International Information Technology University JSC

Faculty of Information Technology

Department of Information Systems

**Final Project**

**«Software requirements specification development»**

**for discipline «Fundamentals of information systems»**

**“IS for Traffic police”**

**Authors: Kemelzhan Daulet**

**Kozhahmetov Alibek**

**Nogai Islam**

**Group: IT1-2013**

**Almaty, 2022Final Project**

**«Software requirements specification development»**

**For discipline «Fundamentals of information systems»**

**Aim of work** – development of full software requirement specification for development of information system.

**Symbols and abbreviations**

VI – Vehicle Info

MIA – Ministry of Internal Affairs

ID - Identity

JSK - Joint Stock Company

**1. INTRODUCTION**

VI - Vehicle Info versatile application created by the request of MIA of Kazakhstan. Application will be used both, for the Traffic Police and ordinary citizens, who owns a vehicle. VI will drastically make it easier to work for Traffic Police. It will give such opportunities as ID-ing the owner of vehicle or checking inspection term of the vehicle. As for the owners of vehicles, our application can show overall information about the vehicle, shows if there are any fines that need to be paid or invitations for vehicle inspection.

**2. GENERAL INFORMATION**

**2.1 Full name of IS and its abbreviation**

Full name of IS: **“Vehicle Info”**

Abbreviation: **“VI”**

**2.2 Information about developers and customers**

Developers:

Name: “Dayaganal” Company.

E-mail: [info@dayaganal.com](mailto:info@dayaganal.com)

Contact number: +7-(775)-674-75-05.

Address: Manas St. 8, Almaty 050000, Republic of Kazakhstan.

Customers:

Name: The Ministry of Internal Affairs of the Republic of Kazakhstan.

E-mail: [kense@mdv.kz](mailto:kense@mdv.kz)

Contact number: +7-(717)-271-51-89.

Address: Tauelsizdik Ave. 1, Nur-Sultan 010000, Republic of Kazakhstan.

**2.3 Project timelines**

Planned start: 05.02.2022

Deadline: 5.09.2022

**2.4 Funding**

Project is sponsored by the MIA of RoK.

**3. PURPOSE OF CREATING INFORMATION SYSTEM**

**3.1 Relevance**

Nowadays, a device that used by traffic police officers is outdated. It contains small amount of RAM, ROM and weak processors by the standards of 2022. Due to frequent complaints from officers, it was decided to create a mobile application that will replace old devices. The advantage of the project is to increase the efficiency of employees and reduce the cost of purchasing and maintaining technical equipment.

**3.2 Usage**

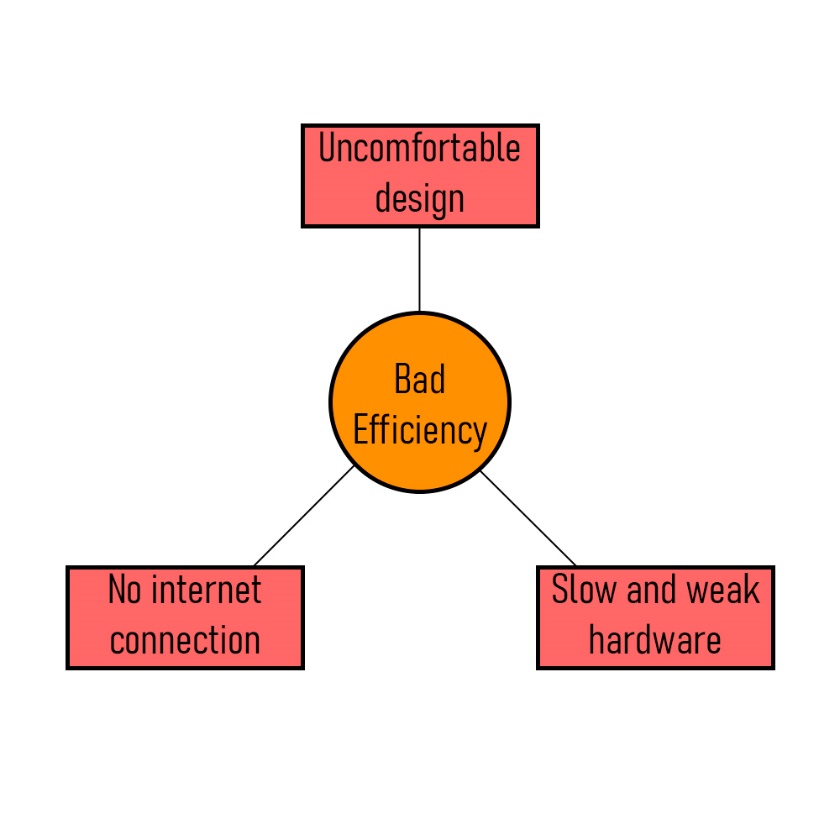
In the field: auto insurance, traffic laws.

Category: vehicles, law and order.

**3.3 Ideology**

Increasing the efficiency of work and improving provision of the information about with the help of new and faster smartphones. Introducing new model of Auto Service Center.

**3.4 Formulation of the problem**

****

**Figure 1.1. Formulation of the problem.**

**3.5 Formalization of the problem**

In case of malfunction or long processing of the request by the main device, the effective solution of given problem can be reached on condition that police officer could swap to personal smartphone and use installed application further for work.

**3.6 The goal**

Our main goal is to reduce the cost of the state for the purchase of new devices, as well as to make application more accessible to everyone and create a newer and more convenient model of Auto-CSC.

**3.7 Objectives**

* Analysis of existing databases on vehicle and citizen resources
* Development of a model
* Development of algorithms which based on user input/output actions.
* Development of IS based on law enforcement tablets.
* Testing IS for bugs and errors.
* Debugging of IS with further fixing it.
* Application of the project after all debugging process.
* Installation of the ready and final variant of the project.

**3.8 Advantages**

* The user can use the app without a network connection.
* Officer can replace device to application in any moment.

**3.9 Disadvantages**

* Without an Internet connection, there will be displayed only the last 5-10 requested vehicle information.
* If mobile device has less than 2 GB of RAM, the work of application will be less stable.

**4. SOFTWARE REQUIREMENTS**

**4.1 Requirements for the structure and functioning of the IS**

**4.1.1 Software technology used**

Languages: Java

Frameworks: IntelliJ IDEA

Front-end: XML

Database: SQLite

DevOps: AWS (Amazon Web Services)

**4.1.2 IS Model**

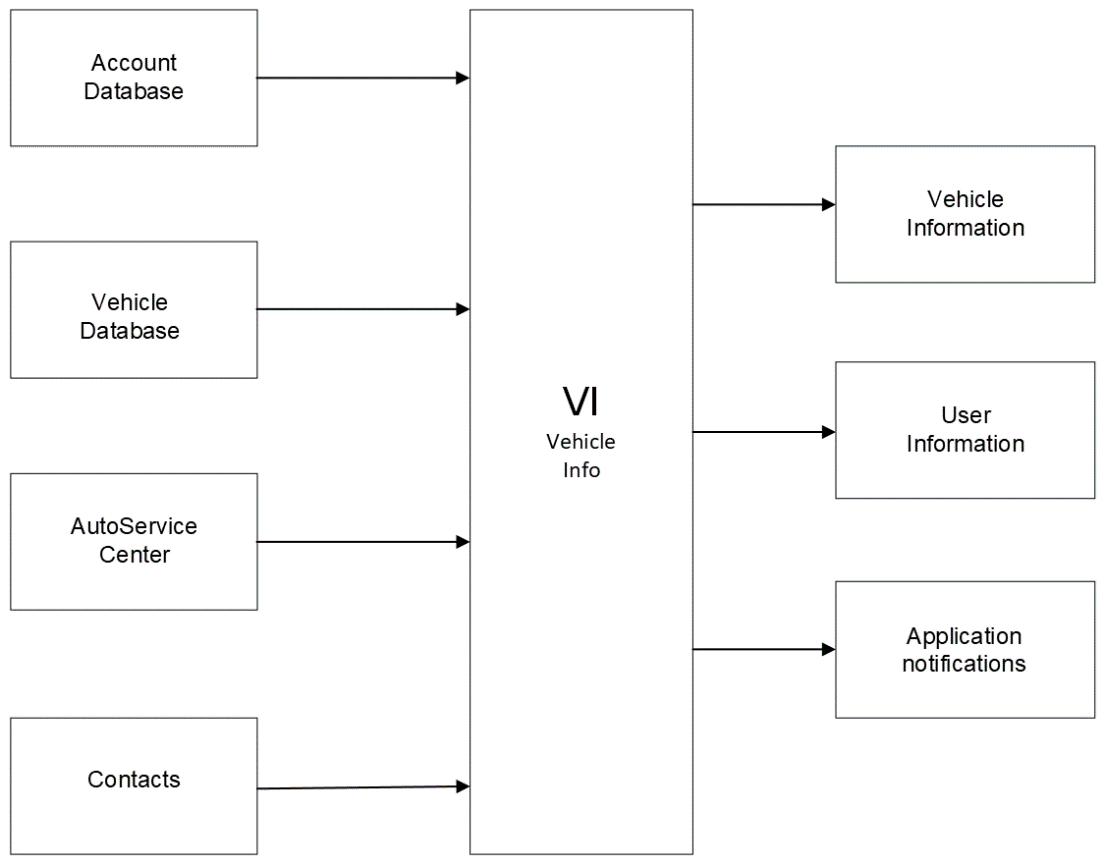
**4.1.2.1 Selection of the model**

Chosen model of IS: Conceptual Model

**4.1.2.2 Justification of the chosen model**

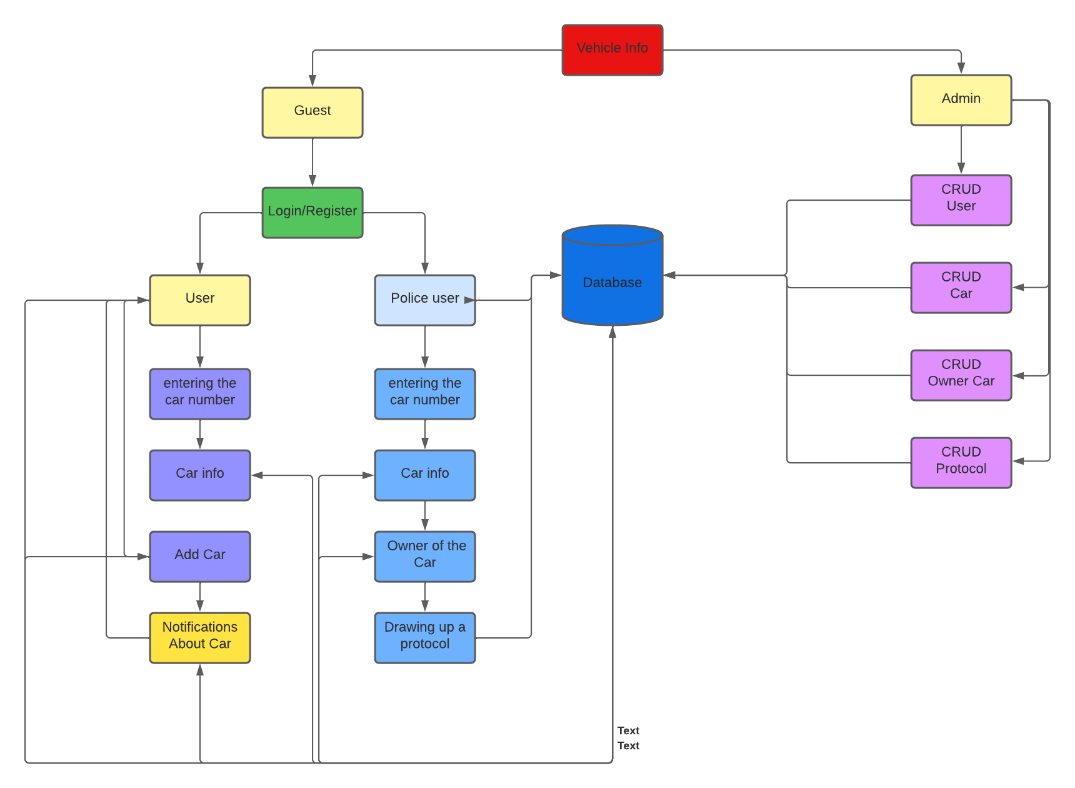
We chose conceptual model of our IS to show basic tools and functions of application.

**4.1.2.3 Construction of the general model**

****

**Figure 1.2. General model.**

**4.1.3 IS Architecture**

****

**Figure 1.3. IS Architecture.**

**4.1.4 Information support requirements**

Volume of text: less than <50, but depends on information

Graphics: Text format, 2D or photos of vehicle/citizen.

Multimedia information in MB: Depends on saved cache of information, approx. – 500 – 1500 MB.

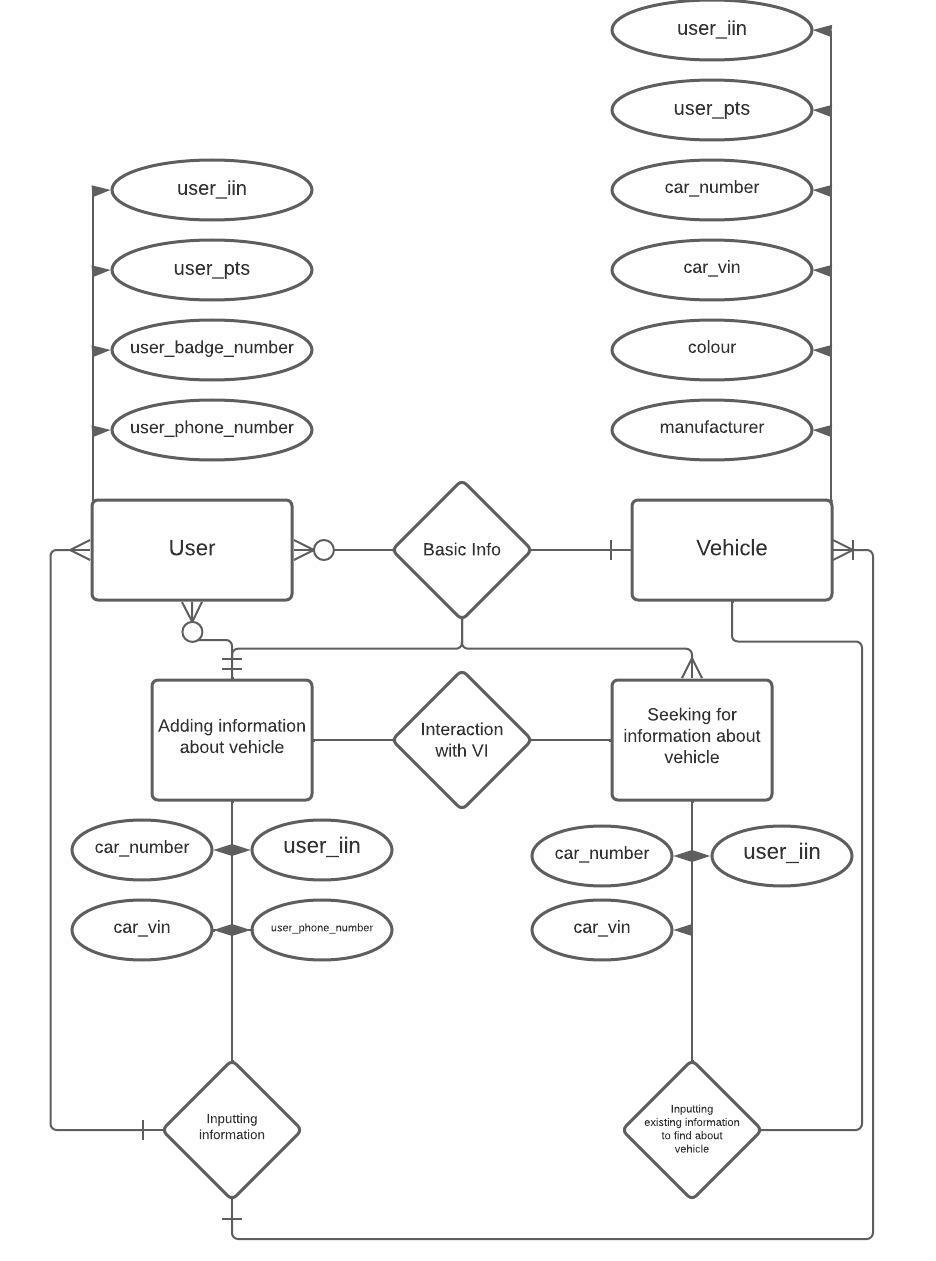
Number of files: Approximately from 5-8 files.

**4.1.5 Software requirements**

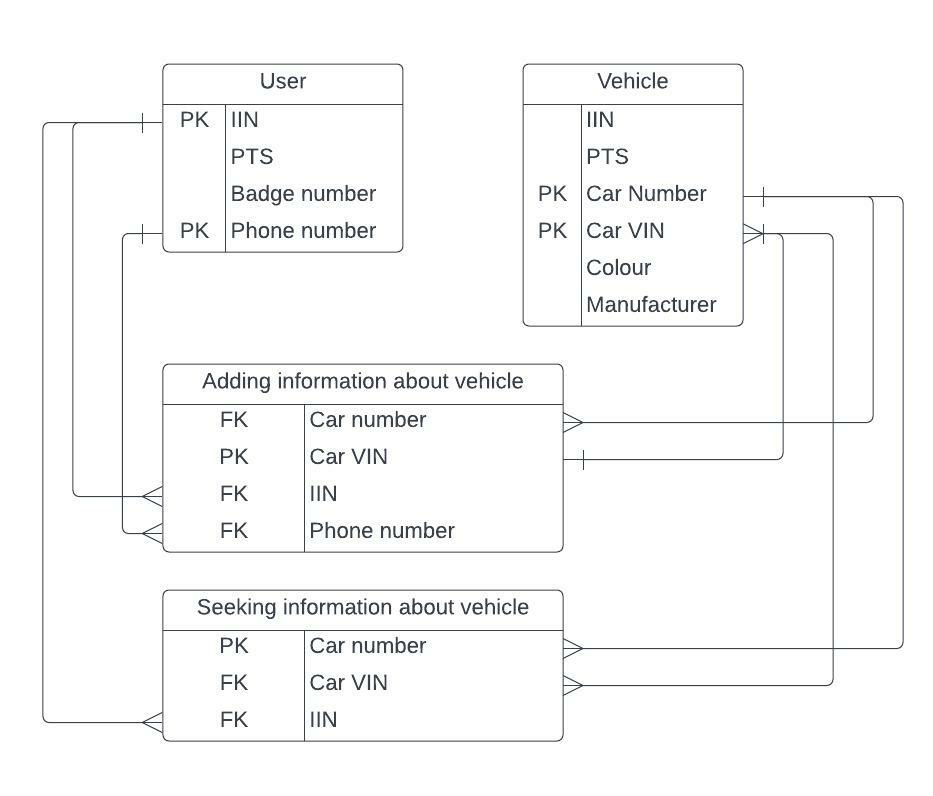
The total amount of software in MB: 667MB.

**4.1.6 Requirements to the construction of the algorithm**

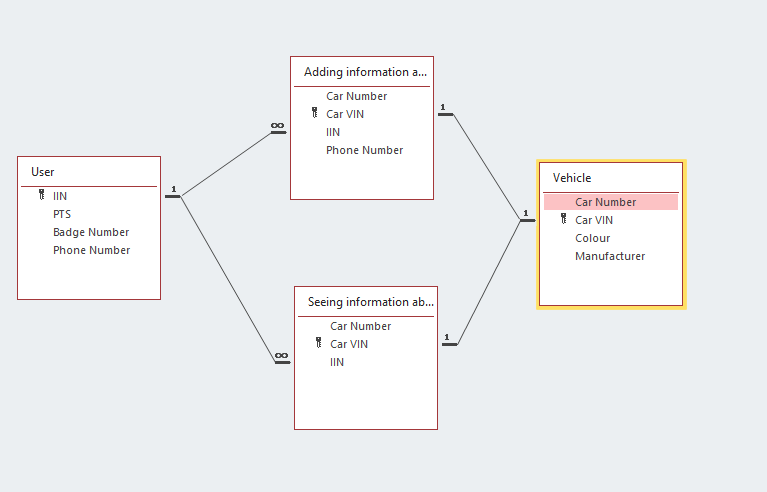
**4.1.6.1 Structure of database**



**Figure 1.4. ER model-diagram.**



**Figure 1.5. Schema plan for database.**

****

**Figure 1.6. Structure of database.**

**4.1.6.2 Data Access Technology**

ODBC- This program interfaces (API) in C applications to connect to different databases. When you connect using ODBC application becomes independent of the data source used (and used by the DBMS). Independence implemented using intermediate libraries, which include specific code for a given database, and which provide a uniform interface for ODBC-applications. These libraries are called ODBC-driver, and usually provide database developers themselves.

**4.1.6.3 Requirements to the user data queries from the database**

For users: inputting information about yourself and owned vehicle to the database, then to show information to police officer, Auto Service Center employee etc., or getting notifications from PD for technical inspection or etc.

For police: inputting ID plate of vehicle to see overall information about vehicle, owner, documents, insurance etc.

**4.1.6.4 Requirements to the source code/programming languages**

For our programming language we chose Java because our project will be multi-platform application that will run on devices that has OS such as IOS, Android or Windows Phone.

**4.1.6.5 Modern theories and methods of IS development**

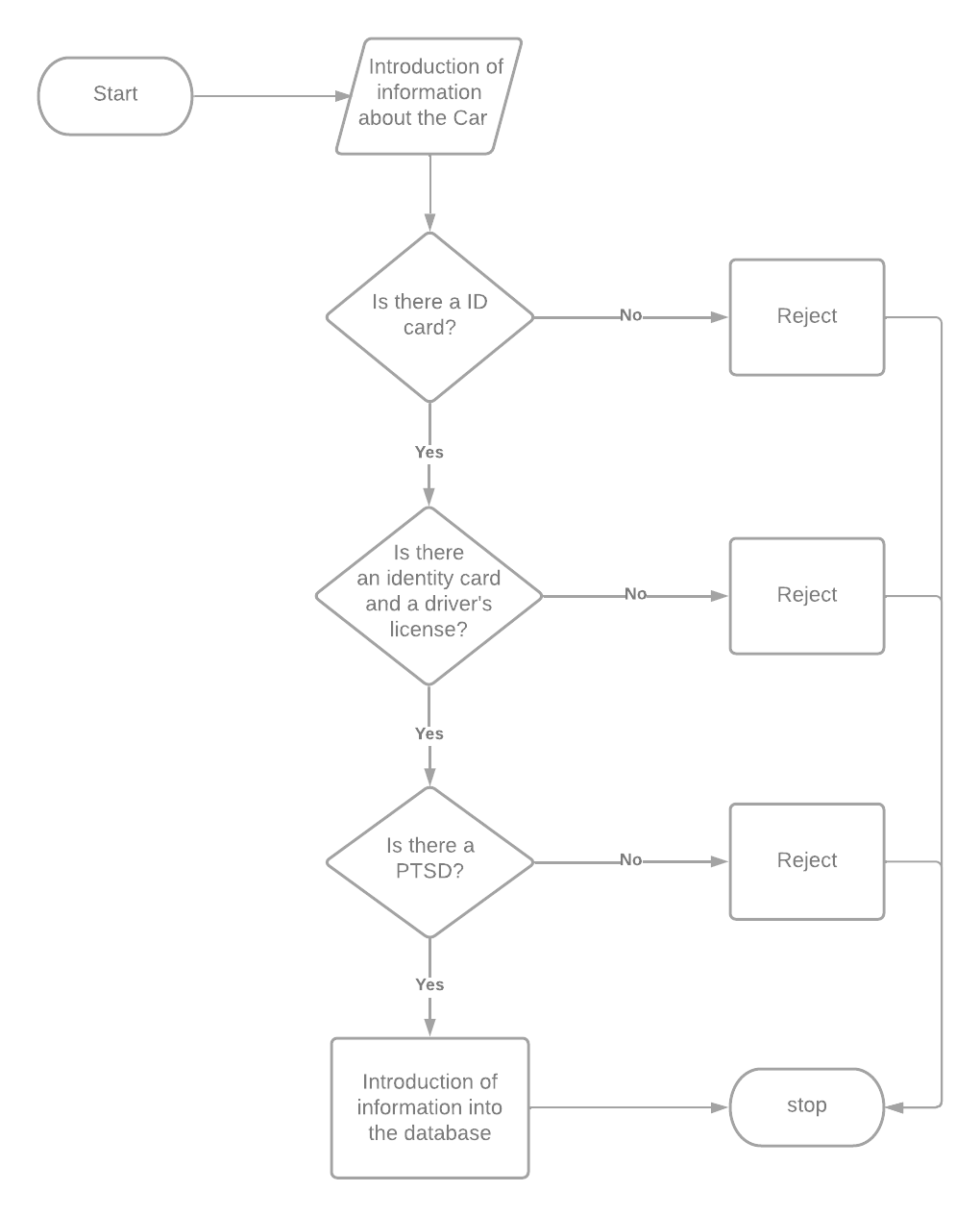
* Modeling of systems and UML
* Decision tree
* Databases and knowledge bases

**4.1.7 OS requirements**

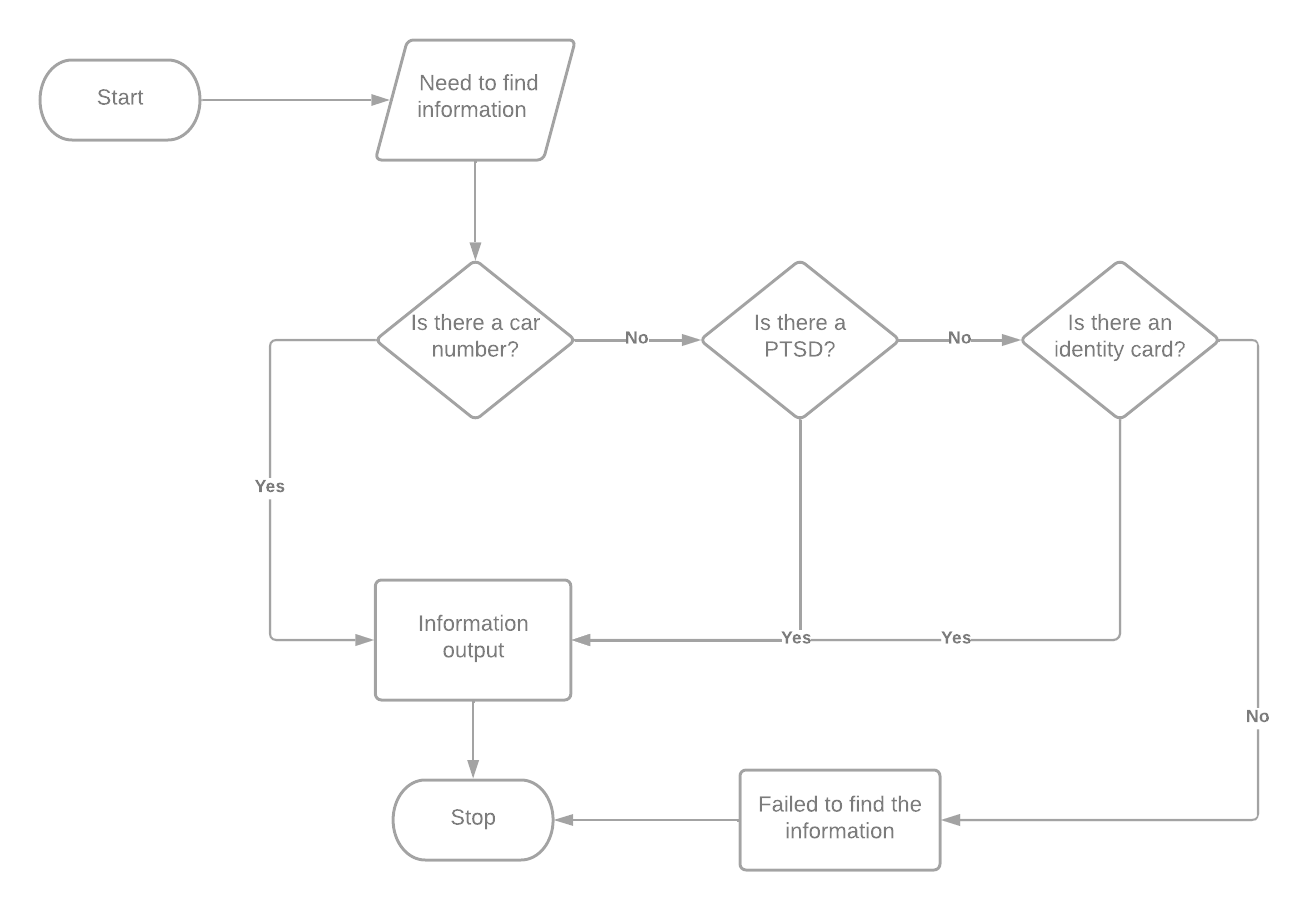
Required operating systems:

* Android 9 or greater
* IOS 13 or greater
* Windows 8.1 Phone 2 or greater

**4.1.8 Construction of the algorithm**

****

**Figure 1.7. User input algorithm**

****

**Figure 1.8. Police officer input algorithm**

**4.2 Requirements to reliability**

The higher the OS version and more RAM of device, the more stable and reliable the app is.

**4.3. IS Security**

**4.3.1. Copyright protection**

Изображение выглядит как текст

Автоматически созданное описание

**Figure 1.9. Copyright certificate**

**4.3.2. Protection of information**

* Using antivirus software
* Store passwords in hashed form
* Configure access rights for each role
* using trusted server hardware

**4.3.2.1. Methods of protection**

* Configure access rights for each role
* Prohibiting the installation of data on an external storage device
* Compiling a "whitelist" of applications
* Working with credentials

**4.3.2.2. Protection algorithm**

DES (Data Encryption Standard) is an algorithm for symmetric encryption developed by IBM and approved by the US government in 1977 as an official standard (FIPS 46-3). The block size for des in is 64 bits. The algorithm is based on a Feistel network with 16 cycles (rounds) and a key having a length of 56 bits. The algorithm uses a combination of nonlinear (S-blocks) and linear (permutations of E, IP, IP-1) transformations.

**4.3.2.3. Anti-virus Protection**

Avast

is an antivirus program designed to protect against various Internet threats. Applications for Windows OS and from Mac OS to tablet detect and block viruses, ransomware, malware, phishing and spyware in real time.

**4.3.2.4. Protection against attacks**

Protection against attacks using hash functions is often used to control the integrity of important operating system files, important programs, and important data. Monitoring can be carried out both as necessary and on a regular basis.

Using the encryption method of a single-key cryptosystem, the same key can be used for both encryption and decryption of information at the same time. This encryption method is called symmetric encryption, also called single-key encryption.

Commonly used symmetric encryption algorithms: AES, SM4, 3DES, IDEA, Blowfish, RC5, IDEA, SKIPJACK, DES, XXTEA.

**4.3.2.5. Protection against hacking**

CRC checksums should be used to protect the application. Any file, string, or block of data can be protected with a checksum, which can then be calculated and compared with a reference. This will not only help against hacking, but also protect programs from a virus or Trojan.

**4.4 Requirements for exploitation**

**4.4.1 Exploitation conditions**

**4.4.1.1 Climatic conditions of exploitation**

This system does not depend on weather conditions as the app works with GPS

**4.4.1.2 Requirements to employees qualification and number**

Front-end developer, Back-end developer, Database developer and System Administrator. Avg number of employees – 4-5.

**4.4.2 Help manual development**

“Manual.hlp” file will be included with IS source code. It will include all the technical instruction.

**4.5 Technical requirements:**

**4.5.1 The recommended monitor resolution range at which software will be viewed is**

400\*900 or more.

**4.5.2 The minimal monitor resolution range at which software will be viewed**

300\*500 or more.

**4.5.3 Recommended smartphone configuration**

For IOS – A15, For Android – Snapdragon 855 or better, 6GB RAM

**4.5.4 Minimal smartphone configuration**

For IOS – A9 or better, For Android – Snapdragon 667 or better, 2GB RAM

**4.6. Non-Technical requirements to IS:**

**4.6.1 Adaptability**

Adaptability is on the first place of our IS. We're developing replacement for old police officers’ tablets, that could take time or crash on not right moment. Thus, our IS will try find information about vehicle as fast as it could, even if it contains lack of information.

**4.6.2 Intellectual development**

Our application will give recommendations or similar requests based on done operations to user. Whether it be question to Auto Service Center or finding information of vehicle.

**4.6.3 Consistency**

Our project is an application software.

**4.6.5 Integrity**

Until the end of whether the design of the software or plug-ins required.

**4.6.6.2 Reliability**

Our project is created to work under different situations.

**4.6.6.3 Ease of application**

The design of our application will be user-friendly, and it will have tutorials and guidance for users.

**4.6.6.4 Effectivity**

Our application will work faster compared to police officer’s tablets and won’t use many phone energy.

**4.6.6.5 Maintainability**

Maintainability of our project will depend on funding from MIA of KZ and requests and feedback from users.

**4.6.6.6 Possibility to learn**

Our application will provide tutorial and provide advice for new users (only for citizens). As for the police officers, they need to contact to their supervisor that will give briefing about “How to use” application.

**4.6.6.7 Modifiability’**

Our project is not open-source project, thus expanding and modifying of it will depend on our decisions or by the request of MIA KZ.

**4.6.6.8 Mobility**

Our IS developing only for smartphones, thus it will have 100% mobility and there’s no plans to release it for PCs.

**4.6.6.9 Finiteness**

The inverse of the frequency of failures is 15-20%.

**4.6.10 Accuracy**

Our IS’s accuracy will be at 90%, depending on basic information of car.

**4.6.6.11 Autonomy**

Our application will work without any support of another software or functions of the smartphone.

**4.6.6.12 Stability**

Stability of our application is on the first place, because it will depend on work of police officers.

**4.6.6.13 Security**

Administrator with full access to program (CRUD etc.) will be in charge of security and stability of application. Also, the databases will be placed and run through Kaspersky Laboratories.

**4.6.6.14 P-documentation**

Not be included.

**4.6.6.15 Informational content**

Not be included.

**4.6.6.16 Sociability**

Our application is not a search engine like Google or Yandex. It won’t show something similar; it will show exactly the vehicle user needs to find.

**4.6.6.17 Time efficiency**

Average time to find the vehicle – 30 to 40 seconds. Results will depend on inputted data.

**4.6.6.18 The effectiveness of memory**

The more memory device will have, the more will be saved information. And, the more RAM device will have, the faster will be response of application.

**4.6.6.19 Efficiency devices**

Depends on hardware, how we wrote before, the more RAM, the faster response.

**4.6.6.20 C-documentation**

Not be included.

**4.6.6.21 Intelligibility**

Not be included.

**4.6.6.22 Structured**

Our project will be structured by division of each application area.

**4.6.6.23 Readability**

Our IS will be readable as much as it could be. It will show only REALLY needed information. There will be a convenient and readable font also.

**4.6.6.24 Extensibility**

Our application extends it size, saves the cache of found information, so it could show information about vehicle, if there’s no connection to the internet.

**4.6.6.25 Modularity**

The property that characterizes the software organization in terms of its programs from discrete components such that a change in one of them has a minimal impact on other components.

**4.6.6.26 Regardless of the device**

Our Information system is a multi-platform application which will work on most smartphones, regardless of brand or model differences.

**5. PSYCHOLOGICAL FEATURES**

**5.1.1 Aesthetic look**

The aesthetic appearance of the application is associated with police officers and their official car.

**5.1.2 Choice of style**

The style of the application was chosen so that the application was clear, simple and logical for the user.

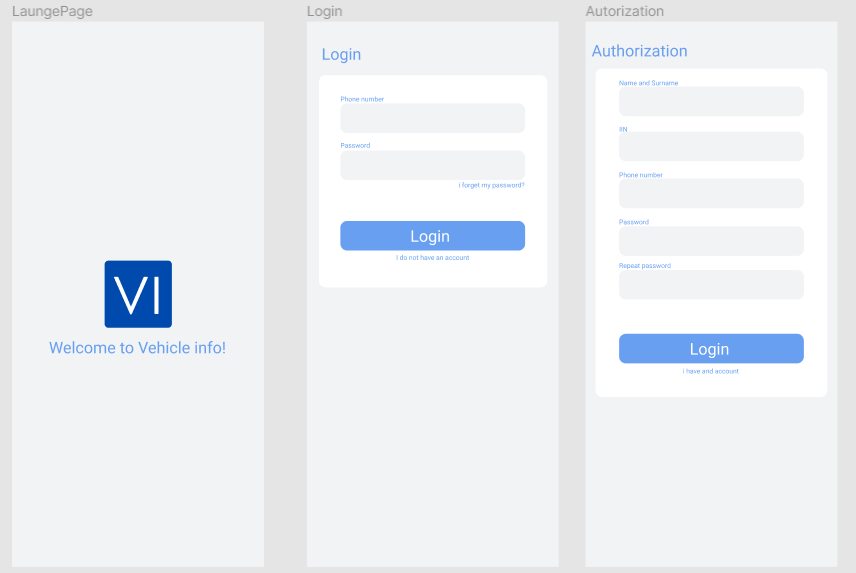
**5.1.3 Color solution**

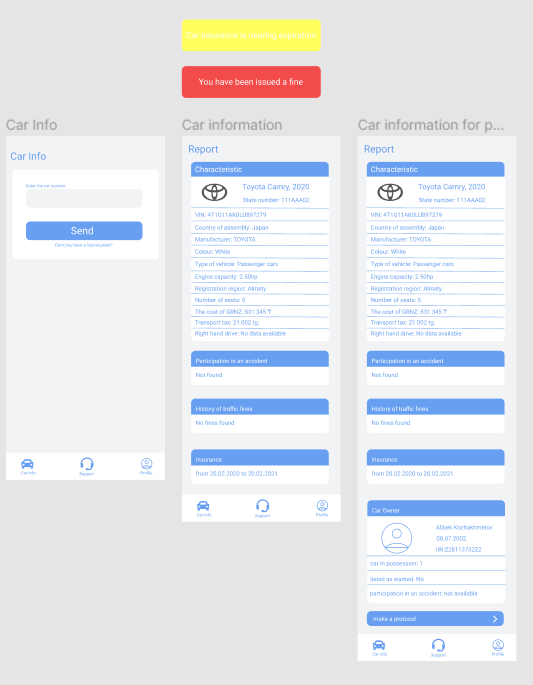
The application was made in the same color scheme as the patrol car coloring.

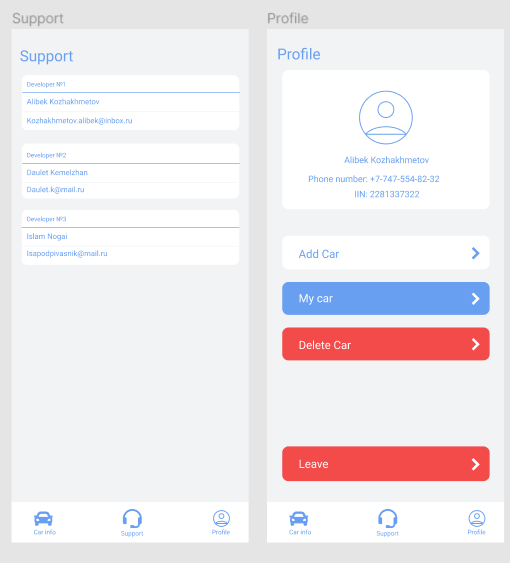
bright blue - indicates the main buttons in the application. Containers containing basic information are painted white. The background in the application was made light gray.

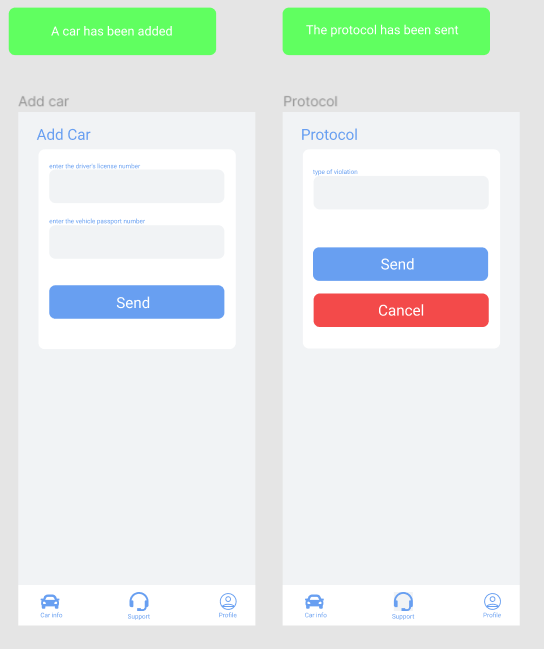
**5.2 Location of interface elements**

<https://www.figma.com/file/x4WeACcnFhBCqgG4qmduqO/Untitled?node-id=0%3A1>









**5.3 Ergonomics**

When opening the application, the boot screen will display "Welcome to Vehicle Info"5.4 Target audience

The main users of the application are police officers and ordinary citizens.

**5.4.1 Age of users**

For users 16 years and older.

**5.4.2 Their mood, temperament, etc.**

It is assumed that the user's mood will be neutral.

**6. ECONIMIC RATIONALE**

**6.1 Developing of IS business plan**

Business Compendium is a technique for calculating the economic cost of the program on IS.

Includes calculations of estimated expenses, the cost of software development, the cost of one CD software, cost-effectiveness.

**6.2 Calculation of IS cost**

**6.2.1 Calculation of cost estimates**

**Изображение выглядит как стол

Автоматически созданное описание**

**6.2.2 Calculation of IS development costs**

**Изображение выглядит как стол

Автоматически созданное описание**

**6.2.3 Calculation of one CD (license) cost**

We use a rough calculation.

To calculate the cost of one CD to analyze market PP and the number of potential consumers (P) of the software. To justify.

In the beginning it is necessary to calculate the cost (C) software. C = cost / P

Suppose potential consumers ≈ 1000 ≈ 10,000 costs

10000 + 20000 = cost of development

30% of all opportunities to recoup the costs +

Calculate the cost of software:

C = 10000/1000 = 10% - costs covered

**6.3 Calculation of economic efficiency**

Economic efficiency means in itself revenues from sales and net profit.

Net profit = revenue - taxes - expenses. Based on this formula to calculate the net income and the calculation result.

Income includes all cash received from the sale of software.

Tax (VAT) = 13% (if the income does not exceed 1 million. $)

**6.4 Building up its РR-campaign**

**6.4.1 Analysis of the market**

To analyze the market for software to determine the place of delivery of the developed program.

**6.4.2 Advertising campaign for the promotion of IS**

A description of all advertising companies (shares), which will be held on the promotion of the software. If the media will be used, indicate their names.

**7. STAGES OF DEVELOPMENT**

1) Writing technical specifications

2) Modeling

3) Analysis

4) Development

5) Testing

6) Debugging

7) Support

8) Promotion and sale

**8. IS TESTING AND DEBUGGING**

**8.1 Testing and Debugging IS**

There are many methods of debugging that can be categorized and classified based on the features of software development. The most promising of them, from the viewpoint of improving the reliability of programs are the methods of automated testing since automation can increase the productivity of the process of debugging and reduce the likelihood of introducing errors at this stage.

Testing IS - a process study (comparison) software on a data set for which the result is known in advance the use or know the rules of conduct of the programs.

IS testing - the process of assessing the quality of the software to detect it possible potential errors.

**Classification errors**

In accordance with the processing steps which occur errors are distinguished:

- Syntax errors - errors recorded by the compiler (compiler, interpreter) when the syntactic and semantic analysis in part, of the program.

- Layout errors - errors detected by the linker (linker) by combining modules of the program.

- Runtime errors - errors detected by the operating system, hardware or user during program execution.

As well as:

* Spelling
* Stylistic
* Syntax
* Punctuation
* Logic
* Functionality
* Technical
* Software
* Algorithmic
* Optimization
* Psychological
* Semantic
* Specific.

**Debugging -** is the process of locating and correcting errors found during testing software. Localization is the process of determining the operator programs the implementation of which cause a malfunction in the computational process. To correct the error, you must determine its cause, t. E define an operator or a fragment containing the error. Causes of errors can be both obvious and very deeply hidden.

**Methods of debugging software IS**

Debugging the program in any case involves thinking and logical understanding of all available information about the error. Most errors can be detected by indirect signs through a careful analysis of texts of programs and test results without more information. In this case, use a variety of methods:

- Manual testing.

- Induction.

- Deduction.

- Backtracking.

**The method of manual testing.** This - the simplest and most natural way to this group. When an error is detected, it is necessary to perform the program being tested manually using the test kit, at work with which the error was detected. The method is very effective, but not suitable for large programs, programs with complex calculations and in cases where the error due to wrong assumptions about the programmer some operations. This method is often used as part of other debugging techniques.

**Method of induction.** The method is based on a careful analysis of symptoms errors that may appear as the wrong calculation results or error message. If the computer is simply "freezes", the fragment displays error calculated from the results obtained and the last user action. Information thus received organize and scrutinize, browsing the corresponding fragment program. As a result of these actions hypothesize error, each of which is checked. If the hypothesis is correct, then the detail information about the error, or - put forward another hypothesis.

The most important stage - identifying symptoms mistakes. Organizing data error, it is advisable to write down everything that is known about its manifestations, and, fix, as the situation in which a fragment of an error is performed normally, and the situation in which the error occurs. If because of examining the data no hypothesis does not appear, you need more information about the error. Further information can be obtained, for example by performing a similar test.

In the process of trying to find evidence that all manifestations of this hypothesis explain the error, if not all, either hypothesis is not correct, or a few errors.

**The method of deduction.** By the method of deduction initially form a variety of reasons that could cause this manifestation of errors, and then analyzing the reasons, exclude those that are contrary to reports. If all excluded reasons, it is necessary to perform an additional test the fragment, otherwise, most likely hypothesis try to prove. If the hypothesis explains the features obtained error, the error is found, otherwise - check the following reason.

The method of backtracking. For small programs to effectively use the method of backtracking. Start from the point O incorrect result. To this point of conjecture about the values of the basic variables that could lead to the existing result. Further, based on this hypothesis, make assumptions about the values of the variables in the previous point. The process is continued until the find the cause.

**Methods and tools for more information**

For more information about the error, you can perform additional tests or to use special methods and tools:

- Debugging output.

- Integrated debugging tools.

- Independent debuggers.

**8.2. Testing methodology**

Specify by what methods IS will be tested, for example the trial-and-error method, alpha or beta testing, and so on.

**8.3. Testing for malicious code**

Testing for malicious code is done by using some antivirus program (Kaspersky, Dr Web, etc.). Specify this program, test and draw a conclusion.

It is required to indicate whether IS will be tested and by what methods.

**9. CONTROL AND ACCEPTANCE PROCEDURE**

**9.1 General requirements for IS acceptance**

**9.1.1 Deadlines**

05.02.2022 — 05.09.2022

**9.1.2. Conditions of IS acceptance**

Software must fully comply with given contractual technical specifications.

**9.2. Test report**

The alpha test period that was between 01.03.2022 – 01.04.2022 have shown goods result with small number of errors. Design still not implemented.

The beta test period that was between 14.04.2022 – 28.04.2022, error and bugs that were in previous build were fixed, implemented new user-friendly design. Results – stable work of software.

**9.3. Acceptance Act**

covering all types of works carried out under the contract.