`International Information Technology University JSC

Faculty of Information Technology

Department of Information Systems

\_\_\_\_\_\_\_\_\_\_\_Tolshina Victoria\_\_\_\_\_\_\_\_\_\_\_\_

**Clothing store system: smart shopping screen**

Laboratory work #1

Almaty, 2021

1 **Introduction**

In our evolving times, people want to try something new all the time. Therefore, the clothing market is constantly increasing. Accordingly, the technologies for servicing it must constantly develop too. One way to develop them is to provide smart shopping screens. These screens can carry out such functions as giving the information about products: their colors, sizes, availability in the store, allows you to pay for these goods and many other things.

**2 GENERAL INFORMATION**

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

The full name of the IS –“ **Facility**”. The word “ facility” in English means many things at the same time - construction, enterprise and ease. And the creation of smart shopping screens combines all these qualities in itself.

**2.2 Information about developers and customers**

Customer:” Inditex Kazakhstan”

Contact phone: **(727) 352-70-01**

Email: inditex@gmail.com

Address: Almaty, Rozybakiev street 247A

Developer: Tolshina Victoria Victorovna

ID card number: 27276

IIN: 020321600998

Address: Almaty, 11 microdistrict, 22

Phone number: **87074392393**

Email: sngraindrops@gmail.com

**2.3 Project timelines**

Collecting and analyzing the information - 27.01.2021 – 03. 02. 2021.

Filling in all the necessary points of the document - 03.02.2021 – 20.02.2021

Coding the appropriate application - 20.02.2021 – 13.03.2021

**2.4 Funding**

All sponsorships and allocated funds belong to the company “Inditex Kazakhstan”.

**3 PURPOSE OF CREATING IS**

**3.1 Relevance**

This system is relevant because now clothing stores are located almost everywhere and require a high level of service. Some reasons why it is necessary:

1. Helps speed up the shopping process;
2. Helps to reduce the number of employees and, accordingly, payments;
3. Will arrange the promotion of other products due to the matching products function.

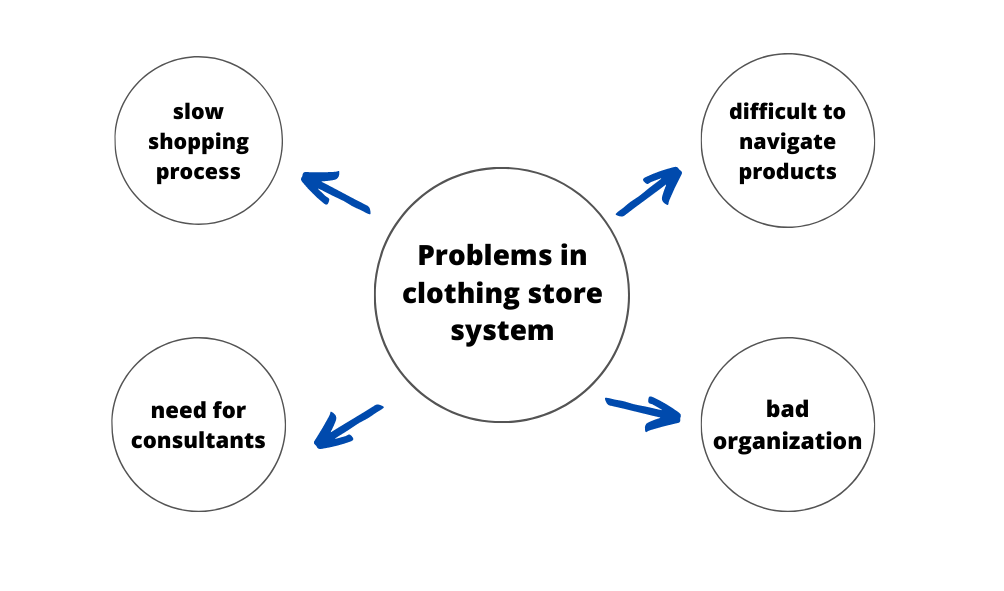
**3.2 Use**

Mostly this IS can be used in clothing stores. But it can also be adapted for other stores.

**3.3 Ideology**

Creation of technology that will help facilitate and automate the shopping process due to the Internet and applications using databases.

**3.4 Formulation of the problem**



**3.5 Formalization of the problem**

If we want to solve the problems mentioned above, we need to

speed up the buying process by requesting from buyers all the data about the thing they need. Also, to solve these problems it is required to provide them with a clear interface with a smart search engine and possible purchase recommendations.

**3.6 The goal**

The main goal is to bring sales to a new level, while using a convenient and understandable service for customers. If earlier such a system was used only by managers and other workers, now the aim is to make this technology available for buyers themselves.

**3.7 Objectives**

* Analysis of weaknesses in this area;
* Developing of algorithm of fast and comfortable shopping;
* Development of the application;
* Testing the application;
* Introduction of the application to everyday usage.

**3.8 Advantages**

- The user can look for any product he wants without the outside help;

- The user can observe all the clothes in just a few clicks;

- Maintenance personnel costs can be reduced;

- The enterprise will seem more comfortable to the buyer

**3.9 Disadvantages**

-Errors may occur in working process;

-May be difficult for people with bad eyesight.

**4 SOFTWARE REQUIREMENTS**

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

**4.1. 1 Software technology used**

Software should be developed by using C# or Python**.**

**4.1.2 IS model**

Model - a simplified representation of reality. Model - a drawing of the system: it may be either a detailed plan and a more abstract representation of the system "bird's-eye". A good model always includes elements that significantly affect the result, and does not include those that are insignificant at this level of abstraction. Each system can be described from different perspectives, which are used for different models, each of which is therefore closed semantic abstraction system. The model can be structural, emphasizing the organization of the system, or behavioral, that is, reflecting its dynamics.

Models allow us to demonstrate the desired structure and behavior of the system. They are also necessary for the visualization and management of its architecture. Models help to achieve a better understanding of the system created by us, which often leads to its simplification and reuse. Finally, the models need to minimize risk.

**4.1.2.1 Selection of the model**

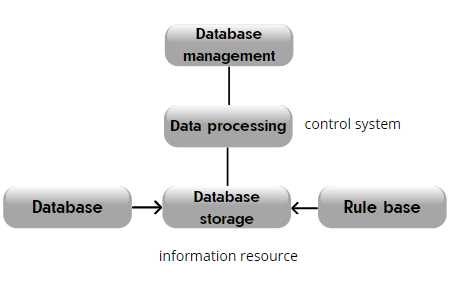
Of these types of models to determine the type model suitable for your PP:

**Conceptual model**.

**4.1.2.2 Justification of the model chosen**

A conceptual model was selected to have an ability to visually represent all the connections between modules, databases and separate blocks required for data interchange within the software.

**4.1.2.3 Construction of the general model**

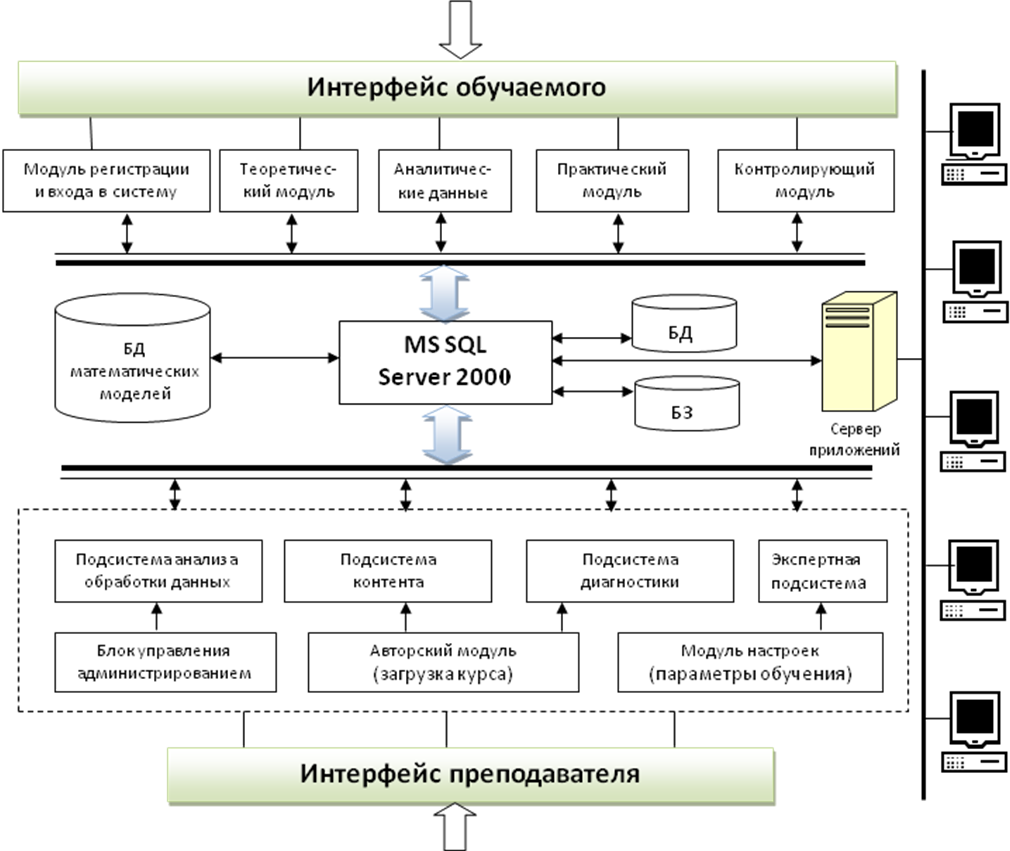


**4.1.3 IS Architecture**

IS architecture – is a base organization of system, embodied in its components (significant solutions, structural elements and its interfaces set), their relationship to each other, attitude and to environment, as well as principles, defining IS design and development.

Architecture - a set of significant decisions about the organization of the system, a set of structural elements and their interfaces by which the assembled software, together with their behavior defined in the interaction between these elements, the arrangement of elements in gradually become larger subsystems, as well as the style of architecture that directs this organization elements and their interfaces, interaction and layout.

In this section, you must draw the architecture of the software.



**4.1.4 information support requirements**

The volume of text – Medium.

Graphics – pictures taken by shop.

Multimedia information in MB – depends on the amount of items, nearly 5GB.

**4.1.5 Software requirements**

Total number of GB – maximum10GB.

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

**4.1.6.1 Structure of databases**

It is necessary to represent the structure of a database as a data logic model. Data logic model (physical model) - a model consisting of tables and relationships between us.

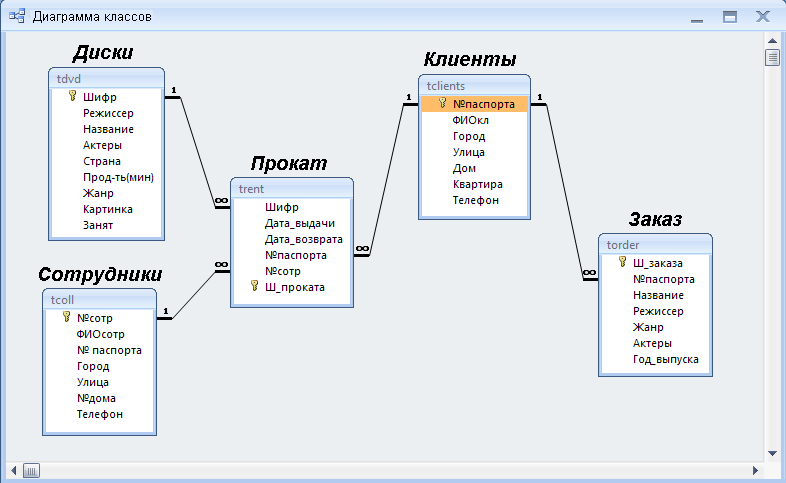


Figure 1.6 - Example of the structure of the database software owner's manual video library (developed in Access)

**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.

OLE DB - hybrid ODBC, COM, that is, to access the data it uses no API in the language C, while COM-interfaces. That is, this technology provides an object-oriented interface for any programming language compatible with COM, and not only for the Visual C ++.

RDO- (Remote Data Objects - remote data objects). DAO - this Data Access Objects (Data Access Objects). Initially, this technology was created as a COM-interface to the database Jet, which lets you work with Microsoft Access databases, and any for which there were drivers ODBC. ADO- ActiveX Data Object (ActiveX-objects for data access). It should be noted that this is one of the most popular (after ODBC) technology developed in this field by Microsoft.

ADO- ActiveX Data Object (ActiveX-objects for data access). It should be noted that this is one of the most popular (after ODBC) technology developed in this field by Microsoft. In fact, ADO - it's just another add-on already existing technologies all the same corporation. It uses ActiveX-component is an add-on API OLE DB, which in itself, in general, not so convenient to use. ADO, of course, introduces an additional level that affects the application performance, but it is so well reflected in the design-time, the technology is much more popular than, in fact, OLE DB. Another plus it - the ability to use objects to access data from scripting languages such as VBScript or Jscript. Here an important role is played by the possibility of its use for programming in ASP to develop web-based applications.

ADO.NET - A new generation of facilities for working with data, where instead of ActiveX-components are used components of .NET. MDAC- Microsoft Data Access Components (Data Access Components Corporation Microsoft) - is the common name ODBC, OLE DB and ADO. Or, to be precise, it is a set of libraries that provide job listings technologies.

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

Is a list of database queries

**Request -** output data, structured in a certain way.

**Example of university DB**:

Show list of students of IT faculty, whose family name begins from letter A, and who have at least one 2nd term exam fail.

Show list of teacher, who profess subjects on specialty of 1 course IS, on which students got the least number of points, but who in questionnaire “Teacher with eyes of students” got the biggest quantity of points.

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

The programming language. Specify what programming environment program will be implemented.

The programming language. Specify what programming environment will be implemented IS.

**4.1.6. 5 Modern theories and methods of IS development**

Select and justify the method of software development. The most important methods of software development - is the PLO and algorithmic (traditional) methods. In the first case, as the basic building block appears or object class. In the most general sense of the object - an entity that is usually extracted from the vocabulary domain or solutions, and the class is a description of the set of similar objects. Each object has its identity (which can be specified or somehow different to distinguish from other objects), state (usually associated with the object are some data) and behavior (with it you can do something or he might have something to do with other objects). In the second case, the basic building block is a procedure or function, and attention is given primarily on the transfer of control and decomposition algorithms larger to smaller. Nothing wrong with that, except for the fact that the system is not too easy to adapt. When requirements change or increase the size of the application (which happens often) to accompany them becomes more difficult

**The most modern theories and methods of IS development:**

* Decision making theory
* Expert systems
* Methods of algorithms and automatons
* Modeling of systems and UML
* Decision tree
* Fuzzy sets and logics
* Numerical methods
* Cluster analysis
* Databases and knowledge bases
* Neural networks and neuro informatics
* Genetic algorithms and natural selection
* Sparse hypercube
* Game theory

**4.1.7 OS requirements**

Operating systems that will operate the software.

**4.1.8. Construction of the algorithm**

«**Algorithm** — is a finite set of rules, which defines sequence of operations for solution of concrete set of problems and which has 5 important characteristics: finiteness, definiteness, input, output, effectivity». (D. E. Knuth)

Must submit a block diagram of the program.



Figure 1.7

**4.2. Reliability requirements:**

Describe the features and characteristics of the software.

Below is an example for online stores:

**4.2 Requirements to reliability:**

Reliability – is a probability of failure free time of IS operating.

Define the time of program recovery, in case of failure of IS.

Presence of failures in system because of incorrect actions of user. If there is, them indicate which failures will be and which actions are needed to be taken for their elimination.

**4.3. IS Security**

**4.3.1. Copyright protection**

Copyright © Facility, 2021.

**4.3.2. Protection of information**

**4.3.2.1. Methods of protection**

The use of antiviruses, authentication systems, programs regulating access to objects.

**4.3.2.2. Protection algorithm**

Creation of a list of employees with administrative rights and empowering them to control the system.

**4.3.2.3. Anti-virus Protection**

Using Kaspersky Antivirus system.

**4.3.2.4. Protection against attacks**

Wi-fi security, installing a Firewall and back-up data.

**4.3.2.5. Protection against hacking**

Using antivirus and staff trainings.

**4.4 Requirements for exploitation**

**4.4.1 Exploitation conditions**

**4.4.1.1 Climatic conditions of exploitation**

Dependence of the program work on climatic conditions

**4.4.1.2 Requirements to employees qualification and number**

List of minimum requirements to the qualification and number of employees.

**4.4.2 Help manual development**

Help manual is developed in \*.hlp format; it contains subject index, search by keyword, indication of the author.

**4.5 Technical requirements:**

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

1024\*768, 640\*480, 600\*800 and etc.

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

640\*480, 600\*800 and less.

**4.5.3 Recommended PC configuration**

2 GHz Processor, 256MB RAM, 64MB Video Card, 120 GB HDD, a CD-ROM or USB-port

**4.5.4 Minimal PC configuration**

1 GHz Processor, 128MB RAM, 100MB available HDD, 16 MB Video Card, and a CD-ROM or USB-port

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

**4.6.1 Adaptability**

Adaptively model is achieved if all component models matched to each other, i.e. developed and strictly enforced rules of engagement component. In addition there should be flexibility in (effective, fast, continuous) reorganization processes (functions, resources) that implement the internal structure of each component subject to the established rules of interaction with other components.

Adaptation of software - a complex of works on changing the standard basic versions of software that implements the requirements of the customer according to the statement. In accordance with the terms of reference, you configure the software to the specifics of accounting in the enterprise. Generated data entry forms and information.

The software product must adapt to the computer and the user.

The possibility of creating a personal agent.

Ability to adapt to different environments without holding for this action, in addition to the pre-specified.

**4.6.2 Intellectual development**

In operation, the IS it can replenish its base of knowledge, as well as to form an opinion about the user. With this in mind, take this or that decision.

**4.6.3 Consistency**

Does the software at the system level? Is the software system?

**4.6.4 Full functionality**

Full-featured - is the presence of software on the same medium.

Discover and prove a fully functional work of program.

**4.6.5 Integrity**

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

**4.6.6 Quality**

Software quality - the ability of software to confirm its specification, provided that the specification is focused on characteristics that the user wants to receive.

Factor of quality assurance - a non-functional requirements of the program, which is usually not described in the contract with the customer, but, nevertheless, it is desirable requirement, which increases the quality of the program.

**4.6.6.1 Functionality**

The ability of the software under certain conditions to solve the problem, users need. Determines what is done for what problems it solves.

Software functionality - the ability to perform a set of software functions:

- Defined in its external description;

- Satisfy given or implied user needs.

**4.6.6.2 Reliability**

Ability to maintain a certain performance under specified conditions.

Software Reliability - the ability of the software to perform certain functions smoothly under predetermined conditions for a predetermined period of time with sufficiently high probability.

Reliability of operation is characterized by a probability of the software product without failure for a specified period of time.

**4.6.6.3 Ease of application**

The ability of the software to be easy to learn and use, as well as attractive to users.

**4.6.6.4 Effectivity**

The ability of the software under the given conditions to provide the necessary performance in relation to the allocated resources to do so. You can define it, and as the ratio obtained by means of the software results to spent on a resource.

The effectiveness of the software - the ratio of the level of services provided by the user software product under specified conditions, to the volume of resources used.

**4.6.6.5 Maintainability**

Ease of all activities related to the support of the program.

Maintainability of software - software product characteristics that minimize efforts to amend it:

- To eliminate errors;

- For modification in accordance with the changing needs of users.

**4.6.6.6 Possibility to learn**

Indicator, reverse the efforts spent by the user to

learn how to work with the software.

**4.6.6.7 Modifiability**

Extensibility, structuring, modularity.

**4.6.6.8 Mobility**

Ability to continue to operate when moving from one environment to another, including organizational, hardware and software aspects of the environment, the ability of software and computer system as a whole continue to operate under its physical movement in space.

Mobility software - the software's ability to work on different hardware platforms or running different operating systems.

**4.6.6.9 Finiteness**

The inverse of the frequency of failures.

**4.6.10 Accuracy**

The ability to give the desired results (e.g., with a certain confidence interval)

**4.6.6.11 Autonomy**

Property that characterizes the ability of the PC to perform intended functions without help or support other software component

software.

**4.6.6.12 Stability**

The ability to maintain a given level of performance in case of failures and irregularities interaction with the environment.

**4.6.6.13 Security**

Ability to prevent unauthorized, ie without the person who is trying to carry it out, and do not allow access to data and applications.

**4.6.6.14 P-documentation**

The property is characterized by the presence, completeness, clarity, accessibility and visibility of the training, guidance and reference documents necessary for the application of the PC

**4.6.6.15 Informational content**

The property is characterized by the presence in the composition according to the information necessary and sufficient for understanding the purpose of the SS, the assumptions made, the existing limitations, the input data and the results of the individual components, as well as the current state of the program in the course of their operation.

**4.6.6.16 Sociability**

Property characterizing the degree to which the SS

facilitates the task description or the input data, and provides useful information issuing in the form and content that are easy to understand.

**4.6.6.17 Time efficiency**

The ability of the software to give the expected results, as well as provide for the transfer of necessary volume of data in the allotted time.

**4.6.6.18 The effectiveness of memory**

Measure characterizing the ability of the PC to carry out its functions under certain restrictions on memory usage.

**4.6.6.19 Efficiency devices**

The measure is cost-effective use of the devices of the machine for the task.

**4.6.6.20 C-documentation**

Properties that characterize in terms of availability documentation of the software requirements and the results of the different stages of development of the 10 software, including capabilities, limitations, and other features of the software, as well as their justification.

**4.6.6.21 Intelligibility**

Indicator, reverse the efforts spent by the user to accept a set of concepts, on which the software and their applicability to solve their problems.

**4.6.6.22 Structured**

The property that characterizes the program PS in terms of organization of related parts into a single unit in a certain way (e.g., according to the principles of structured programming).

**4.6.6.23 Readability**

Readability software - software product characteristics that:

- Minimize user effort in preparing the initial data, application software, and evaluation of the results

- Can cause positive emotions a specific user or implied.

Property characterizing the ease of perception of the text software programs (indentation, fragmentation, formative).

**4.6.6.24 Extensibility**

Property that characterizes the ability of the software to use more memory for data storage or expanding the functionality of the individual components.

**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**

Property characterizing the ability to work on a variety of hardware (different types, brands and models of computers).

**5 PSYCHOLOGICAL FEATURES**

**5.1.1 Aesthetic look**

The interface is minimalistic.

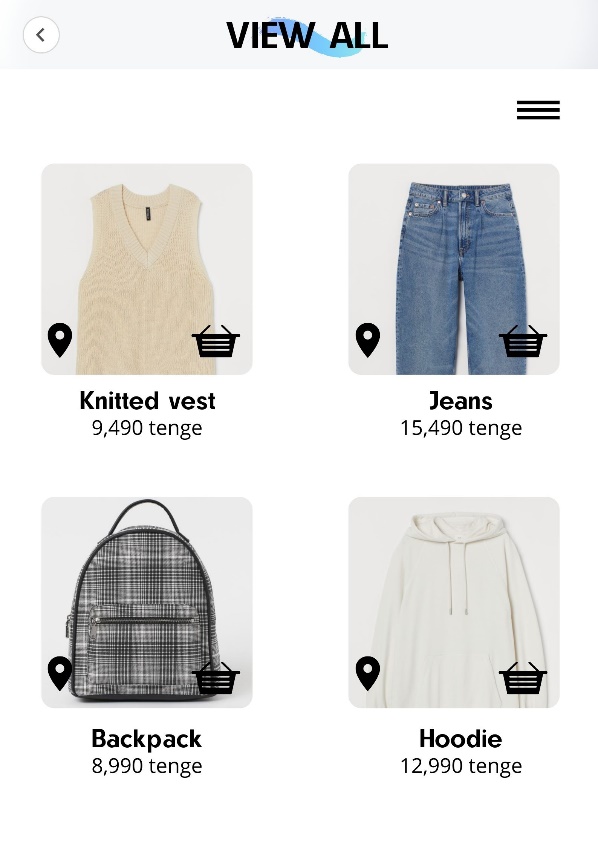
**5.1.3 Color solution**

The predominant colors are black and white, in some places there are gradient elements.

**5.2 Location of interface elements**

“Main menu” and “View all” page look like this:

**Изображение выглядит как текст

Автоматически созданное описание** 

**5.3 Ergonomics**

Friendliness, nice interface, including some instructions for use, FAQ and display settings, where users can change brightness.

**5.4 Target audience**

No special target audience, this technology can be used by people of all ages.

**6 ECONOMIC 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**

In calculating the cost estimates need to calculate all the costs incurred for the implementation of the software, such as the cost of equipment (PC, office equipment), wages, services (utilities, information translated into another language, the Internet, etc.), materials (kants.tovary, CD, table, chair, etc.) , software (Windows, antivirus system, programming languages), and others.

**6.2.2 Calculation of IS development costs**

More in 4 laboratory work

**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.

1. **STAGES OF SOFTWARE DEVELOPMENT**
2. **Writing technical specifications** - filling out the plan for the upcoming work, a rough idea of ​​the future project.
3. **Modeling** - coming up with the design and appearance of the program.
4. **Analysis** - study of store products, their classification and availability.
5. **Development** - writing an application in Python, linking to the store database.
6. **Testing** – trying to check if the system works by viewing the goods and paying.
7. **Debugging** – if something isn`t clear and requires for modernization or just doesn`t work.
8. **Support** - introduction of a system of assistance and training of employees to work with the system.
9. **Promotion and sale** - sale and delivery of the product to all stores of the chain.

**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 et al.

**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 as a result 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 explains 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, on the basis of 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**

Collecting and analyzing the information - 27.01.2021 – 03. 02. 2021.

Filling in all the necessary points of the document - 03.02.2021 – 20.02.2021

Coding the appropriate application - 20.02.2021 – 13.03.2021

Acceptance deadline – 14.03.2021

**9.1.2. Conditions of IS acceptance**

Equipment can only be accepted if at least one purchase is successful, the design and quality has already been shown to customer and eliminated all the shortcomings.

**9.2. Test report**

The test report shall include information about all the tests carried out on this software and their results.

**9.3. Acceptance Act**

Covering all types of works carried out under the contract.