용하는 웨어러블 디바이스의 펌웨어 업데이트 등을 통해 성능이 개선될 수 있으며, 시스템과 호환되는 IoT 디바이스의 범위가 확장될 수 있다. IoT 디바이스에 대해 시스템에서 통제 및

관리 가능한 기능의 범위 또한 확장될 수 있다. IoT 디바이스의 기능 추가와 성능 업데이트에 대응하여 시스템에서 제공하는 서비스가 확장될 수 있다. 또한 반려동물 행동 분석에 사용되는 AI의 성능 개선을 통해 시스템의 반려동물 이상행동 감지 서비스의 기능이 개선될 수 있으며, 네트워크 환경 변화에 대응하여 안정적인 서비스 제공이 가능하도록 시스템이 개선될 수 있다.

4. Supporting Information

4.1. Software Requirement Information

이 요구사항 명세서는 IEEE Recommendation (IEEE Recommended Practice for Software Requirements Specifications, IEEE-Std-830). 서식에 따라 제작되었다.

4.2. Document History

| Date | Version | Description | Writer |
|------------|---------|--|--------|
| 2022/04/25 | 0.1 | 문서작업 시작 | 전체 |
| 2022/04/26 | 0.2 | 반려동물 이상행동 감지 Product Function, User Interface, Use Case, Data dictionary 작성 | 최하늘 |
| 2022/04/28 | 0.3 | 반려동물 건강체크 인터페이스, Organization Requirements 내용 추가 | 양승찬 |
| 2022/04/28 | 0.4 | User Interface 부분 수정 | 양승찬 |
| 2022/04/28 | 0.5 | 반려동물 행동 로그, Product Requirement 작성 | 곽재원 |

| | | | |
|------------|------|---|-----|
| 2022/04/28 | 0.6 | User Interface - IoT 디바이스 등록 및 설정 표 추가 | 김준식 |
| 2022/04/28 | 0.7 | 펫 케어 시스템 User Interface, User requirements 작성, Introduction 작성 | 차현묵 |
| 2022/04/28 | 0.8 | External Requirements, System Evolution 작성 | 최하늘 |
| 2022/04/28 | 0.9 | 사용자 계정 관리 User interface, Use case, Data dictionary 작성 | 이경돈 |
| 2022/04/28 | 0.10 | 실시간 위치 파악 User interface, Use Case, Data dictionary 작성 | 유새하 |
| 2022/04/29 | 0.11 | Product Perspective 작성 | 김준식 |
| 2022/04/29 | 0.12 | Hardware, software Interfaces 작성 Data Dictionary 수정 | 차현묵 |
| 2022/04/29 | 0.13 | Context Model 추가, External Requirements 수정 | 최하늘 |
| 2022/04/29 | 0.14 | 사용자 계정 관리 Communication interface 작성 | 이경돈 |
| 2022/04/30 | 0.15 | System Architecture 추가 Communication Interface 추가 작성 | 차현묵 |

| | | | |
|------------|------|---|-----|
| 2022/04/30 | 0.16 | Sequence Diagram 작성 Communication Interface 추가 작성 | 곽재원 |
| 2022/04/30 | 0.17 | User Interface - IoT 디바이스 등록 및 설정 화면 구현 추가 | 김준식 |
| 2022/05/01 | 0.18 | Process Model 작성 | 김준식 |
| 2022/05/01 | 0.19 | Data Flow Diagram 작성 | 유새하 |
| 2022/05/01 | 0.20 | Use Case Diagram 작성 | 양승찬 |
| 2022/05/01 | 1.0 | 전체 검토 | 전체 |
| 2022/05/01 | 1.1 | Process Model IoT 디바이스 관리 및 반려동물 실시간 위치 파악 수정 | 김준식 |
| 2022/05/01 | 1.2 | Performance Requirements iOS 버전 수정 | 김준식 |
| 2022/05/01 | 1.3 | Introduction, Overall Description 일부 맞춤법 교정 | 김준식 |