



UNIVERSITEIT VAN PRETORIA  
UNIVERSITY OF PRETORIA  
YUNIBESITHI YA PRETORIA

---

NAVUP  
TESTING REPORT

---

ANDRIOD-GIS

MFANA MASIMULA  
12077713

BONGANI TSHELA  
14134790

JOSHUA MOODLEY  
14152152

BOIKANYO MODIKO  
15227678

GITHUB [HTTPS://GITHUB.COM/SIRJOSH/ANDROID-GIS.GIT](https://github.com/SirJosh/Android-GIS.git)

# Contents

<b>Table Of Contents</b>	<b>1</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Scope . . . . .	2
<b>2 Functional Requirements</b>	<b>2</b>
2.1 Create Functionality . . . . .	2
2.1.1 Add a location . . . . .	2
2.2 Remove Functionality . . . . .	3
2.2.1 Remove location . . . . .	3
2.3 Update Functionality . . . . .	4
2.3.1 Update location . . . . .	4
2.4 Request Functionality . . . . .	5
2.4.1 Get all locations . . . . .	5
2.4.2 Get all buildings . . . . .	5
2.4.3 Get location by building name . . . . .	6
2.4.4 Get route . . . . .	6
<b>3 Non-Functional Requirements Tested</b>	<b>6</b>
3.1 Reliability . . . . .	6
3.2 Availability . . . . .	6
3.3 Scalability . . . . .	6
3.4 Recoverability . . . . .	6

# 1 Introduction

## 1.1 Scope

Services to gather, maintain, persist and provide information related to the world serviced by the system. It is about the creation and maintenance of a GIS Map of the campus by using WiFi signal strengths and other available sources of GIS information. This module provide services to search for locations such as landmarks, buildings as well as venues such as offices, lecture halls, labs, etc.

## 2 Functional Requirements

## 2.1 Create Functionality

### 2.1.1 Add a location

Pass/Fail	Mark (10)	Reason
✓	8	The function works well when given the correct parameters for single object insert and fails for batch inserts (tested in the code above).

```

1 //var location = require('request').get('https://contraints/location/locations/');
2 //var cityStats = require('request').get('http://www.citystats.net/');
3
4 var should = require('should');
5 var media = require('media');
6
7 describe('location create function: ', function() {
8   it('Should add new locations: ', function(done) {
9     [
10       {
11         location_type: 'Neighborhood',
12         name: 'Testname',
13         building: 'Testbuilding',
14         lat: 28.762481,
15         lng: 28.293426,
16         level: 4,
17         ground: 1.5
18       },
19     ],
20     function(err, res) {
21       //var helloworldData = {
22         {
23           "location_type": "Neighborhood",
24           "name": "Testname",
25           "building": "Testbuilding",
26           "lat": 28.762481,
27           "lng": 28.293426,
28           "level": 4,
29           "ground": 1.5
30         },
31         {
32           "location_type": "Neighborhood",
33           "name": "Testname",
34           "building": "Testbuilding",
35           "lat": 28.762481,
36           "lng": 28.293426,
37           "level": 4,
38           "ground": 1.5
39         },
40         {
41           "location_type": "Neighborhood",
42           "name": "Testname",
43           "building": "Testbuilding",
44           "lat": 28.762481,
45           "lng": 28.293426,
46           "level": 4,
47           "ground": 1.5
48         }
49       ],
50       function(err, res) {
51         //var helloworldData = {
52           {
53             "location_type": "Neighborhood",
54             "name": "Testname",
55             "building": "Testbuilding",
56             "lat": 28.762481,
57             "lng": 28.293426,
58             "level": 4,
59             "ground": 1.5
60           },
61           {
62             "location_type": "Neighborhood",
63             "name": "Testname",
64             "building": "Testbuilding",
65             "lat": 28.762481,
66             "lng": 28.293426,
67             "level": 4,
68             "ground": 1.5
69           },
70           {
71             "location_type": "Neighborhood",
72             "name": "Testname",
73             "building": "Testbuilding",
74             "lat": 28.762481,
75             "lng": 28.293426,
76             "level": 4,
77             "ground": 1.5
78           }
79         ],
80         function(err, res) {
81           //var helloworldData = {
82             {
83               "location_type": "Neighborhood",
84               "name": "Testname",
85               "building": "Testbuilding",
86               "lat": 28.762481,
87               "lng": 28.293426,
88               "level": 4,
89               "ground": 1.5
90             },
91             {
92               "location_type": "Neighborhood",
93               "name": "Testname",
94               "building": "Testbuilding",
95               "lat": 28.762481,
96               "lng": 28.293426,
97               "level": 4,
98               "ground": 1.5
99             },
100             {
101               "location_type": "Neighborhood",
102               "name": "Testname",
103               "building": "Testbuilding",
104               "lat": 28.762481,
105               "lng": 28.293426,
106               "level": 4,
107               "ground": 1.5
108             }
109           ],
110           function(err, res) {
111             //var helloworldData = {
112               {
113                 "location_type": "Neighborhood",
114                 "name": "Testname",
115                 "building": "Testbuilding",
116                 "lat": 28.762481,
117                 "lng": 28.293426,
118                 "level": 4,
119                 "ground": 1.5
120               },
121               {
122                 "location_type": "Neighborhood",
123                 "name": "Testname",
124                 "building": "Testbuilding",
125                 "lat": 28.762481,
126                 "lng": 28.293426,
127                 "level": 4,
128                 "ground": 1.5
129               },
130               {
131                 "location_type": "Neighborhood",
132                 "name": "Testname",
133                 "building": "Testbuilding",
134                 "lat": 28.762481,
135                 "lng": 28.293426,
136                 "level": 4,
137                 "ground": 1.5
138               }
139             ],
140             function(err, res) {
141               //var helloworldData = {
142                 {
143                   "location_type": "Neighborhood",
144                   "name": "Testname",
145                   "building": "Testbuilding",
146                   "lat": 28.762481,
147                   "lng": 28.293426,
148                   "level": 4,
149                   "ground": 1.5
150                 },
151                 {
152                   "location_type": "Neighborhood",
153                   "name": "Testname",
154                   "building": "Testbuilding",
155                   "lat": 28.762481,
156                   "lng": 28.293426,
157                   "level": 4,
158                   "ground": 1.5
159                 },
160                 {
161                   "location_type": "Neighborhood",
162                   "name": "Testname",
163                   "building": "Testbuilding",
164                   "lat": 28.762481,
165                   "lng": 28.293426,
166                   "level": 4,
167                   "ground": 1.5
168                 }
169               ],
170               function(err, res) {
171                 //var helloworldData = {
172                   {
173                     "location_type": "Neighborhood",
174                     "name": "Testname",
175                     "building": "Testbuilding",
176                     "lat": 28.762481,
177                     "lng": 28.293426,
178                     "level": 4,
179                     "ground": 1.5
180                   },
181                   {
182                     "location_type": "Neighborhood",
183                     "name": "Testname",
184                     "building": "Testbuilding",
185                     "lat": 28.762481,
186                     "lng": 28.293426,
187                     "level": 4,
188                     "ground": 1.5
189                   },
190                   {
191                     "location_type": "Neighborhood",
192                     "name": "Testname",
193                     "building": "Testbuilding",
194                     "lat": 28.762481,
195                     "lng": 28.293426,
196                     "level": 4,
197                     "ground": 1.5
198                   }
199                 ],
200                 function(err, res) {
201                   //var helloworldData = {
202                     {
203                       "location_type": "Neighborhood",
204                       "name": "Testname",
205                       "building": "Testbuilding",
206                       "lat": 28.762481,
207                       "lng": 28.293426,
208                       "level": 4,
209                       "ground": 1.5
210                     },
211                     {
212                       "location_type": "Neighborhood",
213                       "name": "Testname",
214                       "building": "Testbuilding",
215                       "lat": 28.762481,
216                       "lng": 28.293426,
217                       "level": 4,
218                       "ground": 1.5
219                     },
220                     {
221                       "location_type": "Neighborhood",
222                       "name": "Testname",
223                       "building": "Testbuilding",
224                       "lat": 28.762481,
225                       "lng": 28.293426,
226                       "level": 4,
227                       "ground": 1.5
228                     }
229                   ],
230                   function(err, res) {
231                     //var helloworldData = {
232                       {
233                         "location_type": "Neighborhood",
234                         "name": "Testname",
235                         "building": "Testbuilding",
236                         "lat": 28.762481,
237                         "lng": 28.293426,
238                         "level": 4,
239                         "ground": 1.5
240                       },
241                       {
242                         "location_type": "Neighborhood",
243                         "name": "Testname",
244                         "building": "Testbuilding",
245                         "lat": 28.762481,
246                         "lng": 28.293426,
247                         "level": 4,
248                         "ground": 1.5
249                       },
250                       {
251                         "location_type": "Neighborhood",
252                         "name": "Testname",
253                         "building": "Testbuilding",
254                         "lat": 28.762481,
255                         "lng": 28.293426,
256                         "level": 4,
257                         "ground": 1.5
258                       }
259                     ],
260                     function(err, res) {
261                       //var helloworldData = {
262                         {
263                           "location_type": "Neighborhood",
264                           "name": "Testname",
265                           "building": "Testbuilding",
266                           "lat": 28.762481,
267                           "lng": 28.293426,
268                           "level": 4,
269                           "ground": 1.5
270                         },
271                         {
272                           "location_type": "Neighborhood",
273                           "name": "Testname",
274                           "building": "Testbuilding",
275                           "lat": 28.762481,
276                           "lng": 28.293426,
277                           "level": 4,
278                           "ground": 1.5
279                         },
280                         {
281                           "location_type": "Neighborhood",
282                           "name": "Testname",
283                           "building": "Testbuilding",
284                           "lat": 28.762481,
285                           "lng": 28.293426,
286                           "level": 4,
287                           "ground": 1.5
288                         }
289                       ],
290                       function(err, res) {
291                         //var helloworldData = {
292                           {
293                             "location_type": "Neighborhood",
294                             "name": "Testname",
295                             "building": "Testbuilding",
296                             "lat": 28.762481,
297                             "lng": 28.293426,
298                             "level": 4,
299                             "ground": 1.5
300                           },
301                           {
302                             "location_type": "Neighborhood",
303                             "name": "Testname",
304                             "building": "Testbuilding",
305                             "lat": 28.762481,
306                             "lng": 28.293426,
307                             "level": 4,
308                             "ground": 1.5
309                           },
310                           {
311                             "location_type": "Neighborhood",
312                             "name": "Testname",
313                             "building": "Testbuilding",
314                             "lat": 28.762481,
315                             "lng": 28.293426,
316                             "level": 4,
317                             "ground": 1.5
318                           }
319                         ],
320                         function(err, res) {
321                           //var helloworldData = {
322                             {
323                               "location_type": "Neighborhood",
324                               "name": "Testname",
325                               "building": "Testbuilding",
326                               "lat": 28.762481,
327                               "lng": 28.293426,
328                               "level": 4,
329                               "ground": 1.5
330                             },
331                             {
332                               "location_type": "Neighborhood",
333                               "name": "Testname",
334                               "building": "Testbuilding",
335                               "lat": 28.762481,
336                               "lng": 28.293426,
337                               "level": 4,
338                               "ground": 1.5
339                             },
340                             {
341                               "location_type": "Neighborhood",
342                               "name": "Testname",
343                               "building": "Testbuilding",
344                               "lat": 28.762481,
345                               "lng": 28.293426,
346                               "level": 4,
347                               "ground": 1.5
348                             }
349                           ],
350                           function(err, res) {
351                             //var helloworldData = {
352                               {

```

```
Location CREATE function:
  ✓ Should add new Locations: (723ms)
```

**Figure 2:** Result after test is run: status code 200

**Figure 1:** Test code for add location function

## 2.2 Remove Functionality

### 2.2.1 Remove location

Pass/Fail	Mark (10)	Reason
✓	10	The remove function works well when given the correct parameters It will get the exact location that we want to remove, using both the building details and the coordinates

```
1 //var assert = require('assert');
2 //var locations = require('../controllers/locations/locations');
3 //var http_mocks = require('node-mocks-http');
4 var should = require('should');
5 var needle = require('needle');
6
7 describe('Location UPDATE function: ', function() {
8   it('Should update specified Location: ', function(done)
9   {
10     var locationData = {
11       "room": "testRoom2",
12       "building": "testBuilding2",
13       "lat": -23.762492,
14       "lng": 28.293438,
15       "level": 2,
16       "ground": 2
17     };
18
19     needle.patch('localhost:3000/location', locationData, function(err, res)
20     {
21       if (err) { done(); return console.error(err.message); }
22
23       res.statusCode.should.equal(200)
24       //res.body.should.have.property('data');
25
26       // to display body content
27       //console.log(res.body);
28
29       done();
30     });
31   });
32 });
33 });
```

Figure 3: Test code for remove location function

```
Location CREATE function:
✓ should add another new Location: (70ms)
```

Figure 4: Result after test is run: status code 200

## 2.3 Update Functionality

### 2.3.1 Update location

Pass/Fail	Mark (10)	Reason
✓	7	The update function works correctly given the correct parameters. The concern is that it might not get the right or the only location given only the room and the building (as opposed to room, building and the coordinates).

```
1 //var assert = require('assert');
2 //var locations = require('../controllers/locations/locations');
3 //var http_mocks = require('node-mocks-http');
4 var should = require('should');
5 var needle = require('needle');
6
7 describe('Location DELETE function: ', function() {
8   it('Should remove specified Locations: ', function(done)
9   {
10     var locationData = {
11       "room" : "testRoom",
12       "building" : "testBuilding"
13     };
14
15     needle.delete('localhost:3000/location', locationData, function(err, res)
16     {
17       if (err) { done(); return console.error(err.message); }
18
19       res.statusCode.should.equal(200)
20       //res.body.should.have.property('data');
21
22       // to display body content
23       //console.log(res.body);
24
25       done();
26     });
27   });
28 });
29
30
```

Figure 5: Test code for update function

```
Location READ function:
✓ Should retrieve all Locations: (144ms)
```

Figure 6: Result after test is run: status code 200

## 2.4 Request Functionality

### 2.4.1 Get all locations

Pass/Fail	Mark (10)	Reason
✓	10	<p>The retrieve all locations request returns all the location's stored within the database on campus. The request works when new locations are added. Thus showing that the database performs the request correctly. This also holds true when you call this function multiple times sequentially.</p> <p>Overall the function performs what it is defined to do, without causing strain on the database.</p>

```
//var assert = require('assert');
//var locations = require('../controllers/locations/locations');
//var http_mocks = require('node-mocks-http');
var should = require('should');
var needle = require('needle');

describe('Location READ function: ', function() {
  it('Should retrieve all Locations: ', function(done) {
    needle.get('localhost:3000/locations', function(err, res) {
      if (err) { return console.error(err.message); }

      res.statusCode.should.equal(200)
      res.body.should.have.property('data');

      // to display body content
      //for(var i=0; i < res.body.data.length; i++) console.log(res.body.data[i]);

      done();
    });
  });
});
```

Figure 7: Test code for display all locations

```
{
  "data": [
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e708a",
      "attributes": {
        "location_type": "Entrance",
        "room": "N/A",
        "building": "IT",
        "lat": -25.755864,
        "lng": 28.233146,
        "level": 2,
        "ground": 233
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e708b",
      "attributes": {
        "location_type": "Entrance",
        "room": "N/A",
        "building": "IT",
        "lat": -25.755892,
        "lng": 28.233055,
        "level": 2,
        "ground": 233
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e708c",
      "attributes": {
        "location_type": "Cafeteria",
        "room": "N/A",
        "building": "IT",
        "lat": -25.755705,
        "lng": 28.23345,
        "level": 4,
        "ground": 433
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e708d",
      "attributes": {
        "location_type": "Catwalk",
        "room": "N/A",
        "building": "EMB",
        "lat": -25.755607,
        "lng": 28.233481,
        "level": 4,
        "ground": 433
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e708e",
      "attributes": {
        "location_type": "Entrance",
        "room": "N/A",
        "building": "EMB",
        "lat": -25.755393,
        "lng": 28.233289,
        "level": 2,
        "ground": 233
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e708f",
      "attributes": {
        "location_type": "Entrance",
        "room": "N/A",
        "building": "Centenary",
        "lat": -25.753976,
        "lng": 28.232996,
        "level": 1,
        "ground": 133
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e7090",
      "attributes": {
        "location_type": "Entrance",
        "room": "N/A",
        "building": "Thuto",
        "lat": -25.752888,
        "lng": 28.231594,
        "level": 1,
        "ground": 133
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e7091",
      "attributes": {
        "location_type": "Entrance",
        "room": "N/A",
        "building": "Student Services",
        "lat": -25.755237,
        "lng": 28.232293,
        "level": 1,
        "ground": 133
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e7092",
      "attributes": {
        "location_type": "Entrance",
        "room": "N/A",
        "building": "Humanities",
        "lat": -25.755533,
        "lng": 28.230773,
        "level": 3,
        "ground": 333
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e7093",
      "attributes": {
        "location_type": "Entrance",
        "room": "N/A",
        "building": "Merensky Library",
        "lat": -25.755384,
        "lng": 28.230537,
        "level": 3,
        "ground": 333
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e7094",
      "attributes": {
        "location_type": "Entrance",
        "room": "N/A",
        "building": "Theology",
        "lat": -25.755046,
        "lng": 28.229466,
        "level": 1,
        "ground": 133
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e7095",
      "attributes": {
        "location_type": "Entrance",
        "room": "N/A",
        "building": "Chemistry",
        "lat": -25.753052,
        "lng": 28.230763,
        "level": 1,
        "ground": 133
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e7096",
      "attributes": {
        "location_type": "Entrance",
        "room": "N/A",
        "building": "EMB",
        "lat": -25.755062,
        "lng": 28.232988,
        "level": 2,
        "ground": 233
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e7097",
      "attributes": {
        "location_type": "Entrance",
        "room": "N/A",
        "building": "EMB",
        "lat": -25.755207,
        "lng": 28.232936,
        "level": 2,
        "ground": 233
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e7098",
      "attributes": {
        "location_type": "Entrance",
        "room": "N/A",
        "building": "Piazza",
        "lat": -25.754648,
        "lng": 28.232036,
        "level": 0,
        "ground": 033
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e7099",
      "attributes": {
        "location_type": "Entrance",
        "room": "N/A",
        "building": "Piazza",
        "lat": -25.754266,
        "lng": 28.231505,
        "level": 0,
        "ground": 033
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e709a",
      "attributes": {
        "location_type": "Entrance",
        "room": "N/A",
        "building": "Piazza",
        "lat": -25.753976,
        "lng": 28.231567,
        "level": 0,
        "ground": 033
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e709b",
      "attributes": {
        "location_type": "Point",
        "room": "N/A",
        "building": "N/A",
        "lat": -25.755355,
        "lng": 28.232237,
        "level": 0,
        "ground": 033
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e709c",
      "attributes": {
        "location_type": "Point",
        "room": "N/A",
        "building": "N/A",
        "lat": -25.755274,
        "lng": 28.231658,
        "level": 0,
        "ground": 033
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e709d",
      "attributes": {
        "location_type": "Point",
        "room": "N/A",
        "building": "N/A",
        "lat": -25.755211,
        "lng": 28.231305,
        "level": 0,
        "ground": 033
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e709e",
      "attributes": {
        "location_type": "Point",
        "room": "N/A",
        "building": "N/A",
        "lat": -25.755137,
        "lng": 28.230924,
        "level": 0,
        "ground": 033
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e709f",
      "attributes": {
        "location_type": "Point",
        "room": "N/A",
        "building": "N/A",
        "lat": -25.755125,
        "lng": 28.23081,
        "level": 0,
        "ground": 033
      }
    },
    {
      "type": "locations",
      "id": "5909c7a4d063dc70755e70a0",
      "attributes": {
        "location_type": "Point",
        "room": "N/A",
        "building": "N/A",
        "lat": -25.755072,
        "lng": 28.230897,
        "level": 0,
        "ground": 033
      }
    }
  ]
}
```

Figure 8: JSON string returned

```
Location READ function:
✓ Should retrieve all Locations: (54ms)
```

Figure 9: Result after test is run: status code 200

### 2.4.2 Get all buildings

Pass/Fail	Mark (10)	Reason
✓	9	<p>After testing the getBuildingNames request it retrieves all the buildings that are on campus. The request also works when new buildings are added. Thus showing that the database handles the request correctly. This also holds true when you call this function multiple times sequentially.</p> <p>Overall the function performs what it needs to do, without causing the database to crash or hang.</p>

```
//var assert = require('assert');
//var locations = require('../controllers/locations/locations');
//var http_mocks = require('node-mocks-http');
var should = require('should');
var needle = require('needle');

describe('Retrieve building names function: ', function() {
  it('Should retrieve a list of all available buildings: ', function(done) {
    needle.get('localhost:3000/locations/getBuildingNames', function(err, res) {
      if (err) { return console.error(err.message); }

      res.statusCode.should.equal(200)
      res.body.should.have.property('data');

      // to display body content
      //for(var i=0; i < res.body.data.length; i++) console.log(res.body.data[i]);

      done();
    });
  });
});
```

Figure 10: Test code for display all function

```
Retrieve building names function:
✓ Should retrieve a list of all available buildings:
```

Figure 11: Result after test is run: status code 200

```
{
  "data": [
    "IT",
    "EMB",
    "Centenary",
    "Thuto",
    "Student Services",
    "Humanities",
    "Merensky Library",
    "Theology",
    "Chemistry",
    "Piazza",
    "N/A",
    "testBuilding2"
  ]
}
```

Figure 12: JSON string returned

### 2.4.3 Get location by building name

Pass/Fail	Mark (10)	Reason
✓	0	this is some text that is a lot of text can you -lease wraparound so it omdfshjvafkdghjvnadhjkbknadf vjh,kandfb vadfbadfbfsf

### 2.4.4 Get route

Pass/Fail	Mark (10)	Reason
✓	0	this is some text that is a lot of text can you -lease wraparound so it omdfshjvafkdghjvnadhjkbknadf vjh,kandfb vadfbadfbfsf

## 3 Non-Functional Requirements Tested

### 3.1 Reliability

Pass/Fail	Mark (10)	Reason
✓	8	The database successfully supports all CRUD operations. It is very consistent and works smoothly.

### 3.2 Availability

Pass/Fail	Mark (10)	Reason
✓	10	The GIS subsystem is always available and downtime was never experienced during testing.

### 3.3 Scalability

Pass/Fail	Mark (10)	Reason
✓	9	The system is fairly scalable and can handle multiple interactions without actually slowing down

### 3.4 Recoverability

Pass/Fail	Mark (10)	Reason
✓	0	this is some text that is a lot of text can you -lease wraparound so it omdfshjvafkdghjvnadhjkbknadf vjh,kandfb vadfbadfbfsf