

Interactive Learning Application for Teaching the Rusyn Language

Perháč, Ján and Varchola, Peter

Abstract: *With a rapid spread of mobile devices over the market, many mobile apps have been created for various activities. One of the popular and very important type of application are educational ones. The main goal of this paper is a design and implementation of interactive mobile applications for teaching the Rusyn language. We have developed a native application designed for the Android operating system. The course lessons included in the application database are designed for learning the basics of the language, alphabet, and handle a simple conversation. We use a local database to store the data needed for proper application functionality. Because the database contains all the data already in the installation file, the application is fully functional without an internet connection.*

Index Terms: *Android, Native application, Rusyn language, SQLite*

1. INTRODUCTION

NOWADAYS, it is possible to replace the classic way of learning a foreign language, such as drilling vocabulary and grammar or performing tasks, with the usage of modern technologies. With the arrival of smart devices such as smartphones or tablets, new online language learning opportunities have opened up. These apps are popular and effective, especially because smartphone or tablet application teaching can be done anywhere. The market offers two types of language learning applications and apps that offer a story during which you gain knowledge or applications offering a course in a form of traditional lessons.

In this paper, we present the basics methods of application development of mobile operating systems, a design, and the development of a mobile application serving the teaching of a language. We use the codified version of the Rusyn language

Manuscript received Jun 15, 2020.

This work was supported by the Faculty of Electrical Engineering and Informatics at the Technical University of Košice under contract No. FEI-2020-70: "Behavioral model of component systems based on coalgebras and linear logic", and by the project KEGA 011TUKE-4/2020: "A development of the new semantic technologies in educating of young IT experts".

J. Perháč (contact person) is an assistant professor at the Faculty of Electrical Engineering and Informatics, Technical University of Košice, Slovakia (e-mail: jan.perhac@tuke.sk).

P. Varchola - the Faculty of Electrical Engineering and Informatics, Technical University of Košice, Slovakia (e-mail: peter.varchola@student.tuke.sk).

based on the Prešov's dialect [12]. The application base consists of interactive teaching lessons divided according to their difficulties. Users can take lessons in different areas and monitor their learning process. Upon successful completion of the course, they can use basic communication in the Rusyn language.

2. TYPES OF APPLICATIONS

Three types of applications are currently in use on the market. These three types characterize Raluca Budi [6] as follows.

- **Native applications** are installed through an application store, e.g. Google Play or the App Store. Their development is designed specifically for one platform. Such applications can take advantage of all device features, including camera, GPS, compass, contact list, and more. Such applications can also handle different gestures.
- **Web applications** in many ways they look like natural applications. In reality, however, it is a website. Their launching runs through the browser. This type of application is difficult to distinguish from a classic native or hybrid application.
- **Hybrid applications** are popular for one reason - they allow development on multiple platforms. With this step, development costs are noticeably reduced. Another advantage is the reuse of source code on multiple platforms.

For our purposes, we have chosen to develop a native application. Native applications use the built-in features of the device, making it easier and faster for them to work with.

3. PLATFORM AND API

The most important criterium for selecting this platform was its market share. It will allow using the application for as many users as possible. Table 1 shows the proportion of mobile operating systems. We can see the dominant share of the Android operating system on the market in the same quarter of two consecutive years.

TABLE 1: Market share of mobile OS in the second quarter of 2019 and 2020 [3]

Period	Android	iOS	Win. Ph.	Others
Q2 of 2019	75,52%	22,5%	0,23%	1,75%
Q2 of 2020	71.95%	27,45%	0,05%	0,45%

Table 2 shows the market share of the Android API as of May 2020. Due to the market share and the required API functionality, we decided to use the KitKat version 19 interface for our application.

TABLE 2: Share of API versions in May 2020 [2]

Version	Codename	API	Market share
4.4	KitKat	19	1.41%
5.1	Lollipop	22	4.46%
6.0	Marshmallow	23	8.73%
7.0	Nougat	24	6.07%
7.1	Nougat	25	4.13%
8.0	Oreo	26	7.34%
8.1	Oreo	27	11.49%
9.0	Pie	28	36.17%
10.0	Android 10	29	18.55%
Other	-	-	1,63%

Based on the result of the previous two tables, we choose to create an application for one platform. It is clear, this is an Android platform, namely API version 19, which is chosen as the interface with minimal requirements allowing the application to be used on smartphones with API versions 19 and latter.

4. APPLICATION DEVELOPMENT TOOLS AND TECHNOLOGIES

Android application development is in the so-called Host-Target development environment. The application is developed in one environment and started in another [13]. This means that the development of the application takes place on a computer that contains the development environment and other necessary software, and the test phase of the final project takes place on a mobile device, on the smartphone.

Android application development needs four basic tools and components [9]. Developer Environment, Java Development Kit (JDK), Android Development Tools (ADT), and Android Software Development Kit (SDK).

We have used Android Studio from IntelliJ as a development environment. We chose Android Studio mainly because Google officially supports it as an IDE for Android application development and recommends its usage [4]. Android Studio offers a variety of device tools and emulators to test the final application.

As the main programming language, we chose Java. Java is a programming language that is not tied to a specific hardware or operating system [5]. Programs developed in this language can be performed anywhere on any system.

The language learning application must contain a database. In our case, we use a local database to store all the data. MyScript Interactive Ink is an interactive feature for recognizing handwritten text from the screen.

4.1. SQLite

SQLite is a library written in C, which is part of the Android operating system [14]. Supports

standard relational database functions such as SQL syntax, transactions. It requires little memory and is part of every Android device.

SQLite is reliable and, as a database system, it can be selected for many consumer electronic devices [11]. By implementing a library instead of running as a separate running process, each SQLite database is integrated into the application that created it. This minimizes the response and simplifies synchronization.

4.2. MyScript Interactive Ink

MyScript Interactive Ink is an on-line recognition system. Its role is to analyze the sequence of points in 2D. The sequence is based on the tip of the digital pen. The track is defined as a finger trajectory or touch pen from the moment it touches the writing area. The track is represented as the order of 2D points (x; y) arranged according to time (t). MyScript as an input, processes only "digital ink", which can be defined as a series of moves. Digital ink, therefore, refers to a dynamic process that takes into account where the strokes begin, where they end, and in what order they were drawn to the screen of the device [1].

5. ANALYSIS OF LANGUAGE TEACHING METHODS

We can divide language teaching methods into many categories. However, it is enough to separate the classroom teaching methods from those offered by online learning platforms, so-called m-learning, or mobile-learning. M-learning is a form of e-learning and it is available anywhere and at any time, what increases learning comfort.

Antonio Lora in [10] claims that thinking about learning languages using mobile devices is becoming more effective and easier. Especially because of the huge popularity of these devices among students.

5.1. Classical Methods of Teaching

In the history of foreign language didactics, several definitions of teaching methods have emerged. Classical methods of teaching a foreign language include school lessons or language schools, where a pupil should master a language by a means of communication. Great part here is word or grammar drilling or filling lessons in textbooks. This includes all classical grammar and translation methods. Classical methods of language learning are mostly used nowadays in schools, but some of them use additionally interactive devices like smartphones, tablets, or computers.

5.2. Online Teaching Methods

Online learning platforms provide specific training and exercise options that can be accessed through a web browser or other devices [8].

Advances in mobile technologies are rapidly expanding education in areas beyond formal education by enabling flexible access to digital resources. Mobile learning can be an important complementary role in formal education [7]. Potential benefits of mobile learning are very popular for a variety of reasons. One of these is cost savings, omnipresent communication, and learning aids.

Current research [10] has shown that students positively perceive the use of their mobile phones as a learning tool for learning new languages.

This platform offers more non-traditional forms of learning, and learning is more interesting. Using such a device, language learning is more accessible and convenient than ever before.

6. APPLICATION DESIGN AND IMPLEMENTATION

The application design for language learning consists of several parts. First, we created an application lifecycle, based on which we selected points on which we built the application design. The main points include a list of lessons and the content of lessons, question and answer types, and a local database with all data needed to run the application smoothly. The Home screen is a sign-in screen. After signing in, it becomes the main screen with a list of lessons. The application also includes menus with different settings. During the design of the solution, we also focused on implementing interactive responses in the application. Also, the user's data and learning process are stored in the local database.

The application is intended for the teaching of the Rusyn language. Lessons themselves are divided into smaller lessons. We have only one type of user in the designed system because this type of application does not need administration on the system unlike classic web applications. The proposed system offers users to log-on with their Google accounts, if more than one user uses the application on one device, they can select their accounts from a list. It also allows a user to display information about the content of the selected lesson after selecting the lesson. When continuing the learning process after a break, a user gets to the lesson where the break was made. The object of the lesson is then accessed by entering a question about the letters, words, or entire phrases stored in the local database. After completing the lesson, a user can view the results and share it on different social networks.

6.1. Project Structure

The whole project is in one package. Since the project contains many classes, we have divided it into several more packages for better orientation and class management. Fig. 1 shows the package structure.

- The **entities** package contains entity classes that define the attributes of each table of the two databases used.
- The **helpers** package is the DatabaseHelper and UserDbHelper classes that extend the SQLiteOpenHelper class. In both classes, we created the onCreate, onUpgrade methods. Using these methods, tables will be created and they will also take care of their updates. These transactions are used to ensure that the database is always in the current state.
- The **questions** package contains classes that define response activity for the response screen. It also includes transient or summary activity to score scores after each answer and at the end of the lesson. The third file is the CustomAdapter class designed for scoring calculations and allowing you to switch to other questions.
- The **ink** package contains workspaces for writing on the screen by hand. There are implemented Interactive Ink SDKs.
- The **screens** package contains activities that appear when the user is not completing the lesson. This is a sign-in screen, the main screen - a list of lessons and more.

6.2. Local Database

We have created two local databases for proper write and read. The first database is used to store user information and how to use it. The second one is compressed to ZIP and stored in the assets folder. The assets folder is local because it is needed to be distributed in the installation file. It contains tables where all lessons i.e. questions and answers are stored.

For both databases, we have created helper classes. These classes contain CRUD operations such as creating a table or updating it. In addition to CRUD operations, they also include methods representing different SQL commands such as data entry or data selection.

The helper class for the database query contains a method used to copy the database from the compressed file to the package where the application is stored. Copy is triggered after a request coming from the application. The most likely requirement is to select a lesson from the list when the content of the lesson is loaded. When copying, we define paths for the source, where we copy, and the destination where we copy. We store 1024b chunks in the cache, and copying takes place while the source size is greater than 0b. Finally, we close these paths to avoid unwanted data manipulation. Fig. 2 shows the structure of the local database.

6.3. Description of the Application

Every Android device owner must be the owner of Google Account. For this reason, we've decided

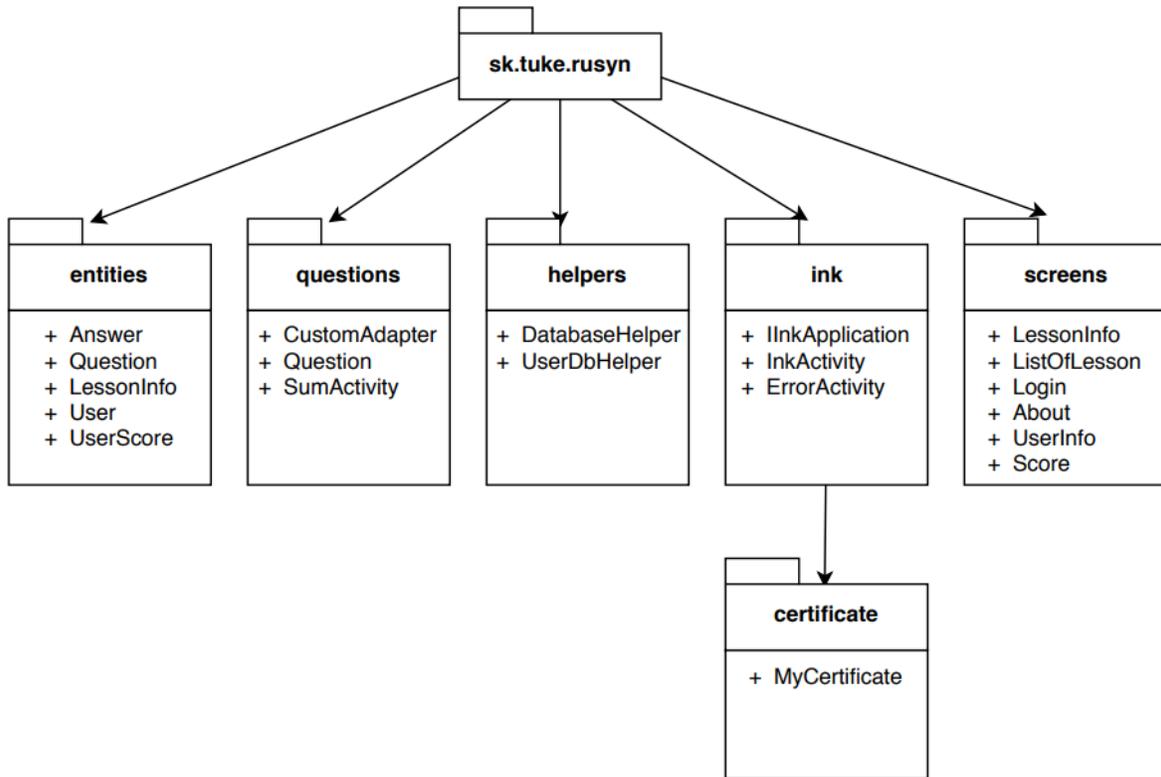


Fig. 1: Package structure

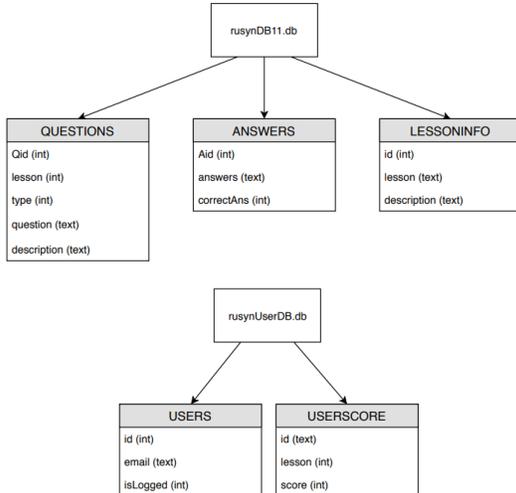


Fig. 2: The structure of the local databases

that Google Authentication will be sufficient to login into the application. We use this authentication method to identify the user and track his progress. After logging in successfully, the user will see activity with a list of all the lessons. The user's profile is also accessible after logging in.

On the main screen with a list of lessons, there are buttons for selecting a lesson. Pressing a button opens a new activity displaying information about what the content of the lesson is. We open

the new activity using the Intent message object, to which we add the number representing the order of the lesson. Based on this number, we pick information from the table about lesson's content and display it on the screen. We only list that data which value is equal to the value sent to the Intent while opening a new activity. The main screen of the application is depicted in Fig. 3.

To change the question while completing the lesson, we used the Adapter. Its main task is to create a bridge between user interface components and the source code of the application. The application's source code then completes the user interface components with the required data.

Using get methods, we find the values for correct plotting of the question. The resulting field length is the number of responses to the current question. To get one item from a worksheet, we use SQL select with the index assigned to the correct question. Each answer in the database is also information about the correctness or mis-statement.

Each question stored in the database has an assigned response type based on the drawing of the user interface components needed to answer the question. The first two types are answers with varying numbers of responses. In these questions, we use the Adapter to draw the number of options. The third type is for typing the answer using the keyboard. All unneeded elements will be hidden,



Fig. 3: The main screen of application

only those that are needed to answer the question will become visible. For the fourth type of question, we open a handwriting activity. Example of lesson's question is depicted in Fig. 4



Fig. 4: Lesson's question

The lesson questions show a summary activity that serves to display the actual score, the number of the next question, and the information

on whether the previous question was answered correctly or not. After pressing the Continue button, intent re-opens the question with the correct rendering of the UI components. Summary of a finished lesson is depicted in Fig. 5.

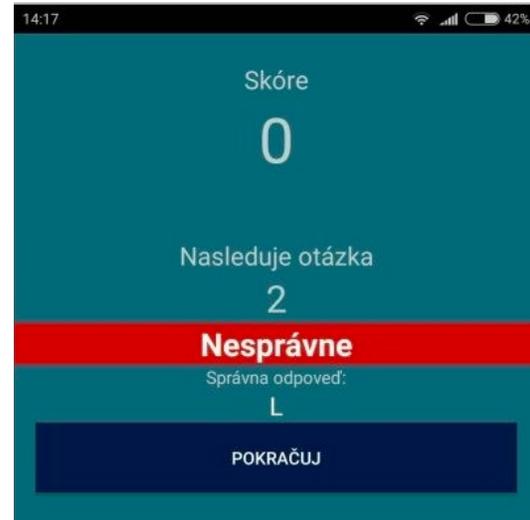


Fig. 5: Lesson's summary

In order to create interactive lessons, we decided to use the type of answer where the user writes the answer by handwriting.

The script's integration into the application proceeds as follows. After registering and downloading the template, we received a unique certification code that is required for the script to work properly. The certificate must be obtainable within the package in order to be used. It is placed in a bundle with other grades used in the handwriting. The downloaded template contains an implementation ready for use on the Android operating system. The template includes features such as capturing and plotting lines text. These functions are located in the UIReferenceImplementation directory. This directory needs to be implemented in the project.

After completing the lesson, the overall lesson will be displayed. This screen also includes the possibility of repeating a lesson or sharing the result on a social network.

7. CONCLUSION

Android is the most widespread mobile operating system. An application for Rusyn language teaching was implemented for KitKat and later APIs 19 which means the application can be used on devices running Android 4.4 or higher. The development was on the officially supported IDE Android Studio in Java.

For developing the application and designing our solution, we first studied the conditions for developing Android applications. Then we created the lesson content. We saved it to the local

SQLite database. The application also contains interactive questions that need to be answered in handwriting.

REFERENCES

- [1] About myscript interactive ink sdk. <https://developer.myscript.com/docs/interactive-ink/1.1/overview/about/>. Online, cit. 2018-06-03.
- [2] Mobile android version market share worldwide. <https://gs.statcounter.com/android-version-market-share/>, 2020. Online, cit. 2020-06-02.
- [3] Mobile operating system market share worldwide. <https://gs.statcounter.com/os-market-share/>, 2020. Online, cit. 2020-06-02.
- [4] Joseph Annuzzi Jr., Lauren Darcey, and Shane Conder. *Introduction to Android application development : Android essentials*. Addison-Wesley, Boston, MA, 2015.
- [5] E Balagurusamy. *Programming with Java : a primer*. Tata McGraw-Hill Education, New Delhi, 2010.
- [6] Raluca Budiu. Mobile: Native apps, web apps, and hybrid apps. <https://www.nngroup.com/articles/mobile-native-apps/>, 2013.
- [7] Jongpil Cheon, Sangno Lee, Steven M Crooks, and Jaeki Song. An investigation of mobile learning readiness in higher education based on the theory of planned behavior. *Computers & Education*, 59(3):1054–1064, 2012.
- [8] Pascal Giehl, Oscar Nierstrasz, and Mircea Lungu. The zeeguu translate application. *University of Berne*, 2015.
- [9] Luboslav Lacko. *Vyvoj aplikaci pro Android*. Computer Press, Brno, 2015.
- [10] Antonio Lora. *New perspectives on teaching and working with languages in the digital era*. Research-publishing.net, Dublin, 2016.
- [11] Reto Meier. *Professional Android 4 application development*. John Wiley & Sons, Indianapolis, IN, 2012.
- [12] Anna Plišková and Kvetoslava Koporová. *Rusyn Language for Beginners*. SARO – Rusín a Ľudové noviny, Prešov, Slovakia, 2011. (in slovak).
- [13] Miroslav Ujbanyai. *Programujeme pro Android*. Grada, Praha, 2012.
- [14] Lars Vogel. Android sqlite database and contentprovider-tutorial. *Java, Eclipse, Android and Web programming tutorials*, 8, 2010.

Ján Perháč received his PhD. in Informatics from the Technical University of Košice, in 2019. Currently, he is an assistant professor at the Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Slovakia. His research topics include computer security, category theory, logical systems, type theory, and semantics of programming languages.

Peter Varchola received his masters's degree in Informatics from the Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Košice, Slovak Republic in 2020. His research interests include mobile application development, and web application development.