Nicole White
Fall 2018

Harvest Moon Trailway

GISD3030: Enterprise GIS
GEOM3040: Database Systems

Centre of Geographic Sciences

Contents

Overview

This project was undertaken as the major term project encompassing two fall semester classes fulfilling the requirements of the Centre of Geographic Sciences' 2-year GIS diploma program. In this time, we collaboratively developed a new enterprise geographic information system for a client.

The work included:

- Use of **Agile/scrum** project management methodologies to collaborate productively

- Regular meetings with client and team members

- Creation and refinement of a **spatial database** design; database management and querying using **Structured Query Language** (SQL).

- Creation and maintenance of **metadata** and **technical documentation.**

- Exploration of **enterprise GIS platforms.**

- Web mapping.

Client Profile

The Annapolis Valley Trails Coalition is a nonprofit group consisting of 22 managing entities responsible for the 110km Harvest Moon recreational trail, situated on the former railway corridor running from Grand-Pre to Annapolis Royal. In addition to the trail segments themselves, the AVTC's purview includes asserts such as bridges, signs, and culverts.

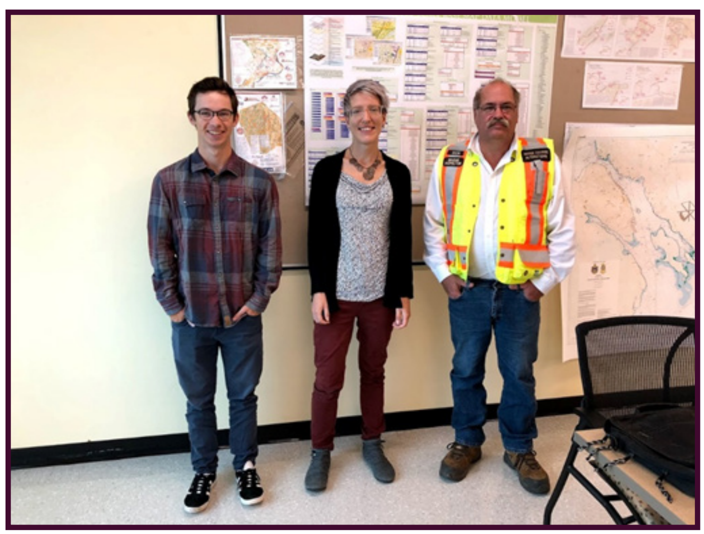
Business operations include trail construction, maintenance, volunteer coordination, and regular field work inspections and risk audits. In recent years, need for construction has diminished and more emphasis is now being placed on maintenance. It is hoped that, moving forward, an increased focus will be placed on research and assessment activities. Much of the documentation related to the AVTC's efforts is currently in hard copy or stored in spreadsheets.

Project Objective

The AVTC require a multiuser geographic information system to be implemented, merging disparate data sources and allowing stakeholders to effectively store, view, query, and manipulate this data. This will allow for **more efficient immediate business operations**, facilitate the **sharing** of data, and provide a **flexible, scalable framework** for future developments.

Client Needs

An initial client meeting was held in September 2018, at the Centre of Geographic Sciences in Lawrencetown, to discuss the project. Rick Jacques, AVTC coordinator, Nicole White and Chris Doyle, COGS GIS students, and Monica Lloyd, COGS GIS faculty, were in attendance.



Several desired features were discussed:

• Inclusion of spatial (trails, bridges, culverts, signage, trailheads, properties, watersheds, road network, nearby tourist attractions, invasive species data, biohazard and other sensitive areas, subsidence zones) and non-spatial data layers (user counts, inspection details, photographs taken during field work). Also, the possibility of integrating the system with other existing trails (Trans Canada, Rum Runners Trail).

• Ability to query features, e.g.: 'How many signs are over ten years old?', 'How many bridges made of steel are in this trail segment?'

• Construction of a web map portal interface, allowing members to view, query, and edit data and containing tools such as measure and draw. The ability to record features during fieldwork was discussed. The web map could also feature a public interface serving trail users and tourists.

The client then identified their top priorities for mapping, which were: Trails (and user counts), Culverts, Bridges, and Signage.

Current Client Resources

Windows 10 PC, Samsung tablet, field GPS device. The individuals who would act as data stewards are computer-literate but have little experience with GIS.

Brainstorming and Industry Research

A Skype meeting was conducted with industry partner Trevor Robar of the Town of Wolfville to gain insights into enterprise GIS solutions. Following this, and using the information provided by Rick in our initial meeting, some independent research was conducted to investigate different workflows and GIS models that could be implemented.

A major consideration at this point was:

Should the solution use Esri's ArcGIS software suite or an open-source alternative?

• Esri's ArcGIS is an industry standard. It is widely supported, and its prevalence may make sharing data with third parties easier. However, licensing issues and the cost of software may be impediments for the client. The finished product in this case would likely consist of a desktop GIS using ArcGIS for Desktop, with web server capabilities provided by ArcGIS Server and ArcGIS Online. Field work could be completed using Collector for ArcGIS and/or Survey123. Esri's desktop software is available only on Microsoft Windows.

• An open-source alternative, such as QGIS, eliminates potential issues associated with licences and fees. Additionally, QGIS Desktop and many similar applications have cross-platform support. However, a greater degree of technical skill, research, and time may be required to successfully implement the best solution. Within the open-source sphere, there exist several different software applications and stacks that could be used in varying combinations in an enterprise GIS framework. Desktop clients include QGIS, uDig, and gvSig. Some open-source GIS web servers are GeoServer, MapServer, and QGIS Server. Mobile apps for field collection are also provided by QGIS and gvSig.

Both proprietary and open-source methods were explored to find the solution most suited to the client's current operations and ongoing needs. Workflow examples, maps, and query demonstrations have been produced for both.

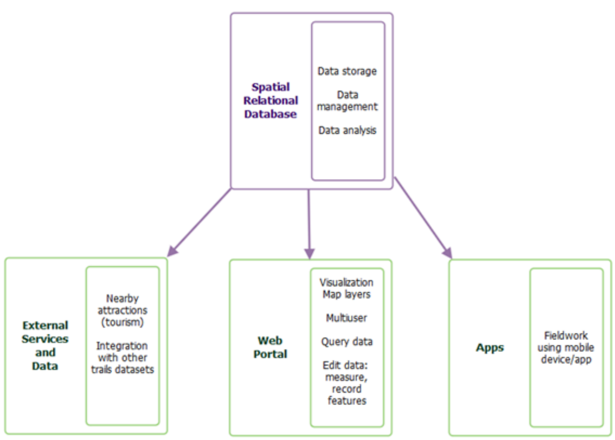
Project Scope

Microsoft Project Professional was used to create a **detailed work breakdown structure (WBS)** and **Gantt chart** to manage the project schedule. These can be found in the appendix.

Deliverables

The provision of a robust, intuitive, centralized enterprise GIS accommodating all operational needs:

- Functioning spatial relational database schema.
- Hosting for data.
- Existing data provided by client migrated into new system.
- Web server allowing users to users to view, query, and edit (update, create, and delete feature instances) spatial data.
- Integration with mobile collection device.



Milestones

September 30	Identification of deliverables.
September 19	Database design complete.
October 19	PostgreSQL database built.
October 31	GIS database built.
November 5	Enterprise GIS created.
November 26	Testing completed.
December 6	Deployment of product.
December 10	Presentation deliverables created.
December 12	Final report completed.

Technical Requirements

Project must adhere to time and budget constraints
Product must be accessible to user with a Windows PC, Android device, and mobile GPS receiver.
Product must be accessible for users with minimal GIS training and no IT background.

Limits and Exclusions

- Timeframe for all development work is limited to approximately three months, from September 5, 2018, to December 14, 2018.
- Rick Jacques will provide timely and accurate responses to requests for necessary information and data that will need to be included in the project.
- Any errors present in the original data supplied by the AVTC (e.g., misalignment of spatial data, typographical mistakes in attribute data entry) is the AVTC's responsibility.
- After the project is complete, the AVTC will be responsible for service fees associated with ongoing data hosting and any future database maintenance required.

Customer Review

Final product to be reviewed by Rick Jacques.

The Spatial Database

Design

A design for a relational spatial database model to be implemented in PostgreSQL was developed by creating an entity relationship diagram. The ERD showed all tables, attributes, relationships and their cardinalities, and schemas, allowing for easy visualization of the product. During this time, the data provided to us by the client was normalized and refined to create a logically structured model.

In summary, the database consists of five schemas:

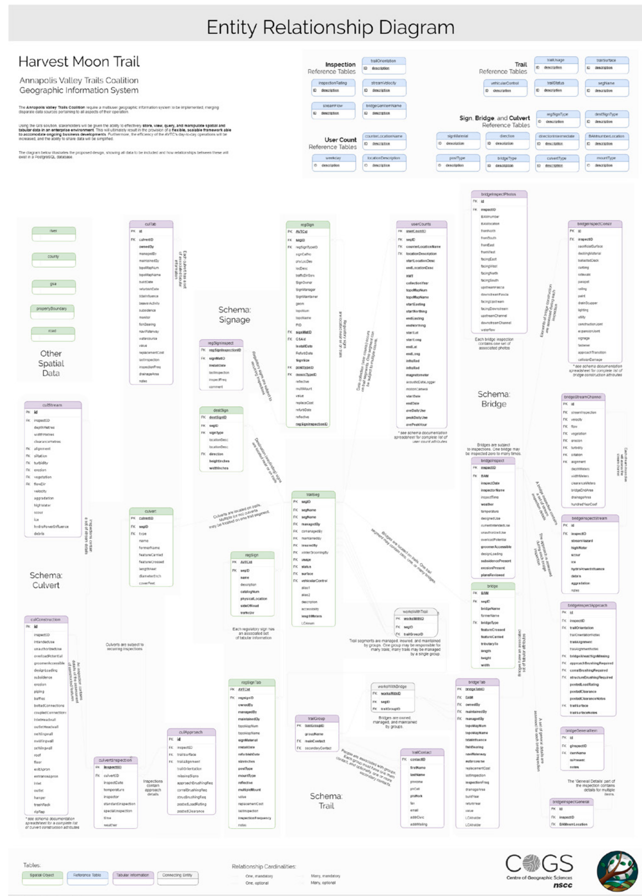
• Trail: Trail segment line features, user count tabular data, and their look-up tables.

• Culvert: Culvert point features and related tabular data, details of inspections, associated look-up tables).

• Bridge: Bridge point features and related tabular data, details of inspections, associated look-up tables.

• Signage: Point features for destination and regulatory signs, tabular data, details of inspections, associated look-up tables

• LUT: Look-up tables containing data (such as water flow direction, inspection 'good/fair/poor' rating, names of the days of the week, etc.) that will need to be referenced by entities in multiple schemas.



The database model was further documented as a shared Excel workbook detailing all the information above, as well as datatypes (integer, varchar, date, etc.) for each individual attribute. The full contents of all required look-up tables were also documented here. This workbook enabled us to have an easy reference ensuring data integrity while collaboratively building the database.

Construction

The database was constructed collaboratively using several tools:

- **PostgreSQL 10**
- **pgAdmin 4 v3.0**
- The **PostGIS** extension for PostgreSQL
- **Python 3** with the **geopandas** module to turn the client's Excel forms into usable data tables.

Three SQL scripts were written for each database entity: one to create the tables, one to populate the tables with attribute data, and one to establish foreign keys and other constraints.

To carry out these required tasks, the following SQL commands were used:

- **Data Definition Language (DDL)** commands: `CREATE, DROP, ALTER`

- **Data Manipulation Language (DML) commands:** `SELECT, INSERT, UPDATE, DELETE`

Regular backups were created of the database during the creation process, and the work underwent several iterations to finally achieve a refined, finished product. Documentation was produced, enabling each step of the build process to be replicated without error.

Queries

Sample queries provided by the client were then able to be executed. The client's business needs can be satisfied by asking questions similar to these examples, and providing answers using SQL queries:

- *"Lands and Forest tells us we must upgrade our stop signs from 12 inches to 16 inches. I need stop sign locations."*

```
select avtcid as “AVTC ID”, description as
“Sign Description”, signsize as “Sign Size
(inches)”, geom
from signage.regsign
join signage.regsigntype on signage.
regsign.regsigntypeid=signage.regsigntype.
id
where signage.regsigntype.description =
‘stop’ and signage.regsign.signsize = 12;
```

- *"Most of our galvanized culverts were installed in the 1950s and we have some extra funding to upgrade. I need the location of all galvanized culverts."*

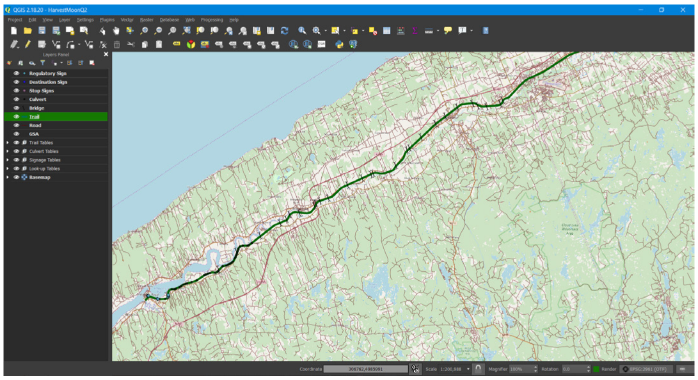
```
select culvert.id as “Culvert ID”
,culvert.name as “Culvert Name”
,culverttype.description as “Culvert
Construction Type”
,tab.buildyear as “Year Built”
,tab.refurbyear as “Year Refurbished”
,segname.description as “Trail Segment
Name”

from culvert.culvert, culvert.culverttype,
culvert.tab, trail.seg, trail.segname

where culvert.culvert.culverttype =
culvert.culverttype.id

and culvert.tab.culvertid = culvert.
culvert.id

and culvert.culverttype.description =
‘Galvanized’

and culvert.tab.refurbyear between 1950 and
1960

and trail.seg.segname = trail.segname.id;
```

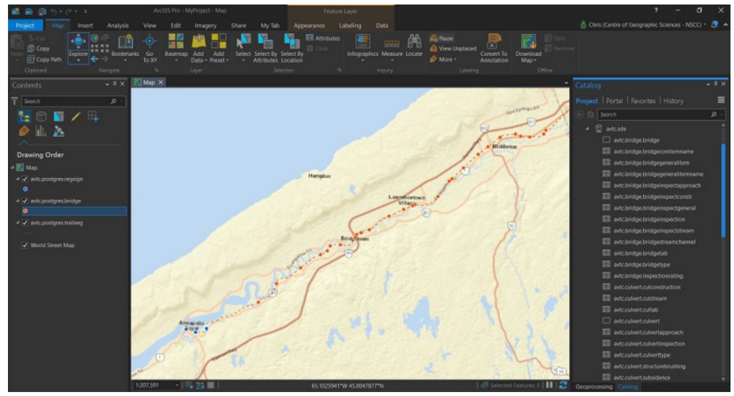
Desktop GIS Integration

A connection to the PostgreSQL database was established within desktop GIS software, enabling us to visualize the spatial data, query results, and perform spatial analysis.

Many desktop GIS applications allow for source data to be loaded directly from the PostgreSQL database, and some, such as QGIS, support direct editing of database geometries and attributes from within the GUI.

Below, AVTC database features are show visualied using QGIS 2.18 and ArcGIS Pro 2.2.2.





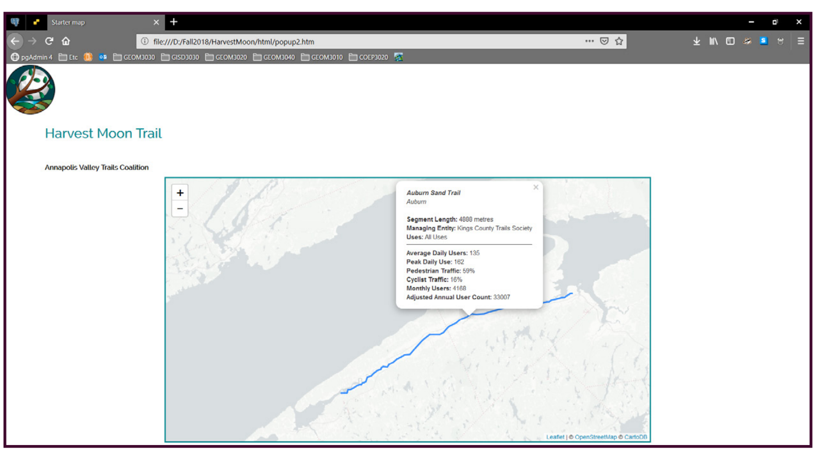
Web-Based GIS

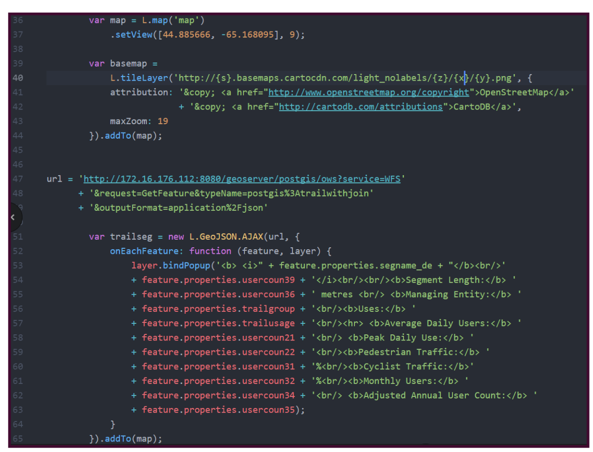
A web-based component is needed for the Annapolis Valley Trails Coalition's stakeholders to view the locations of trail assets and their attributes. A public-facing web map would also be desirable for attracting tourist interest.

Esri, using their ArcGIS Online platform, allow for web maps and feature layers to be created and uploaded from within ArcGIS for Desktop/ArcGIS Pro.

Alternatively, using an open-source GIS server, such as GeoServer, to produce geospatial content layers using data provided by the underlying database. Geoserver can provide data using Web Feature Service (WFS), Web Map Server (WMS), WCS (Web Coverage Service), and Web Processing Service (WPS). These standardized protocols, defined by the Open Geospatial Consortium, allow for serving of a large variety of vector and raster data maps.

The user-end web map application may be built using HTML, CSS, and a Javascript library such as OpenLayers or Leaflet. Javascript and JQuery may be used to provide any additional desired functionality.





Field Mobility

Our client requires the ability to perform field work using a mobile device. They will need to upgrade feature attributes when conducting regular inspections on bridges and culverts. They also need the ability to add new features (for example, when a new trail sign is installed).

We explored a variety of options to provide the AVTC with this functionality, using their pre-existing resources (a Windows PC and an Android tablet). Collector for ArcGIS and QField both showed promise as mobile apps able to be used in this capacity. Appendix 2 contains documentation and workflows describing the entire process using QField, from the initial creation of a project file suitable for use in the field, to the actual collection process, to reintegrating the newly-collected data back into PostGIS.

Options for increasing positional accuracy with the use of external antennas were also explored, with a Trimble R1 being used for some field collection activities and other options, such as the Bad Elf, being researched.

Data Visualization and Static Maps

Two custom maps were created for visualization purposes upon the client's request:

• An updated version of the Leaflet map originally created to explore web mapping technologies, styled to visually express average daily user count by varying the trail segment line width. The map also allows users to query individual trail segments and view user count statistics as pop-ups.

• A static print map displaying each trail segment accompanied by information pertaining to that segment, such as ownership and segment length.

Links to these products are available in Appendix 1.

Appendix 1

Metadata and Supporting Files

Full-size Entity Relationship Diagram (PDF)

Complete Database Documentation (XLSX)

Gantt Chart and Work Breakdown Structure (PDF)

Harvest Moon Trailway Trail Segments Print Map (PDF)

Harvest Moon Trailway Web Map (External Link)

Appendix 2
PostGIS, QGIS Desktop, and QField for Enterprise GIS Field Data Collection

This process has been tested using QGIS 3.4.1 Madeira. This is the first version of this document and represents the efforts of the first group at COGS to use this workflow.

1. Setting up a new QGIS Project for use with QField

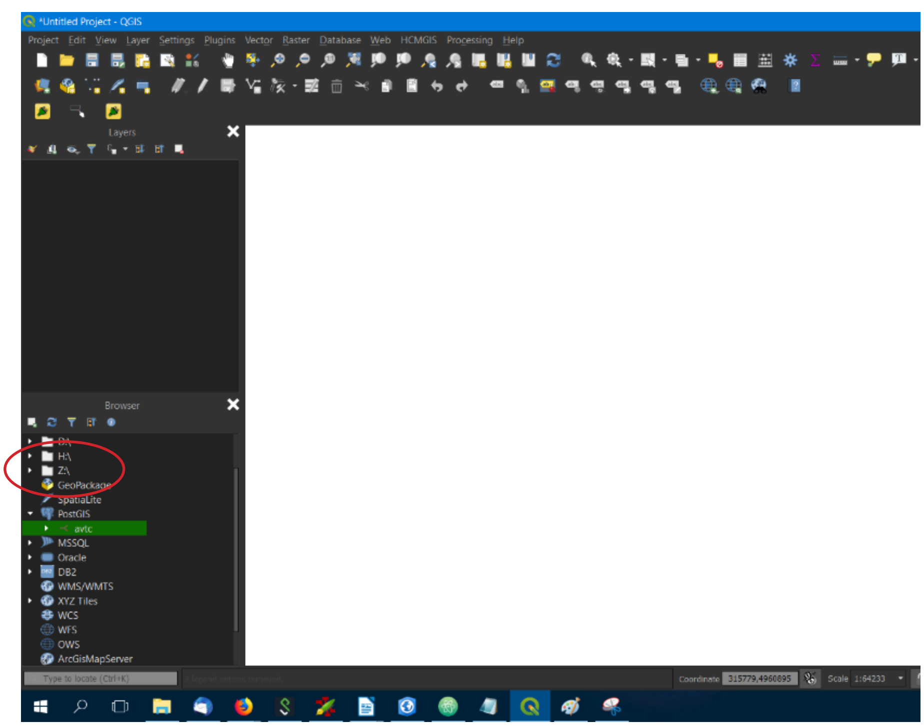
Resources:
- **QGIS official web site:**
https://qgis.org/en/site/

- **QuickMapServices QGIS plugin**
https://plugins.qgis.org/plugins/quick_map_services/

- **QField official web site:**
https://www.qfield.org/

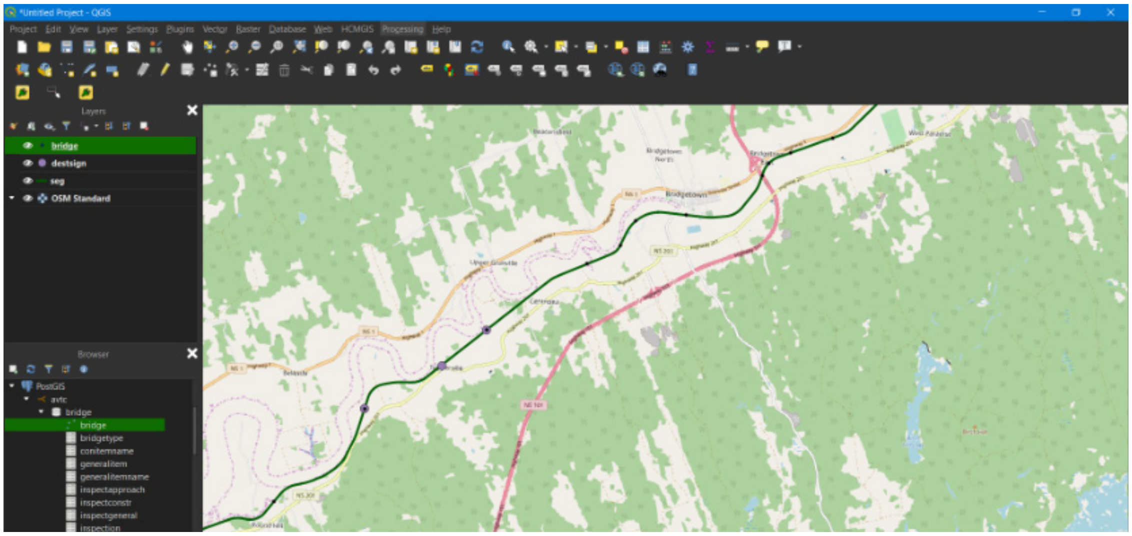
- **QField on Google Play:**
https://play.google.com/store/apps/details?id=ch.opengis.qfield&hl=en_CA

1. Open QGIS and start a new project
2. In the **Browser Panel**, connect to the avtc **PostGIS database**.

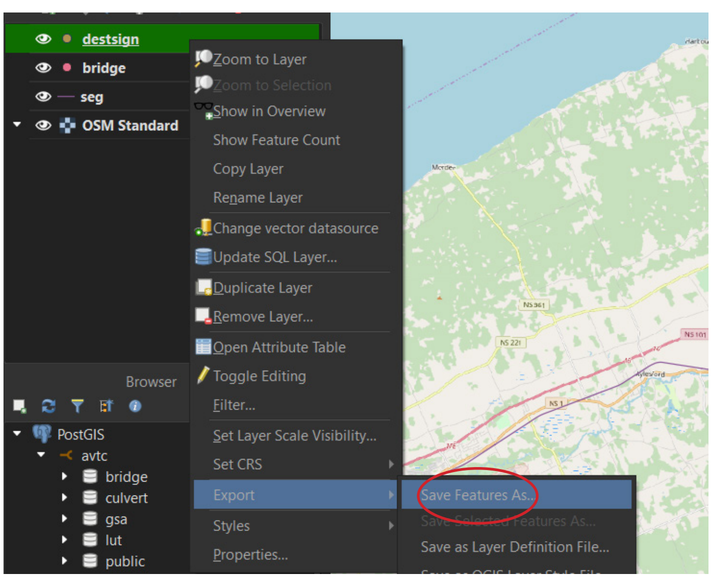


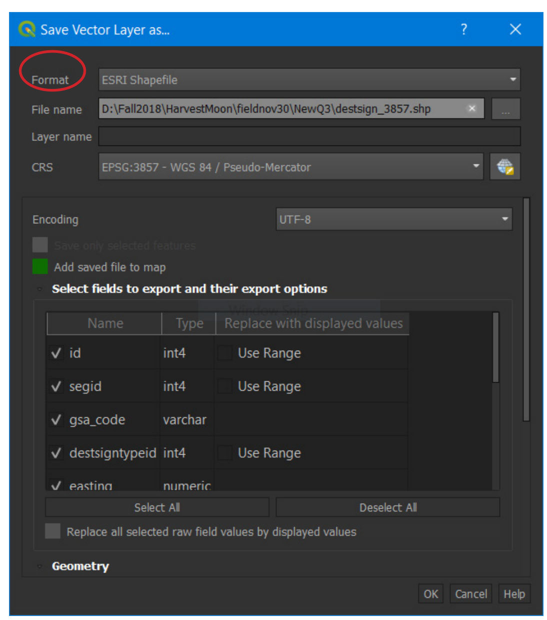
3. Load the appropriate data layers from PostGIS into the map canvas.
   We are collecting Destination Sign data in this example so add:
   - *destsign*
   - *trailseg*
   - *And optionally: bridge, road*

4. Load the OSM Standard basemap using the QuickMapServices plugin.
   - *This will provide location context and make field navigation easier.*



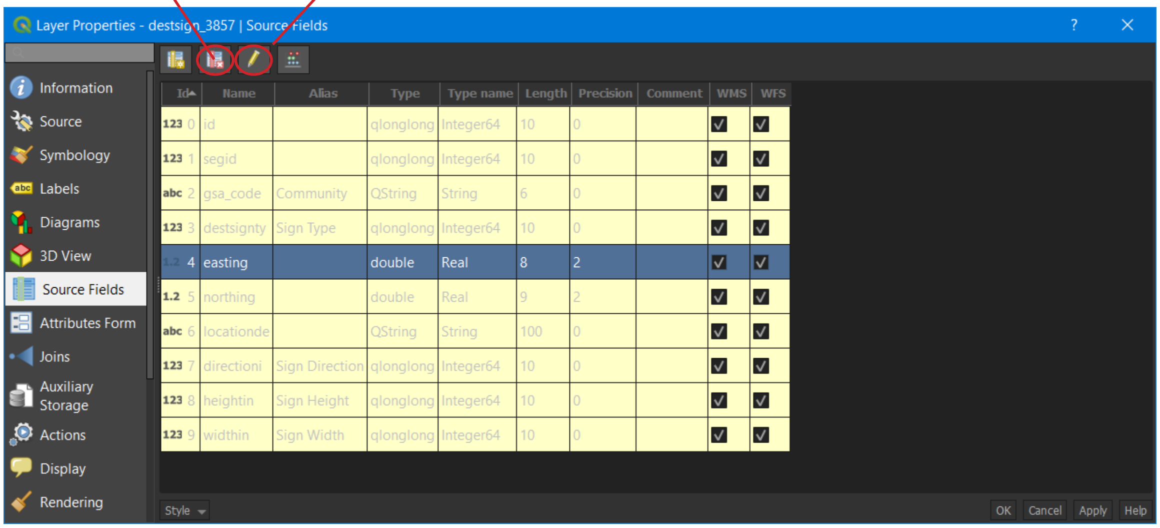
5. **Export and reproject** each layer.
   - *Layers must be in shapefile format to be used on a mobile device.*
   - *As a matter of best practice, we'll reproject all layers to the same CRS as the OSM basemap: WGS 84 / Pseudo-Mercator (EPSG: 3857)*
   - *Save all exported shapefiles to a common working directory on your computer.*





6. Click on the **Source Fields** tab, **enable editing**, and **delete** the following fields:

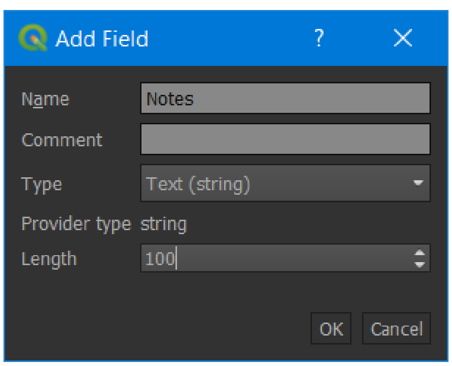
id
segid
easting
northing

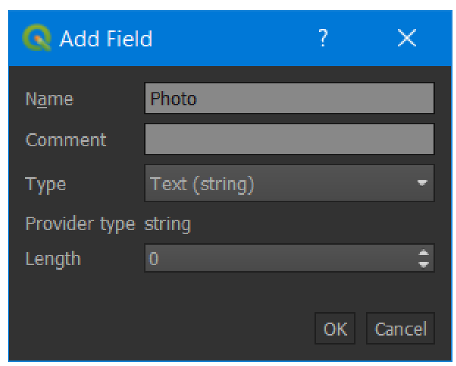


These values can be automatically generated and do not need to be part of field collection.

7. **Add** the following fields;

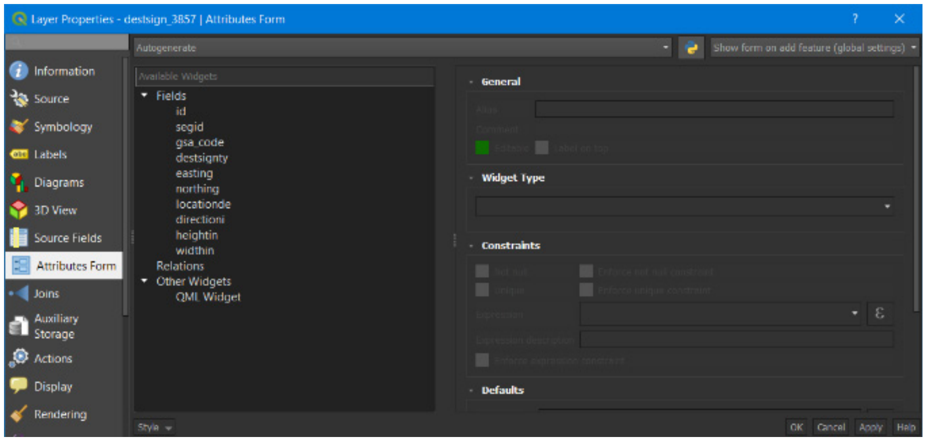
Notes (Type: Text)
Photo (Type: Text)





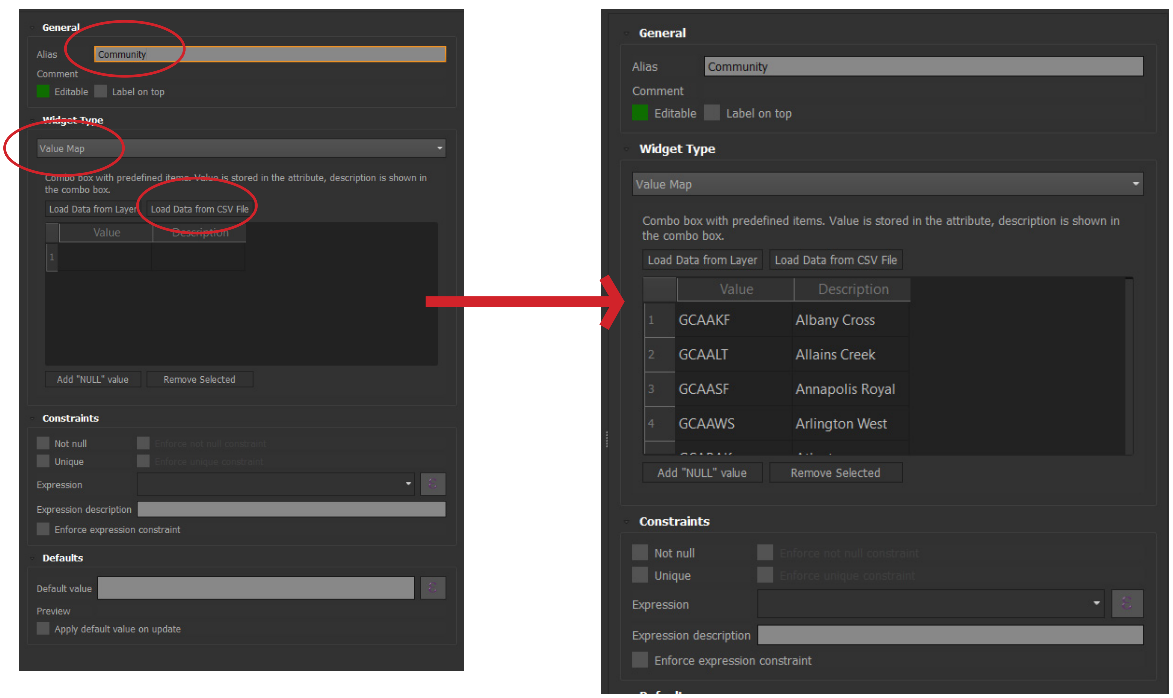
- **Toggle editing off. Be sure to save your edits.**

8. Create **custom collection form widgets** to be used in the field:
- Right-click the reprojected destsign layer (destsign_3857 in this example) and choose **Properties**.
- Click on the **Attributes Form** tab.



- Choose the 'gsa_code' field from the Available Widgets list.
- Add 'Community' as an alias to improve readability in the field.
- Change the widget type to **Value Map**. This allows combo box options to be populated from a prepared LUT.
- Click '**Load Data from CSV File**', navigate to the location of communities.csv and load it.

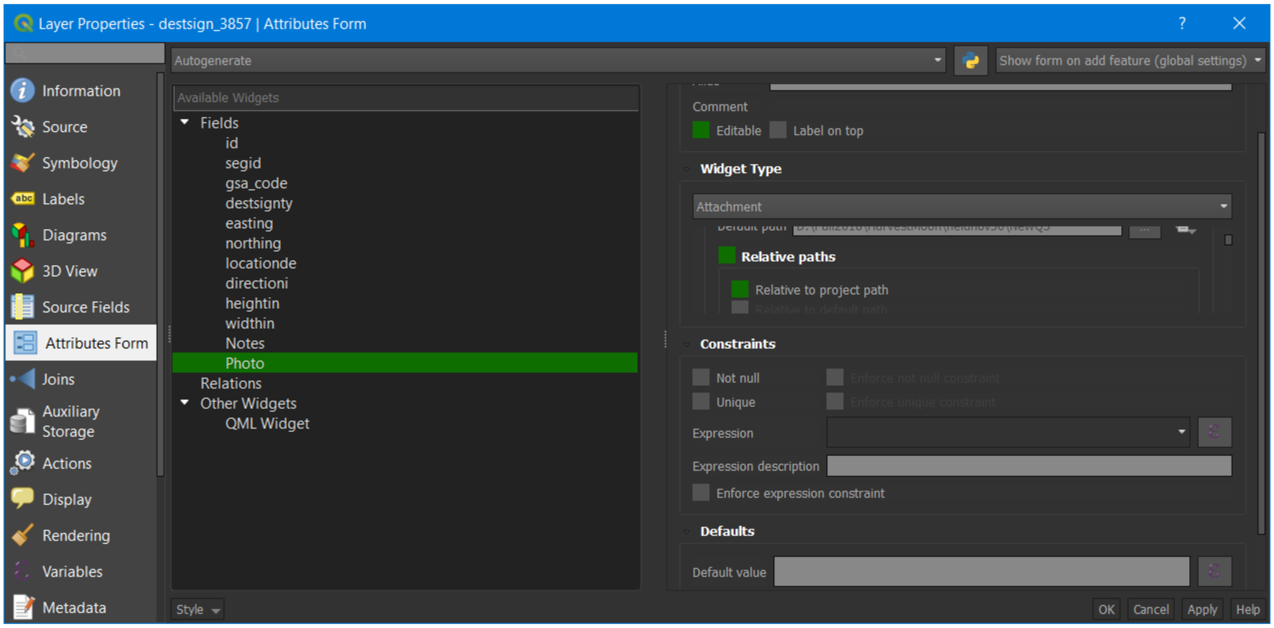
Predefined values matching the foreign key codes from the database with text descriptions should appear.
Tip: Constraints such as Not Null or Unique can be specified for field collection purposes here.



- Repeat Step 8 for the following fields, giving them a user-friendly alias and loading LUT values from the respective csv files:

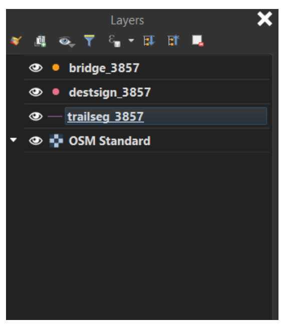
destsigntype
directioni
locationde
heightin
widthin

9. Click on Photo and change its widget type to **Attachment**.



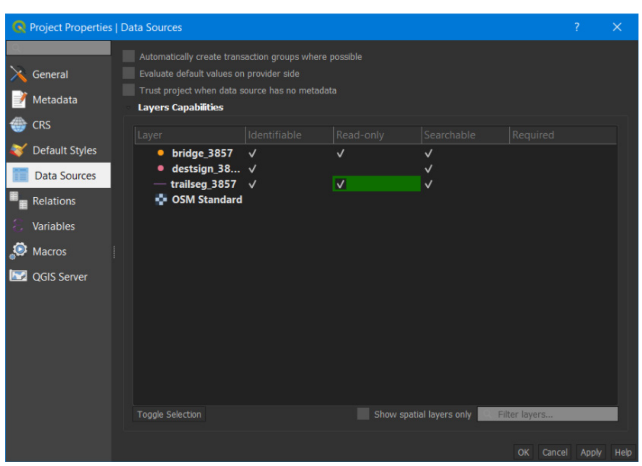
10. Close the Layer Properties dialog and return to the main map view.
- Remove the original PostGIS tables from the Layers Panel (Table of Contents).

  Only layers to be taken into the field should remain.



11. From the main menu, select **Project** > **Properties** > **Data Sources** and configure layer options if desired.

  Recommended: Make all layers except destsign_3857 read-only.

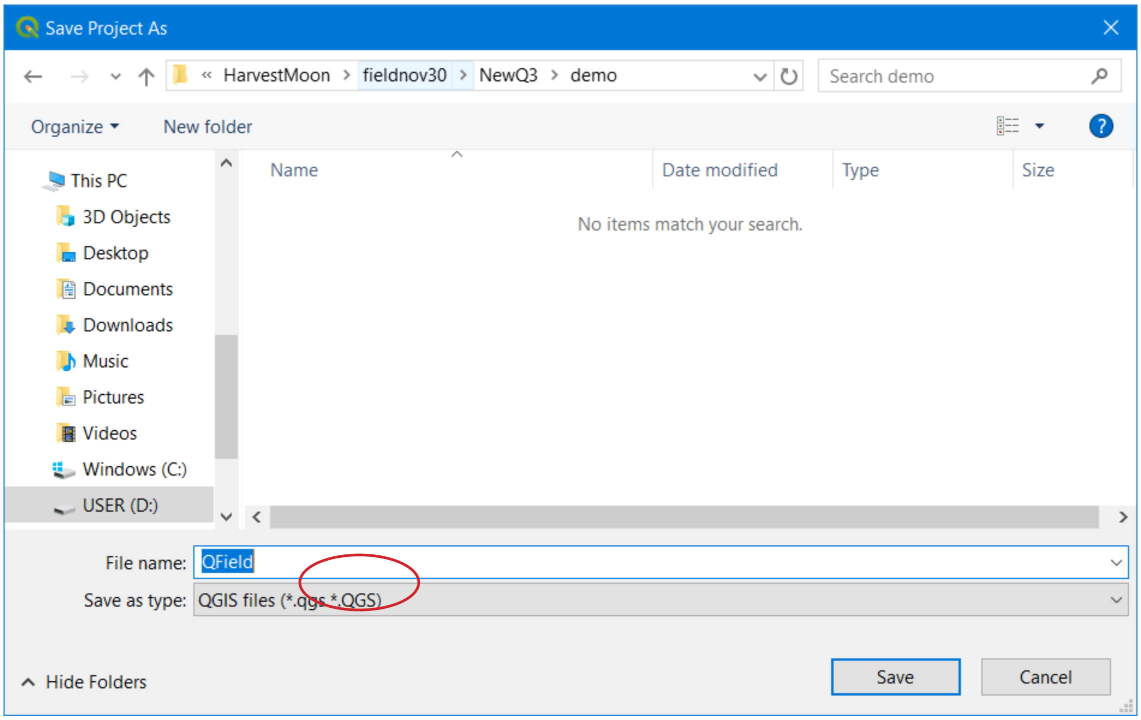


12. Symbolize the project as desired. **How it appears now is how it will appear within QField.**

Recommended: Move point layers to the top of the list in the Layers Panel. Symbolize Destination Signs so that they are prominent.

13. Save the Project. It is now ready to be transferred to a mobile device and used for data collection with QField.

Important: At the time of writing, there was a bug associated with using the .qgz project extension in QField. Save your project as in .qgs format instead.



2. Data Collection Using QField.

Resources

• **QField official web site:**
https://www.qfield.org/

• **QField on Google Play:**
https://play.google.com/store/apps/details?id=ch.opengis.qfield&hl=en_CA

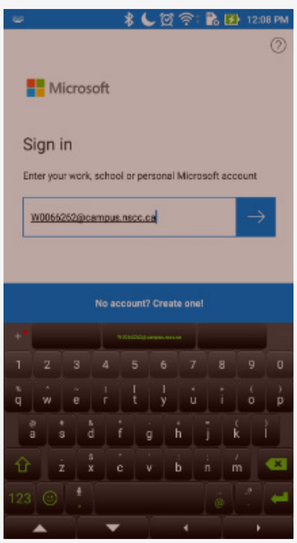
• **Microsoft OneDrive on Google Play:**
https://play.google.com/store/apps/details?id=com.microsoft.skydrive

• **Lefebure NTRIP Client on Google Play:**
https://www.frontierprecision.com/wp-content/uploads/ledstatus.pdf

• **How to Enable Developer Settings in Android:**
https://www.youtube.com/watch?v=v1eRHmMiRJQ

• **Trimble R1 User Guide – LED status:**
https://www.frontierprecision.com/wp-content/uploads/ledstatus.pdf

1. Download the project files from SharePoint and transfer them to your mobile device. If you don't know how to do this, follow the instructions below to transfer files using your campus OneDrive account.



Using Microsoft OneDrive to transfer QField project files.

• Download the OneDrive Android app if you don't already have it.

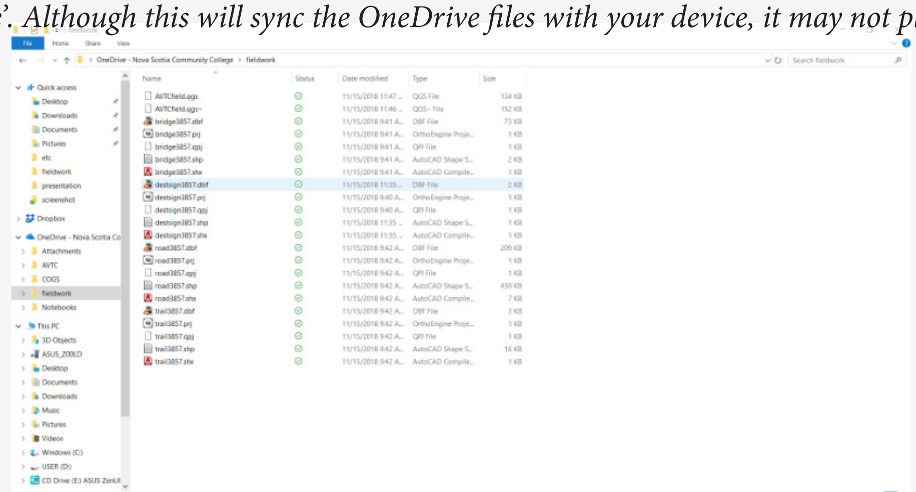
• Launch the app on your phone and log in using your NSCC credentials. You may have to specify your email address with the 'campus' subdomain.

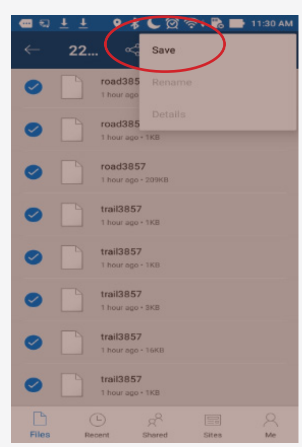
• On your PC, add the files to your OneDrive folder.

• Switching back to your Android, navigate to this same folder using the One-Drive app. The synced files should appear.

• Download the files from OneDrive to the device's local storage.

Note: Make sure you use the ***'Save'*** *option when downloading files from OneDrive.* ***Don't*** *use 'Make Available Offline'. Although this will sync the OneDrive files with your device, it may not play well with QField.*



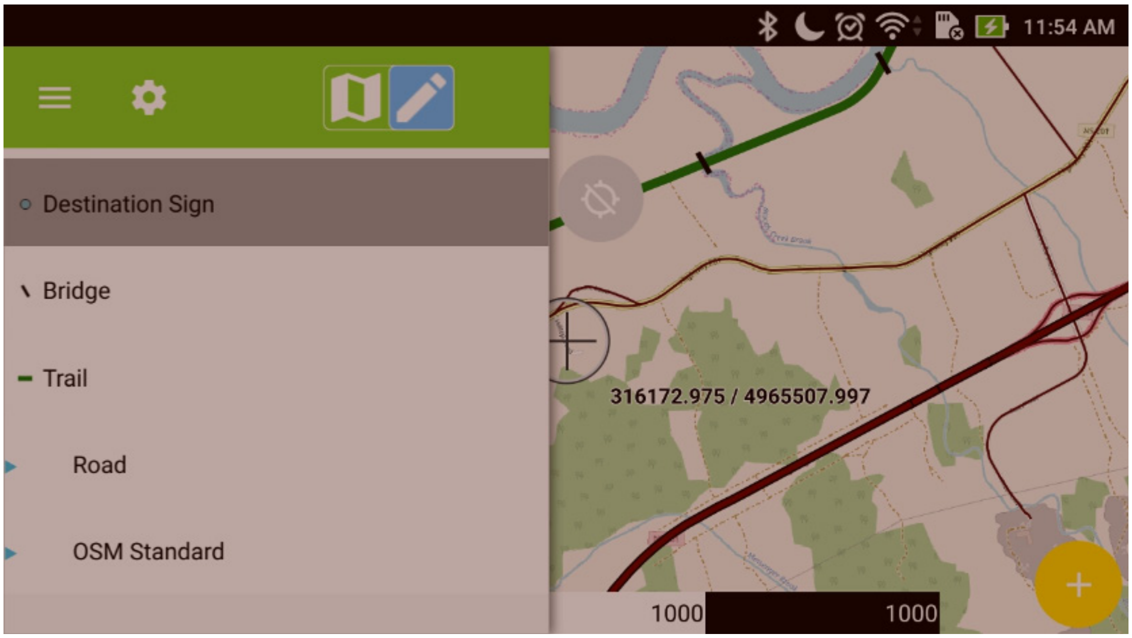


1. Set up the QField Android app.

• Download the QField Android app.

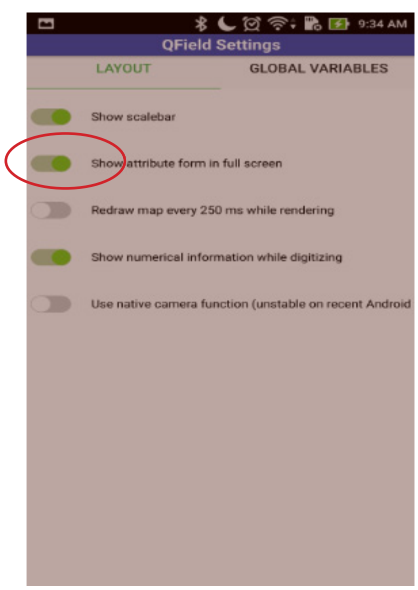
• Launch the QField and open the project file (AVTCfield.qgs) from your phone's internal storage.

• The project should appear as below:



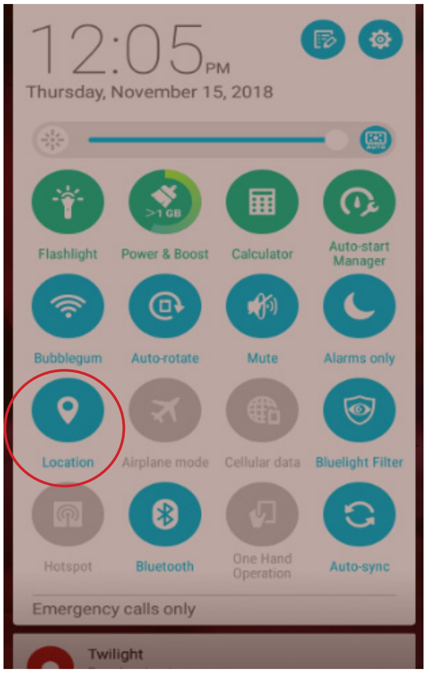
• Go to QField's Settings: click the sandwich button  in the upper left corner, then the cog. 

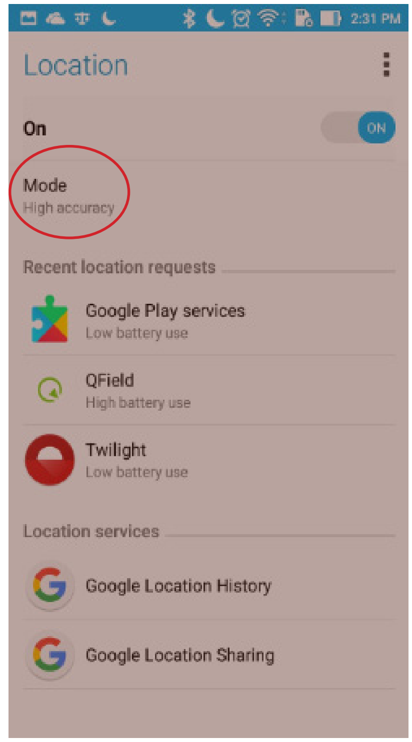
• Select 'Show attribute form in full screen'. This will ensure that the attributes window displays properly on smaller screens.



3. Prepare for Data Collection.

- Turn on **Location** on your device to enable your phone or tablet's GPS functionality.
- If your version of Android has a ***High Accuracy*** setting (available in Location Settings), turn this on.

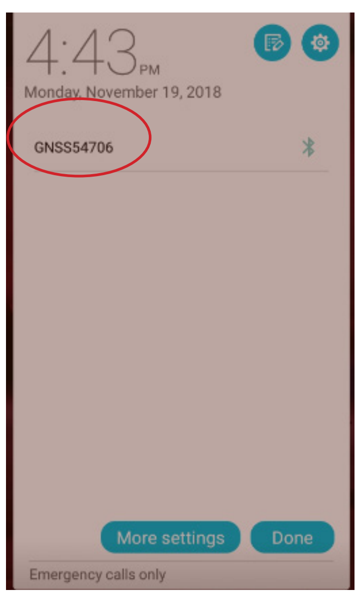




If you're using a Trimble R1, positional accuracy can be increased by connecting it to your device:

- Turn on the R1. The two LEDs on the device indicate its status. Refer to https://www.frontierprecision.com/wp-content/uploads/ledstatus.pdf

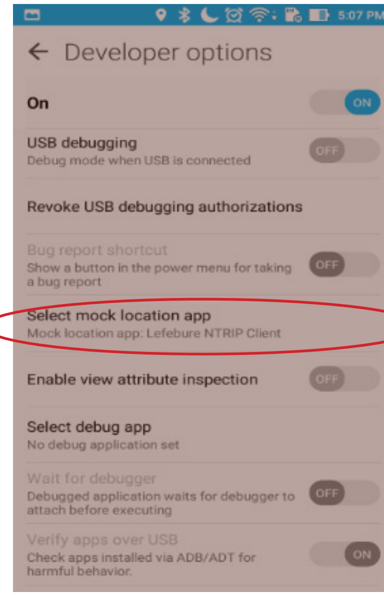
- Enable Bluetooth on your Android and connect to the R1.



• Download the free trial of the Lefebure NTRIP Client Android app. This third-party tool allows your phone to connect to the R1 via Bluetooth.

• Enable Mock Locations on your Android:

• Developer Settings must be enabled. If you don't know how to do this, follow the instructions here: https://www.youtube.com/watch?v=v1eRHmMiRJQ



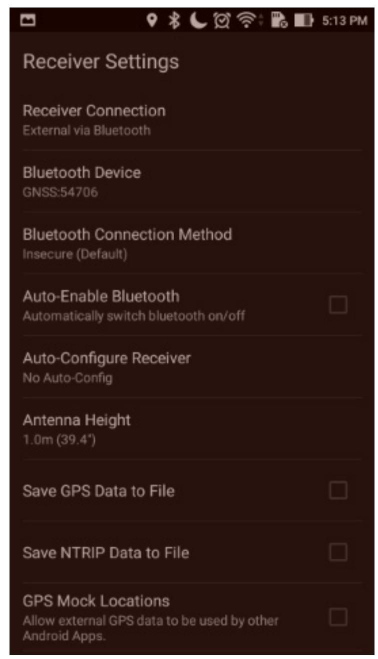
Launch the NTRIP app. Click the Settings button, then choose Receiver Settings.

• Make the following changes:

Receiver Connection: External via Bluetooth

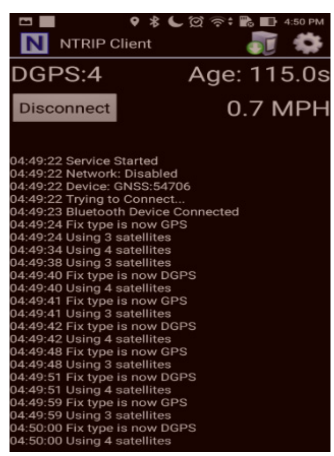
Bluetooth Device: [select the name of your R1]

Antenna Height: 1 metre



• Click Connect. After a few seconds, the main screen should display a feed showing connectivity status.

Tip: If a successful connection is not established, try:
• *Turning your phone's Location feature on and off again.*
• *In NTRIP Client's Receiver Settings, changing 'Auto-Configure Receiver' to 'Custom'.*



4. Fieldwork Process

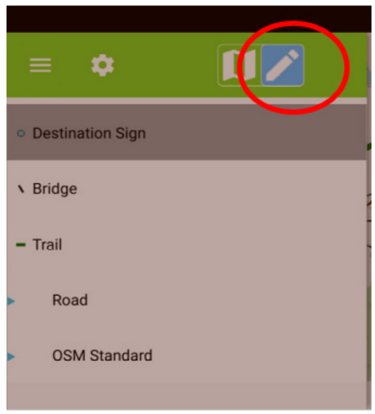
Once you're out in the field and ready to collect, launch Qfield and do the following:

- **Enable Positioning** by clicking the Position button. Blue means it's enabled.

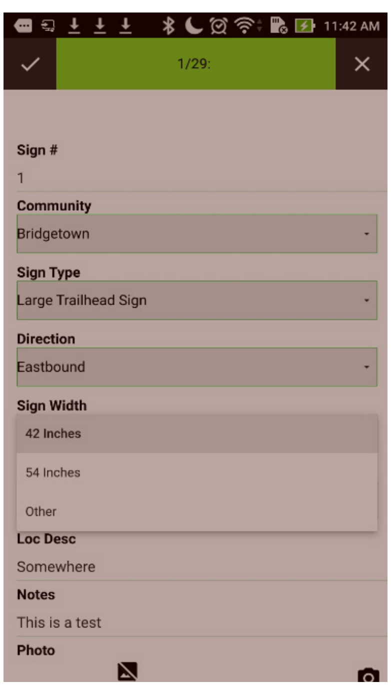
- **Position your phone as close as possible to the feature to be collected. Make sure your map location (the red dot) has updated to reflect this.**

- Click the sandwich icon in the top-left corner and select the Destination Sign layer.

- Switch from Browse mode to **Edit mode** by clicking the blue pencil button above the layer list.



- Pan the map until the crosshair is at the point where you want the feature to be created
- Click the **yellow Plus symbol** in the lower right.



- After you click the Plus symbol, the attribute form will pop up. Fill in the attributes as accurately as you can. Try using the Photo section to take a photo of the sign with your phone's camera.

Tip: We have noticed a bug whereby QField sometimes crashes when the user clicks on a feature to identify/edit it while the map is relatively zoomed-out. A work-around for this seems to be to zoom in closer to the feature before clicking on it.

Tip: The Notes field is a temporary field and its contents will not be stored in the final database tables.

4. Transfer of Field Data Back to Desktop

After you've finished collecting, the project files can be transferred back to your PC.

If you've used the OneDrive method above, here is a workflow you could use:

1. Open the OneDrive app on your device and create a new folder called 'Field Data'.
2. Enter this folder and tap the Plus button to add new files.
3. Choose 'Upload'.
4. Navigate to where the project files have been saved on your Android and select all files.
5. Once back in the office, open the synced OneDrive folder on your PC to access the files.

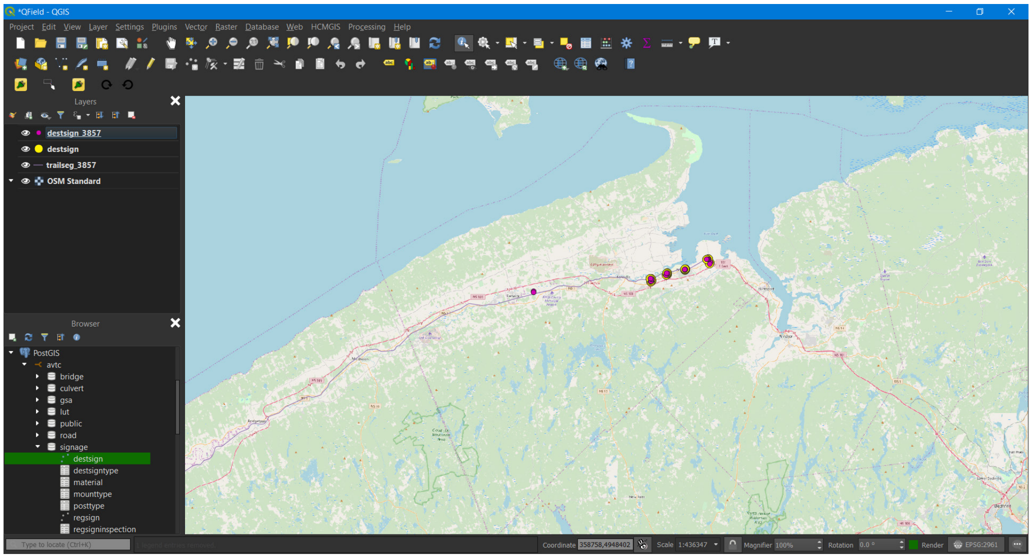
The .qgs project file and its associated shapefiles may be opened in QGIS for Desktop. Any further required processing, including appending newly collected data back into the PostGIS environment, can be performed on your PC.

4. Getting Data Back Into PostGIS

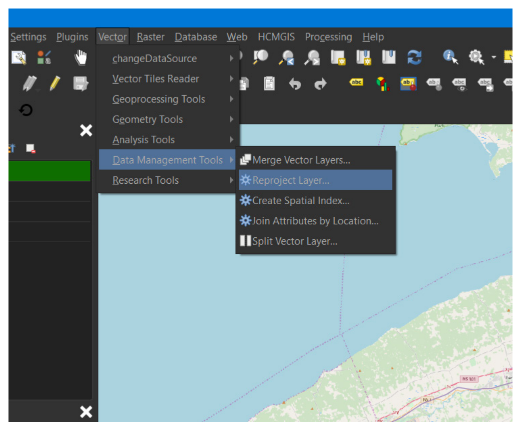
Resources:

Append Features to Layer QGIS Plugin: https://plugins.qgis.org/plugins/AppendFeaturesToLayer/

1. **Import** the shapefile (destsign_3857) containing the newly collected data into a QGIS Desktop project.
2. Add the destsign layer from the **PostGIS** connection.



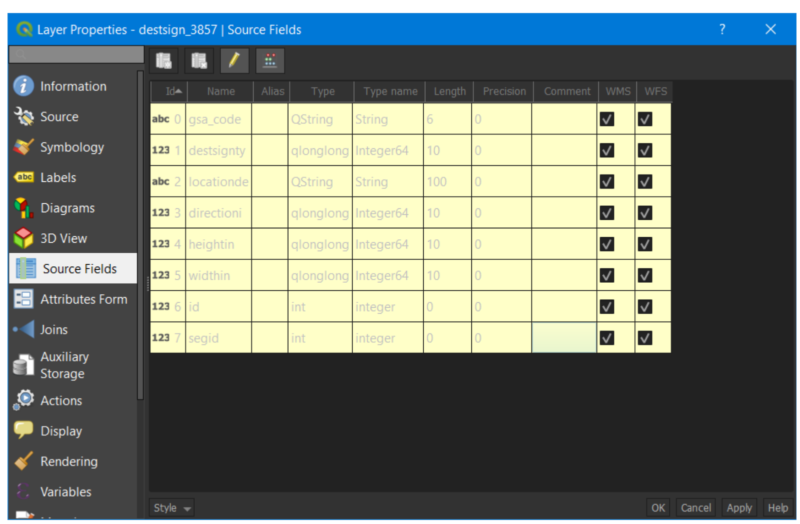
3. **Reproject** destsign_3857 to NAD83 CSRS Z20 (EPSN: 2961). Use **Vector > Data Management Tools > Reproject Layer** or Export Layer to do this. Take a look at the collected points to make sure they have imported correctly and there are no alignment issues.



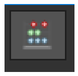
4. Open the Properties of the reprojected destination sign layer. Go to the **Source Field** tab, turn on editing, and delete the following fields:

- Photo
- Notes

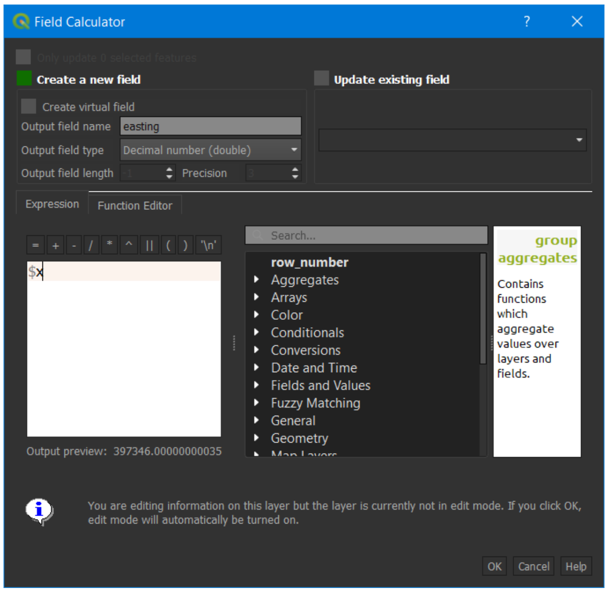
Notes is intended for temporary information and photo storage hasn't yet been implemented in the database.

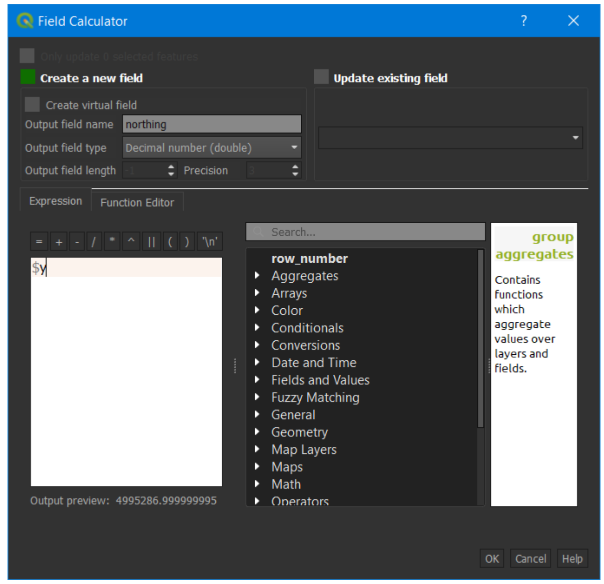


5. **Add** the following fields:
   - id (integer)
   - segid (integer)

6. Use the **Field Calculator**  to automatically populate easting and northing values.
Specify the following parameters:

Create New Field
Output field name: **easting**
Output field type: **Deciminal Number (double)**
Expression: **$x for easting, $y for northing**





7. No automated method for this yet: Open the attribute table of the reprojected layer and manually update the **segid** and **id** fields for the new records only to the correct values. The ID will be serial integer values starting from one after the last record in destsign. segid can be retrieved by identifying the nearest trail segment feature.

Turn off editing mode and **save all edits.**

8. **Select only the newly added/updated features** in the reprojected Destination Sign layer's Attribute Table.

9. Use the **Append Features to Layer** plugin (check '***Selected features only***') to add the new features back into the original PostGIS destsign table.