데이터베이스입문 Project #1

20141334 임재현

1. Entity 생성, Attributes 도출, Key 도출

1.1 User

* 사이트는 회원 정보를 받아 관리해야 한다. 회원 정보에는 ID, 이름, 연령대, 성별, 거주지역, 가입일, 가입 그룹이 있어야 한다. 따라서 회원이라는 entity를 생성, User 로 명명했다.
* 회원 정보에는 ID, 이름, 연령대, 성별, 거주지역, 가입일, 가입 그룹이 있어야 한다. 이에 따라 회원 정보에 포함되어야 하는 항목들이 각각 User entity의 Attribute가 될 것이다. 이들을 각각 아래와 같이 지정했다. 가입 그룹의 경우 1.2에서 Group이라는 Entity를 생성할 것이므로 Foreign Key로 지정 하려 한다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 번호 | Name | Attribute name  (in ERwin) | Domain | 내용 |
| 1 | ID(PK) | User\_id | ID | 회원의 ID를 저장 |
| 2 | Name | User\_name | Name | 회원의 이름을 저장 |
| 3 | Age | User\_age | Age | 회원의 연령대를 저장 |
| 4 | Gender | User\_gender | Gender | 회원의 성별을 저장 |
| 5 | Address | User\_address | Address | 회원의 거주지역을 저장 |
| 6 | Joindate | User\_joindate | Date | 회원의 가입일을 저장 |

* 각 User가 구분되어야 한다. User의 ID는 고유하므로 이를 Primary Key 로 지정했다.

1.2 Group

* 사이트는 회원들이 모여 있는 그룹에 대한 정보를 관리한다. 그룹 정보에는 ID, 이름, 구성원, 활동 지역, 개설일에 대한 정보가 있어야 한다. 따라서 그룹이라는 entity를 생성, Group 으로 명명했다.
* 그룹 정보에는 ID, 이름, 구성원, 활동 지역, 개설일에 대한 정보가 있어야 한다. 이에 따라 그룹 정보에 포함되어야 하는 항목들이 각각 Group entity의 Attribute가 될 것이다. 이들을 각각 아래와 같이 지정했다. 그룹 구성원의 경우 이미 1.1에서 User의 가입 그룹을 Group에서 Foreign Key로 지정했으므로, 비식별관계의 자식 entity가 될 것이다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 번호 | Name | Attribute name  (in ERwin) | Domain | 내용 |
| 1 | ID(PK) | Group\_id | ID | 그룹의 ID를 저장 |
| 2 | Name | Group\_name | Name | 그룹의 이름을 저장 |
| 3 | Address | Group\_address | Address | 그룹의 활동 지역을 저장 |
| 4 | Opendate | Group\_opendate | Date | 그룹의 개설일을 저장 |

* 각 Group이 구분되어야 한다. Group의 ID는 고유하므로 이를 Primary Key로 지정했다.

1.3 Cat

* 사이트는 고양이들을 관리한다. 고양이들은 고유 ID를 가지며, 사이트는 고양이의 이름, 사진, 성별, 나이, 서식지, 성격, 사람에 대한 경계, 털빛, 중성화 여부, 병력, 기타 특이사항 등의 상세정보를 포함해야 한다. 따라서 고양이라는 entity를 생성, Cat 으로 명명했다.
* 고양이 정보에는 ID, 이름, 사진, 성별, 나이, 서식지, 성격, 사람에 대한 경계, 털빛, 중성화 여부, 병력, 기타 특이사항이라는 정보가 있어야 한다. 이에 따라 고양이 정보에 포함되어야 하는 항목들이 각각 Cat entity의 Attribute가 될 것이다. 이들을 각각 아래와 같이 지정했다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 번호 | Name | Attribute name  (in ERwin) | Domain | 내용 |
| 1 | ID(PK) | Cat\_id | ID | 고양이의 ID를 저장 |
| 2 | Name | Cat\_name | Name | 고양이의 이름을 저장 |
| 3 | Picture | Cat\_picture | Picture | 고양이의 사진을 저장 |
| 4 | Gender | Cat\_gender | Gender | 고양이의 성별을 저장 |
| 5 | Age | Cat\_age | Age | 고양이의 나이를 저장 |
| 6 | Address | Cat\_address | Address | 고양이의 서식지를 저장 |
| 7 | Personality | Cat\_personality | String | 고양이의 성격을 저장 |
| 8 | Alert Level | Cat\_alevel | String | 고양이의 경계정도를 저장 |
| 9 | Color | Cat\_color | String | 고양이의 털빛을 저장 |
| 10 | Surgery | Cat\_surgery | Bool | 고양이의 중성화 여부를 저장 |
| 11 | History | Cat\_history | String | 고양이의 병력을 저장 |
| 12 | Etc | Cat\_etc | String | 고양이의 기타 특이사항을 저장 |

* 각 Cat이 구분되어야 한다. Cat의 ID는 고유하므로 이를 Primary Key로 지정했다.

1.4 Foodplace

* 사이트는 고양이들이 식사를 하는 급식소에 대한 정보를 관리한다. 급식소들은 고유 ID를 가지며, 위치, 담당자, 사용하는 고양이에 대한 정보를 포함해야 한다. 따라서 급식소라는 entity를 생성, Foodplace 로 명명했다.
* 급식소 정보에는 ID, 위치, 담당자, 사용하는 고양이라는 정보가 있어야 한다. 이에 따라 급식소 정보에 포함되어야 하는 항목들이 각각 Foodplace entity의 Attribute가 될 것이다. 이들을 각각 아래와 같이 지정했다. 급식 담당자의 경우 1.5 에서 Schedule entity를 생성할 것이므로 Foreign Key로 지정 하려 한다. 또한 급식소를 사용하는 고양이는 1.3 Cat entity에서 고양이의 서식지가 급식소의 위치와 일치하면 사용한다는 가정 하에 Foreign Key로 지정 하려 한다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 번호 | Name | Attribute name  (in ERwin) | Domain | 내용 |
| 1 | ID(PK) | Foodplace\_id | ID | 급식소의 ID를 저장 |
| 2 | Address | Foodplace\_address | Address | 급식소의 위치를 저장 |

* 각 Foodplace가 구분되어야 한다. Foodplace의 ID는 고유하므로 이를 Primary Key로 지정했다.

1.5 Food

* 각 그룹은 사용하는 사료의 재고를 관리한다. 사료의 ID, 사료명, 사료 구입처, 구입 가격, 사료의 잔량, 사료의 위치에 대한 정보를 포함해야 한다. 따라서 사료라는 entity를 생성, Food 로 명명했다.
* 사료 정보에는 ID, 사료명, 구입처, 구입 가격, 잔량, 위치에 대한 정보가 있어야 한다. 이에 따라 사료 정보에 포함되어야 하는 항목들이 각각 Food entity의 Attribute가 될 것이다. 이들을 각각 아래와 같이 지정했다.
* Supplier-and-Parts database를 참고해, 사료 자체의 정보와 직접적인 활동내용을 더해 relation의 형태로 entity를 추가하기로 한다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 번호 | Name | Attribute name  (in ERwin) | Domain | 내용 |
| 1 | ID(PK) | Food\_id | ID | 사료의 ID를 저장 |
| 2 | Name | Food\_name | Name | 사료의 이름을 저장 |
| 3 | Seller | Food\_seller | String | 사료의 구매처를 저장 |
| 4 | Price | Food\_price | Price | 사료의 구입 가격을 저장 |

* 각 Food가 구분되어야 한다. Food의 ID는 고유하므로 이를 Primary Key로 지정했다.

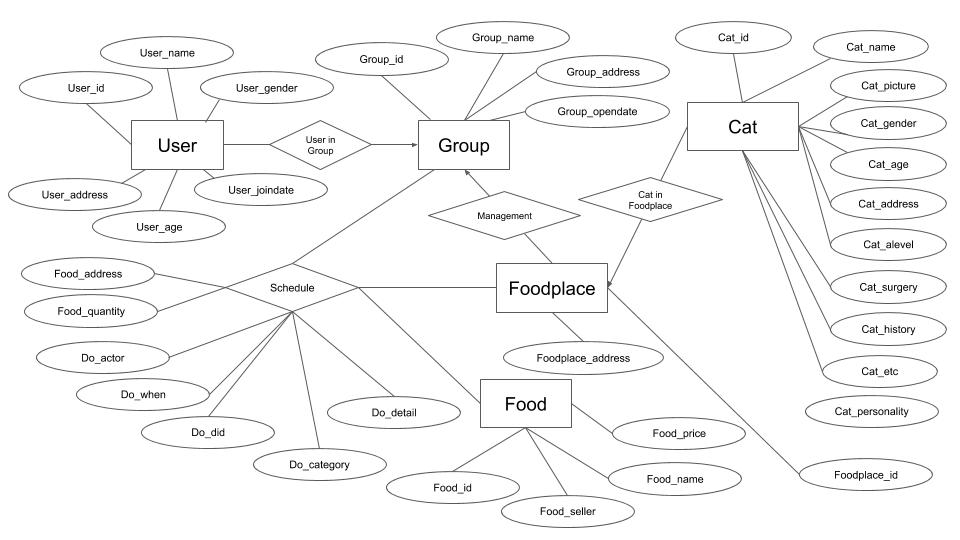
1.6 Schedule

* 운영 시스템에서 급식소의 담당 인원, 담당 요일, 담당 시간, 수행 여부를 확인 할 수 있어야 한다. 따라서 급식 스케줄이라는 entity를 relation의 형태로 생성, Schedule 로 명명했다.
* 스케줄 정보에는 담당 인원, 담당 요일, 담당 시간, 수행 여부에 대한 정보가 있어야 한다. 이에 따라 스케줄 정보에 포함되어야 하는 항목들이 각각 Schedule entity의 Attribute가 될 것이다. 이들을 각각 아래와 같이 지정했다.
* 또한 사용자에게 제공되어야 하는 정보를 감안해 Attribute를 추가했다.
* 급식소의 ID와 관리하는 그룹의 ID, 급식소에서 사용하는 사료의 ID를 Foreign Key로 받아야 한다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 번호 | Name | Attribute name  (in ERwin) | Domain | 내용 |
| 1 | Address | Food\_address | Address | 사료의 저장 위치를 저장 |
| 2 | Quantity | Food\_quantity | String | 사료의 잔량을 저장 |
| 3 | Actor | Do\_actor | Name | 담당자의 이름을 저장 |
| 4 | When | Do\_when | Datetime | 담당 요일과 시간을 저장 |
| 5 | Did | Do\_did | Bool | 활동 여부를 저장 |
| 6 | Category | Do\_category | String | 활동 분류를 저장 |
| 7 | Act | Do\_act | String | 활동 내역을 저장 |
| 8 | Detail | Do\_detail | String | 특이사항을 저장 |

2. Logical Mode, ER diagram

2.1 ER diagram

* 1에서 생성한 Entities 와 그 Attributes를 총괄하는 전체 다이어그램은 다음과 같다.

2.2 Relation & Cardinality 설정

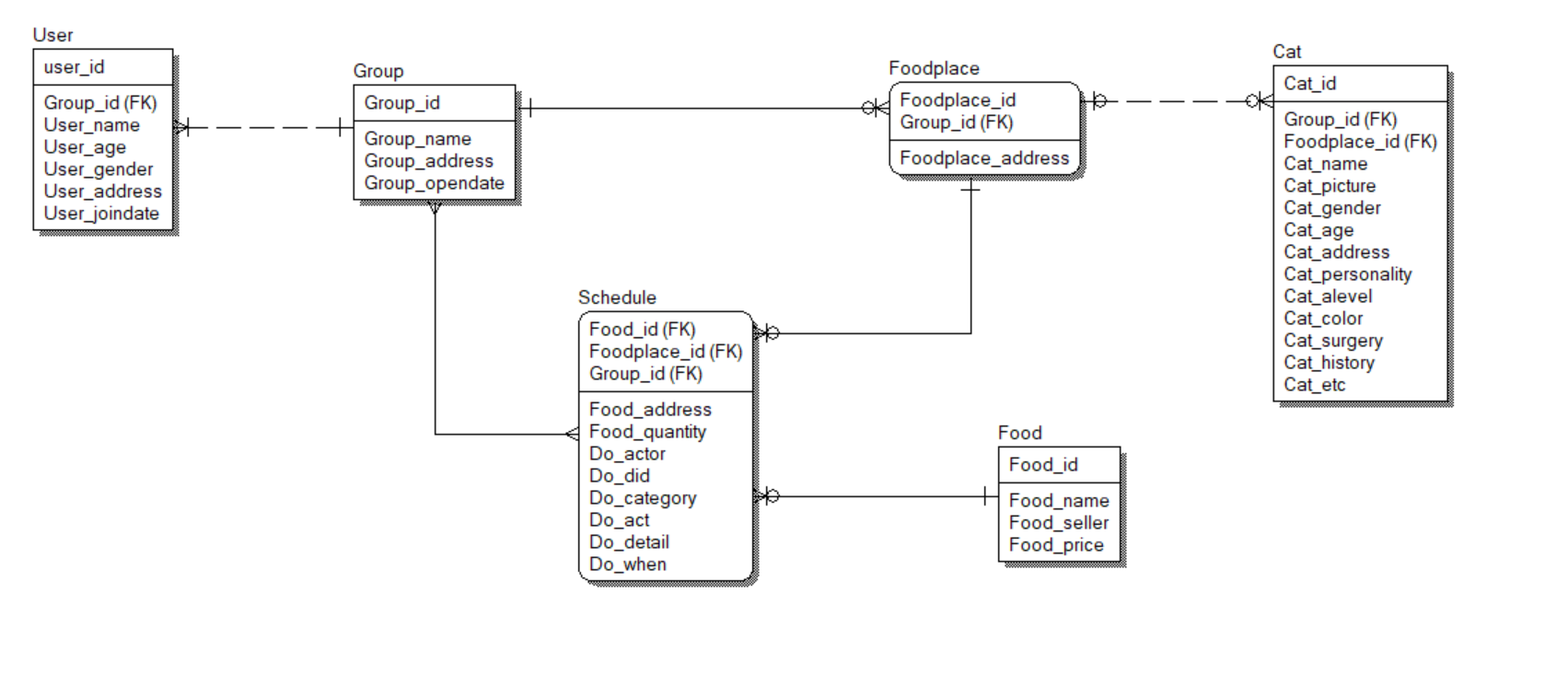
2.2.1 User in Group : User entity와 Group entity 사이의 Relationship으로, 사용자가 그룹에 가입할 경우 사용자의 정보를 관리할 수 있도록 설정하였다. 하나의 그룹에 다수의 사용자가 포함될 수 있으므로 1 : N의 Cardinality를 설정했다. 그룹에는 사용자가 최소 한명이 포함되어 있어야 할 것이므로 N을 one or more로 설정했다.

2.2.2 Cat in Foodplace : Cat entity와 Foodplace entity 사이의 Relationship으로, 고양이가 사용하게 될 급식소를 통해 고양이의 정보를 관리할 수 있도록 설정하였다. 하나의 급식소에 다수의 고양이가 올 수 있으므로 1 : N의 Cardinality를 설정했다. Cat은 Foodplace와는 독립적인 관계이며 따라서 비식별관계이고, Null도 허용한다(오지 않을 수 있으므로). 따라서 N을 zero, one or more로 설정했다.

2.2.3 Management : Foodplace entity와 Group entity 사이의 Relationship으로, 그룹이 담당하는 Foodplace를 관리할 수 있도록 설정하였다. 하나의 그룹이 여러 개의 Foodplace를 관리할 수 있도록 1 : N의 Cardinality를 설정했다. 그룹에서 관리하는 Foodplace는 없을 수도, 다수일 수도 있으므로 N을 zero, one or more로 설정했다. Foodplace는 관리할 Group이 없으면 생성되지 않으므로 식별관계일 것이다.

2.2.4 Schedule : 급식 시스템 스케줄을 운영, 관리할 수 있도록 Foodplace, Food, Group 사이에 Relationship을 설정했다. 각 entity 들이 다양한 조합으로 연결될 수 있으므로 Cardinality는 Many-to-Many 일 것이다.

2.3 Logical Mode

* Logical Mode로 생성한 Entity 사이의 관계는 다음과 같다.

3. Domain 설정

3.1 ID

* ID에 쓰이는 Domain은 String이며, Primary Key가 되므로 Not Null이다.

3.2 Name

* Name에 쓰이는 Domain은 String이며, 이름은 항상 존재하므로 Not Null이다.

3.3 Bool

* Bool의 경우, Yes/No에 대한 값을 출력하므로 String type이며, 수술 여부나 수행 여부는 항상 존재해야 하므로 Not Null이다.

3.4 Datetime

* 가입일/생성일의 Datetime의 경우, 연도/월/일에 대한 데이터를 가지고 있어야 하므로 Domain은 DATE이다.
* Schedule에서 사용할 Datetime의 경우 시간(시, 분, 초) 데이터를 포함해야 하므로 INTERVAL DAY TO SECOND type으로 지정한다.

3.5 Age

* Age의 경우, 고양이의 나이에 쓰일 Age는 제한이 없는 수를 가져야 하므로 INT type으로, User에 사용되는 Age는 연령대이므로 10대, 20대, 30대, 40대, 50대 이상 이라는 제한적인 값을 가지게 하려 한다. 고양이의 나이에 쓰이는 Age 는 특정하기 어려울 수 있으므로 Null 을 허용하고, User의 경우 가입할 때 정보를 기입하므로 Not Null이다.

3.6 Gender

* Gender의 경우, 성 정체성을 받기 위해 Domain을 String으로 지정한다. Not Null이다.

3.7 Address

* Address에 쓰이는 Domain은 String이며, Not Null이다.

3.8 Picture

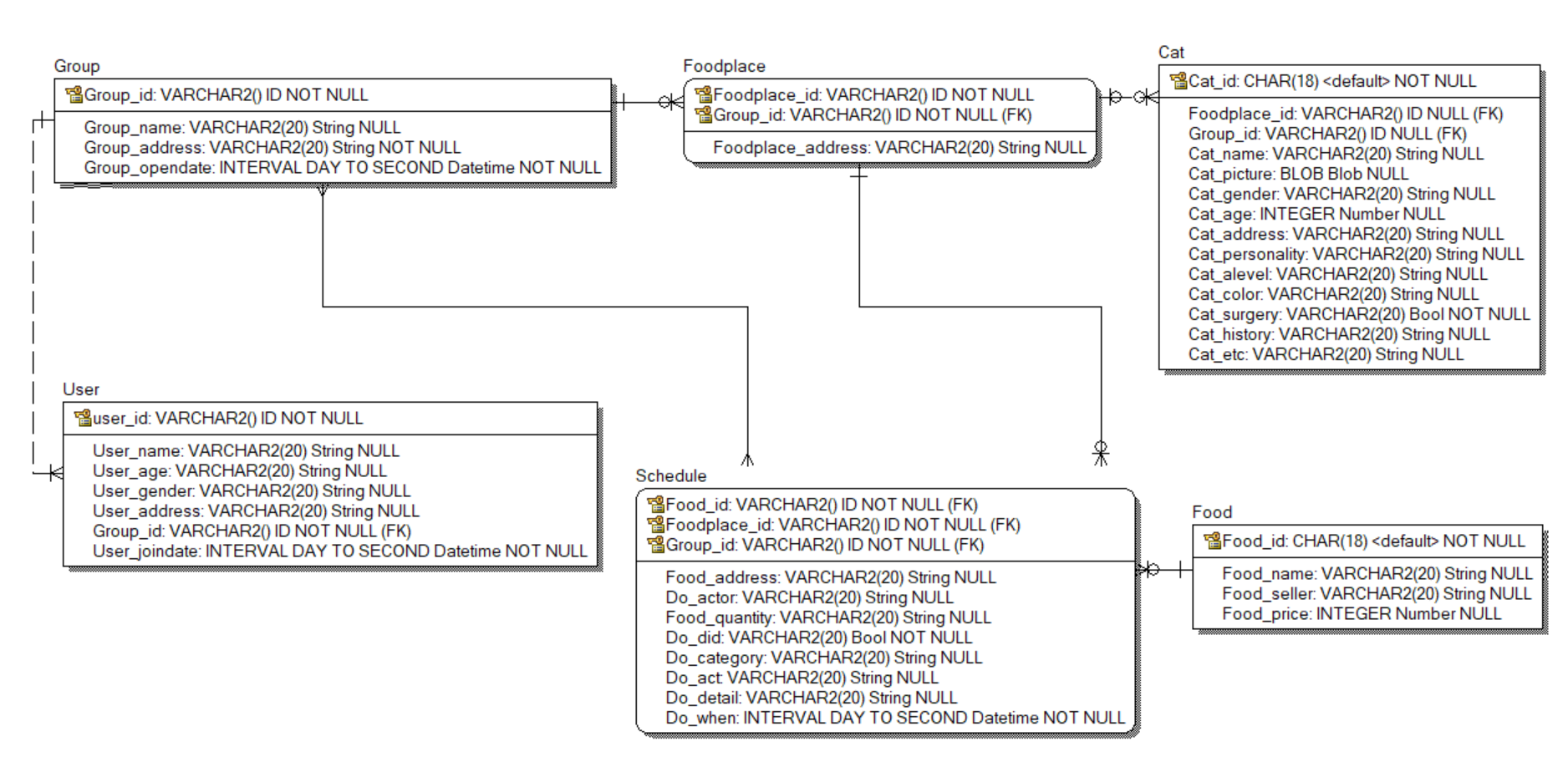
* Picture의 경우 사진을 저장해야 하므로, 이진 데이터 타입 BLOB type으로 지정한다. NULL을 허용한다.

3.9 Price

* Price 의 경우 가격이므로 INT type으로 지정한다.

각 Domain에 대해 알맞게 Null/Not Null을 지정하였다.

4. Physical Mode

- Physical Mode 로 구현한 전체 데이터베이스의 모습은 다음과 같다.

5. 설계에 대한 세부 요구사항 지정

5.1 서강 고양이 돌봄 시스템에서 사용하는 정보 시스템의 요구사항

- 각각 회원 정보는 User, 그룹 정보는 Group, 사료 정보는 Food, 고양이 정보는 Cat, 급식소 정보는 Foodplace entity에서 확인하며, 스케줄 정보는 Schedule Relationship에서 관리한다.

5.2 사용자에게 제공하는 고양이 돌봄 시스템의 요구사항

- 사용자에게 회원가입 신청서를 양식을 통해 받게 되면, 이 정보를 User entity에서 가입하며, 마찬가지로 관리 및 수정한다.

5.3 그룹에게 제공하는 고양이 돌봄 시스템의 요구사항

- 그룹 정보 관리 및 조회는 Group entity에서 하게 되며, 활동 일자/활동 분류/활동 내역/담당자/상세내역을 각각 Schedule Relationship의 Do\_when/category/act/actor/detail을 통해 확인할 수 있다.