



# Meeting Room Pro

Website: <https://navsingh.org.uk/mrpro>

NAVJOT SINGH VIRK

24/August/2017

Available On:

Github/Virksaabnavjot/MRPRO



# Introduction

## MeetingRoom Pro (Find, Review, Book)

What is the project/product about ?



### ► Project Context

A room assistant application that allows the user to

1. Book Rooms
2. Navigate
3. See the Building and the Meeting room on the map
4. Look at room gallery and upload images and review rooms .

And more discussed in further slides.

# Background

What were the reasons to choose this project?

- ▶ Its very hard to find rooms in large corporate buildings.
- ▶ Is very hard to find rooms/location in a building if you are new to the environment.
- ▶ The idea originated during my internship, when it was hard to find and book meeting rooms.
- ▶ People needed easy solution to book and find rooms.
- ▶ People needed an organized system, where they can choose b/w rooms based on their

# Project Goals



- ▶ Build easy and user friendly GUI.
- ▶ Build easy to use features and functionality.
- ▶ Minimal learning curve to start using the application to its fullest.
- ▶ Expand internationally.
- ▶ Happy users and Customer.

# Target Users

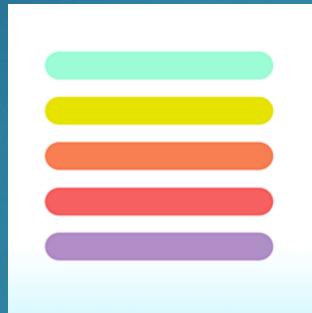
Who can use this product ?

- ▶ Corporate companies and their employees.
- ▶ Schools, Colleges and Universities
- ▶ Teachers and Students
- ▶ Facilities department
- ▶ IT department

# List of Features Users can benefit from ?



SEARCH



LIST AND FIND



MAP AND ROOM  
DETAILS



REVIEWS



GALLERY



PHOTO UPLOAD



BOOK ROOM



MY MEETINGS

# Competitors

## ► Room Finder

Allows to find and book rooms inside Outlook and Office 365.

**Drawbacks:** Limited to outlook / microsoft products, plugin and not an independent application.

## ► Skedda

Online booking and scheduling for meeting rooms.

**Drawbacks:** Only available to publically available places and only web based.

# System (iOS Application)

What are the technologies used ?

Front-end : iOS Application is built with Swift 3 using Xcode IDE.



XCODE



SWIFT 3



JSON



MAPKIT



COCOAPODS



CORE LOCATION

# System (WEB Service)

What are the technologies used ?

Back-end : Web service using PHP, CodeIgniter, webhost using Apache server and MySQL database.



PHP



CodeIgniter



JSON



MySQL



000webhost

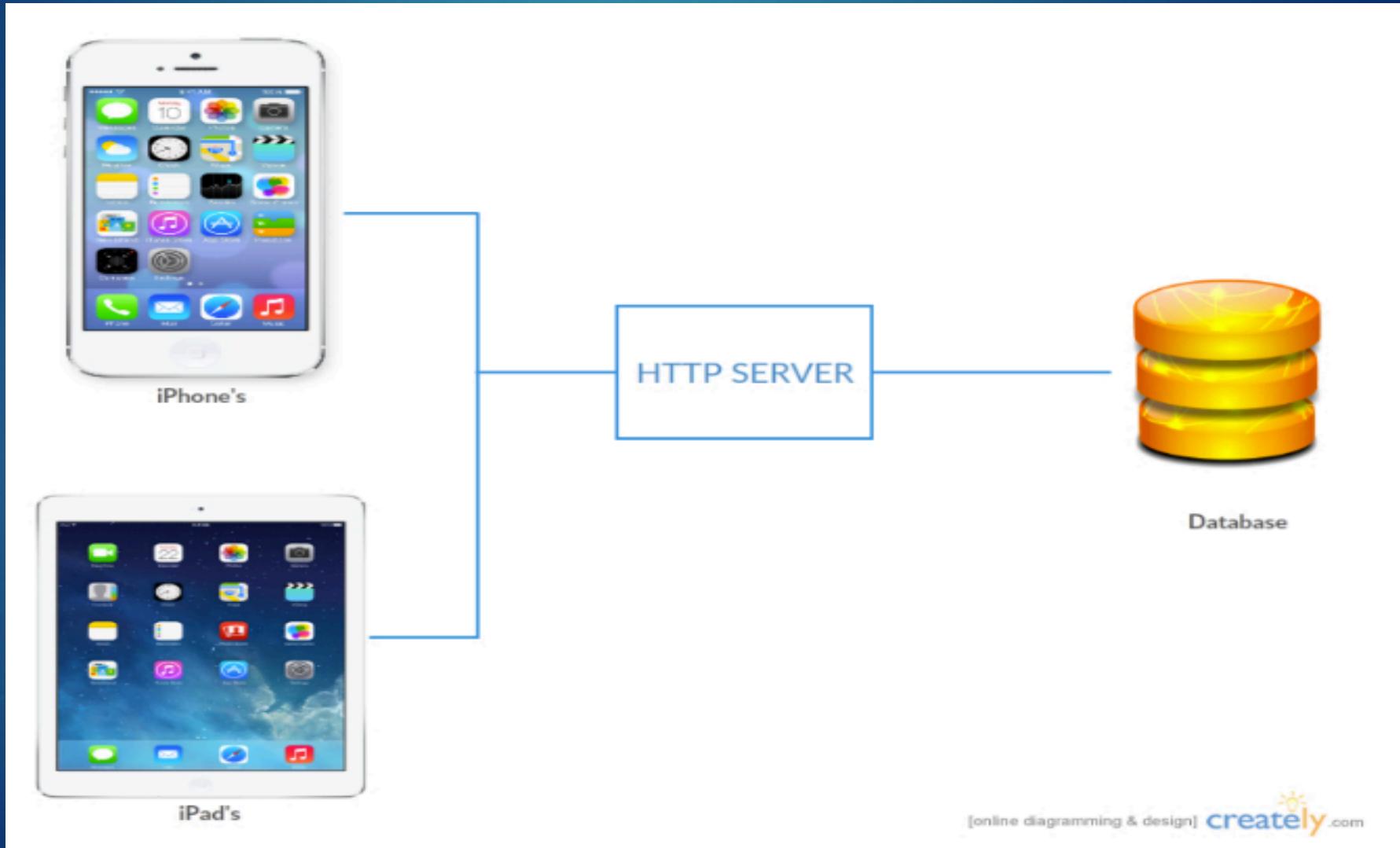
000 Web host



phpMyAdmin

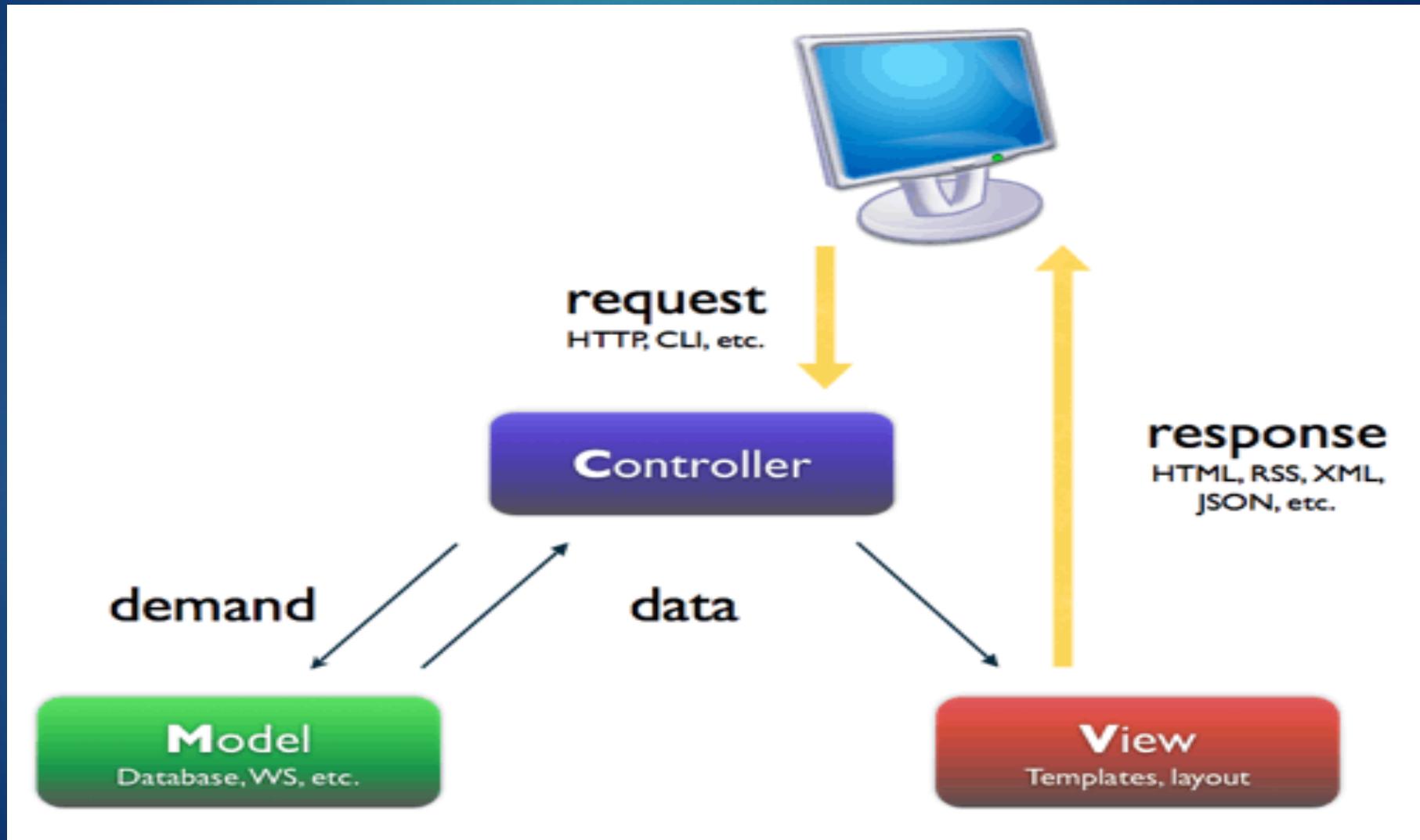
# System

# System Architecture



# System

MVC Architecture (CodeIgniter)



# System

Main algorithms

- Room Booking
- Parsing JSON
- Parsing Geo-spatial data
- Gallery and Image Upload
- Review System
- Room Search

# Design

Overview of the main functional requirements/features.

- Easy to Use GUI
- List and Find (Search)
- Map / Navigation
- My Meetings
- Booking System
- Review
- Gallery and Photo Upload

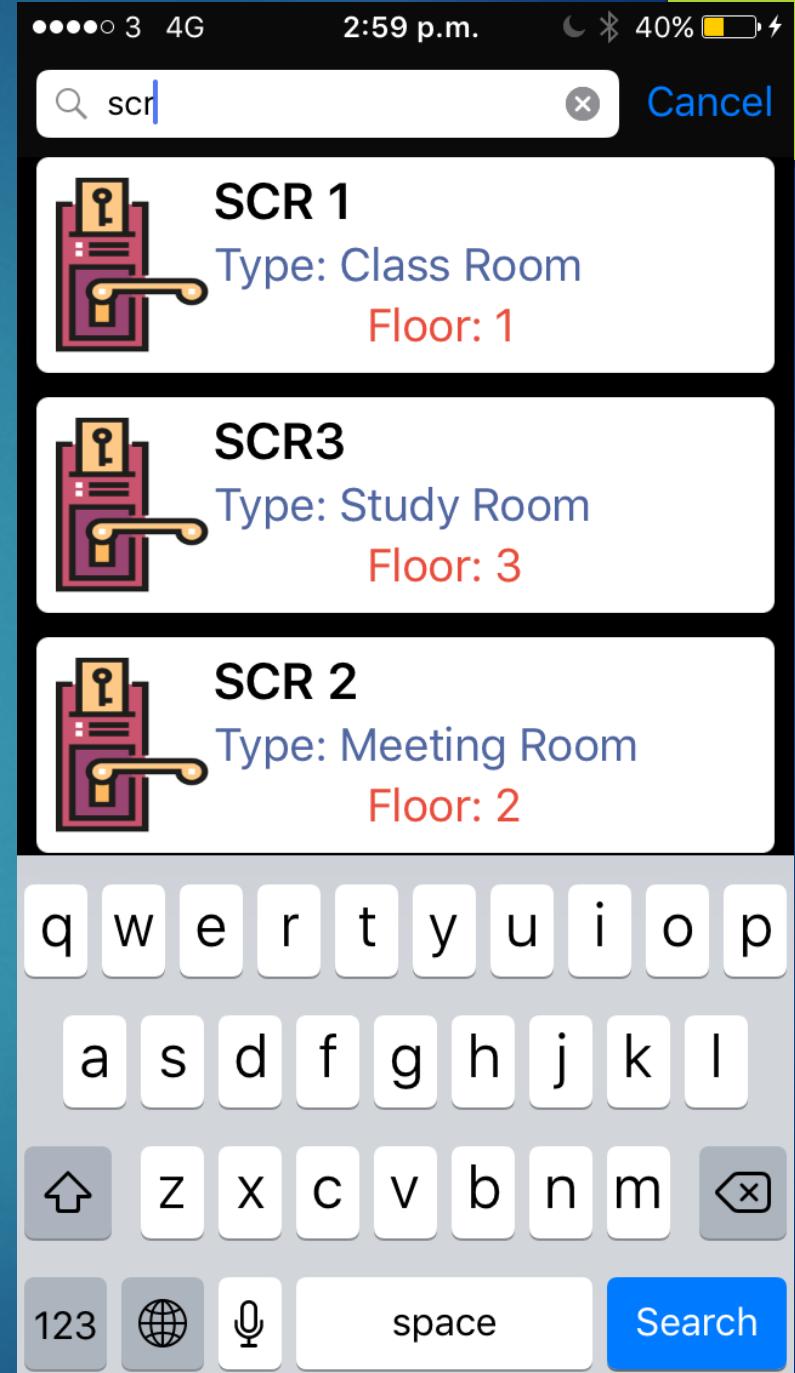
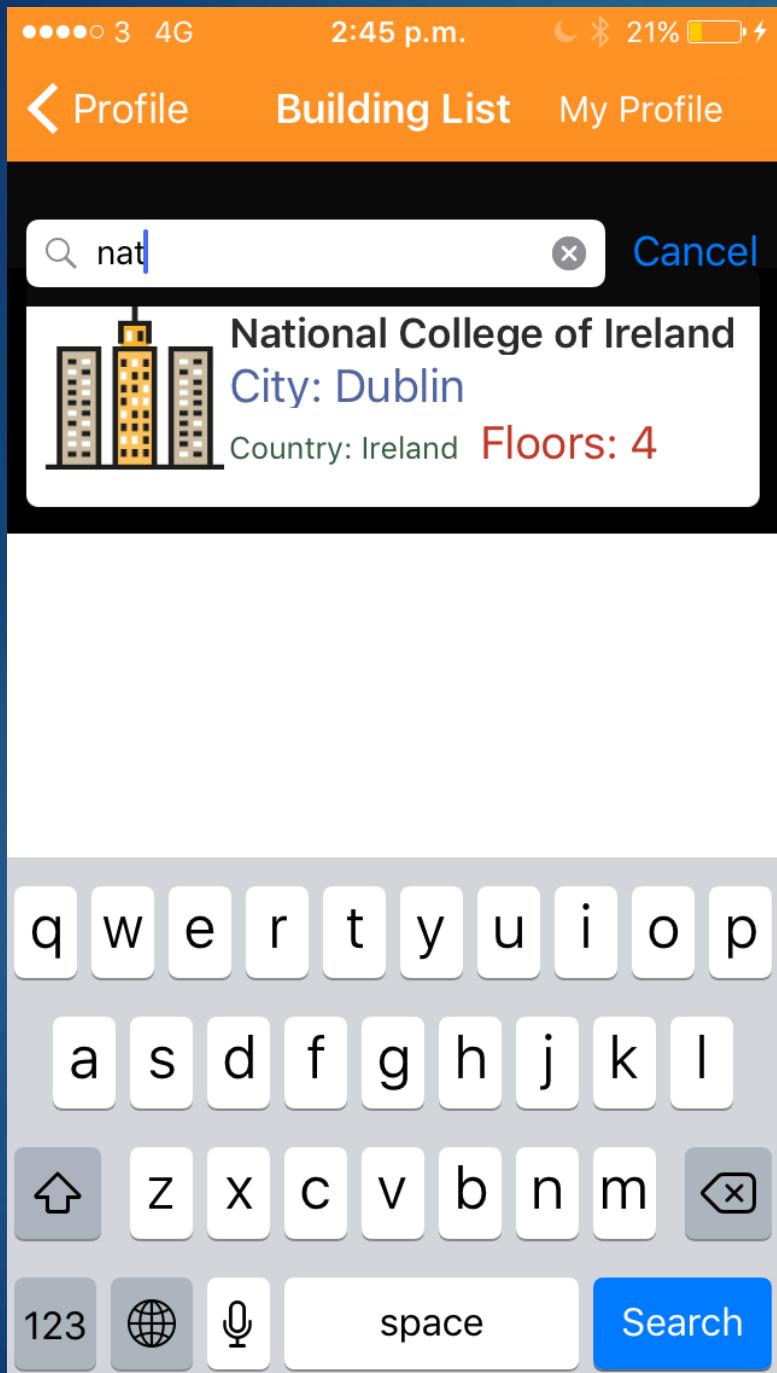


# App Features Summary and Demonstration

## FEATURE 1: SEARCH



Search Buildings  
And  
Meeting Rooms





# App Features Summary and Demonstration

## FEATURE 2: LIST AND FIND



View List of available Buildings  
and rooms available in them  
with relevant hints and  
information

●●●○ 3 4G

2:44 p.m.

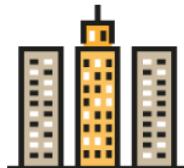
🌙 ⚡ 20% 🔋 ⚡

< Profile

Building List

My Profile

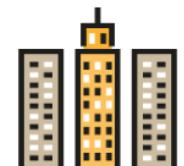
🔍 Search by Building Name



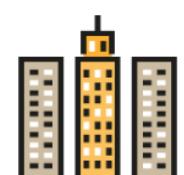
National College of Ireland  
City: Dublin  
Country: Ireland **Floors: 4**



NCI  
City: Dublin  
Country: Ireland **Floors: 3**



St. James  
City: Dublin  
Country: Ireland **Floors: 4**



Jurrys INN  
City: Dublin  
Country: Ireland **Floors: 4**

●●●○ 3 4G

2:59 p.m.

🌙 ⚡ 40% 🔋 ⚡

<

Meeting Room List My Meetings

🔍 Search Meeting Rooms



**Liffy Suite**  
Type: Meeting Room  
**Floor: 4**



**SCR 1**  
Type: Class Room  
**Floor: 1**



**Oriel Suite**  
Type: Conference Room  
**Floor: 3**



**SCR3**  
Type: Study Room  
**Floor: 3**



**College Atrium**  
Type: Meeting Room  
**Floor: 3**

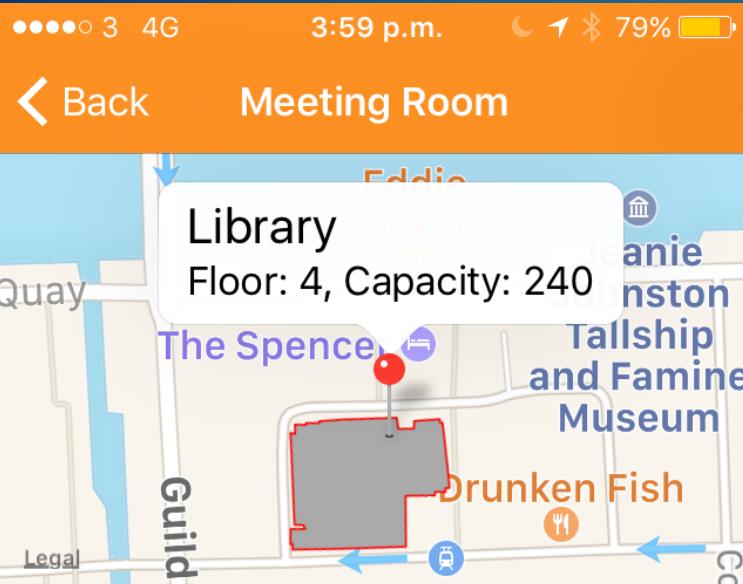


# App Features Summary and Demonstration

## FEATURE 3: MAP AND ROOM DETAILS

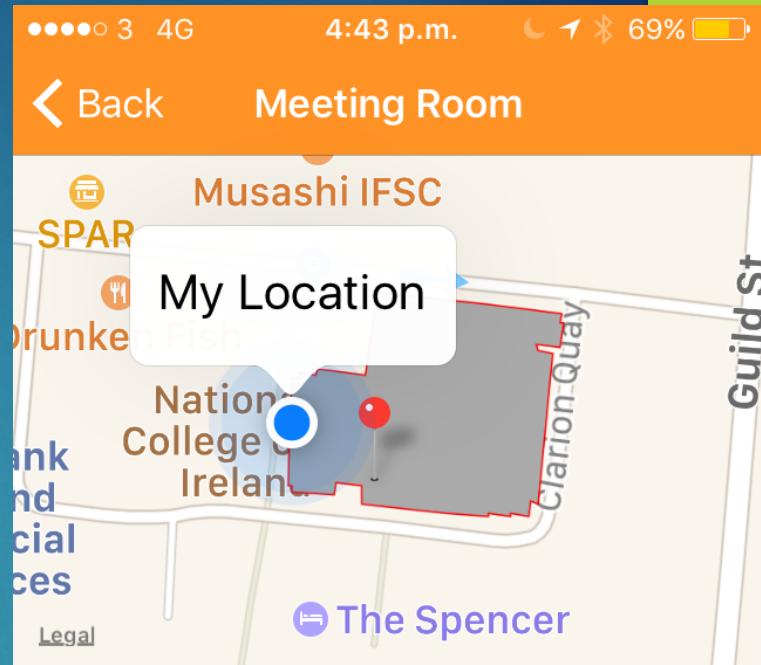


Allows the user to see the building and room on the map, user can opt between 2d or 3d view of the map to see the actual building and places around it and all the related information to the room.



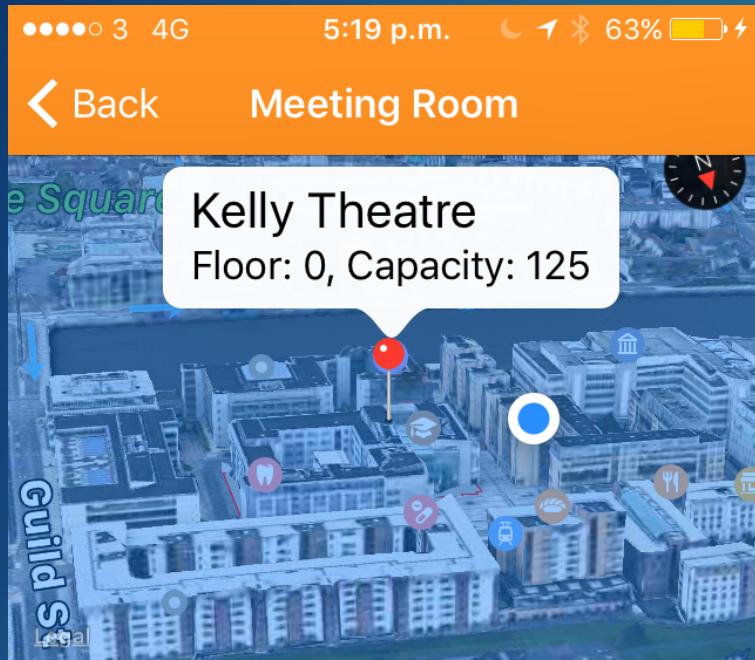
#### Meeting Room Information

Name: Library  
Parent Building: National College of Ireland  
Capacity: 240  
Email: library@ncirl.ie  
Coordinates: 53.348556 , -6.243047  
e: +353892110549  
Type: Study Room  
Floor Number: 4  
Directions : Walk straight from the entrance of your left clode to lifts  
[Book this Room](#)  
[Photo Gallery for this Room](#)



#### Meeting Room Information

Name: Library  
Parent Building: National College of Ireland  
Capacity: 240  
Email: library@ncirl.ie  
Coordinates: 53.348556 , -6.243047  
e: +353892110549  
Type: Study Room  
Floor Number: 4  
Directions : Walk straight from the entrance of your left clode to lifts  
[Book this Room](#)  
[Photo Gallery for this Room](#)



#### Meeting Room Information

Name: Kelly Theatre

Parent Building: National College of Ireland

Capacity: 125

Email: kelly@ncirl.ie

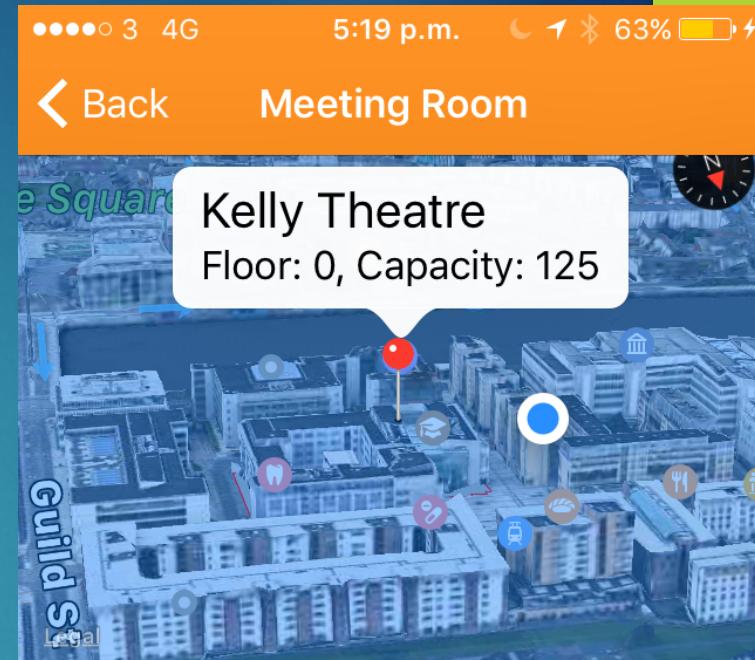
Coordinates: 53.348633 , -6.243084

Phone: +353892110549

Type: Theatre Room

Floor Number: 0

Directions : On entrance walk straight on you



#### Meeting Room Information

Name: Kelly Theatre

Parent Building: National College of Ireland

Capacity: 125

Email: kelly@ncirl.ie

Coordinates: 53.348633 , -6.243084

Phone: +353892110549

Type: Theatre Room

Floor Number: 0

Directions : On entrance walk straight on you



# App Features Summary and Demonstration

## FEATURE 4: GALLERY



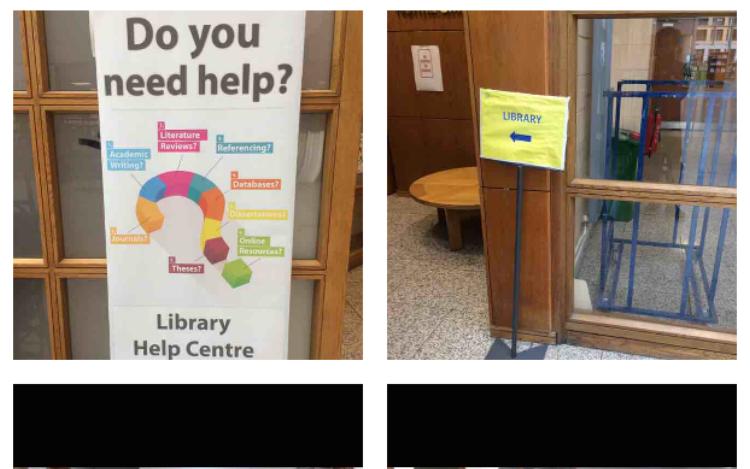
Allows the user to see pictures of the room, which helps him to make decisions based on his/her requirements.

••••• 3 4G

5:07 p.m.

🌙 1 ✉ 56% 🔋

## Meeting Room Photo gallery

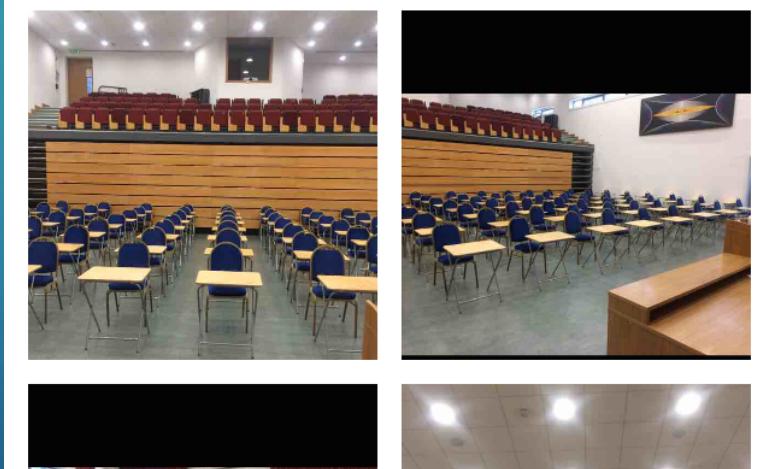


••••• 3 4G

5:08 p.m.

🌙 1 ✉ 55% 🔋

## Meeting Room Photo gallery



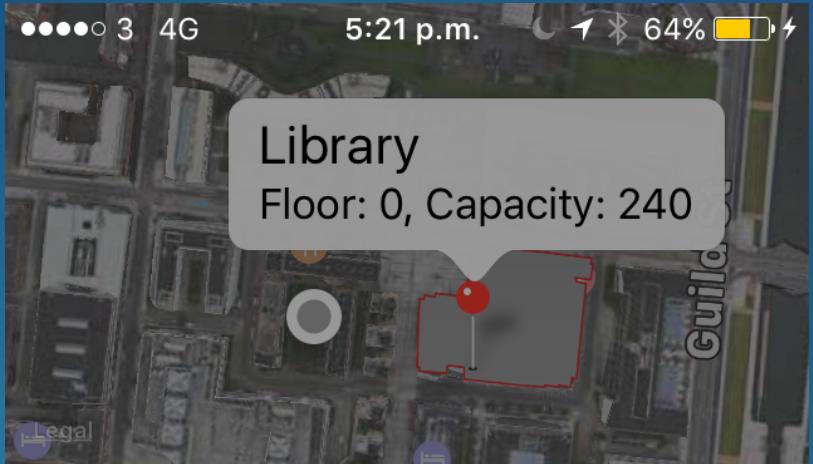


# App Features Summary and Demonstration

## FEATURE 5: PHOTO UPLOAD



Allows the user to add/upload Pictures of the rooms and building to the cloud from the camera or phone's gallery.



### Meeting Room Information

Name: Library

Parent Building: National College of Ireland

Capacity: 240

Select Option

Gallery

Camera

Photo Gallery for this Room

Cancel

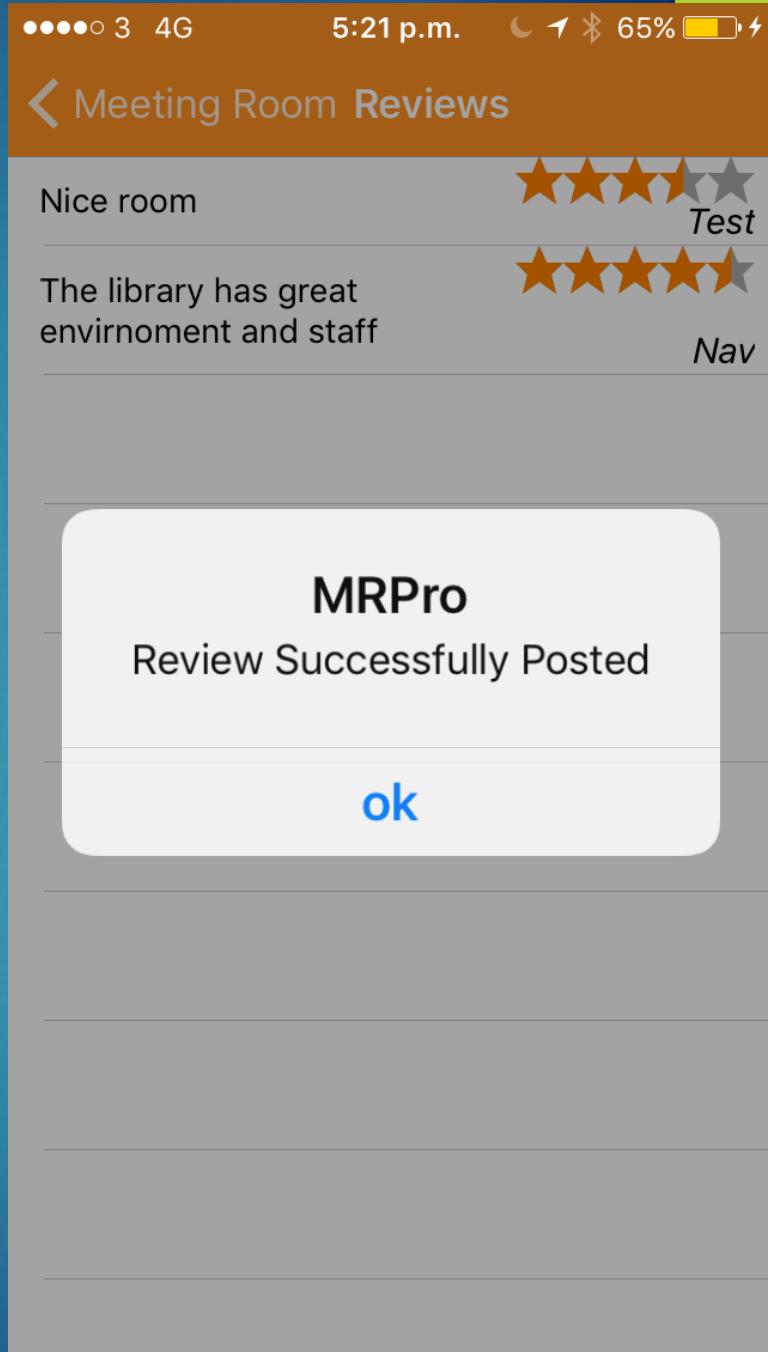
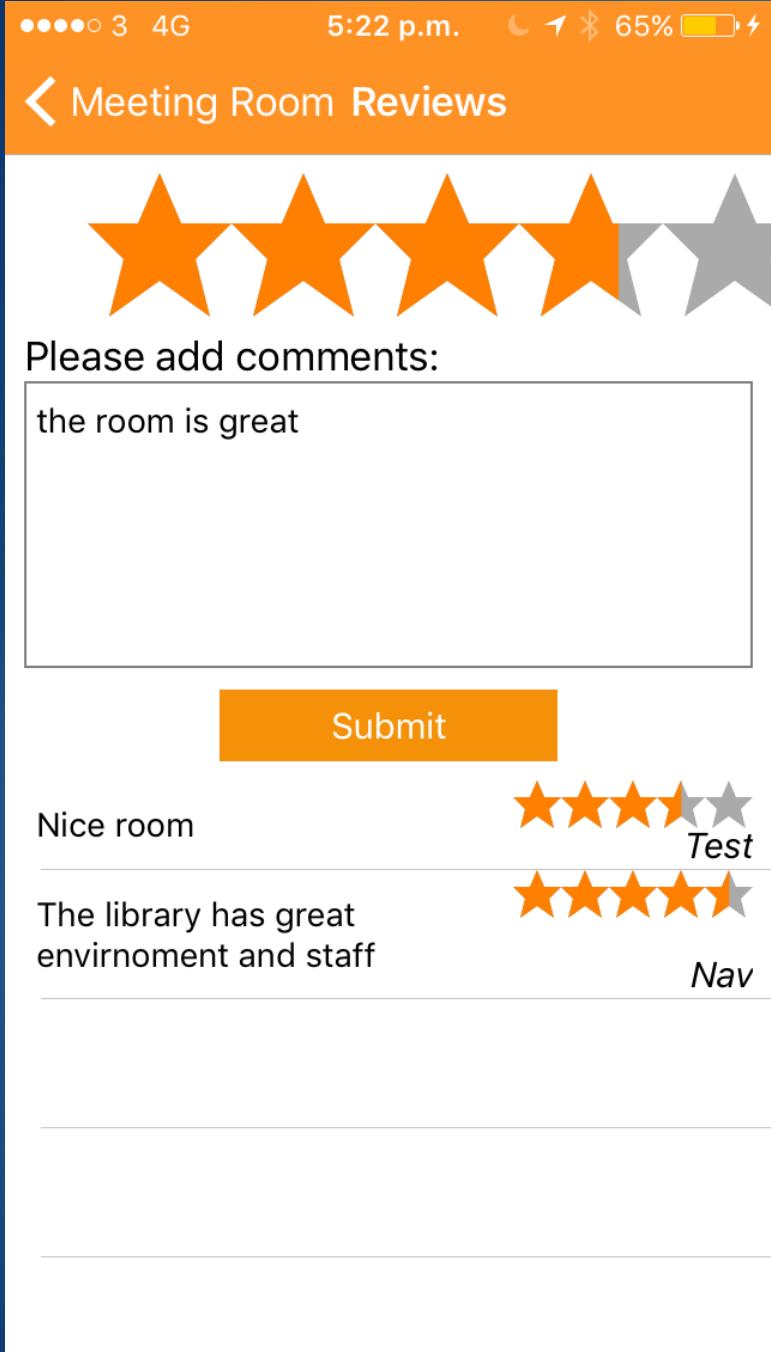


# App Features Summary and Demonstration

## FEATURE 6: USER REVIEWS



Allows the user to review and provide the rating for a room based on their experience with the room, this feature helps other users and admin to get more insights about the room.





# App Features Summary and Demonstration

## FEATURE 7: ROOM BOOKING



Allows the user to book meeting room by choosing a room, title, date and time  
And the people he wants to invite to the meeting.

●●●● 3 4G 5:26 p.m. ⚡ 69%

## Meeting Room Room Booking

Floor

Title: Team meeting

Start Time Jul 31 17, 5:25 p.m.

End Time Jul 31 17, 6:30 p.m.

JD John Doe

NV Navjot Virk

E Eamon

C Cristina

NS Navjot Singh

Book This Room

●●●● 3 4G 5:26 p.m. ⚡ 69%

## Meeting Room Room Booking

Title: Team meeting

Start Time Jul 31 17, 5:25 p.m.

End Time Jul 31 17, 6:30 p.m.

JD John Doe

NV Navjot Virk

E Eamon

C Cristina

NS Navjot Singh

Book This Room



# App Features Summary and Demonstration



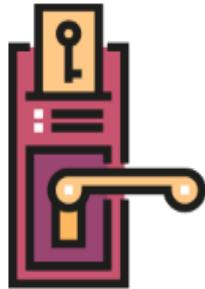
## FEATURE 8: MY MEETINGS



Allows the user to view all  
the booked meetings/rooms.

 Profile

## My Meetings



**SCR 1** → Room name we booked

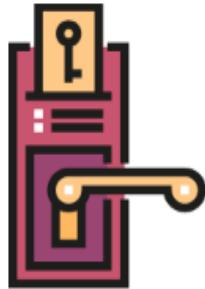
Floor: 1

Building: National College of Ireland

End Time

Jul 31 17, 3:26 p.m.

Jul 31 17, 3:37 p.m.



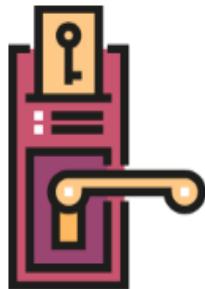
**SCR 1** Meeting start time

Floor: 1

Building: National College of Ireland

Jul 31 17, 3:26 p.m.

Jul 31 17, 3:37 p.m.



**SCR 1**

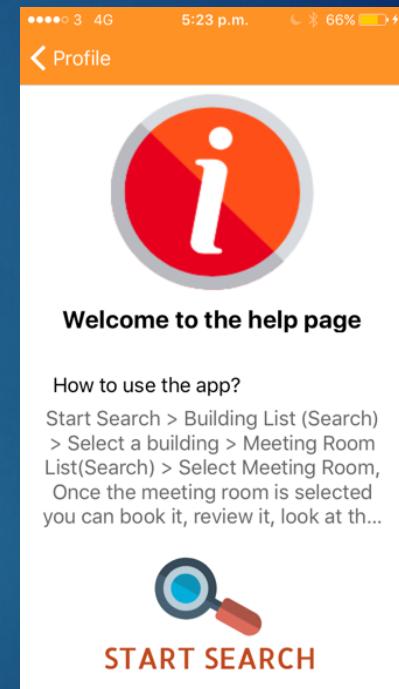
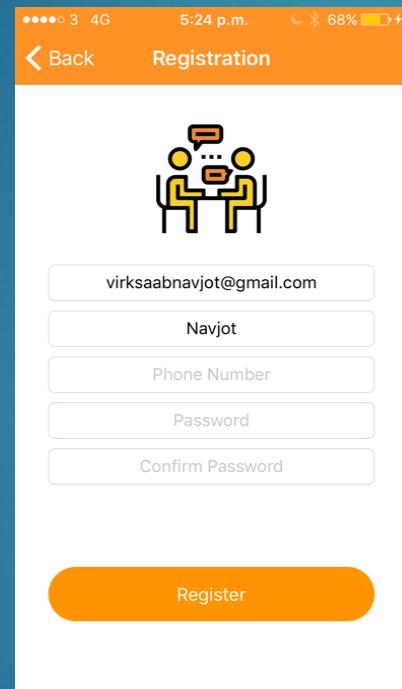
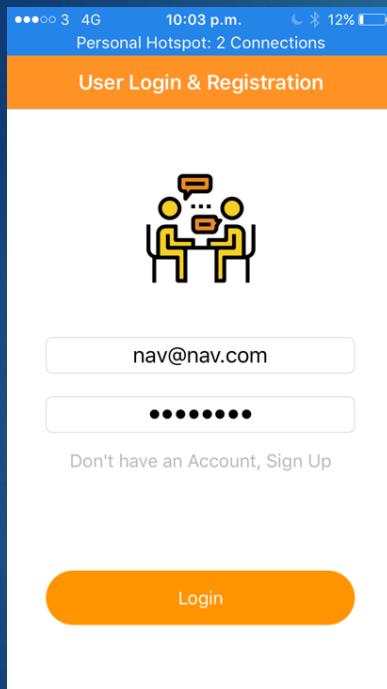
Floor: 1

Building: National College of Ireland

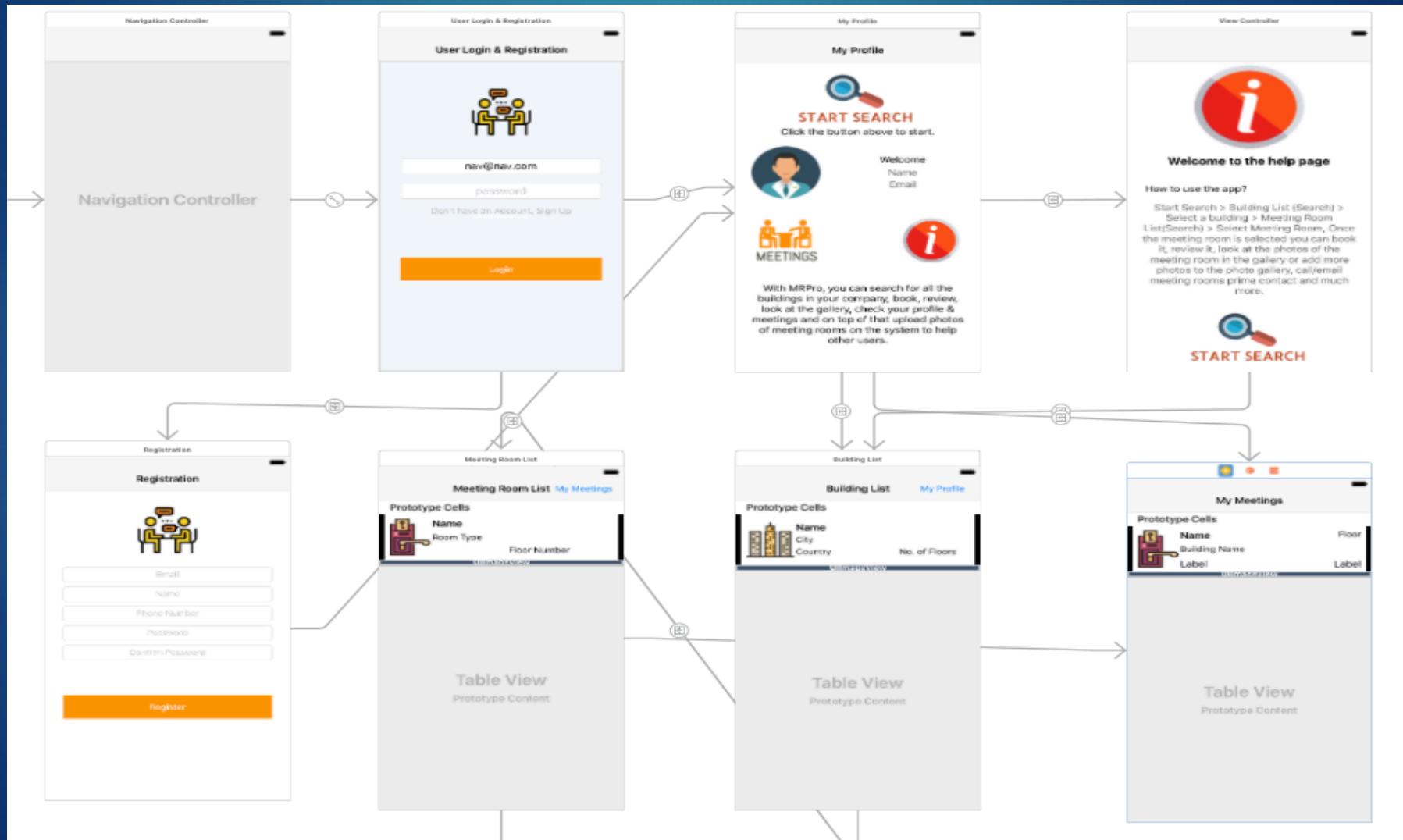
Jul 31 17, 3:26 p.m.

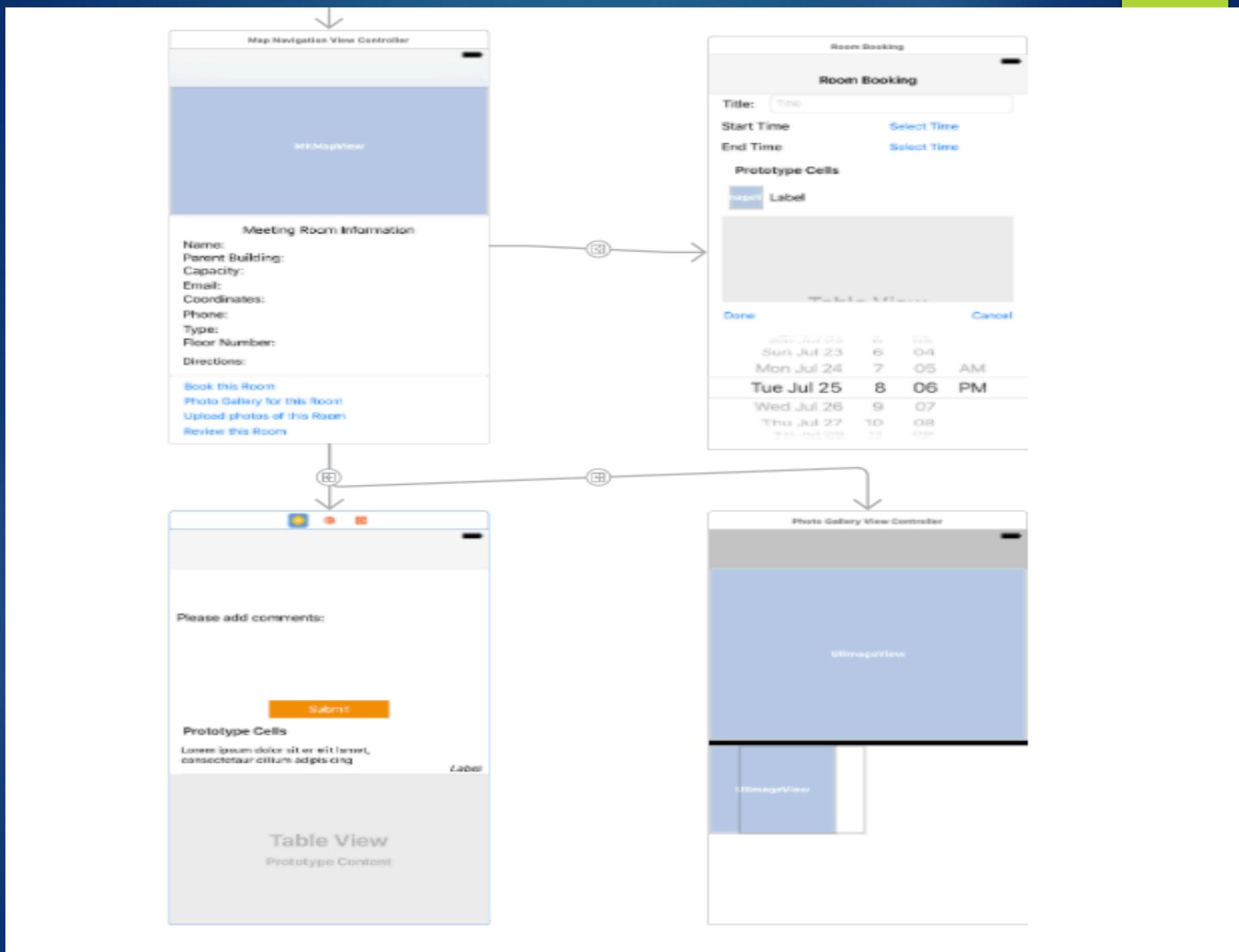
Jul 31 17, 3:37 p.m.

# Others features



# GUI Implementation using Xcode





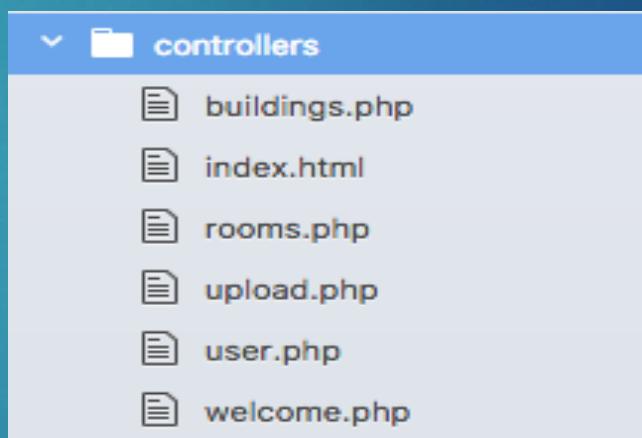
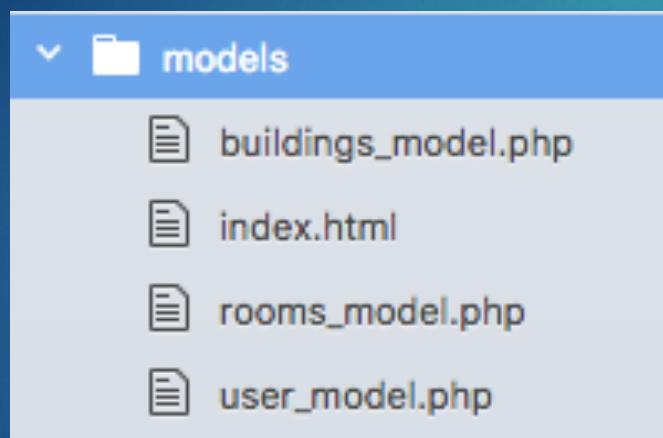
# Environmental Requirements

**Client** – iOS 10+ iOS Devices  
Internet Connection, GPS

**Backend** – MySQL database version with  
Geospatial data support.

# Web Service

Web service is developed using MVC approach in PHP.



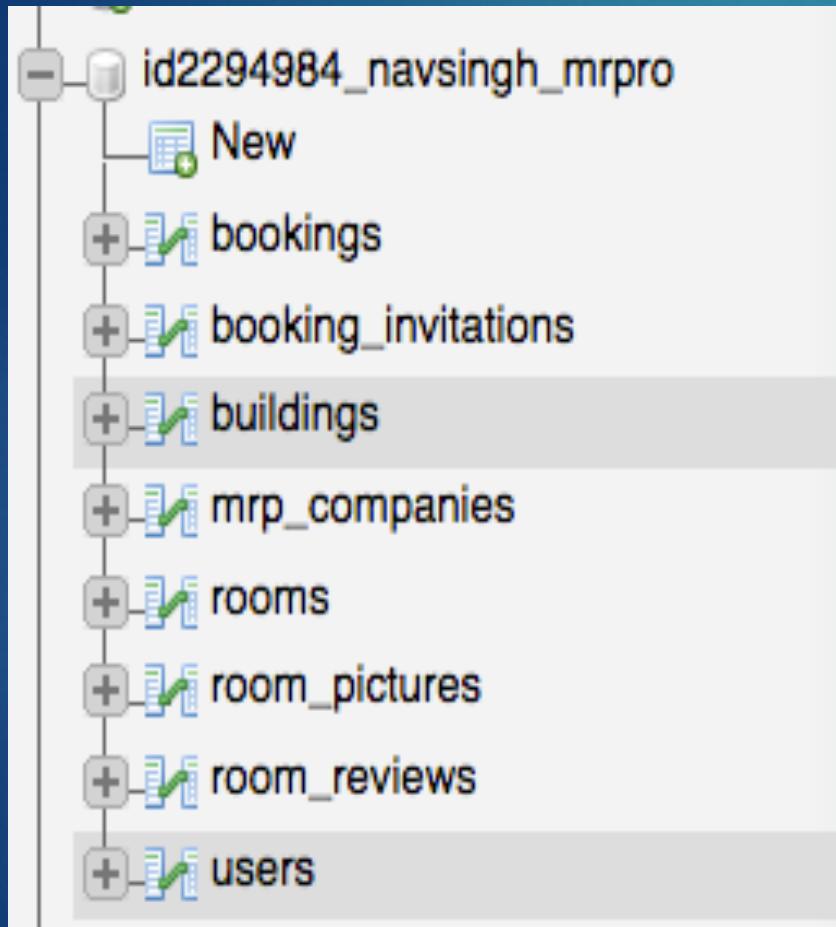
# Data Requirements

The app relies heavenly on data

Explain ?

# Database Design

All Tables



Meeting Rooms Table

#		Name	Type	Collation	Attrib
	1	roomId	int(120)		
	2	buildingId	int(120)		
	3	name	varchar(120)	utf8_unicode_ci	
	4	floorNumber	int(120)		
	5	coordinates	point		
	6	capacity	int(120)		
	7	type	varchar(120)	utf8_unicode_ci	
	8	phone	varchar(120)	utf8_unicode_ci	
	9	email	varchar(120)	utf8_unicode_ci	
	10	directions	varchar(480)	utf8_unicode_ci	
	11	location	varchar(1000)	utf8_unicode_ci	

## Buildings Table

Table structure

Relation view

#	Name	Type	Collation	Attrib
1	<b>id</b> 	int(11)		
2	<b>name</b>	text	utf8_general_ci	
3	<b>city</b>	text	utf8_general_ci	
4	<b>country</b>	text	utf8_general_ci	
5	<b>floors</b>	varchar(10)	utf8_general_ci	
6	<b>address</b>	text	utf8_general_ci	
7	<b>phone</b>	varchar(40)	utf8_general_ci	
8	<b>email</b>	varchar(30)	utf8_general_ci	
9	<b>coordinates</b>	polygon		
10	<b>location</b>	varchar(1000)	utf8_general_ci	

## Reviews Table

Table structure

Relation view

#	Name	Type	Collation
1	<b>id</b> 	int(11)	
2	<b>roomID</b>	int(11)	
3	<b>review</b>	text	utf8_unicode_ci
4	<b>rating</b>	float	
5	<b>userID</b>	int(11)	
6	<b>userName</b>	varchar(100)	utf8_unicode_ci
7	<b>time</b>	timestamp	

# Data Collection



## Building 1- NCI Specifically

53.34888 -6.2435, 53.34875 -6.24351, 53.34853 -6.24352, 53.34853 -6.2435, 53.34851 -6.2435,  
53.3485 -6.24327, 53.34854 -6.24326, 53.34853 -6.24312, 53.34883 -6.24311, 53.34889 -6.2431,  
53.34891 -6.24336, 53.34889 -6.24337, 53.3489 -6.24346, 53.34888 -6.24346, 53.34888 -6.2435

## Building 2 – NCI Area

53.34888 -6.2435, 53.34875 -6.24351, 53.34853 -6.24352, 53.34853 -6.2435, 53.34851 -6.2435,  
53.3485 -6.24327, 53.34854 -6.24326, 53.34853 -6.24312, 53.34847 -6.24312, 53.34844 -6.24266,  
53.34843 -6.24243, 53.34844 -6.24243, 53.34843 -6.24231, 53.34844 -6.24231, 53.34843 -6.24221,  
53.34848 -6.2422, 53.34847 -6.24216, 53.34896 -6.24208, 53.34897 -6.24216, 53.34899 -6.24217,  
53.34898 -6.24204, 53.34906 -6.24202, 53.34906 -6.24204, 53.34909 -6.24203, 53.34912 -6.24243,  
53.34916 -6.24304, 53.34889 -6.2431, 53.34891 -6.24336, 53.34889 -6.24337, 53.3489 -6.24346,  
53.34888 -6.24346, 53.34888 -6.2435

## Building 3 – St james hospital

53.33973 -6.29482, 53.33976 -6.29497, 53.33974 -6.29512, 53.33997 -6.29577, 53.33983 -6.29634,  
53.33986 -6.29638, 53.33981 -6.29663, 53.33978 -6.29663, 53.33962 -6.29734, 53.33965 -6.29738,  
53.33959 -6.29768, 53.33927 -6.29746, 53.33925 -6.29745, 53.3392 -6.29769, 53.33929 -6.2978,  
53.33919 -6.29807, 53.33919 -6.29814, 53.33916 -6.29823, 53.33924 -6.29833, 53.33922 -6.29837,  
53.33948 -6.29864, 53.33951 -6.29859, 53.33948 -6.29846, 53.33944 -6.29847, 53.33955 -6.29829,  
53.3397 -6.29846, 53.33971 -6.29853, 53.33978 -6.29856, 53.33973 -6.29869, 53.33975 -6.29874,  
53.33972 -6.29883, 53.33959 -6.29871, 53.33954 -6.29885, 53.33964 -6.29896, 53.3396 -6.29909,  
53.33953 -6.29903, 53.33936 -6.29944, 53.339 -6.29901, 53.33898 -6.29892, 53.33908 -6.29866,  
53.3391 -6.29889, 53.33914 -6.29861, 53.33927 -6.29873, 53.33869 -6.29809, 53.33867 -6.29816,  
53.33852 -6.29799, 53.33854 -6.29794, 53.33843 -6.29787, 53.33829 -6.29773, 53.33823 -6.29758,  
53.33834 -6.2973, 53.33823 -6.29722, 53.33838 -6.29641, 53.33858 -6.2965, 53.33869 -6.29603,  
53.33837 -6.29579, 53.33842 -6.29555, 53.33876 -6.29575, 53.33874 -6.29582, 53.339 -6.29601,  
53.3393 -6.29463, 53.33918 -6.29455, 53.33932 -6.29391, 53.33946 -6.294, 53.33944 -6.2941,  
53.33964 -6.29424, 53.33959 -6.29445, 53.33941 -6.29435, 53.33939 -6.29443, 53.33942 -6.29447,  
53.33939 -6.29464, 53.33964 -6.29479, 53.3397 -6.29448, 53.33962 -6.29442, 53.33967 -6.29425,  
53.33976 -6.29429, 53.33979 -6.29414, 53.3397 -6.29406, 53.33976 -6.29384, 53.33996 -6.29398,  
53.33972 -6.29492, 53.33973 -6.29482

## Building 4 – SAP

# Web Service/API End-Points

Web service URL

<https://mrpro.000webhostapp.com/MRProApp/MRPro/MRPro/index.php/>

API Request end-points

```
let API_BuildingSearch = "buildings/buildings_search"  
let API_GetAllUsers = "user/getUsers"  
let API_BookRoom = "rooms/bookRoom"  
let API_GetPhotos = "rooms/getRoomsImage"  
let API_UploadImage = "rooms/room_image"  
let API_PostReview = "rooms/postReview"  
let API_MyMeetingRooms = "rooms/getAllBookings"  
let API_RoomRatings = "rooms/getRoomReviews"
```

And more..

# Web Service (Returning list of all buildings) in JSON

```
$ curl -v \
-X POST \
-H "Content-Type: application/x-www-form-urlencoded; charset=utf-8" \
-H "Accept-Language: en-IE;q=1.0, hi-IE;q=0.9" \
-H "User-Agent: MRPro/1.0 (ios.nsv.MRPro; build:1; iOS 10.2.1) Alamofire/4.5.0" \
-H "Accept-Encoding: gzip;q=1.0, compress;q=0.5" \
-d "" \
"https://mrpro.000webhostapp.com/MRProApp/MRPro/index.php/buildings/allBuildings"

{
  data =      (
    {
      address = "Mayor Street, IFSC, Dublin 1, D01 Y300";
      city = Dublin;
      country = Ireland;
      email = "virksaabnavjot@gmail.com";
      floors = 4;
      id = 2;
      location = "53.34888 -6.2435, 53.34875 -6.24351, 53.34853 -6.24352, 53.34853 -6.2435,
53.34851 -6.2435, 53.3485 -6.24327, 53.34854 -6.24326, 53.34853 -6.24312, 53.34847 -6.24312, 53.34844
-6.24266, 53.34843 -6.24243, 53.34844 -6.24243, 53.34843 -6.24231, 53.34844 -6.24231, 53.34843
-6.24221, 53.34848 -6.2422, 53.34847 -6.24216, 53.34896 -6.24208, 53.34897 -6.24216, 53.34899 -6.24217,
53.34898 -6.24204, 53.34906 -6.24202, 53.34906 -6.24204, 53.34909 -6.24203, 53.34912 -6.24243, 53.34916
-6.24304, 53.34889 -6.2431, 53.34891 -6.24336, 53.34889 -6.24337, 53.3489 -6.24346, 53.34888 -6.24346,
53.34888 -6.2435";
      name = "National College of Ireland";
      phone = "+353892110549";
    },
    {
      address = "Mayor Street, IFSC, Dublin 1, D01 Y300";
      city = Dublin;
      country = Ireland;
      email = "virksaabnavjot@gmail.com";
      floors = 3;
      id = 3;
      location = "53.34888 -6.2435, 53.34875 -6.24351, 53.34853 -6.24352, 53.34853 -6.2435,
53.34851 -6.2435, 53.3485 -6.24327, 53.34854 -6.24326, 53.34853 -6.24312, 53.34883 -6.24311, 53.34889
-6.2431, 53.34891 -6.24336, 53.34889 -6.24337, 53.3489 -6.24346, 53.34888 -6.24346, 53.34888 -6.2435";
      name = "NCI";
    }
  )
}
```

Parsed JSON Data Sample  
returned by the Web Service  
through http request.  
Similar data is  
consumed by the client  
Application.

# Evaluation

How will you evaluate the system?

Unit Testing and User Testing and Heuristic Evaluation is incorporated in the project for evaluation.

What is Unit Testing ?

Unit Testing is a testing by which individual units of code are tested to determine whether they fit for use.

What is User Testing ?

User testing is the most effective way to discover any barriers that users will face.

What is Heuristic Evaluation ?

Usability inspection method for computer software.

# Unit Testing

```
/*
function to ensure the json data for building object can be parsed
successfully or not
*/

func testCanParseBuildingInfoFromJson() {

    //Custom class for parsing building and meeting room json
    let jsonParser = CustomJsonParser() //creating instance of parser class
    let jsonData = jsonFromFile() //getting the json from file
    let buildings = jsonParser.parseBuildingJson(jsonData)
    let extractedBuilding = buildings[0]// getting the building at index zero

    //saving all the building information in variables
    let buildingName = extractedBuilding.name
    let buildingId = extractedBuilding.id
    let city = extractedBuilding.city
    let country = extractedBuilding.country
    let numberOfFloors = extractedBuilding.numberOfFloors

    //testing with the data from file and check if tests fail or pass
    XCTAssertEqual(buildingName, "NCI")
    XCTAssertEqual(buildingId, "DUB01")
    XCTAssertEqual(city, "Dublin")
    XCTAssertEqual(country, "Ireland")
    XCTAssertEqual(numberOfFloors, 3)
}
```

```
/*
Function to test if building coordinates from json can be parsed
*/

func testCanParseBuildingCoordinatesFromJson() {
    //creating instance of parser class
    let jsonParser = CustomJsonParser()
    //calling the json from file function that returns json data
    let jsonData = jsonFromFile()

    //creating an array of coordinates
    let buildingCoordinates = [
        CLLocationCoordinate2D(latitude: 53.294974000000003, longitude: -6.4266310000000004),
        CLLocationCoordinate2D(latitude: 53.294846999999997, longitude: -6.4264190000000001),
        CLLocationCoordinate2D(latitude: 53.294288999999999, longitude: -6.4268879999999999),
        CLLocationCoordinate2D(latitude: 53.294325999999998, longitude: -6.427194000000001),
        CLLocationCoordinate2D(latitude: 53.294974000000003, longitude: -6.4266310000000004)
    ]

    //creating a variable and setting it equal to all the building data
    let buildings = jsonParser.parseBuildingJson(jsonData)
    //get the building at index zero
    let extractedBuildingShape = buildings[0]

    //for loop - that iterates through the building coordinates and
    //check if the longitude and latitude at each index matches or not
    for i in 0..<buildingCoordinates.count {

        //long and lat from the created array of coordinates - buildingCoordinates
        let longitude = buildingCoordinates[i].longitude
        let latitude = buildingCoordinates[i].latitude

        //long and lat from the building from json file
        let extractedLatitude = extractedBuildingShape.coordinates[i].latitude
        let extractedLongitude = extractedBuildingShape.coordinates[i].longitude

        //testing
        XCTAssertEqual(longitude, extractedLongitude)
        XCTAssertEqual(latitude, extractedLatitude)
    }
}
```

# User Testing (Survey)

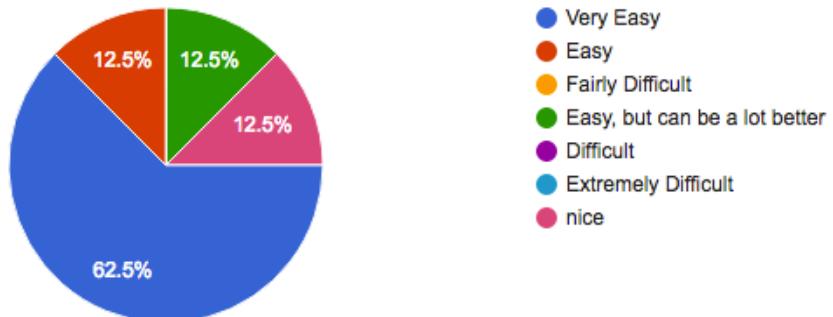
User testing was done by giving users to use app demo and do a survey on google forms.

## Survey Results: Positive Feedback

- ▶ I liked the app would like to see future releases and features of the app, I would like this app at my workplace to find meeting rooms
- ▶ I enjoyed testing the app
- ▶ Nice app
- ▶ Very nice app I think all hotels should have it

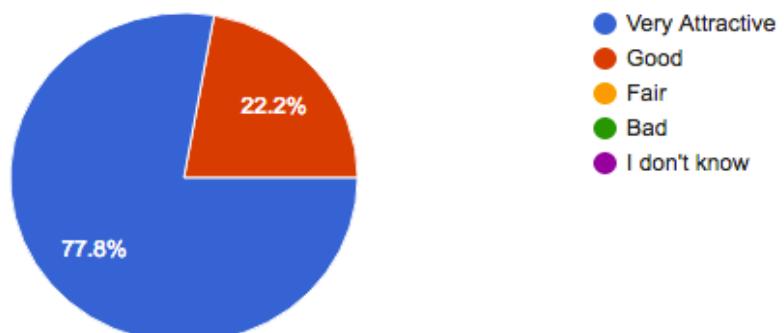
### How easy is to navigate around the app without any help?

8 responses



### How attractive are the features of the app?

9 responses



# Heuristic Evaluation

- ▶ Visibility of System Status
- ▶ Match b/w System and Real World
- ▶ User Control and Freedom
- ▶ Consistency and Standards
- ▶ Error Prevention
- ▶ Recognition rather than recall
- ▶ Flexibility and Efficiency of Use
- ▶ Aesthetic and Minimalistic design
- ▶ Helps users recognise, diagnose, and recover from errors
- ▶ Help and documentation

# Security

The app is very secure.

1. Uses https secure web service
2. Uses Alamofire and SwiftyJSON libraries that helps to securely handle JSON data from web service using https request.

# Demonstration

DEMO OF THE APPLICATION ON REAL DEVICE.

# Discussions

- ▶ Advantages ?

The application is easy to use and very useful.

- ▶ Disadvantages and Limits ?

Huge Data requirements, need many improvements.

- ▶ Opportunities ?

International expansion as the application is very scalable.

- ▶ Future Perspectives?

Indoor Navigation, using beacons or Wi-Fi - triangulation.

# The End



Questions and Feedback.

Thanks