The alleviation of terrain pollution: A web-based Android App contributing to terrain recovery

Make Tao1 *, Qiwei Liu2 *

1 Central South University, Changsha 410000, Hunan province, China
2 Central South University, Changsha 410000, Hunan province, China
*Corresponding author’s e-mail: 8208181411@csu.edu.cn
*These authors contribute equally

Abstract. To present the terrain’s pollution circumstances nationwide more directly and contribute to our further recovery solutions, we design an Android app to provide convenience for both clients and regulators. For clients who are using their land and want to find solutions for their possibly polluted land, our app can visualize their land information and they can submit recovery request. And for regulators who design solutions, our database can visualize all the requests for recovery and help them decide the most urgent request. In our work, we first design the web front-end of our app on Android Studio (a platform for developing apps), which mainly consists of three parts, the register and login part1, the information altering part2 and the request submit part3. Part1 enables clients to register and login by entering their username and password, Part2 allows clients to update their land information in the database, Part3 can record each client’s request (consists of land location, request time and pollute-rate) and save it at the database in web back-end. Then, we use mysql to construct our database in the back-end. Our database contains three tables, table1 for user information (like the username, password), table2 for land information (like the land location, land moisture, land PH, land heavy metal amount), and table3 for recovery requests. Finally we use some functions in JSP (a programming language for dynamic web application) on IntelliJ IDEA (JetBrains program platform) to connect the front-end and back-end to exchange the information. After the designing work, we put our app into practice in lab condition. Most feedback are positive, but crush problems happen sometimes, which inspires us to make further optimization. Therefore, our system has a broad application prospect.

1. Introduction
Recent years have witnessed the increasing number of polluted land nationwide in China. In the report of 2016 [1], the number of land that have different extent of pollution (like the mild contamination, the moderate contamination, and the severe contamination) shares 2.23% of the total available land. Also among these polluted land, over half of them are severely polluted. Despite the comparatively small proportion, China’s total amount of lands for cultivation is 1,432,960 square kilometers [2], thus the amount of polluted land is actually quite large, which will reduce the agriculture yield and affect other good-condition land due to the mobility of pollution.

Except the large number of polluted land, the cases of pollution are also complicated [1]. Nowadays we have over 50 different kinds of pollution in the land, each kind has different amount per land in average. Among these pollution, the amount of mercury pollution and acid pollution are biggest.
According to the severe circumstance of China’s polluted land, it’s urgent to find recovery solutions for these polluted land. The recovery plan for a specific kind of pollution is easily designed with the help of chemical methods, however, how to assign the appropriate plan for each land with various pollution, also given the vast number of polluted land in China, is a realistic problem. Then, we introduce our app to solve this problem. With our web-based app, farmers can easily obtain the information of pollution in their land, and government officials or company regulators can provide corresponding recovery plan for each request.

Also our app is easy to deploy and use, which makes it adaptive for wider range of clients. Therefore, our app can be a feasible solution for polluted land if applied widely in agriculture and industry.

2. Method

2.1. The architecture of our work
Our work can be divided into four parts: front-end, back-end, connection, and deploy. Front-end work is to design interfaces for app (the detail description is in 2.3), back-end work is to design tables to store our information (the detail description is in 2.4), connection work is to connect the front-end and back-end (the detail description is in 2.5), deploy work is to apply our app into real-world cases (the detail description is in 2.6). The architecture is shown at Figure 1.

![Figure 1. The architecture of our work](image)

2.2. Data preparation
We first collect 9 pieces of land information from the 2018 soil test results of the Hunan Longzhou Chiyu Technology Limited Company, each representing one specific district in Changsha. Then we contact with the sub companies of Chiyu Tech in nearby cities and towns and collect other 6 pieces of land information. So totally we have 15 pieces containing information in Changsha and nearby cities, each piece of land information consists of land position, land characteristics and different kinds of heavy metal indexes. Then we examine our data and carry out the soil survey and test on some spots in Changsha to complete some missing values in our data. Finally we reorganize our data of land information into the format of table 2 in the database.

2.3. Front-end
We design four interfaces for our app’s outlook.
2.3.1. Interface1 is for clients to login and register, for register function, we use a form in the Android Studio to allow clients to enter their username, password, true name, ID number, the position of land that they possess, and the number of land they possess in corresponding land position. All these user information will be transported to back-end simultaneously. For login function, clients enter username and password, and we compare password with our database, then allow or reject the clients to login.

2.3.2. Interface2 is to search the information of client’s land in the table2 in the back-end and show the land information in the app, the information includes land position, land number, pollute-rate and pollute-label ( the method to compute pollute-rate and pollute-label will be described in 2.4 ). And client can submit their recovery request in this interface to the back-end if they think the pollute-extent is severe, the request contains information about the client’s ID number, the land position, the pollute-rate and the submit time. We will save the requests in table3 in the database for regulators to see.

2.3.3. Interface3 is a form that allows clients to reenter the land position and land number if they enter wrongly when they register.

2.3.4. Interface4 is another form that clients can enter land characteristics and heavy metal index according to the test results. When clients find the information of their land is missing, outdated or impractical, they can make some test on their own land in real-world to get the updated land characteristics and heavy metal index, then enter the test results to update the land information in the app. We will update in the database as well.

2.4. Back-end

2.4.1. We design three tables in the database to store important information. The first table stores the user information that we collect when users register. The format of our first table is shown at Table 1.

| Username | Password | True name | ID number | Land location | Land number |
|----------|----------|-----------|-----------|---------------|-------------|

2.4.2 The second table stores the land information. The format of our second table is shown at Table 2 and we have loaded our original 15 pieces of land information in 2.2 into the second table. And clients can alter their land information in Interface2.

| Land location | Density | Moisture | PH | Iron | Copper | Aluminum | Mercury | Lead | Zinc | Gold | Silver |
|---------------|---------|----------|----|------|--------|----------|---------|------|------|------|--------|

Then we introduce a method, Nemerow Index [4], to compute the pollute-rate and pollute-label. In this method, we use 8 heavy metal indexes, from iron to silver, to compute the pollute-rate. We calculate the average and the maximum of the 8 indexes as Pave and Pmax. Then use the equation in Figure 2 to compute P as pollute-rate. In the process, we select the heavy metal label corresponding to Pmax as our pollute-label ( like iron ). Also we measure the pollute-extent by introducing a standard in Nemerow Index ( The standard is shown at Table 3 ). Then we transport the three values we compute, pollute-rate, pollute-label and pollute-extent, to Interface2 in front-end to display.
Figure 2. The equation of Nemerow Index

\[
P = \sqrt{\frac{P_{ave}^2 + P_{max}^2}{2}}
\]

Table 3. The standard to measure pollute-extent

| Pollute-rate \(( P )\) | \(P<=0.7\) | \(0.7<P<=1\) | \(1<P<=2\) | \(2<P<=3\) | \(P>3\) |
|--------------------------|-----------|-------------|-------------|-------------|---------|
| Pollute-extent           | Clean     | Relatively Clean | Mild Pollution | Moderate Pollution | Severe Pollution |

2.4.3. The third table stores clients’ requests. The format of third table is shown at Table 4. We also sort the third table in the order of pollute-rate and submit time to put the most urgent request to the top. Then we allow the regulators to access the third table and see the requests in the order of the extent of urgency, which contributes to the further recovery plan.

Table 4. The format of request in third table

| ID number | Truename | Land location | Land number | Submit time | Pollute-rate |
|-----------|----------|---------------|-------------|-------------|--------------|

2.5. Connect
Initially, we construct a class Conn. In Conn, we define the Internet ip address of our web server and two passages, input_stream and output_stream, to serve as the byte buffer to store our exchanged information temporarily. Then we define a method http to connect our interfaces with database through web server and regulate the way to exchange our information.

There are four kinds of information that we need to exchange, which is to transport the user information in Interface1 to the table1 in the database; the pollute-rate, pollute-label and pollute-extent that we compute from table2 in database to Interface2; the request in Interface2 to the table3 in database; the altered land information in Interface3 or Interface4 to the table2 in the database.

For all these information, we build a thread in JSP first. Then we use http method to transport and receive information.

2.6. Deploy
We invite four volunteers to use their cellphones to connect to our web server’s Internet ip address. Then we start to run our server to link each cellphone. Then our app will be installed into each cellphone automatically and start to run. Volunteers can register, login, submit request and alter information in our app. Except some cases when Internet crushes, most times our app works effectively and fast and receives positive feedback from our participants. We hope to deploy our app to our campus next, then to the national market in the future.

2.7. Optimization

2.7.1. Solve the problem of multiple land possessed by one client
Initially, we think the land to the client are one-to-one relationship, that’s how we build our app structure. However, in real-world cases, one farmer can possess two or more land in any place nationwide.
So we think of optimizations in Interface1 and Interface2 to solve the problem of multiple land. In Interface1’s register function, we require the user to enter their multiple land position and corresponding land number using “#” as the separator, and we save the multiple land position and land number to table1 in back-end. Then in Interface2, we now let the database search each land of the multiple land we save, then compute corresponding pollute-rates respectively, then transport the whole results to Interface2 to show.

2.7.2. Set the time limit for Internet response
Initially, we suppose the network always runs well during our app usage, however one of our volunteers report the problem that our app is stuck sometimes and the interface don’t show anything. In our further discussion, we think the problem might be caused by slow network response, which is led by many other processes taking up the Internet passage. This case cannot be avoided but our clients might get impatient about the blank interface without any response. Thus, we introduce a new function, setConnectTimeout, to set the time limit of network response. If our app cannot receive response from web server in 5 seconds, this function will end the app and report to the client that “Network problem occurred, please try again.” That’s our optimization to the improvement of client’s experience.

2.7.3. Introduce an API to help client decide the land position
We also find the circumstance that some clients don’t clearly know the position of their land, then we introduce an API of GoogleMap [6] to our app to help clients better decide their exact position. We use the API when client register, and clients can click the button to the GoogleMap and see their current location, then they can go back to our app and enter the land position precisely. For the construction of API, we first apply for the private key for GoogleMap on the Google Cloud Platform, then use Intent (an Android embedded class) to skip to the web address of GoogleMap app on the same cellphone, then the client can go to GoogleMap to see their position and go back to our app easily.

3. Results
We show the running process of our app below step by step.

3.1. We start the web server, which is shown at Figure 3.

3.2. We open the app, register a client and database updates, which is shown at Figure 4 and Figure 5.
Figure 5. Database shows the newly registered James

3.3. We login into the Interface2, which is shown at Figure 6.

Figure 6. Interface2 to show the land information of James

3.4. We click the button of change personal info then we alter the land position and land number and database updates, which is shown at Figure 7 and Figure 8.

Figure 7. Reenter the land position and land number
3.5. We return to the Interface2 and show the change of land location and land number. Changed Interface2 is shown at Figure 9. Then we click the button of Kaifu, then alter the land information and database updates, which is shown at Figure 10 and Figure 11.

Figure 9. The changed Interface2 compared with Figure 6
3.6. We click the button of submit request in Figure 9 to submit recovery request and database updates, which is shown at Figure 12.

4. Discussion

4.1. Enlarge our data of land information

In our work, we collect the land information in Changsha and its neighboring small towns, totally we have 15 pieces of land information, each representing one spot. However, if one user who have land outside Changsha wants to find his land information, he/she will find no results. Thus, if we want to apply our app to nationwide land, we first must enlarge our data to include all the nationwide land information. Then we find the website of national land database [3], but due to the limitation of national security, we have to access the database on behalf of our university. We examine part of the land information in the database, however, most of them are outdated, focus on characteristics of land and lack of many pollution index, which is not sufficient for our work.

Then in our further discussion, we don’t need to import all the land information at first, actually we can collect data gradually with the help of our clients. Our app allows the users to upload their land information and update in the database, which means we get clients’ land information in the process of their altering land information. For example, when a farmer does some test on his/her land and get results of analysis, and our database don’t include his/her land information or include outdated information, then the farmer can update his/her land information and we get this updated information at the same time. Therefore, with more clients involved in our app, we can access more land information nationwide.
4.2. More applications besides agriculture
In our original goal, we tend to apply this app to farmers, who are directly attached to their land and have first-hand land information. And the feedback also proves that our app is useful. However, considering the influence of industries’s emission of pollution on the land, we propose our app can include some industries that have society responsibility, to help them address the pollution in their factory land. Thus, our app not only benefits the land for cultivation, but also guarantees the good-condition of some industries land, which contributes to the preservation of all land in China.

4.3. More feasible algorithms to compute the pollute-rate
In our work, we use the algorithm of nemerow index [4] to compute the pollute-rate, which mainly focuses on the index of heavy metal. However, sometimes the unusual characteristics of land can indicate the pollution as well, that’s what we neglect in our algorithm. For example, if the PH value of one land becomes unreasonably low, then it may suffer acid pollution. To make the algorithm consider more factors, we compare some commonly used algorithms, like advanced nemerow index [4] and geoaccumulation index [5]. In theoretical analysis, we find the geoaccumulation algorithm contains most parameters, which means it considers most land factors. In experiment, we use the three algorithms to compute the pollute-rate of Changsha’s 15 pieces of land information, respectively. As we don’t collect the official-released pollute-rate, we can’t judge which algorithm is more precise, but we do know is that the computed three pollute-rates are close to one another, all of which can show the approximate pollution extent of each spot. Part of the comparison result is shown at Table 5.

| Land location                  | Pollute-rate of Nemerow Index | Pollute-rate of Advanced Nemerow Index | Pollute-rate of Geoaccumulation Index |
|-------------------------------|-------------------------------|---------------------------------------|--------------------------------------|
| Yuelu district, Changsha      | 0.841                         | 0.876                                 | 0.822                                |
| Yuhua district, Changsha      | 0.764                         | 0.766                                 | 0.789                                |
| Kaifu district, Changsha      | 1.184                         | 1.098                                 | 1.137                                |
| Furong district, Changsha     | 0.541                         | 0.557                                 | 0.549                                |
| Tianxing district, Changsha   | 0.967                         | 0.987                                 | 0.994                                |
| Wangcheng district, Changsha  | 1.786                         | 1.890                                 | 1.652                                |

4.4. More possible functions in our app
In our app, we allow clients to submit their recovery request, but don’t show anything about the recovery plan. In our further discussion, different plans corresponding to different kind of pollution can be pre-set and attached to different labels. After our app computes the pollute-rate and gives the pollute-label, then we can directly show the plan relevant to that label, which can accelerate recovery work.
Because recovery plans need chemical analysis, we haven’t realized this function, but we expect it in future refinement.

Also, we allow the regulators to access the database directly to find the most urgent request. However, this accession may cause data security problem in the database. In our discussion, we can develop a similar app or a website entrance just for regulators and they must login before they can see the data in the database, also we only show the top 10 lines of requests, which avoids the unnecessary and spiteful access.

4.5. Disadvantages of our data of land information
In our 15 pieces of land information, the scope of land location is too large, which means our land information of one specific spot is not precise enough. For example, like the region of Yuelu district, Changsha covers over 558 square kilometers, and the land circumstance of each edge of this region cannot be the same in such a large scale. So actually we generalize land circumstance to a larger scope in our work. For the data in national soil database [3], it specifies the land location to spots that have around 1 square kilometers. As we don’t have the economic condition and techniques to collect data at such precise spots, we had to make trade-off on the precision of our data.

5. Conclusion
In our work, we develop an Android app to serve for the real-world cases of terrain recovery. First, we design four front-end interfaces. Interface1 is for clients to login and register and transfer the user information to the back-end database. Interface2 is to show the information of all the land that the user possesses. Interface3 is to alter land information, the user can upload new land information and our app will update simultaneously in the database. Interface4 is to submit recovery request to the database.

Then we devise three tables in the back-end database, for user information, land information and request, respectively. Finally we connect the front-end and back-end by JSP transmission to build our terrain recovery system. After the design work, we also implement this app in lab condition, and find some practical problems about Internet connection and real-world cases. We think about relevant optimizations to solve the problems to the most extent. In the future, we believe our app is promising to apply to wider range of land and can contribute to the terrain recovery nationwide from an Internet perspective.

References
[1] The Agriculture Department of China, Report of Quality of Nationwide Land For Cultivation, Dec, 2016
[2] Ministry of Land and Resources in China, Report of Available Land NationWide, 2016
[3] National Soil Science Database, http://vdb3.soil.csdb.cn/extend/jsp/introduction
[4] Q. Zou, Yougan Wang, X.-Y. Du, X.-C. Li, Application of nemerow index method in grade evaluation of PAHs pollution in the oilfield soils, Jan 2011
[5] Deng Feng Wang, Hong Juan Wu, Jie Chen, Shi Xing Jiao, Xue Min Dai, Assessment of Heavy Metal Pollution of Urban Roadside Soil with Set Pair Analysis and Geoaccumulation Index, Advanced Materials Research, June 2010
[6] Google Android develop guidance, https://developers.google.cn/maps/documentation/android-sdk/current-place-tutorial
[7] Bill Philips, Chris Steward, Kristin Marsicano, Android Programming, 2018, The Big Nerd Ranch Guide, Third Edition