# **SMALL SNU**

# Design and Planning

2019-04-23

### **Members**

김상철 (2017-13121)

김도현 (2017-10488)

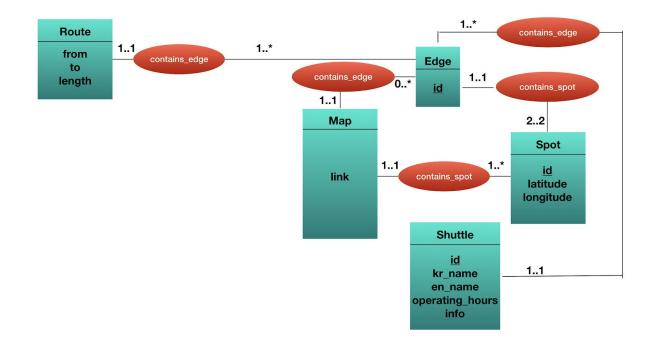
이동학 (2017-12751)

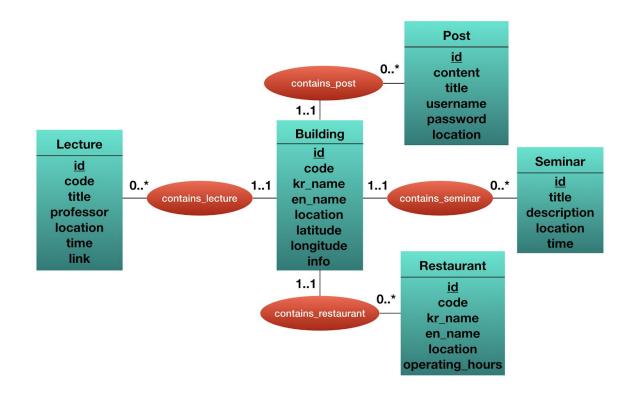
박종석 (2017-11522)

# **System Architecture**

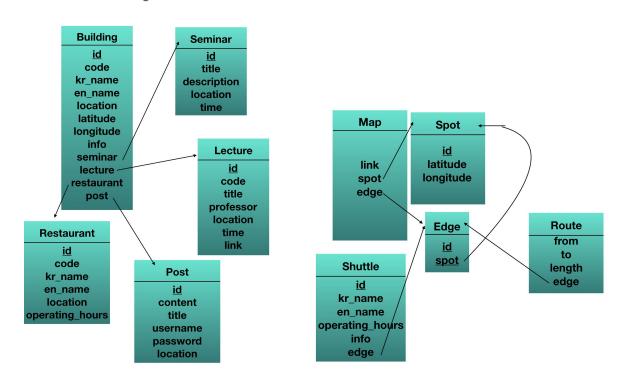
### Model

E-R diagram

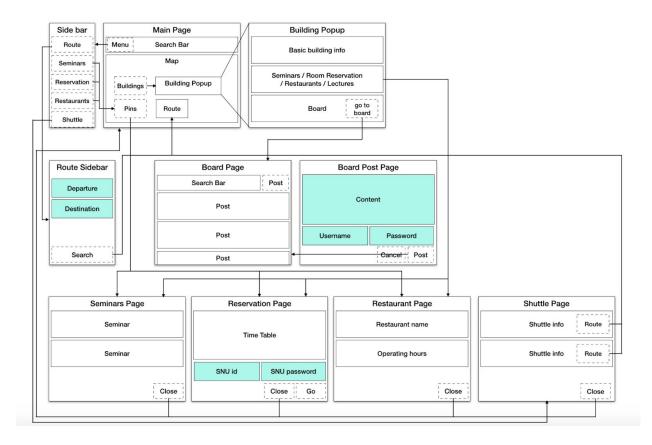




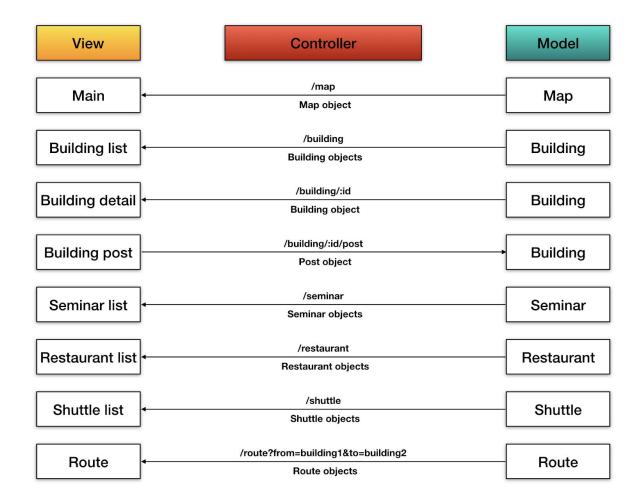
#### relation schema diagram



# View



### Controller



1. /map

frontend에 지도화면을 띄우기 위한 기본적인 정보들을 전달한다.

2. /building

building들의 리스트를 전달한다.

3. /building/:id

한 building의 세부정보를 전달한다.

4. /building/:id/post

해당 building에 글을 게시한다.

5. /seminar

열릴 예정인 모든 seminar를 전달한다.

6. /restaurant

모든 식당의 리스트를 전달한다.

7. /shuttle

모든 셔틀의 리스트를 전달한다.

8. /route?from=building1&to=building2

GET메소드에서 파라미터를 받아, 몇가지 경로들을 리턴해준다.

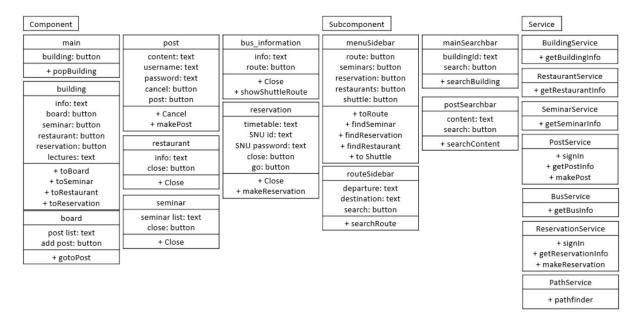
# **Design Details**

# Frontend Design

# **Frontend Components**

Here is frontend components, ssubcomponents, and services. The attributes and the methods of

each component are listed in each box.



### **Frontend Algorithms**

#### **Component:**

- 1. main
  - popBuilding(buildingId: number): show building popup
- 2. building
  - toBoard(buildingId: number): redirect to board page of current building
  - toSeminar(buildingId: number): redirect to seminar page of current building
  - toRestaurant(buildingId: number): redirect to restaurant page of current building
  - toReservation(buildingId: number): redirect to reservation page of current building

#### 3. board

- gotoPost(buildingId: number): redirect to post page of current building

#### 4. post

- Cancel(): redirect to board page
- makePost(username: string, password: string, content: string): call post api to make new post

#### 5. restaurants

Close(): go back to main page

#### 6. seminar

- Close(): go back to main page

#### 7. bus\_information

- Close(): go back to main page
- showShuttleRoute(busId: number): call bus api to get information of selected shuttle bus

#### 8. reservation

- Close(): go back to main page
- makeReservation(buildingId: number, sinceWhen: string, tilWhen: string): call reservation api to make new reservation

#### **Subcomponents**

#### 1. menuSidebar

- toRoute(): redirect to routeSidebar
- findSeminar(): find all buildings hold seminar and display on main page
- findReservation(): find all buildings available to make reservation and display on main page
- findRestaurant(): find all restaurant and display on main page
- toShuttle(): redirect to shuttle page

#### 2. routeSidebar

- searchRoute(departureId: number, destinationId: number): call pathFinder api to display shortest path on main page

#### 3. mainSearchbar

- searchBuilding(buildingld: number): find the building and show information

#### 4. postSearchbar

- searchContent(content: string): call search api to find post that contains search word

#### Services

#### 1. BuildingService

 getBuildingInfo(buildingId: number): call backend api to get basic information of current building

#### 2. RestaurantService

- getRestaurantInfo(buildingId: number): call backend api to get information of current restaurant

#### 3. SeminarService

- getSeminarInfo(buildingId: number): call backend api to get seminar information of current building

#### 4. PostService

- signIn(username: string, password: string): call backend signin api and return the result
- getPostInfo(buildingId: number): call backend api to get board and post information of current building
- makePost(buildingId: number, content: string): call backend api to make new reservation

#### 5. BusService

- getBusInfo(busId: number): call backend api to get information of selected shuttle bus

#### 6. ReservationService

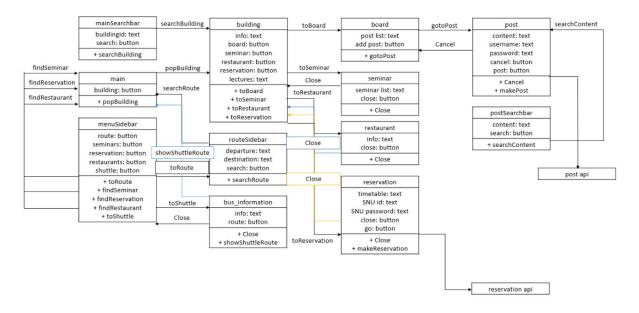
- signIn(username: string, password: string): call backend signin api and return the result
- getReservationInfo(buildingId: number): call backend api to get reservation information of selected building
- makeReservation(buildingId: number, sinceWhen: string, tilWhen: string): call backend api to make new reservation

#### 7. PathService

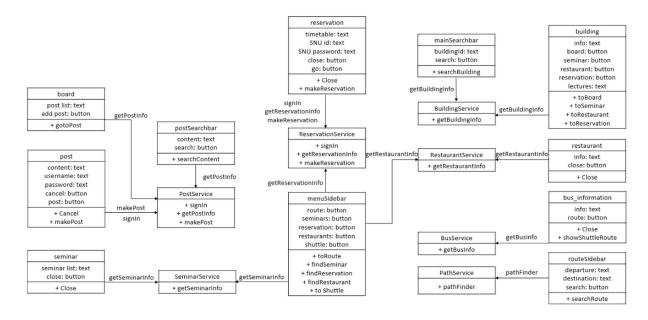
- pathFinder(departureId: number, destinationId: number): call backend api to find shortest path from departure to destination

### **Frontend Relations**

Here are the relations between components.



Here are the relations between services and components.



# **Backend Design**

In the backend design, we use models which have been discussed in MVC architecture section.

#### **RESTful API**

Detailed specifications of RESTful APIs are as following:

Model	API	GET	POST	PUT	DELETE
User	/signin	X	Log in	X	Х
Мар	/map	Get map information	Х	X	Х
Building	/building	Get building list	Х	X	Х
	/building/:id	Get information of specific building	Х	X	Х
	/building/:id/post	Get post list	Create new post	X	delete
Seminar	/seminar	get Seminar information	Х	X	Х
Restaurant	/restaurant	Get restaurant information	Х	X	Х
Shuttle	/shuttle	Get shuttle information	Х	Х	Х
Route	/route ?from=building1 &to=building2	Get shortest route information	Х	Х	Х

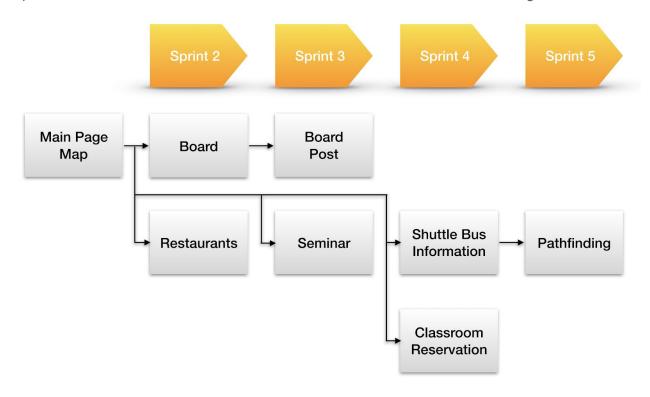
### Algorithm for Route model

구글맵, 카카오맵, 네이버 지도 등 현존하는 지도 프로그램들은 큰 길들의 정보만 저장되어 있고 작은 샛길들의 정보가 저장되어 있지 않아 좁은 범위에서 경로를 탐색할 때 최적화된 경로를 찾아내지 못한다. 그래서 smallsnu는 서울대 내부에 한정해서지만, 작은 길들의 정보를 추가하여 기존 지도 프로그램들 보다 더 최적화된 경로를 탐색하는 알고리즘을 구현할 것이다.

# **Implementation Plan**

구현 계획은 다음과 같다. 기본적인 구현 계획은 Requirements & Specification 문서의 User Stories를 바탕으로 세웠다. User Stories 에서의 각 Page를 하나의 작업 기준으로 세워서 각각을 블록으로 표시하였다.

이 프로젝트는 다루는 기능이 다양하기에, 각 페이지에 대한 dependency는 그렇게 강하지 않은 편이다. 하지만 여전히 dependency를 가지고 있는 기능들이 존재하기에, 그 순서를 1순위로 고려하여 구현 계획을 세웠다. 가장 중요한 지도를 표시하는 'Main Page'부터 sprint2 시작에 구현을 하고, 그 이후 'Board', 'Restaurant'까지 sprint2 내에 완료한다. 그 이후 게시글을 보여주는 'Board'에 dependency를 가지는 'Board Post', 그리고 'Seminar'를 sprint3 내에 완료한다. sprint4에선 'Shuttle Bus Information'과 'Classroom Reservation'을 구현하고, sprint5에서는 이 프로젝트에서 구현이 가장 어려울 것으로 예상되는 'Pathfinding'을 구현한다.



구현 계획에서 Page의 구현 순서 배치는 1차적으로는 dependency, 2차적으로는 구현의 어려움을 고려하였다. 각 Page내의 세부적인 기능과 그 기능 구현에 대한 분담은 다음과 같이 진행할 계획이다.

Page	:	:	:	 Person	Challenge
		culties	(mins)		
		(1~5)			

Main	Show map	1	60	2	박종석	
Main	Click building and show popup	4	240	2	박종석	Setting coordinate of building buttons
Board	Show board lists and posts	1	90	2		
Restaurants	Show restaurant	2	120	2		Investigating every restaurant information of our school
Board	Show board post page and post board	2	120	3		
Seminar	Show seminars	3	240	3		Crawling seminar information from various sources
Shuttle Bus Information	Show shuttle bus information	3	180	4		Investigating bus information and marking its route
Classroom Reservation	Show available classroom for reservation and make reservation	4	240	4		Crawling information from mysnu, Dealing with mysnu id/pw
Pathfinding	Show fastest route to reach one building from another	5	420	5		Dealing with pathfinding algorithm
Main	Show corresponding pins when pick from the sidebar	2	120	2~5		

# **Testing Plan**

# **Unit Testing**

모든 코드의 구성요소에 대해 unit test를 진행한다. 매 sprint에서 다음 framework를 사용하여 테스팅을 진행한다. Coverage는 90% 이상을 목표로 한다.

- React/Redux: Jest

- Django: Python unit test

# **Functional Testing**

프로젝트를 진행할 때 만드는 API에 대해서도 모두 테스팅을 한다. 우리가 계획한 RESTful API에 대해서 sprint3와 sprint4에서 나누어서 testing을 진행한다. 사용하는 framework는 다음과 같다.

- React/Redux: Jest

- Django: Python unit test

# **Acceptance & Integration Testing**

Cucumber를 통해 user stories를 바로 acceptance testing에 사용할 수 있도록 한다. 모든 기능을 테스트하기 위해 이는 sprint5에서 테스트한다. 그리고, integration test에는 수업시간에 다룬 travis CI를 사용한다.

- Acceptance Testing: Cucumber

- Integration Testing: Travis CI