Information System Development For Recording Offenses In Smart City Based On Cloud Technologies And Social Networks

Myroslava Bublyk¹, Yurii Zahreva², Victoria Vysotska³, Yurii Matseliukh⁴, Lyubomyr Chyrun⁵, Olha Korolenko⁶

¹Professor and Deputy Head of Management and International Business Department, Lviv Polytechnic National University, Stepan Bandera Street, 12, Lviv, Ukraine. ORCID: 0000-0003-2403-0784.

²Master's Student of Information Systems and Network Department, Lviv Polytechnic National University, Stepan Bandera Street, 12, Lviv, Ukraine. ORCID: 0000-0002-1057-4126.

³Associate Professor and Deputy Head of Information Systems and Network Department, Lviv Polytechnic National University, Stepan Bandera Street, 12, Lviv, Ukraine. ORCID: 0000-0001-6417-3689.

⁴Postgraduate Student of Information Systems and Network Department, Lviv Polytechnic National University, Stepan Bandera Street, 12, Lviv, Ukraine. ORCID: 0000-0002-1721-7703.

⁵Associate Professor of Applied Mathematics Department Ivan Franko National University of Lviv, University street, 1, Lviv, Ukraine. ORCID: 0000-0002-9448-1751.

⁶Associate Professor of Economics and Business Administration Department, Kryvyi Rih National University, Vitaliy Matusevych Street, 11, Kryvyi Rih, Ukraine. ORCID: 0000-0002-0771-4298.

Abstract
An information system for recording offenses based on cloud technologies is designed and developed. An overview of the current research situation was conducted, and the future bot's fundamental properties - convenience and security- were identified. The use of Telegram represents all these features. We considered systems that the police have long used to record and investigate crimes to analyze the main necessary functions. It gave us an understanding of...
what data should be collected and in what form it should be stored in the future so that employees can use it. When performing this work section, an information system for recording offenses based on cloud technologies was designed. For this purpose, a goal tree was built that reflects the main goal and sub-goals, having achieved which, the main goal will also be performed. The next step was to create a context diagram in IDEF0. It displays all the data that the system receives and processes, and its decomposition reflects the interaction between processes. Also, a hierarchy of tasks was created for the projected plan. Defined four subtasks and functions necessary for the system's proper functioning are described. During the implementation of this section, we will consider the leading technologies that will be used to create Telegram bots for fixing offenses, among them Node.js, TypeScript, DynamoDB, AWS Lambda, Claudia.js. The main advantage is the serverless architecture, which scales and is charged directly to the system load. It will help you avoid system failure entirely if there are excessive requests higher than the average value. During the implementation of this section, we considered a control example of a software product. The analysis showed that the offense recording chatbot is working and performs its direct functions: it receives reports of offenses, tells information about the author, and accepts user questions. In addition, the structure of the database that stores the user's personal data and possible questions was presented.

**Keywords**
Chatbot; Telegram; Information System; Cloud Technologies; Social Networks; Offenses

**Introduction**

In recent years, there has been increasing use of computers in the police as a module of Smart City (Hassan et al., 2019; Bang et al., 2019; Baran et al., 2021; Jalali et al., 2015; Laufs et al., 2020; Jelokhani-Niaraki et al., 2020; Eckhoff et al., 2017; Matseliukh et al., 2021) and Smart House (Moraes et al., 2001; Brown et al., 2007; Filho et al., 2010; Risteiu et al., 2016; Mohammed et al., 2017; Lytvyn et al., 2019; Lytvyn et al., 2020). It is not only in general management (for example, payroll, personnel, e-mail, office suites, etc.) but also as a tool to facilitate the main business activities of the police in crime prevention and detection (O'Donnell et al., 2019; Zamanirad et al., 2020; Koops et al., 2018; Landis et al., 2014; Lopez et al., 2021). The increase in crime rates and the increased awareness of modern criminals entails existing methods and techniques (Chen et al., 2017; Zdraveski et al., 2017; Khan et al., 2012; Kuzmin et al., 2016; Açar et al., 2017). Therefore, there is a clear need to use advanced software technologies to best use limited resources. Many existing systems are internal developments of individual police forces or commercial products. In the world's law enforcement agencies, the total number of crimes has increased by 15.1% over the past decade, while the number of police agencies has increased by only 7.7%. Therefore, software development that would help fix offenses and facilitate activities would be relevant for police officers. The work aims to design and develop a dialog system (Bisikalo et al., 2016; Kubinska et al., 2021; Aksonov et al., 2021) as a chatbot for recording violations based on cloud technologies. The object of the work is the processes of database design and development of crime detection software based on cloud technologies. The work subject is methods and tools for designing and developing a dialog system for recording offenses based on cloud technologies. Completing a work task requires solving the following tasks:
1. Analysis of the current research situation;
2. Familiarisation with systems that can already partially solve this problem;
3. Creating schemas that take into account information flows that detail the system’s functionality;
4. Software implementation of an information system for recording offenses based on cloud technologies.

The practical value of the information system lies in the ability to record and assist law enforcement agencies in investigating any administrative, legal, or other crimes.

Related works

First, let’s look at using chatbots, namely the Telegram bot, to help fix offenses (Bauer et al., 2019; Nayak et al., 2021; Pamela et al., 2021; Tombs et al., 2014; Marino et al., 2014; Chin et al., 2020). Telegram bots are dialog system based online behavior analysis of users in Social Networks inspired by artificial intelligence that can perform many functions: send relevant weather information or helpful news articles, schedule reminders, play ringtones, create to-do lists, and much more (Intyaswati et al., 2022; Dahiya et al., 2022; Raja et al., 2022). Such bots work in Telegram, a popular instant messaging app used by millions of people worldwide. Telegram's security is strong: users can enable end-to-end encryption and communicate via secret chats. This feature is not the only attractive aspect: bots are becoming increasingly popular as third-party applications run in the Telegram environment. However, it is also necessary to consider the very essence of the use and necessity of computer systems in police work. Most, if not all, of the current manual and computerized systems, revolve around investigating crimes already committed. Therefore, they are reactive. Most police forces use various relational database management systems (DBMS) to record and further analyze crimes. Standard or interactive queries are written to create corruption, delinquency, and various statistical models. Software development to perform fingerprint matching, crime scene reconstruction, and vehicle license plate recognition has been several functional studies over the years. These systems can be used as tools for large-scale or large-scale crimes, but their specific nature (i.e., a well-defined purpose) is beyond the scope of the current review of crime analysis. They are also well covered in academic literature, and an interested reader should read the cited literature for details. An investigation of an underlying crime generates much data and information about a particular crime or series of crimes. This data can overwhelm anyone. Most crimes are not severe. Individual crimes are less strict for society as a whole (although victims are still seriously affected). The volume of such crimes creates various problems for the investigator. The high degree of similarity of crimes and the lack of resources for a detailed investigation of any particular crime solve the crime and prevent similar future crimes. The challenge for advanced software developers is to apply state-of-the-art template matching and visualization techniques to fix the problem, for example, based on Chatbot (Minhas et al., 2021; Park et al., 2019; Nuruzzaman et al., 2020; Kim et al., 2021; Chandel et al., 2018; Tu, 2020; Shakhovska et al., 2019). There is a significant difference between the requirements for registering and managing a crime and subsequent investigation. Running complex speculative searches on an online transaction-processing computer will only slow down the system and reduce its operational efficiency, so transferring all data to a warehouse for offline processing is recommended. In a warehouse environment, it will be
possible to combine data from different sources, i.e., crime, management, and control, guardianship, courts, etc.; improve data by creating new fields based on one or more existing features, i.e., linking the time of day of the crime occurred, etc.; and also ensure consistency of data elements in different systems, i.e., road names, etc. in Smart city (Ramaprasad et al., 2017; Rassia et al., 2017; Kirimtat et al., 2020; Lau et al., 2019; Krislata et al., 2020; Katrenko et al., 2020; Lytvyn et al., 2015; Matseliukh et al., 2020; Boreiko et al., 2017; Boreiko et al., 2016; Teslyuk et al., 2018; Podlesna et al., 2020; Bodnar et al., 2020; Makara et al., 2020; Bublyk et al., 2020).

In sizeable criminal series, it is necessary to look for connections in two or more crimes when there is a solid previous belief that the same person(s) committed them. Even the most demanding computer user finds it very difficult to choose explicit communication models for individual crimes regarding large-scale corruption. It is difficult to determine the specifics of a crime in volumes with no clear suspects because specific crimes are lost in the maxis of all registered crimes. From the perspective of crime investigation, several features would be most desirable in any computer system, the main one being interacting with disparate existing systems. Currently, it is quite common to check 4 or 5 different methods to establish a history and find out about a single address or person. A visual representation of information that highlights spatial or temporal relationships based on Big data analysis (Lutskiv et al., 2020; Berko et al., 2018; Berko et al., 2019; Bublyk et al., 2021; Shakhovska et al., 2015; Shakhovska, 2017; Shakhovska et al., 2018; Shakhovska et al., 2019; Duda et al., 2020) is most desirable. Although several commercially available packages are advanced in this area, it is believed that the host system itself should be capable of such representation. The ability to use natural language methods and technologies (Khairova et al., 2019; Khairova et al., 2020; Zhezhnych et al., 2018; Bisikalo et al., 2017; Khomytska et al., 2020; Davydov et al., 2017) for queries and tools for any submitted query can search through all tables in the database without the need to establish relationships (Shakhovska et al., 2018; Khomytska et al., 2020). It would bring a complex search in the domain of an untrained user. A sophisticated indexing system will also be required to ensure an adequate search time. When data is entered into the system, a helpful tool automatically matches any new information to existing information based on user-defined criteria (Matseliukh et al., 2021; Dewobroto et al., 2022). Artificial intelligence techniques are widely used in various commercial products and basic computer software applications, such as fuzzy logic controllers and neural network pattern recognition (Babichev et al., 2020; Berko et al., 2020; Lutvynenko et al., 2020; Sachenko et al., 1999; Golovko et al., 2000; Komar et al., 2013; Tsmots et al., 2015; Klymovych et al., 2020; Danylyk et al., 2020; Smaida et al., 2020; Neskorodieva et al., 2020). Similar different AI methods can be embedded in investigative systems to present the results to the investigator automatically (Wang et al., 2019; Okrenets et al., 2017; Babichev et al., 2018; Semenova et al., 2019; Babichev et al., 2020; Ivanov et al., 2020; Safonyk et al., 2021; Demchuk et al., 2019). The information system plays as a tool based on Chatbot (Modrzyk et al., 2018; IT_everyday, 2021; Telegram, 2021; Dialogflow, 2021; Google, 2021) a vital role in investigating crimes of both significant and mass crimes. Most modern systems are designed for research in any of these categories, but not both, except i2 and Watson, and were written specifically for market niches. Each method works well, judging by its purpose. However, the systems mainly rely on users asking specific questions via Telegram or a Web
application (Baumgartner et al., 2015; Modrzyk et al., 2016) to help with the investigation process. In particular, within the scope of a crime, the user may not know which issue is relevant and cannot effectively link similar crimes in a series.

In both crimes, the human investigator leaves the right questions to the system or interprets the processed information. It requires increased complexity and specialization to get the best software results and gain confidence in the procedures. However, this is difficult to achieve in major crimes due to data's relative rarity and volume, so honing such skills is difficult. For best results, most forces use specialized teams of detective officers supported by trained civil service staff, including trained crime analysts. Such specialization takes time to develop and can vary in the degree of ability for different aspects of crime. Future systems should now use artificial methods that allow automatic learning, formation, and content presentation. However, the police have viewed artificial intelligence systems with great skepticism in the past. It is partly due to a general lack of computer literacy and procedures that require a significant amount of manual knowledge, which users may disagree with, or ability becomes outdated and challenging to update.

In the coming years, thousands of new "smart" video cameras will appear on the streets of Ukrainian cities. It is expected that under their close supervision, the lives of Ukrainians will become safer. The first results are already available: with the help of new technologies, it has been possible to improve the crime situation in frontline cities in eastern Ukraine and the capital. Modern "smart" camcorders easily identify human faces. After recognizing a person from the database, the system instantly displays data, name, and position. With the help of an intelligent video camera, you can quickly find the right person in the crowd. And also to determine the age, route, and even assess the emotional state.

Fighting detectors and psycho-type detections are expected to appear in the capital soon to indicate inappropriate behavior. And the more often a person gets into the camera lens, the more accurately it will be identified. The Smart City concept (Mykich et al., 2016; Burov et al., 2020; Zhezhnych et al., 2018; Zhezhnych et al., 2017; Zhezhnych et al., 2007) based ontology and extensive data analysis has been implemented in Kyiv for several years, partly in the Donetsk region. Its various components are embodied in the experience of other cities around the world: "open budget" - from Boston, "Kyiv City Hub" and the accelerator of urban projects - from Amsterdam, the development of Wi-Fi - following the example of Barcelona and Seoul, e-ticket - from Tallinn, security - as in London and Tel Aviv.

**Materials and methods**

System analysis is performed to study the system or its parts or future design. It is aimed at identifying the goals of the system. This problem-solving technique improves the system or ensures that all its components work efficiently to achieve their goal. The analysis determines what the system should do. One of the best ways to start analyzing a system is to create a goal tree. A goal tree is a diagram used to define criteria and achieve a general goal. Based on the general purpose of creating or improving the system and other needs identified through research, documentation, and other tools, the goal tree is constructed to present the main goal in more specific purposes. Achieving each "smaller" goal brings you one step closer to
achieving the overall plan. At this point, the goal is defined by criteria that provide some measurable requirements for decision-making. The goal tree is hierarchical, where a specific purpose follows from objective needs. It means that there is no way to achieve top-level goals without first reaching a lower level. Thus, the achievement of the general goal is gradual. Usually, each plan should be specified by who will complete it or when it will be completed. It leads to the fact that the expected result's general goal needs to be subjected to a detailed decomposition with the ability, if necessary, to perform several levels of pollution.

Figure 1. Goal Tree of a cloud-based crime detection information system

There are options when each existing and the already defined goal must be divided into sub-goals that specify the task. In turn, they are still divided into sub-goals and so on until the last element is so specific that further decomposition will not be possible. The general goal of the designed system is "An information system for recording offenses based on cloud technologies." The available plan is divided into sub-goals, which are defined as follows: collecting documentation and analyzing the results obtained, planning and designing the system, developing and launching the scheme. These goals represent the next level:

- For "Collection of documentation and analysis of the results obtained," the next-level sub-goals are "User survey and review of comments on analogy systems" and "Review of ISO standards for the storage and processing of users personal data."
- For "System planning and design" – this is "Developing customer-service relationships" and "Planning the necessary tools for creating an architecture."
The last sub-goal, "System development and launch" decomposition, is the following sub-goals: "User interface development" and "Creating and connecting a data warehouse for storing user's data."

Further, criteria were defined for second-level purposes. For "User survey and review of comments on analogy systems" – this is relevant, and for "Review of ISO standards for the storage and processing of personal data" – safe and secure. The following sub-goal, "Developing customer-service relationships," has the following criterion – lightness. "User interface development" must meet the criterion User Friendly. For "Creating and connecting a data warehouse for storing user's personal data," the structure criterion is defined. Notation is selected to specify and describe all system functions IDEF0. Integrated functional modeling definition (IDEF0) is a function modeling method for business systems and an engineering approach for needs analysis. The IDEF0 method uses a window to represent functions and shows the relationship between the child and parent systems. Using this notation, you can get a diagram that will help you better understand the organization of the projected plan. The advantages of using this method are:

- ensuring perfect system analysis;
- providing improved communication methods;
- helps you understand the system and its connections.

Models are descriptions of procedures. To fully describe an information system, you usually create several models or diagrams, each from a different perspective. That is, they work on each one separately. IDEF0 diagrams consist of processes and arrows that are used to indicate input and output data, controllers, and mechanisms. Indicators describe the interaction of processes (systems) with the outside world. Input data is submitted to the system login by an external entity, such as a user, administrator, or others, and processed during the system's operation.

Output data is data that the system processes during operation and outputs to meet the needs of certain entities, such as a user, administrator, or others who work with this system.

Supervisors are specific standards, a set of rules, laws, or procedures that govern the actions of a process. They affect processes, but processes do not process them as in previous cases. There are precedents when the goal of a system is to change rules or standards through one of its processes. They are then served as inputs, not controllers. System mechanisms are specific resources that perform certain activities, such as system initialization. They can be either real people or particular objects. Rectangles indicate processes on diagrams. The name of the process must necessarily begin with a verb since processes are executors of a specific action, for example, changing or processing information. Diagrams are constructed for this system using IDEF0. The first is the context diagram (Fig. 2). The context diagram shows one process – "commit violations" – that will be further decomposed (detailed). The input data for this system is:

- initiating login – the system user must find the design and open it for further operation;
- initiating system start-up – the user needs to start the system for additional operation;
- interface language selection – for convenience, the user will be given a choice of interface language;
- entering a mobile number – for the system to work correctly, the user needs to enter some personal data like a phone number;
- select the "About Author" menu option – by choosing this option, the user can find out the name of the system author;
- select the "Ask a question" menu option – by choosing this option, the user will be taken to a page where they can send a question;
- question text – to get an answer, the user must provide the system with the question text;
- notification of an offense – to report a violation, the user needs to go to the appropriate menu item;
- sending personal data – the user must provide the following passport data: first and last name for the system to record an offense report.
- sending a photo of the offense – to certify the crime and further process it, the user must provide a photo of the incident;
- providing access to the location – to record the scene of an accident, the user needs to share their place with the system;
- sending a text about the details of the offense – also, the user needs to describe all the events that occurred and send this text to the system.

Since the system requires some of the user's data to work and process user requests, the controller for this system will be the ISO/PC 317 standard, which contains a list of all the necessary rules for protecting the user's data when using the system designed in this work.

Output data:

- notification of system features and functions – when logging in, the user sees a list of system features;
- providing the user with a language selection option – for personalization, the user is allowed to choose the interface language;
- notification of the need to enter a mobile number – the system notifies the user that they need to enter their phone number to continue working;
Mechanisms have already been mentioned earlier. They are the user and administrator because they directly impact the system. The next step is to decompose the primary process. As a result of the split, four strategies appear (Fig. 3): log in to the system, get the user's data, provide information about the system, and accept an offense report. Each process has its input and output data, controllers, and mechanisms. The login process includes the following data:

- **input data:** initiating system start-up, initiating login;
- **output data:** notifications about the system's capabilities and functions, providing the user with a language selection option, reports about the need to enter a mobile number, waiting for user responses;
- **this process has one mechanism:** user;
- **no controllers are defined for this process.**

Process Get the user's data. It has the following data (Fig. 3):

- **input data:** select the interface language, enter a mobile number;
- **output data:** confirmation of receipt of the number;
- **This process has one mechanism, and that is the user;**
- **ISO/PC 317 is the only controller for this process.**
Process Provide information about the system. It has the following data:

- input data: select the "About Author" menu option, select the "Ask a question" menu option, question text;
- output data: providing information about the author of the system, waiting for a user's question, answering a question from a live administrator;
- this process has two mechanisms: user and administrator;
- there are no controllers in this process.

Process Accept a report of an offense it has the following data:

- input data: notification of a crime, sending personal data, sending a photo of the offense, providing access to the location, sending a text about the details of the offense;
- output data: request for personal data of the user (last name and first name) request for a photo of the crime, demand for the user's location, request for details of the offense, notification of the beginning of the processing of the offense;
- the user is the only mechanism of this process;
- ISO/PC 317 is the only controller in this process.

We also decompose the process. Provide information about the system. The following details are defined for this process: get information about the author, go to the live administrator, get questions from the user. Process Get information about the author; it has the following data:

- input data: redirect the user to the main menu, select an option "About the author";
- output data: providing information about the author, returning to the main menu;
this process has one mechanism, and that is the user; 
there are no controllers in this process.

Process Go to live admin it has the following data:

- input data: select a menu option "Ask a question";
- output data: waiting for the user's question;
- for this process, the user is a mechanism;
- there are no controllers in this process.

Process Get a question from the user; it has the following data:

- input data: question text;
- output data: answer questions from the live administrator, go to the main menu;
- this process has two mechanisms, and these are the user and the administrator;
- there are no controllers in this process.

The process is described in detail below «Accept a report of an offense.» As a result of decomposition, five sub-processes were obtained: go to the report an offense Section, get the user's passport data, get a photo of the offense, get access to the user's location, get details of the offense. For the process, go to the report violations section. The following data is defined:

- input data: go to the main menu, report violations;
- output data: request for personal data of the user (last name and first name), waiting for personal data;
- mechanism: user.
- no controllers are required for this process.

Figure 4. Process detail «Provide information about the system» (third-level decomposition)

For the process get the user's passport details, the following data is defined:

- input data: sending personal data;
For the process get a photo of the offense, the following data is defined:

- input: sending a picture of the offense;
- output data: request the user's location;
- mechanism: user.
- no controllers are required for this process.

For the process to get access to the user's location, the following data is defined:

- input data: providing access to the place;
- output data: request for details of the offense;
- the only mechanism of this process is the user.
- there are no controllers in this process.

![Diagram](http://www.webology.org)

**Figure 5. Process detail «Accept a report of an offense» (third-level decomposition)**

Process get details of the offense it has the following data:

- input data: send a text about the details of the crime;
- output data: notification of the beginning of the processing of offenses;
- the user is the only mechanism in this process.
- there are no controllers in this process.

A crucial step in system design is to create a hierarchy of tasks. The created diagram shows the structured process of system design and development. It also allows you to see the necessary functions and procedures that should be performed in the designed system. The

http://www.webology.org
main task is "Record violations." It is divided into four subtasks; they can also be called the four main modules of the system: log in to the system, get the user's data, provide information about the system, and accept a report of an offense. The first two subtasks do not have specific functions necessary for them to work correctly. For the third function, as many as three are defined: get information about the author, go to the live administrator and get questions from the user. They will help ensure that the user gets all the necessary information for the convenient operation of the system. The fourth subtask has five functions that will help you work correctly: go to the notify user section, get the user's passport details, get a photo of the offense, access the user's location, and get information about the crime. All functions of this subtask are necessary to ensure that the message about a specific offense is processed.

![Task hierarchy](https://example.com/task_hierarchy.png)

**Figure 6. Task hierarchy for «Information system for recording offenses based on cloud technologies»**

Chatbots are becoming more popular than ever. The global Chatbot Market is expected to grow exponentially between 2016-2023. 85% of customer interactions will be managed without people by 2020, 32% of CEOs say voice recognition is the most common artificial intelligence technology in their business, 6 billion connected devices are actively asking for support by 2018, 44% of managers believe that the most crucial advantage of artificial intelligence is "automated communication that provides data that can be used for decision-making," intelligent agents will manage 40% of mobile interactions by 2020. Chatbots provide a huge opportunity to communicate and engage your customers. 83% of people on the Internet need support while working. The support process can be improved by using chats. They can be programmed to automatically answer repeated questions and direct the request to a natural person when more complex actions are required. This way, the agent focuses more on essential cases rather than a simple question. Companies use "passive customer interaction," meaning they only respond to connected customers and do not initiate communication. Chatbots can also direct customers to responses that will help them, which leads to ticket rejection and increased customer satisfaction. In addition, thanks to the Telegram Bot API, you can easily create bots, promote them, or generate them to make your instant messages more convenient. Bots started appearing after Telegram announced a new bot API in 2015. It allows third parties to create bots that use this messaging app as their primary interface. The list of Telegram bots is long: many developers have experimented with the API provided to create unique bots for various purposes. In short, Telegram bots are programs designed to perform certain functions, follow instructions, and interact with users.
Bots work in the Telegram environment and do not require an additional installation procedure.

By using bots, people can improve the functionality of Telegram. Therefore, using this approach will be very convenient for employees. After all, they need to have their smartphone and internet access. With the manager's help, they will enter all the necessary information and then process it in the required processes. The Telegram bot is a chat designed explicitly for use on the Telegram Messenger platform. These bots work based on artificial intelligence and provide several applications to benefit customers or their target audience. Several companies use and develop Telegram bots, including Zoom and Sisense. One way to use Telegram bots is to provide services to Telegram customers, which simplifies the development and design process for creating a chatbot thanks to its API.

**Experiments, results, and discussion**

Telegram chatbot for recording offenses is designed to collect data and describe crimes, with their location for law enforcement agencies' further processing of this message. It was created using: Node.js, TypeScript, DynamoDB, AWS Lambda, Claudia.js.

The database contains the necessary data for further processing of offenses (Fig. 7). The following user data is stored in the database: FIRST_NAME, IS_ADMIN, LAST_NAME, PHONE-NUMBER, TELEGRAM_ID, USER_LANGUAGE, USER_STATE, USERNAME. The user's last name and first name provided must correspond to those entered in their passport. The user can choose the language themselves. In this version of the bot: Ukrainian or English.

![Database Tree](image)

**Figure 7. Saving user data to the database**

In addition, the user's questions are entered into the database (Fig. 8). There is an item in the menu that the user can click on to ask questions. It consists of the following lines: FIRST_NAME, ID, QUESTION, RESOLVED, TELEGRAM_ID. Therefore, to process the inquiry, the system needs to know the user's name, identification code in the system, the text of the question, or the question that has already been processed - the user's identification number in the Telegram. Waiting for the answer to the question is not clearly defined.
This system will help the Ministry of internal affairs employees reduce the load on call centers by implementing a chatbot system that automatically processes appeals to the police. Recording an offense and calling a patrol car will be carried out via live chat. The main advantage of the service is the serverless architecture, which is scaled and charged directly in proportion to the system load. It will help you avoid system failure if there is an excessive influx of requests.

The following elements are required for the correct functioning of the offense recording chatbot: Telegram Chatbot, API Gateway, Lambda Node.js, S3 Bucket, Dynamo BD (Fig. 9).

Each user of computing resources can register on a particular site, log in to the corresponding group (for example, the "Students" group) and get a set of unique keys for further use. A resolution is a unique string associated with the user-owner and the service for which this key is intended.
The system stores are sets of restrictions for the keys of each user group. Limitations on key usage are expressed in the number of requests per second, per day, and month. This approach allows you to avoid too high a load on the computer system used. If the user accesses the computer system API too often, they receive a response time, after which they will be able to make the request again. Each user (if they have the right to do so) can create more than one key for themselves.

The system uses open projects for service purposes: the Redis key-value repository and the Varnish Cache caching application server. The gateway API system consists of four subsystems, each of which can be used separately from the other. These are user accounting subsystems, a service management subsystem, an access control subsystem, and a key management subsystem. The user accounting subsystem allows users to register and receive keys to access the API. The subsystem is written in Ruby and can use any relational DBMS as a repository. The service management subsystem is designed to administer the list of users, groups, and their rights. The service management subsystem also adds new computing nodes in the cluster. To do this, just specify the address of the added node and select the service provided on this node. The access control subsystem is an add-on to Varnish Cache caching applications. The Add-In allows Varnish Cache to interact with the Redis key-value store. Such a bundle will enable you to process more than 10 thousand requests per second. It will enable you to use several servers as a frontend since Redis also provides a distributed mode of operation. At the same time, using Redis will enable you to achieve a large server bandwidth. The critical management subsystem links the user accounting subsystem and the access control subsystem. Subsystems use a simple REST API to exchange data about keys, downstream services, and restrictions.

AWS Lambda: it is a computing service that allows you to run code for almost any type of application or server service—all without the need for administration. AWS Lambda performs all administration, including server and operating system maintenance, resource allocation, automatic scaling, code monitoring, and logging. All you have to do is provide the code in one of the languages supported by AWS Lambda. DynamoDB is a database of key-value pairs and documents that provides a delay of fewer than ten milliseconds when working at any scale. It is a reliable, fully managed database for Internet-wide applications that operates in multiple regions with multiple leading servers and has built-in security, backup, and restore tools, as well as in-memory caching. DynamoDB can process more than 10 trillion requests per day and handle peak loads exceeding 20 million requests per second. Many of the most advanced companies globally, such as Lyft, Airbnb, and Redfin, and large corporations such as Samsung, Toyota, and Capital One, use the scalable and high-performance DynamoDB service to perform mission-critical workloads.

Hundreds of thousands of AWS customers choose DynamoDB as a database of key-value pairs and documents for mobile, gaming, advertising, internet applications, IoT applications, and other applications that need access to data with minimal latency, regardless of scale. Logically, S3 is a global storage area network (SAN) that resembles a vast hard drive where you can store and query data. From a technical point of view, Amazon's architecture looks a little different. Data (asset) stored and received via S3 are called objects. Objects are stored in
memory areas (bucket). The analogy with a hard disk can illustrate this: objects are files, and memory areas are folders (or directories). Like on a hard disk, objects and memory areas can be found using a unified Resource Identifier (URI).

Telegram Chatbot, API Gateway, Lambda Node.js, S3 Bucket, Dynamo BD are the main parts of the created system, with which the system can allocate loads and resources.

This software product has the following minimum requirements for the user's technical means:

- version Android: 4.1 and higher;
- version iOS: 9.0 and higher;
- minimum required Ram: 100MB;
- processor: quad-core, 1.5 GHz;

To work with this Chatbot, the user needs to install the Telegram program on their device and create an account. Then go to search, find and add to this Chatbot.

The input data for the system will be:

- user's data: first name, last name, phone number, location;
- user questions;
- notification of violations;
- The source data, in turn, will be the following:
- answers to user questions;
- report that information about an offense has been accepted and submitted for processing.

This software product is intended for obtaining information about offenses. A chatbot will reduce the burden on Call Centres in law enforcement agencies and increase the efficiency of processing incoming requests by police officers. This software product will solve the following classes of tasks: ease of reporting offenses, providing detailed information about crimes, and reducing the load on Call Centres. The user will be able to quickly register in the system and be able to describe the emergency and send an image and provide access to their location. It will speed up the entire process of transmitting and confirming information about offenses and help identify groundless calls, increasing law enforcement agencies' efficiency. At the beginning of the operation, the system requires the user to download the Telegram app and confirm the user's identity. In addition, during the process, the system will need access to the user's location.

The system implementation results are presented for a visual inspection of the system's functionality and confirmation of its serviceability. First, the bot welcomes the user to the design and explains its purpose, and provides examples of situations in which you can contact them: an improperly parked car, a littered environment, and facts of corruption in state bodies, traffic violations, and violations of public order. To start working with the bot, you need to click the start button, form the system and allow the user to complete their task (Fig. 10).
Figure 10. Getting started with a chatbot created using the information system for recording offenses

After starting work, the user is given a choice: the system interface can be made in Ukrainian or English. This menu for selecting the language is shown in Fig. 11.

Figure 11. Welcome message and language selection

After the user has decided on the desired interface language, they still need to provide the system with their number to complete registration (Fig. 12).
The first step to report a violation is to select this item in the main menu. After that, the bot will send a message asking for the user’s first and last name, which are indicated in their passport. The next step is for the bot to confirm that a crime has occurred so that it will ask for a corresponding photo (Fig. 14). After the bot receives the image, it will ask the user about its location to fix the exact position in coordinates (Fig. 14). In addition, the last point will ask the user to write about all possible details that the latter remembers, such as the car number, the thief’s facial features, how the event happened, and so on.
Figure 14. The user provides the system with their passport data and requests the plan for photos and locations of the event.

After that, the system thanks to the user for the request and informs them that they will be contacted after processing the request (Fig.15).

Figure 15. Details provided by the user about the violation that the system is currently processing.
Conclusions

In police departments, computers, the Internet, and electronic accounting make information more organized, easily accessible, and most importantly, help you get your work done faster. Technical means of combating crime are increasingly being used. The Criminal Justice industry must use advanced software, tracking systems, etc. However, the technology used must be convenient and easy to use. During the work, many tasks were set, the implementation of which made it possible to implement a software solution in the form of Telegram bot, which helps in fixing offenses based on cloud technologies and perform the following tasks:

- analysis of literature and other sources, which allowed us to assess the current research situation;
- familiarisation and analysis of analogy systems that are already in use;
- performing a system analysis to record a set of functions that need to be implemented;
- software implementation of the solution through Node.js, TypeScript, DynamoDB, AWS Lambda, Claudia js.

The practical value of an information system for recording offenses is based on cloud-based recording technologies and assistance to law enforcement agencies in investigating any administrative, legal, or other crimes.

References

Babichev, S., Durnyak, B., Pikh, I., & Senkivskyy, V. (2020). An Evaluation of the


of computer applications in technology, 39(1-3), 32-39.
Klymovych, O., Hrabchak, V., Lavrut, O., Lavrut, T., Lytvyn, V., & Vysotska, V. (2020). The Diagnostics Methods for Modern Communication Tools in the Armed Forces of
Ukraine Based on Neural Network Approach. CEUR Workshop Proceedings, 2631, 198-208.
Lytvynenko, V., Wojcik, W., Fefelov, A., Lurie, I., Savina, N., Voronenko, M., Boskin, O., & Smailova, S. (2020). Hybrid Methods of GMDH-Neural Networks Synthesis and
Training for Solving Problems of Time Series Forecasting. Lecture Notes in Computational Intelligence and Decision Making, 1020, 513-531.


insurance industry. Knowledge-Based Systems, 196, 105810.
Shakhovska, N., Duda, O., Matsuik, O., Bolyubash, Y., & Vovnyanka, R. (2019). Analysis of the Activity of Territorial Communities Using Information Technology of Big Data Based on the Entity-Characteristic Mode. Advances in Intelligent Systems and
city platform architecture and dashboard. Pervasive Computing, 16(2), 35-43.