UDC 519.816

DOI: 10.15587/1729-4061.2019.168163

DEVELOPMENT OF A MOBILE DECISION SUPPORT SYSTEM BASED ON THE SMART METHOD FOR ANDROID PLATFORM

D. Horpenko*
E-mail: dieznotefeedback@gmail.com

N. Volkova
Senior Lecturer*
E-mail: volkovanp30@gmail.com

M. Polyakova
Doctor of Technical Sciences, Associate Professor*
E-mail: marina_polyakova@rambler.ru

V. Krylov
Doctor of Technical Sciences, Professor*
E-mail: viktor.kryilov@gmail.com

*Department of Applied Mathematics and Information Technologies
Odessa National Polytechnic University
Shevchenka ave., 1, Odessa, Ukraine, 65044

Робота присвячена розробці мобільної системи підтримки прийняття рішень для розв'язання багатокритеріальних задач. Для забезпечення автономної роботи системи запропоновано використовувати трьохрівневу архітектуру. Дана модель, для забезпечення повторного використання і поширення коду, реалізується у вигляді трьох рівнів прикладного коду: рівня представлення, рівня логіки застосунку та рівня даних.

Розробка рівня логіки застосунку в розробленій системі підтримки прийняття рішень передбачає створення підсистеми прийняття рішення, підсистеми взаємодії з базою даних і підсистеми управління повідомленнями. В основі підсистеми прийняття рішень в розробленій мобільній системі був обраний вдосконалений метод Smart. Цей метод відрізняється від класичного методу тим, що особа, яка приймає рішення, використовує елементи матриці рішень в якості оцінок кожної альтернативи за всіма критеріями. Також враховується характер дії над критеріями (максимізація або мінімізація), з урахуванням якого відбувається нормуванням елементів матриці рішень.

Запуск роботи підсистеми взаємодії з базою даних, яка відповідає за передачу і вилучення даних в/з бази даних, відбувається за допомогою інтерфейсу користувача. Для створення бази даних використовувалася система управління реляційної базою даних SQLite. SQLite зберігає всю базу даних (включаючи визначення, таблиці, індекси і дані) в одному стандартному файлі на пристрої, на якому виконується програма. Підсистема управління повідомленнями дозволяє особі, що приймає рішення, відправляти результати розрахунків через мережу Internet або за допомогою служби коротких повідомлень (SMS). Мобільна система підтримки прийняття рішень розроблена на мові Java в Android Studio 3.2.1. В якості використання розробленої мобільної системи розглядалася задача покупки смартфона

Ключові слова: мобільна система підтримки прийняття рішень, особа яка приймає рішення, альтернатива, критерії, метод Smart, трирівнева архітектура, Androi

1. Introduction

The multi-criteria decision-making tasks arise in such areas of human activity as management, politics, social psychology, commerce, and economics. These tasks are, for example, choosing a candidate for the position, choosing the best place for construction, choosing enterprise development strategies, etc. It is necessary to compare different alternatives that are characterized by different criteria to make decisions in such situations. Often, the evaluation of the alternatives may be better by some criteria and be worse by other criteria. Therefore, for solving multi-criteria decision-making tasks, Decision Support Systems (DSS) based on mathematical methods are used [1].

In [2], there are two groups: enterprise-wide DSS and desktop DSS among the DSS. Enterprise-wide DSS is a data-driven decision support system that supports a large group of managers in a networked client-server environment with a specialized data warehouse. Desktop DSS is a small system that runs on a decision maker's (DM's) personal computer. If access to the DSS functionality occurs with web-technologies help, then it is a Web-based Decision Support System (WB-DSS) [3]. WB-DSS are on the server and many decision makers have the ability to work with DSS. The developers of some DSS, such as Decide IT, Expert Choice, Decision Lens, have developed web versions of their systems. The mobile DSS are appropriate to use, if the DM needs to get a solution at any time in off-line mode. Mobile computing devices productivity increases with the development of mobile technology. At the same time, these devices are very compact, which makes them more mobile. Thus, mobile computing devices are an ideal platform for creating new types of decision support systems [4].

However, the problems of the mobile DSS development and usage are:

– ensuring autonomous system operation in case of internet connection unreliability between the client and the server;

– compatibility of the mobile DSS with platforms of specific mobile devices.

Modern DSS, which are based on various mathematical methods, can replace the human resources at the data

Received date: 29.04.2019 Accepted date: 08.05.2019 Published date: 28.06.2019

processing and structuring stage. However, nowadays, the decision maker often needs to perform a comparison of the solution options considering several criteria simultaneously remotely, in off-line mode. With the increasing popularity of mobile devices and adaptation of information technologies to them, there is a need to create the mobile DSS to ensure the decision makers work in off-line mode. Thus, the creation of a mobile DSS for the mobile OS and based on the non-resource-intensive decision-making method for increasing the efficiency of the decision-making process is relevant.

2. Literature review and problem statement

The architecture of the system, that is, the nature of its main components interaction is one of the DSS classification features. The DSS includes: a database, a model base and a software subsystem, which consists of database and model base management systems, a user interface, analytical tools, communications infrastructure and networks [5]. The main mobile applications architecture is the client-server architecture. As a rule, mobile devices are thin or fat clients in the client-server mobile systems [6]. Fat clients can be implemented using one, two or three layers of application code. Thin clients do not have a custom application code. Their functionality is fully provided by the server on which the database is located and the whole application code is implemented. The client sends the entered data for processing to the server, where it is processed and stored when connecting to the Internet. Modern DSS process a large data volume, therefore, cloud servers are used for storing and processing data. Access to cloud servers depends on the connection to the Internet. In [7], the issues of application of multi-criteria decision analysis methods in an offloading operation were considered. The main aim of an offloading operation is to improve the productivity of mobile devices through the use of cloud resources. Theoretical and practical issues of using cloud computing services and cloud data storage are considered in [8]. The cloud computing model suggests network access to cloud computing resources.

In [9], to build mobile DSS the authors have chosen the thin client model, where the client is mobile devices, and all heavy computations are performed by the server. Technologies for developing a scaled and flexible information system for small and medium-sized businesses, and an architecture for building a server were considered. In [10], the three-layer client-server architecture was chosen to develop a mobile DSS for selecting a higher education institution. The database server, the application server and the administrator's PC are connected by a high-speed local area network. The client is a mobile device, connected to the application server via the Internet.

In [11], existing cloud applications were considered for DSS, which are used to support decision-making for rice crop yield and the mobile DSS for farmers was developed. The ThingSpeak.com service was used, which allows collecting, analyzing and processing data, obtained through URL requests. In [12], the mobile information system is presented for decision support during diagnostics and self-monitoring of glycemic status by patients with diabetes. Cloud technologies were used to store patient data and doctor's access to them. Wherein, separate mobile applications for the patient and for the doctor were developed. Cloud services of Firebase were used. In [13], a mobile cloud-based clinical decision support system for drug poisoning in children was developed. A client-server architecture was chosen and cloud technologies were used. The mobile devices or personal computers are as a client. Amazon Cloud infrastructure is used to deploy the application server and data server. In [14], a mobile DSS was developed to help doctors in diagnosis, therapy, searching for information and image analysis. The system consists of 53 information databases and 119 guidelines, including clinical imaging guidelines, developed by the Korean Society of Radiology. A real-time NoSQL database management system of data of Firebase was used to create a data warehouse.

The disadvantage of the client-server architecture is the need of constant communication with the server that is the source of the update and retrieval of data.

To ensure the autonomous operation of the system or in the case of unreliable communication between the client and the server, it is preferable to consider the mobile device as a stationary device on which the application can be deployed. It is necessary to ensure the possibility of modifying individual components of the system, when developing an application for a mobile operating system. The three-layer architecture provides such a possibility. In [15], a mobile application has been developed for selecting the most convenient underground mining method for a mine using multi-criteria decision-making methods, such as TOPSIS, VIKOR, ELECTRE, FMADM and PROMETHEE. The mobile application has been developed on the basis of the three-layer architecture. The application has a user-friendly interface and is designed for users with different levels of training. Components of all application levels: presentation level, application logic level and data level are on a mobile device. This allows the decision maker to use this application for the selection of an underground mining method on mobile devices anywhere without Internet access.

When developing mobile applications, it is also necessary to decide on the platform for which this application will be created; programming language for the selected platform and development environment [16]. Android and iOS platforms are the most popular mobile platforms, today. The iOS platform is a platform only for mobile devices, which are developed by Apple Inc. (USA). The Android platform is the platform for the vast majority of mobile device manufacturers. The most famous companies are: Samsung Electronics (Korea), HTC (China), LG (Korea), Motorola (USA), Huawei Technologies (China), Aser (China), Sony (Japan), Asus (China), Lenovo (China), Xiaomi (China), OnePlus (China), Meizu (China). So mobile DSS in [10–15] have been developed for mobile devices on the Android platform. These applications have been developed to support decision making in specific application areas. The authors determined the criteria for evaluating alternatives in advance. The disadvantage of working with such systems is their narrow specialization.

Furthermore, in [17], the authors have indicated some limitations of using the mobile devices. These limitations are as follows: small screen size, the difficulty of manual data entry, the ability to display only one window, the difficulty to navigate between windows. These limitations must be considered when designing mobile DSS and choosing methods for building a decision rule.

Thus, the analysis of the developed DSS [10–15] revealed the following disadvantages:

– use of the client-server architecture with a thin client model in the mobile DSS, which requires constant communication with local or cloud servers via the Internet;

– most of the mobile DSSs were developed for specific application areas, which makes it impossible to apply such DSS for other purposes.

Analysis of scientific publications [10–15] found out that today DSS are used in various areas of human activity, such as medicine, agriculture, mining, the choice of university. Data processing and storage are performed on local or cloud servers, which are only accessible through an Internet connection. However, the problem consists of the necessity to make a decision in a number of tasks here and now regardless of the Internet connections. It is necessary to take into account that the problem of choice can arise in any subject area.

3. The aim and objectives of the study

The aim of this work is to develop a mobile DSS on the Android platform, using a three-layer architecture. This will allow the decision maker to receive decision support anytime, anywhere without access to the Internet. This architecture contains a presentation layer, an application level and a data layer.

To accomplish the aim, the following objectives have been set:

– to develop a decision-making subsystem based on the improved Smart method;

– to develop a user interface and message management and database interaction subsystems;

– to create a database using the SQLite relational database management system;

– to perform a product software implementation of a mobile decision support system with a three-layer architecture;

– to experimentally research the developed system on the example of the task of buying a smartphone.

4. Development of a decision-making subsystem based on the improved Smart method

For developing the mobile DSS, the three-layer architecture is used: presentation layer; application layer; data layer. This mobile DSS is an extended version of the mobile application, presented in [18], which was implemented as a two-layer architecture. There was no data level in this mobile application. The user entered data, received structured solution, after which data have been clean. The block diagram of the developed mobile DSS (Fig. 1) represents the subsystems DSS interaction in formulating recommendations in the decision-making process. All subsystems are located on the mobile device. This allows the decision-makers to receive decision support in off-line mode, anytime and anywhere.

During the development of the decision-making subsystem, we note that most of the existing DSS allow to solve multiple criteria decision-making problems in the following formulation: we have the set of n alternatives $A_1, A_2,..., A_n$ and N criteria $C_1, C_2,..., C_N$, used to evaluate alternatives. Each alternative has an estimate by each criterion, received from experts or obtained on the basis of objective calculations. It is necessary to build a decision rule based on the decision-maker preferences, which will allow ranking alternatives by quality and finding the best alternative.

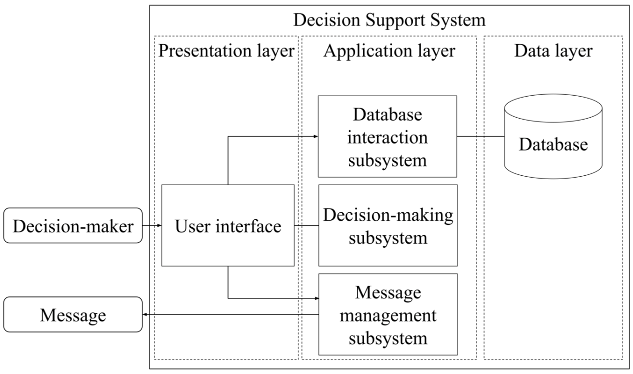


Fig. 1. Block diagram of the mobile DSS

The development of a decision-making subsystem involves using not resource-intensive multi-criteria decision analysis methods. This provides low hardware requirements for DSS. The analysis carried out in [18] showed that the AHP method assumes the filling of the matrix of pairwise comparisons at all hierarchy levels, according to the scale of relative importance. This makes it difficult to work with the DSS based on this method, as there are limitations when working with mobile devices that have been described above. The Smart and Topsis methods are simple, and their calculations are not resource-intensive. But the Smart method is subjective when assigning weights to alternatives and does not allow to take into account actions on criteria (maximize/minimize). Therefore, the Smart method was improved in [18]. The algorithm implementing the improved Smart method is shown in Fig. 2.

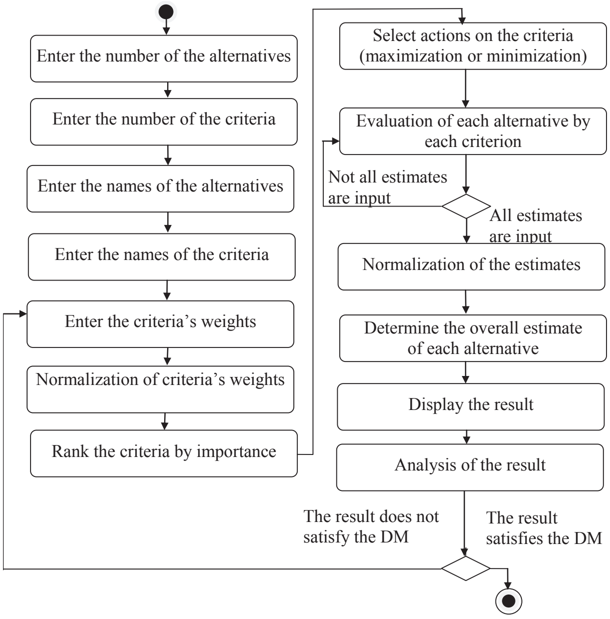

Enter the number of the alternatives
Enter the number of the criteria
Enter the names of the alternatives
Enter the names of the criteria
Enter the criteria's weights
Normalization of criteria's weights
Rank the criteria by importance
Select actions on the criteria (maximization or minimization)
Evaluation of each alternative by each criterion
Not all estimates are input
All estimates are input
Normalization of the estimates
Determine the overall estimate of each alternative
Display the result
Analysis of the result
The result does not satisfy the DM
The result satisfies the DM

Fig. 2. Algorithm of the improved Smart method

Evaluation of each alternative by all criteria on a scale from 0 to 100 points by a decision-maker is one of the Smart method stages. Thus, the evaluation of alternatives by criteria is subjective, and performing this stage is very laborious for the decision-maker when working with a mobile device.

Therefore, it is proposed at this stage to use the decision matrix *X*, as these evaluations. The decision matrix *X* dimension is $m \times n$, where m – the number of alternatives, n – the number of criteria. The elements of the decision matrix are estimations the *i*-th alternative by the *j*-th criterion, $i=1,\ldots,n$, $j=1,\ldots,m$, which are known in advance. The rows of the decision matrix are the alternatives A_i, $i=1,\ldots,n$, and the columns are the criteria C_j, $j=1,\ldots,m$, by which these alternatives are evaluated, and the elements of the matrix are denoted as x_{ij}, $i=1,\ldots,n$, $j=1,\ldots,m$.

$$X=\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{23} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}.$$

Also, to consider the nature of the action on criteria (maximize or minimize), the stage at which the decision maker explicitly indicates the nature of action on criteria was added.

Furthermore, it is proposed to perform a normalization of the decision matrix elements so that the weight of the relevant alternative for each criterion is taken into account when calculating the generalized assessment of each alternative.

It was proposed to take into account the nature of the action on the criteria – maximization or minimization, when normalizing the elements of the decision matrix. Then for the criteria that are maximized, the normalization is performed using the formula:

$$r_{ij}=\frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^{2}}}, \tag{1}$$

where r_{ij} – the normalized value x_{ij} of the decision matrix *X*.

For the criteria that are minimized, the normalization is performed using the formula:

$$r_{ij}=1-\frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^{2}}}. \tag{2}$$

This allows:

– to reduce the subjective role of decision makers in the process of evaluating alternatives by criteria;

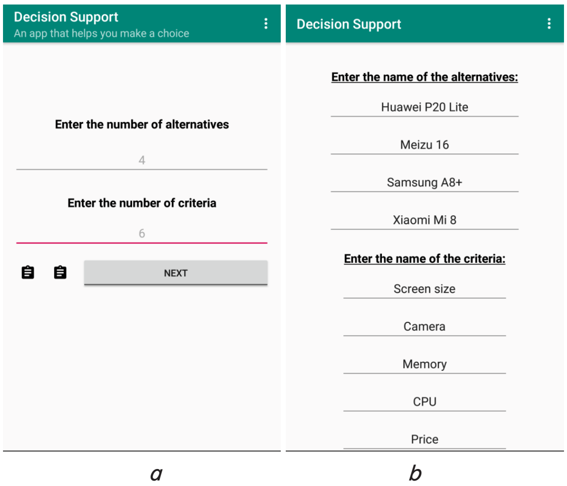
– to consider the nature of the action on the criteria;

– to perform a normalization of the decision matrix so to take into account the nature of the action on the criteria.

As stated earlier, the developed mobile DSS has the three-layer architecture, each level has been implemented in the form of corresponding classes. The decision-making subsystem belongs to the application layer of the DSS.

5. Development of a user interface and message management and database interaction subsystems

Interaction between the mobile DSS and decision makers occurs through the user interface, which belongs to the presentation layer of the DSS. The user interface is responsible for the input of the necessary data for the problem solution. The number of criteria and alternatives is entered in the first window (Fig. 3, *a*). The name of the criteria and alternatives is entered in the second window (Fig. 3, *b*). The evaluation of the alternatives by the criteria, actions on the criteria, and the importance of the criteria are entered in the third window (Fig. 4, *a*, *b*). The first window has the button "Next" (Fig. 3, *a*), the click on which displayed saved solutions from the database (Fig. 5, *b*).



Enter the number of alternatives

4

Enter the number of criteria

6

Enter the name of the alternatives:

Huawei P20 Lite

Meizu 16

Samsung A8+

Xiaomi Mi 8

Enter the name of the criteria:

Screen size

Camera

Memory

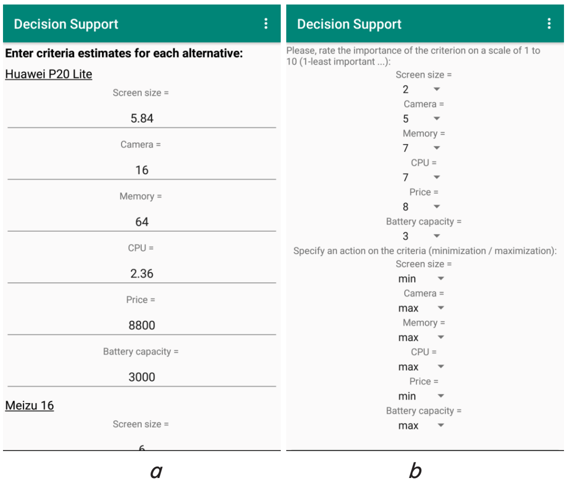
CPU

Price

Fig. 3. User interface windows:
a – first window; *b* – second window

When the decision maker finished entering the necessary data for the problem solution and the button "Get a solution" is pressed in the third window, the decision-making subsystem is started.

Also, the user interface is responsible for displaying the results of the calculation. The name of the best alternative and the diagram, which displays the ranked alternatives are shown in the fourth window (Fig. 5, *a*).



Enter criteria estimates for each alternative:

Huawei P20 Lite

Screen size = 5.84

Camera = 16

Memory = 64

CPU = 2.36

Price = 8800

Battery capacity = 3000

Meizu 16

Screen size = 6

Please, rate the importance of the criterion on a scale of 1 to 10 (1-least important ...):

Screen size = 2

Camera = 5

Memory = 7

CPU = 7

Price = 8

Battery capacity = 3

Specify an action on the criteria (minimization / maximization):

Screen size = min

Camera = max

Memory = max

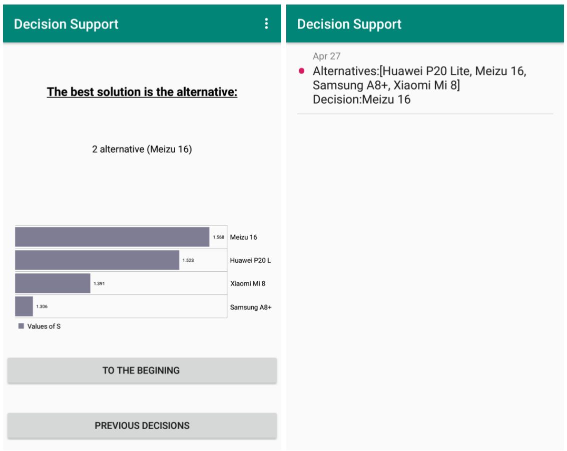
CPU = max

Price = min

Battery capacity = max

Fig. 4. User interface windows: *a*, *b* – third window

The fourth window of the user interface has two buttons: "To the beginning" and "Previous decisions". Pressing the button "To the beginning" allows the decision maker to return to the first window and to make changes in the data for the multiple criteria decision-making problem solution, pressing the button "Previous decisions" opens the list with problems, which were solved earlier (Fig. 5, *b*). If the decision maker long clicks on the note from the list, he takes the opportunity to delete the note from the list. If the decision maker selects the note from the list, the new window displays all solutions of the selected problem (Fig. 6, *a*, *b*).



Decision Support

The best solution is the alternative:

2 alternative (Meizu 16)

TO THE BEGINING

PREVIOUS DECISIONS

Decision Support

Apr 27

- Alternatives:[Huawei P20 Lite, Meizu 16, Samsung A8+, Xiaomi Mi 8]
Decision:Meizu 16

a *b*

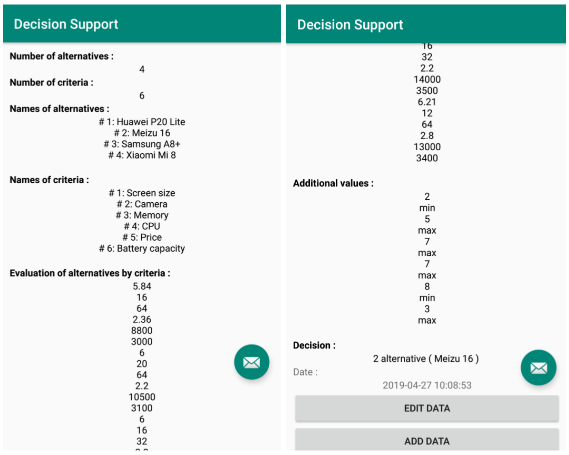
Fig. 5. User interface windows: *a*, *b* — fourth window

If the decision makers clicks the button "Edit data" (Fig. 6, *b*), the third window is displayed, where the decision maker takes the opportunity to edit numerical estimates of alternatives by criteria, edit criteria's weights and enter the optimality criterion. If the decision maker clicks the button "Add data" (Fig. 6, *b*), the first window is displayed and the decision maker takes the opportunity to edit data, add or delete data and make the solution of the new problems.

When the decision maker clicks the graphic object ✉ (Fig. 6, *a*, *b*), the launch of the message management subsystem occurs. This allows the decision maker to send the results of calculations via the Internet using the short message service (SMS).

The launch of the database interaction subsystem, which is responsible for transferring and retrieving data to/from the database, is performed via the user interface.

The message management subsystem and the database interaction subsystem belong to the application layer of the DSS.



Decision Support

Number of alternatives :
4
Number of criteria :
6
Names of alternatives :
1: Huawei P20 Lite
2: Meizu 16
3: Samsung A8+
4: Xiaomi Mi 8

Names of criteria :
1: Screen size
2: Camera
3: Memory
4: CPU
5: Price
6: Battery capacity

Evaluation of alternatives by criteria :
5.84
16
64
2.36
8800
3000
6
20
64
2.2
10500
3100
6
16
32

Decision Support

16
32
2.2
14000
3500
6.21
12
64
2.8
13000
3400

Additional values :
2
min
5
max
7
max
7
max
8
min
3
max

Decision :
2 alternative (Meizu 16)
Date :
2019-04-27 10:08:53

EDIT DATA

ADD DATA

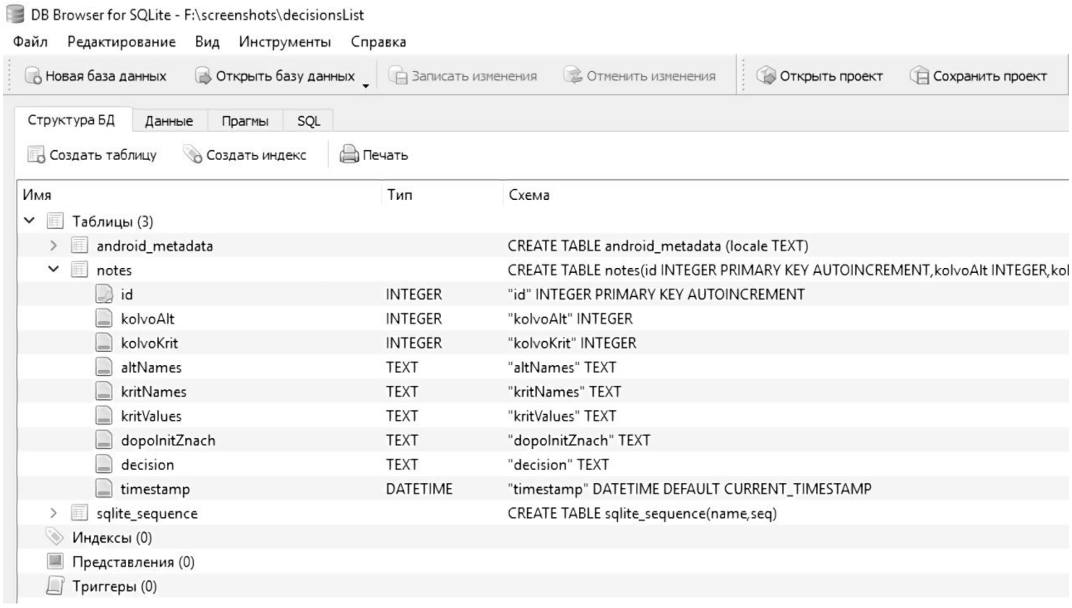
a *b*

Fig. 6. User interface windows: *a*, *b* — firth window

6. Creation of a database

The SQLite relational database management system was used to create a database for the mobile DSS. A feature of SQLite is that it does not use the client-server paradigm. SQLite is a library with which the program is compiled and the kernel becomes an integral part of the program. Thus, SQLite library function calls (API) are used as an exchange protocol. This approach reduces response time and simplifies the program. SQLite stores the entire database (including definitions, tables, indexes, and data) in a single standard file on the device on which the application is run. Before starting the transaction, the file, in which the database is stored, is blocked. This simplifies the implementation. ACID (Atomicity, Consistency, Isolation, and Durability) functions are achieved in particular by creating a log file. ACID is a set of requirements that guarantees the reliable operation of database transactions.

The database of the developed mobile DSS consists of one table. The structure of the developed database is shown in Fig. 7.



DB Browser for SQLite - F:\screenshots\decisionsList

Имя	Тип	Схема
Таблицы (3)		
android_metadata		CREATE TABLE android_metadata (locale TEXT)
notes		CREATE TABLE notes(id INTEGER PRIMARY KEY AUTOINCREMENT,kolvoAlt INTEGER,kol
id	INTEGER	"id" INTEGER PRIMARY KEY AUTOINCREMENT
kolvoAlt	INTEGER	"kolvoAlt" INTEGER
kolvoKrit	INTEGER	"kolvoKrit" INTEGER
altNames	TEXT	"altNames" TEXT
kritNames	TEXT	"kritNames" TEXT
kritValues	TEXT	"kritValues" TEXT
dopolnitZnach	TEXT	"dopolnitZnach" TEXT
decision	TEXT	"decision" TEXT
timestamp	DATETIME	"timestamp" DATETIME DEFAULT CURRENT_TIMESTAMP
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
Индексы (0)		
Представления (0)		
Триггеры (0)		

Fig. 7. Structure of the database

The table contains the primary key field, which is used to ensure the uniqueness of the data in the table. Also, the fields with information on the number of alternatives, the number of criteria, the name of alternatives, and the name of the criteria are indicated. In addition, there are fields in the table:

– numerical evaluation of alternatives by criteria;
– evaluation of the importance of the criterion;
– determination of action over the criteria;
– solution;
– the time of writing the solution to the database.

7. Software implementation of a mobile decision support system

The mobile DSS was developed with the Java language in Android Studio 3.2.1. Java is a strongly typed object-oriented programming language developed by Sun Microsystems, therefore, a Java language program consists of classes and the class diagram is shown in Fig. 8.

The mobile application [18], which has been developed using the two-layer architecture, was used in the implementation of the mobile DSS. This application provided autonomous work of the decision maker, however, it was not possible to save and use the saved data for the multiple criteria decision-making problems solving. The developed mobile DSS uses the three-layer architecture. Each level has been implemented in the form of the corresponding classes.

The presentation level program code was supplemented by the following classes.

The class FirstActivityForDbItem.java implements the first user desktop interface for editing data from the database:

– allows to change the number of criteria and alternatives;
– allows to open a list of tasks that were solved earlier;
– performs verification of the entered data.

The class SecondActivityForDbItem.java implements the second user desktop interface for editing data from the database, namely, it allows to change the names of alternatives and criteria and performs verification of the entered data.

The class ThirdactivityForDbItem.java implements the third user desktop interface for editing data from a database, namely:

– allows you to enter numerical estimates of each alternative according to the relevant criteria;
– allows you to enter criteria estimates;
– allows you to enter an action on the criteria (maximization/minimization);
– implements a modified algorithm of the heuristic Smart method;
– records the data to the database.

The class from_dblist.java implements the display a list of tasks that were solved earlier and allows the user to select a task from the list for viewing.

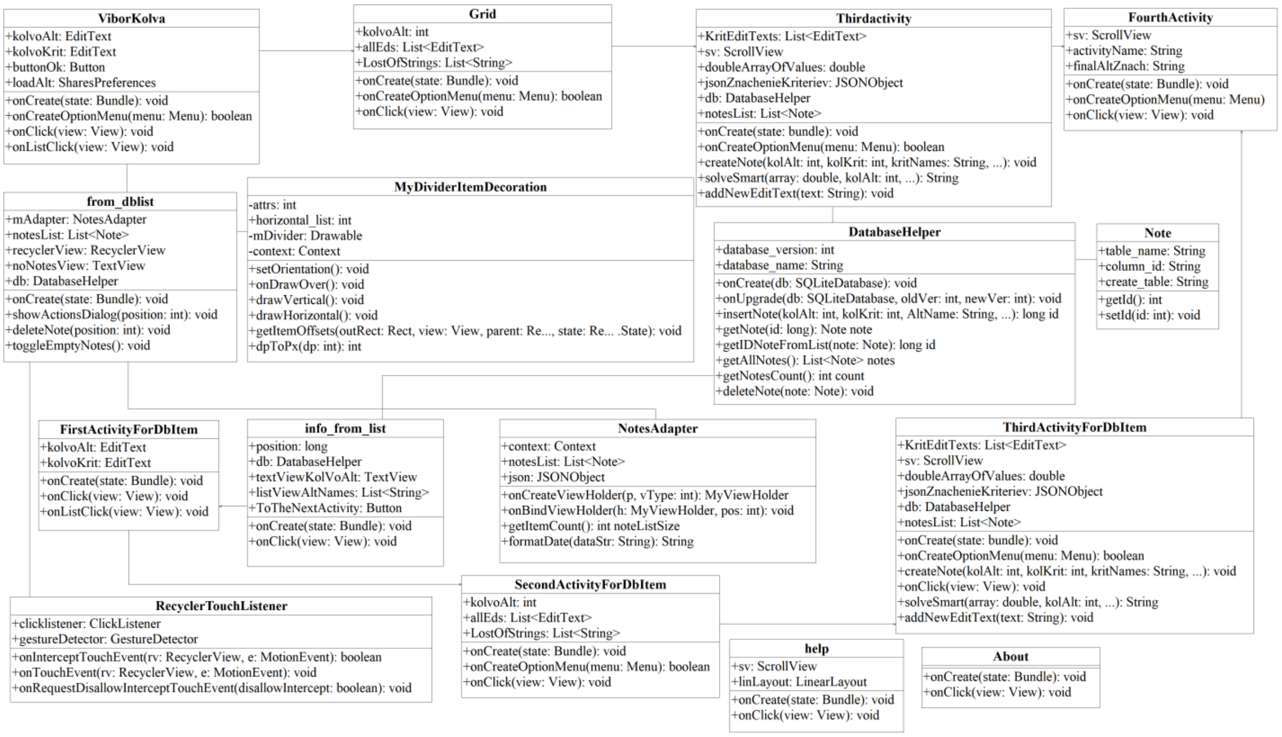
The class info_from_list.java implements the display a task from the list of tasks that were solved earlier and allows the user to select a task from the list:

– to view complete information about the selected task;
– to edit existing data (names of alternatives and criteria, criteria of evaluations of alternatives according to criteria, actions on criteria, importance of criteria) if necessary;
– to make changes (number of alternatives and criteria, names of alternatives and criteria, criteria for evaluating alternatives according to criteria, actions on criteria, importance of criteria) if necessary.

The class MyDividerItemDecoration.java implements a graphical design of the list of tasks that were solved earlier.

The application level program code contains the following classes:

The class Note.java creates an object that contains the complete information about the data that will be added or removed from the database.



ViborKolva
+kolvoAlt: EditText
+kolvoKrit: EditText
+buttonOk: Button
+loadAlt: SharesPreferences
+onCreate(state: Bundle): void
+onCreateOptionMenu(menu: Menu): boolean
+onClick(view: View): void
+onListClick(view: View): void
Grid
+kolvoAlt: int
+allEds: List<EditText>
+LostOfStrings: List<String>
+onCreate(state: Bundle): void
+onCreateOptionMenu(menu: Menu): boolean
+onClick(view: View): void
Thirdactivity
+KritEditTexts: List<EditText>
+sv: ScrollView
+doubleArrayOfValues: double
+jsonZnachenieKriteriev: JSONObject
+db: DatabaseHelper
+notesList: List<Note>
+onCreate(state: bundle): void
+onCreateOptionMenu(menu: Menu): boolean
+createNote(kolAlt: int, kolKrit: int, kritNames: String, ...): void
+solveSmart(array: double, kolAlt: int, ...): String
+addNewEditText(text: String): void
FourthActivity
+sv: ScrollView
+activityName: String
+finalAltZnach: String
+onCreate(state: Bundle): void
+onCreateOptionMenu(menu: Menu)
+onClick(view: View): void
from_dblist
+mAdapter: NotesAdapter
+notesList: List<Note>
+recyclerView: RecyclerView
+noNotesView: TextView
+db: DatabaseHelper
+onCreate(state: Bundle): void
+showActionsDialog(position: int): void
+deleteNote(position: int): void
+toggleEmptyNotes(): void
MyDividerItemDecoration
-attrs: int
+horizontal_list: int
-mDivider: Drawable
-context: Context
+setOrientation(): void
+onDrawOver(): void
+drawVertical(): void
+drawHorizontal(): void
+getItemOffsets(outRect: Rect, view: View, parent: Re..., state: Re... .State): void
+dpToPx(dp: int): int
DatabaseHelper
+database_version: int
+database_name: String
+onCreate(db: SQLiteDatabase): void
+onUpgrade(db: SQLiteDatabase, oldVer: int, newVer: int): void
+insertNote(kolAlt: int, kolKrit: int, AltName: String, ...): long id
+getNote(id: long): Note note
+getIDNoteFromList(note: Note): long id
+getAllNotes(): List<Note> notes
+getNotesCount(): int count
+deleteNote(note: Note): void
Note
+table_name: String
+column_id: String
+create_table: String
+getId(): int
+setId(id: int): void
FirstActivityForDbItem
+kolvoAlt: EditText
+kolvoKrit: EditText
+onCreate(state: Bundle): void
+onClick(view: View): void
+onListClick(view: View): void
info_from_list
+position: long
+db: DatabaseHelper
+textViewKolVoAlt: TextView
+listViewAltNames: List<String>
+ToTheNextActivity: Button
+onCreate(state: Bundle): void
+onClick(view: View): void
NotesAdapter
+context: Context
+notesList: List<Note>
+json: JSONObject
+onCreateViewHolder(p, vType: int): MyViewHolder
+onBindViewHolder(h: MyViewHolder, pos: int): void
+getItemCount(): int noteListSize
+formatDate(dataStr: String): String
ThirdActivityForDbItem
+KritEditTexts: List<EditText>
+sv: ScrollView
+doubleArrayOfValues: double
+jsonZnachenieKriteriev: JSONObject
+db: DatabaseHelper
+notesList: List<Note>
+onCreate(state: bundle): void
+onCreateOptionMenu(menu: Menu): boolean
+createNote(kolAlt: int, kolKrit: int, kritNames: String, ...): void
+onClick(view: View): void
+solveSmart(array: double, kolAlt: int, ...): String
+addNewEditText(text: String): void
SecondActivityForDbItem
+kolvoAlt: int
+allEds: List<EditText>
+LostOfStrings: List<String>
+onCreate(state: Bundle): void
+onCreateOptionMenu(menu: Menu): boolean
+onClick(view: View): void
RecyclerTouchListener
+clicklistener: ClickListener
+gestureDetector: GestureDetector
+onInterceptTouchEvent(rv: RecyclerView, e: MotionEvent): boolean
+onTouchEvent(rv: RecyclerView, e: MotionEvent): void
+onRequestDisallowInterceptTouchEvent(disallowIntercept: boolean): void
help
+sv: ScrollView
+linLayout: LinearLayout
+onCreate(state: Bundle): void
+onClick(view: View): void
About
+onCreate(state: Bundle): void
+onClick(view: View): void

Fig. 8. Class diagram of the mobile DSS

The class NoteAdapter.java implements the output of the information from the database in the list line, which is accessible to the user.

The class RecyclerTouchListener.java implements the processing of clicks on the item in the list of tasks that were solved earlier, while distinguishing between the short and long press. Short pressselects a list item, long press opens a dialog window that allows you to delete the item from the list.

The class Solve.java implements the improved algorithm of the heuristic Smart method.

The class Intent.java implements the sending of calculation results via the Internet or by using the short message service SMS.

The data level program code is represented by the class DatabaseHelper.java. This class provides the following features:

– creates a database;
– updates the database;
– records data;
– receives data;
– removes data.

8. Experimental research of the application of the developed system

The task of buying a smartphone was considered. Four models of the smartphone were considered as alternatives: Huawei P20 (alternative A_1), Meizu 16 (alternative A_2), Samsung Galaxy A8 (alternative A_3), Xiaomi Mi 8 (alternative A_4). Six criteria were chosen to evaluate the variants for buying a smartphone: screen size (criterion C_1), camera resolution (criterion C_2), memory (criterion C_3), processor frequency (criterion C_4), price (criterion C_5), and battery capacity (criterion C_6). All data were entered in the mobile DSS and the recommendation was received. The input data and the computation results are presented in Table 1.

Table 1

Input data and computation results

C_j	Input data						Selected alternatives
A_i	C_1	C_2	C_3	C_4	C_5	C_6	ImprovedSmart
A_1	5.84”	16 Mp	64 Gb	2.36 GHz	8,800 uah	3,000 mAh	1.5229980
A_2	6”	20 Mp	64 Gb	2.2 GHz	10,500 uah	3,100 mAh	1.5680102
A_3	6”	16 Mp	32 Gb	2.2 GHz	14,000 uah	3,500 mAh	1.3061009
A_4	6.21”	12 Mp	64 Gb	2.8 GHz	13,000 uah	3,400 mAh	1.3913106

By applying the improved Smart method we got that the best alternative is the alternative A_2, which corresponds to the smartphone Meizu 16.

Also, this multiple criteria decision-making problem was solved by the expert evaluation method. Five experts were invited to give a recommendation about buying a smartphone. All experts had an equal level of competence. The rank has been assigned to each alternative. So, the alternative to which the expert assigned the highest rating is assigned rank 1. If an expert recognizes several alternatives as equivalent, they are assigned the same rank number (Table 2).

To assess the average degree of the consistency of the opinion of all experts, the Kendall’s coefficient of concordance was calculated. It is equal to 0.808, which indicates about a high degree of the expert opinions consistency. To assess the significance of the Kendall’s coefficient of concordance, the Pearson criterion was calculated. The calculated value χ^2=12.12. It is greater than tabular χ^2=7.81473 for the number of degrees of freedom 3 with a significance level α=0.05.

Table 2

Matrix of ranks

Experts / A_i	1	2	3	4	5
A_1	4	3	4	4	3
A_2	1	2	1	2	1
A_3	3	4	3	3	4
A_4	2	1	2	1	2

So the value of the Kendall’s coefficient of concordance equal to 0.808 is not random and opinions of involved experts are agreed. Then, the obtained results of ranking alternatives are meaningful and can be used when calculating the weights of alternatives.

To calculate the weights of alternatives, the transition from the matrix of ranks to the matrix of transformed ranks is completed (Table 3). If the elements of the matrix of ranks are denoted by x_{ij}, i=1,..., n, j=1,..., m, where n – the number of alternatives, m – the number of experts, then the elements of the matrix of transformed ranks are calculated using the formula $s_{ij}=x_{\max}-x_{ij}$, where $x_{\max}$=4. The weight of an alternative is calculated as the ratio of the sum of the ranks of each alternative to the sum of the ranks of all alternatives.

Table 3

Matrix of transformed ranks

Experts / A_i	1	2	3	4	5	Sum	Weight of the alternatives
A_1	0	1	0	0	1	2	0.06667
A_2	3	2	3	2	3	13	0.4333
A_3	1	0	1	1	0	3	0.1
A_4	2	3	2	3	2	12	0.4
Sum						30	

Thus, research has shown that the expert evaluation method gives the best alternative A_2, which corresponds to the smartphone Meizu 16. The result which was obtained by using the developed mobile DSS is the same as the result of the expert evaluation method. Thus, the developed mobile DSS based on the improved heuristic Smart method can be proposed to the decision maker when solving the multiple criteria decision-making problems in the autonomous mode.

9. Discussion of results of research of the application of the developed system

The results of the research (Table 3) showed that the application of the developed mobile DSS based on the improved heuristic Smart method gives a result that coincides with the experts’ estimation. Thus, the developed mobile

DSS is a high-quality and easy-to-use tool for the assistance to the decision maker to compare solutions taking into account several criteria simultaneously.

Unlike the mobile application developed in [18], this system implements a data layer that contains the developed database using the SQLite relational database management system. The user interface has also been added, a database interaction subsystem was developed, which allows the decision maker to save received recommendations, edit data, and delete data. In addition, a message management subsystem was developed that allows sending results via the Internet or via SMS.

The advantage of using a mobile DSS based on the proposed method compared with the use of known methods such as AHP, MAHP, Electre is the low computation resource intensity of the proposed method, as well as ease of use.

Compared with the considered mobile DSS [10–14], the developed mobile DSS is based on the three-layer architecture. All system services are installed directly on a mobile device with Android OS. This ensures the autonomy of working with the system for decision makers, which leads to an increase in the efficiency of the decision-making process.

The developed mobile DSS is invariant to the subject area, which allows it to be used to solve problems from different application areas.

A limitation in the use of the developed mobile DSS can be the amount of information stored in the database, as well as the use of only one method for solving the multiple criteria decision-making problems in the system. Therefore, further research can be directed to the modification of the developed system, which ensures the use of the three-layer architecture. It is necessary to solve the following problems:

– ensuring the ability to save data in the cloud storage, so as not to violate the autonomy of the decision maker;

– implementation and connection of methods for solving the multiple criteria decision-making problems whose calculations are non-resource-intensive to the decision-making subsystem.

10. Conclusions

1. A decision-making subsystem has been developed that belongs to the application level in the three-layer architecture. This architecture was chosen to facilitate the possible modification of individual components of the developed DSS, if necessary. When developing this subsystem, an improved Smart method was used, which allows reducing the subjective role of the decision maker in the process of evaluating alternatives according to criteria, taking into account the nature of operations on criteria, and normalizing of the matrix taking into account the nature of operations on criteria.

2. A user interface has been developed that belongs to the presentation layer in the three-layer architecture. The interaction between the mobile DSS and decision maker occurs through the user interface. A message management subsystem has been also developed, which allowed the decision maker to send the calculation results via the Internet or via SMS. The developed database interaction subsystem allows the decision maker to save recommendations, edit data, delete data. The message management subsystem and the database interaction subsystem belong to the application level.

3. To create a database for mobile DSS, the SQLite relational database management system was used. SQLite stores the entire database (including definitions, tables, indexes, and data) in one standard file on the device on which the application runs. The created database consists of a single table that contains fields indicating the number of alternatives and criteria, as well as their names. In addition, there are estimates of alternatives on the criteria and the choice of action on the criteria. The database also stores the solution and the time it takes to write the solution to the database.

4. To create a mobile DSS on the Android platform, it was based on the mobile application developed by the authors for solving the multiple criteria decision-making problems using the two-layer architecture. Presentation level and application level of the application have been implemented in the form of corresponding classes. The mobile DSS developed in this paper uses the three-layer architecture. New presentation layer classes were developed to display, modify, or delete stored data and/or solutions from the database. A database of the solutions was also created, which was presented to the data layer. Classes of the application level were developed to implement the database interaction subsystem and message management subsystem.

5. The developed mobile DSS was used when the user chooses the best solution in the problem of buying a smartphone. Four smartphone models were considered as alternatives; six criteria were chosen to evaluate alternatives for buying a smartphone. With the help of the mobile DSS, the recommendation on the buying of a smartphone was received. The result, which was obtained using the developed mobile DSS, coincides with the result of the expert evaluation method with a high degree of consistency of expert opinions (Kendall's coefficient of concordance is 0.808).

References

1. Multiple criteria decision-making techniques and their applications – a review of the literature from 2000 to 2014 / Mardani A., Jusoh A., MD Nor K., Khalifah Z., Zakwan N., Valipour A. // Economic Research-Ekonomska Istraživanja. 2015. Vol. 28, Issue 1. P. 516–571. doi: https://doi.org/10.1080/1331677x.2015.1075139
2. Kuznetsov M. A., Ponomarev S. S. Sovremennaya klassifikatsiya sistem podderzhki prinyatiya resheniy // Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii. 2009. Issue 3. P. 52–58.
3. Power D. J. Web-based and model-driven decision support systems: concepts and issues // 2000 Americas Conference on Information Systems (AMCIS), 2000. California, 2000.
4. Guo X., Díaz López A. Mobile Decision Support System Usage in Organizations // 2013 19th Americas Conference on Information Systems (AMCIS), 2013. Chicago, 2013.
5. Lisetskiy Yu. M. DSS for selecting the elemental basis of corporate integrated information systems // Matematychni mashyny i systemy. 2017. Issue 3. P. 23–37.

6. Gaynanova R. Sh., Shirokova O. A. Sozdanie klient-servernyh prilozheniy // Vestnik Kazanskogo tekhnologicheskogo universiteta. 2017. Vol. 20, Issue 3. P. 79–84.
7. Multi-Criteria Decision Analysis Methods in the Mobile Cloud Offloading Paradigm / Bangui H., Ge M., Buhnova B., Rakrak S., Raghay S., Pitner T. // Journal of Sensor and Actuator Networks. 2017. Vol. 6, Issue 4. P. 25. doi: https://doi.org/10.3390/jsan6040025
8. Shishkin Yu. E. Oblachnye servisy v sistemah podderzhki prinyatiya resheniy // Avtomatika. Vychislitel'naya tekhnika. 2017. Issue 1 (14). P. 19–20.
9. Tupalo Y. Architecture of expert support and decision making system // Kompiuterni zasoby, merezhi ta systemy. 2017. Issue 16. P. 146–155.
10. Kostoglou V., Kafkas K. Design and development of an interactive mobile-based decision support system for selecting higher education studies // Balkan Region Conference on Engineering and Business Education. 2017. Vol. 3, Issue 1. P. 240–248. doi: https://doi.org/10.1515/cplbu-2017-0032
11. Decision Support System Using Mobile Applications in the Provision of Day to Day Information about Farm Status to Improve Crop Yield / Ogunti E. O., Akingbade F. K., Segun A., Oladimeji O. // Periodicals of Engineering and Natural Sciences. 2018. Vol. 6, Issue 2. P. 89–99.
12. Obelets T., Kiforenko S. Mobile information system for decision support in diabetology applying cloud services // Mezhdunarodniy nauchniy zhurnal «Internauka». 2017. Issue 8 (30). P. 56–59.
13. The Design of a Cloud-based Clinical Decision Support System Prototype / Barigou B. N., Barigou F., Benchehida C., Atmani B., Belalem G. // International Journal of Healthcare Information Systems and Informatics. 2018. Vol. 13, Issue 4. P. 28–48. doi: https://doi.org/10.4018/ijhisi.2018100103
14. Implementation of Korean Clinical Imaging Guidelines: A Mobile App-Based Decision Support System / Lee J. H., Ha E. J., Baek J. H., Choi M., Jung S. E., Yong H. S. // Korean Journal of Radiology. 2019. Vol. 20, Issue 2. P. 182–189. doi: https://doi.org/10.3348/kjr.2018.0621
15. Iphar M., Alpay S. A mobile application based on multi-criteria decision-making methods for underground mining method selection // International Journal of Mining, Reclamation and Environment. 2018. P. 1–25. doi: https://doi.org/10.1080/17480930.2018.1467655
16. An Empirical Study of Investigating Mobile Applications Development Challenges / Ahmad A., Li K., Feng C., Asim S. M., Yousif A., Ge S. // IEEE Access. 2018. Vol. 6. P. 17711–17728. doi: https://doi.org/10.1109/access.2018.2818724
17. Perez I. J., Cabrerizo F. J., Herrera-Viedma E. A Mobile Decision Support System for Dynamic Group Decision-Making Problems // IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans. 2010. Vol. 40, Issue 6. P. 1244–1256. doi: https://doi.org/10.1109/tsmca.2010.2046732
18. Kozina Y., Volkova N., Horpenko D. Mobile Application for Decision Support in Multi-Criteria Problems // 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP). 2018. doi: https://doi.org/10.1109/dsmp.2018.8478499