

ТЕХНОЛОГІЇ РОЗРОБКИ ТА СУПРОВОДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

УДК 004.415.2

**Oleshchenko L.M., Lapchuk D. A.**  
**National Technical University of**  
**Ukraine "Igor Sikorsky Kyiv**  
**Polytechnic Institute"**

# SOFTWARE DEVELOPMENT USING CARTOGRAPHIC DATA

*The article considers the main tools for developing Android applications using data mapping services on the Internet. The resulting information is systematized to identify the tools used in developing a software system for collecting and analyzing data on the movement of mobile users. The programming languages and database management systems that are appropriate for solving this problem are considered. The description of the interaction with the database and the analysis of the users of the developed application with the help of Firebase Analytics and BigQuery is given. Describes the DBaaS architecture, lists and describes the modules of an Android application that simplifies data processing and facilitates scalability of the code and ensures proper non-excessive connectivity of code components. The algorithm of smoothing of geotracks, which is used in the application is presented. The main advantages of using Firebase Authentication technology for registration and authorization of users of the developed system are considered.*

*У статті розглянуті основні засоби для розробки Android-застосунків з використанням даних картографічних сервісів мережі Інтернет. Отримана інформація систематизована для визначення інструментів, що використовуються при розробці програмної системи для збору та аналізу даних про переміщення користувачів мобільних пристроїв. Розглянуто мови програмування та системи керування базами даних, які доцільно використовувати для вирішення даної задачі. Наведено опис взаємодії з базою даних та аналізу даних про користувачів розроблюваного додатку за допомогою Firebase Analytics і BigQuery. Описано архітектуру DBaaS, наведено перелік та опис модулів розроблюваного Android-застосунку, що спрощують роботу з обробки даних та полегшують масштабованість коду та забезпечують належну не надлишкову зв'язність компонентів коду. Наведено алгоритм згладжування геотреків, який використовується в додатку. Розглянуті основні переваги використання технології Firebase Authentication для реєстрації та авторизації користувачів розроблюваної системи.*

*В статье рассмотрены основные средства для разработки Android-приложений с использованием данных картографических сервисов сети Интернет. Полученная информация систематизирована для определения инструментов, используемых при разработке программной системы для сбора и анализа данных о перемещении пользователей мобильных устройств. Рассмотрены языки программирования и системы управления базами данных, которые целесообразно использовать для решения данной задачи. Приведено описание взаимодействия с базой данных и анализа данных о пользователях разрабатываемого приложения с помощью Firebase Analytics и BigQuery. Описаны архитектуру DBaaS, приведен перечень и описание модулей разрабатываемого Android-приложения, упрощающие работу по обработке данных и облегчают масштабируемость кода и обеспечивают должным образом не избыточную связность компонентов кода. Приведен алгоритм сглаживания геотреков, который используется в приложении. Рассмотрены основные преимущества использования технологии Firebase Authentication для регистрации и авторизации пользователей разрабатываемой системы.*

**Keywords:** software, mapping services, Google Maps, geo track, Android, Firebase, BigQuery, Java.

### **Introduction and problem statement**

At present, it is popular and in demand to create software for mobile users using map data. By constructing and recording geo-tracks, it is possible to maintain statistics on the movement of people of various categories, the construction of optimal routes and the tracing of the exact location of children for the purpose of their control and search in case of their disappearance.

Data on moving people in the future can be used to track the most popular places and identify potential areas for doing business and trade. For collecting and analyzing data on the movement of mobile users, special technologies and data repositories are needed to handle large amounts of data. The purpose of this study is to analyze and allocate the necessary technologies for software development, which collects and analyzes data on the movement of mobile users using data mapping services.

### **Select cartographic service**

*Google Maps* Cartographic Services is a collection of applications built on the basis of Google's free cartographic service and technology. The service is a map and satellite images of the whole world. The service has an integrated business directory and a map of highways with route search. Routing information is available in all browsers, including mobile, as well as Google Maps for Android and iOS.

*Google Street View* provides users with the ability to "get around" in a three-dimensional street projection over the Internet. This functionality has become possible thanks to the help of round-the-field photography of special equipment in real time. The result is a lot of spherical panoramas with geographic coordinates and an opportunity to switch between them, using an intuitive interface for this, creating a sense of movement in space. You can use the service to create your own products by third parties.

The Google Static Maps API allows you to build static maps using custom URL. There are also API versions for different types of mobile devices [1].

*OpenStreetMap* is an open source community-based project to create world-class mappings. Primary and basic information for OpenStreetMap are tracks collected by volunteers using devices equipped with GPS receivers and shipped to the OSM database. It is behind the tracks that the main network of roads is created [2].

### **Choosing a database management system to**

### **interact with mapping services**

*Firebase* is a cloud-based NoSQL database for real-time applications that provides an API that allows developers to store and synchronize data between multiple clients. *Firebase* NoSQL provides a database for real-time applications as a service. This service provides an API for developers, allows you to synchronize data and store them in a *Firebase* cloud. The RTDB synchronization feature provides the client with all the missing files after connecting the recovery, which makes it very valuable for the stable operation of the application. The company also took note of the possibility of integrating with Android, IOS, JavaScript, Java, Objective-C and Node.js. Developers can use the API provided by *Firebase* to encrypt data. Using the *Firebase* SDK for storing and scaling data ensures complete security and reliability. Information can be sent directly from the client to *Firebase* Cloud, various privacy settings can be adjusted to restrict access to specific groups or files. *Firebase* is supported by Google Cloud, so it can offer developers virtually endless storage space [12].

In the real-time *Firebase* database, data is stored in JSON format. This should be taken into account when developing the database schema so that the data is easily accessible, avoiding the insertion of the child nodes. JSON data is in the form of a tree, so each data unit can contain several more pieces of data, which in turn may also contain other data units. Using JSON does not allow structuring data at all, although it is not recommended. *Firebase* allows you to store data up to 32 levels. This allows you to have millions of data that contain millions of data. Most databases require HTTP calls to receive and synchronize data.

Most databases provide data only when the user requests their receipt. If the developer uses *Firebase*, the connection is made using WebSocket. WebSockets are much faster than HTTP. All data is synchronized automatically through one WebSocket as fast as the client network can carry them. Because the data is stored in a cloud rather than locally on a user's device, it allows the user to have access to data from any device if it is authorized by its own account. It also means that the application requires Internet connection, and the data must be uniquely linked to the user.

The main benefits of *Firebase* are:

- The JSON repository means the absence of a barrier between data and objects;
- easy serialization of the application;

- easy customization for use in the application;
- easy access to data, files, authorization, etc .;
- You do not need to configure the server infrastructure for the application;
- Great potential for storing large amounts of data;
- is a real-time database;
- high reliability and security;
- Easily integrates with other Google technologies.

The main disadvantages of Firebase are:

- very limited capabilities for queries and indexing;
- there is no possibility of aggregation;
- there is no map decrease;
- There is no way to create a request to view the list of users or saved files.

*SQLite* is a relational database management system. Incorporated in the form of a library, which implements many of the standard SQL-92. The SQLite code can be used without limitation and free of charge for any purpose.

The SQLite feature is that it does not use the client-server paradigm, that is, the SQLite engine is not a separate process that the application interacts with, but provides the library with which the program is compiled and the engine becomes an integral part of the program. In this way, SQLite library functions calls are used as a protocol. This approach reduces overhead, response time, and simplifies the program. SQLite stores the entire database (including definitions, tables, indexes, and data) in a single standard file on the computer on which the application is run. The ease of implementation is achieved due to the fact that before the transaction is started, the entire file that stores the database is blocked; ACID functions are achieved in particular through the creation of a log file. Multiple processes or threads can simultaneously read data from a single database without problems. Recording to the database can be done only if no other requests are currently served; otherwise the write attempt fails, and the program returns the error code. Another variation of events is the automatic repetition of attempts to record within a given time interval. The delivery package also includes a functional client part in the form of executable `sqlite3` file, which demonstrates the implementation of the functions of the main library. The client part runs from the command line, and allows you to access the database file based on the default OS functions. Thanks to the engine architecture, SQLite can be

used on both embedded systems and on dedicated machines with gigabytes of data.

Features of SQLite:

- Atomic, sequential, isolated, strong (ACID) transactions even after system crashes and power failures;
- installation without configuration - does not require any installation or administration;
- implements a large part of the SQL92 standard;
- the database is stored in one cross-platform file on a disk;
- support for terabyte database sizes and gigabyte-sized rows;
- Small code size: less than 350KB fully configured, and less than 200KB with lowered advanced features;
- faster than popular client / server database engines for the most common operations;
- Simple, easy-to-use API;
- well commented rust code with 100% test coverage of branches;
- Available as the only ANSI C Core file that can be easily inserted into another project;
- autonomy: no external dependencies;
- comes with an offline client-side command-line interface that can be used to manage SQLite databases.

SQLite has proven itself as an extremely reliable database system that is used in many consumer electronic devices and applications, including some MP3 players, iPhone, iPod Touch, Mozilla Firefox and others [14]. Microsoft SQL Server is a commercial database management system distributed by Microsoft. The language used for queries is Transact-SQL, created jointly by Microsoft and Sybase. Transact-SQL is the implementation of the ANSI / ISO standard for the structured language of SQL queries with extensions. It is used both for small and medium-sized databases, and for large scale enterprise databases. For many years, it has been successfully competing with other database management systems.

Microsoft SQL Server, as a query language, uses the SQL version called TRANSACT-SQL (T-SQL). T-SQL allows you to use additional syntax of stored procedures and provides transaction support (database interaction with the application manager). Microsoft SQL Server and Sybase ASE for network interfaces use the application level protocol called Tabular Data Stream (TDS, tabular data transfer protocol).

SQL Server supports mirroring and clustering databases. SQL Server Cluster is a collection of

identically configured servers. All servers have one virtual name, and the data is distributed according to the IP addresses of cluster machines during the work. Also, in the event of a failure or failure on one of the cluster servers, an automatic transfer of the load to another server is available [15].

Because software requirements require registration, authorization and authentication, in this case it is advisable to use the capabilities of the Firebase Authentication framework. It provides backend services, an easy-to-use SDK, and ready-made UI libraries to authenticate users into the application. It supports registration with a password, a mobile number, popular inclusive providers such as Google, Twitter, Facebook and others. Firebase Authentication is closely integrated with other Firebase services and uses the standards of OAuth 2.0 and OpenID Connect, and can be easily integrated with the backend of the application being developed. Firebase Auth also includes a user management system. The developer can store some basic information about the registered users of the application [3].

Firebase Analytics for Android lets developers understand how people use the app. The SDK automatically captures a number of user events and properties, as well as allows you to define your own events for measuring metrics that are important for an application being developed. Once the data is received, they are available in the Firebase console control panel. This panel provides a detailed view of all data, from

aggregated data such as the number of active users and demographic statistics, to more detailed data, such as the number of views of an application page. Analytics also integrates with a number of other Firebase features.

Analytics helps to understand how users behave when using the application [4]. Since the main task of the application being developed is the ability to analyze data in a qualitative way, using Firebase Analytics can greatly facilitate this task and provides a wide range of tools. If a developer needs to perform his own analysis of data obtained through Firebase Analytics or to combine data with other sources, it is possible to associate analysis of data with BigQuery. This allows for more complex analysis, for example, requests for large data sets and a combination of multiple data sources. BigQuery is a RESTful web service that allows interactive analysis of large data sets in conjunction with Google Storage. It is an infrastructure like a service (IaaS), which can be used in the MapReduce add-on. There is also no infrastructure to manage and there is no need for a database administrator so that the developer can focus on data analysis using a familiar SQL syntax. BigQuery can process data per second or data in one minute. Combining the Firebase application with BigQuery allows you to access the primary complete event data along with all the user's parameters and properties. Once configured, Google Analytics will export event data to the relevant BigQuery table on a daily basis [5].

*Table 1  
Comparison of database management systems  
Firebase Realtime Database, Microsoft SQL Server, SQLite*

Name	Firebase Realtime Database	Microsoft SQL Server	SQLite
Description	Cloud Real-time database	Microsofts Relational database management system	Widely used database management system
Database	Document Storage	Relational database management system	Relational database management system
Model	Used	Not Used	Not Used
Cloud Storage	Free scheme	Needed	Needed
SQL	Not Used	Used	Used
Data Sheet External Keys	Not Used	Used	Used
Concept of Transactions	Used	ACID	ACID
Parallel	Used	Used	Used

BigQuery manages the technical aspects of data storage, including compression, encryption, replication, performance adjustments and scaling. BigQuery supports a standard database concept

that includes tables, columns, rows, and sections. The bandwidth of BigQuery analytics is measured in slots. The slot is the unit of computing power required to execute SQL queries. BigQuery

automatically calculates how many slots you need for each query, depending on the complexity of the queries. BigQuery table rows can be not only simple key-value pairs. They may look like rows of JSON objects that contain simple data (such as strings, integers, and floats), but also more complex data such as arrays, structures, or even arrays of structures. This means that one line in the BigQuery table can contain an array of user properties, as well as an array of events, which, in turn, also have their own array of event parameters [6].

BigQuery supports two SQL dialects: standard SQL and legacySQL. Earlier, BigQuery used non-standard SQL dialects called BigQuery SQL. After releasing the BigQuery 2.0 version, standard SQL support was created, while BigQuery SQL was renamed to legacy sql.

The advantages of standard SQL are:

- Layout using WITH and SQL functions;
- subqueries in the fields SELECT and WHERE;
- correlated subtasks;
- data types ARRAY and STRUCT;
- Inserts, updates, deletes.

Taking into account the above advantages and the more convenient and understandable sql syntax, it was decided to use this particular dialect for the developed application [7]. In order to use BigQuery for the application being developed, the BigQuery Java API was used. The API allows all requests with access tokens OAuth 2.0.

The Java programming language is selected for the development of the system, because:

- Kotlin language is not yet sufficiently developed and has fewer documentation than Java;

- The Firebase project database selected for the project has a convenient and versatile Java API.

As the DBMS, it was decided to choose the Firebase Realtime database, because:

- It is a real-time database and allows you to send updates to all users of the application;
- Has access to Google Cloud, which allows you to use a nearly unlimited storage for document storage;
- Provides hosting for the application, authentication and authorization tools;
- Easily integrates with other Google services.

### Description of the architecture of the application to be developed

All classes of software developed can be divided into three groups:

- models;
- services;
- user interface.

The group of models can include all POJO classes for the entities used in the application. The services group includes entities validators, component injections, actions in response to user events, events that respond to user interface changes in active screens, backend implementation, and more. The UI group includes settings, all Activities and Fragments, but they are all separated from the code associated with the data logic as much as possible (fig. 1).

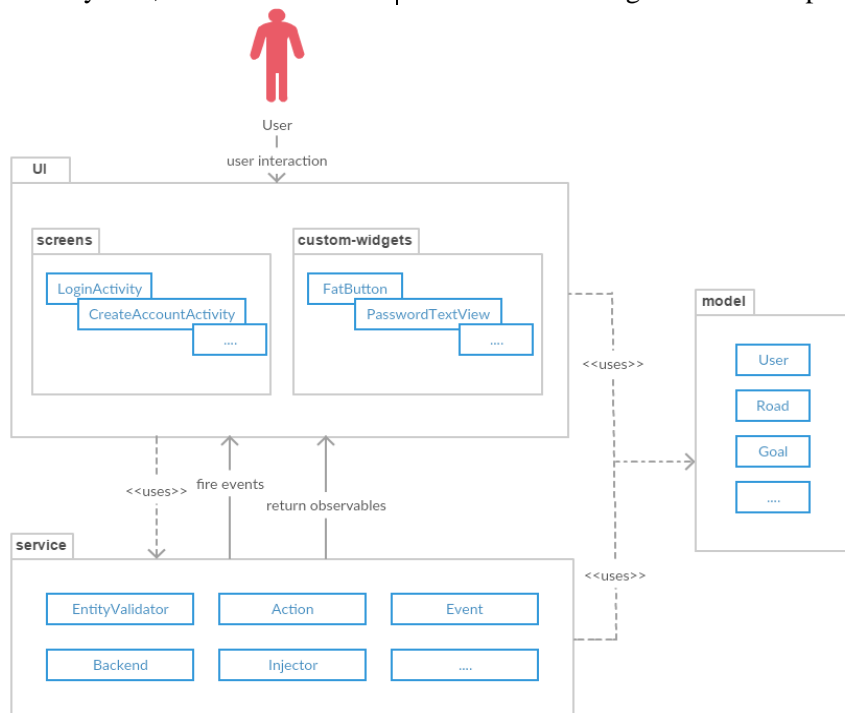


Fig. 1. DBaaS Software Development Architecture

DBaaS (Database as a Service) is an integrated solution for automated creation and provision of any type of database through the self-service portal. A database-like service (DBaaS) is a cloud computing service model that provides the user with some form of access to the database without the need to create physical hardware, software installation, or setup for work [8]. The Firebase

API allows you to easily synchronize and store data in real time (fig. 2). Technology allows you to unpack the application without having to manage servers or write code on the server side. The API supports standard CRUD operations with data within a database, as well as data requests. Firebase uses WebSockets to communicate between clients in real time.

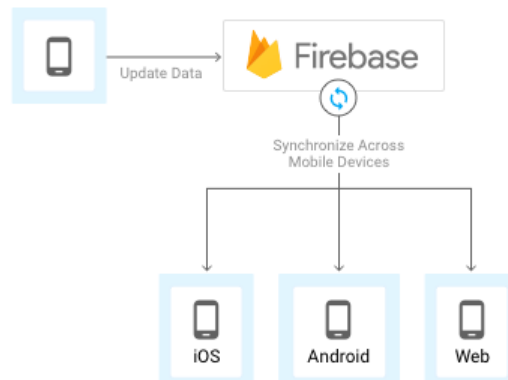


Fig. 2. Firebase database interoperability framework with devices [9]

The main features of such an architecture are:

- Data storage using JSON in the Firebase database in real time, as well as files in Firebase Storage;
- sending messages from Firebase Cloud Messaging;
- Automated data synchronization in real time between multiple devices;
- authentication of users through various services;
- rapid development of the backend.

In the software structure, it is possible to allocate the following modules, which the server part interacts with:

- database work module;
- registration and authentication module;
- Google Maps interaction module;
- comment module;
- personal information module;
- module for analyzing the results.

The database module provides functions for working with the database. Accordingly, the functions of this module are as follows:

- providing a single point of access to the database;
- ensuring the integrity of the data;
- synchronization of data of all devices in real time;
- recording, updating and deleting data in a database;
- conversion of data stored in the database into the respective application models and vice versa.

All system data is stored in the NoSQL Firebase database in JSON format. Google Maps interaction module is responsible for working with maps.

One can distinguish the following main functions of the module:

- displaying cards with special notations;
- Get coordinates of the geographic position of the user when recording a path.

Google Maps interacts with the Google Maps Android API. Firebase Cloud Storage is used as a cloud storage. If the network connection on the user's device is bad, the user may repeat the operation later where it was interrupted. Cloud storage stores files in Google Cloud Storage, making them accessible through Firebase and through Google Cloud. This allows you to download and download user files through Firebase SDKs and perform server-side processing, such as image filtering or video transcoding using the Google Cloud Platform [10].

Firebase SDKs to integrate with Firebase Authentication for user authentication, which allows you to set access control to individual files or group of files, so that the user can make the files public and private, at their own will. The Data Update Module provides a single access point to functions that provide data updates in the multi-level embedded structure, regardless of the level to which these data relate. The registration

and authorization module performs functions related to the authentication of users in the system and the control of access to data.

Main features:

- registration of users in the system;
- Authorization of users in the system;
- ability to change password, mailbox address;
- control access to private user information.

This module uses Firebase Authentication technology. In order to authorize a user in an application, you must first obtain login information from the user. These credentials may be the user's email address and password, or the OAuth token. These data are then transferred to the Firebase Authentication SDK. The server checks these credentials and returns the response to the client. After a successful authorization, the user can access the basic information about his profile, and the developer can control the user's access to the data stored in the database [11].

The analysis module of the data obtained allows you to analyze the collected travel data and draw diagrams that reflect the travel statistics. The HelloGraphLibrary library is chosen to build the charts, since it has expanded documentation, graphics, and a clear interface [12].

To use Firebase services, a Firebase account and a new project in it were created. Since the Firebase database exists in the Firebase console. You can also specify restrictions on user properties such as name, password, email address etc. in the settings. Firebase is used by listeners to monitor changes in a particular node. This is similar to event handlers in the sense that the code runs under certain precise circumstances. In our case, if changes in the data of the site occur, the listener automatically provides an update to the application, this is called snapshot.

The application may use this information to update the user interface. There are several types of listeners in Firebase, and each of these listeners has its own kind of inverse call that starts when the listener is activated.

- ValueEventListener;
- ChildEventListener.

The ValueEventListener expects data changes in a specific place in the database, nodes. ValueEventListener has one inverse call for the onDataChange () event to read static snapshot content at a specific address. This method is called once when the listener is tied up and every time the data, including the child nodes, is updated. An inverse call sends a snapshot containing all the data in that node, including child nodes. If no data is available, snapshot returns null. If the event can not be performed,

the following method is called onCancelled () [13].

The Firebase database makes it easy to read and write a POJO with its built-in serialization capabilities. BigQuery and Firebase Analytics are used to analyze data from users. Integration with BigQuery is very relevant to the application being developed, since this service allows you to handle very large data sets in a short time. Firebase Analytics automatically captures the following main events:

- first\_open – an event when users launched the application for the first time after installing or reinstalling;
- in\_app\_purchase – an event where users made an app purchase through the App Store or iTunes or Google Play;
- user\_engagement – the frequency with which the application was in the foreground;
- session\_start – an event where users used an app for a time that is longer than the minimum session after a period of inactivity that exceeded the session timeout;
- app\_update – an event where the user updates the application to the new version and launches it again;
- app\_remove – an event when an application is uninstalled or removed from a mobile device;
- os\_update – an event when the user's operating system has been upgraded to a new version;
- app\_exception – an event when an application stops working or causes an exception when using a user [34].

In addition, there is the possibility of creating a developer of his own event. During the development of the application, the following events were created:

- select\_content – an event when a user browses this or that informative page in the application;
- add\_mark – an event where a user adds or changes the rating of a infrastructure object;
- add\_road – an event where a user writes his own geo track;
- add\_message – an event when the user writes his own comment to the infrastructure object.

In addition, BigQuery imports the following data by default:

- the exact date and time of the event;
- id user.

Also, the developer can add his own user properties and event properties. For each Firebase

application that is configured to integrate with BigQuery, the data set is added with the name of the package or bundle ID at the base of the name.

Firestore Analytics integration with BigQuery is configured in Firestore Console. For each day of export, a separate table is created. This table has the format "app\_events\_YYYYMMDD". In addition, there is a table in which data is exported for the current day. It's called "app\_events\_intaday\_YYYYMMDD" and is filled in real-time when events are gathered. Each

row in the table corresponds to the bundle of the event loaded with the Firestore SDK [14].

All user parameters are stored together in the same line. Instead of performing any type of join over a separate user\_properties table, all user properties are included in one line as an array of structures. Event options are included inside an event as an array of structures. And the events themselves are also stored inside the array. One single data line in BigQuery will always contain 2 or more Firestore Analytics events recorded together (fig. 3).

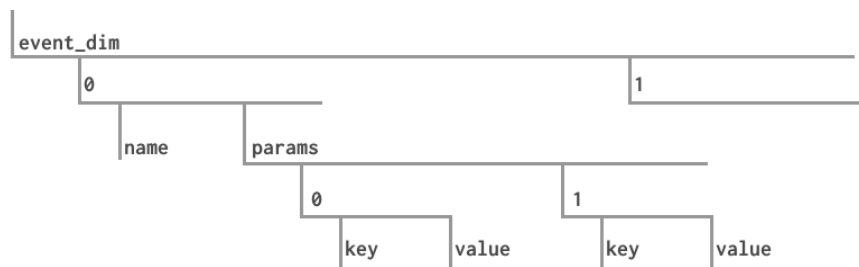


Fig. 3. The structure of the event parameter table in BigQuery

That means a single row in BigQuery can contain an array of user properties or an array of events that, in turn, can have their own array of properties of events. As mentioned in the previous section for writing queries to tables used StandardSQL. To work with arrays, use UNNEST function. This function takes an array and splits it into individual elements.

### The smoothing algorithm of geo tracks the positioning

Accuracy of GPS receivers under ideal conditions ranges from five to ten meters. In fact, however, as a rule, much worse than a few sources of error.

For example, there may be less than four satellites in view that are required for accurate calculation of three-dimensional position. Even in the presence of a sufficient number of satellites in view, they can be badly located, resulting in a high position dilution of precision values (PDOP).

This leads to the identification of such a GPS location, which differ greatly from the actual status of the receiver. There is also the issue of the so-called warm / cold start. Also, there are random errors caused by, for example, through the specifics of the satellites or receivers, atmospheric and ionospheric disturbances, multipath signal reflection, or blocking the signal.

The GPS signal is reflected off buildings, walls or surfaces and the position of the GPS are often scattered around the actual position of the

receiver. Lock signal, on the contrary, leads to the fact that the GPS points are not enough. All these problems require an algorithm that allows to approximate the points obtained during the trip to build geo track on the map. To smooth the path the application uses a smoothing with a Gaussian kernel.

### Conclusion

When designing software for collecting and analyzing data on moving mobile users, Google Maps was selected as a mapping service, as this service: it has the best among competitors for the coverage of the globe; it has various variants of landscape display (map, satellite, map, public transport); it has an API that is easy to integrate with the Android platform and Java programming language. Android Studio was chosen as the development environment, since this IDE has the most functionality among its competitors. To analyze the data received in the application, Firestore Analytics and BigQuery technology were selected. The automatic synchronization of these two services is configured. The main advantages of using these technologies are: the ability to create advanced reports by merging data from several other resources, such as CSV files, Google AdWords, or their own MySQL databases; no need to think about server size and data security; BigQuery is very fast, allowing you to query for large multi-year data and get fast results; Firestore captures some events automatically, such as First\_open, app\_updated, etc.; using complex



SQL queries you can get reports of any complexity.

#### References

1. BigQuery [Електронний ресурс]. – Режим доступу: <https://goo.gl/WjWk5o>.
2. What is BigQuery [Електронний ресурс]. – Режим доступу: <https://goo.gl/b9tFvn>.
3. Cloud database [Електронний ресурс]. – Режим доступу: <https://goo.gl/4nBIuу>.
4. Mobile app backend services [Електронний ресурс]. – Режим доступу: <https://goo.gl/x5yFyR>.
5. Firebase Cloud Messaging [Електронний ресурс]. – Режим доступу: <https://goo.gl/SaL8v4>.
6. Firebase Authentication: simple, free multi-platform sign-in [Електронний ресурс]. – Режим доступу: <https://goo.gl/dYf9nn>.
7. HelloGraphLibrary [Електронний ресурс]. – Режим доступу: <https://goo.gl/Fn5aG3>.

8. Firebase Retrieving data [Електронний ресурс]. – Режим доступу: <https://goo.gl/6abQ9B>.
9. Firebase: Writing POJOs [Електронний ресурс]. – Режим доступу: <https://goo.gl/7aEwQc>.
10. Firebase: Log events [Електронний ресурс]. – Режим доступу: <https://goo.gl/nZYcWR>.
11. Importing Firebase Analytics Data Into BigQuery [Електронний ресурс]. – Режим доступу: <https://goo.gl/CvTsgt>.
12. BigQuery tip: The UNNEST Function [Електронний ресурс]. – Режим доступу: <https://goo.gl/Mjccnv>.
13. Processing GPS Raw Data Without Additional Information [Електронний ресурс]. – Режим доступу: <https://goo.gl/D1EaGd>.
14. Firebase Test Lab for Android Robo test [Електронний ресурс]. – Режим доступу: <https://goo.gl/JWNGmW>.
15. Google Analytics for Firebase [Електронний ресурс]. – Режим доступу: <https://goo.gl/wQvJGG>

#### Information about authors:



**Oleshchenko Liubov** – PhD (Engineering), Senior Lecturer of Computer Systems Software Department, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”. Scientific interests: information technology in transport systems, mathematical modeling, computer networks, programming.

**E-mail:** oleshchenkoliubov@gmail.com



**Lapchuk Daryna** – Student of Computer Systems Software Department, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”. Scientific interests: programming, client-server technologies, creating mobile applications.

**E-mail:** dalapchuk@gmail.com