**NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**Faculty of Engineering**

Department of Ocean Operations and Civil Engineering

# Guideline for Using Visualization Platform

This guideline is a supporting document of the first author's Ph.D. Thesis:

## GIS-based Approach to Flood Modeling and Condition Assessment of Sewer Network

Lam Van Nguyen

Razak Seidu

**Water and Environmental Engineering Group**

**NTNU Ålesund**

**12/2022**

## Objectives of the guideline

The overall aim of this guideline is to provide a user with a step-by-step implementation for using a visualization platform of sewer conditions on mobile phones and HoloLens devices. This platform is only used for a visualization based on the specific objects relating to the first author's Ph.D. thesis.

# Contents

# 1. Mobile Application

## *1.1. Downloading and Installing Applications*

There are two versions of the mobile application, named "*WaterNet on GoogleMap*" and "*WaterNet*". The downloaded links of these applications are provided in **Table 1.1**. The basic difference between the two versions is described as follows:

- The "*WaterNet on GoogleMap*" application uses the Google map as a background image and the components of the sewer network (i.e., manholes or pipes) are overlayed on this image. This 2D visualization application was designed for supporting the users to easily look for components of the sewer network on the field.
- The "*WaterNet*" application supports 2D and 3D visualizations of sewer components and presents predicted results of the sewer condition.

**Table 1.1.** Downloaded links to the applications on android devices

| Name | Download link |
|---|---|
| WaterNet_Google Map | https://www.mediafire.com/file/lw2gote7dwjmgl3/WaterNetOnGoogleMap.apk/file |
| WaterNet | https://www.mediafire.com/file/ua4qzwb2582bb6b/WaterNet.apk/file |
| Data sample | https://www.mediafire.com/file/okq6nn95frc41ic/Data_Sample.zip/file |
| Python code | https://www.mediafire.com/file/piwvybtyc2uy7b4/Unity_Python.py/file |

These applications are distributed as an Android package with the file extension **APK**. Therefore, these applications are only compatible with android devices. After downloading applications from the above links, the users easily install these applications as any android application by clicking on them.

The interfaces of these applications are currently optimized for android phones with a screen resolution of 720x1600 pixels. Other android devices with different screen sizes maybe have slightly different experiences.

## *1.2. The "WaterNet on GoogleMap" Application*

After installing the "*WaterNet on GoogleMap*" application on the android device, the symbol  will appear on the menu of the device (**Figure 1.1**a).

To get the geographical coordinates of the user's position, the GPS function on the device should be enabled before the application is opened. After the application is activated, the current

position of the user is shown in the center of the screen (the blue dot) and the geographical coordinates (including latitude, longitude, and altitude) of the user also are shown on the screen (**Figure 1.1**b).

When the user moves, the user's location and geographical coordinates are updated in real-time. The user also moves the screen from any position to the current position by clicking on the symbol  in the top-right corner of the screen (**Figure 1.1**b).

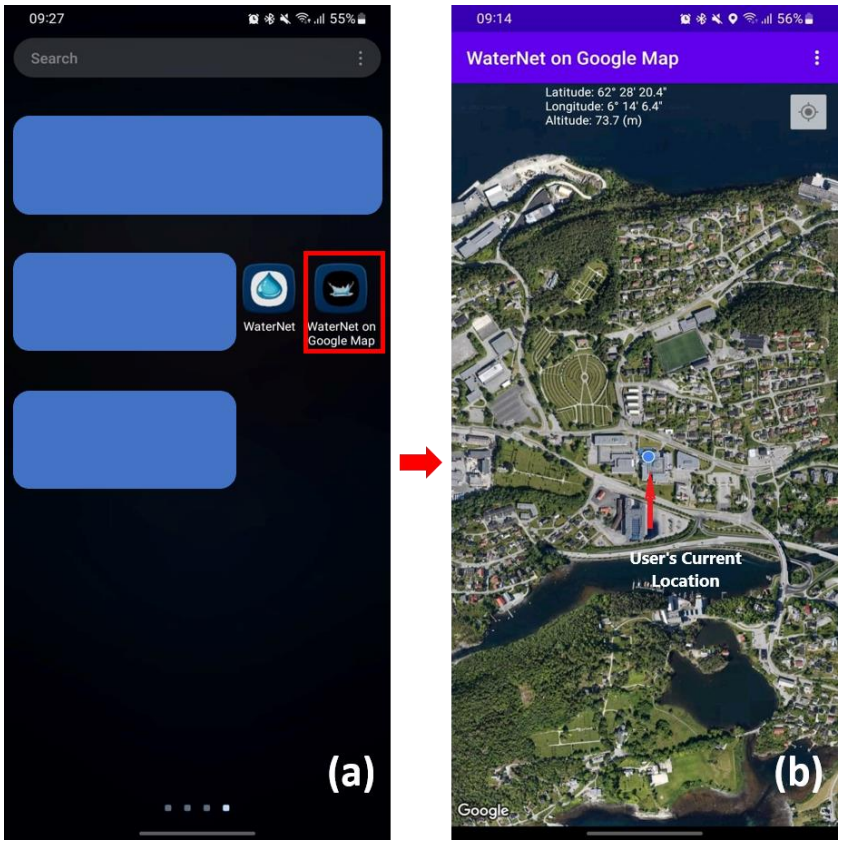



**Figure 1.1.** The interface of the *"WaterNet on GoogleMap"* application

This application allows the user to import some components of the sewer network (such as manholes and pipes) from CSV format files. The structure of CSV files is shown in **Figure 1.2**.

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | PSID | Latitude | Longitude | Z | BOTTOM_Z | DIAMETER |
| 2 | 3822 | 62.46013019 | 6.26396 | 26.85 | 25.34 | 1.2 |
| 3 | 3823 | 62.46009518 | 6.26495 | 27.14 | 24.67 | 1.2 |
| 4 | 3824 | 62.46019717 | 6.26575 | 27.9 | 25.18 | 1.2 |
| 5 | 58881 | 62.45994931 | 6.263119 | 25.89 | 24.12 | 1.2 |
| 6 | 58882 | 62.45996656 | 6.263101 | 25.89 | 24.1 | 1 |
| 7 | 58883 | 62.45997333 | 6.263135 | 25.97 | 23.26 | 1.2 |
| 8 | 58884 | 62.46014482 | 6.263969 | 26.84 | 24.9 | 1 |
| 9 | 58886 | 62.46014821 | 6.264929 | 27.27 | 24.89 | 1 |
| 10 | 58887 | 62.46008889 | 6.264968 | 27.12 | 25.3 | 1.2 |
| 11 | 58888 | 62.46019299 | 6.26577 | 28.08 | 26.27 | 1 |
| 12 | 58907 | 62.46035042 | 6.263297 | 26.76 | 25.07 | 0.65 |
| 13 | 59019 | 62.46022278 | 6.262976 | 26.01 | 24.66 | 1 |
| 14 | 59021 | 62.45959161 | 6.266194 | 31.81 | 29.7 | 1 |
| 15 | 112182 | 62.46052082 | 6.263792 | 27.6 | 26.05 | 1.2 |
| 16 | 112760 | 62.46020249 | 6.264139 | 26.9 | 25.56 | 1 |
| 17 | 112765 | 62.46011305 | 6.264231 | 26.2 | 24.94 | 0.9 |

(a) Sewer's manhole data format

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | PipeID | Latitude | Longitude | Diameter | Z_Value |
| 2 | 5372 | 62.4601 | 6.26495 | 0.6 | 24.9 |
| 3 | 5372 | 62.46012 | 6.264264 | 0.6 | 24.96 |
| 4 | 5372 | 62.46013 | 6.264077 | 0.6 | 25.14 |
| 5 | 5372 | 62.46013 | 6.26396 | 0.6 | 25.31 |
| 6 | 5373 | 62.4602 | 6.26575 | 0.6 | 25.26 |
| 7 | 5373 | 62.4601 | 6.26495 | 0.6 | 24.9 |
| 8 | 59098 | 62.46011 | 6.264021 | 0.315 | 25.15 |
| 9 | 59098 | 62.45995 | 6.263119 | 0.315 | 24.11 |
| 10 | 59099 | 62.45995 | 6.263119 | 0.315 | 24.11 |
| 11 | 59099 | 62.45991 | 6.262961 | 0.315 | 23.99 |
| 12 | 59099 | 62.45977 | 6.262401 | 0.315 | 24.42 |
| 13 | 59100 | 62.46014 | 6.263969 | 0.4 | 24.9 |
| 14 | 59100 | 62.45997 | 6.263101 | 0.4 | 24.07 |
| 15 | 59101 | 62.46022 | 6.262976 | 0.16 | 24.66 |
| 16 | 59101 | 62.46015 | 6.262997 | 0.16 | 24.53 |
| 17 | 59101 | 62.46 | 6.26307 | 0.16 | 23.83 |

(b) Sewer's pipe data format

**Figure 1.2.** Example of the data format used in the "*WaterNet on GoogleMap*" application

**Figure 1.3** illustrates the steps for importing manholes and pipes from CSV files into the application. By coloring different characteristics (i.e., material, network type, etc.,) of each pipe, the user easily distinguishes sewer pipes in the field using this application.

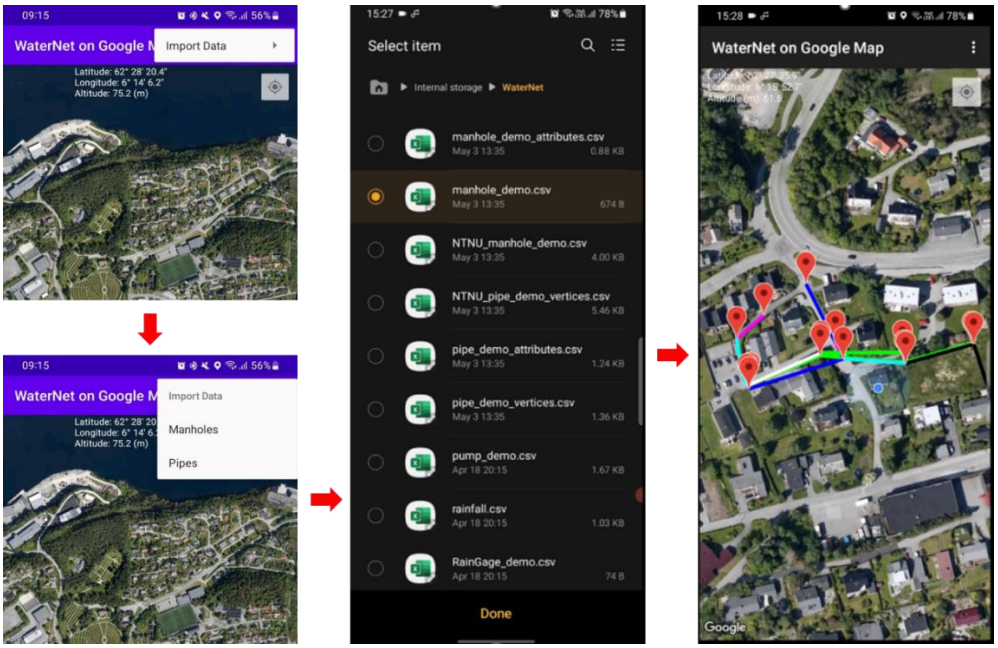



**Figure 1.3.** Importing sewer components into the application

**Figure 1.4** shows the process of random checking of the "*WaterNet on GoogleMap*" application. The experimental results show that the difference between the actual sewer's manhole location and their visualized location on the "*WaterNet on GoogleMap*" application is smaller than 3 m. This difference satisfies for identifying sewer manholes in the field.

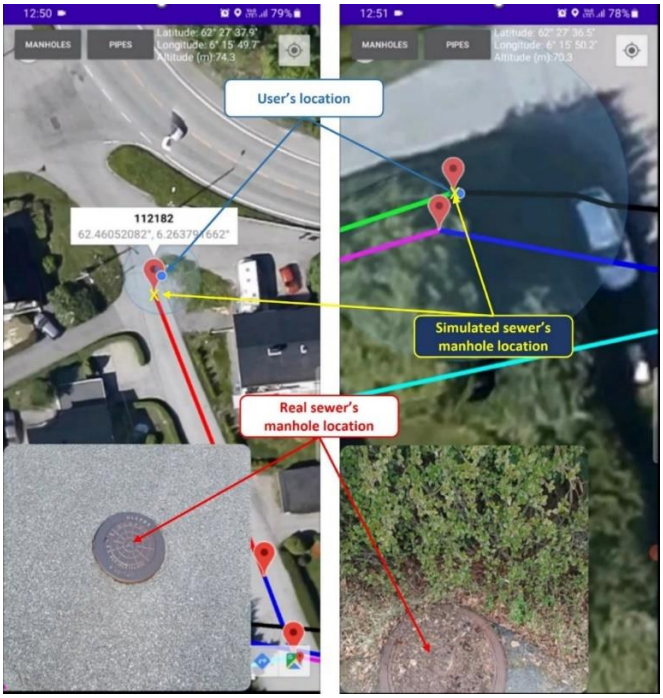



**Figure 1.4.** Accuracy of the "*WaterNet on GoogleMap*" application

It is worth noting that other point-based objects (such as locations of pumps, outfalls, dividers, storage units, etc.,) can be imported using the "*Manholes*" function in this application.

### *1.3. The "WaterNet" Application*

The "*WaterNet*" application was designed for 3D visualization, integration of 3D models, and results predicted from machine learning models and real-time visualization. The interface of the "*WaterNet*" application is shown in **Figure 1.5**.

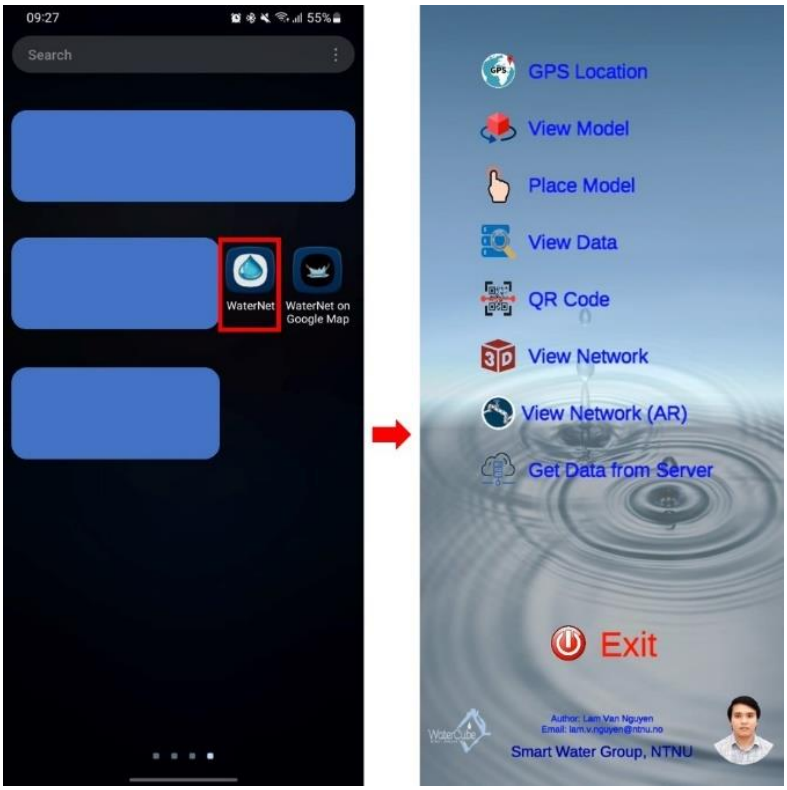



**Figure 1.5.** The interface of the "*WaterNet*" application

The "*WaterNet*" application provides the user with some options to visualize data that are presented as follows:

#### *1.3.1. GPS Location*

This function allows the user to locate the device's location in the World Geodetic System 1984 (WGS84) coordinate reference from Global Positioning System (GPS) signal. This function is useful to pinpoint the problems in the field (i.e., cracks' locations, notice points, etc.,) and the output can be saved in the Comma-Separated Values (CSV) type that is easily opened by a text editor such as Notepad or Microsoft Excel. The user can use this function to store notes in the field in CSV format on their device, they after that can transfer data from the android device to a personal computer (PC) and process them for different purposes. The basic steps for implementing this function are shown in **Figure 1.6**.

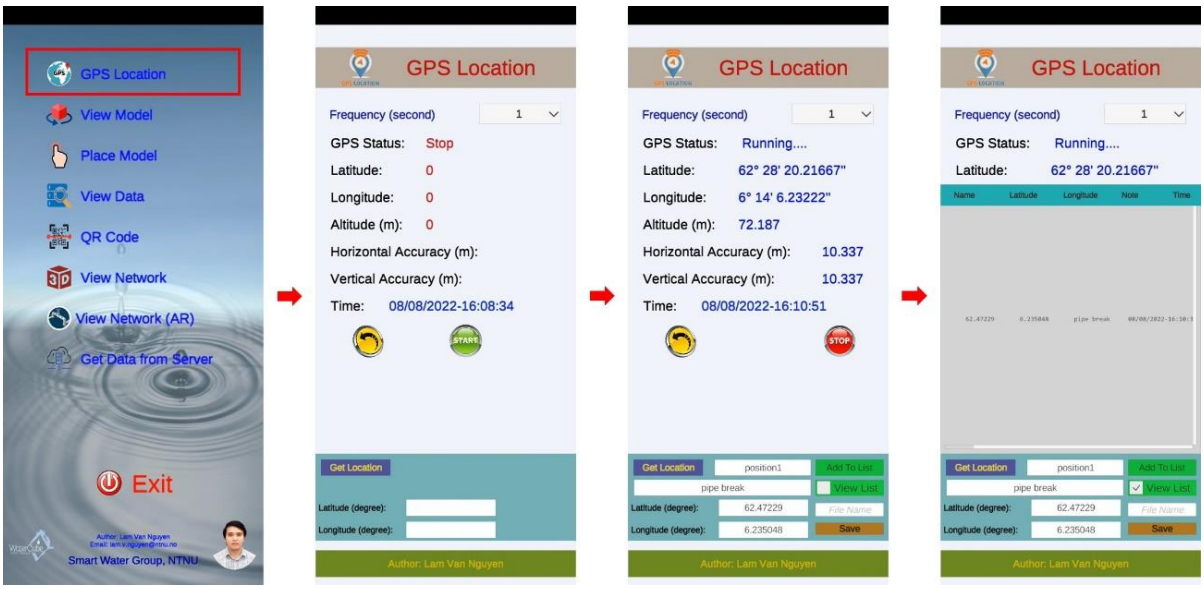



**Figure 1.6.** Receiving GPS signal in the "*WaterNet*" application

The detailed steps for implementing this function are presented as follows:

- *Step 1:* Activate this function by selecting the symbol 

on the device screen.
- *Step 2:* Define the frequency (in seconds) that the user wants to collect the GPS location.
- *Step 3:* Select the button 

to begin receiving GPS signal for coordinate definition.
- *Step 4:* Click on the button "*Get Location*" at the wanted positions to get latitude and longitude. The user can set up names and notes for pinpointed locations (**Figure 1.7**).

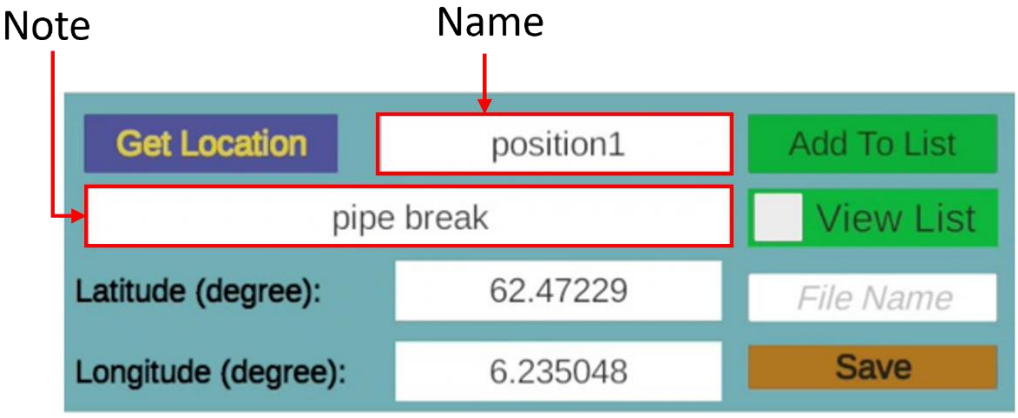



**Figure 1.7.** Pinpointing in the "*WaterNet*" application

- *Step 5:* Select the button "*Add To List*" to add pinpointed location to the database. The user also views the points that are saved in the list by clicking on the checkbox "*View*

*List*". The list of the point is directly shown on the device screen (**Figure 1.8**). The user can use the finger to move and see all information of the saved points. Based on this list, the user can check the information of the saved points or take more measurements on the field.

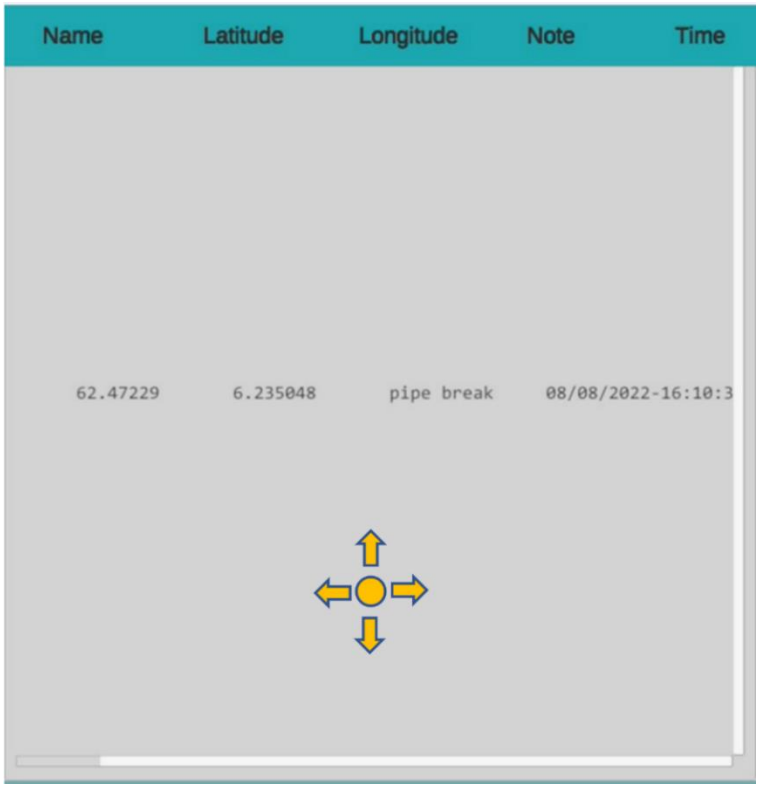



**Figure 1.8.** List of saved points in the "*WaterNet*" application

- *Step 6:* After the user finishes measuring in the field, the data file must be named in the area 

(**Figure 1.7**). To save data in the storage of the device, the user must click on the button "*Save*" in **Figure 1.7**.

- *Step 7*: To finish this function, the user clicks on the button 

. Selecting the button  will quit this function and the user will be delivered to the main screen (**Figure 1.5**).

The saved data from **Figure 1.8** is transferred to a PC, an example of the structure of this data is shown in **Figure 1.9**.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Name | Latitude | Longitude | Note | Time |
| 2 | position1 | 62.47238 | 6.234922 | pipe break | 06/12/2022-09:33:38 |
| 3 | position2 | 62.47252 | 6.234916 | landslide | 06/12/2022-09:34:27 |

**Figure 1.9.** An example of a record created by the "*WaterNet*" application

### *1.3.2. View Model*

This function provides an option for the user to view a 3D model from assigned images. The steps for implementing this function are shown in **Figure 1.10**. In this function, by using different assigned images, the user can view the different corresponding 3D models.

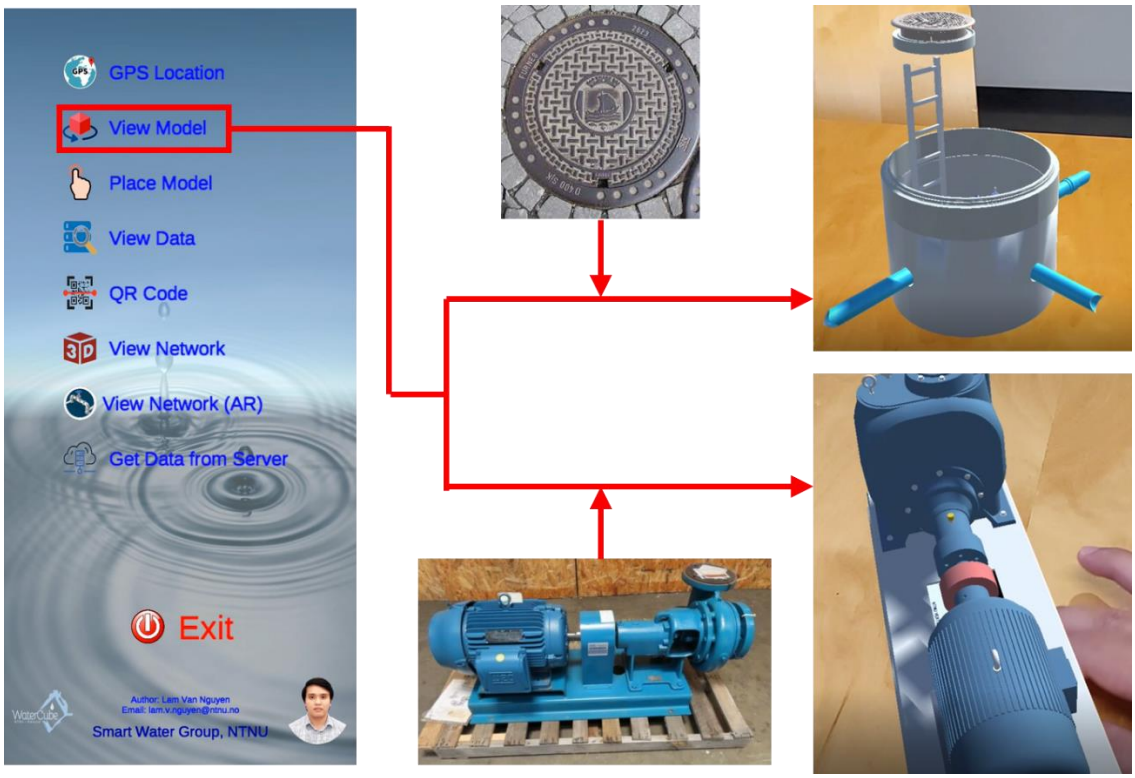



**Figure 1.10.** Illustration of viewing the 3D model in the "*WaterNet*" application

### *1.3.3. Place Model*

This function provides an option to view 3D objects on a real scale using an Augmented Reality perspective (**Figure 1.11**).

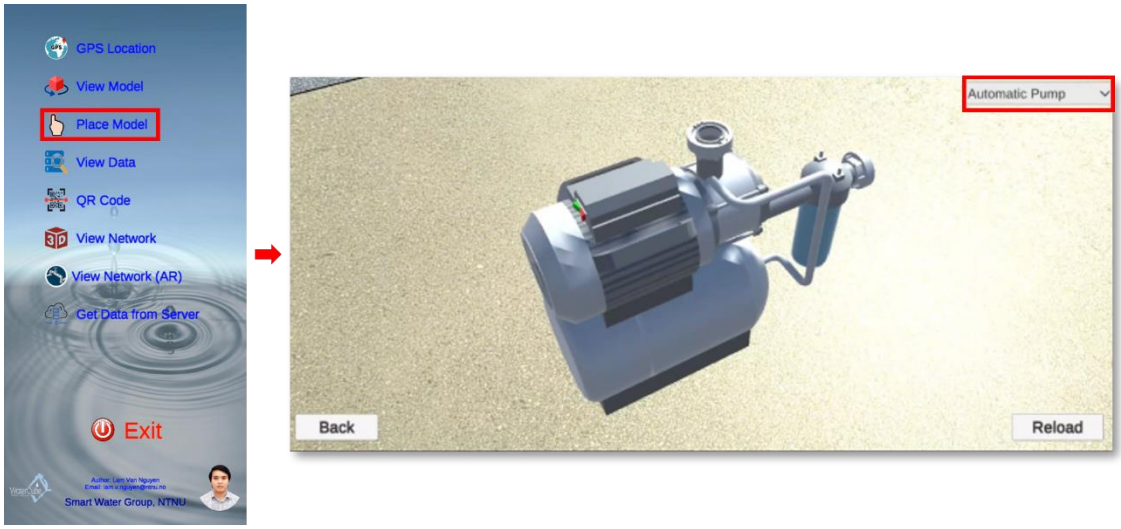



**Figure 1.11.** Example of placing the 3D model in the "*WaterNet*" application

The user can view different objects by using the combo box in the top-right corner of the screen. This function provides two different 3D models to view: the pump (**Figure 1.12**a) and the fire hydrant (**Figure 1.12**b). It is worth noting that other 3D models can be further added depending on the specific purposes.

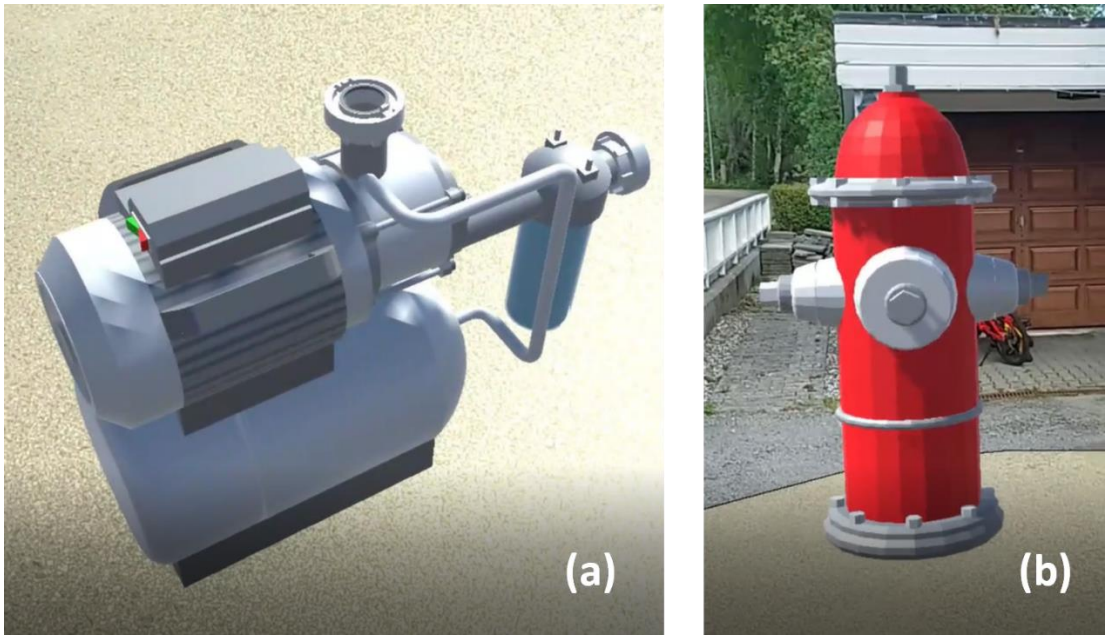



**Figure 1.12.** Example of a 3D model in the "*WaterNet*" application

### *1.3.4. View Data*

This function allows the user to directly view the CSV file using the "*WaterNet*" application (**Figure 1.13**).

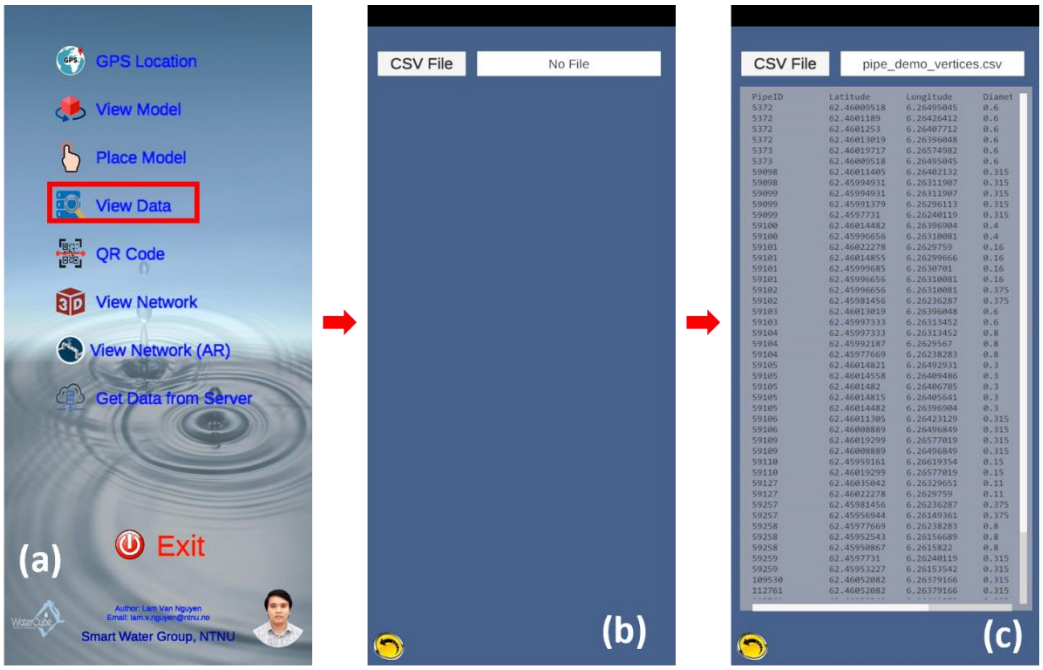



**Figure 1.13.** Viewing CSV data in the "*WaterNet*" application

To use this function, the user first selects the symbol 

on the menu screen of the application (**Figure 1.13**a). Next, select the button 

on the next screen (**Figure 1.13**b), and specify the location of the CSV file that the user wants to open. Finally, the entire CSV file is shown in a window at the center of the screen (**Figure 1.13**c). To exit this function, the user clicks on the button  in the bottom-left corner of the screen, the user afterward is led to the main menu screen.

### *1.3.5. QR Code*

This function allows the user to create and read QR codes. The attributes of sewer components can be coded and decoded using a QR code.

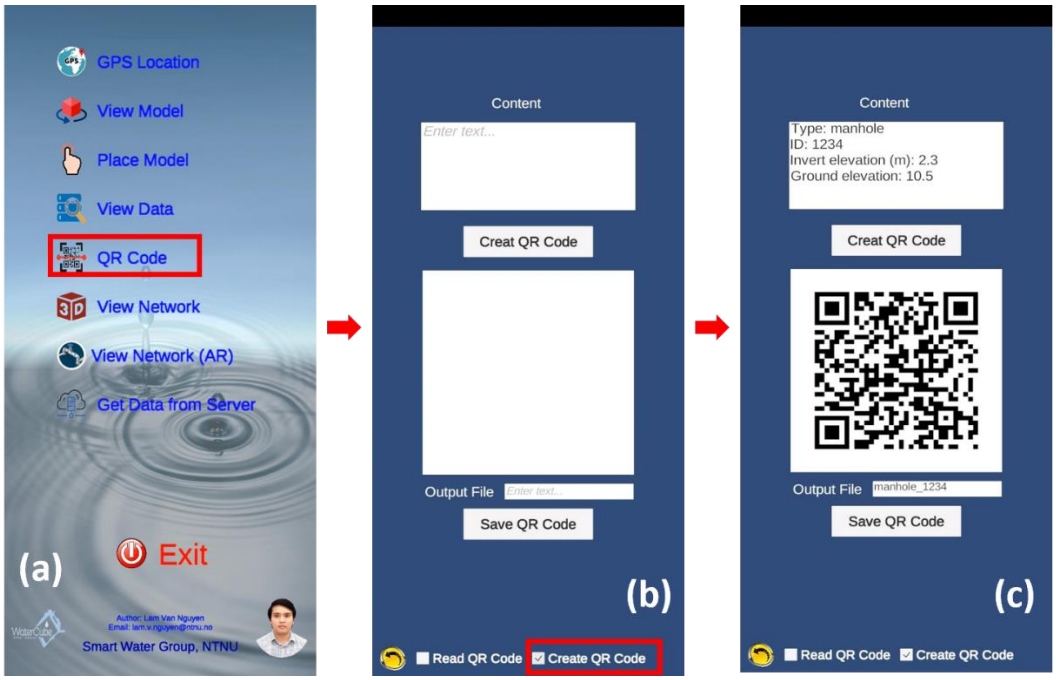



**Figure 1.14.** Creating a QR code in the "*WaterNet*" application

To create a QR code for assigning the attributes of the sewer component, the user first selects 

on the menu screen of the application (**Figure 1.14**a). Next, select the check box "*Create QR Code*" in the bottom-right of the screen (**Figure 1.14**b). Assign the wanted attributes for the object at the content box, select the button 

to create a new QR code, and assign created attributes for the sewer component. A new QR code is generated at the center of the screen (**Figure 1.14**c). To save the QR code, the user specifies the name of the QR code in the field "*Output File*", selects the button 

, then specifies the location to save the QR code on the device. Finally, this saved QR code can be transferred to a PC for storage and use for other purposes.

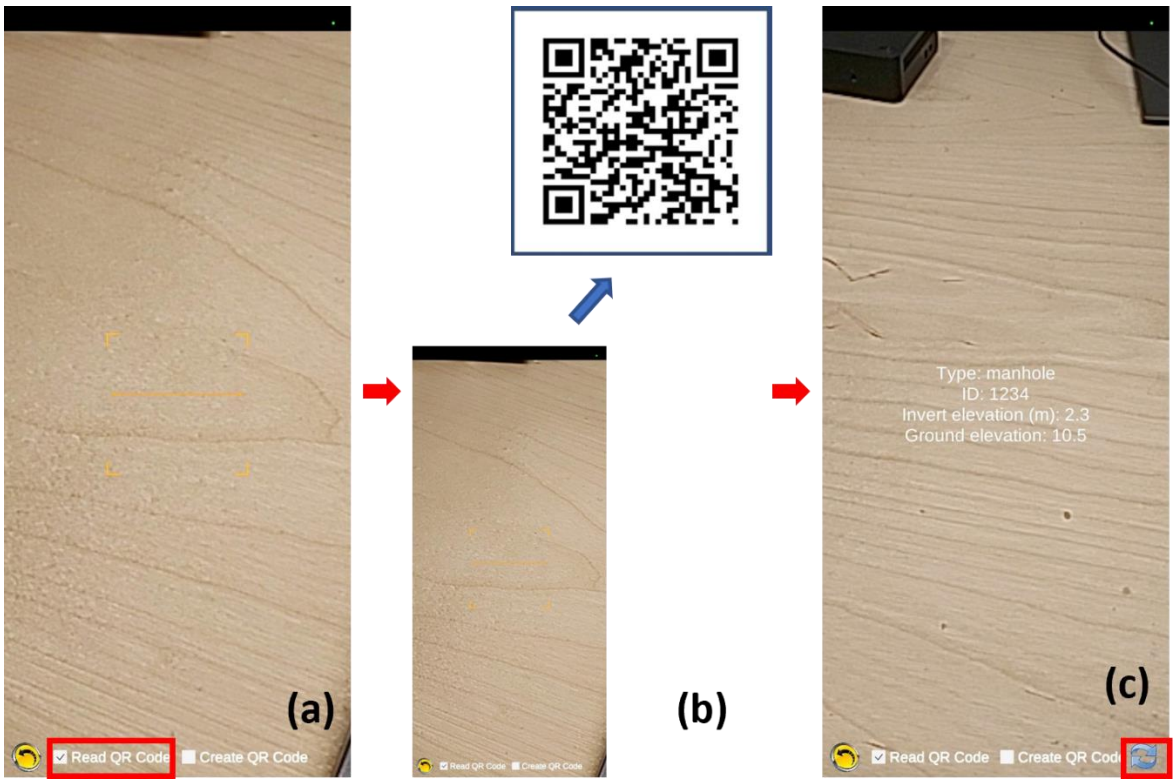



**Figure 1.15.** Reading a QR code in the "*WaterNet*" application

To read a QR code, the user first selects the symbol 

on the menu screen of the application (**Figure 1.14**a). Next, select the check box "*Read QR Code*" in the bottom-right of the screen (**Figure 1.15**a). Move the screen to the QR code that the user wants to read, it makes sure that the yellow square on the center of the screen covers the QR code (**Figure 1.15**b). The assigned attributes of the object will appear in the center of the screen (**Figure 1.15**c). To delete current information on the screen and read a new one, the user selects the symbol  in the bottom-right corner of the screen (**Figure 1.15**c). By clicking on the button  in the bottom-left corner of the screen, the user is led to the main menu screen.

### *1.3.6. View Network*

This function allows the user to import attributes of manholes or pipes from a CSV file and visualize them in real-time (**Figure 1.16**a), the user also visualizes the sewer conditions using data from a CSV file (**Figure 1.16**b). To activate this function, the user clicks on the symbol 

on the main menu of the screen.

To view the attributes of the sewer components in real-time, the user first selects the button 

on the top-left corner of the screen (**Figure 1.16**a). Next, specifies the location of the CSV file that contains the attributes of sewer components. An example of the attribute file

is shown in **Figure 1.17**. It is worth noting that the number of columns in this file is unlimited, the user can add as many columns as possible. Next, the user selects the check box 

, and an "*Attribute Table*" panel will appear on the right-hand side of the screen (**Figure 1.16**a). By moving and rotating the device screen to the target marker  on the center of the screen hit the wanted object, the corresponding attributes of the object will appear in the "*Attribute Table*" panel (**Figure 1.16**a).

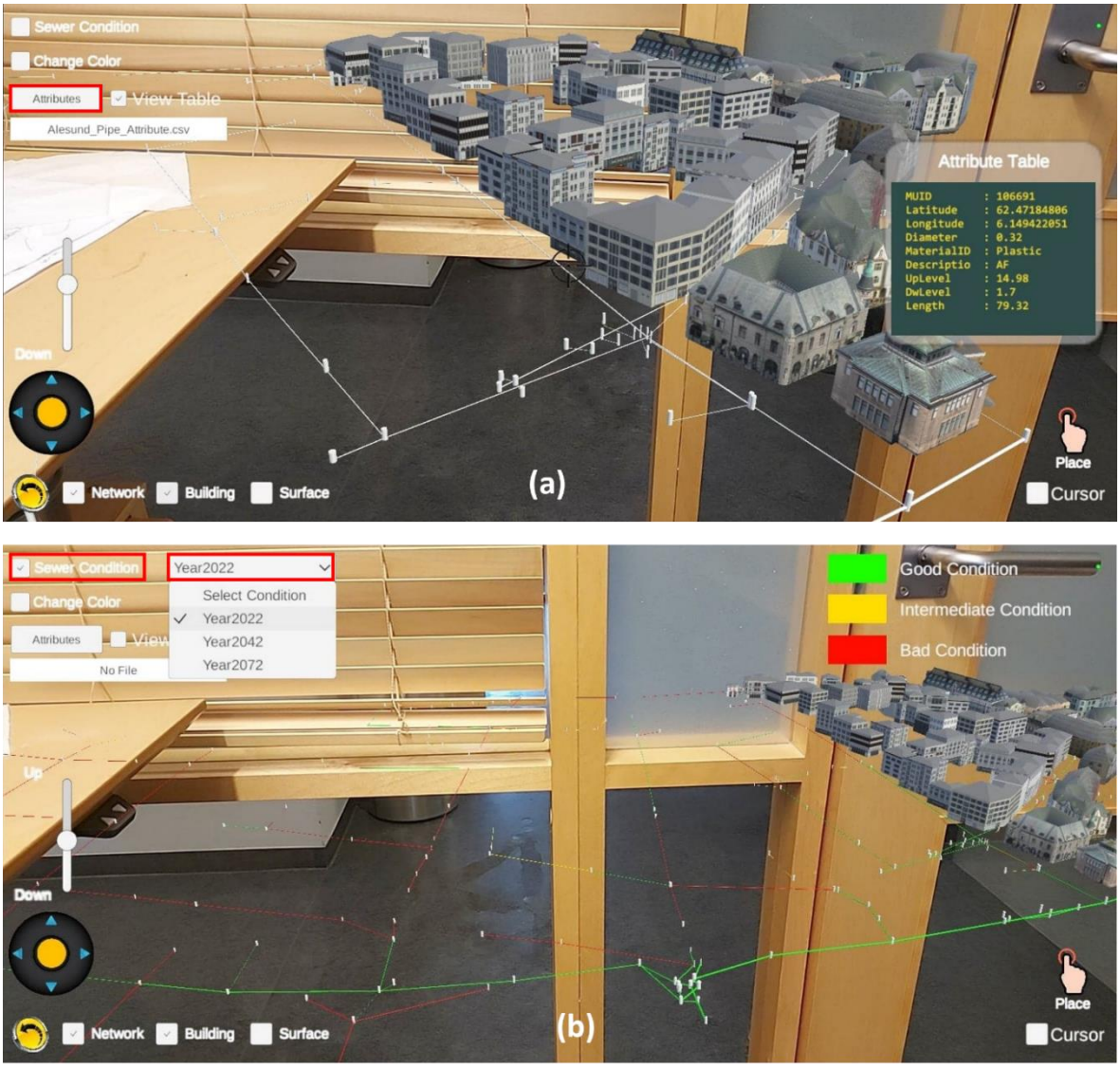



**Figure 1.16.** Viewing the sewer network in the "*WaterNet*" application

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | MUID | Latitude | Longitude | Diameter | MaterialID | Descriptio | UpLevel | DwLevel | Length |
| 2 | 99557 | 62.46256 | 6.167135 | 0.16 | Plastic | OV | 20.21 | 15.07 | 45.16 |
| 3 | 99557 | 62.46256 | 6.167143 | 0.16 | Plastic | OV | 20.21 | 15.07 | 45.16 |
| 4 | 99557 | 62.46257 | 6.167184 | 0.16 | Plastic | OV | 20.21 | 15.07 | 45.16 |
| 5 | 99557 | 62.46259 | 6.167282 | 0.16 | Plastic | OV | 20.21 | 15.07 | 45.16 |
| 6 | 99557 | 62.46262 | 6.167382 | 0.16 | Plastic | OV | 20.21 | 15.07 | 45.16 |

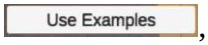
**Figure 1.17.** An example of the attribute file

To view the sewer conditions, the user first selects the check box 

in the top-left corner of the screen (**Figure 1.16**b).


**Figure 1.18.** Option for viewing sewer conditions

A window will appear on the center of the screen that requires the user to either use an example data or connect to a PC to transform the predictive conditions (**Figure 1.18**).

- If the user selects the button 

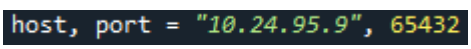
, the condition of sewer pipes in the year 2022 is showed up. This data was predicted using the machine learning model from one of the first author's works and it was set up as default values for example visualization.

- If the user selects the button 

, the application will send a request to a PC that is running a python code which is provided in **Table 1.1**. After getting the request from the "*WaterNet*" application, this python code will run a function to get the conditions of sewer pipes from a pre-registered location on the PC and send the data back to the android devices for visualization. This approach will be efficient in the case the user wants to modify the sewer conditions file for visualization for different scenarios. To transfer data from PC to android devices by using this approach, the user needs to do below steps:

  - *Step 1:* Ensure the PC for running the python codes and android devices are connected to the same internet network.

  - *Step 2:* Run the python codes provide in **Table 1.1** by using any Integrated Development Environment used for programming in Python (such as PyCharm, Spyder, etc.,). In this tutorial, we use Spyder to run python codes.

  - *Step 3:* Change the host address 

in the python code by the IP address created by the "*WaterNet*" application (**Figure 1.18**). Please

notice that this IP address will be different depending on each specific android device.

- *Step 4:* It takes several seconds to read and transfer data from a PC to an android device depending on the capacity of the data and internet connection.
- *Step 5:* If the data is loaded successfully, the combo box in the top-left corner of the screen will contain the years of sewer conditions (**Figure 1.16**b).

The data structure of the sewer conditions is partly shown in **Figure 1.19**. The first column contains the name/ID of sewer pipes in this data format. From the second column, the condition of the sewer pipes each year is stored in each column. From the second row, each sewer pipe's name and corresponding state are stored in each row.

| | A | B | C | D |
|---|---|---|---|---|
| 1 | ID | Year 2022 | Year 2042 | Year 2072 |
| 2 | 99484 | Good Condition | Good Condition | Bad Condition |
| 3 | 99468 | Good Condition | Good Condition | Bad Condition |
| 4 | 90235 | Good Condition | Good Condition | Good Condition |
| 5 | 90234 | Bad Condition | Bad Condition | Bad Condition |
| 6 | 89593 | Good Condition | Good Condition | Good Condition |
| 7 | 89590 | Good Condition | Good Condition | Good Condition |
| 8 | 88937 | Good Condition | Good Condition | Good Condition |
| 9 | 88935 | Good Condition | Good Condition | Good Condition |
| 10 | 88550 | Good Condition | Good Condition | Good Condition |
| 11 | 88467 | Good Condition | Good Condition | Good Condition |
| 12 | 88453 | Good Condition | Good Condition | Good Condition |
| 13 | 88449 | Good Condition | Good Condition | Good Condition |
| 14 | 88444 | Good Condition | Good Condition | Good Condition |
| 15 | 88417 | Good Condition | Good Condition | Good Condition |
| 16 | 88393 | Good Condition | Good Condition | Good Condition |
| 17 | 88344 | Good Condition | Good Condition | Good Condition |
| 18 | 88332 | Good Condition | Good Condition | Good Condition |

**Figure 1.19.** An example of the sewer conditions

### *1.3.7. View Network (AR)*

This function allows the user to view the sewer network using the Augmented Reality technique. Because of the limitation of the hardware of Android devices, this function is only applied to a small area of the study area. This function will define the relative relationship between the user's location and objects of the sewer network using a GPS signal. Therefore, this function does not work on HoloLens devices that do not have GPS receivers. Additionally, the accuracy of the objects detected using this function mainly depends on the accuracy of the GPS receiver on the android device. To activate this function, the user clicks on the symbol 

on the main menu of the screen. The interface of this function is shown in

**Figure 1.20**.

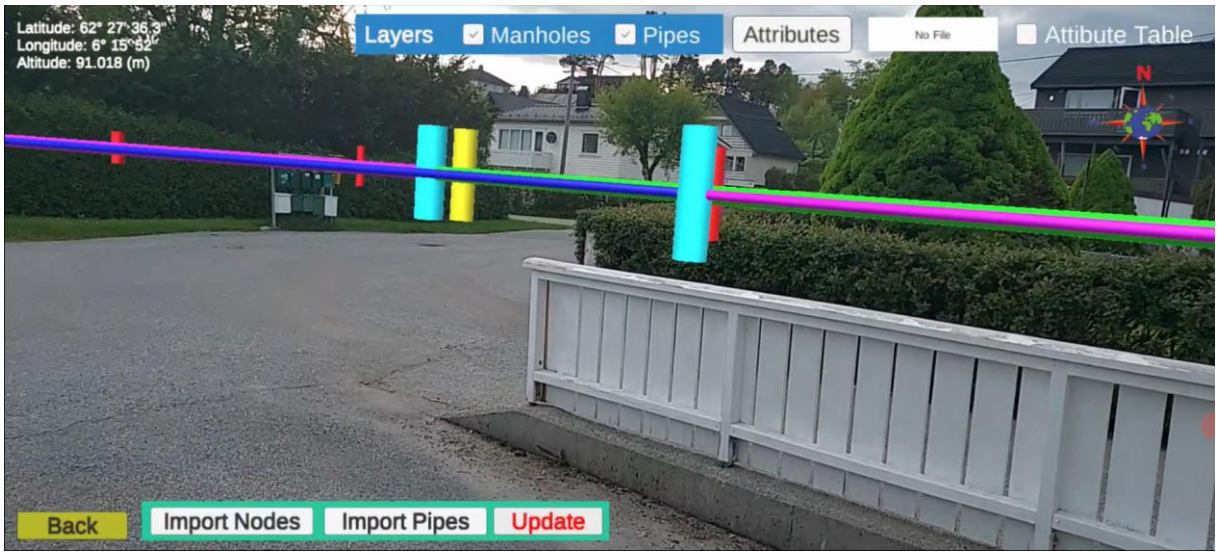



**Figure 1.20.** The interface of the "*View Network (AR)*" function

The below steps describe how to use this function:

- *Step 1:* Ensure that the location function on the android devices is enabled before activating this function.
- *Step 2:* Click on the button Import Nodes and/or button Import Pipes to import manholes and/or pipes from the CSV file. The structures of the CSV file containing the manholes and pipes are shown in **Figure 1.2**a and **Figure 1.2**b, respectively.
- *Step 3:* The symbol  shows the different angles in degrees relative to the geographic North pole and the android device's compass. It is recommended to wait for the geographical locations at the top-left corner of the screen to be shown up and this angle is approximately zero before clicking on the button Update to put the sewer network on the screen.
- *Step 4:* To turn display/hide the manhole layer or pipe layer on the screen, the user can check/uncheck the corresponding check boxes Manholes or Pipes, respectively.
- *Step 5:* To view the attributes of manholes or pipes, the user must import the CSV file that contains the attributes of manholes or pipes by clicking on the button Attributes and checking on the check box "*Attribute Table*" Attibute Table at the top-right corner of the screen. By moving the screen to the point at the center of the screen hit the objects, the attributes of corresponding objects will appear on the small panel at the left-hand side of the screen (**Figure 1.21**).

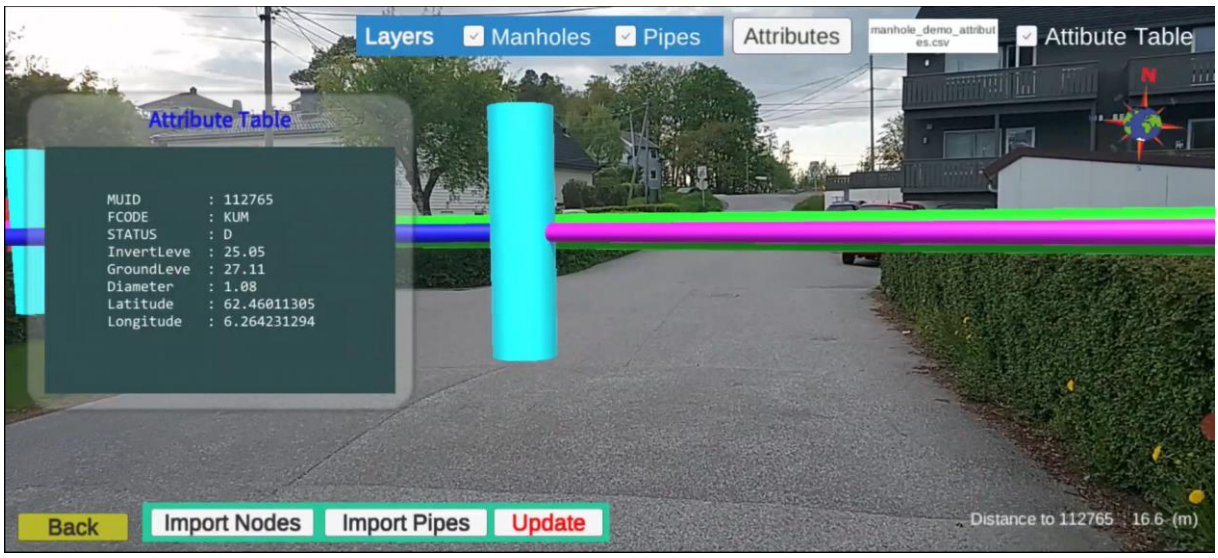



**Figure 1.21.** Viewing the object's attributes in the "*View Network (AR)*" function

### *1.3.8. Get Data from Server*

This function allows the user to access and visualize data from the provided server in real-time. In general, accessing the server to get data requires authorization (i.e., username and password), therefore, we only illustrate how to access and get data from the "*StaalCloud*" server controlled by Ålesund municipality in this example.

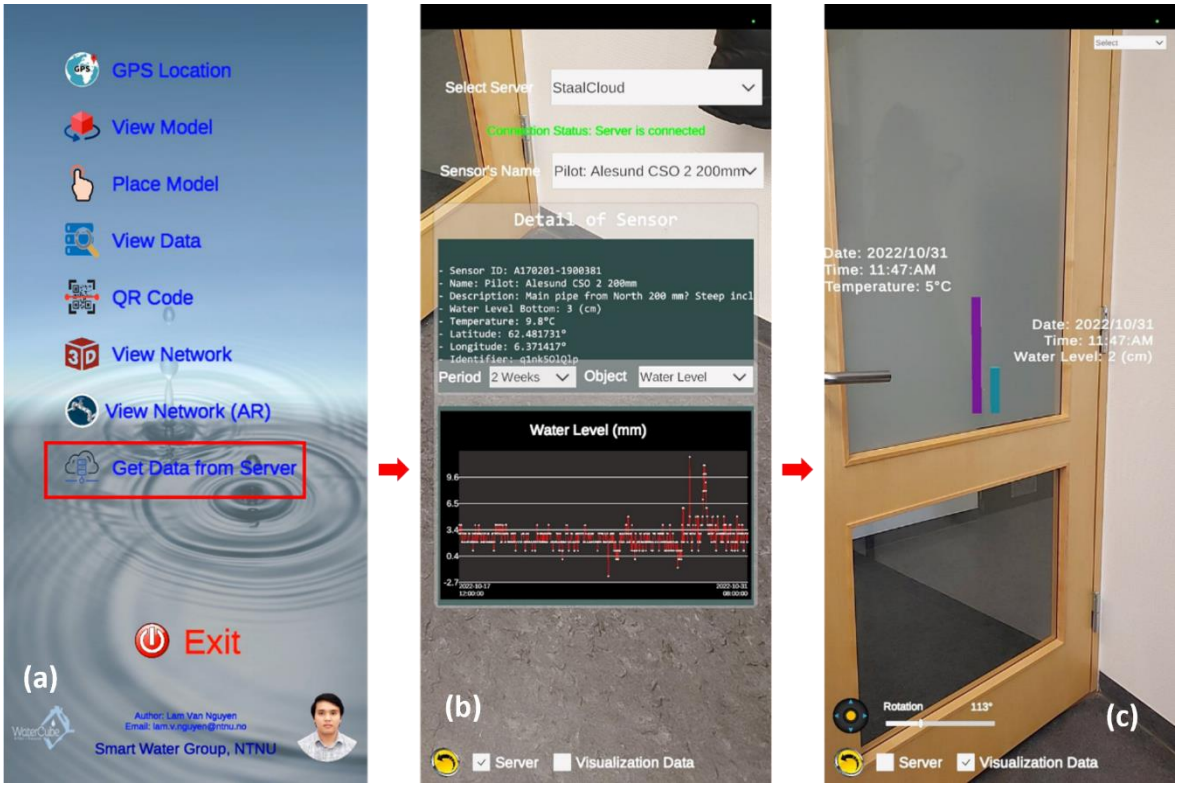



**Figure 1.22.** The interface of the "*Get Data from Server*" function

The below steps describe how to use this function:

- *Step 1:* Ensure that the internet connection is connected to the android devices. Select the symbol 

from the main menu of the application (**Figure 1.22**a).
- *Step 2:* Select the combo box "*Select Server*" to connect to the server. If the application connects to the server successfully, the red text "Connection Status: Not connected" will change to the green text "Connection Status: Server is connected" (**Figure 1.22**b).
- *Step 3:* The combo box "*Sensor's Name*" lists all sensors on the server. A comparison of the number of sensors on the server and the number of sensors obtained by this function is shown in **Figure 1.23**.

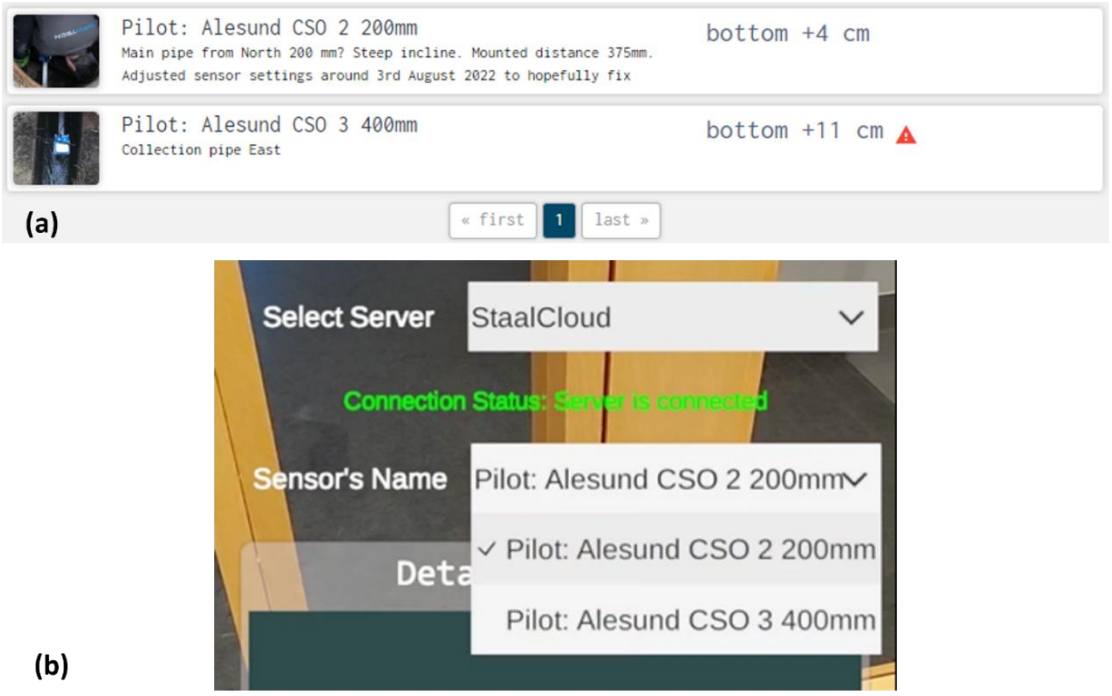
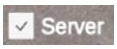



**Figure 1.23.** Sensors on the server **(a)** and the result obtained by the "*Get Data from Server*" function **(b)**

- *Step 4a:* To view data in real-time, the user selects the checkbox 

at the bottom of the screen (**Figure 1.22**b). This function allows the user to view the water level and water temperature received from sensors by changing the values of the combo box "*Object*" 

in **Figure 1.22**b. Additionally, this function allows the user to view these aforementioned values in 1 day, 2 days, 1 week, 2 weeks, and 1 month by changing the values of the combo box "*Period*" 

in **Figure 1.22**b. **Figure 1.24** illustrates an example of real-time data received from the server using the "*Get Data from Server*" function.

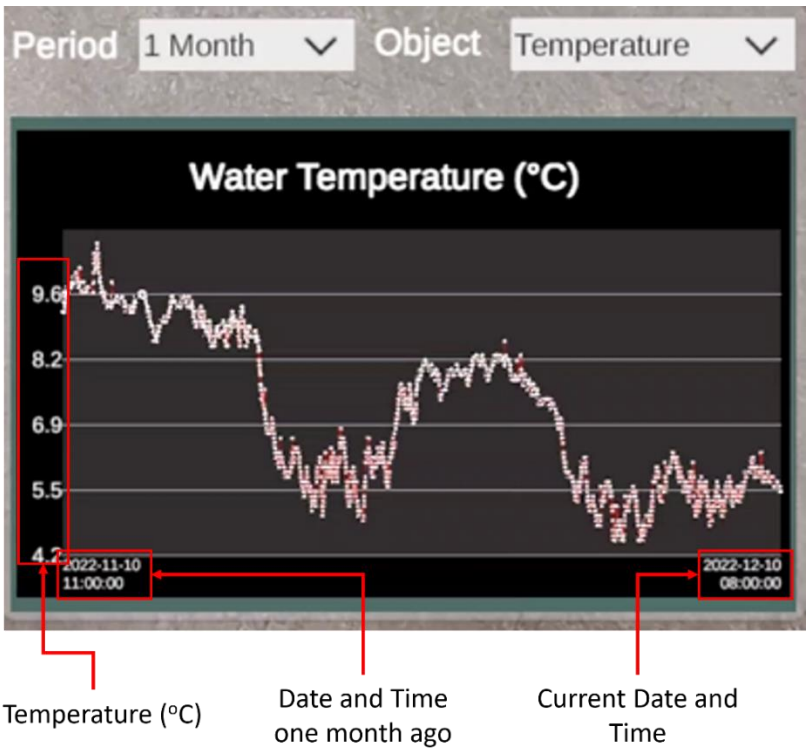



**Figure 1.24.** Real-time data visualization obtained from the server

- *Step 4b:* To visualize data in 3D, the user selects the checkbox 

at the bottom of the screen. The water level and water temperature will be visualized simultaneously (**Figure 1.22**c).

## 2. HoloLens Application

The "*WaterNet*" application used on the HoloLens device is distributed by a project that can be implemented via Unity. To download the Unity project, the user accesses the link provided in **Table 2.1**.

**Table 2.1.** Downloaded link of the "WaterNet" application on the HoloLens device

| Name | Download link |
|---|---|
| WaterNet_HoloLens | https://www.mediafire.com/file/lgqg6k2kxdi54bk/WaterNet_HoloLens.zip/file |

The steps for implementing and running this application are presented as follows:

- *Step 1:* Download and extract the file from the link that is provided in **Table 2.1**.
- *Step 2:* Download and install the Unity software (https://unity.com/download). The Unity version 2020.3.32f1 is recommended for the implementation of this application.
- *Step 3:* Open Unity and import this project. Make sure that Unity software is optimized

for HoloLens devices by selecting *File/Build Setting…* In the *Platform* section, select "*Universal Windows Platform*" and *HoloLens* is selected in the section *Target Device* (**Figure 2.1**).

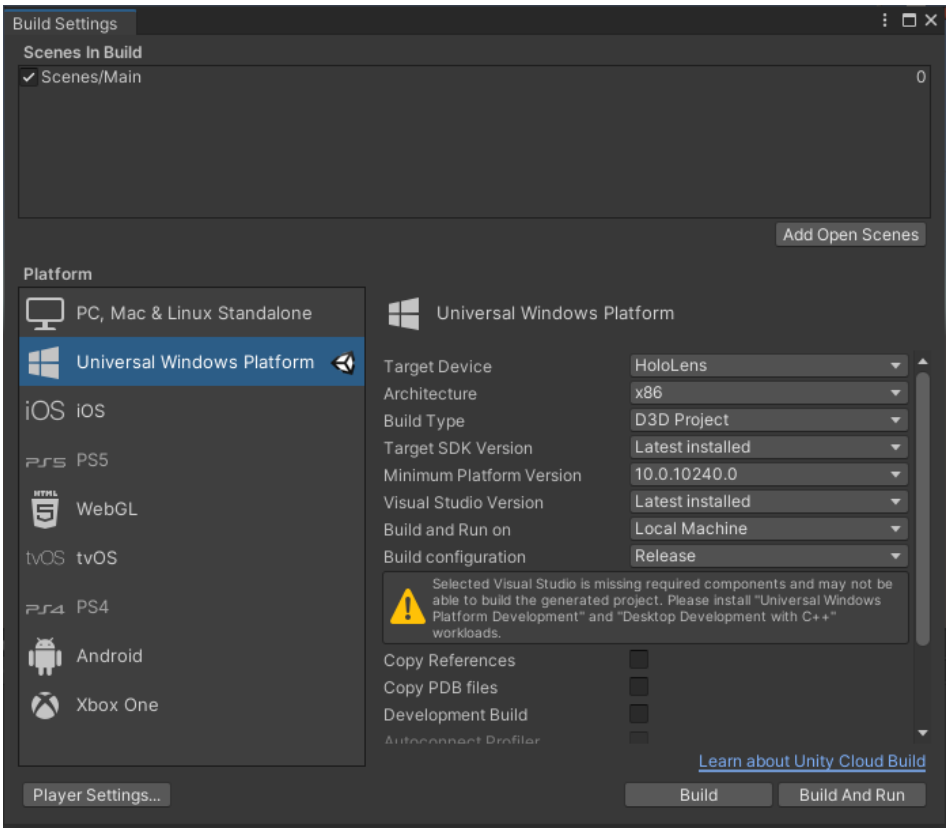



**Figure 2.1.** Unity configuration for HoloLens device

If the "*Universal Windows Platform*" option is not installed, the user can install this package via Unity Hub (**Figure 2.2**).

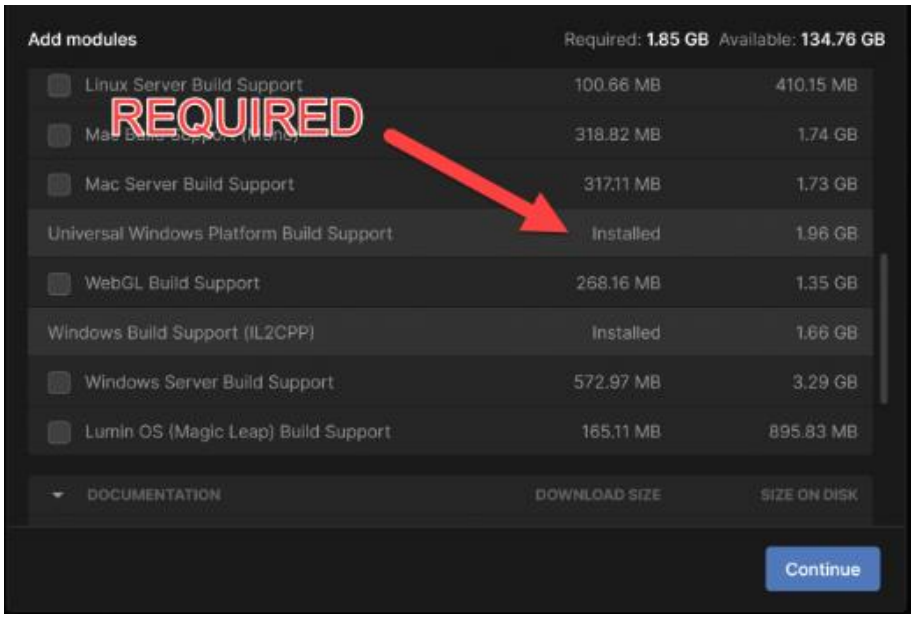



**Figure 2.2.** Installing the "*Universal Windows Platform*" package

If the user sees the notice reminding to install "*Universal Windows Platform Development*" and "*Desktop Development with C++*" workloads as in **Figure 2.1**, please open "*Visual Studio Installer*" from the ***Start*** menu and install them as in **Figure 2.3**.

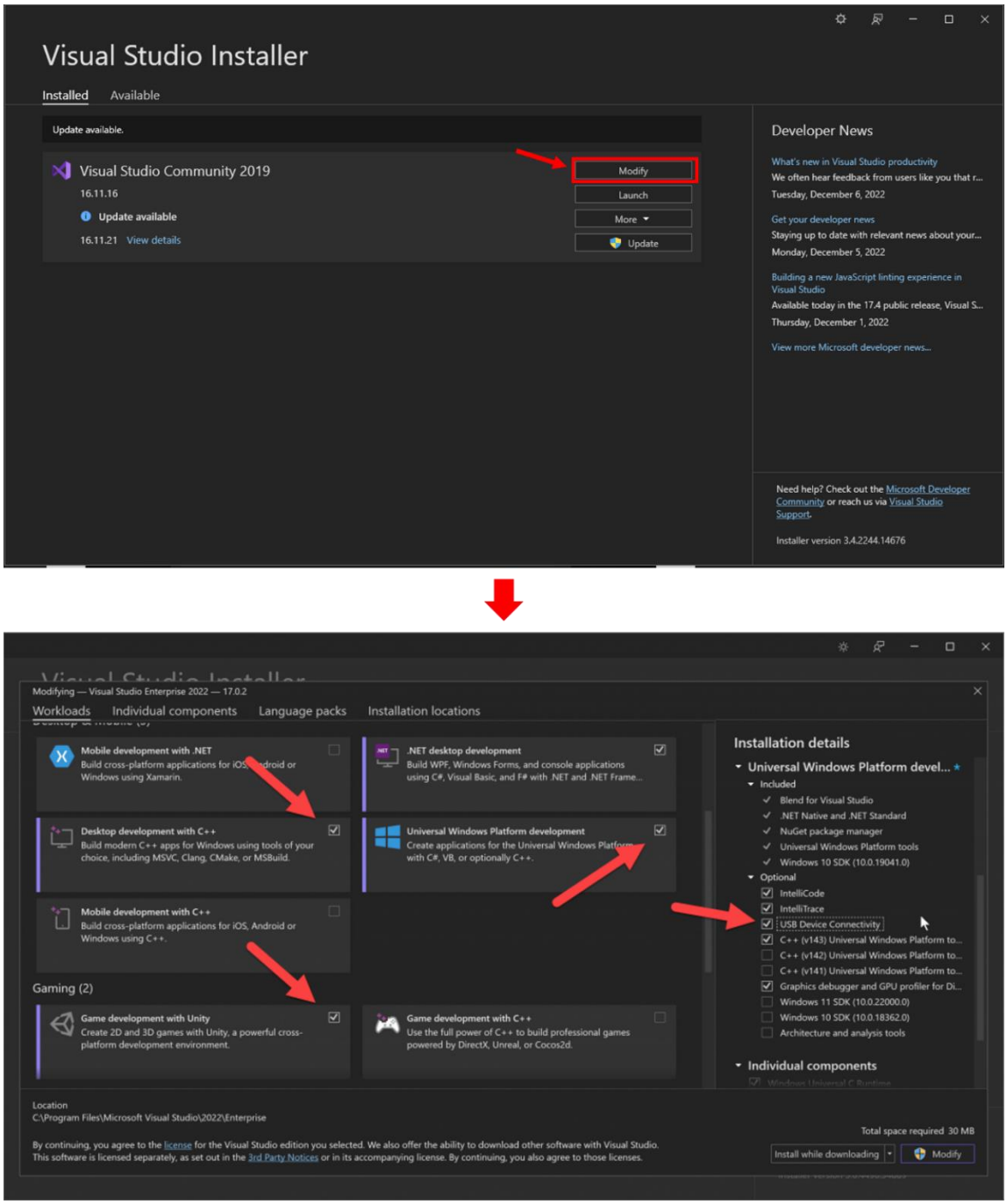



**Figure 2.3.** Installing the "*Universal Windows Platform Development*" and "*Desktop Development with C++*" workloads

After installing the packages and workloads successfully, the configuration window for HoloLens looks as in **Figure 2.4**. Before compiling this application on a HoloLens device, the user must add the "*Main*" scene in the section "*Scenes In Build*" (**Figure 2.4**) and activate the object "*SceneDescriptionPanelRev*" as in **Figure 2.5**.

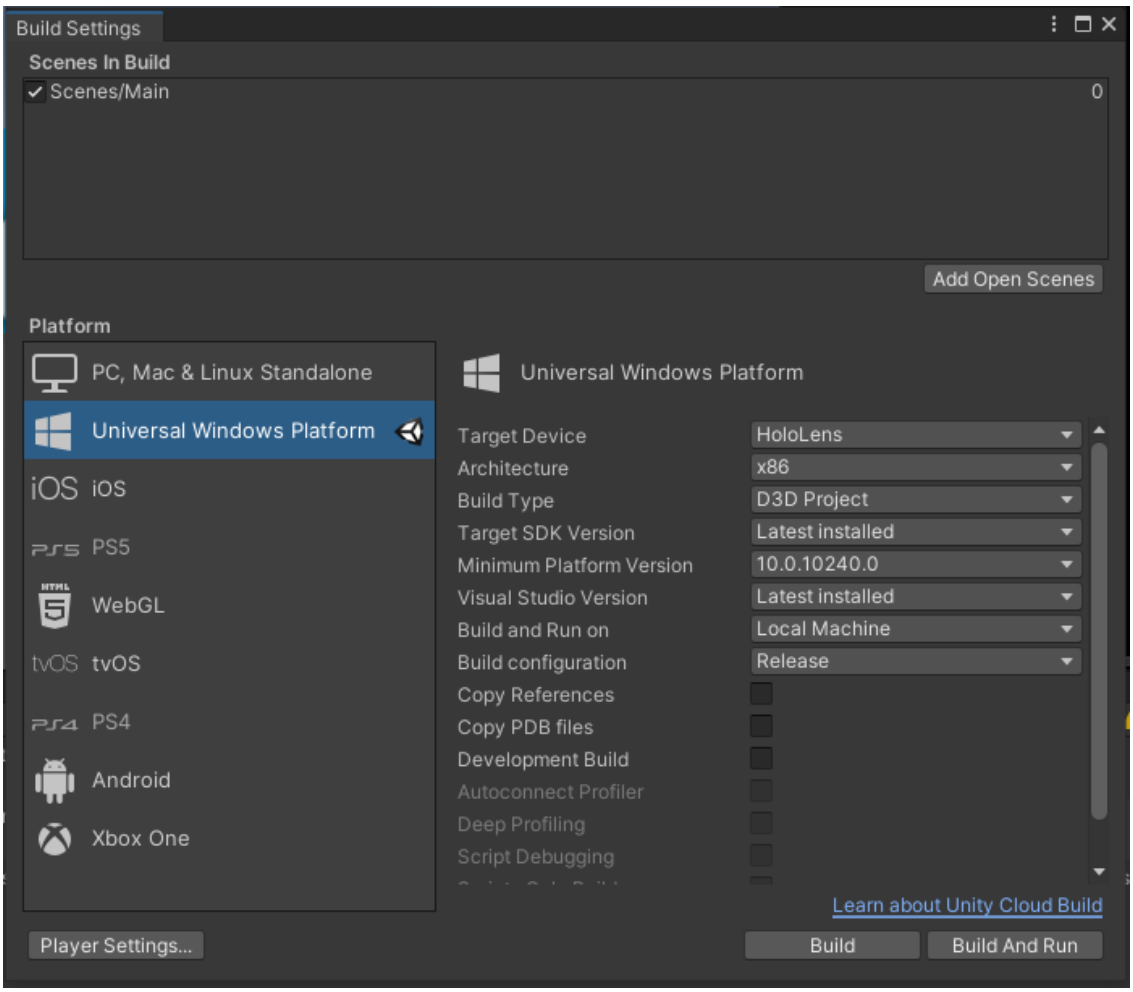



**Figure 2.4.** Successful configuration for HoloLens device in Unity

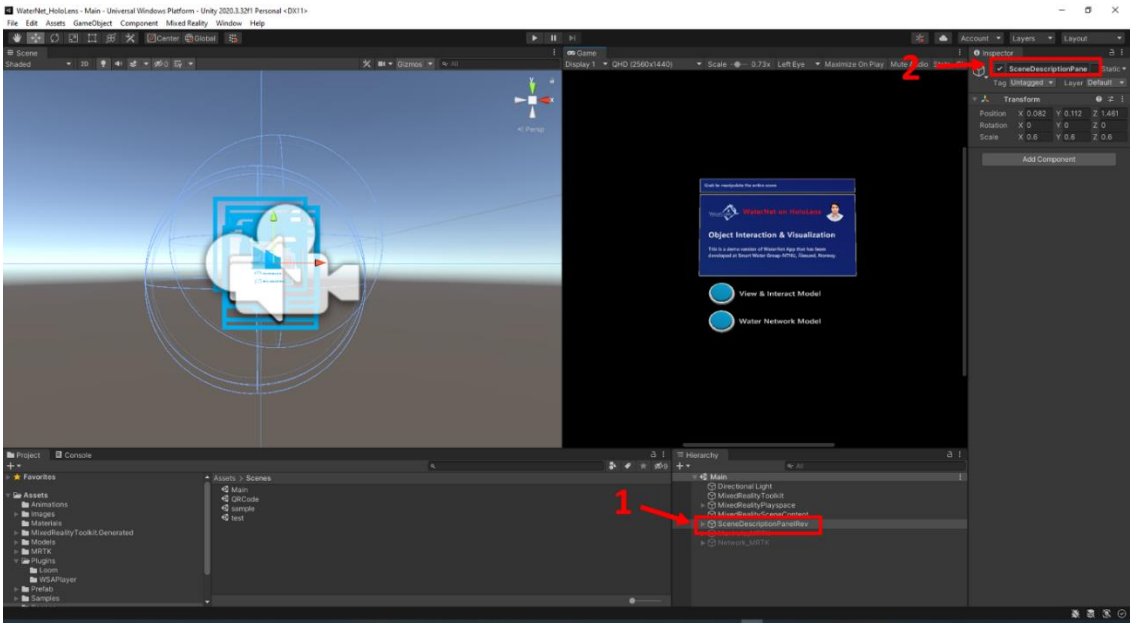



**Figure 2.5.** The "*WaterNet*" project in Unity

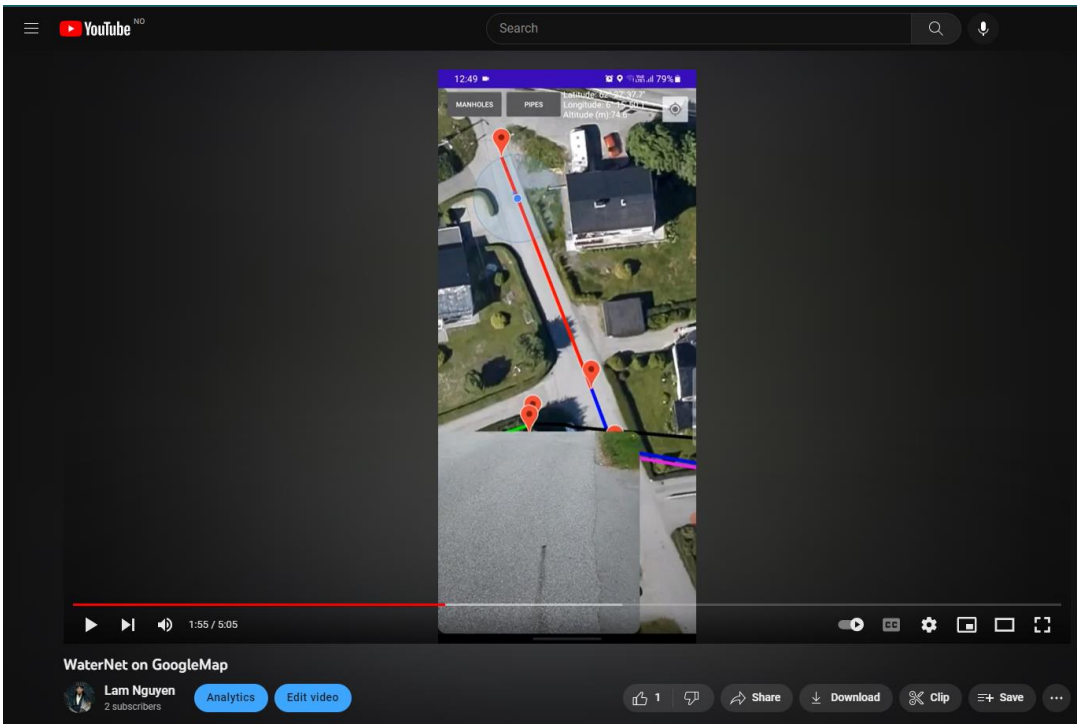
➢ *Step 4*: Connect the PC with the HoloLens device. Select *File/Build Setting…/Build and Run* to compile the application on the HoloLens device.

# 3. Video examples

This section provides video examples of using the above functions.

## *3.1. A Video Example of the "WaterNet on GoogleMap" Application*

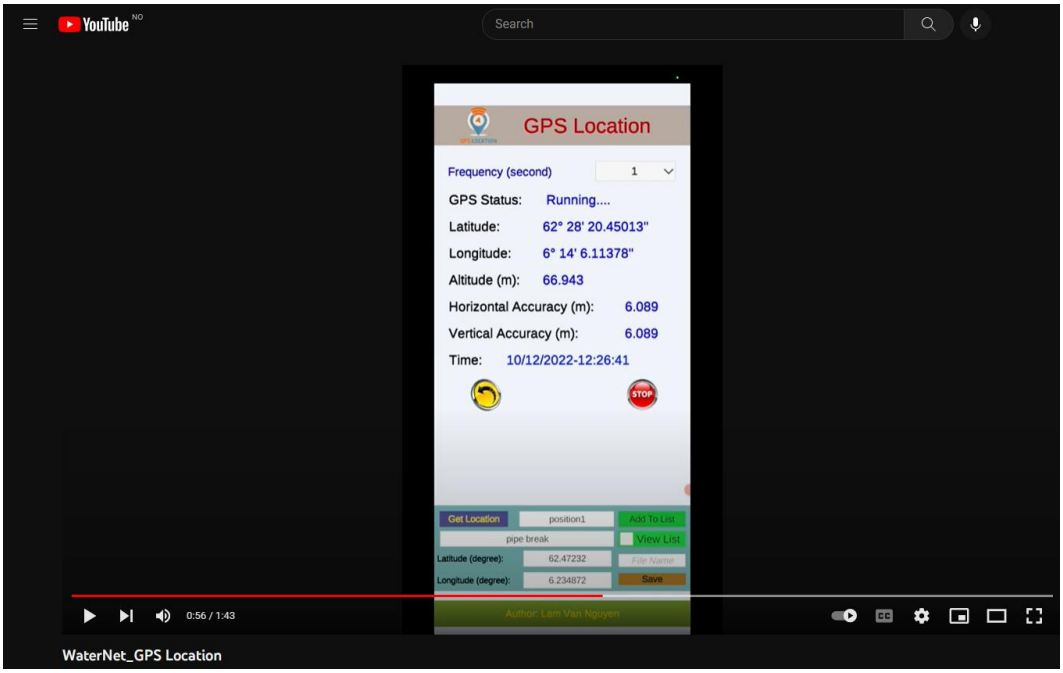
Video link: https://www.youtube.com/watch?v=3jfjDVP6Cag




## *3.2. Video Examples of the "WaterNet" Application on an Android Device*

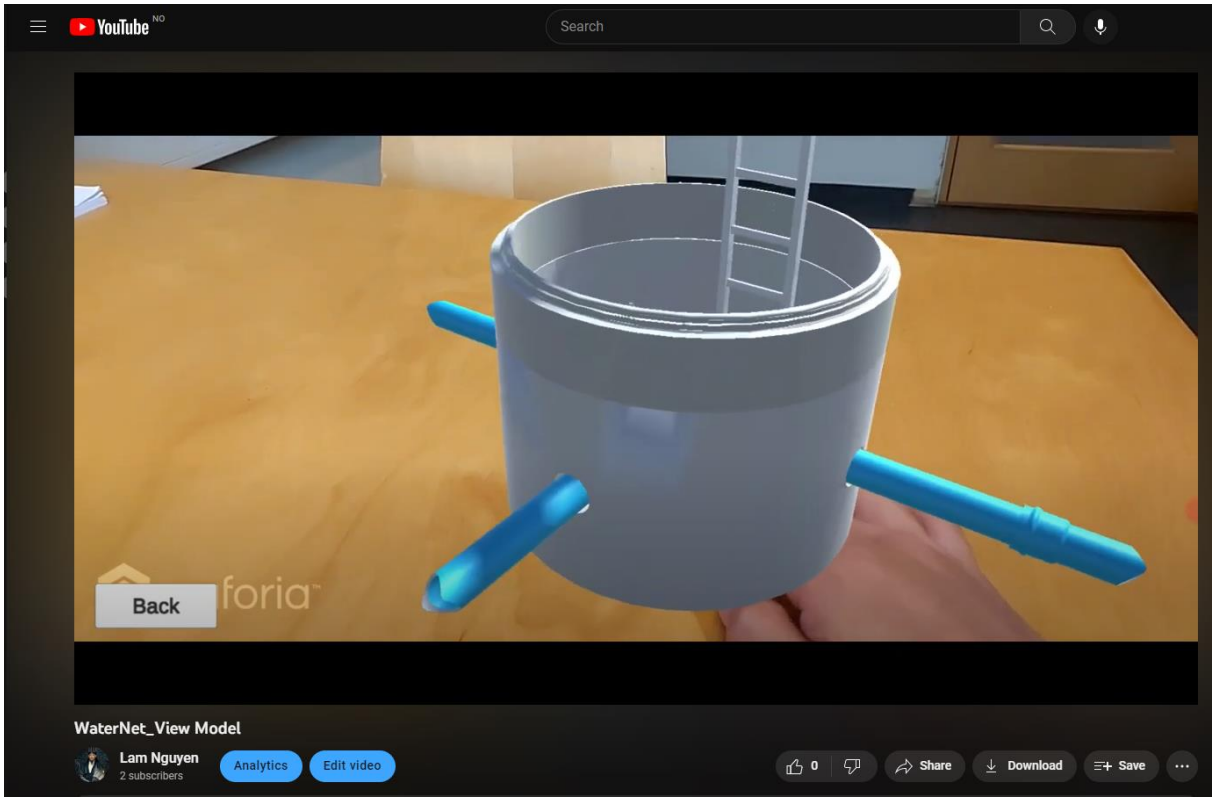
### *3.2.1. GPS Location*

Video link: https://www.youtube.com/watch?v=u_gcxmjHVmk



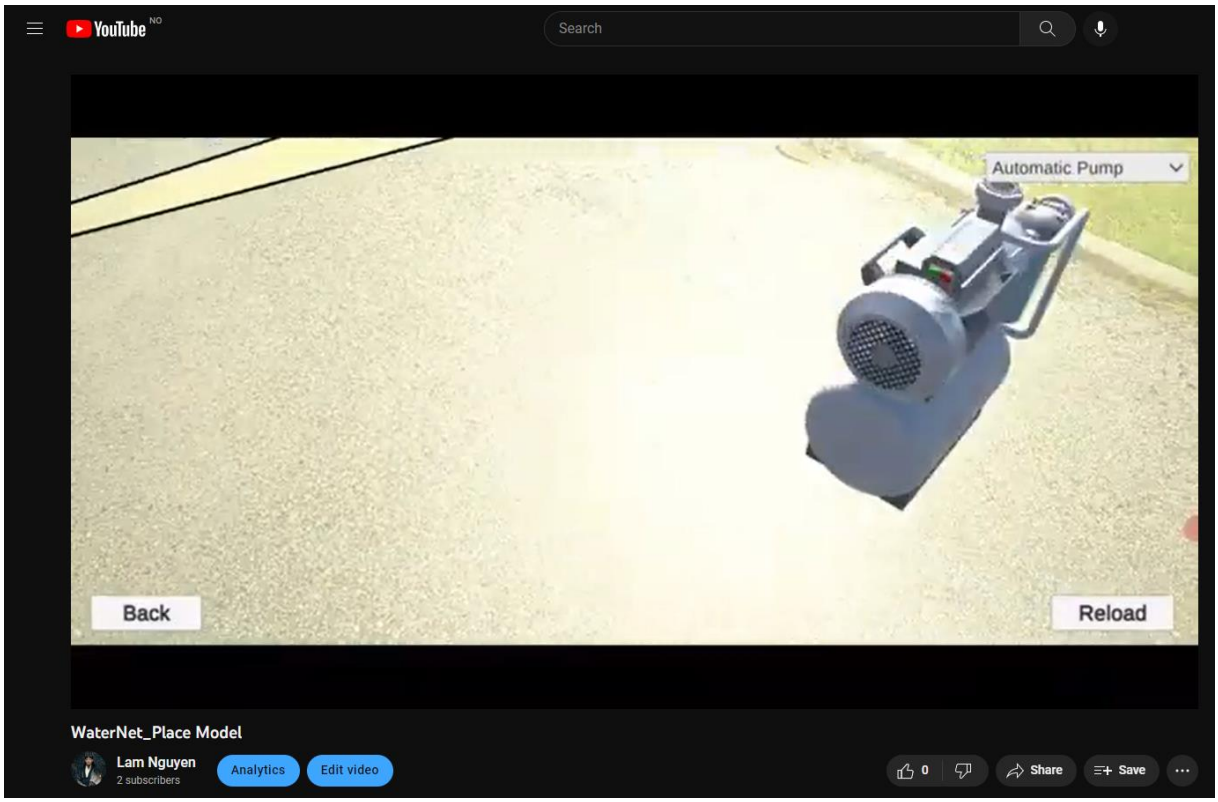
### 3.2.2. *View Model*

Video link: https://www.youtube.com/watch?v=pBOkH39HMbw




### 3.2.3. *Place Model*

Video link: https://www.youtube.com/watch?v=WpE02c0Lo5U



### 3.2.4. *View Network*

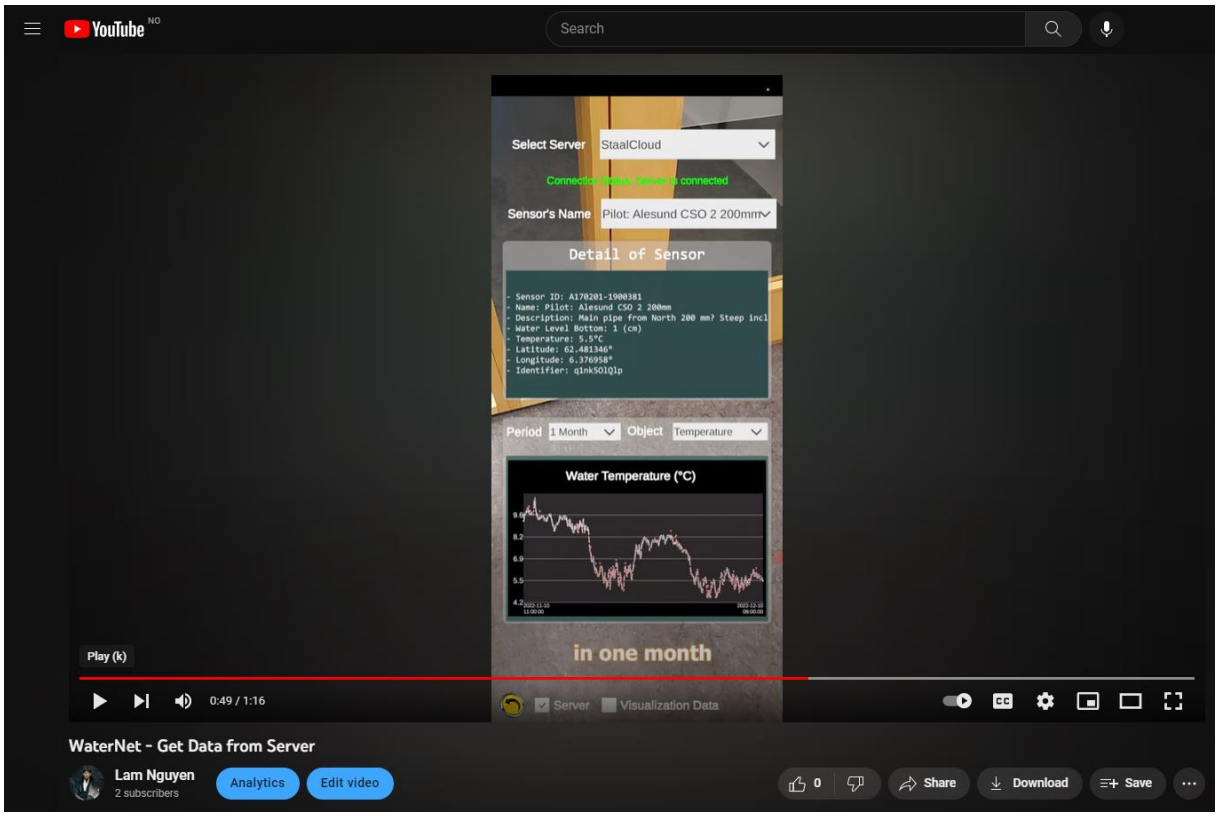
Video link: https://www.youtube.com/watch?v=sK93RZY0Lsg


### 3.2.5. *View Network (AR)*

Video link: https://www.youtube.com/watch?v=aOKMS2HQd8A


### 3.2.6. *Get Data from Server*

Video link: https://www.youtube.com/watch?v=WZU25vCQ4FA




### *3.3. A Video Example of the "WaterNet" Application on HoloLens*

Video link: https://www.youtube.com/watch?v=bjv-whMe4RA