Annals of Spiru Haret University.
Economic Series
Since 2000
ISSN:2393-1795 ISSN-L:2068-6900

Issue 4/2021

SMART HOUSE WEB APPLICATION: DESIGN AND IMPLEMENTATION USING JAVA EE, MVC FRAMEWORK AND ARDUINO MICROCONTROLLER

Strahinja ĐORĐEVIĆ[1], Svetlana JEVREMOVIĆ[1], Jovana TOŠIĆ[1], Nina STOJANOVIĆ[2]

[1] ***Information Technology School, 34 Cara Dušana Street, Zemun, Belgrade, 11080, Email: strahinja7518@its.edu.rs, svetlana.jevremovic@its.edu.rs, jovana.tosic@its.edu.rs***

[2] ***Faculty of Contemporary Education, Svetozara Miletića 12, Belgrade, 11 000, Serbia, Email: nina.stojanovic@fsu.edu.rs***

How to cite: ĐORĐEVIĆ, S., JEVREMOVIĆ, S., TOŠIĆ, J., & STOJANOVIĆ, N. (2021). "Smart house web application: design and implementation using Java EE, MVC framework and Arduino microcontroller." *Annals of Spiru Haret University. Economic Series, 21*(4), 251-266, doi: https://doi.org/10.26458/21413

Abstract

The automation of the environment as an outcome of hectic modern life has resulted in applications that would simplify and facilitate everyday life. This paper aimed to explore the possibilities of designing and implementing the web application "Smart House", which would make it easier for all users, especially those with limited or disabled mobility, to control the device in the house. In the initial part of the paper, the technologies used to develop the web application "Smart House" is theoretically processed. The research is illustrated with diagrams and tables. Documentation on application design and implementation is processed by Larman's method. At the same time, theoretical analysis of used technologies refers to the literature of authorities in the field of research of advanced concepts of Java, Java EE platform and Arduino microcontroller open-source development system, which allow modularity and ease of modification. The research examines cases of using the application when the actor is a user, and on the other hand, cases of using the application when the actor is an admin. Further, the system operations that need to be designed are observed. This is followed by the application *implementation process, in which testing is performed simultaneously, which is the last and final phase of software system development to facilitate troubleshooting.*

Keywords: *Internet of Things; web application; smart house; Java; Servlet; MVC; Arduino.*

JEL Classification: O14

Introduction

Due to the fast way of life, there is a growing need to automate the environment in the modern world. The smart house system can provide control of subsystems such as heating and air conditioning, lighting, video surveillance, alarm system, and electrical devices such as refrigerator, water heater, stove, washing machine and the like [H. T. Lin, 2013]. Additional services include a driving simulator that assesses the driver's ability; smart shutters where the windows have automated shutters that can be adjusted with a remote device; a smart mailbox that notifies tenants every time shipments arrive; a smart water leak detector where sensors in the kitchen or garage aim to detect leaking water and notify tenants; a smart stove that warns the tenant if it is left on, a smart floor where sensors monitor the movement of the occupants; a cognitive assistant who reminds the tenants of scheduled obligations with audio and visual signals or, e.g. taking medication, etc. [Helal, 2005]. Consequently, smart houses are increasingly present to make everyday life easier for people.

The appearance of the smart house system was a consequence of the desire of people to make their life as easy and comfortable as possible. Ten years ago, smart houses looked like an expensive futuristic project and Science fiction idea and, after that, the privilege of exclusively rich people. Today, smart house systems are available to the majority of the population due to reducing IT technologies cost.

The smart house is a group of electronic and electro-mechanical devices that allow the owner to remotely monitor and control the processes in the apartment.

Today, smart house systems are often integrated into the general home information network and connected to the user via the internet. Therefore, IoT (connected systems to the internet) is a critical component of smart houses and building automation in general.

In the book "Towards A New Architecture", Le Corbusier described the futuristic house as "a machine for living in" with numerous connected sensors and

Annals of Spiru Haret University.
Economic Series
Since 2000
ISSN:2393-1795 ISSN-L:2068-6900

computers. [10]. While Le Corbusier's attitude only applied to materials and construction, the development of technology, in the meantime, also encompasses service automation into modern homes.

The main functions of a smart house are:

●**Maintaining maximum user comfort** in everyday life (monitoring living space characteristics such as microclimate and lighting, as well as automating household chores, such as washing, cooking cleaning)

●**Avoiding security incidents** (burglary, uncontrolled gas leaks, etc.)

●**Efficient and economical use** of electricity and water.

Smart houses can improve energy savings, thermal comfort, health, and safety. Programmed applications achieve control of these systems: "e.g. if the room is empty, reduce the heating to minimum and turn off the light [Callaghan 12] More advanced smart house systems can learn and make self-program based on monitoring user habits, adapting to a person's behaviour.

One of the arguments for adopting such systems is that energy savings can be significant, [Davidsson 98] estimates up to 40% of energy consumption.

Smart house includes three groups of components:

●**Controller -** controls the system in automatic mode according to user settings. The system configuration allows one or more devices.

●**Sensors -** monitor the state of the environment in residential space. (temperature, humidity, movement of people). The sensors send the appropriate signals to the controller, which processes them and issues commands in order to adjust the parameters according to the user's order;

●**Executive devices** perform a specific job or action when they receive an order from the controller. These include air conditioners, ventilation units, sirens, sockets, switches and so on;

Optional equipment is routers, autonomous power supply for devices, network hubs and cables, for the well-coordinated operation of the entire system that functions as a local network. For that user to control the system outside the home, it must be connected to a global information network - the internet.

Short history

The idea of creating an intelligent home system appeared at the beginning of the 20th century at the same time as electrification and the appearance of the first models of household appliances (vacuum cleaners, washing machines, irons, toasters, refrigerators)

1898. Nikola Tesla invented the first remote control. In addition, the Serbian-American inventor, electrical and mechanical engineer and futurist, made dozens of discoveries in the production, transmission and application of electricity. He invented the first AC motor and developed AC production and transmission technology, without which the technology that drives SmartHome would not be possible.

The first control devices for smart houses were developed and implemented in the 50s and 60s of the last century. For example, the ECHO IV kitchen computer (1966 - 1967) was the first smart device that calculated shopping lists, controlled the Temperature in the home, and turned devices on and off.

Figure 1. Jim Sutherland sits at the ECHO IV computer. (Pittsburgh Post-Gazette, 1966)

Issue 4/2021

In the 1970s, the X10 home control standard, a relatively inexpensive automation system, was developed in the United States and became quite widespread. X10, a home automation platform, sends digital information via radio frequency waves to existing electrical installations. Early users of this technology could remotely control devices in their homes using a command console and modules.

The advent and widespread use of smartphones and tablets, which control home devices, has dramatically accelerated the development of smart house technology worldwide.

Today's smart houses - are more about safety and environmental life. Smart houses are sustainable. Further development of smart houses goes in the direction of developing technologies that learn about user habits.

Figure 2. Schematic diagram of an IoT-enabled smart house depicting the employment of smart sensing devices indifferent utilities (IoT for smart house)

Automation easily controls all devices, thus achieving energy savings and more accessible day-to-day activities. Furthermore, management can be realized using

Annals of Spiru Haret University.
Economic Series
Since 2000
ISSN:2393-1795 ISSN-L:2068-6900

applications that use the internet as a portable medium, fixed or mobile phone, remote control or dedicated devices, and the implementation methods are combined according to the needs and wishes of users [Radenković, 2017].

The paper deals with the research of designing and implementing the web application "Smart House", with the ultimate goal of achieving significant changes in everyday life and household management and making life easier for users who decide to implement this system. This web application makes the most significant contribution to users with difficulty or disability, allowing them to control the device from a mobile phone or computer. Also, for users who lead a modern lifestyle, automation and remote control of the devices offered by this system can alleviate or at least reduce stress and care at home. By analyzing the literature cited in the paper, the technologies used for the development of the web application "Smart House" were investigated: Java web application programming, advanced Java concepts, as well as the implementation of the server part of the application using Java EE (Java platform for enterprise application development) and controller applications via Arduino microcontroller (open source development system, used to manage other systems and modules).

1. Theoretical framework of research - defining applied technologies

Before analyzing the documentation on the design and implementation of the web application "Smart House", the technologies used during the development phases of the application were analyzed. Then, the paper focuses on the design and implementation of the web application "Smart House" using Java EE components organized on the MVC model and the Arduino platform. In the beginning, the Java EE platform and specific technologies are explained, which make it easier for developers to develop the application and, at the same time, ensure a high level of application security. Then, an MVC pattern is processed, representing an organization of individual parts of the application into components. Each component has a specific role in the application. After that, basic information about the Arduino microcontroller used to control the smart house device is given.

1.1 Java EE

The Java platform for the development of enterprise applications (Java EE) can be divided into four parts that follow the appropriate technologies and services [Vlajić, 2005]:

Annals of Spiru Haret University.
Economic Series
Since 2000
ISSN:2393-1795 ISSN-L:2068-6900

Issue 4/2021

1. Web technologies used in the development of the presentation level of JEE or stand-alone web application;
2. Enterprise JavaBeans (EJB) technologies used in the development of business logic JEE applications;
3. Java XML technologies for developing applications that process XML documents and implement web services and
4. JEE platform services using all of the listed technologies.

JEE applications consist of JEE components that are installed on various computers. Computers that run components have different roles in the system, divided into the client, application logic, and Enterprise Information System (EIS) levels.

1.1.1 Servlets

Servlets are a popular way to build interactive web applications. It is a server-side component that runs exclusively within a Java virtual machine. Servlets provide a sophisticated way to create a server-side following a standard JEE environment and using a highly portable Java programming language. The role of the servlet consists of receiving and reading detailed data sent by the client (data from the form); receiving and reading implicit data sent by the client (header requests); generating results; sending detailed data to the client (HTML), and sending implicit data to the client (status codes about the header response). Servlet containers are generally components of a web or application server, such as BEA WebLogic Application Server, IBM WebSphere, and Sun Java System Web Server. Servlets are not designed for specific protocols—http Java packages [Jevremović, 2016].

1.1.2 JSP

Java Server Pages - is a java technology for generating dynamic content based on XHTML. MVC uses this technology as the default for generating views, i.e. view part of the application. It contains a set of XHTML and web-oriented UI components that are easily embedded in JSP pages and facilitate user interface development. The basic types of dynamic elements of JSP technology are expressions, scriptlets, declarations, directives.

1.1.3 Java Beans

Enterprise beans are Java EE components that implement EJB (Enterprise JavaBeans) technology. They are written in the Java programming language and represent components that run on the server-side and which, as a rule, encapsulate

Annals of Spiru Haret University.
Economic Series
Since 2000
ISSN:2393-1795 ISSN-L:2068-6900

Issue 4/2021

the business logic of the application. Enterprise beans are executed within an EJB container that represents the executable environment of the application server [Ball et al., 2006]. Enterprise beans are portable components, so it is possible to put together different applications that will use existing EJB components (reusability). Using Enterprise beans significantly simplifies the development of complex, distributed applications and enables the creation of software components that can be reused when developing new applications.

1.2 MVC

The MVC model is a model architecture and consists of three components: Model, View, and Controller. A model is a component that contains the structure of a business system and its operations; that is, it contains data and data processing operations. The View component provides a user interface through which the user communicates with the system. It also sends the user reports obtained from the model. A controller is a component that is responsible for managing the execution of system operations. It accepts the request from the client, then calls the operation defined in the model and controls its execution.

1.3 Arduino

Arduino is an open-source development system, an electronic computing platform created in 2005 at Ivery University in Italy. The Arduino system is used to manage other systems and modules. The hardware is a simple Arduino board design that includes an Atmel AVR microcontroller and accompanying input-output components. Arduino programming is done in a programming language based on the Wiring language. In essence, it is a C ++ programming language with many facilitations and predefined functions for end-users or developers. The Integrated Development Environment IDE is based on Processing.

The Arduino IDE is written in Java and can be used on programming platforms such as Windows, Linux and Mac. A program written for the Arduino is called a sketch. The syntax of the Arduino IDE is similar to the syntax of the C ++ programming language.

2. Design and implementation of the web application "Smart House."

After researching adequate technologies, detailed documentation preparation followed, including the following phases: specification of requirements, analysis of requirements, design, implementation and testing of the web application "Smart

House". Documentation on the design and implementation of the application was processed by Larman's method.

2.1 Specification with requirements

To implement the smart house system, it is necessary to provide reasonable documentation and instructions for users and a microcontroller and the database and web application. The smart house includes lighting (ordinary and RGB), control of yard irrigation, control of blinds, monitoring of house temperature, and air humidity. In the first phase of preparation of documentation (cases of use based on verbal model), the behaviour of the system will be analyzed when one user and one admin are logged in who administers users, controls access rights and monitors events in the system through logs.

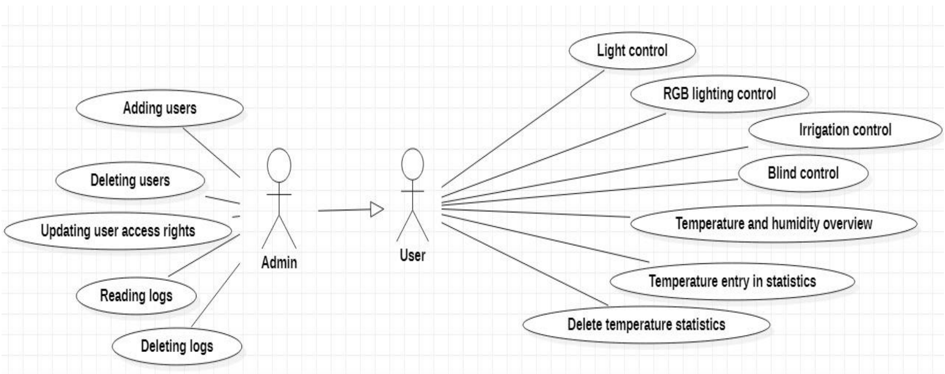

Figure 3. Use-case diagram of all use cases

2.1.1 Use cases

Based on the verbal model, the following use cases have been identified when the actor is a user: light control, RGB lighting control, irrigation control, shutter control, temperature and humidity review, temperature and humidity entry in statistics, and deletion of statistics.

Example - SK2: RGB lighting control

• Name: RGB lighting control

• Actors: User

- Participants: User and system
- Prerequisites: The system is switched on, the user is logged in, and the RGB lighting control form is displayed
- Baseline scenario:

1. The user enters the colour of the lighting (APUSO);
2. The user confirms the entered parameters and calls the system to make changes (APSO);
3. The system changes colour or lighting mode (SO) and
4. The system displays a colour change or mode (IA) success message.

- Alternative scenarios:

3.1. The system recognizes that the user has entered the same colour or mode, displays a message that no change (IA) has occurred, and

3.2 The system fails to change colour or mode; the user physically checks the lighting condition (ANSO).

Based on the verbal model, the following cases of use were observed when the actor was an admin: adding users, deleting users, reading logs, deleting logs, updating user access rights.

The description of the use cases when the actor is an admin also implies primary and alternative scenarios.

2.2 The analysis

The analysis phase is divided into:

1. System sequence diagrams for SK - user (DSSK1: Light control, DSSK2: RGB lighting control, DSSK3: Irrigation control, DSSK4: Blind control, DSSK5: Temperature and humidity overview, DSSK6: Temperature entry in statistics, and DSSK7: Delete temperature statistics) with the definition of the contract on system operations - user. Contracts are made for each of the observed system operations. Contracts describe the behaviour of a system operation by describing what the operation does, but not how. Thus, one contract is tied to one system operation.

The following system operations that need to be designed have been identified:

- control lights (bulbID, intensity)
- controlRGB (red, green, blue)
- controlIrrigation (percentage)
- check shutters (position)
- read Temperature and humidity ()
- enterStatistics (temperature)

Annals of Spiru Haret University.
Economic Series
Since 2000
ISSN:2393-1795 ISSN-L:2068-6900
ICCS Institutul Central de Cercetare Științifică
Fundația România de Mâine — Universitatea Spiru Haret — Știință, Cultură

Issue 4/2021

• outlineStatistics (statisticsID)

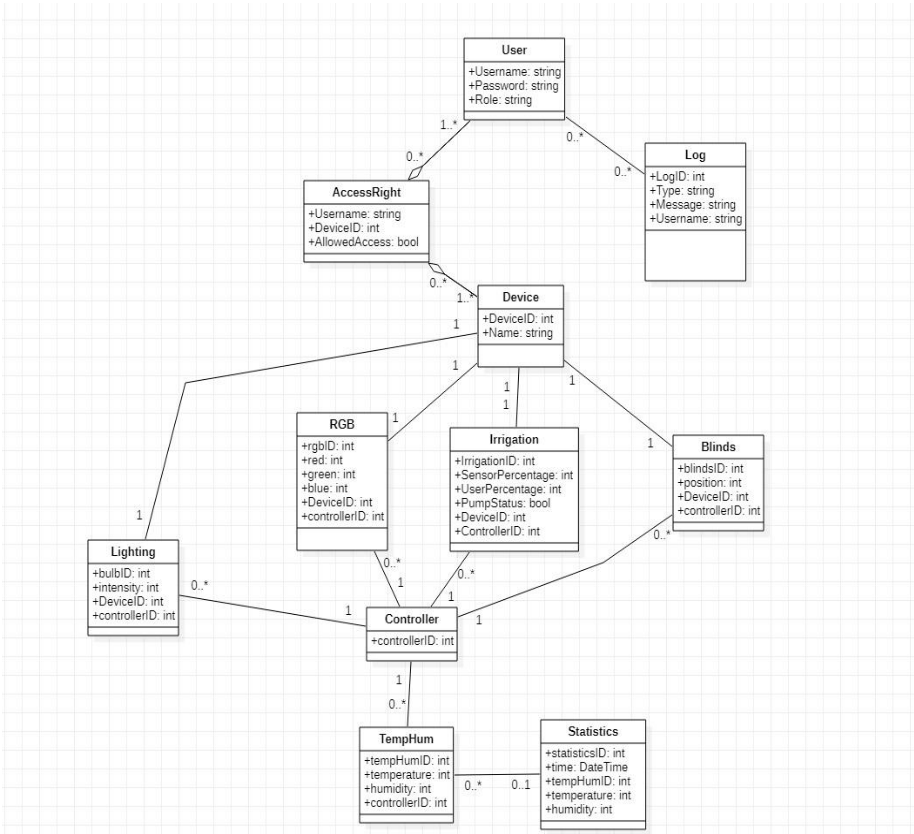

Figure 4. Conceptual model

2. System sequence diagrams for SK - admin (DSSK1: Adding users, DSSK2: Deleting users, DSSK3: Reading logs, DSSK4: Deleting logs, and DSSK5: Updating user access rights) with defining contracts for system operations - admin.

The system operations to be designed are:

• addUser (username, pass)
• user outlines (username)
• readLogs ()

• obrisiLog (logID)
• update Access Rights (user ID, device ID, right)
3. Conceptual (domain) model created based on data from the functional requirement and use cases.
4. A relational model can be made based on a conceptual model and will design a relational database.
• User (Username, Password, Role)
• Device (EditID, Name)
• Access Right (Username, Device ID, Allowed Access)
Usernamereferences to User (Username)
DeviceID refers to Device (DeviceID)
• Log (LogID, UserID, Type, Message,)
UserID refers to User (UserID)
• Controller (controllerID)
• Blinds (blindsID, controllerID, DeviceID position)
controllerID refers to Controller (controllerID)
DeviceID refers to Device (DeviceID)
• Irrigation (IrrigationID, ControllerID, DeviceID, SensorPercentage, UserPercentage, Pump Status)
controllerID refers to Controller (controllerID)
DeviceID refers to Device (DeviceID)
• RGB (rgbID, controllerID, DeviceID red, green, blue)
controllerID refers to Controller (controllerID)
DeviceID refers to Device (DeviceID)
• Lighting (bulb ID, controller ID, Device ID, intensity)
controllerID refers to Controller (controllerID)
DeviceID refers to Device (DeviceID)
• TempHum (tempHumID, controllerID, temperature, humidity)
controllerID refers to Controller (controllerID)
DeviceID refers to Device (DeviceID)
• Statistics (statisticsID, tempHumID, weather, temperature, humidity)
tempHumID refers to TempHum (tempHumID)

2.3 Designing a web application "Smart House"

The application is divided into two parts, the server part which is implemented using Java EE and the controller part implemented via Arduino. The communication between the controller and the server part is performed via the HTTP protocol, whereby the controller sends requests to the server part to which

Annals of Spiru Haret University.
Economic Series
Since 2000
ISSN:2393-1795 ISSN-L:2068-6900

Issue 4/2021

the server part responds with messages about the current state in the system, thus maintaining the state of the device set by the user.

The application design process includes creating Sequence Diagrams and collaboration diagrams for system operations - User, according to defined contracts, and then Sequence Diagrams and collaboration diagrams for system operations - Admin. After creating the diagram, the data warehouse is designed based on the software classes of the structure. As a result, tables of the relational database management system are obtained.

Special attention is paid to the design of the user interface - User and the user interface - Admin.

2.3.1 User Interface Layout - User (SK2: RGB Lighting Control)

To illustrate the appearance of the user interface, the case of use when the actor is the user - RGB lighting control was chosen.

Prerequisites: The system is turned on, the user is logged in, and the RGB lighting control form is displayed

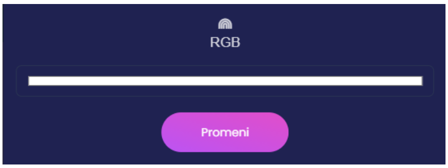

Figure 5. User interface display - RGB lighting control

Baseline scenario:

1. The user enters the colour or lighting mode
2.

Figure 6. User input display - RGB lighting control

Annals of Spiru Haret University.
Economic Series
Since 2000
ISSN:2393-1795 ISSN-L:2068-6900

Issue 4/2021

2. The user confirms the entered parameters and calls the system to make changes

Action Description: The user presses the "Change" button, which calls the system operation controlRGB (red, green, blue, mode)

3. The system changes colour or lighting mode

4. The system displays a colour or mode change success message

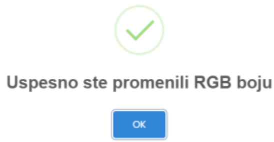

Figure 7. Display a successful change message

Alternative scenarios:

The system recognizes that the user has entered the same colour or mode, displays a message that no change has occurred

Figure 8. Display of change failure message

2.4 Web application implementation and testing phase

The “Smart House” web application is implemented in the Java programming language. The “Eclipse IDE” software was used as the development environment. The “MySql” database was used for data storage. The connection between the application and the database is performed and controlled by “mysql-connector-java-8.0.23”. Execution of the application is performed on the server “Apache TomCat v8.5”.

The microcontroller used during the implementation is a version of the Arduino microcontroller with an integrated network card called “Arduino Node Mcu v3".

The device control code inserted on the microcontroller is implemented in the Arduino programming language, a customized C ++ programming language version. The development environment used to implement the code is "Arduino IDE". During the implementation, testing is also performed, which represents the last and final phase of software system development. Testing was performed after the addition of each new component for easier troubleshooting.

Conclusion

The research presented in this paper presents a detailed and complex process of designing and implementing the web application “Smart House” and the expansive possibilities of its application. The results obtained by this research can be further used and directed in two directions: to examine the possibilities of these platforms and technologies used to design and develop other similar applications that would make life easier for users, and on the other hand, to improve the application “Smart House” by adding new features for users. It is essential to point out that the system can be efficiently improved with easy and quick changes because it is organized in a unique way that allows it a high level of modularity. Some of the ideas for further research and improvement of the web application “Smart House”, which appeared during the implementation, are: control of cooling and heating devices, finishing the security of the house by adding an alarm system and sending notifications via email service if an alarm is activated, as well as adding a video surveillance section.

References

[1] Ball, J. *et al.* (2006) *The Java™ EE 5 Tutorial*, Third Edition: For Sun Java System Application Server Platform Edition, Melbourne: Addison Wesley Professional.
[2] Radenković, B. *et al.* (2017) *Internet inteligentnih uređaja*, Beograd: Fakultet organizacionih nauka Univerziteta u Beogradu.
[3] Jevremović, S. (2016) *Java Programiranje Web aplikacija*, 1. izdanje, Beograd: Visoka škola strukovnih studija za informacione tehnologije – ITS.
[4] Lin H. T. (2013) *Implementing Smart Homes with Open Source Solutions*, International Journal of Smart Home, vol. 7, no. 4, pp. 289-296.
[5] Helal S. et al. (2005) *The Gator Tech Smart House: a programmable pervasive space*, computer, vol. 38, no. 3, pp. 50-60.
[6] Vlajić, S. (2005) *Napredni koncepti Jave i web programiranje u Javi*, 4. izdanje, Beograd: Fakultet organizacionih nauka Univerziteta u Beogradu.

Annals of Spiru Haret University.
Economic Series
Since 2000
ISSN:2393-1795 ISSN-L:2068-6900

[7] Vlajić, S. (2004) *Projektovanje programa (Skripta)*, Beograd: Fakultet organizacionih nauka Univerziteta u Beogradu.
[8] Official web page Arduina: https://store.arduino.cc/collections/boards/products/arduino-uno-rev3 (Accessed 17.09.2021).
[9] Official web page Java Community Process-a (JCP): http://jcp.org/en/home/index (Accessed 14.07.2021).
[10] Le Corbusier (1923), Vers une architecture (Towards A New Architecture). Paris: G. Crès & Cie.
[11] Callaghan V, Clarke G, Colley M, Hagras H Chin JSY, Doctor F, (2004), "Inhabited Intelligent Environments", BT Technology Journal, Vol.22, No.3. Kluwer Academic Publishers,Dordrecht, Netherlands,
[12] Smart Homes for Healthcare, Cindy Zhirui Li, Elizabeth M Borycki (2019), https://pubmed.ncbi.nlm.nih.gov/30741210/
[13] How to Design Smart Homes? 8 Tips for Incorporating Domotics into Architecture, José Tomás Franco, Archdaily, https://www.archdaily.com/908468/how-to-design-smart-homes-8-tips-for-incorporating-domotics-into-architecture?ad_source=search&ad_medium=projects_tab&ad_source=search&ad_medium=search_result_all