## Database:

MapTbl

* MapID (Primary Key and will be used as a foreign key later on) – Autonumber
* ImgLocation (Location of where the image of the map is stored) - String
* Corner1GPSLatitude (A Corner point latitude coordinate for the map) - double
* Corner1GPSLongitude (A Corner point longitude coordinate for the map) - double
* Corner2GPSLatitude (A Corner point latitude coordinate for the map) - double
* Corner2GPSLongitude (A Corner point longitude coordinate for the map) - double
* Corner3GPSLatitude (A Corner point latitude coordinate for the map) - double
* Corner3GPSLongitude (A Corner point longitude coordinate for the map) - double
* Corner4GPSLatitude (A Corner point latitude coordinate for the map) - double
* Corner4GPSLongitude (A Corner point longitude coordinate for the map) - double

LocationDataTbl

* LocationID (Primary Key and will be used as a foreign key later on) Autonumber
* TypeofPoint (There are three types of points. **Entrance/Exit**, to be connected to other maps ect, **Destination**, the buildings or destinations selected by the people inputting the data into the system and **NodalPathPoint**, which are the created points in the paths used in the adjacency list.) String
* MapID (Foreign key linking the point to a map)
* ImageCoordinateX (The coordinates of the point within the image of the map) Int
* ImageCoordinateY (The coordinates of the point within the image of the map) Int
* GPSLatitude (The GPS latitude coordinate of the point)
* GPSLongitude (the GPS Longitude coordinate of the point)
* Designation (Optional, will be NULL if the point is a **NodalPathPoint** or entrance/exit or there has been no data inputted, can handle more than one by using delimiter: **“ ,”**. String
* Paragraph (Optional, will be NULL if the point is a **NodalPathPoint** or entrance/exit or there has been no data inputted, some information about the destination) String
* LocationOfImage (Optional, will be NULL if the point is a **NodalPathPoint** or entrance/exit or there has been no data inputted, image of the destination) String

InternalMapTbl (this is to allow one map to be a map which is inside another, ie a map of sentae house within a map of East Campus, just for some extra features)

* InternalMapID\_PrimaryKey (Primary Key) Autonumber
* ExternalMapID (foreign Key from MapTbl)
* InternalMapID (foreign Key from MapTbl)

PathTbl (Adjacency List, which will store all the paths on all the respective maps)

* PathID (Primary Key) Autonumber
* CurrentNode (LocationID foreign key connection)
* PreviousNode (LocationID foreign key connection)
* ~~MapOfPath (MapID foreign key connection)~~ Not Needed since it will not make the DB in 3rd Normal Form

UserTbl (users who are allowed to add/remove data)

* UserID (Primary Key) Autonumber
* Username String
* Password String