

2021-2 소프트웨어보안

# 학식 알리미 시스템

3 조

20164028 이종원

20165196 김현진

20173595 임태균

20183907 모예림

20184887 박지균

## 목차

---

1. SW 요구사항 .....	3
1. 사용자 요구 사항 .....	3
2-1. 시스템 요구 사항 도출 .....	3
2-2. 시스템 요구 사항 .....	5
2. 시스템 테스트 .....	6
3. 시스템 모델링 .....	8
1. 클래스 다이어그램 .....	8
2. 시퀀스 다이어그램 .....	8
3. 활동 다이어그램 .....	11
4. 상태 다이어그램 .....	11
4. 시스템 신뢰성 .....	12

# 1. SW 요구사항

---

## 1. 사용자 요구 사항

<학식 알리미 시스템>은 캠퍼스와 식당별로 그 날의 학식 메뉴를 알려주고, 메뉴별로 만족도조사를 실시하고 랭킹을 매겨 제공함으로써 이용자의 선택을 돕는다.

### 2-1. 시스템 요구 사항 도출

#### 1. 사용자 스토리

##### 스토리 1.

푸앙이는 중앙대 흑석캠퍼스 학생으로, 오늘의 점심 메뉴를 선택하려 한다, "서울캠퍼스"의 "중식"을 선택한 뒤, 모든 식당의 메뉴들을 확인한다. 다이어트를 하는 푸앙이는 칼로리 등의 정보도 알고 싶어한다. 또한 메뉴별 랭킹을 참고하여 오늘의 점심을 선택하고 싶어한다. 밥을 다 먹은 이후에는 만족도 조사에 직접 참여해 먹은 메뉴의 점수를 매기고, 이가 반영된 랭킹을 확인한 뒤 프로그램을 종료한다.

##### 스토리 2.

안성캠퍼스에 다니고 있는 고니는 친구인 푸앙이를 만나기 위해 흑석캠퍼스에 찾아가려 한다. 안성에서 점심을 먹을지, 흑석에 가서 점심을 먹을지 고민하다 각 캠퍼스별로 메뉴를 확인한다. 또한 캠퍼스를 오가다 보니 운영시간이 언제인지, 학식이 어디에 있는지 잘 몰라 관련 정보를 간편하게 확인한 뒤 점심을 선택한다.

##### 스토리 3.

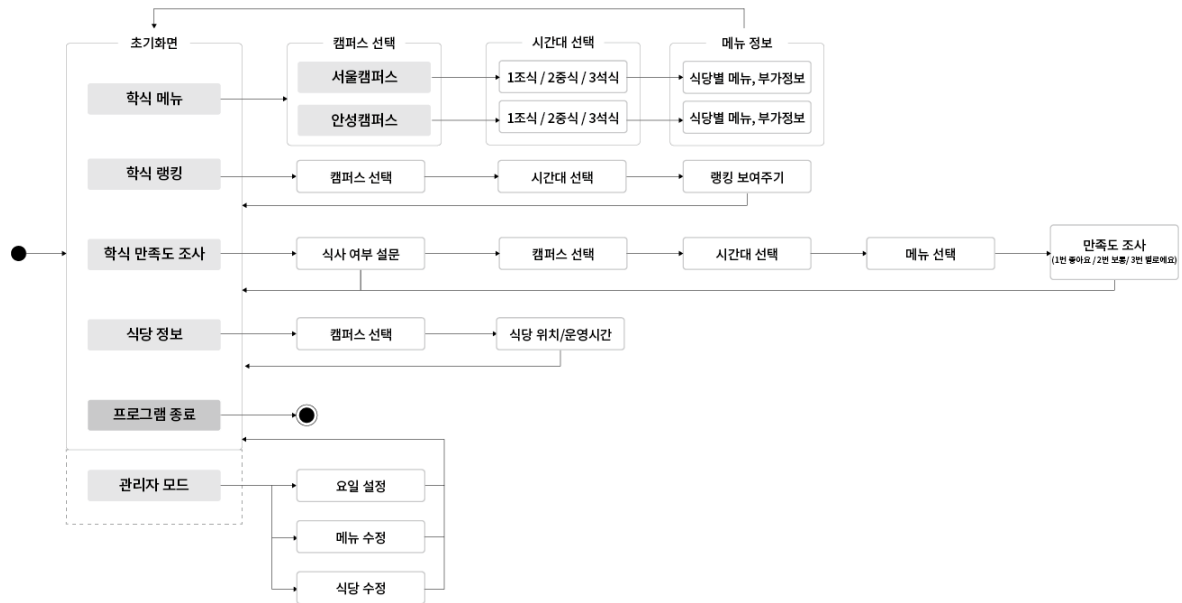
학식 관리자 김영양씨는 12 월의 학식정보 등록을 하려 한다. 특수 코드들을 입력해 관리자 모드에 입장한 뒤 11 월의 메뉴 정보를 삭제하고, 바뀌는 12 월의 메뉴들을 새로 입력한다. 마지막으로 바뀐 정보들을 이용자들이 볼 수 있도록 오늘의 요일 설정을 한 뒤 프로그램을 종료한다.

## 2. 사용자 시나리오

### 시작 상황에 대한 설명

- 학식 메뉴 / 랭킹 / 만족도 조사 / 식당 정보 / 프로그램 종료 탭을 나누어 사용자가 선택할 수 있게 한다. 특수 코드(999-banana-cat)를 입력하면 관리자 모드에 진입할 수 있다.

### 이벤트의 정상적인 흐름에 대한 설명



- [학식 메뉴] 에서 캠퍼스와 시간대를 선택하면, 해당 항목의 모든 식당별 메뉴들과 칼로리가 나열된다.
- [랭킹] 에서 캠퍼스와 시간대를 선택하면 해당 항목의 메뉴 랭킹을 확인할 수 있다.
- [만족도 조사] 에서는 사용자의 식사 여부를 설문하고, 먹은 사람에 한해 만족도 조사를 실시한다.
- [식당 정보]를 선택하면, 식당에 관련된 정보들을 확인할 수 있다.
- [관리자 모드]에서는 각 요일, 메뉴, 식당의 정보를 수정할 수 있다.

### 잘못됐을 때의 대처방안

- 지정된 값 외의 번호를 입력했을 때, 안내문을 띄우고 다시 입력하게 한다.

### 다른 동시 활동에 대한 정보

- 동시 활동이 없다.

### 시나리오 완료 시 상태에 대한 설명

- 초기 화면으로 돌아간다.

## 2-2. 시스템 요구 사항

### 기능적 요구사항

- 초기 화면에 각 메뉴들을 분리하여 원하는 항목에 바로 접근할 수 있게 한다.
- 캠퍼스, 시간대를 구분하여 메뉴 정보를 제공해야 한다.
- 식당별로 메뉴를 알려주되, 메뉴 외에도 칼로리나 가격 등의 추가 정보를 동시에 표기해야 한다.
- 단순한 숫자 버튼을 활용하여 다음 단계로 넘어갈 수 있게 한다.
- 이용자가 지정된 값 이외의 숫자를 입력했을 때는 다시 입력하게 한다.
- 단순한 만족도조사를 실시하여, 그 결과가 랭킹에 반영될 수 있도록 한다.
- 메뉴 데이터베이스를 분리하여 따로 관리할 수 있도록 한다.
- 분리된 메뉴 데이터베이스에 정보들과 만족도 점수를 저장할 수 있게 한다.
- 메뉴 외에도 식당 위치 및 운영시간 등의 공지를 알려줘야 한다.
- 관리자 모드를 두어 요일, 메뉴, 식당정보 등을 수정할 수 있게 해야 한다.
- 초기 화면에서 특수 코드를 입력하면 관리자 모드에 접근할 수 있게 한다.

### 비기능적 요구사항

- 이용자들에게 빠르고 간편하다는 인상을 주도록 한다.

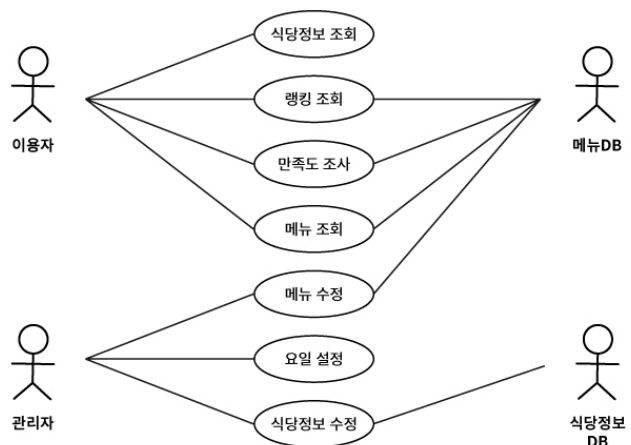
#### Actor



#### Use Cases



#### 유스케이스 다이어그램



## 2. 시스템 테스트

---

### 1-1. 학식 시간은 조식, 중식, 석식 중 하나가 돼야 한다.

Times 함수를 실행시키고, ans 에 1 or 2 or 3 을 입력했을 때 return 이 1 or 2 or 3 으로 제대로 나오지 않으면 오류이다.

```
1  import unittest
2  from choice import *
3
4  class TddTest(unittest.TestCase) :
5
6      def test_time(self):
7          time = times()
8
9          test_time = [1, 2, 3]
10
11         if time not in test_time:
12             print("time은",time)
13             print("Time should return 1, 2, 3 but it doesn't")
14         else:
15             print("정상적인 값을 입력받았습니다.")
16
```

### 1-2. 캠퍼스는 서울, 안성 캠퍼스 중 하나가 돼야 한다.

Campus 함수를 실행시키고, ans 에 1 or 2 를 입력했을 때 return 이 1 or 2 로 제대로 나오지 않으면 오류이다.

```
18     def test_campus(self):
19         cam = campus()
20
21         test_campus = [1, 2]
22
23         if campus not in test_campus:
24             print("cam은", cam)
25             print("Campus should return 1, 2 but it doesn't")
26
27
28 if __name__ == '__main__' :
29     unittest.main()
```

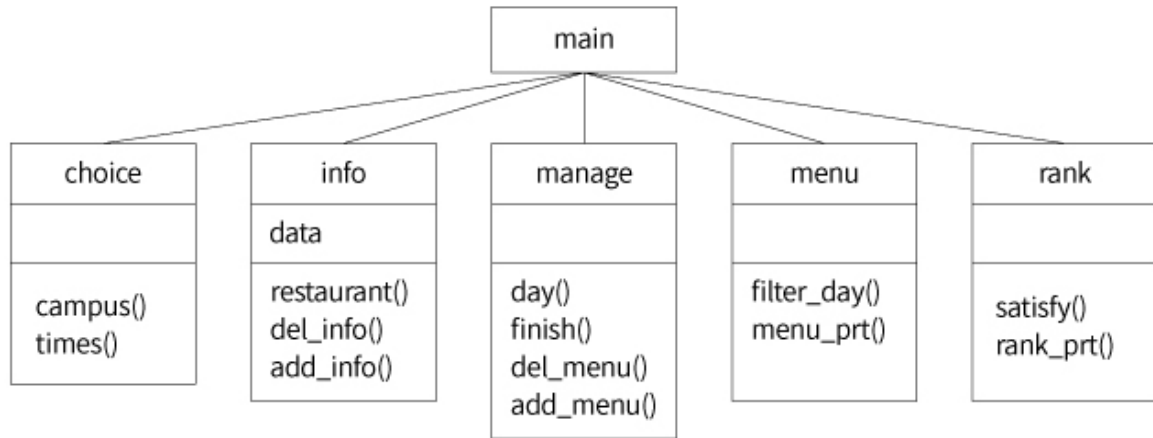
## 2. 만족도 조사는 별로예요, 보통이에요, 좋아요 중 하나가 돼야 한다.

Satisfy 함수를 실행시키고, ans 에 1 or 2 or 3 을 입력했을 때 return 이 1 or 2 or 3 으로 제대로 나오지 않으면 오류이다.

```
1  import unittest
2  from rank import *
3
4  class TddTest(unittest.TestCase) :
5
6      def test_satisfy(self):
7          satisfy_choice = satisfy(data)
8
9          test_satisfy = [1, 2, 3]
10
11         if satisfy_choice not in test_satisfy:
12             print("satisfy_choice은", satisfy_choice)
13             print("satisfy_choice should return 1, 2, 3 but it doesn't")
14         else:
15             print("정상적인 값을 입력받았습니다.")
```

### 3. 시스템 모델링

#### 1. 클래스 다이어그램

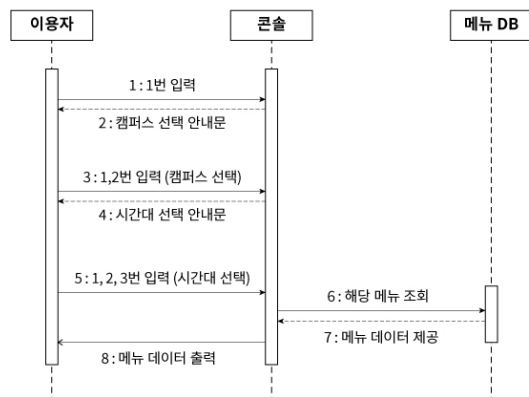


프로그램에 클래스를 사용하지 않고 함수만으로 구현하였기 때문에 모듈 사이의 연관성을 다이어그램으로 표시했다.

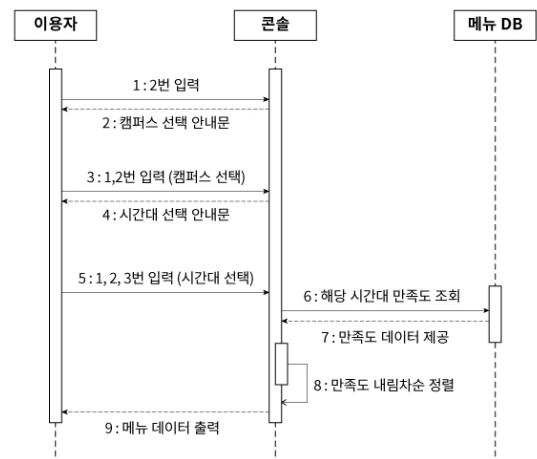
#### 2. 시퀀스 다이어그램

-이용자-

##### 1. 학식 메뉴 확인

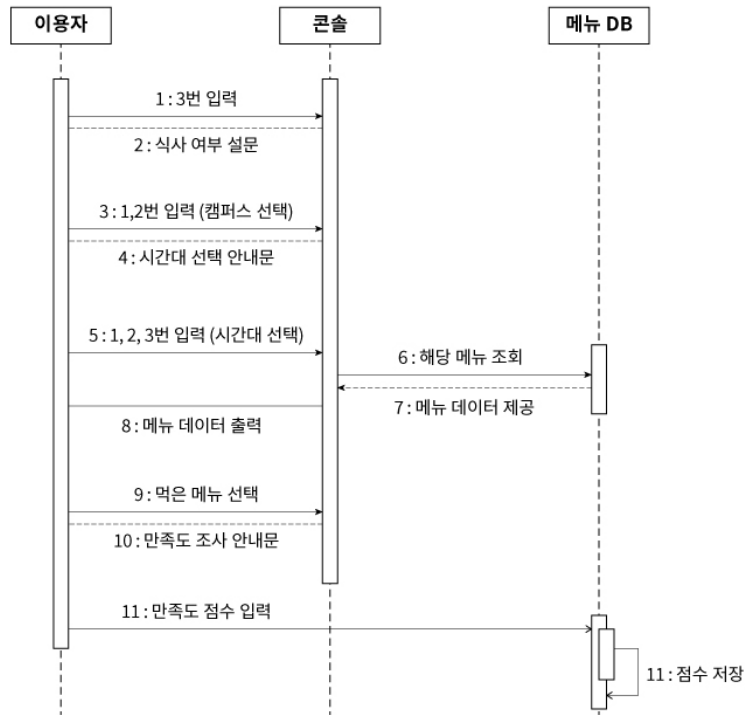


##### 2. 랭킹 확인

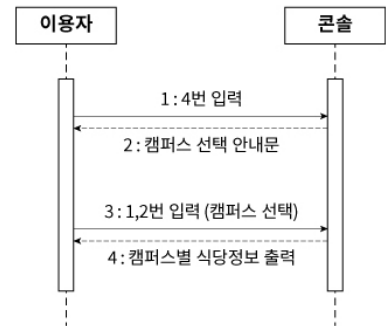




### 3. 만족도 조사

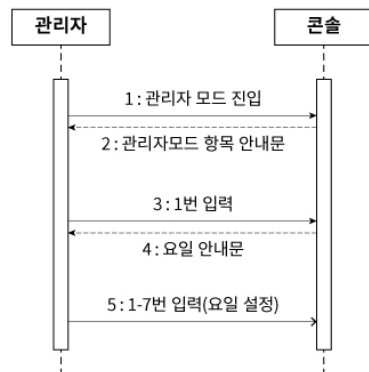


### 4. 식당 정보 확인

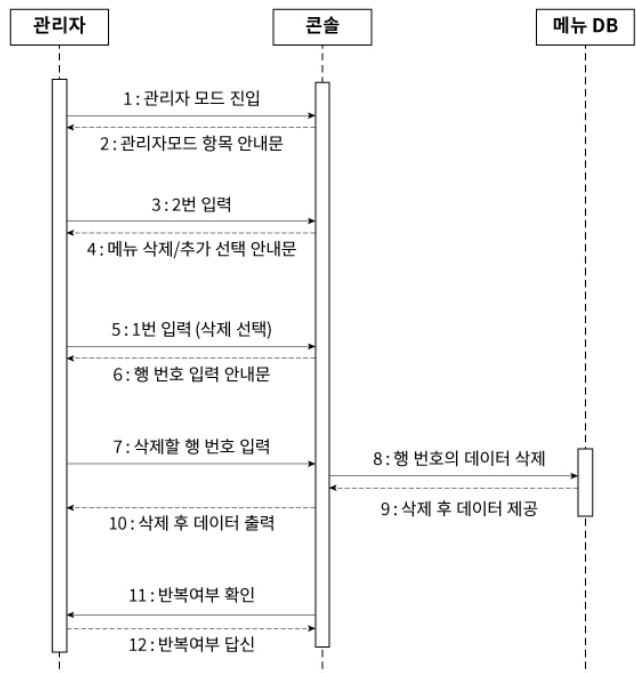


## -관리자-

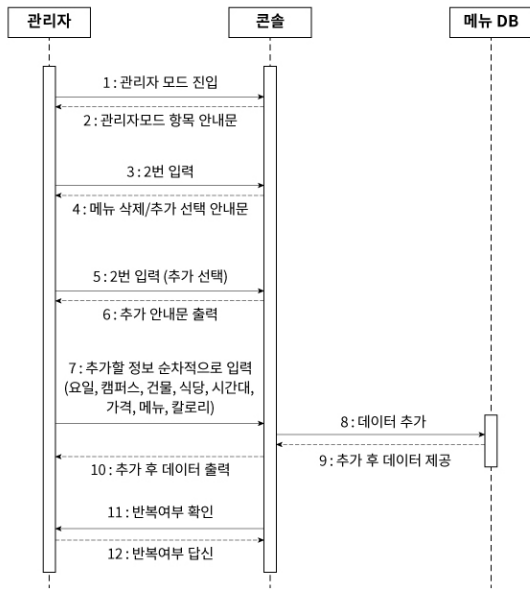
### 1. 요일 설정



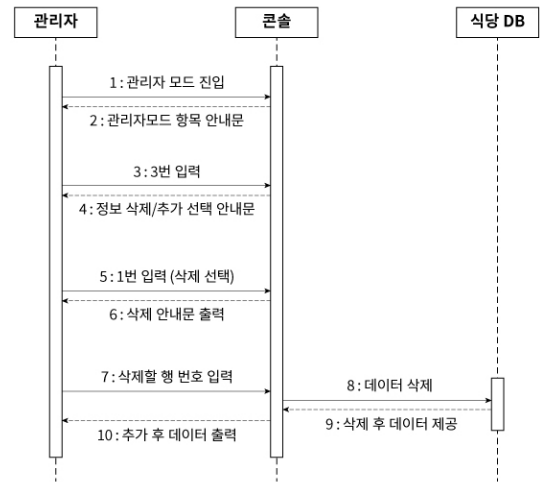
### 2. 메뉴 삭제



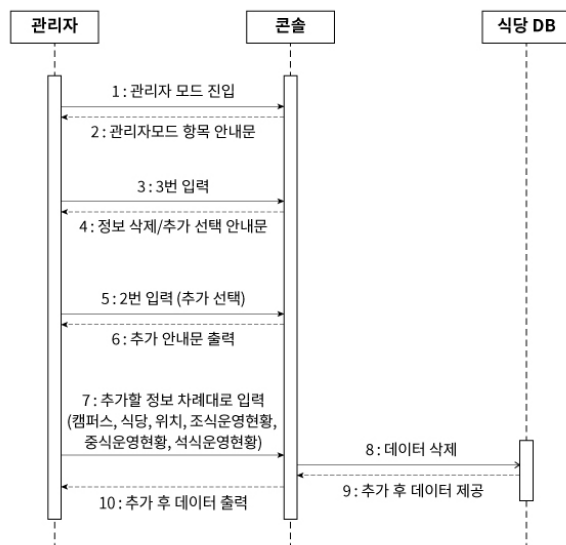
## 2-1. 메뉴 추가



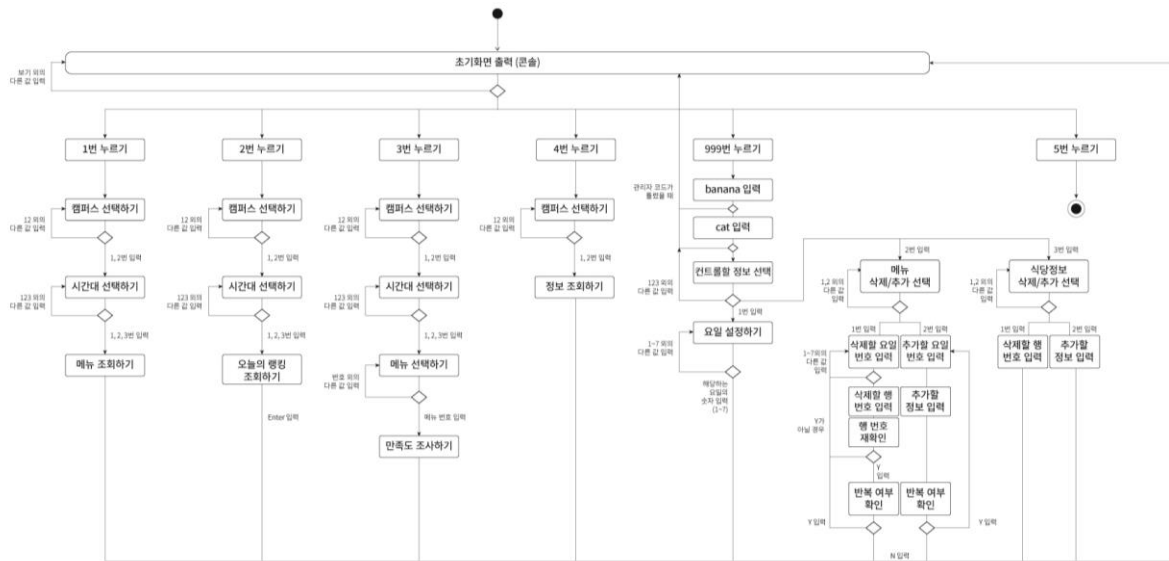
## 3. 식당정보 삭제



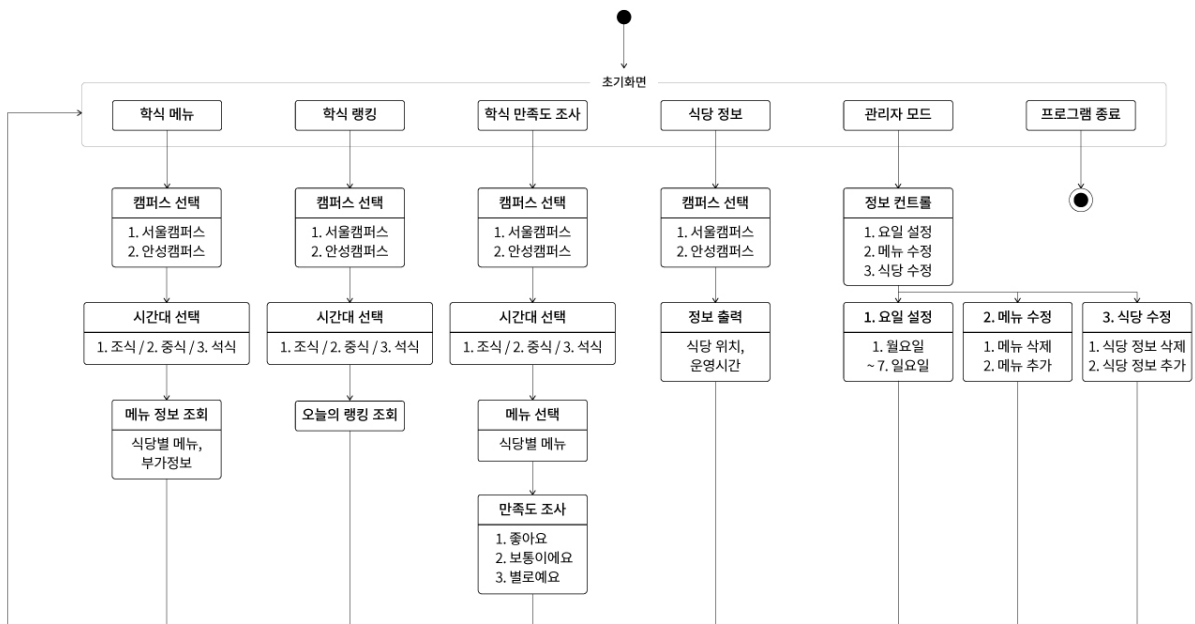
## 3-1. 식당정보 추가



### 3. 활동 다이어그램



## 4. 상태 다이어그램



## 4. 시스템 신뢰성

---

### 가용성

- 프로그램을 가동할 환경이 되면 언제든지 실행할 수 있다.

### 신뢰성

- 동일한 사용자가 만족도 조사에 중복적으로 참여할 가능성을 배제하지 못했다.
- 하지만 주된 기능인 정보 제공은 원활하게 이루어지며, 신뢰성이 있는 정보를 제공했다.

### 안전

- CSV 파일이 형식과 다르게 입력되었을 경우 정보 출력에 있어 장애가 생길 수 있지만 이런 일이 발생할 경우는 적다.

### 보안

- 일반 사용자가 1,2,3 등의 단순한 숫자를 이용해 프로그램에 진입할 수 있게 한 대신, 관리자 모드에는 특수한 3 중 코드(999-banana-cat)를 부여하여 일반 사용자가 쉽게 접근하지 못하도록 하였다.

### 복원성

- 데이터를 제공하거나 수정하는 부분에 있어서 단계마다 CSV 파일에 저장해서 추출하는 방식으로 진행했기 때문에, 본 프로그램의 구동에 오류가 생기더라도 데이터는 손상 위험이 적다.

### 유지보수성

- 관리자 모드를 통해 메뉴를 삭제 및 추가하는 등의 유지보수가 가능하게 했다.

### 오류 허용

- 사용자가 지정되지 않은 값을 입력하면 다시 입력할 수 있도록 안내 문구를 띄우거나 지정된 화면으로 돌아가게 했다.