**System Design Document**

**Intelligent Manhole Cover system**

by

Yihuai Zhang 1064213

Jinyu Wang 1025853

March 14th, 2021

Department of Computer Science

Wenzhou-Kean University

About the System Design Document

**Version Control**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Vision No. | Title | Author | Date | Authorization |
| 1.0 | SDD Report | Zhang Yihuai  Wang Jinyu | 3/14/2021 |  |
|  |  |  |  |  |

**Document Properties**

|  |  |
| --- | --- |
| **Items** | **Details** |
| Document Title | Intelligent Manhole Cover System |
| Author | Zhang Yihuai  Wang Jinyu |
| Creation Date | 3/14/2021 |
| Last Updated | 3/14/2021 |

**Table of Contents**

1. Introduction 1

1.1 Purpose of the System 1

1.2 Design Goals 1

1.3 Definitions, acronyms, and abbreviations 2

1.4 References 3

1.5 Overview 3

2. Current Software Architecture 4

3. Proposed Software Architecture 4

     3.1 Overview 4

     3.2 Subsystem Decomposition 5

     3.3 Hardware/Software Mapping 7

     3.4 Persistent Data Management 7

     3.5 Access Control and Security 9

     3.6 Global Software Control 9

     3.7 Boundary Conditions 10

4. Subsystem Services    10

     3.1 User Authentication 10

     3.2 Map Display 11

     3.3 Manhole Cover State Update 11

     3.4 Alarm and Maintenance 12

1. **Introduction**
   1. **Purpose of the system**

A smart city is a collection of intelligent systems and end-to-end services, and the data and information needed for the is the most widespread infrastructure, manhole covers can be found in every corner of the city. As the most widespread infrastructure, manhole covers can be found in every corner of the city. Manhole covers are a very important part of the urban drainage system, connecting the sewer to the ground. In the past few years, incidents of people's property loss due to lost or damaged manhole covers or construction have been common, and lost or damaged manhole covers often pose a great threat to the safety of the public. Ensuring that manhole covers are intact provides great safety for pedestrians and vehicles on the road, and also ensures the normal operation of the municipal drainage system. The goal of this project is to create a smart manhole cover based on Arduino microcontroller combined with various sensors, which comes with various functions and connected with big data platform. It can realize real-time monitoring of environmental temperature, humidity and other data, and display on the big data platform whether the manhole cover is missing or offset, and monitor whether the sewer is blocked.

* 1. **Design Goals**

The system shall allow the users to:

1. Accurately locate the position of the manhole cover by GPS module.

2. Be able to receive alarm messages from the smart manhole cover when it is damaged or lost.

3. Have the ability to view the data collected by the smart manhole cover on the Android operating system.

4. Monitor the temperature, humidity and rainfall of the ground detected by the sensors through the data platform.

* 1. **Definitions, acronyms, and abbreviations**

Arduino: Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

GPS: The Global Positioning System (GPS), originally Navstar GPS, is a satellite-based radionavigation system owned by the United States government and operated by the United States Space Force.

SDD: Software Design Document, a written outline of the development of a course or a description of a software product

RF chip: Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify and track tags attached to objects.

DHT11: The DHT11 is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). It’s fairly simple to use, but requires careful timing to grab data.

NodeMCU: NodeMCU is a low-cost open source IoT platform. It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which was based on the ESP-12 module. Later, support for the ESP32 32-bit MCU was added.

IDE: An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.

* 1. **References**

What is an Arduino? https://learn.sparkfun.com/tutorials/what-is-an-arduino/all#

"GPS: Global Positioning System (or Navstar Global Positioning System)" Wide Area Augmentation System (WAAS) Performance Standard, Section B.3, Abbreviations and Acronyms. Archived April 27, 2017, at the Wayback Machine.

IEEE Standard for Information Technology--Systems Design--Software Design Descriptions. doi:10.1109/ieeestd.2009.5167255. ISBN 978-0-7381-5925-6.

Automatic Identification and Data Collection (AIDC) Archived May 5, 2016, at the Wayback Machine

https://shop.evilmadscientist.com/productsmenu/716

Zeroday. "A lua based firmware for wifi-soc esp8266". Github. Retrieved 2 April 2015.

"Programming software and the IDE". BBC Bitesize. Retrieved 10 October 2018.

IoTDB User Guide (V0.10.x)-Architecture https://iotdb.apache.org/zh/UserGuide/V0.10.x/Overview/Architecture.html

* 1. **Overview**

This document completely describes the system at the architecture level, including subsystems and their services, hardware mapping, data management, access control, global software control structure, and boundary conditions. The SDD should define a virtual machine that implements all requirements in the SRS, and it should provide a foundational guide for further implementation details all the way to an executable solution.

The document contains the purpose of the system, the design goals, the current software architecture, the proposed software architecture, and the subsystem services.

SDD is a "live" document that should be incrementally expanded and refined during review cycles.

1. **Current software architecture**

This project needs a GPS Module, DHT11 as a temperature and humidity sensor, a water senser, an Arduino board, a ESP8266 NODEMCU module to accomplish the function structure shows below:

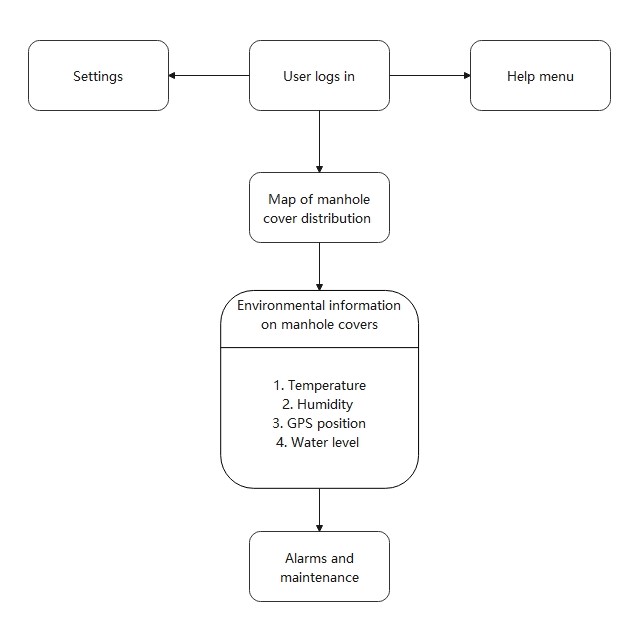
****

Figure 1. A NodeMCU module

**3. Proposed Software Architecture**

**3.1 Overview**

Our proposed system create a smart manhole cover that connected with big data platform, and realize real-time monitoring of parameters of that manhole. This system makes use of Arduino microcontroller combined with various sensors. It can be either displayed wither on the big data platform or be accessed by users whether the manhole cover is missing or offset. Meanwhile, more complex functions are provided to the staff of the sewerage administration. They can monitor the environmental temperature, humidity and whether the sewer is blocked directly through their phones with our mobile application.

The hardware control is based on Arduino boards. The data management part makes use of IoTDB, a newly developed data management engine with advantages of light weight structure, high performance and usable features. And the user interfaces shown on mobile phone will be displayed on an Android app. The application will be designed to provide different services to 3 groups of users. One group is granted the access to data of all sensors as well as request of moving the manhole covers, such that our identification strategies should be powerful enough to avoid the losses or damage of manhole covers.

**3.2 Subsystem Decomposition**

The whole architecture can be decomposed into four subsystems:

User Authentication:

The main function of this part is to deny access from illegal user. This should include not only password or SMS checking but also check whether it’s a staff member or average user. The system needs to decide which operations are allowed for that user. You can see how the users are classified in Figure 2.

Map display:

After the user passes login authentication, the application will request the user’s location and get a map that display his surroundings. You can see the relationships between map and other subsystems in Figure 2.

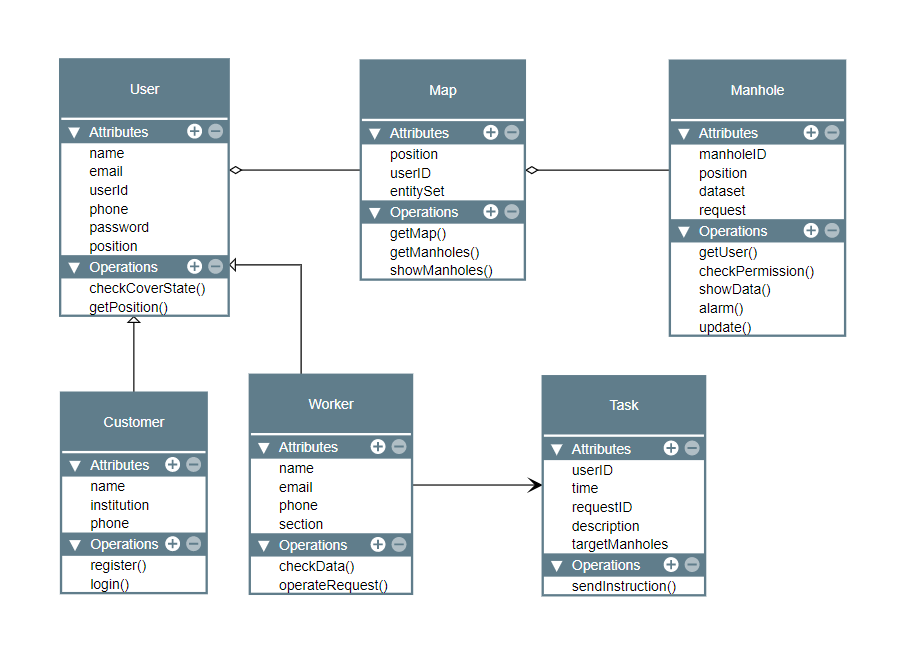


Figure 2. UML class diagram of Intelligent Manhole Cover system

Manhole cover state update:

After the map is displayed, the system will search for manhole covers in this area with the help of GPS positioning module in each cover. The server will return the result along with the state of that cover (missing/offset/intact), and data of other sensors just for staff. Icons which represent manhole covers will be displayed on the map at corresponding positions. You can see how each manhole get updated in Figure 2.

Alarm and maintenance:

Once the anomalous state is detected, an alarm message will be sent to the users or staff. For the staff, they can send a request to the server and tell it he will go for maintenance of which manhole. The platform will also show whether each problem are being handled by which staff member.

**3.3 Hardware/Software Mapping**

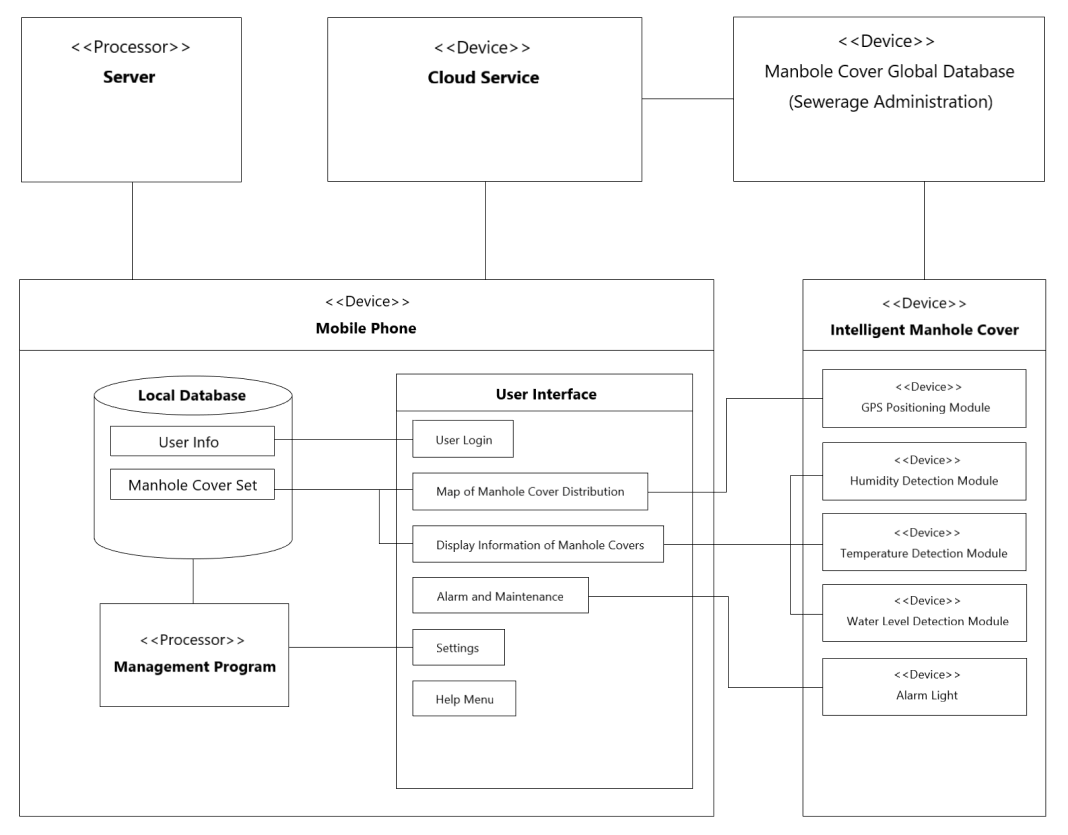


Figure 3. UML deployment diagram of Intelligent Manhole Cover system

**3.4 Persistent Data Management**

IoTDB (Internet of Things Database) is an integrated data management engine designed for timeseries data, which can provide users specific services for data collection, storage and analysis. Users can use JDBC to import timeseries data collected by sensor on the device to local/remote IoTDB. Data are written to the TsFile (local or on HDFS). For the data written to IoTDB and local TsFile, users can use TsFileSync tool to synchronize the TsFile to the HDFS. For the data written to local TsFile, users can use TsFile-Hadoop-Connector or TsFile-Spark-Connector to allow Hadoop or Spark to process data. The results of the analysis can be write back to TsFile in the same way. Also, IoTDB and TsFile provide client tools to meet the various needs of users in writing and viewing data in SQL form, script form and graphical form. The architecture of IoTDB is shown below in figure 4:

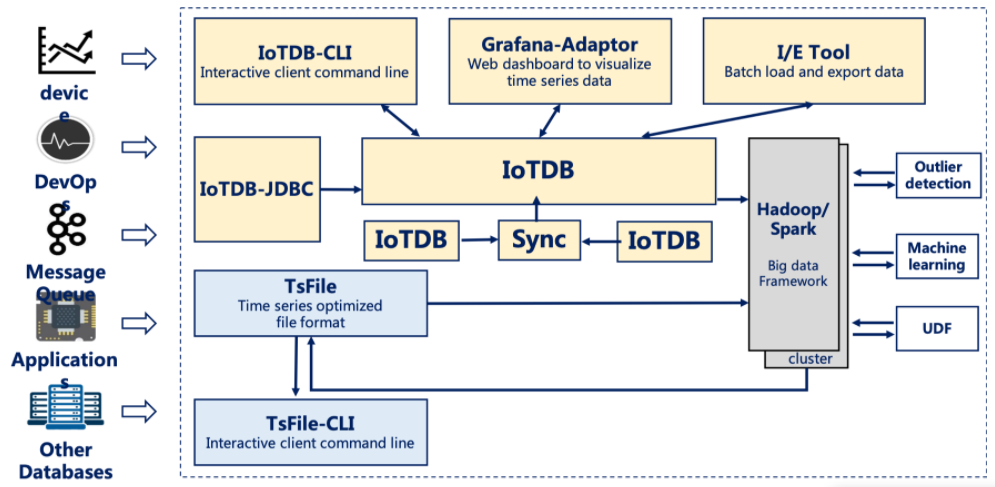


Figure 4. UML deployment diagram of Intelligent Manhole Cover system

Data stored in IoTDB follows a hierarchical relationship: power group layer - power plant layer - device layer - sensor layer. All the data and paths stored in IoTDB are organized by sensors. IoTDB can locate data files accurately with the help of timeseries. A timeseries path can be thought of as the complete path of a sensor that produces the time series data. All timeseries paths in IoTDB must start with root and end with the sensor. A timeseries path can also be called a full path. Users can set any prefix path as a storage group. Thus, we can group sensors on the same device or a series of same devices together and manage it, which also enables IoTDB to store all devices under it in the same folder.

When the time series is written and encoded as binary data according to the specified type, IoTDB compresses the data using compression technology to further improve space storage efficiency. The compression method (SNAPPY) compresses the binary stream, so the use of the compression method is no longer limited by the data type.

**3.5 Access Control and Security**

This Intelligent Manhole Cover System allows not only users but also visitors to access the application and check the positions and states of manhole covers. However, for visitors or average users, they can also check whether a manhole cover is missing or offset. This system is also planning to provide services to the staff of the sewerage administration. They can access the data transmitted from sensors. So the first to do is to justify whether the user is a staff member, then to decide if he can access more data.

As for the authentication mechanism, there are also some more efficient approaches other than user/ password authentication identification and SMS authentication, such as counting the number of special users to avoid illegal access. Since the number of staff of the sewerage administration is constant (the government maintain the number and even detailed information of each staff member, and those data must be updated automatically and thus can be regarded as constant), the system can grant those staff “staff” identifications only during their work hours. If any of those identifications granted are all online, any other “staff” access will be denied.

**3.6 Global Software Control**

The control flow of this system is periodic. Once the user login, the application start working until the user choose to logout or when it runs in the background for a long time, the system will automatically end it for less power cost. This system does not include manual data exchange, any data update operations is done by the server and cloud services, so no synchronization occurs during the interaction with the user. If more than one sensors on the same manhole cover activate alarm, message will be sent only once. Some items in Settings may change the rules of applications and once it happens, refresh will be implemented at once, and thus other subsystems can respond to some conditions.

**3.7 Boundary Conditions**

Once the anomalous state is detected, an alarm message will be sent to the user first, then the map updates to show which one is anomalous and highlight the icon on the map with another color like red. The map will be updated once per 5 seconds. For average users, the icon changes its color only when the manhole cover is missing or offset, while for the staff it will also change its color to yellow if data from other sensors is out of safe range. Another thing is the content of the alarm message shown. For average users, they can get alarms only if the position of the manhole cover changes, while the staff will receive alarms from other sensors.

**4. Subsystem Services**

**4.1 User Authentication**

This subsystem is responsible for rejecting illegal users, recognizing legal users and granting them proper authorities. At least two types of login are provided: username\password and SMS. There should be two input forms, username and password, for the first type. For the second type, two input forms for phone number and message code. If the username or phone number does not exist, page jumps to register page. If password or message code is incorrect, an alarm will be shown at the bottom. Otherwise, it’s a legal user and then the system will detect if it’s a staff member or average user. Thus the services provided include at least:

Login

Register

Automatic SMS function

Credential checking (for password and message code)

User class confirm

**4.2 Map Display**

This subsystem shows the top view of the user’s surroundings. It needs request of user’s locating service, map display, zoom in and zoom out:

User locating service

Map zoom in and zoom out

Map display scope

**4.3 Manhole Cover State Update**

This subsystem needs to search for manhole covers within specific region (the map displayed in last subsystem), get information of those covers and finally update the map with these datasets. Make sure that the user class confirm service in User Authentication subsystem plays an important role here. Services in this stage provided are:

Access cloud server

Update information of each manhole cover periodically (server)

Get datasets from the cloud server

Draw icons on the map

**4.4 Alarm and Maintenance**

In this subsystem, an alarm message will be displayed with proper information, and the staff member sends a request to the server, and the server will update the manhole cover according to that request. Thus, the subsystem needs to be able to do the following:

Send alarm messages

Send a request

Update after receiving requests (server)