Preference system for guided the selection of improvement type and assessment of soil based on expert opinion with weighted system

T B Kurniawan¹, F Usman¹, R C Omar¹ and K A Sanusi²

¹Institute of Energy Infrastructure, Universiti Tenaga Nasional, Kajang, 43000, Malaysia
²Grid Solution Expertise, Grid Division, Tenaga Nasional Berhad (TNB), Malaysia

Corresponding author: fathoni@uniten.edu.my

Abstract. The soil in different regions has many fundamental problems, including a high groundwater table, deformation of highly compressive clays, peat, etc. Most areas of these zones are waterlogged, swampy, and characterised by weak soils, making it difficult to construct them without addressing the stability of such structures. In many cases, soil conditions and soil materials are not ideal for the proposed or planned development. In such cases, the geotechnical engineer will consider how to resolve possible issues with feasible solutions carefully. This paper presents work resulting in the development of software that is used as a tool to guide and select the type of improvement and soil assessment. It allows users to learn about soil improvement and ground modification in engineering practice while providing the user with a tool that helps to solve soil improvement problems. An expert opinion was designed and conducted to obtain the weighted matrix for evaluation selection of improvement type based on specific criteria. A Software Development Life Cycle (SDLC) was followed, and in terms of design, implementation, testing and evaluation were conducted. The evaluation result based on the System Usability Scale was performed and evaluated, and it shows the proposed tool was accepted and suitable to use by the end-user.

1. Introduction
In many cases, soil conditions and soil materials are not ideal for the proposed or planned development. In such cases, the geotechnical engineer will consider how to resolve possible issues with feasible solutions carefully. Modification of soil materials or soil stabilisation may provide a means to achieve the desired goal of ensuring good engineering properties and responses for a variety of applications and conditions [1].

Depending on the initial soil conditions, soil properties, and desired outcomes, the engineer may choose from a wide range of soil improvement and soil stabilisation techniques [2]. It will help solve the challenges of poor site conditions, low soil quality, mitigation of potential problems, or remedial work. The soil in different regions has many fundamental problems, including a high groundwater table, deformation of highly compressive clays, peat, etc. Most areas of these zones are waterlogged, swampy, and characterised by weak soils, making it difficult to construct them without addressing the stability of such structures [3]. The land-based system is as reliable as its foundation. The soil is, therefore, a critical element that influences the success of any structure placed on it. Due to the existence of the soils in
certain regions, fundamental problems such as deformation, accumulation of highly compressible clays, and settlement are inevitable [4]. Based on these problems, the process of recovery of the soil structure is a very critical issue. One good idea is to provide a tool which can be used as guided and preference system to help the engineer to make the recovery of soil structure.

This paper presents work resulting in the development of software that is used as a tool to guide and select the type of improvement and soil assessment. It allows users to learn about soil improvement and ground modification in engineering practice while providing the user with a tool that helps to solve soil improvement problems. The process of selection was developed based on expert opinion method with a weighted system.

2. Literature review

2.1. Soil improvement and grounding modification
Soil is a broad term used to describe loose deposits produced from underlying rocks by physical, synthetic, and natural processes that fluctuate with time, area, and ecological conditions and result in a wide variety of soil properties [5]. For certain areas, soils are fluviatile for nature, consisting of water-logged, overflowing soil, protected by peat. The soil in coastal areas is distinguished by a fluctuating aggregation of sand and clay, with a high concentration of clay occurring about 10 m below the surface of the ground [6]. The soil in these regions has a low load-bearing limit, water-filled waste materials; in this way, their major engineering issues include an abundance of surface and groundwater, poor drainage, high compressibility, low bearing capacity, and differential settlements. It is emphasised that extensive salt leaching by rainwater and groundwater harms soil structure [7].

Improvement of soil is any method of modification of unsuitable in situ or borrowed soil to improve the selected engineering characteristics at a lower cost and with improved quality control than can be accomplished by removing, bridging, or moving through unsuitable materials [8]. From the above definition, the inference is drawn that the improvement of the soil is the modification of existing site-based soils to provide better performance under design or operational loading conditions.

For some projects, soil improvement is needed to optimise the quality of the natural soil for a given construction purpose. Modification of the soil increases strength, bearing capacity and resistance to deteriorating of the forces of nature and the human-made climate. Decreases the pattern of volume change tendency, settlement, controls permeability and provides long-term permeability over decades of operation under harsh environmental conditions [9]. Improvement of the soil improves the soil’s engineering properties to allow field construction to take place on weak soils. The long-term performance of structures, whether roads, buildings, slopes, embankments, or any other arrangement, depends on the soundness of the underlying soils; unsuitable soils can create significant problems.

2.2. Web-based application
A web application is a computer program that utilises web browsers and web technology to perform tasks over the Internet [10]. Web applications use a combination of server-side scripts (PHP and ASP) to handle the storage and retrieval of the information, and client-side scripts (JavaScript and HTML) to present information to users. The web application allows users to interact with the company using online forms, content management systems, shopping carts and more. Besides, the applications allow employees to create documents, share information, collaborate on projects, and work on common documents regardless of location or device. Figure 1 shows how the web-based application works.

The app can be written in various programming languages and make use of multiple technologies and frameworks. A web-based application will run on the client browser of the computer no matter what operating system is installed. It makes web-based apps one of the universal cross-platform solutions available today.
2.3. **ASP.NET Core MCV**

ASP.NET Core, built by Microsoft and the community, is a free and open-source web framework and successor to ASP.NET [11]. It is a modular framework that operates on both the full .NET Framework, Windows, and cross-platform .NET Core. However, ASP.NET Core version 3 works only on .NET Core discarding support for .NET Framework. The architecture is a full rewrite that integrates a single programming model with the previously separate ASP.NET MVC and ASP.NET Web API. It has a high degree of concept compatibility with ASP.NET despite being a new system, designed on a new web stack. ASP.NET Core applications allow side by side versioning where different forms can target different versions of ASP.NET Core running on the same computer. Previous versions of ASP.NET would not allow this [12].

The following diagram shows the three main components and which ones reference the others, as shown in Figure 2.

![Diagram of Model-View-Controller (MVC)](image)

**Figure 2.** The main concept of Model-View-Controller (MVC)

3. **Methodology**

3.1. **Design framework**

The proper selection and assessment of a soil improvement technique for use at a particular site are neither a simple nor a one-off proposal. Local conditions and specificity, as well as expertise and judgment, are integral parts of the decision-making process. In applied engineering training settings, this process leads to treatment using a computer-based design and assessment approach, relying on a
comprehensive database of ground modification techniques and an associated decision-making tool [13].

The objective of the process is to provide tools that can help the engineer to select the most feasible type of improvement based on soil conditions or criteria. The design framework, as shown in Figure 3, shows the process flow of these tools. Figure 3 shows that the user acts as an actor to input the data to the database. They are the category of improvement, the type of improvement, the criteria, and the items of each criterion.

![Figure 3. Basic design framework](image)

3.2. Soil improvement weighted matrix

Based on the data entered in subsection 3.1, the soil improvement matrix was built as Figure 4. A weighted value is needed for each type of improvement crossed by the selection of each item.

![Figure 4. Soil improvement matrix](image)
The weighted value is set to a floating-point between 0 and 1 \([0 – 1]\). It was using the Normalisation Formulation, as shown in Equation 1, to normalise the weighted value. Normalisation is essential as no information that is more important than others for the type of improvement or selection.

\[
x_{\text{normalized}} = \frac{x-x_{\text{minimum}}}{x_{\text{maximum}}-x_{\text{minimum}}}
\]

(1)

3.3. Design model evaluation
An approach of expert opinion evaluation for soil improvement selection was designed and conducted as Figure 5, below. Figure 5 shows the weighted matrix to be inputted by an expert. Each selection item has a few weight values for each selection improvement type. In this experiment, we submit the questionnaire to ten experts in the expertise area of soil improvement and ground modification. All experts return the survey, and the results were analysed and summarised.

![Figure 5. Soil improvement questionnaire for experts opinion evaluation](image)

3.4. Weighted evaluation
Each response from the expert is analysed and examined for its weighted. Whether it is completed or not, and the range of value is check. The result is also compared with other result response from other experts. Each column of improvement type is first converted using the normalised formulation. It was mentioned in subsection 3.2. After that, from all result, we calculate the average value for each item. Then the value will be used as input in the tool later.

4. Implementation

4.1. Software development life cycle – extreme programming
The tool is developing based on ASP.NET Core MVC. The development process was followed by a Software Development Life Cycle (SDLC), Extreme Programming (XP), one of the Agile method. The basis of the whole XP software development is code-writing and testing [14]. XP methodology is primarily designed for smaller teams with two to ten members, who work on frequently or less known
assignments. Projects which take long or have difficulties in getting feedback (e.g., from a technological point of view) are not suitable for this methodology. Automatic testing or version assembly is necessary for its implementation. XP is a flexible Agile methodology emphasising the interconnection of the proposal and implementation stages.

XP practices Test-Driven Development (TDD), pair programming, short interaction, team code ownership, and acceptance tests. The primary activities in XP are [15]:
- Planning and Managing
- Designing
- Coding
- Testing

The design processes were carried out and shown in Section 3, which consists of the basic design framework, soil improvement matrix, model design assessment, and weighted assessment. Next, the coding and testing process will be discussed in the following section.

4.2. Coding
This web-based application has been built with the programming of ASP.NET Core MVC. The MVC has a template for the Visual Studio 2019 Community Version. The flow process of the tool was built on this template. Figures 6 and 7 display the Visual Studio 2019 editor with the MVC template and the main user interface (UI) of the application. The tool is complete with the user management system and the task of granting the right of access to the authority of each user. The other facilities are master data management, which helps the user to control categories, data type improvement, selection criteria, and data item selection, as shown in Figure 8 as an example. It is designed to allow users can easily to manage: add, edit, and remove data. The user does not need to change the coding if the number of data has been changed.

Based on the design in subsection 3.2 on Soil Improvement Weighted Matrix, Figure 9 shows the implementation part. The user only needs to enter the value and then move the cursor to another cell, and the data will be stored in the database. The user will notice that since the value will be changed in bold formatting and it will be shown in different colours. Zero value means that the type of improvement is not suitable for that selected item so that the type of improvement can not be included in results.
Figure 7. The main user interface (UI) of the program

Figure 8. The category list maintenance module

Figure 9. The soil improvement weighted matrix

The user can be used the tool as a preference system to select the proper improvement type by selecting the [Selection Wizard] menu. First, the user asks to create a new project to identified and record some project information as documentation of the program. Then the user needs to input some information, such as project name, location, and data of data collection. After that, the tool shows some
selection criteria to select by user. Based on the user selection, the feasible improvement type of soil will be displayed by their rank values. The calculations are based on weighted was inputted previously in Soil Improvement Weighted Matrix, based on expert opinion, as shown in Figure 10.

4.3. Testing

Testing has many advantages, and cost-effectiveness is one of the most significant. Testing in your project can save money in the long run. Software development consists of several stages, so if bugs are found in earlier stages, it costs a lot less to repair them. That is why it is crucial to get the testing completed as soon as possible.

There are two kinds of testing in web-based development or other software development, white-box testing and black-box testing. The white-box testing is considering the processes in the coding part; on the other hand, the black-box testing only considers the input and output, not include the process of the program.

![Selection Wizard](image)

**Figure 10.** The selection wizard module

In ASP.NET Core MVC, the white box testing can be achieved by using unit testing for the automatic testing procedure. A unit test is meant to test a single part of our application logic. It runs entirely in memory and process which does not communicate with the file system, the network, or a database. Unit tests should only test our code. Due to these reasons, unit tests should execute exceptionally quickly, and we should be able to run them frequently. Ideally, we should run unit tests before each change is pushed to the source control repository and also with every automated build on the build server.

There are 63 unit tests built and running for this tool, and Figure 11 shows the unit testing after complete running and show the results. The figure shows that all unit testing was executed, and all results status show a success. The black-box testing can be done using user interface (UI) testing. UI testing is a testing technique used to identify the presence of defects is a product/software under test by using a graphical user interface (GUI). One of the excellent tools for UI testing is Selenium, which we used in this tool development.
Selenium is a portable web application testing framework. Selenium offers a playback method for conducting practical experiments without the need to learn a software scripting language. It also provides a test domain-specific language (Selenese) for writing tests in several popular programming languages, including C#, Groovy, Java, Perl, PHP, Python, Ruby, and Scala. The testing can then be run against most modern web browsers. Selenium is running on Windows, Linux, and macOS, which is based on an open-source software released under the Apache License 2.0. Figure 12 shows the Selenium can be used as a UI testing in our tool development.

Figure 11. The selection wizard module

5. Evaluation and discussion

5.1. Evaluation matrix
The UI evaluation was conducted to evaluate the user usability of the tool. In this research, the System Usability Scale (SUS) is used. In 1986, John Brooke, then working at DEC, developed the System Usability Scale (SUS) [16]. The standard SUS consists of the following ten items (odd-numbered items worded positively; even-numbered items worded negatively).

Based on ten items from the SUS standard, we define a list of questionnaires to the end-user to give the evaluations. The questionnaire obtained from [16] was used. A matrix was constructed to evaluate the result from the end-user response, as shown in Figure 13. Three aspects will be measured, the adjective rating, the grade scale, and the acceptability ranges. Each element has a variety of evaluations.
5.2. Result and discussion

There are 25 of end-users surveyed, and the response results were analysed and evaluated. Table 1 shows the final results of end-user responses.

Table 1. Final value evaluation results.

|    | \( \sum_{\text{evaluator}} x 2.5 \) | Total |
|----|------------------------------------|-------|
| R1 | 35 x 2.5                           | 87.5  |
| R2 | 35 x 2.5                           | 87.5  |
| R3 | 37 x 2.5                           | 92.5  |
| R4 | 36 x 2.5                           | 90    |
| R5 | 36 x 2.5                           | 90    |
| R6 | 38 x 2.5                           | 95    |
| R7 | 34 x 2.5                           | 85    |
| R8 | 37 x 2.5                           | 92.5  |
| R9 | 34 x 2.5                           | 85    |
| R10| 37 x 2.5                           | 92.5  |
| R11| 35 x 2.5                           | 87.5  |
| R12| 35 x 2.5                           | 87.5  |
| R13| 34 x 2.5                           | 85    |
| R14| 35 x 2.5                           | 87.5  |
| R15| 34 x 2.5                           | 85    |
| R16| 34 x 2.5                           | 85    |
| R17| 30 x 2.5                           | 75    |
| R18| 33 x 2.5                           | 82.5  |
| R19| 32 x 2.5                           | 80    |
| R20| 35 x 2.5                           | 87.5  |
| R21| 35 x 2.5                           | 87.5  |
| R22| 35 x 2.5                           | 87.5  |
| R23| 36 x 2.5                           | 90    |
| R24| 36 x 2.5                           | 90    |
| R25| 34 x 2.5                           | 85    |

Average \( \frac{21805}{25} = 87.2 \)
Based on Table 1, the result shows the average value is 87.2. From the perspective user evaluation, the tool is suitable for SUS evaluation, as shown in Figure 13:

1. From the adjective rating aspect, the result value is showing in the excellence category.
2. From the grade scale aspect, the result value is showing in group B.
3. From the acceptability aspect, the result value is showing in an acceptable group.

The result evaluation shows that the proposed tool was accepted and suitable by the end-user and can be used without any modification or redesign and re-implementation.

6. Conclusion

From the design process, the implementation, and finally, the evaluation process, the proposed tool can be used by the end-user. The evaluation result shows that the tool was accepted and suitable to use by the end-user. The improvement can be proposed to enhance the tools by adding other facilities and functionality, such as settlement process, guide process to improve the soil development as an interactive approach, etc. These proposed other improvements to the tool can be made in future research.

References

[1] P. G. Nicholson, 2014. Soil Improvement and Ground Modification Methods.

[2] L. Soto, M. Galleguillos, O. Seguel, B. Sotomayor, and A. Lara, Jan, 2019. “Assessment of soil physical properties’ statuses under different land covers within a landscape dominated by exotic industrial tree plantations in south-central Chile,” J. Soil Water Conserv., 74, pp. 12–23.

[3] I. Virto, M. J. Imaz, O. Fernández-Ugalde, N. Gartzia-Bengoetxea, A. Enrique, and P. Bescansa, 2015. “Soil degradation and soil quality in Western Europe: Current situation and future perspectives,” Sustain., 7, no. 1, pp. 313–365.

[4] A. S. B. and J. G. W. T., 2019. “Reliability of Soil and Ground Improvement Techniques on Peaty Clay Soil – A Review,” Int. J. Trend Sci. Res. Dev., 3, no. Issue-3, pp. 682–690.

[5] J. K. Mitchell and K. Soga, 2005. Fundamentals of Soil Behavior, 3rd Editio. John Wiley & Sons Inc.

[6] M. Long et al., Jun, 2019. “Site characterisation and some examples from large scale testing at the Klett quick clay research site,” AIMS Geosci., 5, pp. 344–389.

[7] B. N. Olorunfemi, 1984. “Factors influencing the geochemistry of lateritic soils in the greenish-chlorite-amphibolite meta-tectonic belt, southwestern Nigeria,” Chem. Geol., 42, no. 1, pp. 261–275.

[8] B. M. Das, Ed., 2011. Geotechnical Engineering Handbook. J. Ross Publishing Inc.

[9] K. Onyelowe, Jan. 2012. “A COMPARATIVE REVIEW OF SOIL MODIFICATION METHODS,” 1

[10] A. M. Qadir and P. Cooper, 2020. “GPS-based Mobile Cross-platform Cargo Tracking System with Web-based Application,” in 2020 8th International Symposium on Digital Forensics and Security (ISDFS), pp. 1–7.

[11] J.-H. Xu, 2017. Practical Cross-Platform Charts and Graphics for ASP.NET Core MVC.

[12] G. Krasner and S. Pope, Jan.1998. “A cookbook for using the model - view controller user interface paradigm in Smalltalk - 80,” J. Object-oriented Program. - JOOP, 1

[13] S. Sadek and G. Khoury, 2000. “Soil and Site Improvement Guide: An Educational Tool for Engineered Ground Modification,” Int. J. Eng. Educ., 16, no. 6, pp. 499–508.

[14] M. Alshayeb and W. Li, 2006. “An empirical study of relationships among extreme programming engineering activities,” Inf. Softw. Technol., 48, no. 11, pp. 1068–1072.

[15] R. Fojtik, 2011. “Extreme Programming in development of specific software,” Procedia Comput. Sci., 3, pp. 1464–1468.

[16] J. Brooke, 1996. “A ‘Quick and Dirty’ Usability Scale,” in Usability Evaluation in Industry, B. Jordan, P.W., Thomas and B. A. Weerdmeester, Eds. McClelland (eds.), pp. 189–194.