МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кафедра комп’ютерних систем та мереж

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
__________________Жуков І.А.
« ____ » ____________ 2020 р.

ДИПЛОМНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ “МАГІСТР ”

Тема: Інформаційна система ремонтної агенції
з використанням Spring Boot

Виконавець:____________________________________ Мамчур В.В.

Керівник:______________________________________ Телешко І.В.

Нормоконтролер з ЄСКД::________________________ Надточий В.І.

Київ 2020

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
NATIONAL AVIATION UNIVERSITY

Computer Systems and Networks Department

PERMISSION TO DEFEND GRANTED

The Head of the Department

_____________________ Zhukov I.A.

“_____”______________________2020

MASTER’S DEGREE THESIS

(EXPLANATORY NOTE)

Specialty: 123 Computer Engineering

Educational-Professional Program: Computer Systems and Networks

Topic: _________*Information system of repair agency*_______________________
____________________*with using of Spring Boot*__________________________

Completed by: ______________________________________Mamchur V.V.

Supervisor: ____Teleshko I.V.

Standards Inspector__________________________________Nadtochiy V.I.

Kyiv 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп’ютерної та програмної інженерії

Кафедра комп’ютерних систем та мереж

Освітній ступінь магістр

Спеціальність 6.050102 “Комп’ютерна інженерія”

Освітньо-професійна програма: «Комп’ютерні системи та мережі»

ЗАТВЕРДЖУЮ
Завідувач кафедри

____________Жуков І.А.

« ____ » ___________ 2020 р.

ЗАВДАННЯ
на виконання дипломної роботи

Мамчур Віталій Володимирович

(прізвище, ім'я, по батькові)

1.Тема проекту (роботи): “ Інформаційна система ремонтної агенції з використанням Sping Boot”

затверджена наказом ректора від " 25 " вересня 2020 року № 1793/ст.

2.Термін виконання проекту (роботи): з 01.*10*.2020 до 25.12.2020

3. Вхідні дані до роботи (проекту*): вимоги до системи*

4. Зміст пояснювальної записки:

Аналіз роботи ремонтного агенства

Аналіз вимог

Структура системи

Робочий прототип системи ремонтної агенції

5. Перелік обов'язкового графічного (ілюстративного) матеріалу:

ілюстративний матеріал в Power Point.

6. Календарний план-графік

№ п/п	Етапи виконання дипломного проекту	Термін виконання етапів	Примітка
1	Узгодження технічного завдання з керівником проекту	25.10.20	
2	Підбір та вивчення науково-технічної літератури за темою дипломного проекту	26.10.20 – 31.10.20	
3	Опрацювання матеріалу щодо створення систем продаж	01.11.20 – 09.11.20	
4	Аналіз впроваджених технологій	10.11.20 – 12.11.20	
5	Розробка практичної частини	13.11.20 – 01.12.20	
6	Оформлення пояснювальної записки	02.11.20 – 07.12.20	
7	Оформлення графічних матеріалів проекту та представлення роботи на кафедру	08.12.20 – 10.12.20	

7. Дата видачі завдання «25» вересня 2020 р.

Керівник дипломного проекту____________________ Телешко І.В.__________
(підпис)

Завдання прийняв до виконання____________________ Мамчур В.В.__________
(підпис студента)

NATIONAL AVIATION UNIVERSITY

Faculty of cyber security, computer and soft engineering

Department: *Computer Systems and Networks*

Educational and Qualifications level: *Master Degree*

Specialty: *123 "Computer Engineering"*

Educational-Professional Program: "Computer Systems and Networks"

APPROVED BY

The Head of the Department

____________________ Zhukov I.A.

"_____"____________________2020

Graduate Student's Degree Thesis Assignment

Mamchur Vitalii Volodymyrovych

1. The Project topic: *" Information system of repair agency with using of Spring Boot"*

Approved by the Rector's order of 25.09.2020 № 1793/st.

2. The Thesis to be completed between 01.10.2020 and 25.12.2020

3. Initial data for the project (thesis): *requriment to the system*

4. The content of the explanatory note (the list of problems to be considered):

Analysis of the repair agency work
Analysis of requirments
System structure
Working prototype of repair agency system

5. The list of mandatory graphic materials:

illustrative material in Power Point.

6. Timetable

#	**Completion Stages of Degree Project**	**Stage Completion Dates**	**Remarks**
1	Technical task coordination with the supervisor	25.10.20	
2	Selection and study scientific literature on the topic of master degree thesis	26.10.20 – 31.10.20	
3	Work with theoretical materials	01.11.20 – 09.11.20	
4	Analysis of technology	10.11.20 – 12.11.20	
5	Work with practical part	13.11.20 – 01.12.20	
6	Diploma note preparing and forming	02.11.20 – 07.12.20	
7	Graphical materials preparation	08.12.20 – 10.12.20	

7. Assignment issue date: 25.09.2020

Diploma Thesis Supervisor _____________________________ Teleshko I.V.
(Signature)

Assignment accepted for completion _____________________Mamchur V.V.
(Signature)

ABSTRACT

The Explanatory Note to the Master's Degree Thesis "Information system of repair agency with using of Spring Boot": 73 pages, 38 figures, 15 references.

The Goal of the Master's Degree Thesis – create the information system of repair agency with using of Spring Boot.

Main Tasks – automation of the process of creating of request for repair of any staff, its confirmation, processing and closing.

The method of development - an object-oriented approach.

Technical and software – Operational system – Linux Mint 19.4, Development environment - Intelij Idea Ultimate, JDK – 1.8.0.

Practical Usage. The result is working web-site which can be used by any repair agency to improve their work.

CONTENT

LIST OF SYMBOLS, ABBREVEATIONS, TERMS 7

INTRODUCTION 8

PART 1 ANALYSIS OF REPAIR AGENCY WORK 9

- 1.1. Maintenance of tools 9
- 1.2. Causes of automation software appearance 10
- 1.3. Advantages of automation system implementation 10
- Conclusion on the first part 18

PART 2 REQUIREMENTS FOR THE REPAIR AGENCY SYSTEM 19

- 2.1. Analysis of software requirements 19
- 2.2. Functional requirements for the project 20
- 2.3. Non-functional requirements for the project 22
- Conclusion on the second part 26

PART 3 SYSTEM STRUCTURE 27

- 3.1. Project architecture 27
- 3.2. Spring Boot 27
- 3.3. MVC pattern 30
- 3.4. ThymeLeaf 32
- 3.5. MySQL 33
- 3.6. Maven 36
- 3.7. Hibernate 40

CSN department				NAU 20 72 67 – 000 EN				
Done By	*Mamchur V.V.*			Information system of repair agency with using of Spring Boot	***Letter***		***Page***	***Pages***
Supervisor	*Teleshko I.V.*						*5*	*65*
Consultant					123 CE 222a			
S-inspector	*Nadtochiy V.I.*							
Head of dep.	*Zhukov I.A.*							

3.8. Bootstrap 43

3.9. Spring Security...45

3.10. Classes Diagram..47

Conclusion on the third part 49

PART 4 SYSTEM PROTOTYPE 50

4.1. System prototype in general 50

4.2. Implementation of a system for a repair agency...............................51

4.3. Repair agency apperarance 60

CONCLUSIONS..67

REFERENCES 72

LIST OF SYMBOLS, ABBREVEATIONS, TERMS

LAN	Local Area Network
WAN	Wide Area Network
ISO	International Standardization Organization
OSI	Open System Interconnection
TCP	Transmission Control Protocol
IP	Internet Protocol
OSPF	Open Shortest Path First
UDP	User Datagram Protocol
FTP	File Transfer Protocol
NIC	Network Interface Card
IEEE	Institute of Electrical and Electronics Engineers
LLC	Logical Link Control
DHCP	Dynamic Host Configuration Protocol
NAT	Network Address Translation
VLAN	Virtual Local Area Network
MAC	Media Access Control

INTRODUCTION

Due to the steady growth of consumer needs, companies are forced to increase their productivity. At the same time, they face a number of problems, among which are the deterioration of quality and optimization costs, as well as the growing consumption of human resources. Production automation systems can solve the above difficulties, as well as open new frontiers of productivity.

Production automation is a process in which previously performed human control and management functions are fully transmitted to automatic devices. very often automation is confused with mechanization. To avoid confusion, it should be remembered that mechanization is the simplification of human work through the use of automated equipment, and production automation is the use of an automatic system to control and manage the entire process. in other words, automatic control systems (ACS) exclude the presence of a person during operations and control the stages of execution.

PART 1
ANALYSIS OF REPAIR AGENCY WORK

1.1. Maintenance of tools

Issues of maintenance and repair of equipment are always relevant for any manufacturing enterprise. At some companies, the number of tools waiting to be repaired can reach up to 50%. This is not only unacceptable from the point of view of lean production, but also lead to significant use of own products, as a result of maintenance of equipment adds to its value from 10% to 40%. The issues of repairs are especially relevant for enterprises in the CIS countries, as many of them have a technical park since Soviet times. Obsolete physical and moral equipment, which has long since exhausted its resource, is constantly in need of repair and simple.

One of the options for solving the problems of proper organization of maintenance and repair of equipment is the introduction of the software package "Management of repairs and maintenance". The introduction of modern technologies for resource management of enterprises, in addition to them, the maintenance of specialized software, helps to increase the efficiency of business management, reduce investment and solve a number of major production problems, reducing this solution. There is always a huge gap between the finance department and the repair department number of documents.

1.2. Causes of automation software appearance

There is always a huge amount of documentation between the finance department and the repair work. The best these services can say about such different "languages" that often understands one of them that it is very important. Finances can't figure out yesterday, telling mechanics how to educate and analyze their work, what and why they need money. Technicians, in turn, do not want to spend time to get acquainted with the peculiarities of the work of accountants and the intricacies of economic accounting. As a result, their cooperation becomes inefficient, time consuming and provides many unclear situations.

With the use of this system, the company gets the opportunity to optimize costs, try an internal document, improve the registration and analysis of the needs of various departments and services. The process of conducting an inventory of production assets is also simplified and takes less time to learn, increasing the effectiveness of cooperation between departments.

1.3. Advantages of automation system implementation

First of all, the introduction of accounting with software affects the financial and economic services. Economists are able to accurately calculate the budget needed for repairs and correctly spread the costs of financial accounting items. The allocation of funds for them becomes transparent and clear. As a result, the efficiency of the use of assets increases significantly, their structure changes for the better.

First of all, it is the simplicity of accounting, control and statistics. It becomes easy to draw up and control schedules for inspection and repair of tools, the technical documentation is put in order. It is not only convenient, but also profitable at work.

It is also important that technical services are beginning to pay more attention to the efficiency of equipment use, its reliability. They also have the opportunity to clearly justify the need for financial and material resources, based on reliable information on the state of instruments. As a result, repair services are less idle, which means that the efficiency of the enterprise itself increases.

Experience shows that by introducing an automated system of enterprise resource management and statistical analysis, it is possible to achieve that restoration repairs will be carried out 20% less often. And this saves time, money and human resources.

Timely information from services and available data for past periods, allow MTO services to more clearly plan their work and optimize inventory. Their employees can quickly provide repairmen with the necessary materials, more accurately determine what may be needed in the future and make appropriate requests for supply.

HR professionals, using the openness and transparency of the system, have the opportunity to develop and implement an effective personnel policy and a simple clear system of increasing staff motivation.

We should not forget about the shareholders or owners of the enterprise. Automated control systems allow you to easily and reasonably explain the need for certain costs. Reduction of technological risks and increase of return of tools, optimization of work of "MTO" services allows to save a considerable part of expenses and at the expense of it to increase profit.

Starting work on the implementation of process automation, first of all, you should understand - what they include. That is, we actually automate.

Maintenance usually includes:

- Collection and storage of information - it includes data on musical instruments, including resources needed to repair it;
- Management of "MTO", including planning of work of a part of these services which are engaged in maintenance of repair divisions - it is necessary for maintenance of optimum activity of services of "MTO" and a warehouse on supply of repairmen with spare parts and other materials;
- Direct management of repair works;
- Management of financial issues related to - including the formation of quantitative indicators for production assets using statistical information.

Most often, two systems are used to automate processes - repair modules "ERP-systems" or a full-fledged "EAM-system". Both allow to carry out both planned (calendar, as of, service, independent, capital, current, etc.), and unscheduled (because of sudden failure of the equipment) repairs.

In favor of repair modules "ERP-systems", as a rule, those who carry out automation of all processes of the enterprise at the same time make the choice. It should be remembered that "ERP-systems" are not able to cover all business processes and the maximum that can be achieved from them is equal to - 60%. In any case, such systems will later have to be refined and expanded a set of integrated solutions.

These tools were compared according to the following criteria:

- availability of technical and economic standards for repair work;

- availability of time norms for repair works;

- availability of maps of technological (repair) processes;

- display of equipment status;

- availability of project technological documentation;

- design documentation;

- the possibility of planning a system of timely prevention and provision of work;

- the method of entering the initial data about the equipment available and its characteristics during the implementation of the system (there are two options: manual filling of the database by operators, or collecting information from the system);

- the possibility of integration with existing control systems (An important characteristic that shows how in detail the EAM-system will reflect the state of the equipment);

- the possibility of integration with existing control systems (An important characteristic that shows how in detail the EAM-system will reflect the state of the equipment);

- the ability to optimize the purchased assets in the warehouse, which allows you to optimally plan the volume of purchases - to avoid the purchase of some components to the detriment of others;

- support for the log of repairs and maintenance of equipment to collect statistics and identify patterns, and through them the causes of breakdowns;
- use for planning repairs of the internal software module or external information system;
- support of composite (component) outfits;
- the possibility of differentiating the rights of users (the presence of a security system);
- support for users through the web interface (submission of applications, monitoring the progress of work);
- technical requirements and the necessary platform for the client.

In the above list, the first six criteria characterize such an important part of any information system as database maintenance. In the case of an EAM system, the databases contain certification information (data on existing equipment).

The introduction of a specialized "EAM-system" does not automate all management processes at once, in the future will have to integrate all solutions into a single system.

There are the following benefits received by their companies from the introduction of management systems of production assets:

- increasing the service life of equipment;
- increasing the productivity of repair services;
- increase the productivity of tools;
- more prompt execution of restoration repairs;
- reduction of unplanned downtime;
- increase in the coefficient of readiness (service life);
- reducing the number of breakdowns;
- increasing the return on the company's fixed assets;
- effective implementation of systems of this class allows the company to reduce the cost of maintaining music in working order due to;
- proper organization of repairs;

- creation of an effective inventory management system and supply system;
- full control of maintenance and repair costs;
- creating a history of equipment;
- other measures, the application of which in Western enterprises has long been an element of management culture.

The results of the implementation of (EAM) -systems in the world indicate their extremely high return. The vast majority of projects pay for themselves in less than one and a half to two years. A reduction of 20% or more in repair costs is typical.

With regard to quantitative estimates, according to the research of the consulting group, the studied cases of application of the systems were characterized by obtaining, on average, the following benefits:

- increase in productivity of works by 29%;
- increase the coefficient of readiness by 17%;
- reduction of cases of stock shortage by 29%;
- increase in the share of scheduled repairs by 78%;
- reduction of overtime work by 22%;
- reduction of waiting time for spare parts by 29%;
- more favorable prices for purchased goods and materials by 18%.

If we express the improvement of the system in numerical values, the changes after automation can reach the following values:

1. Improving economic performance:

1.1. increase of productivity of works to 55%;

1.2. increase the readiness factor by 17%;

1.3. reduction of capital expenditures (replacements) by 50-90%;

1.4. reduction of operating costs by 10-40%;

2. Reduction of costs for excess goods and materials:

2.1. more favorable prices for purchased goods and materials by 18%;

2.2. reduction of urgent purchases of goods and materials by 29%;

3. Increasing the degree of propensity to plan repairs:

3.1. increase in the share of scheduled repairs by 78%;

3.2. reduction of unscheduled works and simple EIA equipment by 30-40%;

4. Reduction of time indicators:

4.1. reduction of repair duration by 20-50%;

4.2. reduction of waiting time for spare parts by 29%.

The obtained results allow to carry out the economic substantiation of works on automation. As prospects for further development, the results obtained will be developed into a method of assessing economic efficiency, which, ultimately, will intensify the development and implementation of automated process support systems.

Automation can be one step towards a modern enterprise. But only a clear strategy, which covers the right goals and objectives, will be able to bring such an enterprise to a leading position in the market.

Work kinds	Service description
Simple repair	Incomplete disassembly - assembly of equipment, lubrication, cleaning of optics, prevention of the device, replacement of: fuses, buttons, regulators, broken conductors, heads without adjustment (carrying out minor repairs of the equipment)
Complex repair	Deep disassembly - assembly, replacement of chips and faulty elements, mechanical components and parts, setting up the device. Deep cleaning, debugging and repair of the video projector
Non-standard repairs	Faults that were associated with improper operation of the equipment (overvoltage in the network, mechanical actions,

	flooding, etc.)
Diagnostic	Carrying out disassembly - cleaning, identification of the reasons of malfunction, preparation of the estimate of repair work (at carrying out repair work the cost of diagnostics is included in the price of repair and is not paid separately)
Guitar service	
Tool maintenance	- Replacement of a set of pegs on any type of guitars, without taking into account the cost of pegs (without plugging old holes) + reconstruction; - Replacement of a set of pegs on any type of guitars, without taking into account the cost of pegs (with clogging of old holes and) + reconstruction; - Replacement of strings on any type of guitar with a fixed bridge; - Replacement of a threshold (top / bottom) or adjustment of height established earlier; - Reconstruction of the measuring distance and the optimal height of the strings above the fingerboard on any type of acoustic guitars; - Reconstruction of beaker distance and optimum. the height of the strings above the fingerboard on electric guitars with tremolo.
Working with the fingerboard	- Replacement of an anchor rod (bilateral) + replacement of an overlay; - Replacement of frets on any type of guitars, without taking into account the cost of frets; - Replacement of frets on any type of guitars with a varnished overlay of a fretboard, without consideration of cost of frets (the price includes repainting of an overlay); - Replacement of the fingerboard on any type of guitar (wood of choice: rosewood, ebony, maple); - Alignment of geometry of an overlay of a fingerboard;

	- Folding the neck.
Working with the case	- Coloring of a guitar fretboard, monophonic color, glossy, matte varnish; - Painting the body of the guitar, solid color, matte varnish; - Restoration and gluing of the junction of the body and the fingerboard (five) guitars; - Restoration and gluing of a string holder of acoustic and classical guitars with painting; - Restoration and gluing of a string holder of acoustic and classical guitars without painting; - Bonding and restoration of the head of the guitar fretboard without further painting;

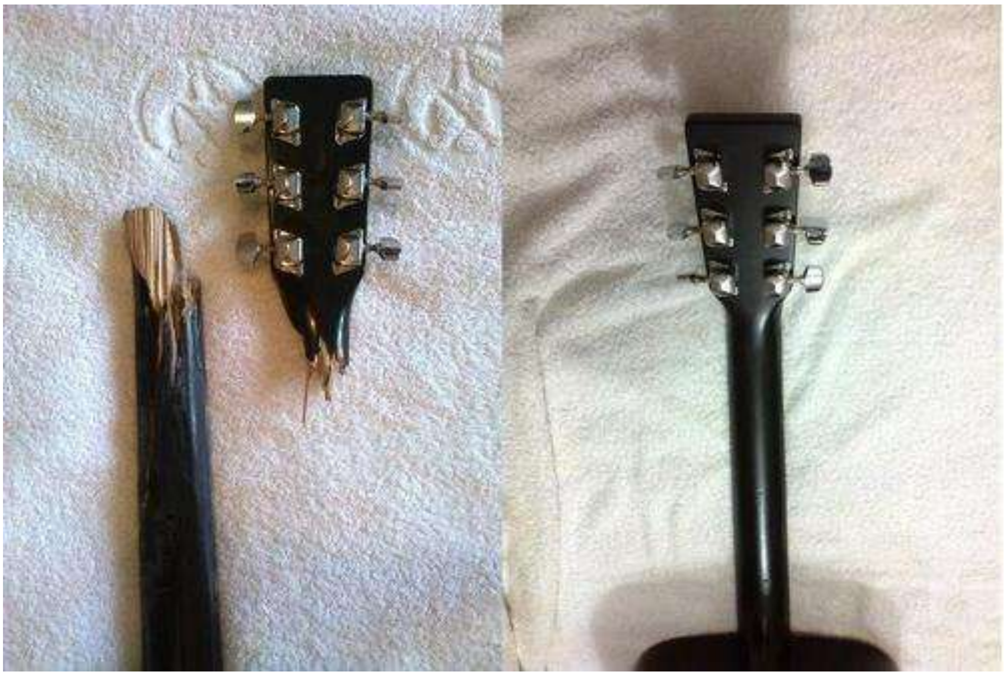


Fig.1.1 An example of guitar repair

Conclusion

In the first section the detailed analysis of work of system of repair agency of musical instruments is carried out. Describes in detail how maintenance is performed. The main advantages of automation implementation for the repair agency system are also named.

PART 2
REQUIREMENTS FOR THE REPAIR AGENCY SYSTEM

2.1. Analysis of software requirements

Before you start a project, be sure to know what result (product) you want to get. And sometimes this product needs to be described most carefully. In other words, you need to know what requirements the customer has for the product. The complete set of these requirements is called the requirements catalog, or specification.

Large and complex projects usually have thousands of requirements. Business analysis allows you to identify problems and determine what is needed to overcome them. In large projects, such as software development, requirements collection is one of the most important stages of a project life cycle, which can take several weeks / months.

To identify the requirements, a series of structured interviews with customers are conducted, which allow to accurately determine their wishes for the finished product. Trying to find out directly from the customer what results he needs may end in failure: the customer will make more and more new requirements, so you just will not be able to meet them. Remember, any requirement affects the duration and cost of the project. Accordingly, when receiving a detailed list of requirements, you need to know if they are:

- Required
- Desirable
- Optional

2.2. Functional requirements for the project

Functional requirements are the tasks for the developer. Everything that is not specified in the requirements is done at the discretion of the developer, which often differs from the product manager's view of the expected result. Therefore, the requirements must contain answers to all possible questions on the task.

Functional requirements for the authorization page:

- The client must be able to register in the system
- The client must be able to log in to the system
- The client must be able to select the interface language
- During registration, the client must enter the login e-mail and password
- The client must enter an email and password during authorization

Functional requirements for the home page:

- The client must be able to select the interface language
- The client must be able to move to “Login” page
- The client must be able to move to “Registration” page

Functional requirements for the user page:

- The client must be able to select the interface language
- The client should see a link on “Create Repair request” page
- The client should see a link on “My Request List” page
- The client must have the opportunity to Logout
- The client should see the main description of service

Functional requirements for the client “Create Repair request” page:

- The client must be able to select the interface language
- The client must have the opportunity to Logout
- The client should have the possibility to make a repair request
- The client should have the possibility write theme and description of repair

• The client should see an “green” alert if request was send sucessfuly

• The client should see an “red” warning-alert if request was send with troubles

Functional requirements for the client “My Requests List” page:

• The client must be able to select the interface language

• The client must have the opportunity to Logout

• The client should see a list of all his repair requests

• The request should consist of: Theme, Description, Price, Status, Feedback fields and Send feedback button

• The client is unavailable to send feedback till the request was processed by manager and closed by master

• If client try to send feedback till repair request is not closed, he should see warning alert

• If send feedback after repair request is closed, he should see a success alert

Functional requirements for the manager’s home page:

• The manager must be able to select the interface language

• The manager should see a link on “Repair requests List” page

• The manager must have the opportunity to Logout

• The manager should see the main description of service

Functional requirements for the manager “Repair Request List” page:

• The manager must be able to select the interface language

• The manager must have the opportunity to Logout

• The manager should see a list of all repair requests from all users

• The request should consist of: Theme, Description, Request creator, Price fields and Accept and Decline buttons

• The manager must be able to specify repair price and accept the request

• The manager must be able to click decline button and after confirmation window will appear

• The manager should see a success alert after acception of request

• After manager accept specify the price, the price should appear in repair list in user whose request was accepted

Functional requirements for the master's home page:

- The manager must be able to select the interface language
- The manager should see a link on "Check Requests List" page
- The manager must have the opportunity to Logout
- The manager should see the main description of service

Functional requirements for the master "Check Requests List" page:

- The master must be able to select the interface language
- The master must have the opportunity to Logout
- The master should see a list of all repair requests accepted by manager with price
- The request should consist of: Theme, Description, Request creator, Price fields and Close button
- The master must be able to close the repair request
- The master should see successful alert if repair request was closed sucessfuly
- After master close the request, the status should be changed in repair list in user whose request was closed

2.3. Non-functional requirements for the project

In addition to standard requirements for behavior (or functionality), it is extremely important to identify and document so-called non-functional requirements. They will be responsible for the stability and reliability of the program, which is no less important assessment of the success of the project, after the developed application allows you to perform the main functions assigned to it.

First of all, when we write about the requirements for the design of a web service, namely their development, we must remember that they are part of the process of designing a site. Design requirements should allow any user to manage the site, even without knowledge of web programming. Of course, how the design requirements will be made and how they will be implemented, so will the efficiency of the site. After all, design is the second most important component of an effective site. It depends on how attractive the page is, and this ultimately further attracts potential users and customers.

The design requirements for my web service are as follows:

- the minimum number of images in order not to overload the user and the service;
- Own style of the site, which helps it to differ from other sites and be remembered by the user;
- The web page should be minimized to get the smallest size in order to load as quickly as possible;
- transparent and logical structure of information placement;
- no spelling and syntactic errors;
- navigation on the Web page - in fact, it does not allow the user to get lost in the site;
- a button that automatically allows you to display the site interface for the main user actions;
- own logo;
- site design should be modern and concise;
- adaptive design;
- integrity of site design.

Over the years, the demands on modern design have increased as the number of websites has grown. Having opened an illiterate and ugly page, the user will want to leave it soon and look for another, more pleasant to read and general perception.

Of course, it is possible and necessary to adhere to the requirements that are standard, which apply to the design itself and its location, and in principle, everything related to it. But keep in mind that there are a lot of similar design sites, all of which are designed to meet general requirements, so I think there should be a unique design.

Also, one of the main requirements will be to create a layout that will reflect the design of the web service. Mostly layouts are drawn in Adobe Photoshop or Gimp. Layout is one of the main steps in creating a website.

The main task of web service design requirements, in my opinion, is to build a bridge between a business owner and a potential customer.

When we make requirements for site design, there should not be present:

- elements that would distract users from viewing information;
- complexly structured information, difficult to perceive;
- sections that fall out of the overall design of the site.

The home page of the website was designed as required. It consists of all the main structural elements. The development of the site includes the process of creating a layout of web pages, which will later be strung all the other elements.

A well-designed website is a pleasure to use. It attracts the attention of users, conveys information that they may need, and convinces them to connect to the service. The website also reflects the company's attitude towards its customers and their ability to keep up with the times.

When creating requirements for the design of a web service from previous experience, the main rules were made for themselves:

1. Reduce the user's cognitive effort

You need to make the web page obvious. This should not require detailed explanations for its operation. You need to keep the navigation structure of the site intuitive and clear.

2. Do not waste user time on trifles

When you need a user to subscribe to a service, the process should be as simple as possible. 3-4 steps on one page usually limit the number of steps a user is willing to perform to subscribe to your service. If you need more steps, place them sequentially on one page.

3. Trying to focus the visitor

If a site sells a product for a limited period of time, you should use a web page design to focus the user's attention on it. Do not use pop-up style banners to convey a message.

4. Effectively demonstrate the capabilities of the site

If the site offers 9 sections that provide access to other pages, you need to make sure that all 9 main titles are easily visible. In doing so, it is possible to effectively demonstrate everything I offer.

5. Write effective content

broken down into categories and titles that are easy for the user to view.

6. Do not complicate things

Users typically visit a website to access information, not design. In fact, they are looking for information despite the design. Try to simplify this process by making things as simple as possible.

7. Use spaces effectively

Using space on your website does not mean that it will be cluttered. In most cases, this is a battle to make things as simple as possible. Using empty space on your site to differentiate categories is an effective design strategy.

8. Communicate effectively

When using a website as a means of visual communication, be sure to check the content and design for what needs to be communicated and the clarity with which it is communicated. Also, keep visual cues in a way to enhance distinction and emphasize.

9. Use familiar formats

Small things, such as placing a search box in places where users expect it, using familiar terms such as registering and logging in, and placing basic links in familiar places, will help reduce the user learning curve for your website.

10. Create a logo

A graphic or text symbol that represents a specific object, such as a company, organization, individual, or product. And when a person recognizes the image of the logo known to him, the person knows that the thing marked with the logo belongs to this company and if the trust is already established, the user will be immediately positive, which will allow to interact with him.

Conclusion

Therefore, when creating requirements for a web service and its design, it is necessary to clearly understand the stages of creation and implementation of the site, to have direct experience in this topic to create a more productive web service. Keep in mind that individual requirements are what will help the site to stand out among others and be remembered by the user in the implementation and in the future. There is a possibility that a developer working in this field may miss the creation of requirements, because he still remembers them, but he does not miss this stage and automatically already implements the site taking into account the requirements.

In addition to the requirements that have been described, there are a number of no less important requirements: technical requirements for the content of the site, the design of the site, the content management system and the like. It was the requirements for web service and design that were described, because they played an important role in the implementation of my site and took a lot of my time, because I hesitated over them for a long time.

Many factors need to be considered when creating requirements. Of course, to develop a quality site, you need to know the basic languages used in writing sites and search engine promotion technology, but only if you follow all the rules you can count on the successful development and quality of the web resource and requirements are no exception.

A true professional will not miss any of the stages of creating a web service and knows that creating requirements is one of the most important steps.

PART 3
SYSTEM STRUCTURE

3.1. Project architecture

The software product is created using Spring Boot framework in the Java programming language paired with ThymeLeaf to render HTML templates from server. The main pattern of developing this project is MVC (Model, View, Controller). As a database I chose MySQL. Also I used a build automation tool – Maven which is primarily used for Java projects. For automatic connection with database I used Hibernate framework. And to give the project good appearence I used some Bootstrap to make it crossbrowser.

3.2. Spring Boot

Spring Boot is a brand new framework from the team at Pivotal, designed to simplify the bootstrapping and development of a new Spring application. The framework takes an opinionated approach to configuration, freeing developers from the need to define boilerplate configuration. In that, Boot aims to be a front-runner in the ever-expanding rapid application development space.

The Spring IO platform has been criticized over the years for having bulky XML configuration with complex dependency management. During last year’s SpringOne 2GX conference, Pivotal CTO, Adrian Colyer acknowledged those criticisms, and made special note that a goal of the platform going forward is to embrace an XML-free development experience. Boot takes that mission statement to the extreme, not only freeing developers from the need for XML, but also, in some scenarios, releasing them from the tedium of writing import statements. In the days following its public beta release, Boot gained some viral popularity by demonstrating the framework’s simplicity with a runnable web application that fit in under 140-characters, delivered in a tweet.

Spring Boot is not, however, an alternative to the many projects that comprise the "Foundation" layer of the Spring IO platform. Indeed, the goal of Spring Boot is not to provide new solutions for the many problem domains already solved, but rather to leverage the platform in fostering a development experience that simplifies the use of those already-available technologies. This makes Boot an ideal choice for developers who are familiar with the Spring ecosystem, while also catering to new adopters by allowing them to embrace Spring technologies in a simplified manner.

In pursuit of such an improved development experience, Spring Boot — and, indeed, the entire Spring ecosystem — has embraced the Groovy programming language. Groovy’s powerful MetaObject protocol, pluggable AST transformation process, and embedded dependency resolution engine are what facilitate many of the shortcuts that Boot affords. At the core of its compilation model, Boot utilizes Groovy to build project files, so that it can decorate a class' generated bytecode with common imports and boilerplate methods, such as a class' main method. This allows applications written with Boot to remain concise, while still offering a breadth of functionality.

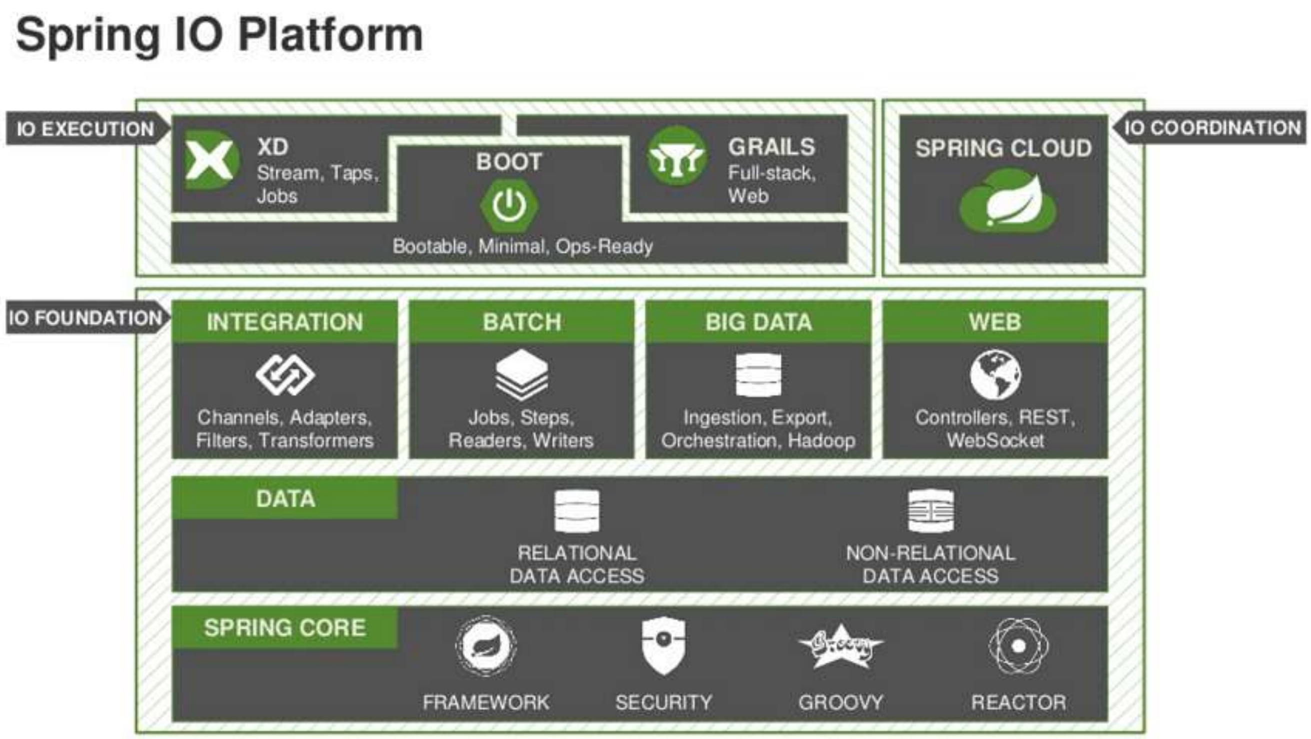


Fig.3.1 Spring IO Platform

Spring Boot is a project at the IO Execution level (runtime) of the IO Spring Framework

Spring Boot is the next step in Spring to make it easier to configure and develop applications. With Spring Boot, Spring configuration is minimized as much as possible. Spring Boot supports embedded containers, which allows web applications to run independently and without the need for a Web Server.

You can use spring Boot to create a Java Web application running via the "Java-jar" line command, or export a War file to use on the Web Server as usual. Spring Boot gives you a "CLI Tool" for running spring scripts

Spring Boot can be explained simply by the illustration below:

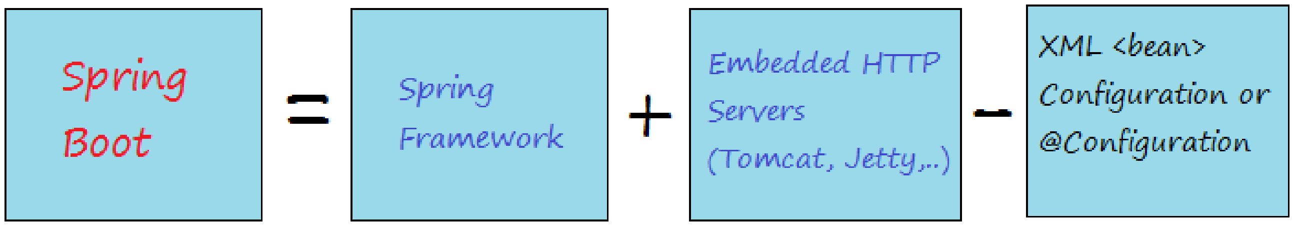

Spring Boot
=
Spring Framework
+
Embedded HTTP Servers (Tomcat, Jetty,..)
–
XML <bean> Configuration or @Configuration

Fig.3.2 Simple explanation of Spring Boot

The following are the benefits of Spring Boot:

- Easily used to develop a Spring based application with Java or Groovy Spring

- Minimizes development time and raises productivity
- Avoids writing a lot of boilerplate, Annotations and XML configuration
- Easily allows you to interact with Spring Boot applications with Spring ecosystems like Spring JDBC, Spring ORM, Spring Data, Spring Security, etc.
- Follows the "Default Configuration Principles" approach to minimize the time and effort invested in developing applications.
- Provides Embedded HTTP servers like Tomcat, Jetty ... to quickly and easily develop and test web applications
- Provides CLI (Command Line Interface) tools for developing and testing Spring Boot applications (Java or Groovy) from the command prompt very easily and quickly
- Provides many plugins for quickly developing and testing Spring Boot applications using Build tools like Maven and Gradle

  Offers many plugins for easy handling of embedded databases and in-memory Databases.

3.3. MVC pattern

The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller. This is done to separate internal representations of information from the ways information is presented to and accepted from the user. Traditionally used for Desktop GUI apps, the pattern became popular with the advent of web apps. Today almost all popular languages support this architecture.

Components

- The model is the central component of the pattern. It is the application’s dynamic data structure, independent of the user interface. It corresponds to all the data-related logic that the user works with. It directly manages the data, logic and rules of the application.

- A view can be any output representation of information, such as a chart or a diagram. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.
- The third part or section, the controller, accepts input and converts it to commands for the model or view. It acts as an interface between Model and View components to process all the business logic and incoming requests, manipulates data using the Model component and interacts with the Views to render the final output.

Interactions

In addition to dividing the application into three kinds of components, the model–view–controller design defines the interactions between them.

The model is responsible for managing the data of the application. It receives user input from the controller.

The view means the presentation of the model in a particular format.

The controller responds to the user input and performs interactions on the data model objects. The controller receives the input, optionally validates it and then passes the input to the model.

These interactions can be aptly represented by the following flow diagram:

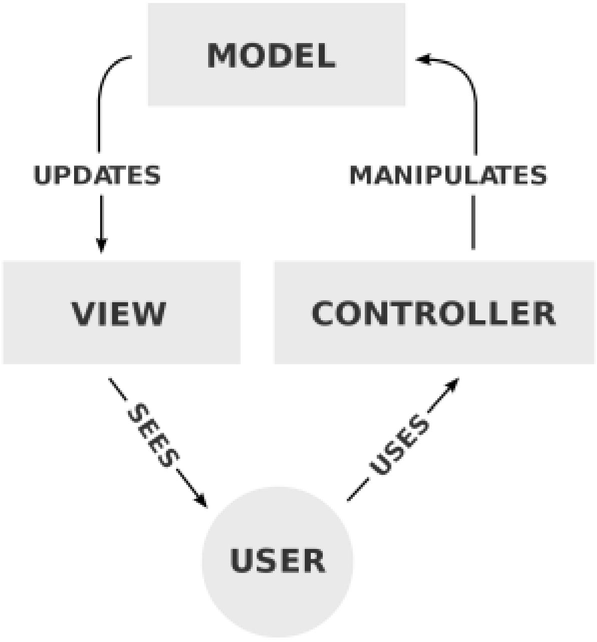


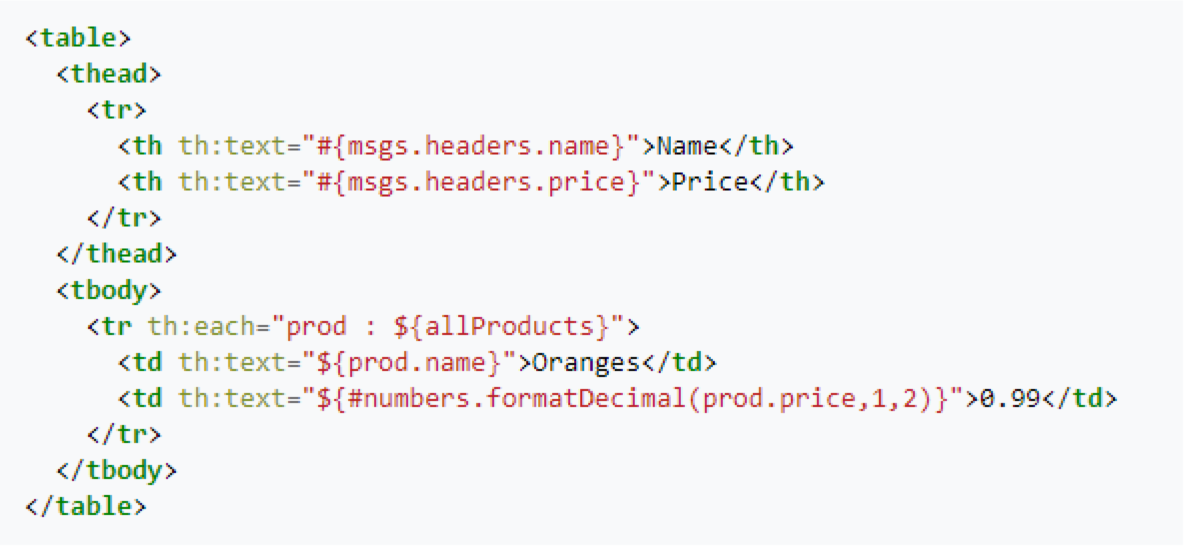
Fig.3.3 MVC Interaction Pattern

The following advantages of the MVC design pattern make it suitable to be used in software development:

- Simultaneous development — Multiple developers can work simultaneously on the model, controller and views.
- High cohesion — MVC enables logical grouping of related actions on a controller together. The views for a specific model are also grouped together.
- Low coupling — The very nature of the MVC framework is such that there is low coupling among models, views or controllers
- Ease of modification — Because of the separation of responsibilities, future development or modification is easier
- Multiple views for a model — Models can have multiple views

3.4. ThymeLeaf

Thymeleaf is a Java XML/XHTML/HTML5 template engine that can work both in web (servlet-based) and non-web environments. It is better suited for serving XHTML/HTML5 at the view layer of MVC-based web applications, but it can process any XML file even in offline environments. It provides full Spring Framework integration.



```
<table>
  <thead>
    <tr>
      <th th:text="#{msgs.headers.name}">Name</th>
      <th th:text="#{msgs.headers.price}">Price</th>
    </tr>
  </thead>
  <tbody>
    <tr th:each="prod : ${allProducts}">
      <td th:text="${prod.name}">Oranges</td>
      <td th:text="${#numbers.formatDecimal(prod.price,1,2)}">0.99</td>
    </tr>
  </tbody>
</table>
```

Fig.3.4 ThymeLeaf using example

3.5. MySQL database

What is MySQL?

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use
- MySQL uses standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use
- MySQL is developed, distributed, and supported by Oracle Corporation
- MySQL is named after co-founder Monty Widenius's daughter: My

The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows.

Databases are useful for storing information categorically. A company may have a database with the following tables:

- Employees
- Products
- Customers
- Orders

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

MySQL is an important component of an open source enterprise stack called LAMP. LAMP is a web development platform that uses Linux as the operating system, Apache as the web server, MySQL as the relational database management system and PHP as the object-oriented scripting language. (Sometimes Perl or Python is used instead of PHP.)

Originally conceived by the Swedish company MySQL AB, MySQL was acquired by Sun Microsystems in 2008 and then by Oracle when it bought Sun in 2010. Developers can use MySQL under the GNU General Public License (GPL), but enterprises must obtain a commercial license from Oracle.

Today, MySQL is the RDBMS behind many of the top websites in the world and countless corporate and consumer-facing web-based applications, including Facebook, Twitter and YouTube.

How MySQL works

MySQL is based on a client-server model. The core of MySQL is MySQL server, which handles all of the database instructions (or commands). MySQL server is available as a separate program for use in a client-server networked environment and as a library that can be embedded (or linked) into seperate applications.

MySQL operates along with several utility programs which support the administration of MySQL databases. Commands are sent to MySQLServer via the MySQL client, which is installed on a computer.

MySQL was originally developed to handle large databases quickly. Although MySQL is typically installed on only one machine, it is able to send the database to multiple locations, as users are able to access it via different MySQL client interfaces. These interfaces send SQL statements to the server and then display the results.

Core MySQL features

MySQL enables data to be stored and accessed across multiple storage engines, including InnoDB, CSV, and NDB. MySQL is also capable of replicating data and partitioning tables for better performance and durability. MySQL users aren't required to learn new commands; they can access their data using standard SQL commands.

Getting started with MySQL tutorial for beginners

MySQL is written in C and C++ and accessible and available across over 20 platforms, including Mac, Windows, Linux and Unix. The RDBMS supports large databases with millions records and supports many data types including signed or unsigned integers 1, 2, 3, 4, and 8 bytes long; FLOAT; DOUBLE; CHAR; VARCHAR; BINARY; VARBINARY; TEXT; BLOB; DATE; TIME; DATETIME; TIMESTAMP; YEAR; SET; ENUM; and OpenGIS spatial types. Fixed- and variable-length string types are also supported.

For security, MySQL uses an access privilege and encrypted password system that enables host-based verification. MySQL clients can connect to MySQL Server using several protocols, including TCP/IP sockets on any platform. MySQL also supports a number of client and utility programs, command-line programs and administration tools such as MySQL Workbench.

Offshoots of MySQL, also known as forks, include the following:

- Drizzle, a lightweight open source database management system in development based on MySQL 6.0;
- MariaDB, a popular community-developed "drop-in" replacement for MySQL that uses MySQL APIs and commands; and
- Percona Server with XtraDB, an enhanced version of MySQL known for horizontal scalability.

MySQL vs SQL

Before 2016, the main difference between MySQL and SQL was that the former could be used on multiple platforms, whereas the latter could only be used on Windows. Microsoft has since expanded SQL to support Linux, a change which went into effect in 2017. When MySQL is installed via Linux, its package management system requires custom configuration to adjust security and optimization settings.

MySQL also allows users to choose the most effective storage engine for any given table, as the program is able to utilize multiple storage engines for individual tables. One of MySQL's engines is InnoDB. InnoDB was designed for high availability.

Because of this, it is not as quick as other engines. SQL uses its own storage system, but it does maintain multiple safeguards against loss of data. Both systems are able to run in clusters for high availability.

SQL Server offers a wide variety of data analysis and reporting tools. SQL Server Reporting Services is the most popular one and is available as a free download. There are similar analysis tools for MySQL available from third-party software companies, such as Crystal Reports XI and Actuate BIRT.

3.6. Maven

Maven, a Yiddish word meaning accumulator of knowledge, began as an attempt to simplify the build processes in the Jakarta Turbine project. There were several projects, each with their own Ant build files, that were all slightly different. JARs were checked into CVS. We wanted a standard way to build the projects, a clear definition of what the project consisted of, an easy way to publish project information, and a way to share JARs across several projects.

The result is a tool that can now be used for building and managing any Java-based project. We hope that we have created something that will make the day-to-day work of Java developers easier and generally help with the comprehension of any Java-based project.

Maven’s Objectives

Maven’s primary goal is to allow a developer to comprehend the complete state of a development effort in the shortest period of time. In order to attain this goal, Maven deals with several areas of concern:

- Making the build process easy
- Providing a uniform build system
- Providing quality project information

- Encouraging better development practices
- Making the build process easy

While using Maven doesn’t eliminate the need to know about the underlying mechanisms, Maven does shield developers from many details.

Providing a uniform build system

Maven builds a project using its project object model (POM) and a set of plugins. Once you familiarize yourself with one Maven project, you know how all Maven projects build. This saves time when navigating many projects.

Providing quality project information

Maven provides useful project information that is in part taken from your POM and in part generated from your project’s sources. For example, Maven can provide:

- Change log created directly from source control
- Cross referenced sources
- Mailing lists managed by the project
- Dependencies used by the project
- Unit test reports including coverage
- Third party code analysis products also provide Maven plugins that add their reports to the standard information given by Maven.

Providing guidelines for best practices development

Maven aims to gather current principles for best practices development and make it easy to guide a project in that direction.

For example, specification, execution, and reporting of unit tests are part of the normal build cycle using Maven. Current unit testing best practices were used as guidelines:

- Keeping test source code in a separate, but parallel source tree
- Using test case naming conventions to locate and execute tests
- Having test cases setup their environment instead of customizing the build for test preparation

Maven also assists in project workflow such as release and issue management.

Maven also suggests some guidelines on how to layout your project's directory structure. Once you learn the layout, you can easily navigate other projects that use Maven.

While takes an opinionated approach to project layout, some projects may not fit with this structure for historical reasons. While Maven is designed to be flexible to the needs of different projects, it cannot cater to every situation without compromising its objectives.

If your project has an unusual build structure that cannot be reorganized, you may have to forgo some features or the use of Maven altogether.

What is Maven Not?

You might have heard some of the following things about Maven:

- Maven is a site and documentation tool
- Maven extends Ant to let you download dependencies
- Maven is a set of reusable Ant scriptlets

Feature Summary

The following are the key features of Maven in a nutshell:

- Simple project setup that follows best practices - get a new project or module started in seconds
- Consistent usage across all projects - means no ramp up time for new developers coming onto a project

- Superior dependency management including automatic updating, dependency closures (also known as transitive dependencies)
- Able to easily work with multiple projects at the same time
- A large and growing repository of libraries and metadata to use out of the box, and arrangements in place with the largest Open Source projects for real-time availability of their latest releases
- Extensible, with the ability to easily write plugins in Java or scripting languages
- Instant access to new features with little or no extra configuration
- Ant tasks for dependency management and deployment outside of Maven
- Model based builds: Maven is able to build any number of projects into predefined output types such as a JAR, WAR, or distribution based on metadata about the project, without the need to do any scripting in most cases.
- Coherent site of project information: Using the same metadata as for the build process, Maven is able to generate a web site or PDF including any documentation you care to add, and adds to that standard reports about the state of development of the project. Examples of this information can be seen at the bottom of the left-hand navigation of this site under the "Project Information" and "Project Reports" submenus.
- Release management and distribution publication: Without much additional configuration, Maven will integrate with your source control system (such as Subversion or Git) and manage the release of a project based on a certain tag. It can also publish this to a distribution location for use by other projects. Maven is able to publish individual outputs such as a JAR, an archive including other dependencies and documentation, or as a source distribution.
- Dependency management: Maven encourages the use of a central repository of JARs and other dependencies. Maven comes with a mechanism that your project's clients can use to download any JARs required for building your project from a central JAR repository much like Perl's CPAN. This allows users of Maven to reuse JARs across projects and encourages communication between projects to ensure that backward compatibility issues are dealt with.

3.7. Hibernate

Hibernate is an Object-Relational Mapping (ORM) solution for JAVA. It is an open source persistent framework created by Gavin King in 2001. It is a powerful, high performance Object-Relational Persistence and Query service for any Java Application.

Hibernate maps Java classes to database tables and from Java data types to SQL data types and relieves the developer from 95% of common data persistence related programming tasks.

Hibernate sits between traditional Java objects and database server to handle all the works in persisting those objects based on the appropriate O/R mechanisms and patterns.

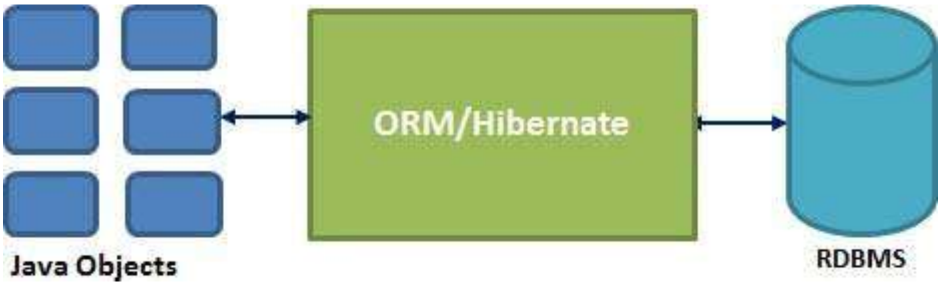


Fig.3.5 Hibernate Position

Hibernate Advantages

- Hibernate takes care of mapping Java classes to database tables using XML files and without writing any line of code.
- Provides simple APIs for storing and retrieving Java objects directly to and from the database.
- If there is change in the database or in any table, then you need to change the XML file properties only.
- Abstracts away the unfamiliar SQL types and provides a way to work around familiar Java Objects.
- Hibernate does not require an application server to operate.
- Manipulates Complex associations of objects of your database.
- Minimizes database access with smart fetching strategies.

- Provides simple querying of data.

Supported Databases

Hibernate supports almost all the major RDBMS. Following is a list of few of the database engines supported by Hibernate

- HSQL Database Engine
- DB2/NT
- MySQL
- PostgreSQL
- FrontBase
- Oracle
- Microsoft SQL Server Database
- Sybase SQL Server
- Informix Dynamic Server

Supported Technologies

Hibernate supports a variety of other technologies, including

- XDoclet Spring
- J2EE
- Eclipse plug-ins
- Maven

In order to build the project, Maven provides developers with options to mention life-cycle goals and project dependencies (that rely on Maven plugin capabilities and on its default conventions). Much of the project management and build related tasks are maintained by Maven plugins.

Developers can build any given Maven project without the need to understand how the individual plugins work. We will discuss Maven Plugins in detail in the later chapters.

Features of Maven:

- Simple project setup that follows best practices
- Consistent usage across all projects
- Dependency management including automatic updating
- A large and growing repository of libraries
- Extensible, with the ability to easily write plugins in Java or scripting languages
- Instant access to new features with little or no extra configuration
- Model-based builds − Maven is able to build any number of projects into predefined output types such as jar, war, metadata
- Coherent site of project information − Using the same metadata as per the build process, maven is able to generate a website and a PDF including complete documentation
- Release management and distribution publication − Without additional configuration, maven will integrate with your source control system such as CVS and manages the release of a project
- Backward Compatibility − You can easily port the multiple modules of a project into Maven 3 from older versions of Maven. It can support the older versions also
- Automatic parent versioning − No need to specify the parent in the sub module for maintenance
- Parallel builds − It analyzes the project dependency graph and enables you to build schedule modules in parallel. Using this, you can achieve the performance improvements of 20-50%
- Better Error and Integrity Reporting − Maven improved error reporting, and it provides you with a link to the Maven wiki page where you will get full description of the error

3.8. Bootstrap

What is Twitter Bootstrap?

Bootstrap is a sleek, intuitive, and powerful, mobile first front-end framework for faster and easier web development. It uses HTML, CSS and Javascript.

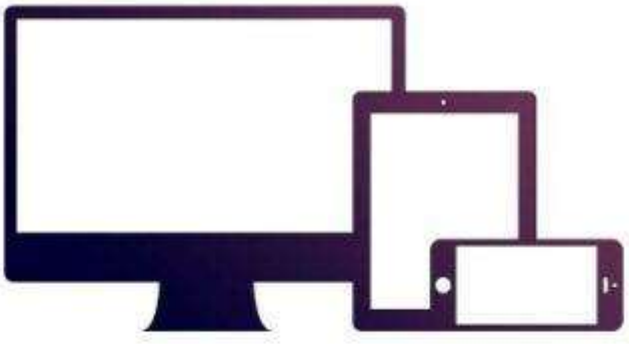
History

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter. It was released as an open source product in August 2011 on GitHub.

Why Use Bootstrap?

- Mobile first approach − Bootstrap 3, framework consists of Mobile first styles throughout the entire library instead them of in separate files.
- Browser Support − It is supported by all popular browsers.

- Easy to get started − With just the knowledge of HTML and CSS anyone can get started with Bootstrap. Also the Bootstrap official site has a good documentation.
- Responsive design − Bootstrap's responsive CSS adjusts to Desktops, Tablets and Mobiles. More about the responsive design is in the chapter Bootstrap Responsive Design.



- Provides a clean and uniform solution for building an interface for developers.
- It contains beautiful and functional built-in components which are easy to customize.
- It also provides web based customization.
- And best of all it is an open source.

What Bootstrap Package Includes?

- Scaffolding − Bootstrap provides a basic structure with Grid System, link styles, and background. This is is covered in detail in the section Bootstrap Basic Structure
- CSS − Bootstrap comes with the feature of global CSS settings, fundamental HTML elements styled and enhanced with extensible classes, and an advanced grid system. This is covered in detail in the section Bootstrap with CSS.
- Components − Bootstrap contains over a dozen reusable components built to provide iconography, dropdowns, navigation, alerts, pop-overs, and much more. This is covered in detail in the section Layout Components.
- JavaScript Plugins − Bootstrap contains over a dozen custom jQuery plugins. You can easily include them all, or one by one. This is covered in details in the section Bootstrap Plugins.
- Customize − You can customize Bootstrap's components, LESS variables, and jQuery plugins to get your very own version.

3.9. Spring Security

Spring Security is a powerful and highly customizable authentication and access-control framework. It is the de-facto standard for securing Spring-based applications.

Spring Security is a framework that focuses on providing both authentication and authorization to Java applications. Like all Spring projects, the real power of Spring Security is found in how easily it can be extended to meet custom requirements

Features:

- Comprehensive and extensible support for both Authentication and Authorization
- Protection against attacks like session fixation, clickjacking, cross site request forgery, etc
- Servlet API integration
- Optional integration with Spring Web MVC
- And a lot of other

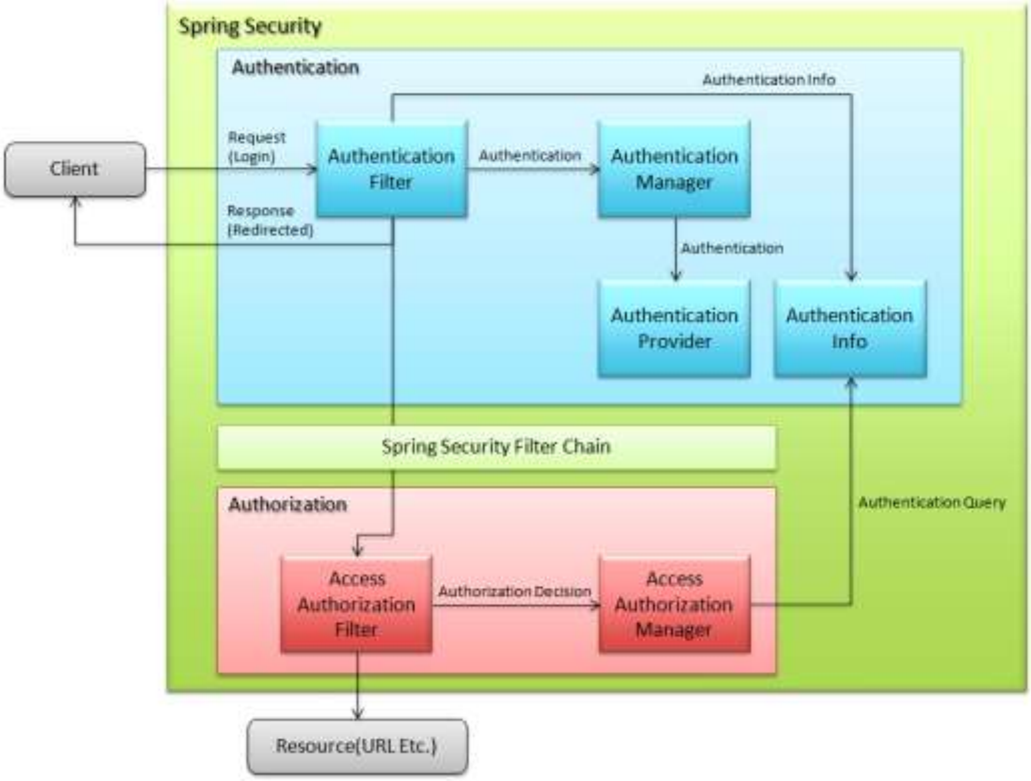
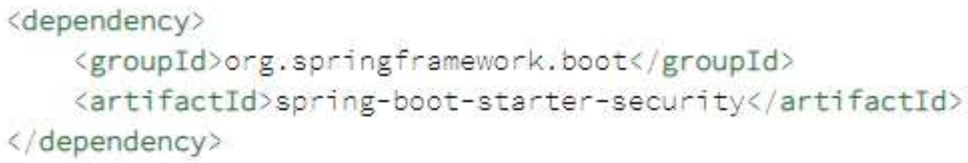


Fig.3.6 Spring Security Overview

Default Security Setup

In order to add security to our Spring Boot application, we need to add the security starter dependency:



```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

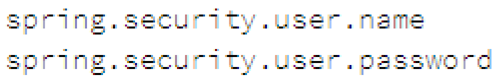
Fig.3.7 Maven Spring Security dependency

This will include the SecurityAutoConfiguration class – containing the initial/default security configuration.

Notice how we didn't specify the version here, with the assumption that the project is already using Boot as the parent.

Simply put, by default, the Authentication gets enabled for the Application. Also, content negotiation is used to determine if basic or formLogin should be used.

There are some predefined properties, such as:



```
spring.security.user.name
spring.security.user.password
```

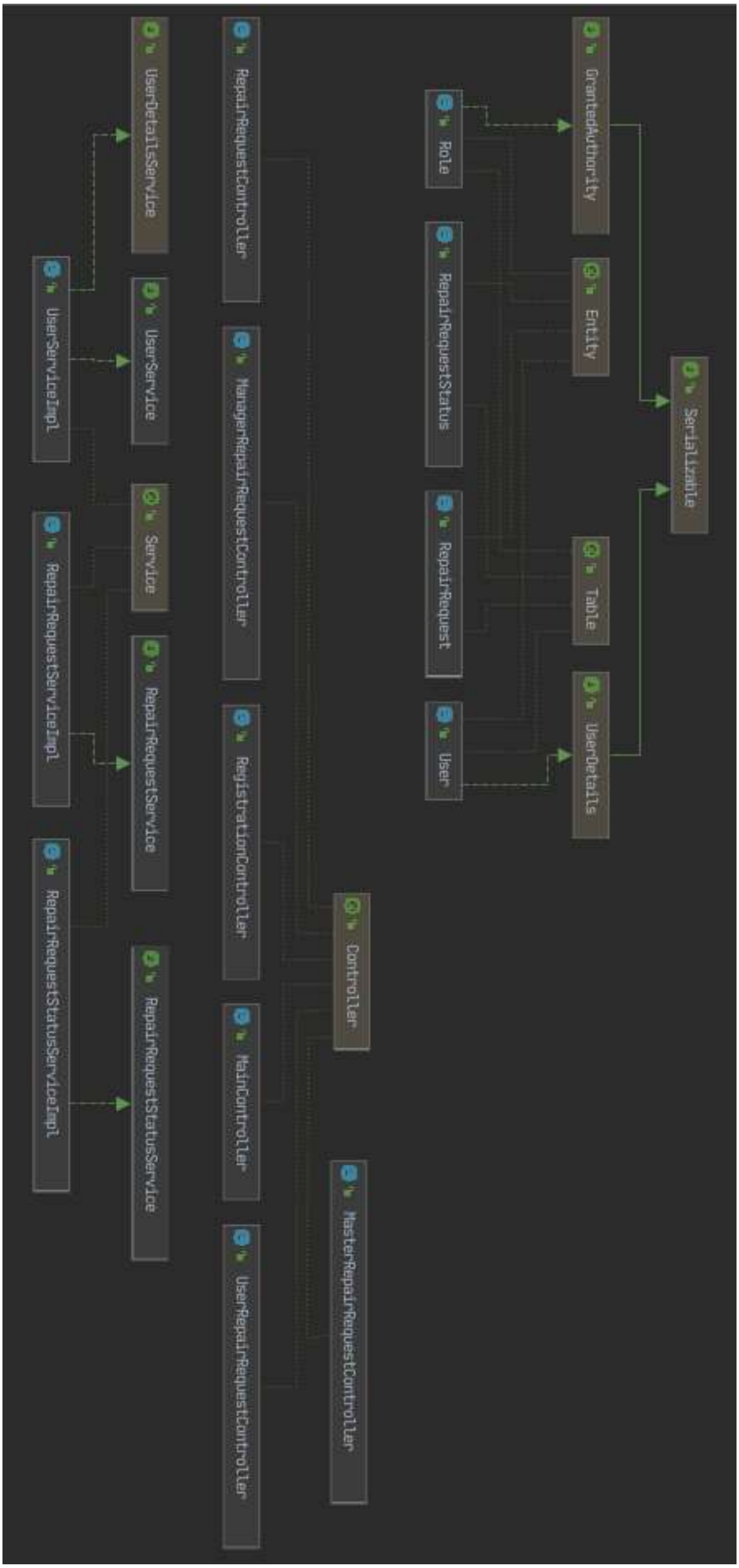
Fig.3.8 Predefined propeties

If we don't configure the password using the predefined property spring.security.user.password and start the application, we'll notice that a default password is randomly generated and printed in the console log:

Using default security password: c8be15de-4488-4490-9dc6-fab3f91435c6

3.10. Classes diagram

The class diagram shows the classes and their relationships, thus representing the logical aspect of the project. A separate class diagram is a certain perspective of the class structure. At the analysis stage, we use class diagrams to highlight the general roles and responsibilities of the entities that provide the required system behavior. At the design stage, we use a class diagram to convey the structure of the classes that make up the system architecture. The two main elements of a class diagram are classes and their basic relationships.



Pic.3.9 Classes diagram

Conclusion on the third part

In the third section of the diploma project it was determined that the system of monitoring the processes of the technical department of the Internet provider will be built using the architectural template MVC (model-representation-controller).

I also constructed a diagram of the components of a thesis project based on the MVC principle, which shows how the model provides data and responds to controller commands, view is responsible for displaying model data to the user, and the controller interprets user actions, notifying the model of changes.

To implement the future prototype of the system, a class diagram was constructed, which describes in detail the composition of the future system for monitoring the processes of the technical department of the Internet provider.

The deployment diagram showing the topology of the system and the distribution of its components by nodes, as well as the routes of information transfer between hardware nodes, was also presented and described in detail.

PART 4
SYSTEM PROTOTYPE

4.1. System prototype in general

Software prototyping is becoming an increasingly popular and frequently used process in IT companies. The reasons are as follows: on the one hand - this is a tribute to fashion, on the other - prototyping promises the company a number of significant advantages.

However, making the prototyping process useful and efficient is not an easy task. There are pitfalls, questions arise. Who and when should prototype? How to make prototypes? How to use them? The answers to these questions and the next steps determine the success and usefulness of the innovation. If they are incorrect - prototyping can be not only harmful but also extremely expensive.

Here we no longer use the prototype, but our customers. When accepting works, they now, in addition to, and sometimes instead of checking the functions of the vehicle, compare the implemented system with the prototype. This is beneficial to both parties. The customer has the right to make a claim if something is wrong with the implemented system as in the prototype. But we as a contractor can protect ourselves from claims, if we implement the system in the same way as in the prototype. The customer simply will have no reason to be dissatisfied. Thus, the executor gives, and the customer receives exactly what was agreed - no more, no less. Nobody does extra work, and everyone is happy.

Further in the section the prototype of the developed product of system of repair agency will be described.

4.2. Implementation of a system for a repair agency

1. Realization of default ApplicationRunner class.

```
@SpringBootApplication
public class ApplicationRunner {
    public static void main(String[] args) {
        SpringApplication.run(ApplicationRunner.class, args);
    }
}
```

Annotation SpringBootApplication indicates a configuration class that declares one or more @Bean methods and also triggers auto-configuration and component scanning. And also, “SpringApplication.run” is a default method which tells Spring Boot to start collect project together.

2. Realization of User entity class.

```
@Entity
@Table(name = "usr")
public class User implements UserDetails {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    private String username;
    private String password;
    private boolean active;

    @OneToMany(fetch = FetchType.EAGER, mappedBy = "repairRequestCreator")
    private Set<RepairRequest> statements;

    @ManyToMany(fetch = FetchType.EAGER)
    @JoinTable(name  =  "user_roles",  joinColumns  =  @JoinColumn(name  =  "user_id",
referencedColumnName = "id"),
            inverseJoinColumns = @JoinColumn(name = "role_id", referencedColumnName
= "id"))
    private Set<Role> roles = new HashSet<>();

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    @Override
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }

    @Override
    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public boolean isActive() {
        return active;
    }

    public void setActive(boolean active) {
        this.active = active;
    }

    public Set<RepairRequest> getStatements() {
        return statements;
    }

    public void setStatements(Set<RepairRequest> statements) {
        this.statements = statements;
    }

    public Set<Role> getRoles() {
        return roles;
    }

    public void setRoles(Set<Role> roles) {
        this.roles = roles;
    }

    @Override
    public Collection<? extends GrantedAuthority> getAuthorities() {
        return getRoles();
    }

    @Override
    public boolean isAccountNonExpired() {
        return true;
    }

    @Override
    public boolean isAccountNonLocked() {
        return true;
    }

    @Override
    public boolean isCredentialsNonExpired() {
        return true;
    }

    @Override
    public boolean isEnabled() {
        return isActive();
    }
}
```

This class describes the state and behavior of the user in the system. To do this, user has id, username, password, active fields.

There are also access methods for the get and set fields. They are used to adhere to the principle of encapsulation in OOP programming methodology.

Implemented constructors for easy assembly of objects in other classes.

Annotation @Entity tells the Hibernate to make auto-connection within database without manual work

Annotation @Table tells Hibernate that we want to make the separete table in database with name “usr”

Also, in this class there are fields like statements (user repair requests), roles(what authority user has in system). And annotations over this fields are also for Hibernate, to let it know which type of connections and what types of SQL Joins to use.

As you see, there is no business logic in this class, such as different methods for work with this entity, because of the using of MVC pattern.

3. Realization of Role entity class

```
@Entity
@Table(name = "roles")
public class Role implements GrantedAuthority {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    @Column(name = "role")
    private String role;

    @ManyToMany(mappedBy = "roles")
    private Set<User> users;
}
```

There are Role class fields which is necessary for our system and also some Hibernate configuration for it.

Also system uses another two Entity classes, RepairRequest and RepairRequestStatus with their own necesassary fields and annotations.

To make Hibernate work properly we need to add some configuration in main Spring Boot configuration file, which is called “application.properties”. This file stores in resources of the project

```
spring.thymeleaf.cache=false
server.error.whitelabel.enabled=false
server.port=8888
spring.datasource.url=jdbc:mysql://localhost:3306/final_project_spring
spring.datasource.username=root
spring.datasource.password=root
spring.jpa.generate-ddl=true
spring.jpa.hibernate.ddl-auto=update
```

This is how “application.properties” file look.

- “`server.error.whitelabel.enabled=false`” this raw disable default “404 page” of Spring Boot, since in this project another custom error page is used
- “`server.port=8888`” this raw indicates which server port will be used by the system
- “`spring.datasource.url=jdbc:mysql://localhost:3306/final_project_spring`” this one is an url for connection with MySQL local database
- Next two raws are just username and password for connection with DB
- “`spring.jpa.generate-ddl=true`” JPA has features for DDL generation, and these can be set up to run on startup against the database. This is controlled through this exteranal property
- “`spring.jpa.hibernate.ddl-auto=update`” this property has some standart values like none, validate, update and create-drop. “update” is used, due to necessity of updating the database with every next start of application

4. Description of WebSecurityConfig class

```
@Configuration
@EnableWebSecurity
@EnableGlobalMethodSecurity(prePostEnabled = true)
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {
    @Autowired
    UserServiceImpl userService;

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.csrf().disable()
                .authorizeRequests()
                .antMatchers("/", "/index", "/registration", "login").permitAll()
                .antMatchers("/request", "/userlist").hasAuthority("ROLE_USER")
                .antMatchers("/masterlist").hasAuthority("ROLE_MANAGER")
                .antMatchers("/masterlist").hasAuthority("ROLE_MASTER")
                .anyRequest().authenticated()
                .and()
                .formLogin()
                .loginPage("/login")
                .permitAll()
                .and()
                .logout()
                .permitAll();
    }

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {
        auth.userDetailsService(userService)
                .passwordEncoder(NoOpPasswordEncoder.getInstance());
    }
}
```

This is main class which provide security to the system. Spring Boot in pair with Spring Security gives an easy-editable tool to forbid different link acess to users with different roles. The system consists of different end-points and controllers, and everything developer need to do to customize it in the way it’s required.

To work with different request from users, managers and masters, system uses different controllers, due to MVC architecture which receives request from client side(browsers) and processed data and transmit it to services, which about will be written a little further. There are examples, of some controllers.

5. Realization of MainController class

```
@Controller
public class MainController {

    @RequestMapping(value = {"/", "/index"}, method = RequestMethod.GET)
    public String mainPage() {
        return "index";
    }
}
```

This class is an example of the simplest controller. All it does – is returns the user “index” endpoint, if urls like “/” or “/index” was used. The client will see the main page of the system.

Using of @Controller annotation tells the Spring Boot, that this class is Controller. And also over method there is @RequestMapping annotation which is one of the most common annotation used in Spring Web applications. This annotation maps HTTP requests to handler methods of MVC and REST controllers.

Also this annotation has two paraters, first one is “values”, where is necessary to send all path urls that should return the main page of the system. And the second parameter is “method”. Since there is no necessity to send any data, we use Get method.

6. Realization of RegistrationController class

```
     @Controller
     public class RegistrationController {

         private static final Logger LOGGER = 
Logger.getLogger(RegistrationController.class);
         @Autowired
         UserService userService;

         @GetMapping("/registration")
         public String registration() {
             return "registration";
         }

         @PostMapping("/registration")
         public String createUser(User user, RedirectAttributes redirectAttributes) {
             User createdUser = userService.create(user);
             if (createdUser == null) {
                 LOGGER.warn("Can't do registration. User with such username already 
exist");
                 redirectAttributes.addFlashAttribute("message", "User already exist. Try 
another username!");
                 redirectAttributes.addFlashAttribute("type", "danger fade show");
                 return "redirect:/registration";
             }
             LOGGER.info("New user was created. Username: " + user.getUsername());
             return "redirect:/login";
         }
     }
```

RegistrationController is more complex compared with MainController. Same as MainController class it should be indicated as @Controller but there is much more functionality. In this class an object of class “Loger” should be presented, since it’s necessary to log all requests which was send through this controller and if something went wrong to let developer know about weaknesses in system.

Next field is a Spring Bean, the object of “UserService” class which was injected here with the main feature of Spring Boot – Dependency Injection.

Due to Dependency Injection Spring Boot takes only necessary beans from Inversion of Control container whenever it is necessary. To mark the object as a “bean” to let Spring know where and how to take it, @Autowired annotation is used.

User get the registration page by Get method, in this controller used simpler annotation @GetMapping. And there is also the Post method, which will be called after user inputed all information such e-mail and password and click “Registration” button. After the previous one was clicked, user service tries to create new user and write information about him into database and return the result. And after some checks should be done like null reference check. If user service returned “null” it means that such user is already exist and an exception should be thrown and redirect to registration page must be done. But if everything was good. The user will be redirected to login page.

Also to make system more secured in data transition between repositories model and controllers. There were added some services. Service is the additional layer data transfer model.

7. Realization of RepairRequestStatusServiceImpl class

```
@Service
public class RepairRequestStatusServiceImpl implements RepairRequestStatusService {

    @Autowired
    RepairRequestStatusRepository repairRequestStatusRepository;

    @Override
    public RepairRequestStatus findByStatus(String status) {
        return repairRequestStatusRepository.findByStatus(status);
    }
}
```

This is example of the simplest service which was realized in the system. @Service annotation tells Spring Boot how to communicate with this class, and also this class implements “RepairRequestStatusService” interface.

In this class dependency injection principle is present. There is Repair Request Status Repository is injected, with the help of which the “findByStatus” method accesses the database and return the status of necesassary repair request.

8. Realization of UserServiceImpl class

```
@Service
public class UserServiceImpl implements UserDetailsService, UserService {

    @Autowired
    UserRepository userRepository;
    @Autowired
    RoleRepository roleRepository;


    @Override
    public User loadUserByUsername(String username) throws UsernameNotFoundException
{
        return userRepository.getByUsername(username);
    }

    public User findByUserDetails(User user) {
        return userRepository.getByUsername(user.getUsername());
    }

    @Override
    @Transactional
    public User create(User user) {
        if (userRepository.findByUsername(user.getUsername()).isPresent()) {
            return null;
        } else {
            user.setActive(true);
            user.getRoles().add(roleRepository.findByRole(Role.Names.ROLE_USER));
            return userRepository.save(user);
        }
    }
}
```

There is described all methods for work with user. This service also marked with @Service annotation. This service exists to provide easy access to database by using JPA. There are two simple methods just to return necessary user by some parameters.

But also there is one “create” method from CRUD. It has @Transactional annotation which is used when two requests to database is present in one method. Because of the possibility, when first request was received with success, but during second request something went wrong, the first request which have already could made some changes in database would be rolled back.

Also, localization feature has been implemented into project. It was done with the help of Resource Bundle. To do this one class “LocaleConfig” should be added into configuration package. And some property files should be added. These files separated by languages you want to implement in your project. These files should have same name by with adding of locale indicator with underscore.

To support Cyrillic letters, it’s necessary to use Unicode, like:

```
	#index page
	index.title=\u0420\u0435\u043c\u043e\u043d\u0442\u043d\u0435\u0020\u0430\u0433\u043
5\u043d\u0441\u0442\u0432\u043e
	index.service.presentation=\u0420\u0435\u043c\u043e\u043d\u0442\u043d\u0435\u0020\u
0410\u0433\u0435\u043d\u0441\u0442\u0432\u043e
	index.service.description=\u0422\u0443\u0442\u0020\u0432\u0438\u0020\u043c\u043e\u0
436\u0435\u0442\u0435\u0020\u0437\u0430\u043b\u0438\u0448\u0438\u0442\u0438\u0020\u0437\u
0430\u043f\u0438\u0442\u0020\u043d\u0430\u0020\u0440\u0435\u043c\u043e\u043d\u0442\u0020\
u0432\u0430\u0448\u043e\u0433\u043e\u0020\u043e\u0431\u043b\u0430\u0434\u043d\u0430\u043d
\u043d\u044f

	#login page
	login.title=\u041b\u043e\u0433\u0443\u0432\u0430\u043d\u043d\u044f
	login.login=\u041b\u043e\u0433\u0456\u043d
	login.button.login=\u0423\u0432\u0456\u0439\u0442\u0438
	login.button.registration=\u0417\u0430\u0440\u0435\u0454\u0441\u0442\u0440\u0443\u0
432\u0430\u0442\u0438\u0441\u044c
	login.link.to.registration=\u041d\u0435\u043c\u0430\u0454\u0020\u0430\u043a\u0430\u
0443\u043d\u0442\u0443\u003f
```

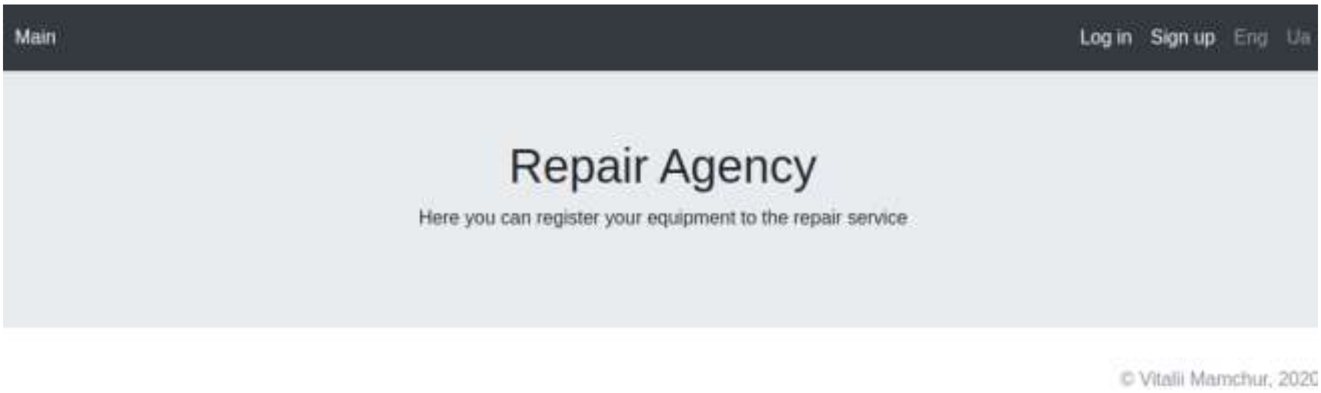
And this is how it looks with latins letters:

```
	#index page
	index.title=Repair agency
	index.service.presentation=Repair Agency
	index.service.description=Here you can register your equipment to the repair 
service

	#login page
	login.title=Login
	login.login=Log In
	login.button.login=Log in
	login.button.registration=Sign Up
	login.link.to.registration=Do not have an account?
```

4.3. Repair agency apperarance

1. View of the main page



Main

Log in Sign up Eng Ua

Repair Agency

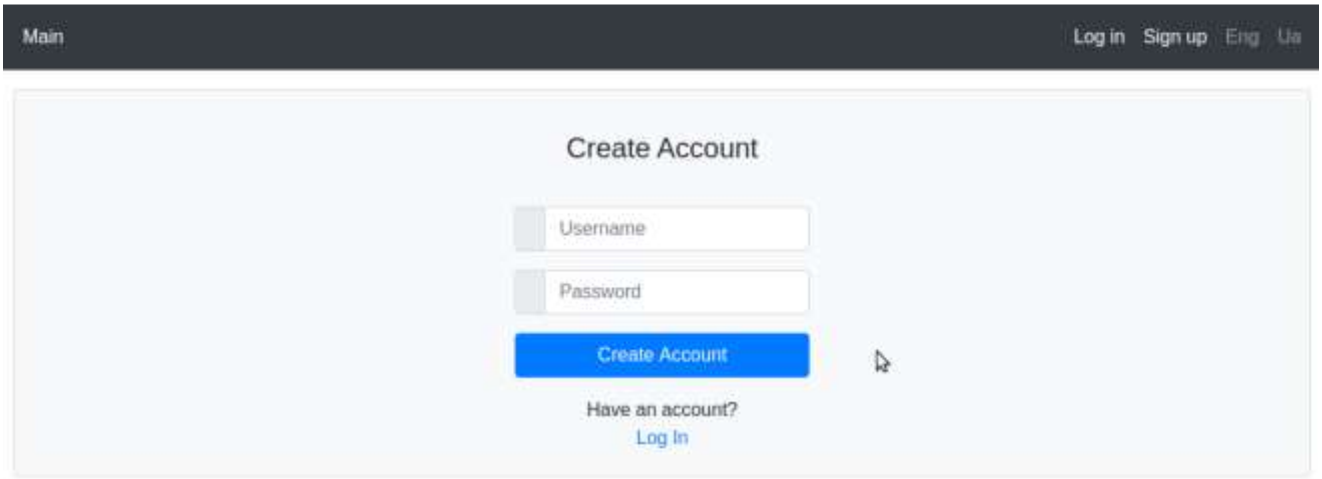
Here you can register your equipment to the repair service

© Vitalii Mamchur, 2020

Fig.4.1 Main page of system

There is the possibility to re-render main page of system, log in into system or sign up and also change the language of system.

2. View of registration page



Main

Log in Sign up Eng Ua

Create Account

Username

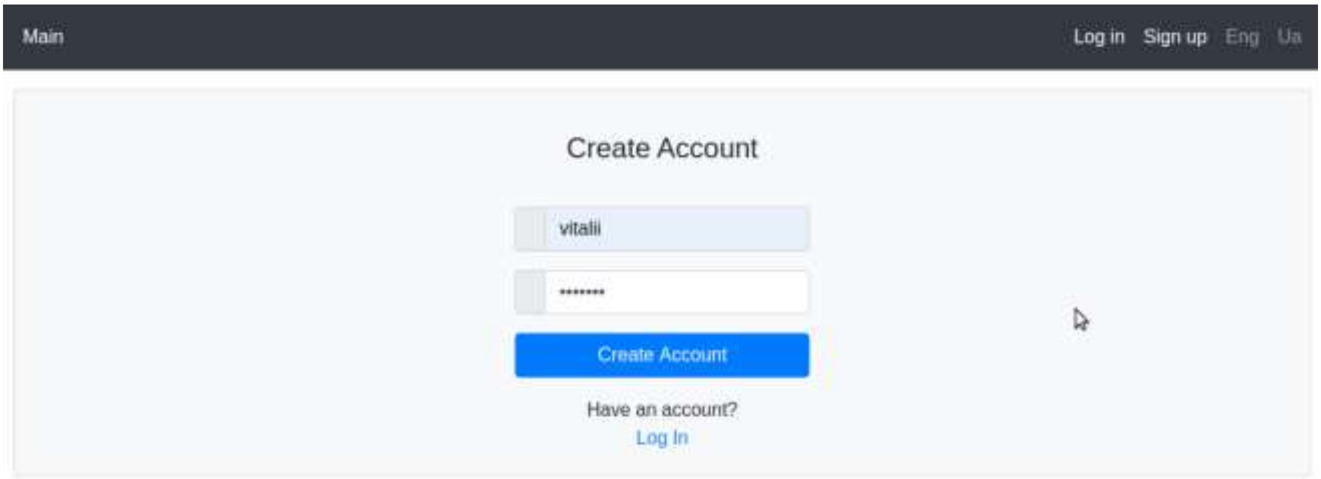
Password

Create Account

Have an account?
Log In

Fig.4.2 Registration page

On this page, the user can sign up in system with unic username and password which will be encrypted.

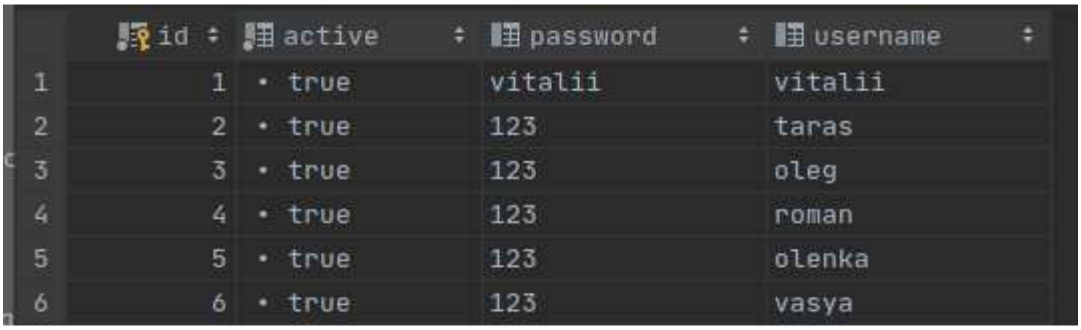


vitalii

****

Fig.4.3 Registration page with registration data

If there is no such user in database and all registration data is good and data base request was returned with success, the user will be added into the system’s database



	id	active	password	username
1	1	• true	vitalii	vitalii
2	2	• true	123	taras
3	3	• true	123	oleg
4	4	• true	123	roman
5	5	• true	123	olenka
6	6	• true	123	vasya

Fig.4.4 Table “usr” in database

3. View of Log in page

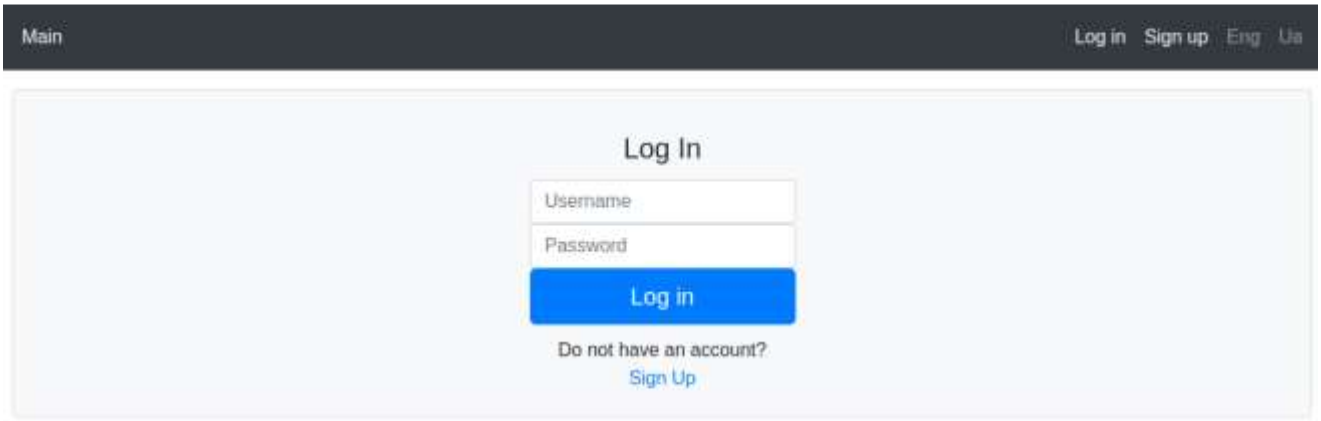
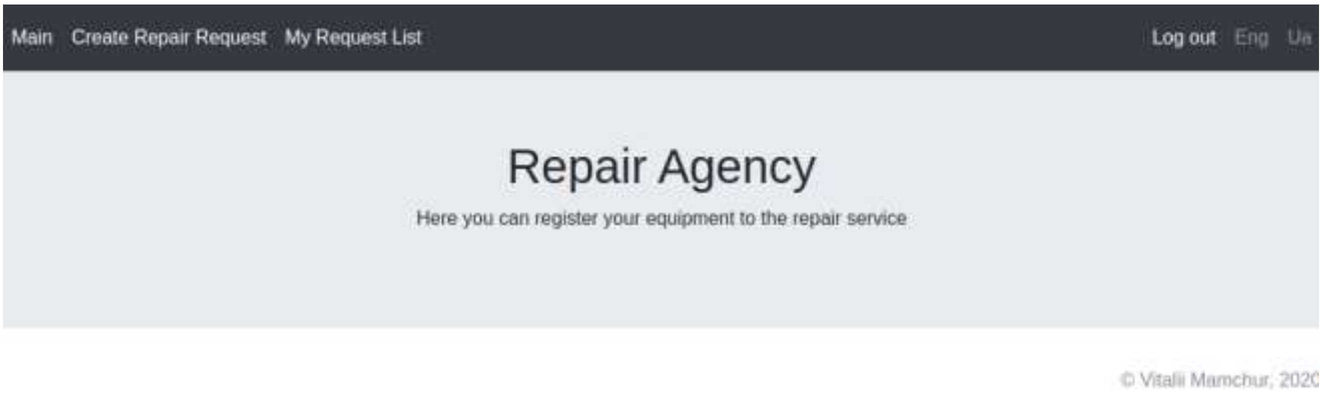


Fig.4.5 Log in page

There the client can log in into system, if he has been registrated earlier, or go to Sign up page, if user do not have an account yet.

4. View of user main page



Repair Agency

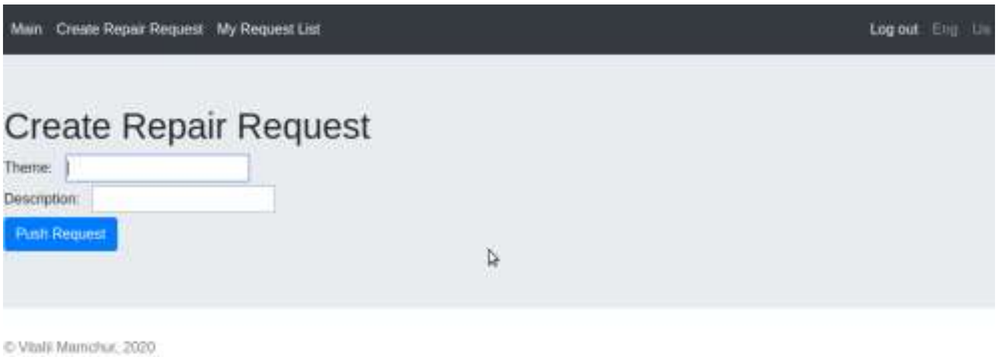
Here you can register your equipment to the repair service

© Vitalii Mamchur, 2020

Fig.4.6 Main page for logged in user

The main page, which logged in user will see. There is some additional links added, like “Create Repair Request” page and “MyRequestList”. Also, user has opportunity to log out any time.

5. View of Create Repair Request page for user



Create Repair Request

Theme:

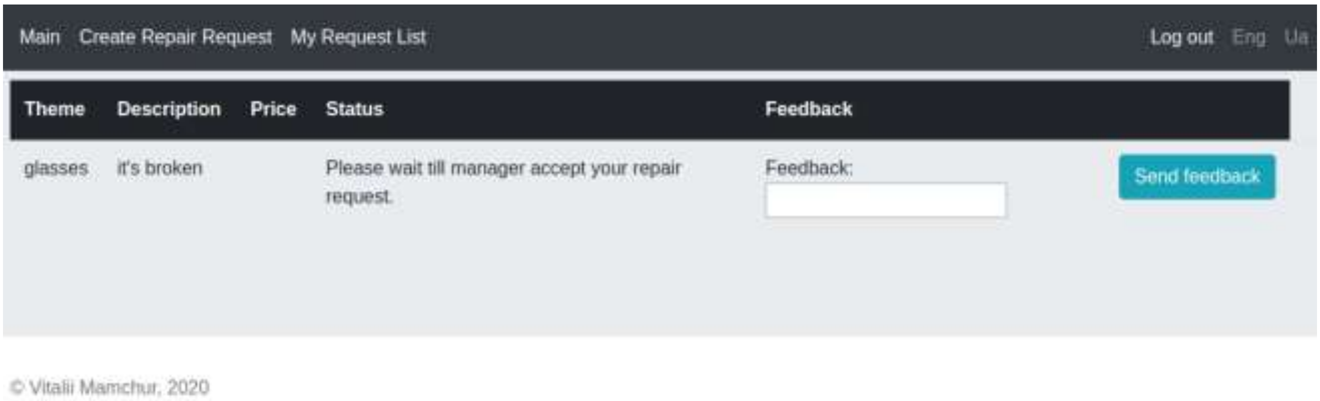
Description:

© Vitalii Mamchur, 2020

Fig.4.7 “Create Repair Request” page for logged in user

On this page, user may create a repair request for any staff he has. The client just need to write the object of repair and description what happened.

6. View of My Request List for user

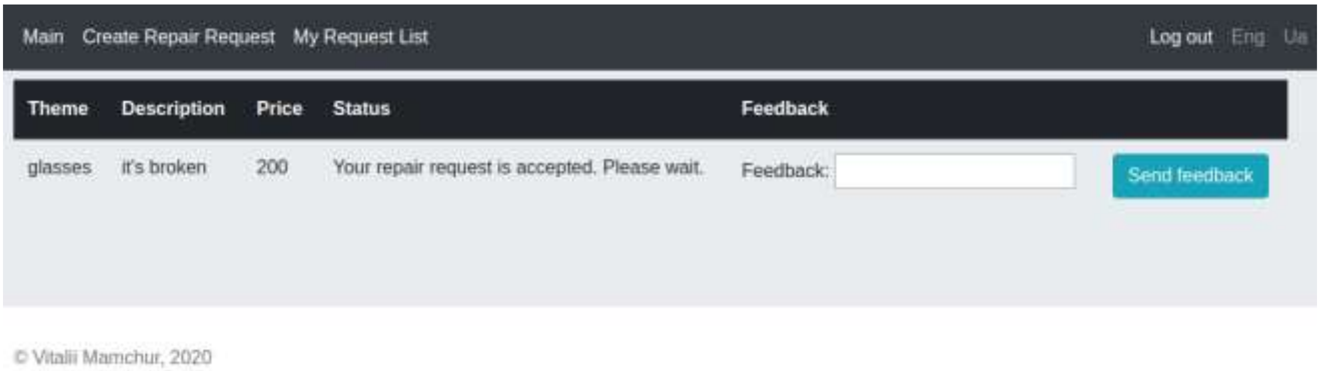


Theme	Description	Price	Status	Feedback
glasses	it's broken		Please wait till manager accept your repair request.	Feedback:

© Vitalii Mamchur, 2020

Fig.4.8 List of the client requests

Here, the client can see all repair requests he requested in this system. Check their status, wait till manager put the price and send feedback after the work was done

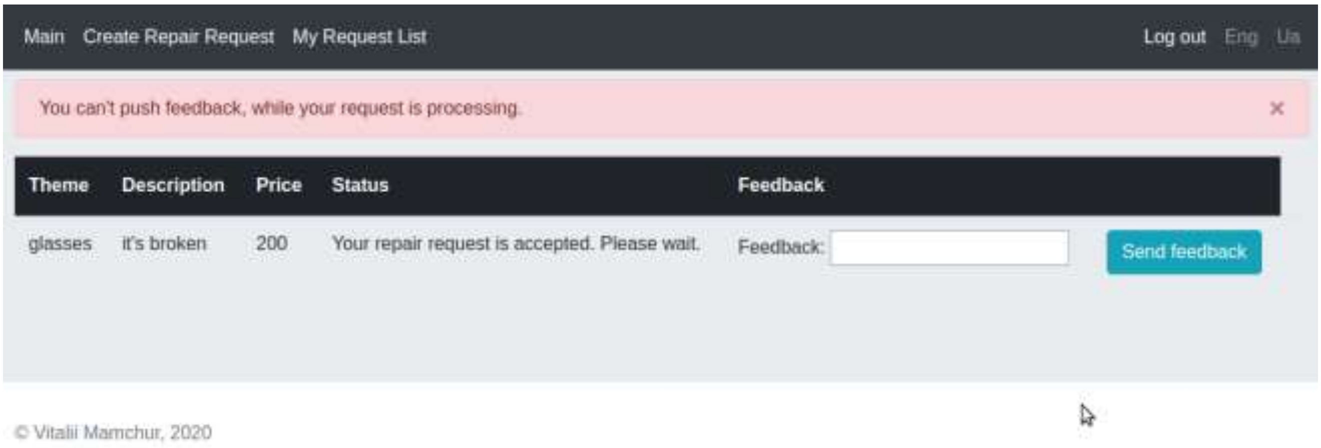


Theme	Description	Price	Status	Feedback
glasses	it's broken	200	Your repair request is accepted. Please wait.	Feedback:

© Vitalii Mamchur, 2020

Fig.4.9 Changed client's list of request with price and other status

This is updated list, after the manager put the price on the repair and status had been changed.



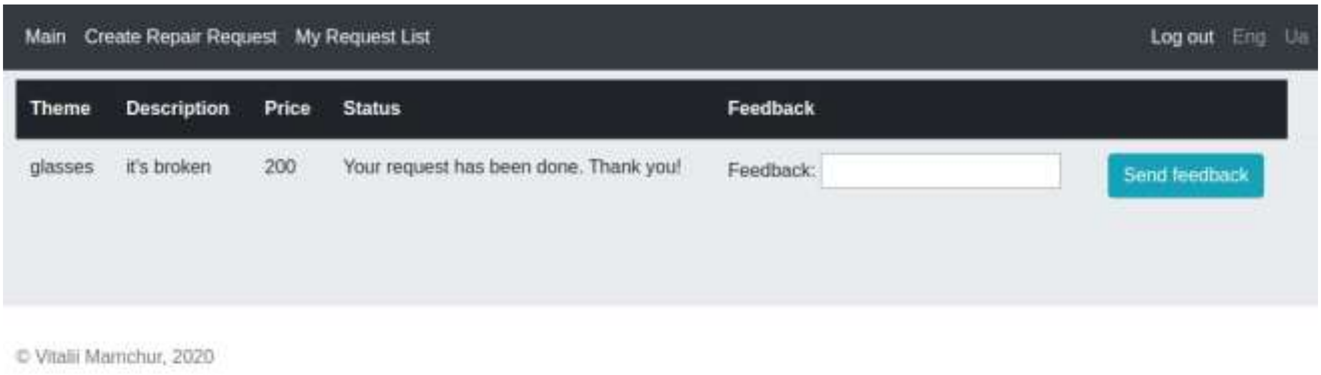
You can't push feedback, while your request is processing.

Theme	Description	Price	Status	Feedback
glasses	it's broken	200	Your repair request is accepted. Please wait.	Feedback:

© Vitalii Mamchur, 2020

Fig.4.10 Warning popup due to feedback

And also, if user try to send feedback till the repair request would be closed, he will see an exception about it.

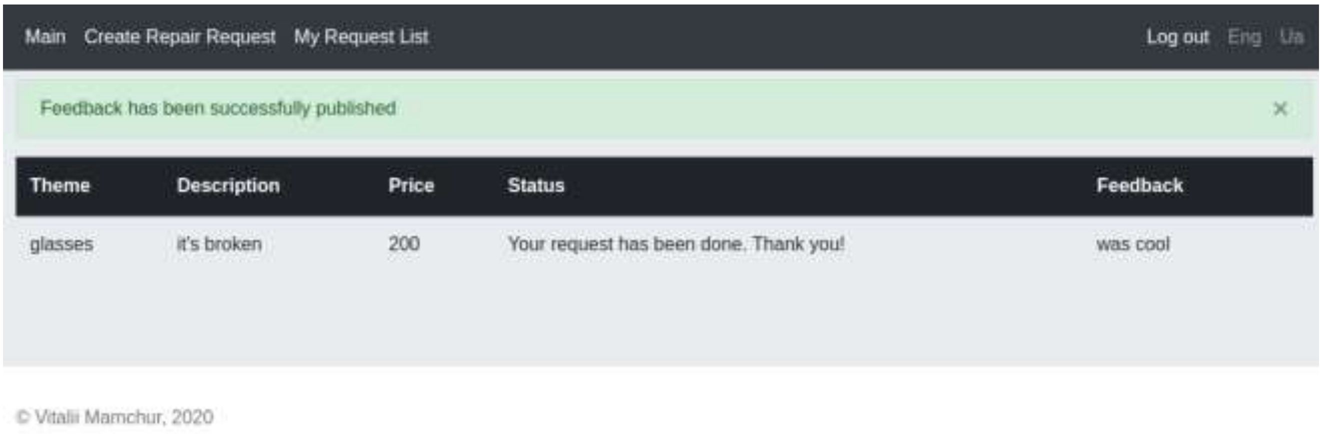


Theme	Description	Price	Status	Feedback
glasses	it's broken	200	Your request has been done. Thank you!	Feedback:

© Vitalii Mamchur, 2020

Fig.4.11 Changed status of request

This case is when the repair request has been done and closed by some master, status has been updated and user is able to leave feedback about it.



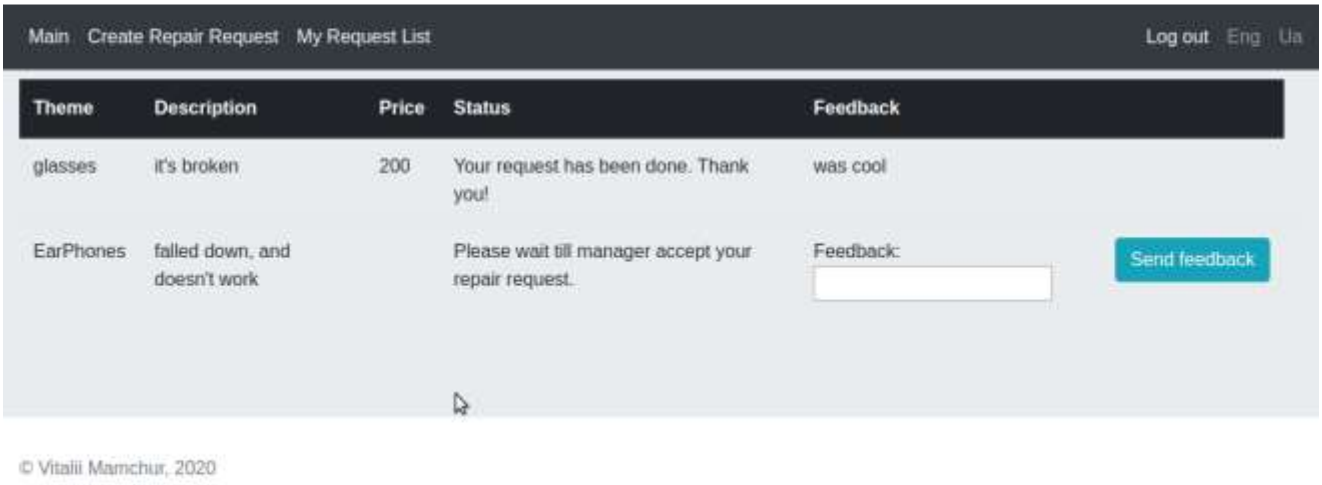
Feedback has been successfully published

Theme	Description	Price	Status	Feedback
glasses	it's broken	200	Your request has been done. Thank you!	was cool

© Vitalii Mamchur, 2020

Fig.4.12 Feedback added to request

After the client leave the feedback, he will see success notification about it, and feedback field will be updated.



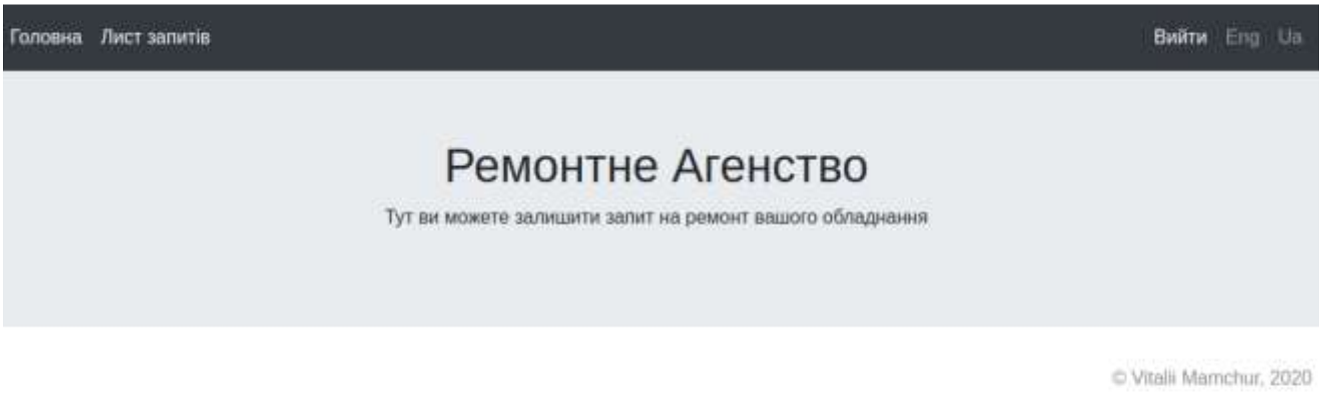
Theme	Description	Price	Status	Feedback
glasses	it's broken	200	Your request has been done. Thank you!	was cool
EarPhones	falled down, and doesn't work		Please wait till manager accept your repair request.	Feedback:

© Vitalii Mamchur, 2020

Fig.4.13 Second repair request added

This list could be replenished, when user create new repair request.

7. View of main page for manager



Ремонтне Агенство

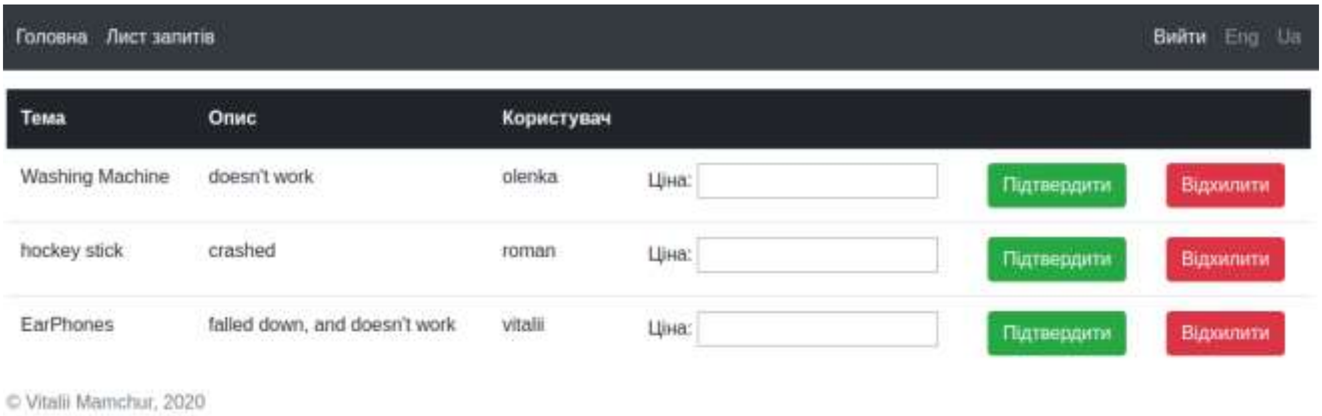
Тут ви можете залишити запит на ремонт вашого обладнання

© Vitalii Mamchur, 2020

Fig.4.14 Main page for manager

This is main page for manager translated to Ukrainian language. Here manager can see Repair Requests List from all users. Has possibility to log out and to change language.

8. View of Repair Request List for manager page

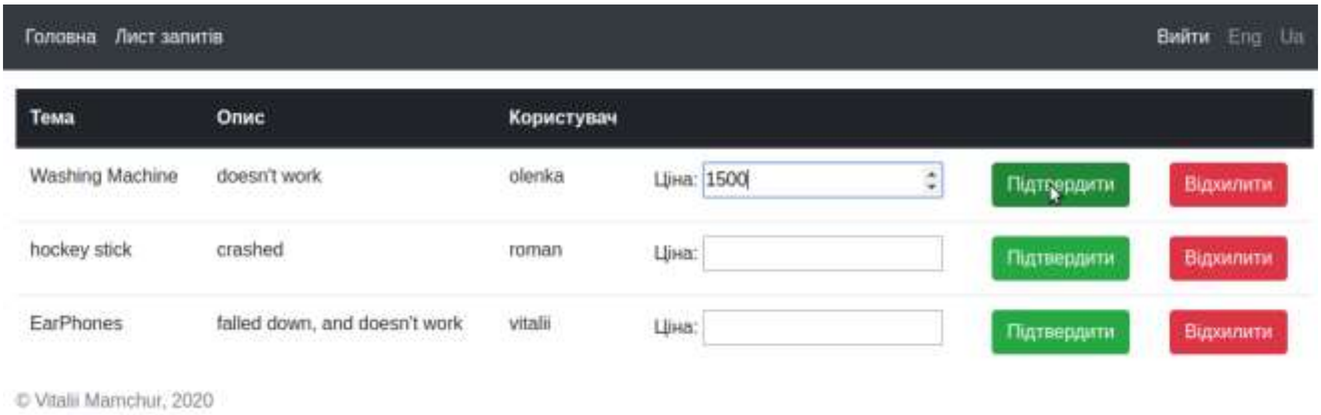


Тема	Опис	Користувач			
Washing Machine	doesn't work	olenka	Ціна:	Підтвердити	Відхилити
hockey stick	crashed	roman	Ціна:	Підтвердити	Відхилити
EarPhones	falled down, and doesn't work	vitalii	Ціна:	Підтвердити	Відхилити

© Vitalii Mamchur, 2020

Fig.4.15 List of repair requests from all clients

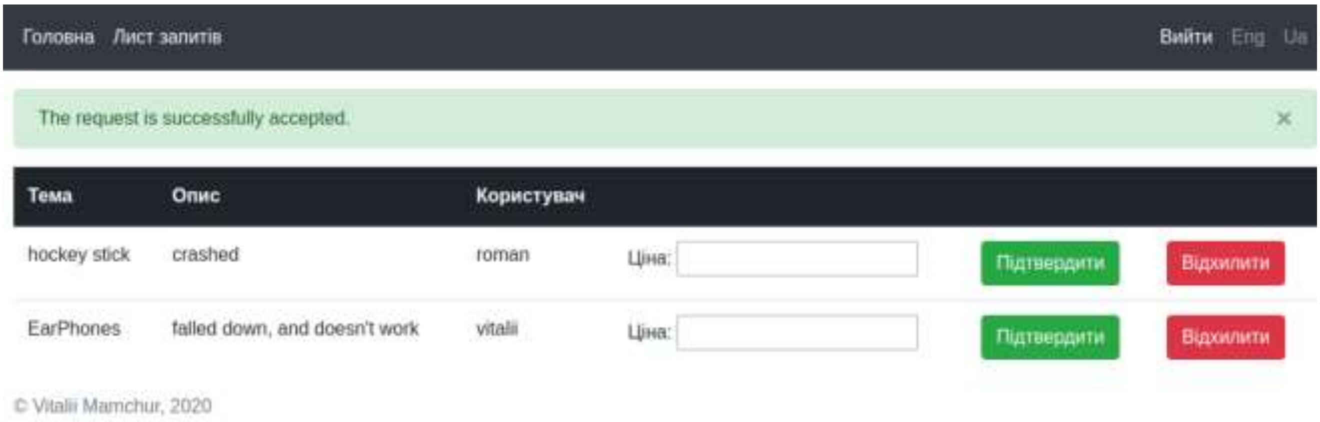
Manager can see all requests from all users that have been received. Manager have to analize the object of repair and what happened with it and put the price if he wants to accept it.



Тема	Опис	Користувач			
Washing Machine	doesn't work	olenka	Ціна: 1500	Підтвердити	Відхилити
hockey stick	crashed	roman	Ціна:	Підтвердити	Відхилити
EarPhones	falled down, and doesn't work	vitalii	Ціна:	Підтвердити	Відхилити

© Vitalii Mamchur, 2020

Fig.4.16 Repair request with inputed price



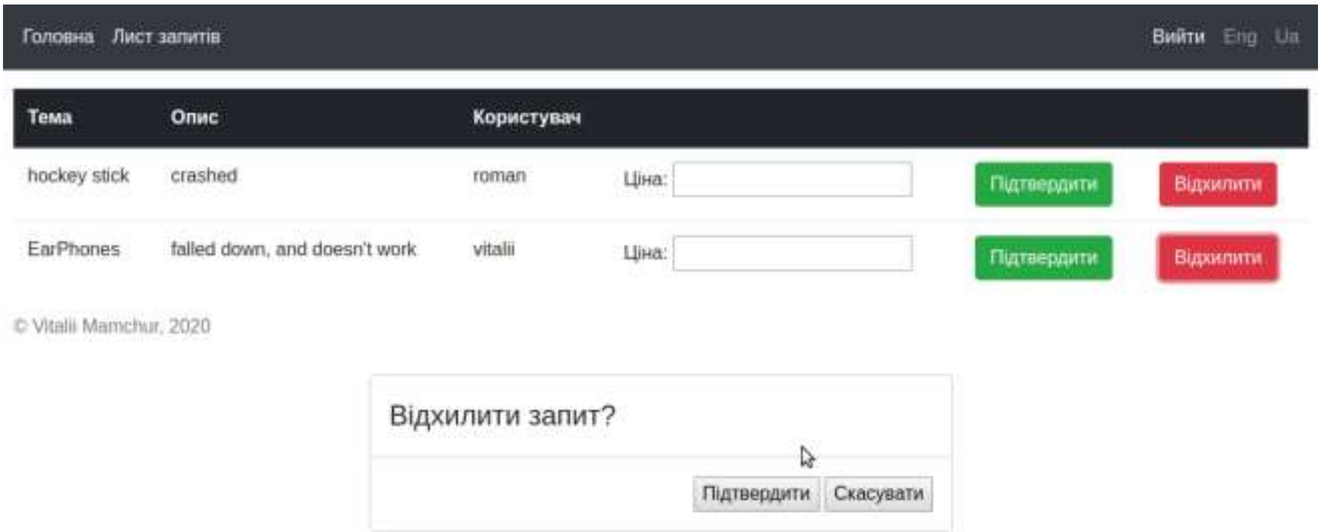
The request is successfully accepted.

Тема	Опис	Користувач			
hockey stick	crashed	roman	Ціна:	Підтвердити	Відхилити
EarPhones	falled down, and doesn't work	vitalii	Ціна:	Підтвердити	Відхилити

© Vitalii Mamchur, 2020

Fig.4.17 Succesfuly accepted repair request

Acceptance of the client request. On the side of client, the price was added to his request and status has been changed.



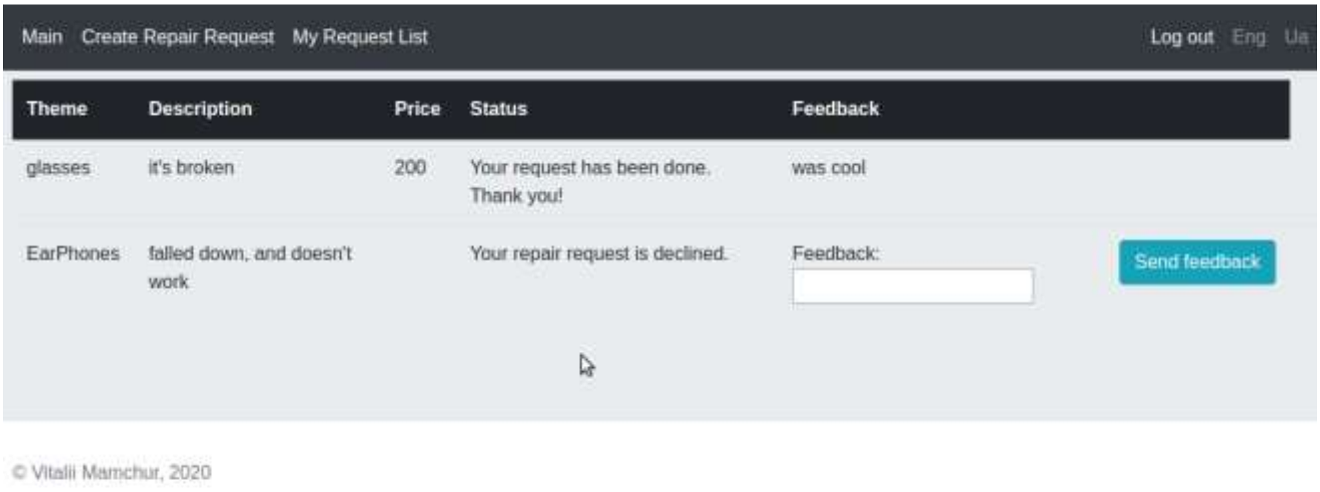
Тема	Опис	Користувач			
hockey stick	crashed	roman	Ціна:	Підтвердити	Відхилити
EarPhones	falled down, and doesn't work	vitalii	Ціна:	Підтвердити	Відхилити

© Vitalii Mamchur, 2020

Відхилити запит?

Fig.4.18 Decline confirmation window

And also, manager is able to decline the repair request. Before it he will see a confirmation window to approve his decision to prevent fake click on decline button.



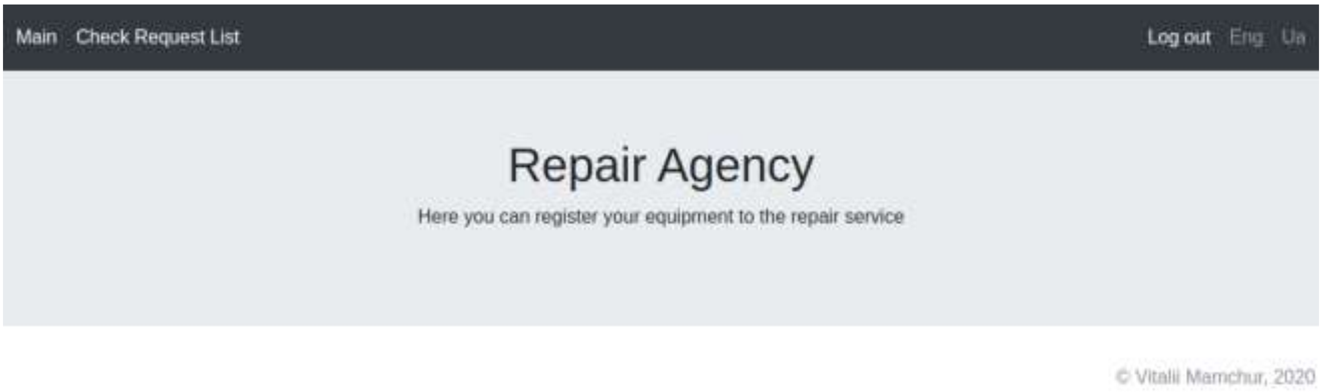
Theme	Description	Price	Status	Feedback	
glasses	it's broken	200	Your request has been done. Thank you!	was cool	
EarPhones	falled down, and doesn't work		Your repair request is declined.	Feedback:	Send feedback

© Vitalii Mamchur, 2020

Fig.4.19 Declined request status

This is what the client see if his request was declined. And also, it is mean that this request is closed and will not more be processed by the system.

9. View of main page for master



Repair Agency

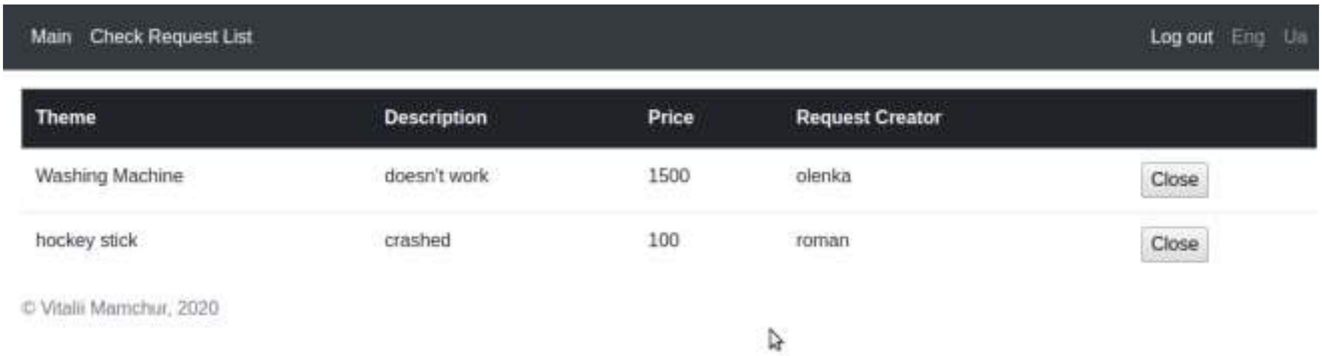
Here you can register your equipment to the repair service

© Vitalii Mamchur, 2020

Fig.4.20 Main page for master

Master should see when he is logged in, only link on Check Request List, and should have the possibility to log out and change the language.

10. View of Check Requests List for master

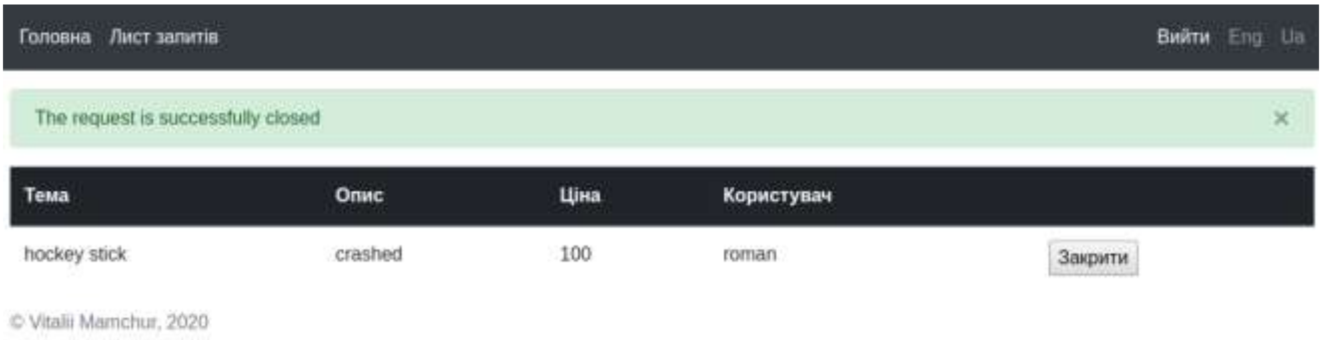


Theme	Description	Price	Request Creator	
Washing Machine	doesn't work	1500	olenka	Close
hockey stick	crashed	100	roman	Close

© Vitalii Mamchur, 2020

Fig.4.21 List of accepted repair requests for master

There the master can see all already accepted requests by manager and decide which he want and can to do. After he finish his work he can close the request in the system.



The request is successfully closed

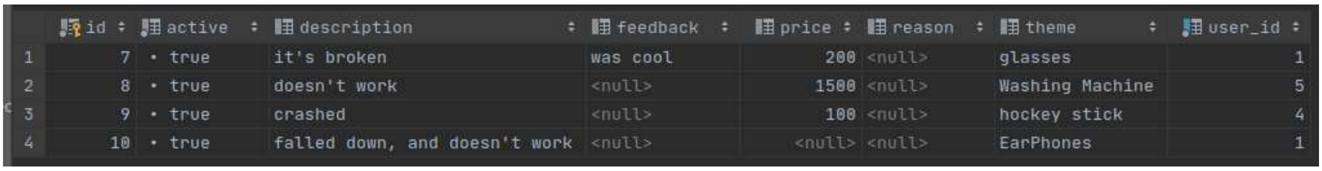
Тема	Опис	Ціна	Користувач	
hockey stick	crashed	100	roman	Закрити

© Vitalii Mamchur, 2020

Fig.4.22 Sucessfuly closed requst

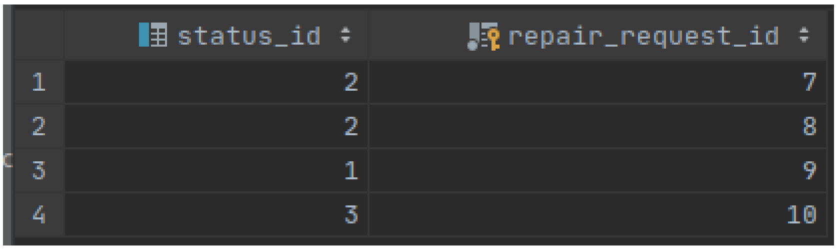
After master close the request, he will see the success popup. The status of request will be changed in user’s requests list and he will be able to leave the feedback.

11. Database tables view in project



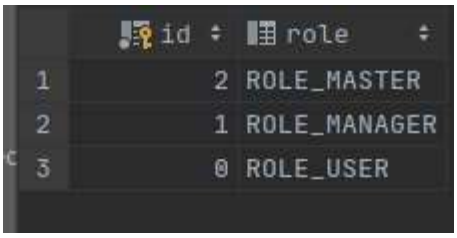
	id	active	description	feedback	price	reason	theme	user_id
1	7	• true	it's broken	was cool	200	<null>	glasses	1
2	8	• true	doesn't work	<null>	1500	<null>	Washing Machine	5
3	9	• true	crashed	<null>	100	<null>	hockey stick	4
4	10	• true	falled down, and doesn't work	<null>	<null>	<null>	EarPhones	1

Fig.4.23 Repair requests table



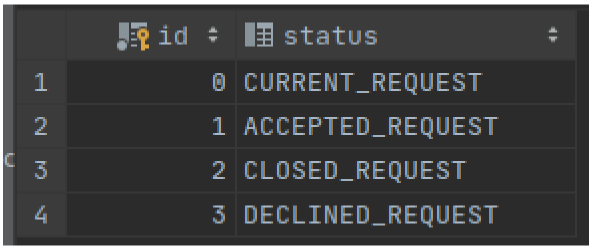
	status_id	repair_request_id
1	2	7
2	2	8
3	1	9
4	3	10

Fig.4.24 Connection of repair request with status by id



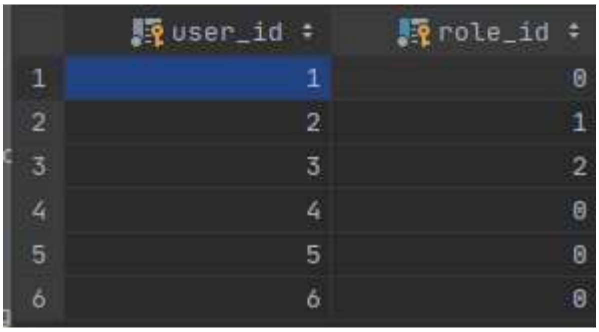
	id	role
1	2	ROLE_MASTER
2	1	ROLE_MANAGER
3	0	ROLE_USER

Fig.4.25 Roles which is used in system



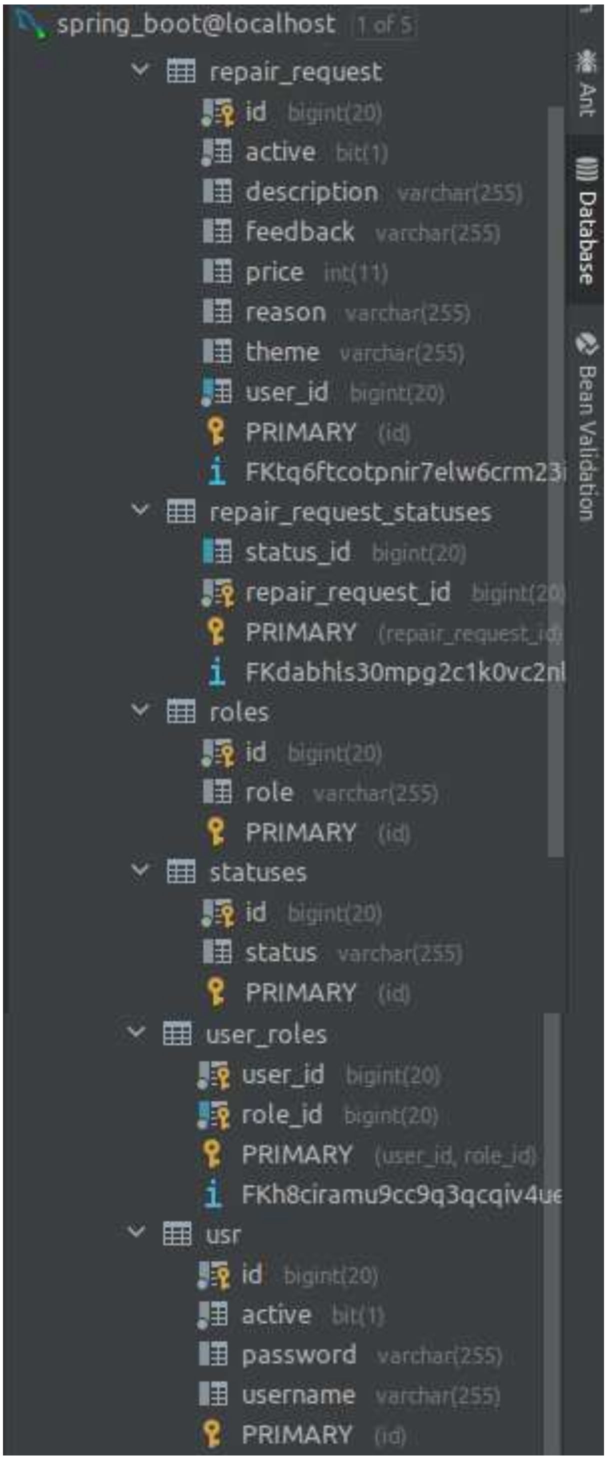
	id	status
1	0	CURRENT_REQUEST
2	1	ACCEPTED_REQUEST
3	2	CLOSED_REQUEST
4	3	DECLINED_REQUEST

Fig.4.26 Repair request statuses with id



	user_id	role_id
1	1	0
2	2	1
3	3	2
4	4	0
5	5	0
6	6	0

Fig.4.27 Connection of users with roles



```
spring_boot@localhost  1 of 5
    repair_request
        id  bigint(20)
        active  bit(1)
        description  varchar(255)
        feedback  varchar(255)
        price  int(11)
        reason  varchar(255)
        theme  varchar(255)
        user_id  bigint(20)
        PRIMARY  (id)
        FKtq6ftcotpnir7elw6crm23
    repair_request_statuses
        status_id  bigint(20)
        repair_request_id  bigint(20)
        PRIMARY  (repair_request_id)
        FKdabhls30mpg2c1k0vc2nl
    roles
        id  bigint(20)
        role  varchar(255)
        PRIMARY  (id)
    statuses
        id  bigint(20)
        status  varchar(255)
        PRIMARY  (id)
    user_roles
        user_id  bigint(20)
        role_id  bigint(20)
        PRIMARY  (user_id, role_id)
        FKh8ciramu9cc9q3qcqiv4ue
    usr
        id  bigint(20)
        active  bit(1)
        password  varchar(255)
        username  varchar(255)
        PRIMARY  (id)
```

Fig.4.28 Full structure of MySQL database, created by Hibernate

CONCLUSIONS

During the diploma project work, an analysis of the subject area of the repair agency system was conducted. The reasons for the emergence of software automation systems for repair agencies are analyzed, as well as the advantages of data data automation.

Next, the functional and non-functional requirements for the software were clearly defined. The very concept of requirement is given and defined.

All technologies, necessary to create repair agency automation system were investigated and described.

The structured software uses Spring Boot technology, which helps to build a repair agency system safe and reliable.

The diagram showed for which cycles the whole web service works.

The final stage was the development and implementation of a system prototype.

REFERENCES

1. Vince Barnes Why Create A Website?/ Vince Barnes // Non-Technical Introduction. – 2009. – #1. – 1

2. Exploring Micro-Frameworks: Spring Boot [Web resource]. – Access mode: https://www.infoq.com/articles/microframeworks1-spring-boot/#:~:text=Spring%20Boot%20is%20a%20brand,need%20to%20define%20boilerplate%20configuration.

3. Pete Zaborszky The 5 Best Ways to Build a Website in 2014 [Web resource]. – Access mode: http://www.make-a-web-site.com/5-best-ways-build-website-2014/

4. Joe Burns To Use or Not to Use a Database? / Joe Burns // Database Basics. – 2009. – #1. – 4

5. Curtis Dicken Top 5 Databases for Web Developers / Curtis Dicken // Database/SQL Primer. – 2011. – 1

6. SQL Server [Web resource]. – Microsoft, 2014. – Access mode: http://www.microsoft.com/en-us/server-cloud/products/sql-server/

7. Introduction to the Oracle Database [Web resource]. – Oracle Corporation. – Access mode: http://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm

8. MySQL 5.7 Reference Manual [Web resource]. – Oracle Corporation, 2015. – Access mode: http://dev.mysql.com/doc/refman/5.7/en/

9. Spring Boot for begginers [Web resource]. – DocForge, 2010. – Access mode: https://o7planning.org/ru/11267/spring-boot-tutorial-for-beginners

10. MVC pattern [Web resource]. – Access mode: https://medium.com/@anshul.vyas380/mvc-pattern-3b5366e60ce4

11. W3School [Web resource]. – Access mode: https://www.w3schools.com/php/php_mysql_intro.asp

12. Apache Maven Project Documentation [Web resource]. – Access mode: https://maven.apache.org/what-is-maven.html

13. Hibernate overiview [Web resource]. – Access mode: https://www.tutorialspoint.com/hibernate/hibernate_overview.htm

14. Tutorials point “Simply Easy Learning” [Web resource]. – Access mode: https://www.tutorialspoint.com/maven/maven_overview.htm

15. Spring Security documentation [Web resource]. – Access mode: https://spring.io/projects/spring-security