

# GIS team Gladios

April 20, 2017



**UNIVERSITEIT VAN PRETORIA  
UNIVERSITY OF PRETORIA  
YUNIBESITHI YA PRETORIA**

Peter Rayner u14001757  
Nardus van der Vyver u15012698  
Mandla Mashinini u29630135  
Minal Pramlall u13288157

## 1 Introduction

The aim of the GIS module is to provide services to gather ,maintain and provide information related the the spatial data. We used postgreSQL with the postGIS extension to be able use a java interface to be able to communicate with other modules and the postgreSQL database. PostGIS was our server of choice as it proved far superior to its competitors in the time it took for data to be collected,which is very important in an application such as NavUP, and handles multiple calls by using a queue.

## 2 Physical overview

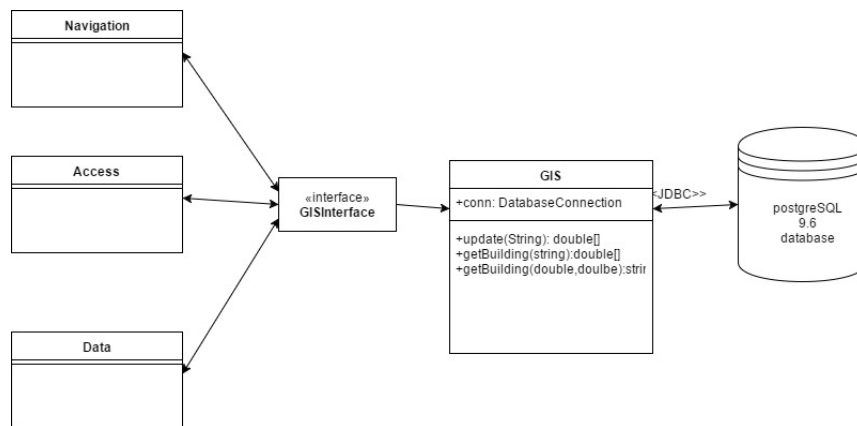


Figure 1: Overview of how GIS interacts

The GIS class requires postgreSQL 9.6 database and a GISInterface for other modules to interact with. The GIS class makes use of a JDBC driver for postgreSQL in order to communicate to the database. All other modules interact with the GISInterface class which then enters directly with the main GIS class itself.

### 3 GIS Class

The GIS class consists of the following functions:

```
public GIS()
```

Default constructor for the class that will initialize the fields and establish a map to the database using the default database name, user name and password to authenticate to the database.

```
public GIS(String username, String database, String password)
```

Constructor for the class that will initialize the fields and establish map to the database.

Parameters:

username String value that holds the username of the database

database: String value that holds the name of the database

password: String value that holds the password which will

```
GIS getInstance()
```

Returns an instance of this current GIS object

```
public void createMap()
```

createMap creates the tables for buildings, campus buildings , entrances, level2 ,level3 ,walls and stairs It also reads in the data from the given JSON objects in the data file to the database. createMap creates the tables for buildings, campus buildings , entrances, level2 ,level3 ,walls and stairs It also reads in the data from the given JSON objects in the data file to the database.

```
public void insertBuilding(String name, String description, String geometry, String coordinates, String table)
```

Inserts a new Building into the database

Parameters:

name: the name of the building  
description: the description of the building  
geometry: the geometry of the building . ie polygon  
coordinates: the coordinates of the building  
table: the table we want to insert the data into.

```
public String getBuilding(String name)
```

Returns the name ,description and geometry of a building

Parameters:

name: the name of the building that is being requested  
return: a String that contains the name description and geometry  
of the building

```
public void addMoreData(String dataFile)
```

Allows us to add more buildings to the buildings file Note: this  
only works with files in the same format as the building file and  
wont work with stairs

Parameters:

dataFile: the name of the file that is requested to update.

```
public ArrayList(String) getLectureHall(String building)
```

gets all the lecture hall names in a given building

Parameters:

building: the name of a given building  
return: an array of strings with lecture hall names

```
public ArrayList(Double) getLectureCoordinates(String room)
```

gets the room name and returns the coordinates of the lecture venue

Parameters:

room: the lecture hall that is being requested eg IT 4-2

return: returns the an arraylist of the coordinates

```
public ArrayList(Double) getBuildingCoordinates(String name)
```

Gives the coordinates of a building name that is passed through to it

Parameters:

name: the name of the building that has been requested

return: the coordinates of the building if it exists in the database

```
public ArrayList(String) getAllBuildings()
```

Gives the names of all the possible buildings.

```
public ArrayList(String) getBuildingInRadius(double mLat, double mLon, double radius)
```

Gets the buildings given in a given radius

Parameters:

mLat: the persons current latitude

mLon: persons current longitude

return: an arraylist of all buildings in the given radius

## 4 Reasoning for use of postgresSQL

The reason we decided to use postgresSQL was due to the way it was able to handle multiple calls simultaneously and the speed and efficiency we were able to achieve with it. PostgresSQL with the postGIS extension enabled us to retrieve data faster than any other databases which we explored. These databases included mongoDB mySQL and SQL server. we found that on average we were able to retrieve data almost a third faster than other databases were able to retrieve data. This meant that we are able to provide very low latency to the rest of the system and allows the system to be very responsive, making it the ideal solution for this sort of application. postgresSQL also allows for interaction directly through the terminal and through Java( as we have used ). This allows extra functionality to be added on later should the need arise and allows multiple different ways to interact with the database