

# LookVie Design Specification

Team LookVie

김명선, 송민준, 이의섭, 유진, 조범준, 한승남

## 1. Architecture

A detailed definition of the system's software components.

- the major **modules** and their functionality

Searching module - Search movie (Maybe show the Box Office ranking).

Naver Map API - Search for cinemas nearby the user and calculate the distance between the user and the cinema.

Cinema Recommendation module - Compares the arrival time to each cinemas nearby and the screening time of each cinemas nearby to get the best solution(fastest way to watch the movie).

CGV Web Scraping data - Data of CGV cinemas and its screening time schedule. And scraps each theaters seat map.

- the **interfaces** between modules

Searching module - Naver Map API

-When getting the search value get the location of the user by the gps of the phone

-Search for the cinemas nearby the user

Naver Map API - Cinema Recommendation module

-Gives the arrival time of each cinemas nearby to the CR module

Naver Map API - CGV data - Cinema Recommendation module

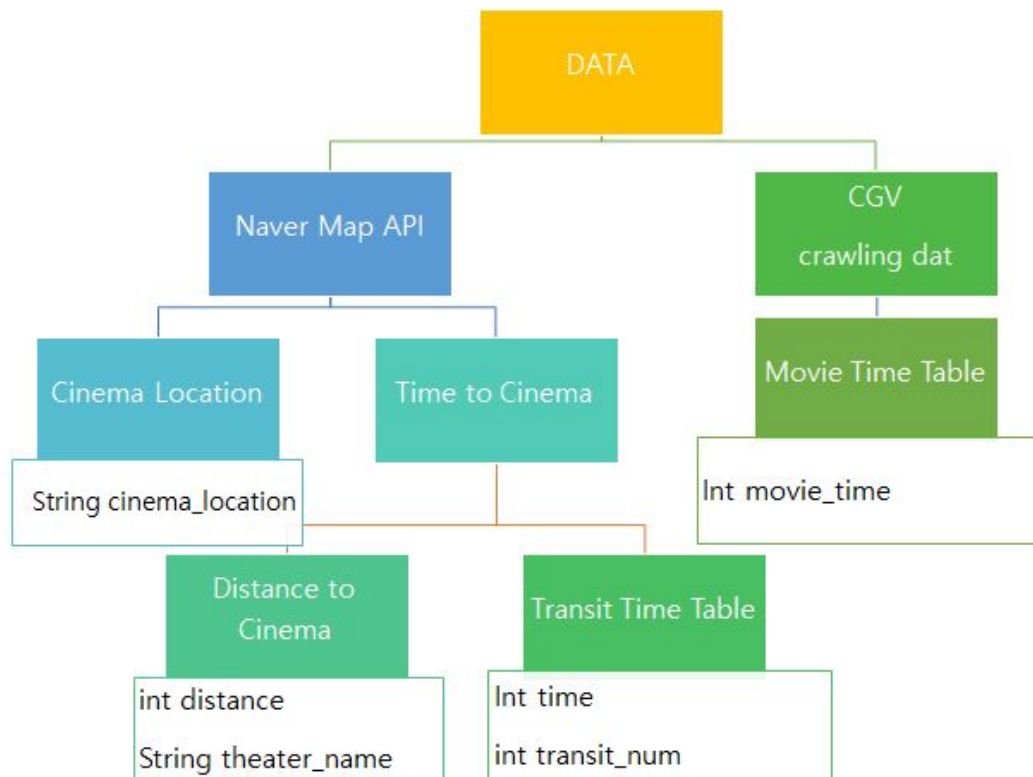
-Gives the information of nearby CGV cinemas to the CGV Web Scraping data

-Find the screening time of the nearby cinemas and send it to CR module

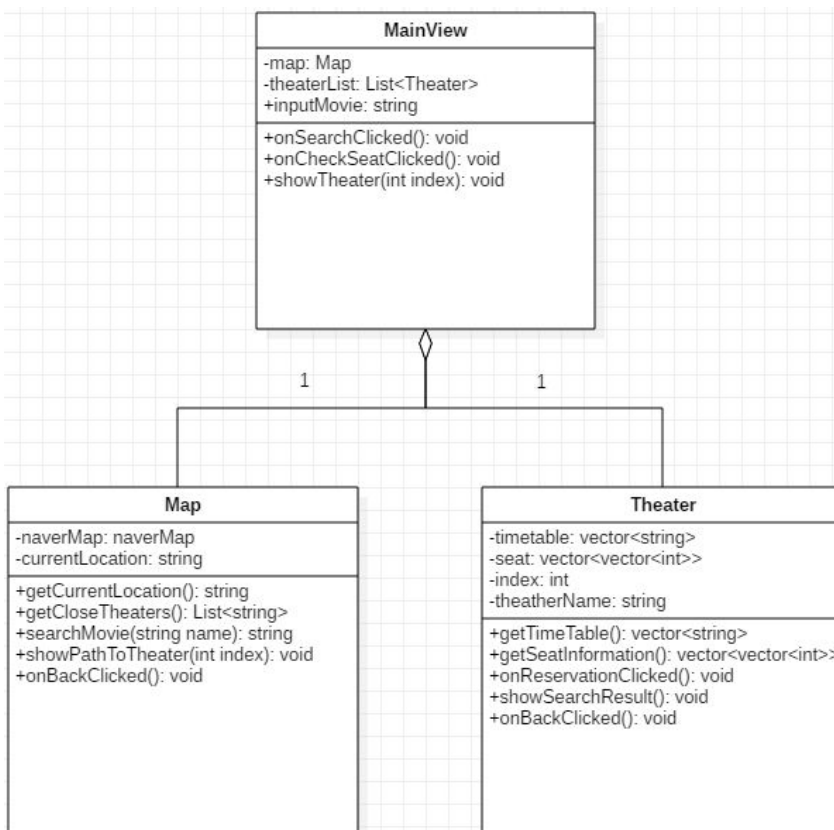
Cinema Recommendation module - CGV data

-After getting the best cinema to watch, let the CGV data find that cinema and give the seat map to the user

- **data** description (database schema if possible)

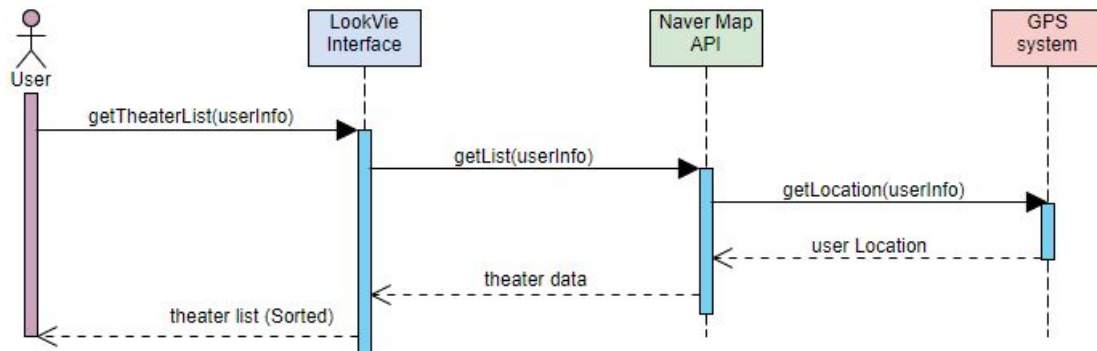


- design **alternatives**

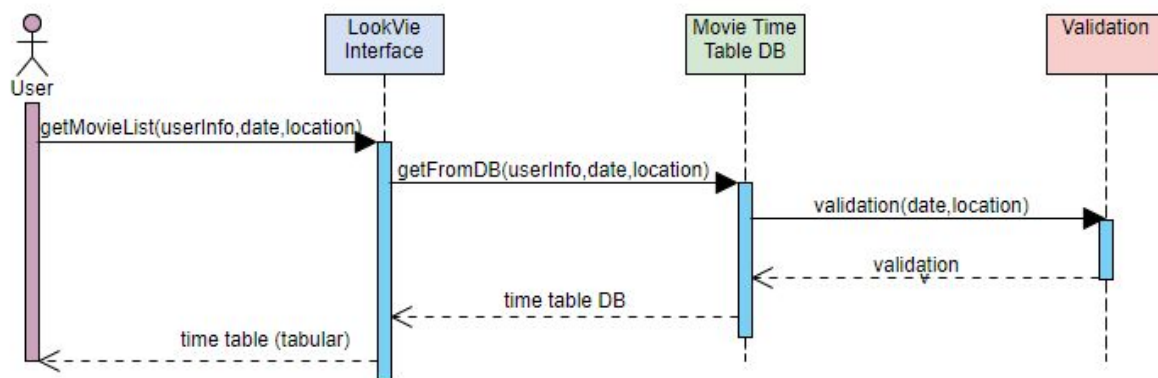


## - assumptions

- a. search movie list via Naver Map API



- b. show close theater list from DB



## 2. Process

### - Risk assessment

#### - Database update issue

We plan to save data to a device(local). Since this application should have the most recent database, we will update this application every week.

#### - Crash with certain Android versions

Since we are mainly going to design with the Pie (API 28) version, there might be clash with older versions of Android API.

### - Project schedule

	WEEK										
	1	2	3	4	5	6	7	8	9	10	11
Design proposals				M I D T E R M	Design Specifi cation						F I N A L  D E M O
Design DB Schema											
Create crawling model / Scrape data											
Implement basic application features											
UI Design											
Implement map algorithm											
Implement additional function											
Test and prepare final demo								A L P H A			

- Team structure

- App Development / Map API & Algorithm  
Myeongseon Kim, Beomjun Cho
- Crawling Data  
Minjoon Song, Seungnam Han
- DB Management  
Euisup Lee, Minjoon Song
- UI Design  
Yujin, Euisup Lee
- Document

Seungnam Han, All

- Test plan

-Test by using Android Studio test tools such as

- AndroidJUnitRunner : Android test runner based on JUnit. Test the methods, classes, components of the code
- Espresso : UI Testing Framework, good for testing UI in the app

- Documentation plan

- Will be uploaded to our github : <https://github.com/lookvie>
- Documents will contain sources of reuse softwares
- Bugs occurred during development phase

- Coding style guidelines

-We will write source code with strict rules for contributing Java code to the Android Open Source Project

-<https://source.android.com/setup/contribute/code-style#javatests-style-rules>