Diploma in Software Development

DSE 770 Robotics and Internet of Things

Assessment: Project Report

Total marks: 100

Course Weighting: 100%

Due Date: November 8, 2018

Student Names: Pavel Sobolev

Student Declaration

We declare that all work delivered is our own work and is not copied from any published or unpublished work of others. We confirm that we have read and understood the EDENZ Colleges' regulations on plagiarism as stated in the Student handbook.

Student Name: Pavel Sobolev

Student IDs: EDZ00005KO

Signatures: ____

Date: 10/09/2018

By submitting this assessment electronically, you are deemed to have signed the above declaration.

1 EXECUTIVE SUMMARY

The project “Brain.iot” is dedicated to the development and implementation of software and hardware system for creation of interface between a human brain, computer (BCI – brain-computer interface) and devices of internet of things. The BCI-based system being developed can be used by any interested person primarily for self-development and self-education in the area of human psychology and promotion of level of personal mental effectiveness (by develop ability to control personal level of attention (or activity) and level of calmness (or relaxation)). Also, such a system can be used by people with physical disabilities. Hardware part of the project will include BCI-device and a hardware system for providing a user with live feedback showing demonstrative effect of his brain activity. Two technical systems are chosen to implement such a feedback:

1. home automation system which utilizes modern technologies of IoT (Internet of things) and technologies of smart home.
2. smartphone and mobile software which are aimed to give a user possibility to train one’s ability to consciously control two mentioned above physiological processes.

Both mentioned systems are going to be controlled by data which is received from BCI based headset.

This project peruses the following goals:

1. Planning, assembly and implementation of software and hardware system which can provide users with possibility to interact with computing devices by direct receiving and interpretation of the user’s brain activity data;
2. Studying of modern state of BCI (brain-computer interface) technologies for usage of their capabilities for the system being created.
3. Choosing of appropriate BCI hardware systems which are suitable for achieving of project’s objectives.
4. Choosing of appropriate APIs, protocols, programming languages and frameworks which are optimal for achieving of project’s objectives.

2 INTRODUCTION

2.1 Background

Interaction between a computer and a human can be implemented in different ways. The way which is used for interaction between them is called human-computer interface (Zaphiris & Ang, 2008). Classical human-computer interfaces emerged from the principle of functioning of computing devices. They significantly rely on and depend from the capabilities and inner architecture of used computing facilities. Classical computer is provided with built-in capability to get data from outside and send data to outside through its peripheral devices and inner data streams (input and output data streams). Very important to notice that classical Neumann's computer is not originally capable to adapt itself to abilities and necessities of a user. Vice versa, a human should train and adapt himself in order to be able to use a computer. There are some classical instruments of interaction between computers and their users. Here can be mentioned mice, trackballs, touch screens, keyboards etc. Interaction with computer by using of these devices can be called convenient in some sense but cannot be called natural.

More natural ways of interaction were always desired by wide range of computer's users. And finally, computer architecture and technology has developed to the stage of maturity and effectiveness when they became capable to give adequate response to people's requirements. The primary goal of finding of new and natural ways of interaction between people and computers is to make this activity more convenient and effective namely for users. This can be possible not only by developing of hardware component of computing devices but also by implementation of modern technologies in the area of software, such as artificial intelligence, machine learning, neural networks, big data storage and processing, internet of things and smart home technology, natural language processing etc. All these factors should create new reality where computers study their users and adapt themselves to them, but not vice versa. In this reality natural interfaces will allow people to use computers without preliminary learning efforts and time consumption. Another crucial field of usage of such interfaces is help and assistance for people with physical disabilities.

Nowadays many technologies for implementing of human-computer interface exist. Let's enumerate some the most interesting and prominent of them (Kumar & Arjunan, 2016):

1. Brain-computer interfaces;
2. Muscle's electricity detection interfaces (myoelectric interfaces):
   a. Hand gesture recognition (Thalmic Labs, 2018),
   b. Eyes movement detection (electrooculography) (The Tobii Group, 2018),
   c. Mouth movement detection (leaps and/or tongue movement detection (glossokinetic potential detection)).
3. Interfaces based on technology of video capturing:
   a. Had moving detection and recognition,
   b. Hands movement detection and recognition,
   c. Eyes movement detection and recognition,
   d. Leaps movement detection and recognition, voiceless speech recognition.
4. Speech recognition interfaces:
   a. Microsoft Cortana, Apple Siri, Google assistant, Yandex Alisa, Amazon Alexa.

As was mentioned above this project is devoted to implementation of BCI (brain-computer interface) for controlling of home automation system (smart home system). Before implementation of such a project it is important to understand the principal idea of these kind of interfaces and realize how to use them in practice. It wouldn't be exaggeration to say that human brain is the most complicated object in the known Universe. Understanding of its functioning still remains one of the hardest tasks of modern science. Nevertheless, some practical results of modern physiology, neurology and other science about brain and human psychology allow to create and implement technical devices which can catch some physical aspects of working brain and interpret these data for useful and rational usage.

BCI is a technology which is used to directly send data of human brain activity to process by some computing device (microcontroller or computer) for further interpretation and usage in practical goals. Idea of BCI is based on the results of medical sciences which had being developed for long period of time. BCI uses specially designed devices connected with computing facilities by some kind of interface (wired or wireless). Usually (but not always) the devices which are some way attached to the scalp are compact modern versions of well-known encephalograph which is used for many decades by scientists, medics and researchers of human brain. Data from an encephalograph provides interested doctor or researcher with data about current statement of human brain in the from of EEG (electroencephalogram) which shows neural oscillations (so-called "brain waves"). Interpretation of EEG allows to make valuable conclusion about health of subject's brain and other aspects of brain functioning. So, crucial part of any BCI is the device which can read and transmit brain's data. Nowadays the are called BCI headsets (Bijholt, Femke, Gerritsen, Poel, & Braam, 2015).

Modern BCIs are based on some common practices which were evolving during many decades in the scientific studying of the brain. The only difference that nowadays computers and new technologies (such as artificial intelligence) are used to analyse and interpret brain data. Two standard approaches are used to implement BCI – noninvasive and invasive (Zaphiris & Ang, 2008).

Noninvasive methods don't intrude inside human skull and use electrodes attached to skin of the head in order to read EEG data (fig. 1). Also, these methods can use transcranial magnetic stimulation in order to influence the brain and provide two-way interaction between the brain and the technical system (fig. 2). So, TMS allows send data to the brain.

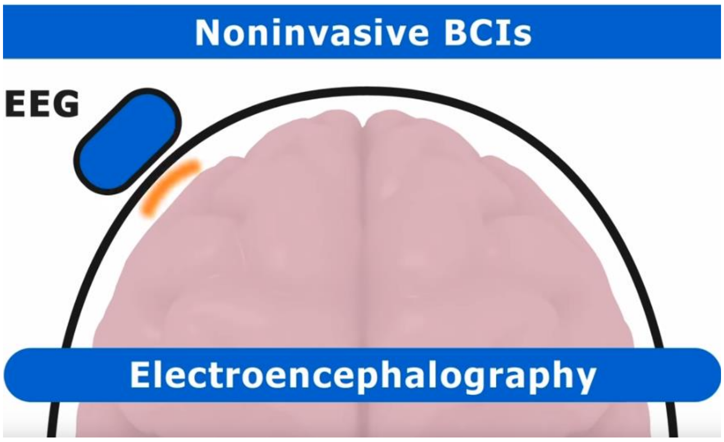


Figure 1: noninvasive BCI schema

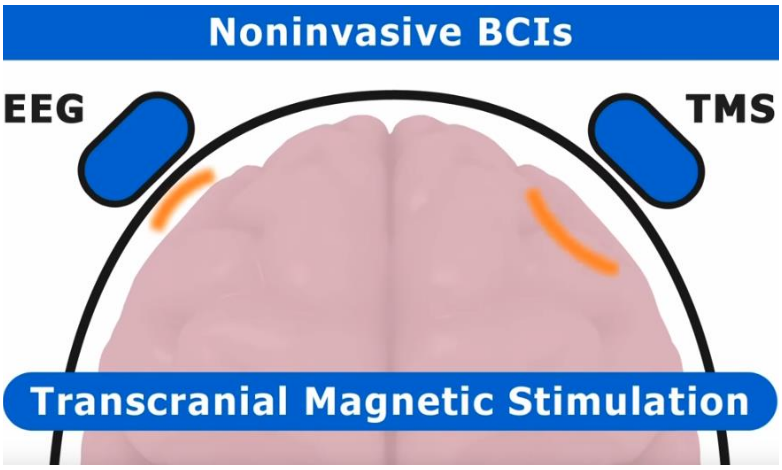


Figure 2: noninvasive BCI schema with backward response

Invasive methods expose the skull to some surgery in order to intrude inner neuro system and get more precise data from the brain. Two types of invasive method can be called. Technology of electrocorticography (fig. 3) allows to attach a sensor to the surface of the brain. Method of intracortical electrodes (fig. 4) goes deeper inside the brain and allows to connect specially designed prongs of sensor with inner brain structures.

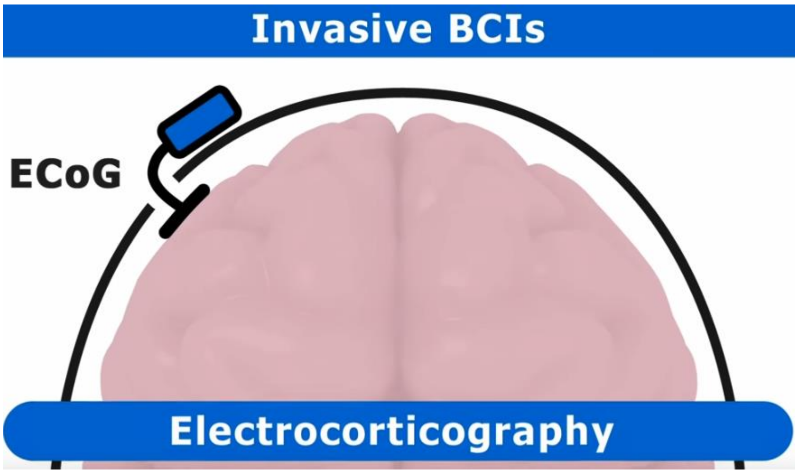


Figure 3: invasive BCI with sensor attached to the surface of the brain

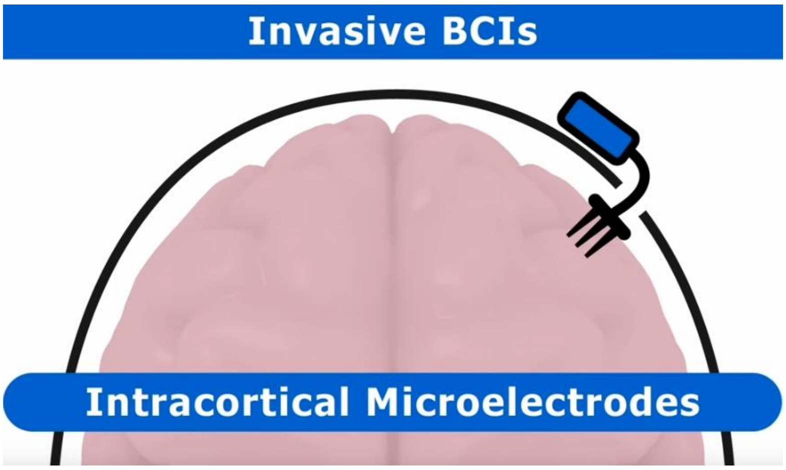


Figure 4: invasive BCI with sensor implanted into the brain

Evidently that this project can use only noninvasive technique. There are some devices on the market which can be used as effective instrument for BCI in the sense of purposes of the project. Some devices were considered as possible hardware part of BCI for this project: Emotive EPOC devices (Emotiv Inc., 2018), InteraXON Muse devices (InteraXon Inc., 2018), Neurosky MindWave devices (NeuroSky, Inc., 2018). Comparison and final choice are described in the "Architecture, Design and Implementation" section. These devices use electroencephalography as a source of data and are based on usage of outer electrodes attached to subject's scalp.

One of the most interesting and important questions in the sense of project's aims is how to interpret main brainwaves' data for practical usage. This project should provide a user with real-time neurofeedback by giving him ability to include brain-computer interface (BCI) to his home automation system ("smart home" system). Such a system will allow to control subject's mental processes by receiving feedback from a smart device being used. At the same time, promoted ability to control own mental processes (with natural limitations) potentially can make possible to intentionally operate home automation devices by the brain data (mental concentration and mental relaxation).

2.2 Scope

The system being created will consist of hardware and software. The principal idea is to organize wireless interaction between a human and a computational system by the intermediary of internet services in order to implement mentioned above functionality. This task can be achieved by implementation of the following conceptual architecture (fig. 5).

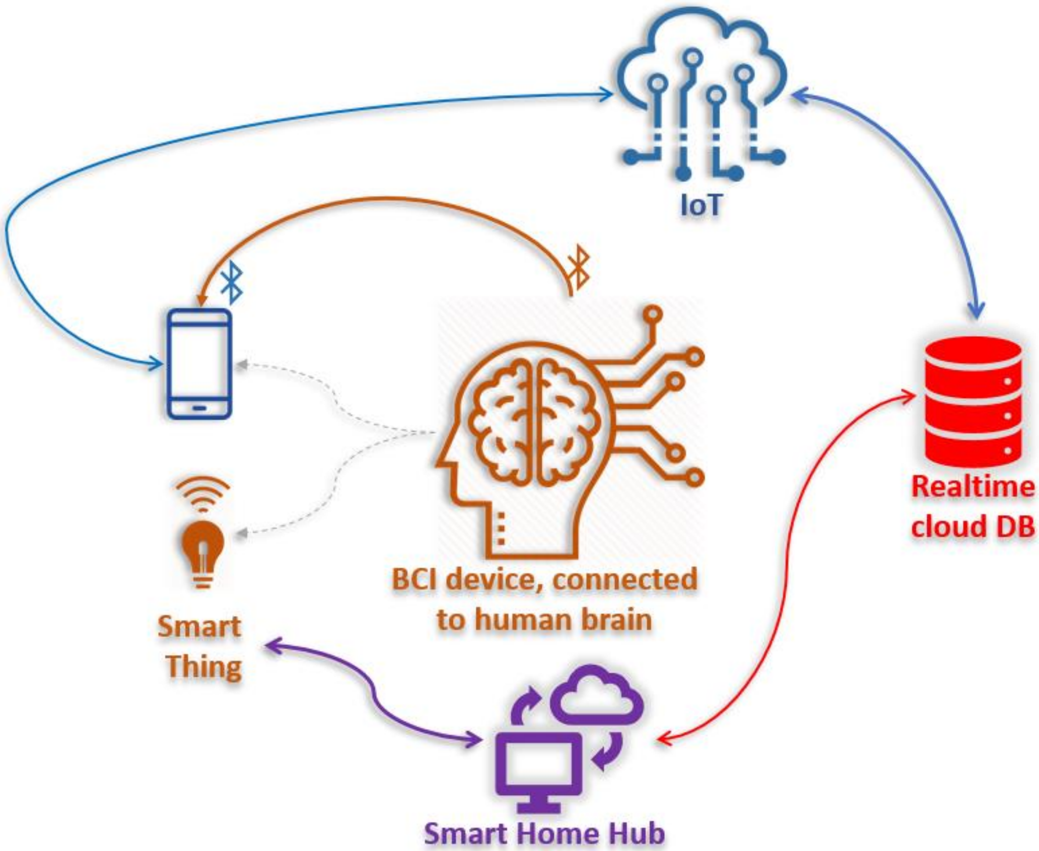


Figure 5: principal schema of the project with usage of internet connection (all connections are wireless)

It is important to notice that project is not about design or creation of hardware part. If to say about this part, it is needed to choose suitable components (and protocols of data exchange) and use them for creation of software which is capable to control functioning of the hardware being used.

So, at the beginning stage of the project it is needed to choose and evaluate suitable hardware devices (which include needed sensors and other parts), protocols for data exchange, available and suitable internet services for IoT. According to figure 5 planned system should contain the following parts:

1. BCI device for getting and transmitting of human brain data in digital form. This device should transmit brain data through some non-proprietary wireless protocol (such as Bluetooth, WiFi, etc.). Saying about brain data we mean information from EEG-sensors, which is the most widely spread and reliable enough method of understanding and interpreting of inner brain processes.
2. Computer (smartphone/desktop/laptop etc.) to receive data from BCI-device. The computer is needed to control and customize the process of interaction between BCI-device and other devices of the system. Also, this computer should be equipped with hardware which is capable to receive data transmitted by BCI-device and should support this device's protocol of data interchange.
3. Internet of things' service and real-time reactive database which has possibility to receive and transmit discrete data stream. This data storage must have function of sending of real-time notification to the interested recipients, which are devices of Internet of things or smart home hubs.
4. Hub is a part of a system of home automation (smart home) which actually is a computer (probably narrowly specialized). It should be capable to receive live notifications from Internet and execute received commands. Devices of this part should be always online. They must have open (not proprietary) and well documented interfaces of commutation with outer systems. Proprietary protocols and devices are not suitable for this project because they exist and function in its own "ecosystem" and cannot be adjusted and programmed from external systems.
5. Smart device is some electrical equipment which has microcontroller and can be controlled directly from the Internet or from hub of home automation. This can be a bulb, a door lock, a sensor of some type, an outlet and etc.

3 PROJECT PLANNING AND EXECUTION

3.1 Software development life cycle

According to formulated goals of the project and control dates laying within range of 8 weeks it is needed to plan and choose appropriate SDLC. Software development lifecycle is a process which controls both overall and concrete activities within the process of creation of software product. Considering studied literature (Langer, 2016; Pressman, 2001) it is possible to say that taking of decision about model of DSLDC to be used for implementation of project should pay attention to a lot of different aspects. Some concrete considerations can be listed according to the mentioned above sources and existing definitions and descriptions of modern models of SDLS being used in the software engineering. Concerning "BCI-IoT-Smart Home" project set of questions were compiled and on the base of the prevailing answers final SDLC model was chosen.

Table 1: comparison of SLDC models for "BCI-IoT-SmartHome" project

Characteristic of the project / Does model correspond this characteristic?	Waterfall	V-model	Spiral
Not all the requirements are well known and easily detectable from the beginning stage of the project.	✗	✗	✓
Not all the requirements were defined preliminary.	✗	✗	✓
Requirements are to be changeable.	✗	✓	✓
There is a need to implement some requirements on early stages of the project (especially it concerns connectivity of devices).	✗	✓	✓
Project demands development of overall system architecture in the area which is new for the developer (BCI, smart home).	✗	✗	✓
There are some unknown issues of software development in this project (Bluetooth programming, programming of smart home devices, programming of cloud data storage).	✗	✗	✓
Tools of development relatedly new to the developer.	✗	✓	✓
There is not team working on the project. Changing of roles is not applicable.	✓	✓	✗
It is possible to get some additional training for the project developer during the process of development.	✓	✓	✓
Customers are going to be involved in the process of development (in the person of the project's supervisor).	✗	✓	✓
Possible users can evaluate current condition of the project (in the person of the project's supervisor and developer).	✗	✓	✓
Consumers are planned to participate in each phase of project SDLC.	✗	✗	✓
Consumers (in the person of the project's supervisor) can observe the process of development.	✗	✗	✓
This project isn't middle or large scaled project.	✓	✗	✗

The project doesn't extend existing projects.	✓	✓	✗
This project is expected to be exploited for some period in the future.	✗	✓	✓
Ready program is expected to be changed after finishing of the project.	✗	✓	✓
The project has strict time limitation.	✓	✓	✗
Created components of the projects can be reusable.	✗	✗	✓
There are enough resources for the development of the project (time, development tools).	✓	✓	✗
Overall score	6/20	12/20	**15/20**

So, according to undertaken analysis spiral model of SDLC is the most appropriate for this project.

3.2 Project Plan and Gantt Chart

The project planning period is from Monday, 10th of September till Thursday, 1st of November (8 weeks).

Table 2: description of project's plan

	Description of activity	W1	W2	W3	W4	W5	W6	W7	W8
1	Planning and design of overall project's architecture (considering hardware and software components for future IoT infrastructure of the project). Choice of SDLC.								
2	Studying of brain computer-interface.								
2.1	Analysis of hardware components, choice of the BCI-device. Evaluation of suitable technology, sensors, etc.								
2.2	Analysis of software components for the device (API, SDK, suitable programming languages).								
2.3	Creation and testing of mobile application's prototype for interaction with chosen BCI-device.								
3	Studying of smart home technologies.								
3.1	Analysis of hardware of home automation systems with the respect to project's aims. Choice of equipment. Choice of home automation hub.								
3.2	Analysis of software components of home automation systems, APIs, SDKs, protocols.								
3.3	Creation and testing of prototype of software for intercommunication between mobile application and smart home device.								
4	Studying of technologies for implementation of IoT.								
4.1	Choice of suitable technology for IoT implementation (considering performance, ethics issues, security issues, consistency of data being used). Design of database structure.								
4.2	Analysis of software components of IoT: APIs, SDKs, programming languages.								

	Description of activity	W1	W2	W3	W4	W5	W6	W7	W8
4.3	Creation and testing of the prototype of the application for intercommunication between BCI and IoT, home automation system and IoT.								
5	Overall project development according to proposed architecture.								
5.1	Assembling all the hardware components in one system. Testing of previously created prototypes for suitability for using in the final software being developed.								
5.2	Development and testing of mobile application for communication between EEG BCI-device and IoT component of the project.								
5.3	Development of application for home automation hub for implementation of interconnection between IoT component, home automation server and smart device.								
5.4	Overall testing of all components working together.								
6	Report writing.								

3.3 Risk Management

Risk management is a methodology which allows developers to elicit possible risks that can happen during the process of development and, thus, be prepared in the case of their emerging. Three parts of risk management can be called: assessment of possible risks, mitigation of risks which happened or precautions for prevention or mitigation, and risk costs evaluation (Stoneburner, Goguen, & Feringa, 2002).

On the base of analysing of sources (Arnuphaptrairong, 2011; Wallace & Keil, 2004) all software project risks can be categorised by so called risk dimension, which defines the perspective of view for determining source of possible emerging risks. According to mentioned authors 6 risk dimensions can be elicited: risks coming from users or customers; risks connected with software requirements; risks coming from project complexity; risks from project planning and control; risks coming from the project team; risks coming from organizational environment; risks coming from the used equipment and software. Result of the analysis is presented on the following list.

1. Users/customers risks;
2. Requirements risks;
3. Project complexity;
4. Control and planning risks;
5. Risks from development team;
6. Environment risks;
7. Software and hardware risks.

Based on studied sources which devoted to risk management (Arnuphaptrairong, 2011; Wallace & Keil, 2004; Stoneburner, Goguen, & Feringa, 2002), 4 possible groups of risks for “BCI-IoT-SmartHome” project were detected (table 3).

Table 3: "BCI-IoT-SmartHome" project risks

#	Possible risk	Possible effect	Precautions for prevention
1	Hardware and electrical equipment risks		
1.1	Issues of the device of brain-computer interface.	Deadline of project can be broken. Not suitable or adequate data of brain activity are provided.	Thorough following of the device instructions. Handling of errors and exceptions of device's connection during execution of software being created.
1.2	Issues of home automation device.	Deadline of project can be broken. Impossible to implement demanded functionality and provide adequate neurofeedback.	Thorough following of the device instructions. Only approved (according technical specifications) and tested data should be sent to the device.
1.3	Computer failures (first of all, data storage system failures).	Possibility to lose software projects files, failure of projects due dates.	Using of software for control versioning; usage of cloud storages.
1.4	Smartphone issues.	Malfunction in the software while connecting with BCI or remote internet services.	Handling of errors and exceptions of wireless protocols' connections during execution of software being created.
1.5	Failures of hardware and equipment suppliers (or delivery services) because some components were bought online.	Due dated can be broken.	Choice of reliable suppliers (such as Amazon.com) or nearly located suppliers (in New Zealand or in Australia).
1.6	Not permittable smart devices were chosen (some technologies of home automation are limited by local legislation; for example, working frequencies of ZigBee of Z-Wave devices must comply with demands of concrete country (see table4)).	Due dated can be broken.	Thorough following the demands of New Zealand's law and regulations in the area of used radio frequencies.
2	Problems with requirements		
2.1	No all the requirements were elicited.	Project's functionality does not correspond to	Thorough studying and following of the 770RIoT course demands.

		required learning outcomes.	Discussion of requirement with project's scientific supervisor before starting the project and during its implementation according to chosen spiral SDLC methodology.
3	Problems with overall architecture of the system.		
3.1	Not correct or not adequate choice of the system nodes (devices, protocols) due to lack of experience in the area of the project.	Failure of due date of the project because of possible	Discussion of architecture components with scientific supervisor during all the SDLC of the project. Short public presentations for groupmates.
4	Software related problems including testing		
4.1	Lack of experience in the area of brain-computer interface. Not enough experience of programming and handling of streams of empirical data.	Not correct conclusions about data which are being received during sessions of working with the device. Not correct algorithms applied to the data being received.	Studying of manuals of manufacturer of BCI-devices about data which is provided by the concrete device and their sense in the BCI-system (for example, how to measure level of attention or tranquillity of the subject's brain, etc.). Studying materials about EEG and sense of its data (brain waves and etc.).
4.2	Lack of experience of programming of IoT.	Choice of not correct or not enough adequate IoT services which are not compatible with other components of the system being designed.	Thorough study of existing IoT services and their abilities, prices and limitations. Discussion of IoT components with scientific supervisor during all stages of the SDLC.

4 SYSTEM DESIGN AND IMPLEMENTATION

4.1 Discussion of technologies, hardware and software used

According to previously described architecture (figure 5) it is needed to analyze and choose suitable devices and other relevant hardware components to build this architecture. Also, it is needed to decide on data interchange protocols and software development interfaces.

4.1.1 Discussion of mobile devices for brain-computer interface

Nowadays the idea of brain-computer interfaces is becoming more and more popular because of miniaturization of devices which are capable to read data of human brain activity. These devices became wireless and wearable. This means that know a consumer does not need to sit inside special laboratory or hospital in order to get and use data about his brain activity. This project has educational purposes and because of that I need to choose some compromise devise which is capable enough and at the same time is affordable by its price and its capabilities. Today we can call at least three manufacturers which provide modern market with portable EEG-devices. Some devices were considered as possible hardware part of BCI for this project: Emotive EPOC devices (Emotiv Inc., 2018), InteraXON Muse devices (InteraXon Inc., 2018), Neurosky MindWave devices (NeuroSky, Inc., 2018). All of these devices use electroencephalography as a source of data and are based on usage of outer electrodes attached to subject's scalp. Resulting preciseness and reliability depends from number of sensors, their sensitivity and characteristics of algorithms being used inside devices to calculate and interpret final results of measurements. Following table #3 shows comparison of the most available mobile BCI-devices. Three basic devices were considered as possible candidates for this project (figures 6-8).

Figure 6: NeuroSky MindWave Mobile 2 Headset

Figure 7: Muse 2 - The Brain Sensing Headband

Figure 8: Emotiv Insight Headset

Table 4: comparison of technical specifications of mobile BCI EEG devices (according to (Emotiv Inc., 2018; InteraXon Inc., 2018; NeuroSky, Inc., 2018))

	Muse 2: The Brain Sensing Headband	**NeuroSky MindWave Mobile 2 Headset**	**Emotiv Insight Headset**
Number of EEG-sensors	4	1 + 1 reference	5 + 2 references
Providing of raw EEG-data	Yes	Yes	Yes
Wireless interface	Bluetooth Low Energy v5 + Endpoint detection and response (EDR)	Bluetooth Low Energy	Bluetooth Low Energy
Wired interface	USB (charging only)	Not provided	USB
Brain waves recognition	Absolute and relative power for delta, theta, alpha, beta, and	Alpha (2 ranges), Beta (2 ranges), Theta, Gamma (2 ranges)	Alpha (2 ranges), Beta (2 ranges), Theta, Gamma (2 ranges)

	gamma, for each channel.		
Derived mental characteristics being supplied	Mellow, concentrate	Attention, meditation, mental effort (desktop API only), familiarity (desktop API only)	Excitement, Engagement, Relaxation, Interest, Stress, Focus
Facial expressions	Blinks, jaw clenches	Blinks	Blinks, Wink L/R, Surprise, frown, smile, jaw clench.
Additional sensors	Accelerometer, gyroscope, heart beat sensor, posture sensor, breath frequency sensor	Gyroscope	Accelerometer, gyroscope, magnetometer
Electrical noise suppression	50/60HZ (regional)	50/60HZ (regional)	50/60HZ (regional)
Power supply method	Internal Lithium Polymer battery	AAA battery, not rechargeable	Internal Lithium Polymer battery 480mAh
Working time	Up to 6 hours	Up to 8 hours	Up to 8 hrs (USB), 4 hrs (Bluetooth)
Supported operating systems	iOS, Android, Windows 10 (UWP)	Windows, Linux, MacOS, iOS, Android	Windows (7 min), MacOS (9 min), iOS (9 min), Android (4.4 min)
Data exchange protocols	Raw binary, JSON	Raw binary, text (JSON) through socket API.	Raw binary, text (JSON) through HTTP.
Supported means of outer interactivity for 3d party software	Bluetooth	Net sockets (through ThinkGear connector API), Bluetooth.	RESTful API, Bluetooth
3d party development support	Muse SDK, Muse direct (partially free)	Free API (last update was in 2016)	Cortex SDK (free API for noncommercial use only); developer site and community
Supported programming languages	Java, C++, C#, Swift	Java, C#, C#, Objective C, Python	Java, C#, C++, Python, Ruby, NodeJS, PHP
Basic price (USD) for New Zealand	274	120	410

After analysis of described devices and with accordance with formulated goals and educational nature of the project I chose “NeuroSky MindWave Mobile 2 Headset”. This device has rather powerful functionality along with affordable price. Very important feature is that manufacturer provides fully functional free application programming interface and software development kits for the most widely used programming languages. Furthermore, manufacturer has special service application “ThinkGear Connector” which provides additional protocol of connection for third party applications (this service works as a net socket and calculates not only level of attention and meditation, but also additional characteristics of mental activity, such as level of familiarity and level of mental efforts; usage of this service allows developers of client applications not to program raw data from Bluetooth connection but to use more convenient ways of data interchange in the form of JSON). Slight drawbacks of the chosen device are following: usage of AAA batteries (which makes it more expensive in exploitation), "ThinkGear Connector" interface and additional data of mental activity is not available yet for mobile devices' SDKs. Also, practical usage of the device showed me one disappointing phenomenon. The device being used fails to provide reliable data about eyes blinks of a user (for some people MindWave can detect all the blinks, but for other not even one blink was detected correctly). Anyway, usage of the device allowed to implement goals of the project (but data about blinks would give possibility to implement richer functionality). Strong side of the device is that it is accompanied by well documented application programming interface (NeuroSky Inc., 2018). Also, the manufacturer provides detailed description of data formats in which streams of data are generated.

According to the goals of the project and factual functionality of the chosen device it is possible to implement handling of two major characteristics of subject's mental activity: ***level of attention*** and ***level of mediation*** (these names of characteristics were given by the manufacturer of the device (NeuroSky Inc., 2018)). In general words it is can be said that level of attention measures how active and concentrated person's brain during the session of measurement. Vice versa, level of mediation shows how relaxed and "empty" the mind of a person during the session of measurement.

NeuroSky MindWave headset calculates levels of attention and meditation in the form of integers within the range from 1 to 100 (value of 0 means that device is not functioning properly). These numbers can be received through the usage of API provided by manufacturer of the device. These integers along with data about brain waves (such as alpha, beta, theta, gamma) are sent through the Bluetooth channel in the from of raw binary packets of data (NewroSky Inc., 2018). Manufacturer's high-level API provides data types and methods for retrieving of data through the Bluetooth channel and decoding of fields of raw binary data packages. High-level libraries are provided for Google Android, Microsoft Windows and Apple MacOS operating systems.

The project is aimed to provide its users with ability to implement interactivity between the BCI-device with some IoT device through the intermediary of some computing device (because BCI-devices by themselves don't have internet connection yet). Smartphone was chosen as the primary intermediate device for the project. First of all, smartphones are the most used personal computing devices. Usage of smartphone allows a user not to be bound to the place where the PC is located (if PC or Mac is used) or not to carry with himself heavy laptop. Furthermore, almost any modern smartphone provides Bluetooth connectivity and supports this data exchange protocol (which is needed for this project). According to modern statistics Google Android is the most widely used mobile operating system for smartphones. This circumstance shows that system being developed should be firstly aimed for using with smartphones running Google Android operating system. As was mentions before, NeuroSky provides high-level Java library for interaction between Android apps and its MindWave devices.

4.1.2 Discussion of used technologies for IoT implementation

Internet of things provides with the ability to interact with usual devices (such lamps, outlets, kettles, door locks, etc.) through the Internet connections. Furthermore, things of Internet can interact between themselves directly to provide a human with high level of comfort and service. Functioning of IoT is implemented through various types of protocols and industrial standards which emerged not much time ago. Nowadays we can observe booming of this technology when scientists, industrial manufacturers invent and suppose new ways of IoT connectivity and interactivity. So, this area is very far from some type of unification and widely accepted standardization (CISCO, 2017).

This project should provide a user with the ability to control smart home device by the data of his brain activity through internet channel. This can be achieved by different ways. One of ways is to try using of some patented and proprietary protocols and technologies. This way can does not provide ability to freely adapt functionality of the system to the needs of the project itself. The problem is that such proprietary technologies very often consist of closed set of devices and closed protocols which can't accept data from outside (for example, from BCI-devices because none of them is a part of these kind of IoT system). This reason points out that we need to use open protocols which at the same time are well described and widely supported by the community of developers of software and manufacturers of hardware. One of the most popular hardware platforms for implementation of IoT is Raspberry PI compact computer. This computer which operates under Linux operating system is secure and effective (for tasks of controlling of IoT) at the same time (Hosmer, 2018). Raspberry PI computer has small size, relatively cheap and supports wireless Internet connections, Bluetooth connections and can be expanded by third party devices and sensors.

The idea of the project is to use Raspberry PI computer as a controller of smart home devices. This computer should be connected to the Internet permanently. In the architecture of the project this controller must be capable to get commands from outside and transform them into the commands directed to the smart devices. Inbound data will be received from NeuroSky MindWave BCI-device and then sent to the controller through Internet channel. In general, this idea is illustrated by figure 9.

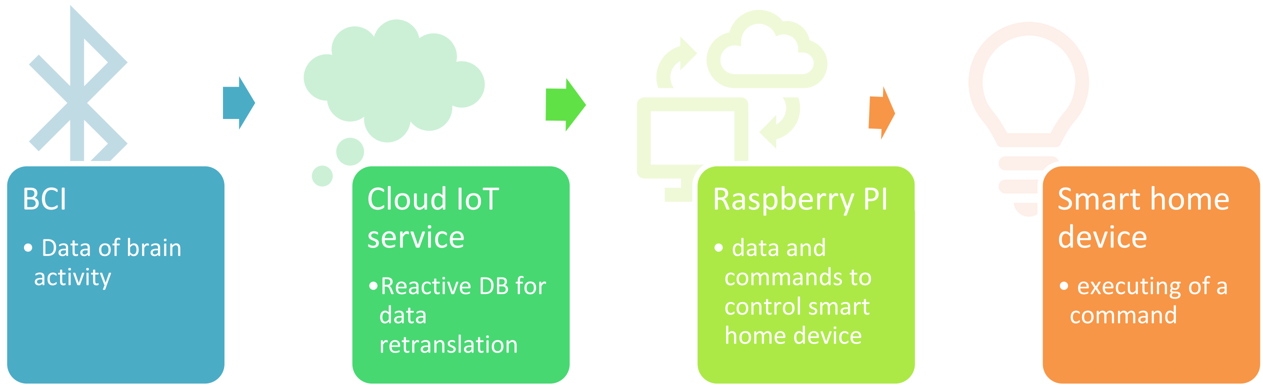


Figure 9: data translation in the system being developed

Second part of the system should be an Internet service which is capable to accept and retranslate live stream of data generated by devices connected to the internet as a part of IoT system. If to say about BCI-device being used, it can produce significant amount of data during short period of time. Brain activity is very changeable and is measured by many characteristics. For example, during the period of active development, debugging and testing of the system's software (from 10 to 23 of October 2018) more than 1.7 GB of brain waves data was generated by the brain of the developer (figure 10).

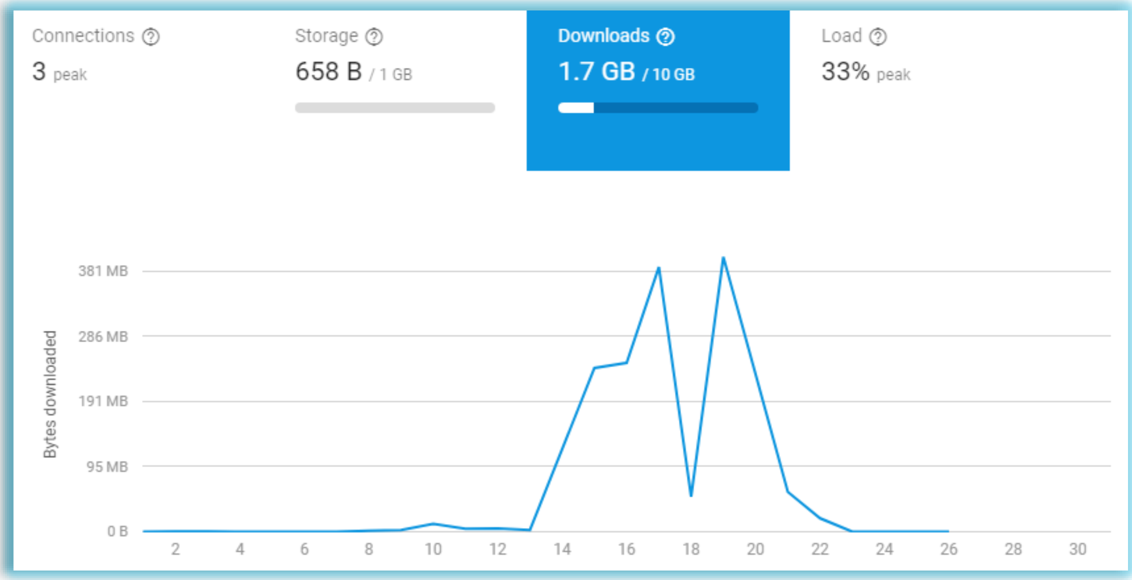


Figure 10: statistics of generated data from BCI-device being used (Google Firebase console)

Furthermore, it is vitally important for the project to be capable to send this data in real-time mode and provide smart home device with latest changes in user's brain activity. Transferring data between devices of IoT is implemented by public commercial and open Internet services. Here some the most prominent services can be called (Grousse, 2018):

1. AWS IoT (Amazon Corp., 2018);
2. IBM IoT platform (IBM Corp., 2018);
3. Google Cloud IoT (Google LLC., 2018);
4. Microsoft Azure IoT (Microsoft Corp., 2018).

In the sense of the project's aims IoT service should provide:

1. data storage for live stream of data from BCI sensors,
2. events architecture which allows receive and notify immediately the system controller (Raspberry PI) about changes of existing data or about arriving of new data.

Considering that the project is educational and noncommercial it is important to choose such a service which provide mentioned functionality for free during significant period of time. From this perspective of view Google "Firebase" service (as a part of Google IoT service) looks as the most suitable. Google Firebase (Google LLC., 2018) is a reactive real-time cloud NoSQL database which can be used for free with some limitations which are not significant for this project (free plan gives ability to use database of size up to 1GB and download up to 10GB of data every month; another great feature is ability to establish up to 100 of simultaneous connections). Firebase is not pure IoT platform as itself, but it can be used to build IoT architecture because it has ability to collect big amount of data in the cloud and reactively notify connected consumers about any change which happened. In the sense of the project's aims this functionality is adequate and sufficient. After each change of state of user's brain, the BCI MindWave headband sends new data to smartphone which analyses and resends needed data to Firebase internet storage. Then inner mechanisms of Firebase engine will notify registered instances of interested clients about arrival of new data.

In the case of this project, interested client is a process running on the Raspberry PI computer and capable to interact with home automation system being used.

Also, it is important to mention that Google Firebase database protects data by providing instruments for restriction of access to the parts of database or database in general. Third party applications or extraneous users are not allowed to freely access these data. Considering that the database of this project holds personal data of user's brain activity this approach prevents some possible ethical and security issues.

Figure 11: Google Firebase logo

Summarizing this part of the report I can say that IoT infrastructure of this project will be implemented with utilization of Raspberry PI computer as IoT client and Google Firebase DB as IoT server.

4.1.3 Discussion of home automation technologies being used

System of home automation (also called "domotics" or smart home) is a combination of specially designed hardware, software and protocols of data which used to provide consumers with ability to control devices in their home environments. Nowadays systems of home automation are tightly connected with Internet of things and very often they are considered as an integral part of IoT technologies (Hersent & Boswarthick, 2012) which allow to control home appliances remotely (even from outside of home). Depending on the technologies and protocols which are used smart home system contains of different parts. But anyway, the main part of any such system is a smart device. Smart device is any electrical appliance or device which can be controlled remotely by sending commands in some format (which depends from used protocol of interconnection). Smart home devices can accept data and commands from outside and send data and commands to other devices and outer systems (for example, SMS or email notifications) (Goodwin, 2013).

Nowadays market of smart devices is booming and includes may kinds of home automation hubs and controllers (for overall control of the smart home system), electric equipment (LED bulbs, switches, outlets, energy consumption meters), plumbing hardware (leakage sensors, valves), security appliances (smart door locks, sensors of movement and presence, CCTV systems, etc.), home climate control systems and many others.

Smart home includes not only devices itself but also protocols which are used to control them. Modern state of home automation systems shows diversity of used protocols (Hersent & Boswarthick, 2012). But there is a process of some unification and standardization of technologies which demands some clear and unified rules in this area. Some protocols for IoT and smart home technologies can be called:

- Bluetooth mesh networking,
- Near-field communication (NFC),

- Wi-Fi,
- Z-Wave,
- ZigBee,
- HaLow,
- Long-range wide-area networking.

Analysis of modern market of shows that majority of devices belong to some limited number of protocols. Some of the most widely used protocols are ZigBee (ZigBee Alliance, 2018), fig. 12 and Z-Wave (last version of protocol is called Z-Wave Plus) (Z-Wave Alliance, 2018) fig. 13. These are mesh protocols which provide ability to transfer data between devices of smart home by utilizing of radio channel. Both protocols are supported by relatively wide range of manufacturers which form two corresponding alliances. Some of the most important characteristics of these protocols are listed in the table 4.

Figure 12: Logo of ZigBee Alliance

Figure 13: Logo of Z-Wave Alliance

Table 5: characteristics of ZigBee and Z-Wave (on the base of (Charara, 2018)**)**

	Z-Wave	**ZigBee**
Year of creation	2001	1998
Number of members of alliance	About 700	About 400
Companies which manufacture production with support of the protocol	Aeotec, Samsung, Honeywell, Huewei, Fibaro, LG, Logitech, and others	Philips, Samsung, Amazon, Hive, Honeywell, Ikea, Belkin, LG and others
Number of certified devices on the market	About 2400	About 2500
Carrier of the signal of the protocol	Radio signal in the band of 2.4 GHz (in some countries this value can be different and depends from national regulatory laws:	Radio signal in the band of 2.4 GHz (in some countries this value can be different and depends from national regulatory laws:

	China, European Union allow 868.4 MHz; USA – 908.4, Russia – 869 MHz, Australia and New Zealand – 921.42 MHz)	China allows 784MHz; USA, Australia and New Zealand – 915 MHz, Europe – 868 MHz)
Coverage distance	Up to 100 meters	Up to 30 meters
Devices connectivity	Mesh network	Mesh network
Number of devices in the same mesh	Up to 232	Up to 65000
Interoperability of devices	Fully implemented (each device can “talk” to another)	Fully implemented in the latest version (old devices can interoperate only with devices from the same “semantic” group (i.e. a bulb cannot “talk” to a heartbeat sensor))
Probability of interference with signals from other sources (Wi-Fi, Bluetooth)	Not possible	Possible, bur very low (with Wi-Fi)
Presence of centralized hub (connected to the Internet)	Hub is required (Internet connection is optional)	Hub is required (Internet connection is optional)
What can be used as a hub	Any computer with specially designed dongle (for example, USB-dongle); specially designed device.	Any computer with specially designed dongle (for example, USB-dongle); specially designed device.
Security implementation	128-bit key encryption	128-bit key encryption
Approximately estimated number of certified devices in the use	More than 94 million	No data

On the base of this comparison it can said that both protocols have similar functionality. At the same time there are some universal hardware and software platforms which can provide programmability and interoperability with both of them (utilization of such platforms can allow to use any protocol). According to the goals of the project both protocols are suitable. Final choice was made in favor of Z-Wave protocol because of its low probability of interference with other radio-based protocols and longer distance of coverage.

After making the choice of the protocol of home automation system we need to opt some concrete equipment and software which can be used for the goals of the project. As was mentioned in the table 2, Z-Wave protocol demands presence of so-called hub which us used to send commands to devices in the home automation network. This hub should be connected to the internet. According to previous discussions where Raspberry PI computer was chosen as a hardware platform for IoT infrastructure of the project we need to turn this computer to the hub of home automation. This can be made by the usage of specially designed USB-dongle, which can be plugged in the USB slot of this computer. There are some different USB-dongles of this type of different manufacturers on the market. Aoeotec Z-Stick Gen5 (figures 14, 15) was chosen as the Z-Wave controller for this project. Major advantage of this device is that it supports band working frequency allowed in New Zealand (921.42 mHz). Also, this device supported by all major operating systems (Linux, Windows, MacOS). Interaction with this controller is implemented through standard COM-port (which significantly simplifies its programming).

Figure 14: USB Z-Wave Controller "Z-Stick Gen5" being used in the p roject (Aeotec) (Aeotec Inc., 2018)

Figure 15: "Raspberry PI" and "Z-Stick Gen5" controller which are used in the project (27 October 2018, EDENZ Colleges)

Next stage of the project is the choice of the smart device (connected to Z-Wave network) for automation. This project needs to automate a smart device by sending to it commands which should be chosen and configured on the base of data retrieved from user's brain activity. Smart LED bulb was chosen as a controlled smart device because of the simplest way of delivering of visual and comprehensible neurofeedback. Such devices are capable to change level of brightness and hue of color along with obvious ability to be remotely switched on and off. This circumstance opens possibility to map data of brain activity to levels of brightness or hues of color of the LED-bulb being used. Concrete algorithms of such mappings are discussed below in the chapter devoted to software development.

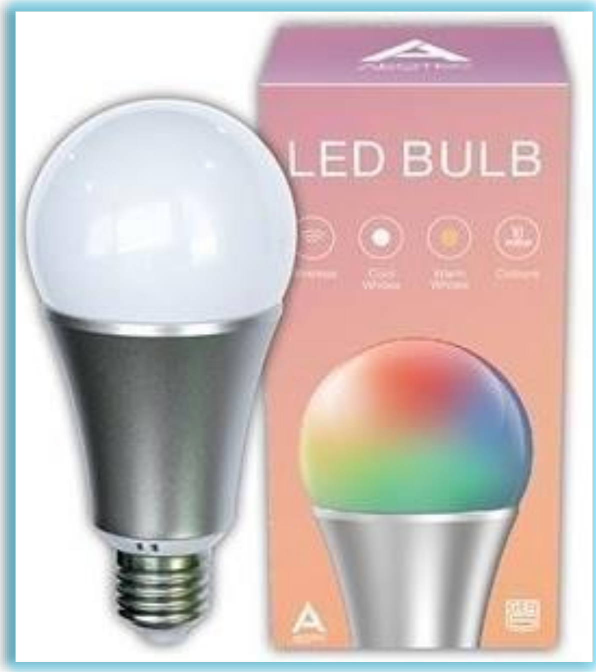


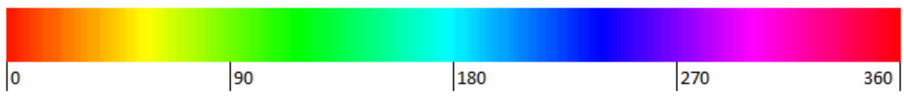
Figure 16: LED Bulb used in the project (Aeotec) (Aeotec Inc., 2018)

This project uses "Aeotec LED Bulb Gen5 Multi-Colour" device which supports Z-Wave protocol and has the following tech specifications (figure 17).

Figure 17: specification of smart device being used (Aeotec Inc., 2018)

According to (Aeotec Inc., 2018) the bulb is can be controlled in some standard ways.

Table 6: controlled states of the bulb being used

#	Controlled parameter	Accepted values
1	Turning on and off	0 (off), 1 (on).
2	Brightness	0 (of)…100 (maximum brightness).
3	Colour coding according to HSV (hue, saturation, value) system.	Hue within range from [0; 360]; brightness within range [0;100]. 

And finally, I need to opt some software platform for implementation of interactivity between Z-Wave controller being used and software which is planned to be executed on the home automation hub. According to the planned architecture this software is going to be a script which can get data from remote EEG device (through Google Firebase real-time service over Internet connection) and convert this data to commands directed to Z-Wave controller. The main idea of such software is to organize permanently running server which can receive and resend commands for smart home devices (commands can be formed locally or remotely from internet).

Figure 18: Logo of “Domoticz” – home automation server

There are a lot of servers of home automation on the market. Some of them are free, and some others are to be licensed and paid. According to recommendations of manufacturer of used Z-Wave equipment (Aeotec Inc., 2018), free “Domoticz” (figure 18) home automation server was chosen as software platform in this project (Domoticz, 2018). Other examples of analogical software systems are LinuxMCE, zVirtualScenes, OpenControl, FHEM, OpenHAB, etc. Manufacturer of appliances being used (Aeotec) nevertheless recommends using of “Domotics” server because devices were tested for compatibility with this system of home automation. This program has some very useful features which make it suitable for the project being developed.

- This software is cross platform with open source codes. It can be installed under Linux, Windows, MacOS operating systems. So, it can be used in Raspberry PI computer which was chosen as a hub for home automation system.
- This software is permanently running Web-server, which accepts commands on the address http://localhost:8080 or http://(computer_ip_adress):8080. This Web-server has open RESTful API which is well documented. This type of interface allows local and remote internet queries (Domoticz team, 2018).
- This server is programmable through the usage of its RESTful API which supports JSON format for data and commands interchange.
- This software is compatible with different types of IoT and home automation protocols (in the case of presences of their hardware controllers in the system). So, “Domoticz” server can be used to control both Z-Wave and ZigBee devices by intercommunicating with their dongles.
- A lot of additional features are implemented (daily scheduled control, seasonal scheduled control, usage of building’s plans for placing and controlling of smart devices, sending push notifications to messengers and mobile phones (SMS) and many other).

4.2 Final architecture of the project

Based on previous discussions, final architecture of the project is designed (figure 19). For better understanding of the diagram a color coding was introduced:

- Gray color depicts physical devices (with sensors, actuators and etc.) used in the system being created. This project uses Z-Wave controller (fig. 14), smart Z-Wave bulb (fig. 16), mobile EEG device (NeuroSky MindWave, fig. 6).
- Yellow color indicates computing devices (computers) controlled by standard operating systems (such as Linux, Android, Windows and etc.). This project uses Raspberry PI (see fig. 15).
- Pink color indicates third party software which is used a service to achieve the goals of the project. This project uses Google Firebase cloud service and server of home automation "Domoticz" (along with operating systems (Debian Linux, Google Android, Microsoft Windows), drivers of devices and executing environments (such as Java Virtual machine, Node.js and etc., which will be discussed further)).
- Light green color indicates software which is developed by the author of the project.

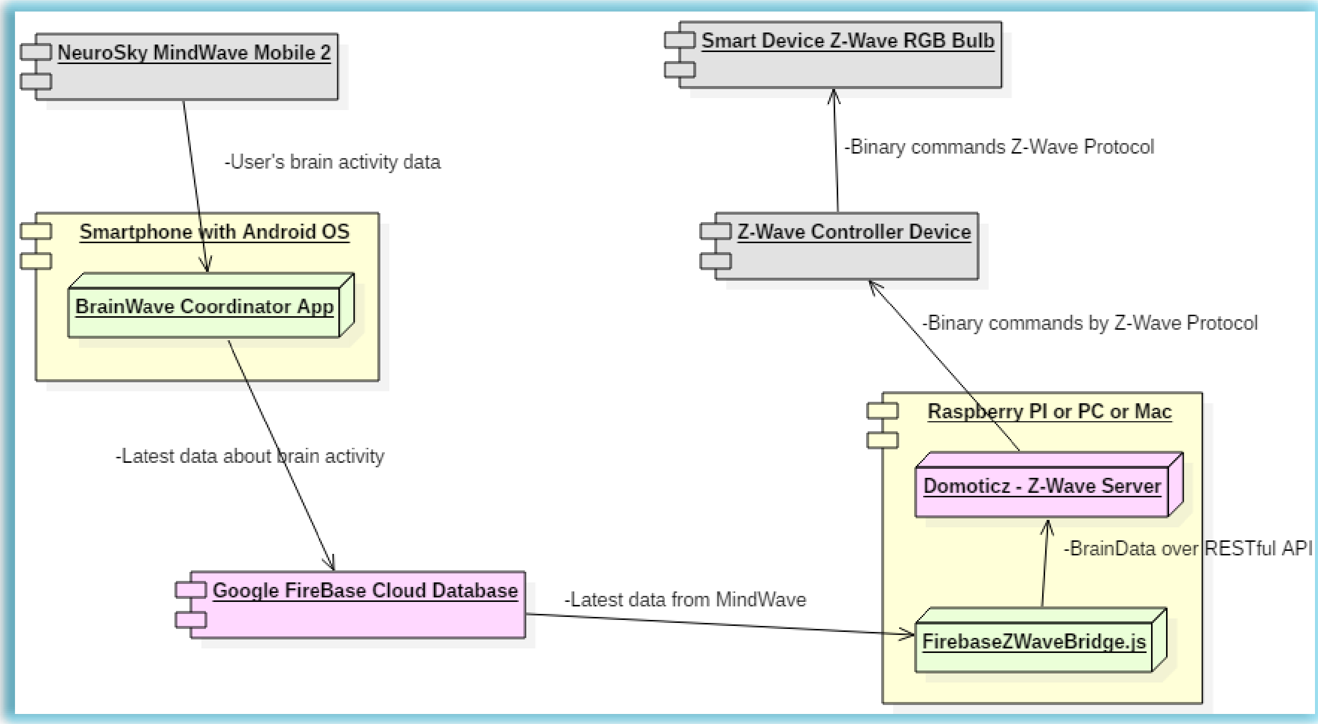


Figure 19: final software and hardware architecture of the project

Two pieces of software should be developed to fulfill the project aims.

First is "Brainwave coordinator application" which should retrieve raw data of human brain activity produced by NeuroSky MindWave headband. This application is designated for Android operating system and should comply with the requirements listed below in the table 5. Main goal of this software is to provide real-time neurofeedback by getting live stream of data from user's brain. Program must show these data in visually evident and easily understandable form. Another function of the application is to give a user ability to choose ways of controlling of smart device by providing him with appropriate GUI.

Second is a script for implementation of interaction between server of home automation system and stream of data from real-time database, containing information about user's brainwave data and commands directed from user's smartphone to controlled smart device (in this project smart bulb is meant). This script allows to distantly control functioning of smart device by converting brainwaves data to commands of home automation system.

Figure 20: final architecture in action

4.3 Reflection

This part of the project was devoted to overall system architecture. During this process I gained a lot of new knowledge from different areas. First of all, brain-computer interfaces should me mentioned. This is modern and extra interesting field of knowledge which combines neurosciences and informatics. Both of these sciences are in active process of growing and continuous evolving. Every day brings us new facts about human brain and new computer technologies. Our knowledge about the brain is still incomplete and many questions are still unresolved. Nevertheless, direct interaction between human brain and a computer is possible already and can be used outside of some sophisticated laboratory environment. Today we have consumer electronics which can read the brain's encephalogram and digitize this data. Usage of such device is the base of this project. Another significant interest is how to use data from BCI devices. I use data of brain to control state of outer smart device which can give visible and easily detectable neurofeedback. I gained new knowledge about home automation systems, protocols of their functioning and software which can be used to interact with these systems. Also, internet of things is a crucial part of the project. IoT is dynamically evolving technology and combination of IoT, brain-computer interface and home automation system became some challenge for me. I designed the architecture of the system which combine these three technologies together. Following development of the project demonstrated practical applicability of this architecture in the sense of feasibility of the project's aims.

5 HARDWARE AND SOFTWARE REQUIREMENTS

Table 7: requirements for hardware and software parts of the system

HR – Hardware requirements	
HR-1	System should provide a BCI wireless device with ability to provide user with at least two independent characteristics of a subject's brain activity.
HR-2	BCI-device should transfer data through Bluetooth channel.
HR-3	System should provide a receiver (namely, smartphone) which is equipped with subsystem for receiving Bluetooth signal from BCI-device.
HR-4	The smartphone being used should have permanent internet connection.
HR-5	The system should provide the hub (Raspberry PI computer; min. version is 2) for home automation component.
HR-6	The hub of home automation should have permanent internet connection.
HR-7	The system should provide the controller of home automation which support Z-Wave (or Z-Wave Plus) protocol.
HR-8	The controller of home automation should have USB-interface.
HR-9	System of home automation should have permanent Internet connection.
HR-10	System should provide equipment (namely, LED bulb) for home automation system which is capable to receive commands from the hub being used.
SR – Software requirements	
GDSR – general requirements to data and software	
GDSR-1	BCI EEG device should transfer data through Bluetooth channel in raw binary format (binary packages) according to technical specification of this device.
GDSR-2	The smartphone should run Android operating system (version 6 or higher).
GDSR-3	Raspberry PI computer should run Linux operating system (4.8 core version or later).
GDSR-4	The system should provide mobile application (for Android OS) for interaction between BCI EEG device and other components of the system. This software should be capable of getting data from BCI-device through Bluetooth connection and process them according to requirements (MR) which are given below.
GDSR-5	System should provide access to appropriate real-time service of IoT (Google Firebase) in order to implement live interaction between BCI EEG device and the home automation system.
GDSR-6	The real-time service of IoT being used should be able to store structured data about user's brain activity and other concomitant information in JSON format.
GDSR-7	The real-time service of IoT that is used should be able to immediately notify the system of home automation of changes which have been applied to data.
GDSR-8	System should provide access to server software for handling of requests to the system of home automation (namely, "Domoticz" server).
GDSR-9	Software of home automation hub should handle requests in the form of HTTP RESTful queries and return respond in the form of JSON format.
GDSR-10	System should provide intermediary software which is capable to get data from IoT being used and resend it to the server of home automation system (converting received data to the appropriate HTTP requests).
MR – Mobile software requirement	
MR-1	Mobile application should be run under Google Android operating system.
MR-2	Graphical user interface should comply with Google "Material design" conventions (Google LLC, 2018)
MR-3	All messages in the application's GUI should be in English.
MR-4	The software should check availability of Internet connection prior to launching of its main functionality.
MR-5	In the case of absence of internet connection, the application should show appropriate message and wait until connection will be established.
MR-6	The software should check availability of Bluetooth connection prior to launching of its main functionality.
MR-7	In the case of absence of Bluetooth connection, the application should show appropriate message and wait until connection will be enabled.
MR-8	The software should check availability of NeuroSky MindWave device which is should be preliminary paired with mobile device which is used.

MR-9	In the case of absence of appropriate NeuroSky MindWave device, the application should stop its functioning and show error message to the user.
MR-10	The application should check the state of the detected NeuroSky MindWave device before starting of its main functionality.
MR-11	In the case of erroneous state of the NeuroSky MindWave device (device is off) the application should show appropriate message and wait until the device will be turned on.
MR-12	The application should monitor the state of the NeuroSky MindWave device while functioning. In the case the device becomes unavailable the application should return user to the screen of connection establishing and show the appropriate message.
MR-13	The application should constantly read data from NeuroSky MindWave device and show in the title of the main window.
MR-14	The application should monitor two characteristics of brain activity of the user: level of brain activity (concentration), level of brain relaxation (meditation).
MR-15	The application should provide user with ability to distantly control functioning of LED bulb (section 4.1.3 of the report).
MR-16	The application should provide the user with GUI for choosing of one of three modes of working: Mode 1) control of the LED-bulb by the value of the level of his brain's attention (concentration), Mode 2) control of the LED-bulb by the value of the level of his brain's relaxation (mediation), Mode 3) control of the LED-bulb by the value calculated on the base of combination of two mentioned levels.
MR-17	The application should provide visible neurofeedback by accepting data from the brain of the user (through BCI EEG device being used) and converting it to the commands directed to the home automation server which change behaviour of the bulb in each of modes mentioned in MR-16: Sub-mode (a) state of the smart LED-bulb (on/off) (not for mixed mode), Sub-mode (b) brightness, Sub-mode (c) colour.
MR-18	The application should provide visible neurofeedback by accepting data from the brain of the user (through BCI EEG device being used) and converting it to the demonstrative graphical representation in the main window of the application.
MR-19	**Mode 1, sub-mode a.** Level of user attention should be converted to two possible states of the smart LED lamp: on and off.
MR-20	**Mode 1, sub-mode a.** The smart LED lamp should be turned on (by the application) when the level of attention exceeds maximum threshold value during about 10 consecutive seconds.
MR-21	**Mode 1, sub-mode a.** The smart LED lamp should be turned off (by the application) when the level of attention becomes less then minimum threshold value during about 10 consecutive seconds.
MR-22	**Mode 1, sub-mode a.** Maximum threshold value should be stored in the cloud Google Firebase database (applies for attention and meditation levels).
MR-23	**Mode 1, sub-mode a.** Minimum threshold value should be stored in the cloud Google Firebase database (applies for attention and meditation levels).
MR-24	**Mode 1, sub-mode a.** The application should provide GUI for changing of minimum and maximum threshold values (applies for attention and meditation levels).
MR-25	**Mode 1, sub-mode a.** Changes of maximum threshold should be applied in the cloud database immediately after confirmation from the user (applies for attention and meditation levels).
MR-26	**Mode 1, sub-mode a.** Changes of minimum threshold should be applied in the cloud database immediately after confirmation from the user (applies for attention and meditation levels).
MR-27	**Mode 1, sub-mode a.** Changes of maximum threshold should be applied in the GUI immediately after user confirms them (applies for attention and meditation levels).
MR-28	**Mode 1, sub-mode a.**

	Changes of minimum threshold should be applied in the GUI immediately after confirmation from the user (applies for attention and meditation levels).
MR-29	**Mode 1, sub-mode a.** Changes of maximum threshold should immediately affect the behavior of the smart LED lamp after confirmation from the user was received (applies for attention and meditation levels).
MR-30	**Mode 1, sub-mode a.** Changes of minimum threshold should immediately affect the behavior of the smart LED lamp after confirmation from the user was received (applies for attention and meditation levels).
MR-31	**Mode 1, sub-mode b.** Level of the user's attention should be converted to appropriate level of brightness of white color of the smart LED-lamp (see table 6). The dependence of the level of brightness of the bulb on the level of brain's concentration should be linear (level 0 should be ignored, because indicates improper functioning of the NeuroSky MindWave device (NeuroSky Inc., 2018)).
MR-32	**Mode 1, sub-mode b.** Level of user attention should be correspondingly depicted in the graphical user interface of the application in order to allow the user to comprehend level of his mind activity. This should be the chart with visual meter which changes its position in accordance with the state of level of activity of the user.
MR-33	**Mode 1, sub-mode c.** Level of the user's attention should be converted to appropriate color of the smart LED-lamp (see table 6). Color of the bulb should be in the range form green (minimum level of attention) to red (maximum level of attention).  The dependence of the color of the bulb on the attention level should be linear (level 0 should be ignored, because indicates improper functioning of the NeuroSky MindWave device (NeuroSky Inc., 2018)).
MR-34	**Mode 1, sub-mode c.** Level of user attention should be correspondingly depicted in the graphical user interface of the application in order to allow the user to comprehend level of his mind activity. This should be the rectangle with pointer indicating the current color. Indicator should move along rectangle in accordance with the state of level of brain's activity of the user.
MR-35	**Mode 1.** The app should provide the user with additional graphical interface for training of user's ability to promote his level of attention by concentrating on specially organized animation.
MR-36	**Mode 1.** The app should provide concise information about concentration (attention) and how to develop this state of his mind.
MR-37	**Mode 2, sub-mode a.** Level of user's mind relaxation (meditation) should be converted to two possible states of the smart LED lamp: on and off.
MR-38	**Mode 2, sub-mode a.** The smart LED lamp should be turned on (by the application) when the level of user's mind relaxation exceeds maximum threshold value during about 10 consecutive seconds.
MR-39	**Mode 2, sub-mode a.** The smart LED lamp should be turned off (by the application) when the level of user's mind relaxation becomes less then minimum threshold value during about 10 consecutive seconds.
MR-40	**Mode 2, sub-mode b.** Level of the user's mind relaxation should be converted to appropriate level of brightness of white color of the smart led lamp (see table 6). The dependence of the level of brightness of the bulb on the level of brain's relaxation should be inversely proportional, i.e. the more relaxed the subject's mind, the weaker the brightness (level 0 should be ignored, because indicates improper functioning of the NeuroSky MindWave device (NeuroSky Inc., 2018)).
MR-41	**Mode 2, sub-mode b.** Level of user's mind relaxation should be correspondingly depicted in the graphical user interface of the application in order to allow the user to comprehend level of his mind relaxation (meditation). This should be the rectangular chart with visual meter inside which changes its position in accordance with the state of level of activity of the user.
MR-42	**Mode 2, sub-mode c.**

<table>
<tr><td></td><td>Level of the user’s meditation should be converted to appropriate color of the smart LED-lamp (see table 6). Color of the bulb should be in the range form green (minimum level of mediation) to dark purple (maximum level of meditation).

The dependence of the color of the bulb on the meditation level should be linear (level 0 should be ignored, because indicates improper functioning of the NeuroSky MindWave device (NeuroSky Inc., 2018)).</td></tr>
<tr><td>MR-43</td><td>Mode 2, sub-mode c.
Level of user’s mind meditation should be correspondingly depicted in the graphical user interface of the application in order to allow the user to comprehend level of his mind activity. This should be the rectangle with pointer indicating the current color. Indicator should move along rectangle in accordance with the state of level of brain’s relaxation of the user.</td></tr>
<tr><td>MR-44</td><td>Mode 2.
The app should provide the user with additional graphical interface for training of user’s ability to promote his level of relaxation by concentrating on specially organized animation. It is often recommended to close the eyes to bring up the level of meditation. Considering this circumstance, the app should provide user with audio neurofeedback. During this exercise the app should play the relaxing tune which volume of which depends on the level of user’s brain meditation.</td></tr>
<tr><td>MR-45</td><td>Mode 2.
The app should provide concise information about mind relaxation (meditation) and how to develop this state of his mind.</td></tr>
<tr><td>MR-46</td><td>Mode 3, sub-mode a
Not applicable.</td></tr>
<tr><td>MR-47</td><td>Mode 3, sub-mode b
The app should calculate the difference between the level of attention and the level of relaxation. On the base of this difference brightness of the smart LED lamp should be changed using the following scale:
<table>
<tr><td>Range of value of difference</td><td>Corresponding value of brightness of the led-bulb</td></tr>
<tr><td>0..20</td><td>10</td></tr>
<tr><td>21..40</td><td>31</td></tr>
<tr><td>41..60</td><td>51</td></tr>
<tr><td>61..80</td><td>71</td></tr>
<tr><td>81-100</td><td>90</td></tr>
</table>
Difference is should be calculated if values are overlapped. This means that sum of both values should be equal or greater than 100. The more difference between two levels the brighter the lamp.</td></tr>
<tr><td>MR-48</td><td>Mode 3, sub-mode c
The app should calculate the difference between the level of attention and the level of relaxation.
A) If difference is less or equal to 30, the app should report that user’s mind is in balanced state and change color of the bulb to yellow.
B) If difference is more than 30 and level of attention prevails, the app should report that user’s mind is highly focused and change color of the led-bulb to red.
C) If difference is more than 30 and level of relaxation value prevails, the app should report that user’s mind is significantly relaxed and change color of the led-bulb to purple.</td></tr>
<tr><td colspan="2">SR - Server-side home automation script requirements</td></tr>
<tr><td>SR-1</td><td>Script should be running continuously on the hub of home automation (Raspberry PI).</td></tr>
<tr><td>SR-2</td><td>Script should retrieve notification form Google Firebase about changes in control data of the smart led-bulb and to immediately send them to the “Domoticz”. New data should be sent in the form of HTTP RESTful requests according to API specifications of “Domoticz” server.</td></tr>
<tr><td>SR-3</td><td>The script should not have runtime errors.</td></tr>
<tr><td>SR-4</td><td>The script should be automatically started by operating system of the host in the process of system launching.</td></tr>
<tr><td>SR-5</td><td>The script should keep writing log file about events which were handled. Every event should have information about date and time, type of event, applied data, server response (status of the led-bulb after applying of new data). Only events of two latest weeks should be stored. Earlier events’ records should be erased from the log file at 00:00:00 of every new day.</td></tr>
<tr><td colspan="2">PR – System performance requirements</td></tr>
<tr><td>PR-1</td><td>System should guaranty reaction on events of changing of the user’s EEG data during not more than 1 second (i.e. the smart led-bulb should change its color or brightness immediately after corresponding EEG data was changed).</td></tr>
</table>

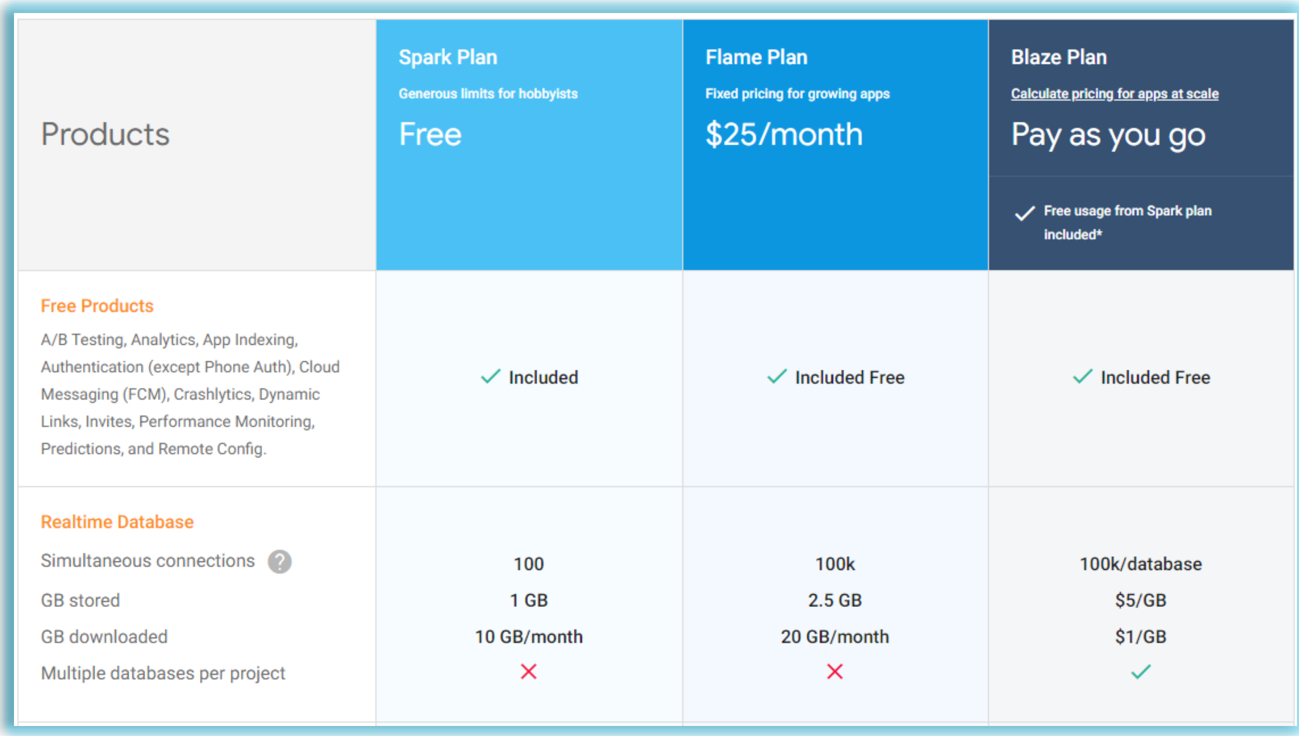
6 SOFTWARE IMPLEMENTATION

6.1 Database technology and schema

According to the aims of the project and requirements (section 5, table 7, GDSR-5,6,7), software being developed should implement reactive online reactions on changes which is constantly happening during period of the software functioning. New data is arriving from EEG BCI device and need to be demonstrated in GUI of mobile application and immediately change the state of smart home device (led-bulb). Furthermore, resending data of BCI should be implemented through internet in order to guarantee wireless connection between all devices of the architecture.

Cloud Google Firebase database was chosen as an DBMS for this project (section 4.1.2). This DBMS supports NoSQL database only. Data of Firebase DB should have JSON text representation. Firebase provides real-time immediate notifications

Concerning the database being developed I need to mention that BCI-device produces significant amount of data. But converting of BCI-device it into the element of IoT infrastructure demands to store its data in order to resend them dynamically to other parts of designed architecture. It was said in section 4.1.2 that volume of data for short period of time exceeded 1GB. In real scenario this circumstance means that storing of each element of data stream in the database can be significantly costly (especially having in mind that this project uses free pricing plan of Firebase which limits the size of downloaded data and size of database (fig. 21)).



Products	Spark Plan Generous limits for hobbyists Free	Flame Plan Fixed pricing for growing apps $25/month	Blaze Plan Calculate pricing for apps at scale Pay as you go Free usage from Spark plan included*
Free Products A/B Testing, Analytics, App Indexing, Authentication (except Phone Auth), Cloud Messaging (FCM), Crashlytics, Dynamic Links, Invites, Performance Monitoring, Predictions, and Remote Config.	✓ Included	✓ Included Free	✓ Included Free
Realtime Database			
Simultaneous connections	100	100k	100k/database
GB stored	1 GB	2.5 GB	$5/GB
GB downloaded	10 GB/month	20 GB/month	$1/GB
Multiple databases per project	✗	✗	✓

Figure 21: pricing plans of Google Firebase services (Google LLC, 2018)

Considering this circumstance there is no vital need to store absolutely each changes of data of user's brain activity. To achieve the goals of the project the database being developed should store the most recent data of user's brain activity delivered by used BCI-device. Also, data which affect state and behavior of the smart led-bulb should be presented in the database. On the base of these considerations the following structure of "MindwaveBCI" database was developed.

"MindwaveBCI" database comply with NoSQL rules and represents data in JSON format according with necessities of the project. Set of data fields is described in the table 8.

Table 8: "MindwaveBCI" database

Field name	Data type	Sense of data
BrainDataToUse	Integer	This field is used to determine what data from BCI to use (0 means applying of level of attention, 1 means applying of level of meditation, 2 means applying of combination of those two levels).
LastAttentionData	Integer	Last value of user's mind attention (concentration).
LastMeditationData	Integer	Last value of user's mind meditation (relaxation).
LastModified	Integer	Last date and time when changes happened.
Hue	Integer	Hue of color of the led-bulb which is calculated on the base of brain data being used (according to the value of BrainDataToUse field).
LightLevel	Integer	Level of brightness to be applied together with Hue.
IsOn	Integer	Last state of the led-bulb.
LightBrightness	Integer	Level of the brightness the led-bulb which is calculated on the base of brain data being used (according to the value of BrainDataToUse field).
ParamToApply	Integer	Value of this field is used to determine what characteristics should be changed after new data of user's brain activity arrived ("-1 means that no action is needed, 0 means turning on/off, 1 corresponds to the level of brightness, 2 corresponds to hue and brightness).
State	Integer	Code of state of the led-bulb which this device returns after receiving of some command from home automation hub (0 – error, 1 – ok).
offThreshold	Integer	Level of attention of meditation which is used to turn off the led-bulb.
onThreshold	Integer	Level of attention of meditation which is used to turn on the led-bulb.

As was mentioned above, "MindwaveBCI" database has JSON format. A snapshot of database during its exploitation is shown by figure 22. Snapshot of database in the GUI of Google Firebase console can be seen on figure 23.


```
{
    "BrainDataToUse": 1,
    "Hue": 215,
    "IsOn": 1,
    "LastAttentionData": 60,
    "LastMeditationData": 48,
    "LastModified": 274000000,
    "LightBrightness": 51,
    "LightLevel": 100,
    "ParamToApply": 2,
    "State": 1,
    "offThreshold": 40,
    "onThreshold": 67
}
```


Figure 22: "MindwaveBCI" database snapshot in the form of raw JSON format

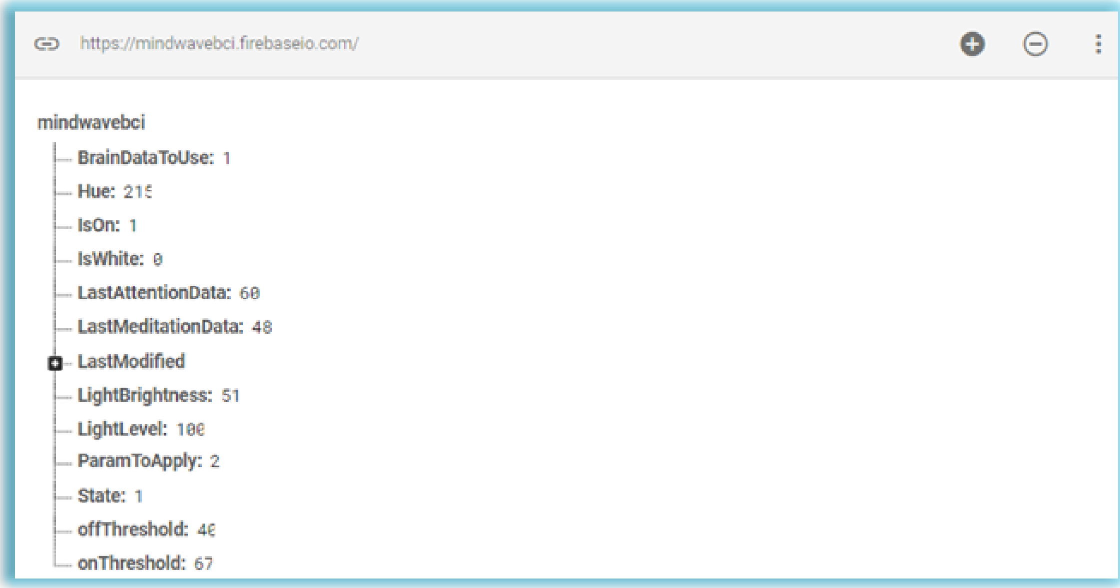


Figure 23: ““MindwaveBCI” database in the Google Firebase console Web-GUI

6.2 Mobile software technologies

This project mobile application as one of the components of its architecture (fig. 19). According to requirements (GDSR-4) this should be Android application. Structure of the application and its functions also ensues from requirements which are formulated in table 7 (section MR). Below are discussions of what technologies were used to meet the requirements of the project.

6.2.1 Programming language

Kotlin programming language (Samuel & Bocutiu, 2017) was used to create mobile software. Reason for choosing of this language lays in the personal educational interest of the author of the project. Nowadays there some possible options of programming languages for Android development which are officially supported by Android (C++, Java, Kotlin). Kotlin has full access to all the built-in features of Android SDK and to Android Java Virtual Machine as well. So, this a fully functional instrument for Android development which is officially supported by Google as a new programming language which is independent from Java which is possessed by Oracle Corp.

Kotlin is an object-oriented general-purpose technology (Samuel & Bocutiu, 2017). Kotlin includes programming language and set of underlying libraries and instruments that can provide a programmer with some interesting possibilities. First of all, it should be mentioned that Kotlin allows to build executables files of several types. These types cover formidable area of possible necessities of software developers. Kotlin has compilers for creation of applications for JVM (in this case compiler creates JVM-compatible byte code) and for creation of applications in native binary executable formats for major operating systems (Windows, Linux, MacOS). Kotlin is officially supported language for Android development since 2017, meanwhile the language was created by JetBrains company in 2011.

6.2.2 Bluetooth connectivity

One of the most important and primary tasks of software being created is to get data of user's brain EEG activity which is supplied by "NeuroSky MindWave headband" (EEG BCI-device). BCI-device translates data using Bluetooth low energy protocol. So, establishing of Bluetooth connection is vital task for the software.

Usage of Bluetooth connection can be implemented by performing of some rather standardized steps. First of all, manifest of project should be customized by inserting two xml-tags (fig. 24).

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

Figure 24: xml-tags for enabling of Bluetooth connections within the app

"BluetoothAdapter" class of standard Android SDK is used to establish connections over Bluetooth channel. Following code snippet shows how the application establishes Bluetooth connection.

```
fun startConnection(): Pair<Int, String> {
    try {
        bluetoothSocket = BluetoothAdapter.getDefaultAdapter()

        var reason: Int = 0
        var message: String = ""

        if (!isInet) reason += 1
        if (bluetoothSocket == null || !bluetoothSocket!!.isEnabled()) reason += 2

        message = when (reason) {
            1 -> res.getString(R.string.no_inet)
            2 -> res.getString(R.string.no_blue_tooth)
            3 -> res.getString(R.string.no_blue_tooth_inet)
            else -> ""
        }

        if (reason > 0) return Pair(0, message)
    }
    catch (ex: Exception) {
        return Pair(1, ex.toString() + "Impossible to establish connection to device. " +
                "Try to check your headset and restart the app.")
    }

    return Pair(2, "Connecting the NeuroSky MindWave device ... ")
}
```

Figure 25: code snippet for establishing of Bluetooth connection

Main methods for solving this task are getDefaultAdapter() and isEnabled().

After establishing of connection, the app needs to find and connect concrete preliminary paired BCI-device. It can be done by iterating through the full list of existing paired devices (which is returned by calling of bindedDevices) and comparing their system names with string "mindwave" which is omnipresent part of the name of any NeuroSky MindWave Mobile device.

6.2.3 Firebase database connectivity

Connection with Google Firebase database can be programmed by using object model provided by Firebase SDK which has versions for major programming languages and operating systems (Google LLC., 2018).

Using of this API in Android application is possible through performing of some standard steps.

1. Registration of mobile application in Firebase console (in the project where database is stored). Registration implies getting of specially calculated hash SHA code (can be calculated by CLI, provided by Google in Android Studio).
2. Getting of specially formed JSON-file ("google-services.json") which will be used by Gradle system in Android Studio for building the app. JSON-file is created online by Firebase registration tool and is downloaded to the folder of the project. Part of this file is shown in figure 26.
3. Gradle scripts of the project should be updated by inserting instructions for downloading of Firebox libraries into the project (figure 27).

```
{
  "project_info": {
    "project_number": "870956849009",
    "firebase_url": "https://mindwavebci.firebaseio.com",
    "project_id": "mindwavebci",
    "storage_bucket": "mindwavebci.appspot.com"
  },
  "client": [
    {
      "client_info": {
        "mobilesdk_app_id": "1:870956849009:android:0b36e9826dc2852f",
        "android_client_info": {
          "package_name": "sobolev.bciot_getdata"
        }
      },
```

Figure 26: Firebase connection specification file

```
    implementation 'com.google.firebase:firebase-database:16.0.4'
    implementation 'com.google.firebase:firebase-core:16.0.4'
}

apply plugin: 'com.google.gms.google-services'
```

Figure 27: instructions for enabling of usage of Firebase libraries

The main class for programming Firebase in the project is "FirebaseDatabase" from package "com.google.firebase.database".

1. `var bciMindWaveDBReference = FirebaseDatabase.getInstance().getReference()`

allows to create active connection with Firbase database which is specified in the Firebase specification file.

2. **override fun** onDataChange(dataSnapshot: DataSnapshot) { ...

   allows to get immediate notification from remote Firebase server in the case when data is changed (also this call is used to read data from database).

3. **DataSnapshot** class represents the immediate copy of data from Firebase database.

4. **bciMindWaveDBReference**.child(**"offThreshold"**).setValue(value)

   this call allows to write new data directly to the remote Firebase database (it is needed to point out the name of the data field and its value).

6.2.4 NeroSky MindWave headset interaction

Programming of getting data from NeuroSky MindWave headset is implemented by using of API which is provided by the manufacturer of this device (NeuroSky Inc., 2018). The manufacturer provides developers with several versions of their API for different programming languages (C++, Java, C#). This API is a set of classes which provide third party apps with ability to read data provided by the headset. This API is free and implements the main goal of the device usage in third party apps (namely it allows to ger real-time data from user's brain activity in the form of attention level, mediation level, frequencies of alpha waves, beta waves, delta waves, theta waves).

Appropriate copy of API libraries for aimed operating platform and programming language can be obtained from NeuroSky site in their app store (acquisition is free). After downloading and unzipping these libraries should be copied to the directory of the project and imported by standard tools of Android studio.

The app which need to use classes of NeuroSky API should import its standard packages.

Table 9: packages of NeuroSky API "com.neurosky.connection"

ConnectionStates	Module contains declarations of possible states of the device during the process of its functioning.
DataType.MindDataType	Module contains definitions of possible types of signals which can be generated by the device during the process of its functioning.
EEGPower	Module contains of data structures for storing of data generated by the device during the process of its functioning.

TgStreamHandler	This module contains definition of Java interface which should be implemented in the app in order to handle events coming from the device during the process of its functioning.
TgStreamReader	This module contains definition of the class which is used to read data from the device while its functioning.

The principal rules of this API libraries are the following:

1. Usage of classes and interfaces of the API in the app is possible only if Bluetooth connection was established successfully.
2. The app must implement **TgStreamHandler** interface which should be used to interact with device. Namely the primary goal of this implementation is to send commands to the device and get device's status which is important in real use cases. For example, if battery of the device is low then Bluetooth signal disappears, and the app should react accordingly because in other case it can misinform the user and provide him with false data. Example of such implementation given on figure 28. Methods of the interface is provided (by Bluetooth connection) with respond codes which signal about concrete state of the device.
3. The app must create instance of standard **Handler** JVM-class which allows to receive and handle custom (non-system) events generated by the device while the period of its functioning. This object directly receives codes of device's events (code snippet of figure 29) and executes specific operations with data which accompany this concrete event. Inside methods of this object (instance of Handler class is meant) it is possible to use instance of **TgStreamReader** class to get real data from the device.

```
ConnectionStates.STATE_DISCONNECTED -> {
    connectionMessage = "Device is disconnected. Turn headset on to continue and..."
    connectionStatus = ConnectionStates.STATE_DISCONNECTED
    BrainWaveData.IsErrorState = true
    setChanged()
    notifyObservers()
}
ConnectionStates.STATE_ERROR -> {
    connectionMessage = "Connection error was encountered. Touch the image above..."
    connectionStatus = ConnectionStates.STATE_ERROR
    BrainWaveData.IsErrorState = true
    setChanged()
    notifyObservers()
}
ConnectionStates.STATE_FAILED -> {
    connectionMessage = "Your headset is off. Put on the headset properly and tu..."
    connectionStatus = ConnectionStates.STATE_FAILED
    BrainWaveData.IsErrorState = true
    setChanged()
    notifyObservers()
}
```

Figure 28: example of device's events handling in the mobile app

```
MindDataType.CODE_MEDITATION -> {
    BrainWaveData.LastMeditationData = msg.arg1
}

MindDataType.CODE_ATTENTION -> {
    BrainWaveData.LastAttentionData = msg.arg1
}

MindDataType.CODE_EEGPOWER -> {
    val power = msg.obj as EEGPower
    if (power.isValidate) {
        BrainWaveData.WaveDelta = (power.delta.toDouble())
        BrainWaveData.WaveTheta = (power.theta.toDouble())
        BrainWaveData.WaveLowAlpha = (power.lowAlpha.toDouble())
        BrainWaveData.WaveHighAlpha = (power.highAlpha.toDouble())
        BrainWaveData.WaveLowBeta = (power.lowBeta.toDouble())
        BrainWaveData.WaveHighBeta = (power.highBeta.toDouble())
        BrainWaveData.WaveLowGamma = (power.lowGamma.toDouble())
        BrainWaveData.WaveHighGamma = (power.middleGamma.toDouble())
    }
}
```

Figure 29: example of device's events handling in the mobile app

6.3 Server-side software technologies

Saying "server side" I mean software which used to provide functionality of home automation hub (server). Home automation system is a Raspberry PI computer which executes special software ("Domoticz" server) for controlling of devices connected to the Z-Wave mesh network. In this project I need to handle data from BCI-device to the node of Z-Wave network (smart led-bulb) (figure 30).

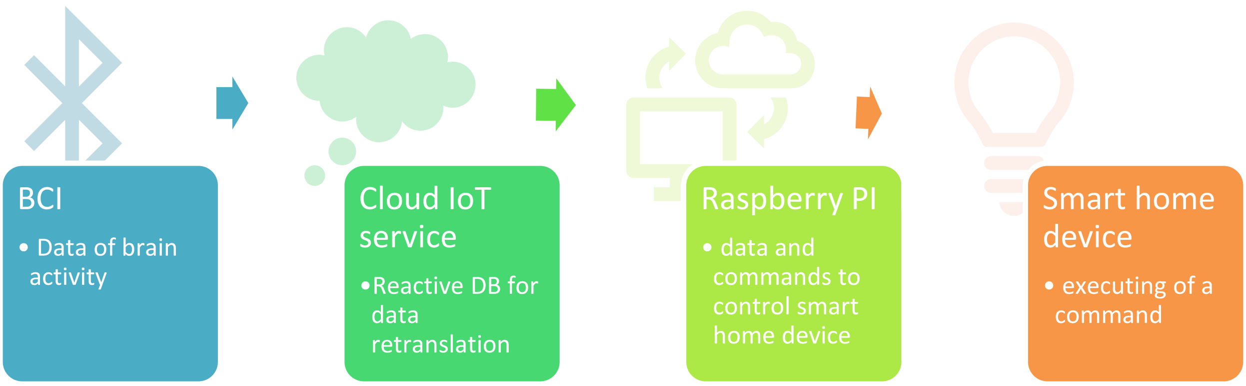


Figure 30: data translation in the system being developed

According to overall architecture (section 4.2), it is needed to develop software which can provide interaction between Google Firebase and "Domoticz". This software is needed because "Domoticz" does not have built-in functionality to receive notifications from Google Firebase. But "Domoticz" is functioning as web-server which provides open REST API, which allows to send commands with data to desired smart device. Sender must know inner ID of device or should inform "Domoticz" what type of device needed to be controlled. "Domoticz" supports vast set of smart devices and almost all of the standard home automation protocols (major of them are ZigBee and Z-Wave). After receiving of commands from the client this server converts them into the binary form which corresponds of rules of the protocol which is used by some concrete device. After installing of “Domoticz” this program starts automatically along with operating system.

The system being created needs to control smart led-bulb. Smart bulbs are controlled by the “Domoticz” by the special subset of commands in REST format, which is described in appropriate manual (Domoticz team, 2018). Commands which can be used to control bulbs are shown in the table 10. All commands are sent in the form of HTTP-requests to the specified HTTP address (which can be assigned during the process of installation). General format is presented below (Domoticz team, 2018) (username and password are optional and can be omitted if not used).

```
http://<username:password@>domoticz-ip<:port>/json.htm?api-call
```

- <username:password@> = the username and password to access Domoticz, this is optional.
- domoticz-ip = the IP-address or hostname of your Domoticz installation.
- <:port> = the port number of your Domoticz installation, this is optional.

Figure 31: format of REST command for Domoticz server

For example:

http://192.168.0.198:8080/json.htm?**type=command**&**param=switchlight**&idx=2&**switchcmd=On**

where idx – inner system identifier of a device which is registered in “Domoticz” service. Address of http server also can be replaced by “localhost” or “127.0.0.1” if used from inside the system where it was installed.

Table 10: REST API of “Domoticz” server for controlling of led-bulbs (Domoticz team, 2018)

Operation	**Command**
Turn a light on	?type=**command**&param=**switchlight**&idx=**99**&switchcmd=**On**
Turn a light off	?type=**command**&param=**switchlight**&idx=**99**&switchcmd=**Off**
Set a light to a certain color (hue and brightness are controlled)	?type=**command**&param=**setcolbrightnessvalue**&idx=**99**&hue=**274**&brightness=**40** &iswhite=**false**
Set a light brightness to a certain value (intensity of white color is controlled)	?type=**command**&param=**setcolbrightnessvalue**&idx=**99**&brightness=**100**&iswhite=**true**

Connection between Google Firebase and “Domoticz” was implemented through intermediary software which was written using Java Script programming language. Created script is dedicated for running under Node.js executive environment. This technology was chosen because of strong functionality presented in Node.js which allows to effectively solve given task. Java Script and Node.js were created specifically to support implementation of internet communication between clients (usually Web browsers) and Web servers. Task of this project is exactly about that type of communication. So, using of this pair Java Script and

Node.js can be justified by that fact that they have everything needed for the task. Furthermore, Google provides Node.js API for Firebase connectivity programming. Usage of Firebase API for Node.js demands preliminary installation of packages. This can be done as follows.

In the CLI of targeted operating system (in this project Debian Linux for Raspberry PI was used; but Node.sj is cross platform technology and commands are the same for any operating system) the following command should be executed:

sudo npm install -g firebase-tools

After installing of Firebase tools for Node.js it is possible to address installed library in the scripts by using of the following command:

```
import { initializeApp, database } from "firebaseLink";
```

Figure 32: import of Firebase tools in Java Script program

Connection to Firebase must be configured before establishing of connection by creating of Java Script object as follows:

```
// Initialize firebaseLink
var config = {
    apiKey: "AIzaSyDTXNPch8lU4M4bu-uzhZqMr3OMQCfm6Q8",
    //authDomain: "<PROJECT_ID>.firebaseLinkapp.com",
    databaseURL: "https://mindwavebci.firebaseLinkio.com",
    storageBucket: "mindwavebci.appspot.com",
};
initializeApp(config);
```

Figure 33: configuring of database connection

Value of apiKey field is calculated by Android Studio CLI command "keytool". Manipulation with data in Firebase data storage being used can be through getting a link to the root element of JSON data object which represent data in NoSQL database.

```
const newData = database().ref('/');
```

Figure 34: retrieving of root element of Firebase database

```
// new data arrival (when any part of realtime database is changed)
newData.on('value', (snapshot) =>
{
    let paramToApply = parseInt(snapshot.val().ParamToApply);
```

Figure 35: handling of event of new data arrival (snapshot is a part of JSON data)

6.4 Reflection

This project gave me excellent possibility to learn a lot of new technologies in the area of software development. Here I can name the most interesting things I managed to learn (or improve my knowledge):

1. Programming of Bluetooth connection and handling of real-time streams of data which are retrieved from some “physical” device (EEG BCI data is meant).
2. Learning of Kotlin programming language for mobile development. I studied new (for me) features of the language (local classes and objects; lambdas; null-reference safe operations; control flow statements, which can be used on the right side of assignment operator; syntactical structure of classes’ elements; etc.). Kotlin demonstrated syntactic flexibility and clearness of the source code which contributed solving of the tasks of this project.
3. Designing and usage of NoSQL database. This is my first practical experience in the area of usage of NoSQL database. This type of database was suitable for this task because structure of data is not very complicated and does not demand creation of vast network of relations. Also working with real-time Firebase Especially was helpful. This DBMS allows immediately resend any changes to all interested clients, which allowed to implement necessities of IoT infrastructure.
4. Learning of home automation was especially exciting. Before this project I had only some general idea of this phenomenon and I always wanted to “dive” deeper. This project gave me such an opportunity. Nowadays smart home technology is controversial and not very well standardized area, which is not typical for modern information technologies. But usage of home automation systems relies of standard and well-developed technologies which simplifies programming of smart home servers and devices.
5. Overall understanding of “philosophy” of internet of things. This allowed me to design my own architecture of such system and implement its software components. Internet of things is booming technology. Understanding of this phenomenon and practical skills in its implementation could be useful for self-promotion on the labor market.

7 SYSTEM DESIGN

7.1 Overall system behavior

Overall system design was discussed in section 4.2. For further understanding of the system functioning it is important to describe sequence of actions and activities during the process of its functioning. Standard UML activity diagram is used to describe behavior of the system being designed.

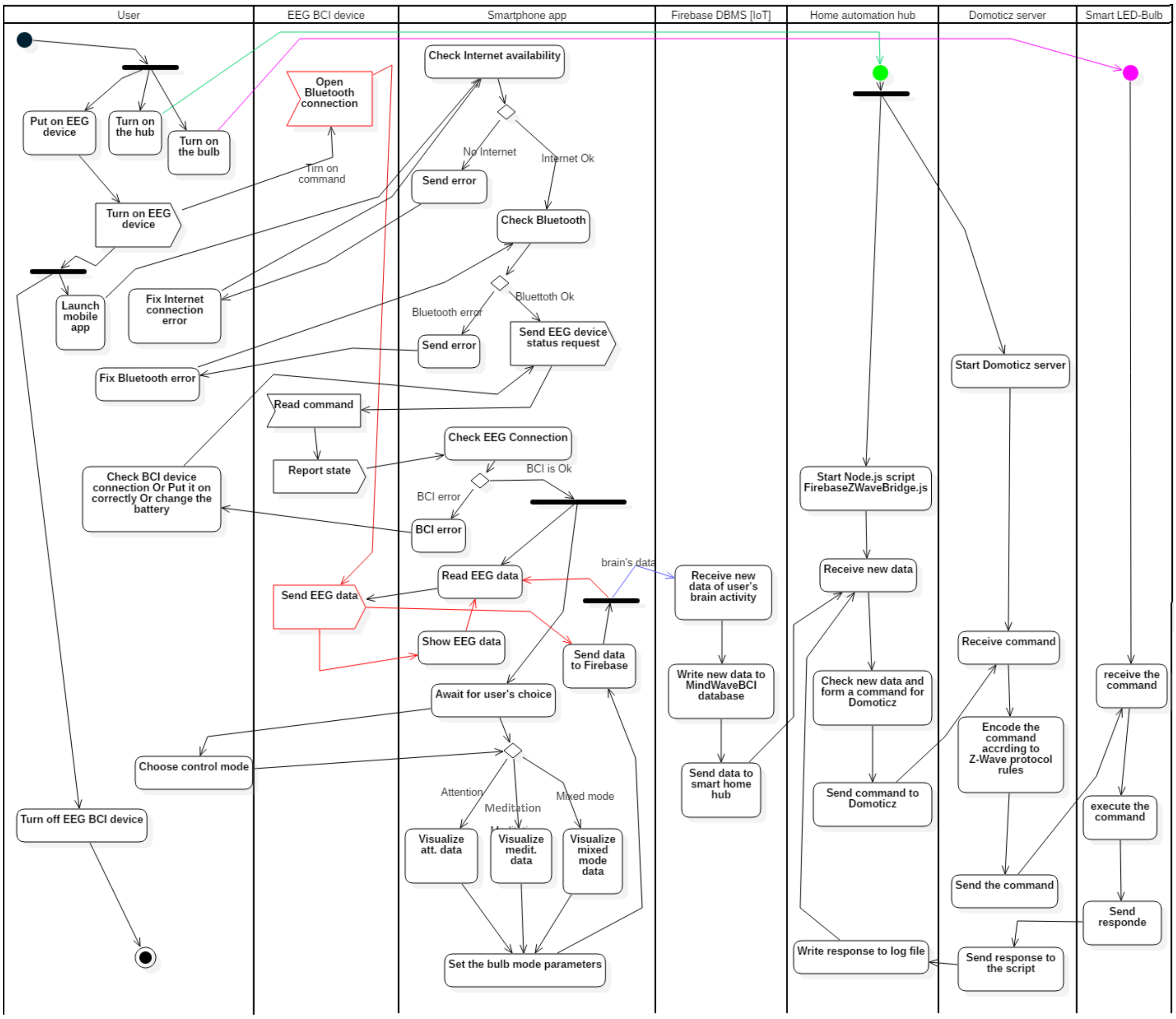


Figure 36: Activity diagram of overall system functioning

7.2 Software design

Mobile app “Control with brain wave” is one of the crucial parts of the system being developed. Structure of the software includes packages, classes and objects (in the sense of Kotlin programming language object is a static class), and enumerations.

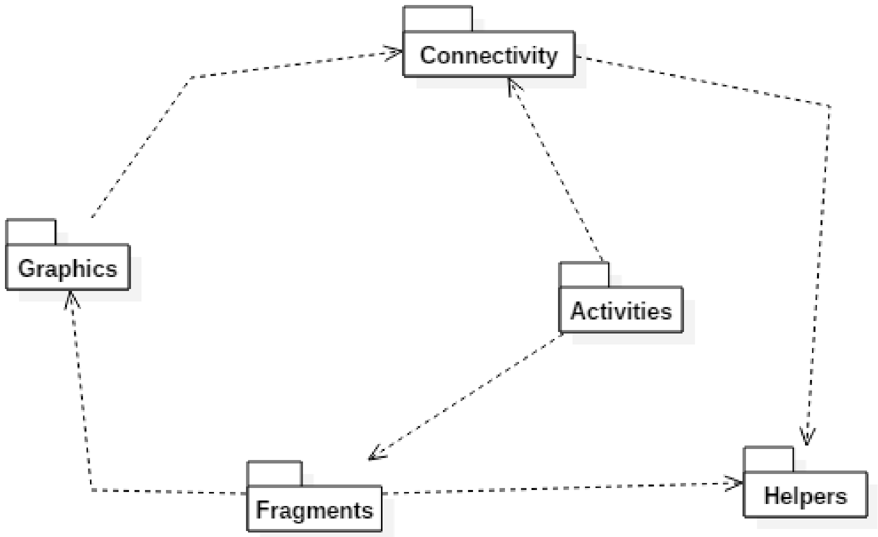
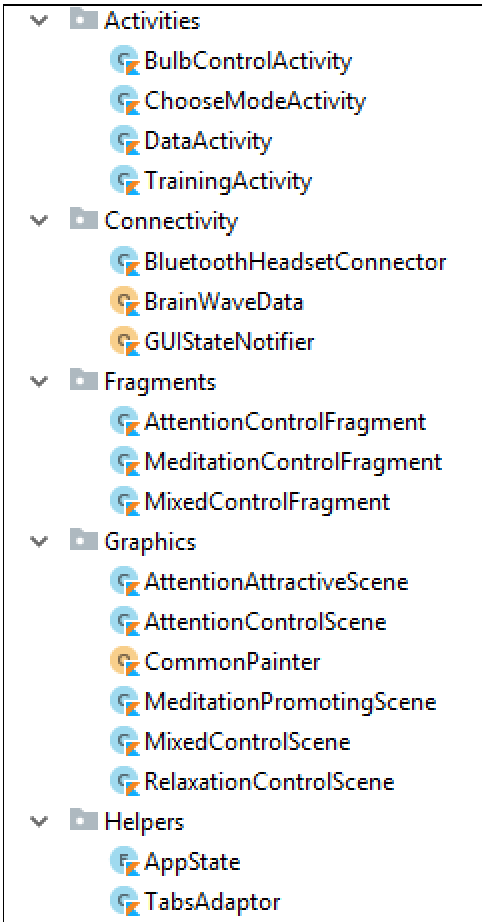


Figure 37: UML package diagram of the software

Table 11: description of software's packages



Package name	**Package description**
Connectivity	This package contains classes which provide the app with Bluetooth connection (to EEG BCI device) and Firebase DBMS connection.
Activities	This package contains classes of the app's GUI Android activities.
Fragments	This package contains classes of GUI fragments.
Graphics	Classes of this package provide graphical rendering on the surface of the fragments (they depict graphical representation of user's brain data).
Helpers	Classes of the package provide ancillary functionality for other classes of the project.

Relationships between classes are shown on the standard UML class diagram (figure 38).

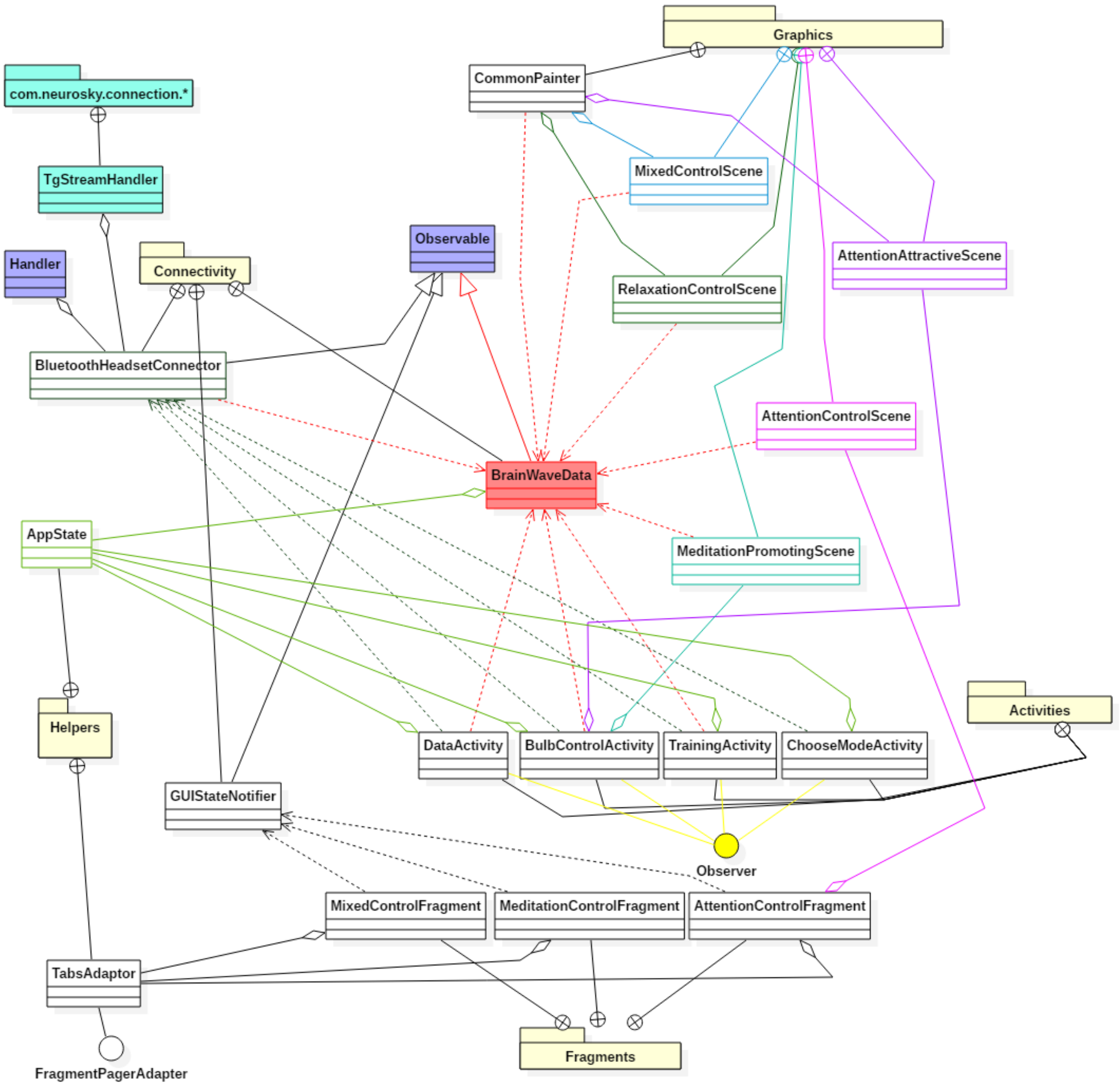


Figure 38: software class diagram

7.3 GUI design

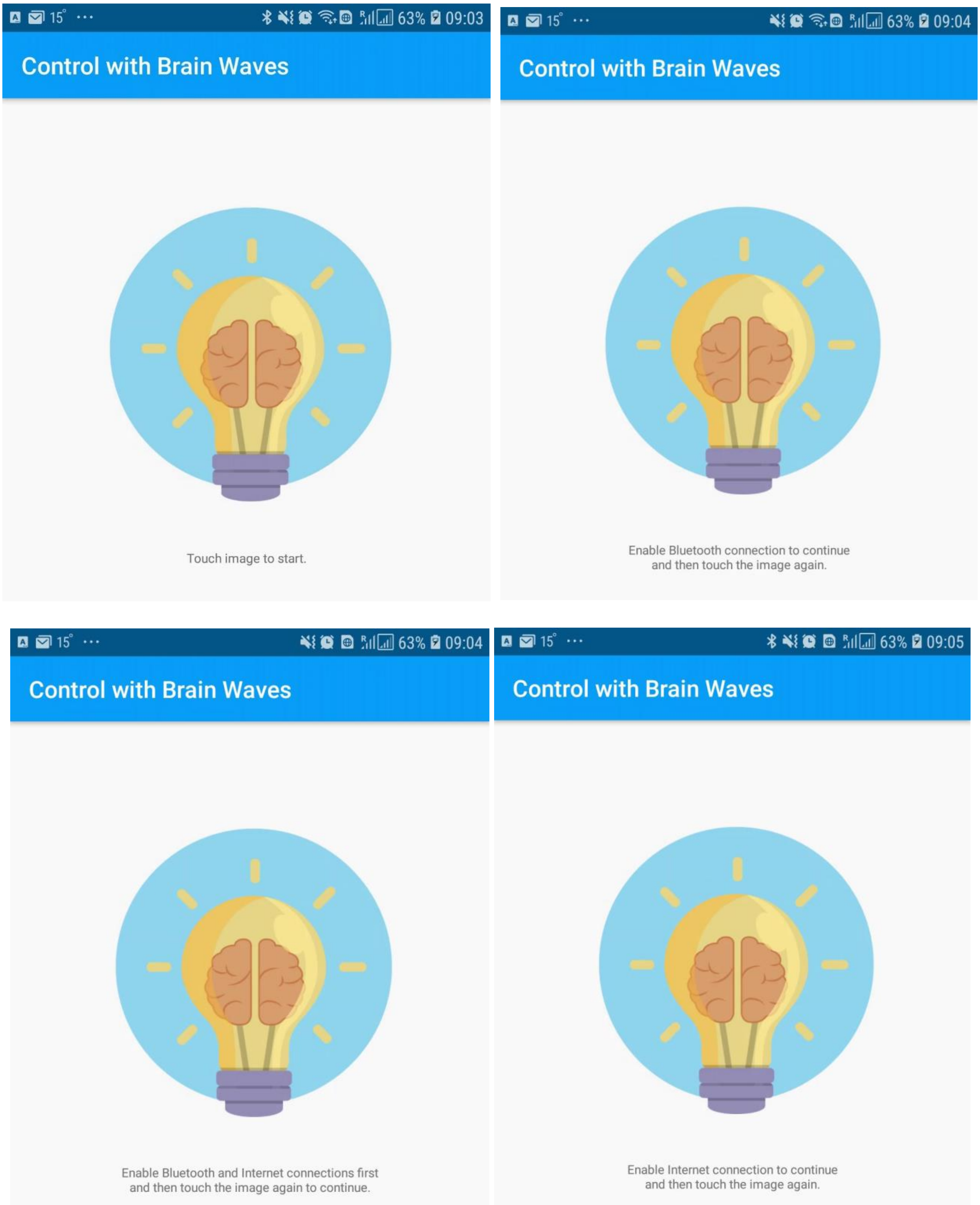


Figure 39: messages during the process of app launching

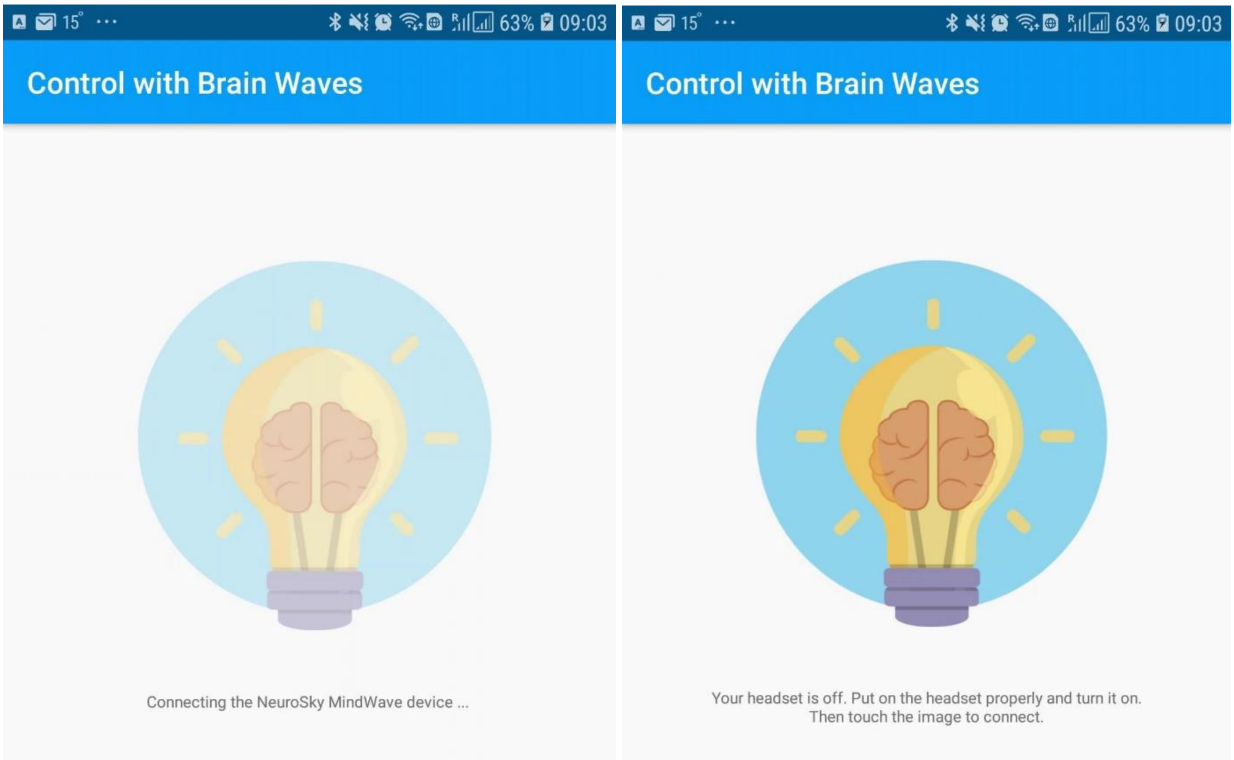


Figure 40: messages during the process of app launching

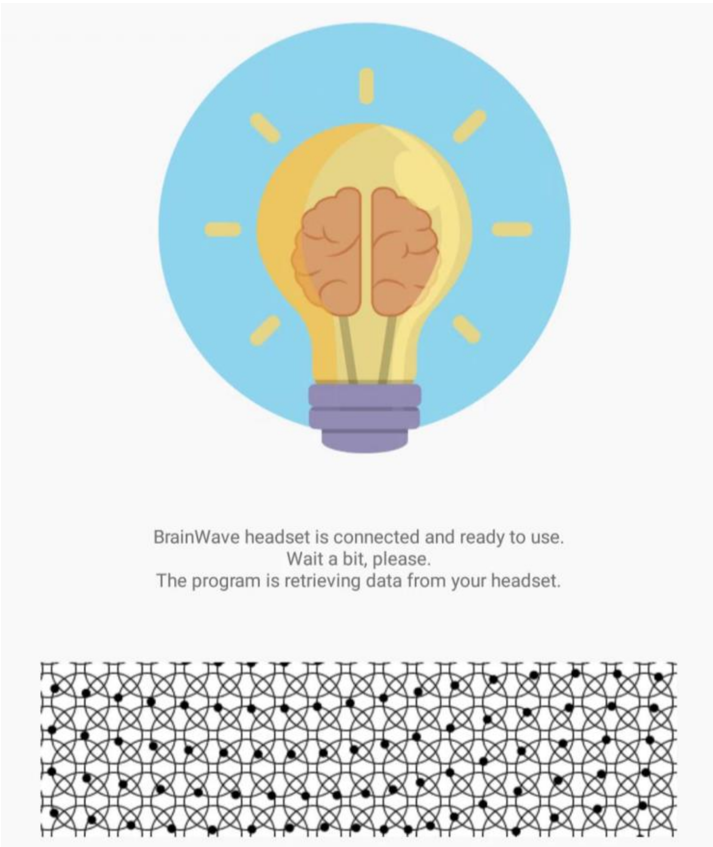


Figure 41: connecting to NeuroSky MindWave Headband GUI

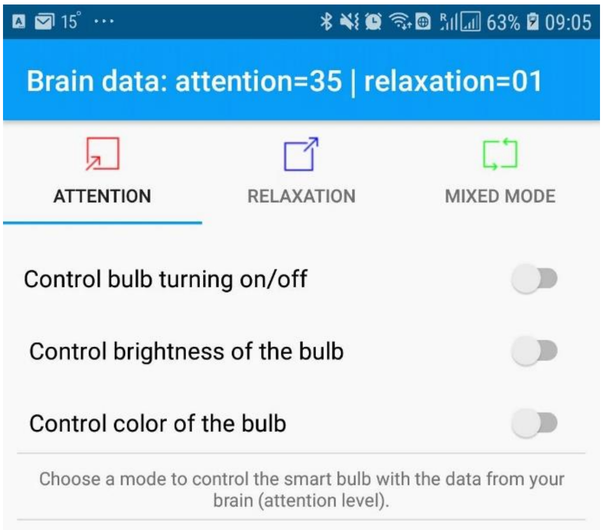


Figure 42: attention mode GUI

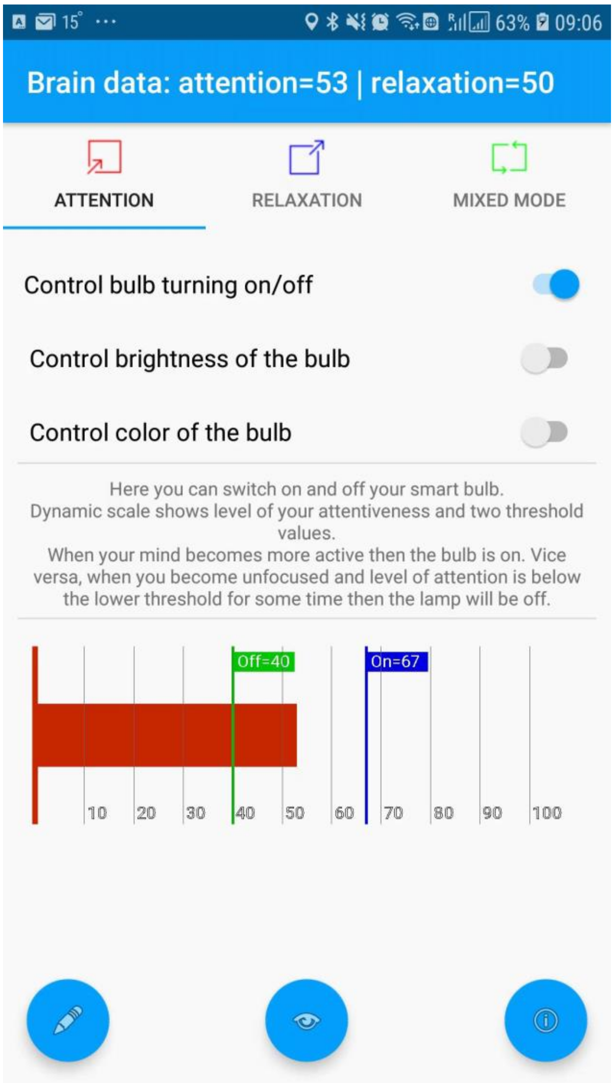


Figure 43: attention mode GUI

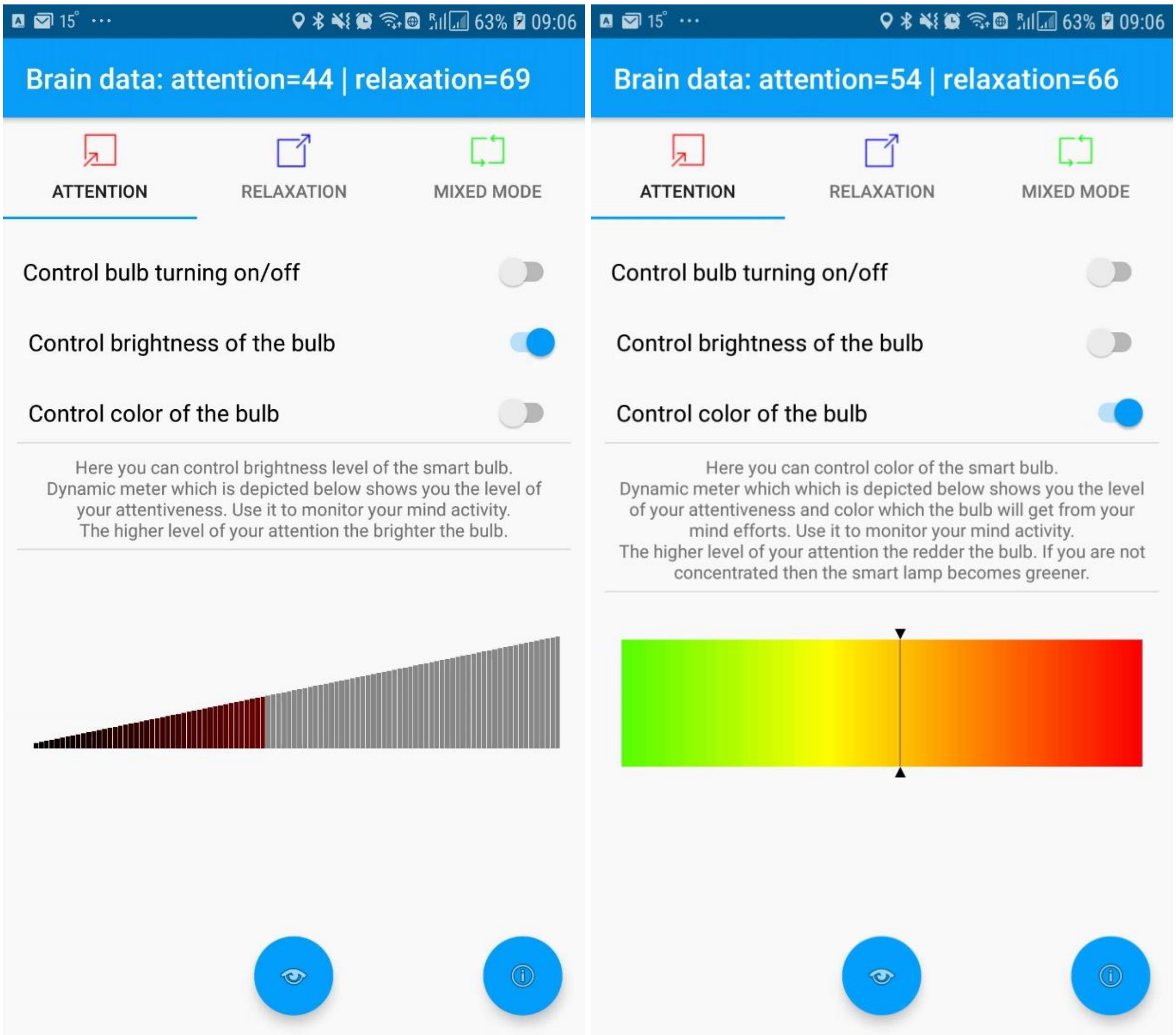


Figure 44: attention mode GUI

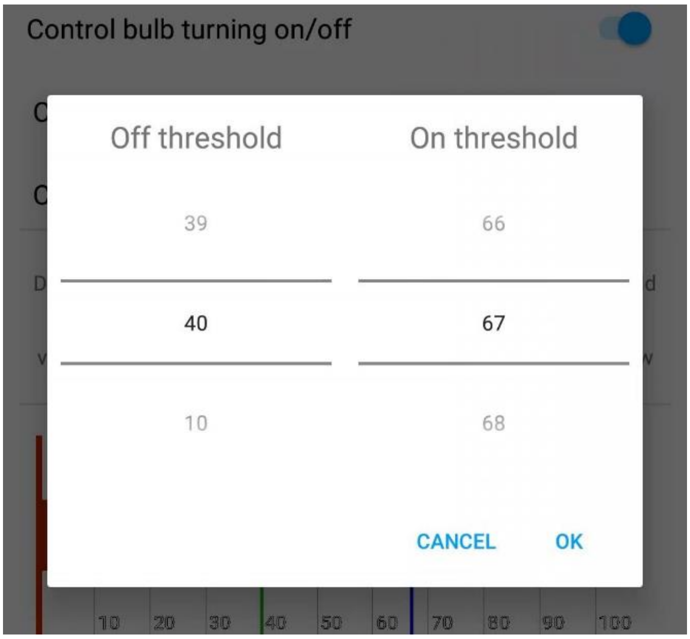


Figure 45: thresholds editor window

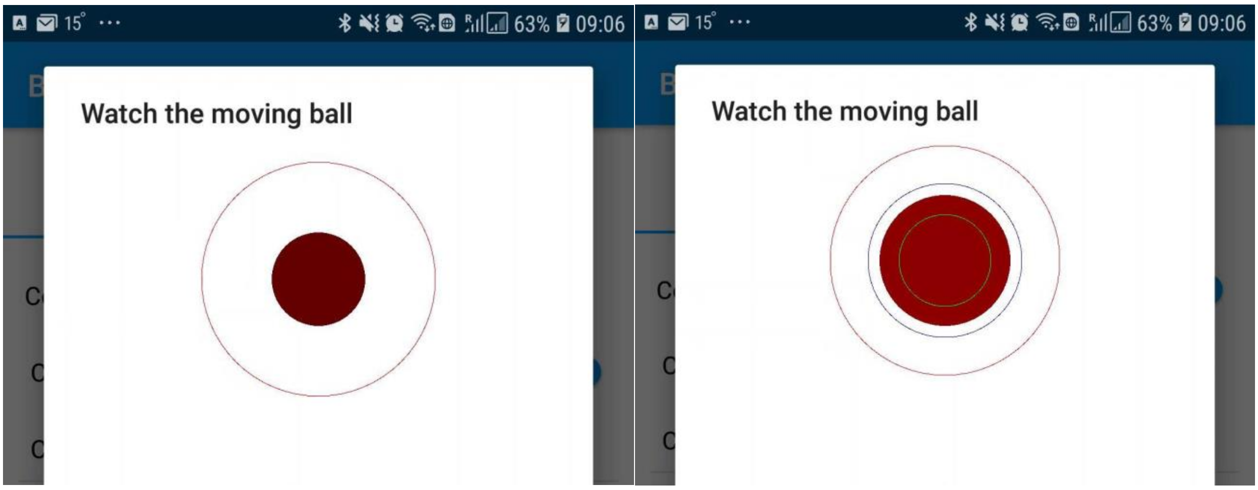


Figure 46: attention training windows

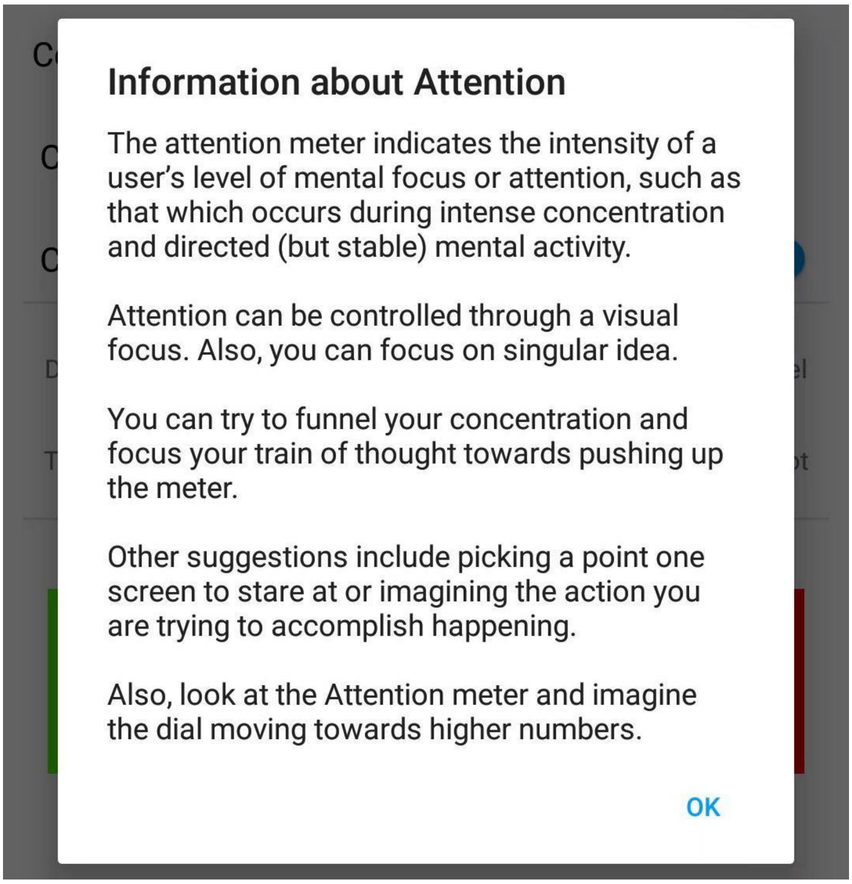


Figure 47: attention help window

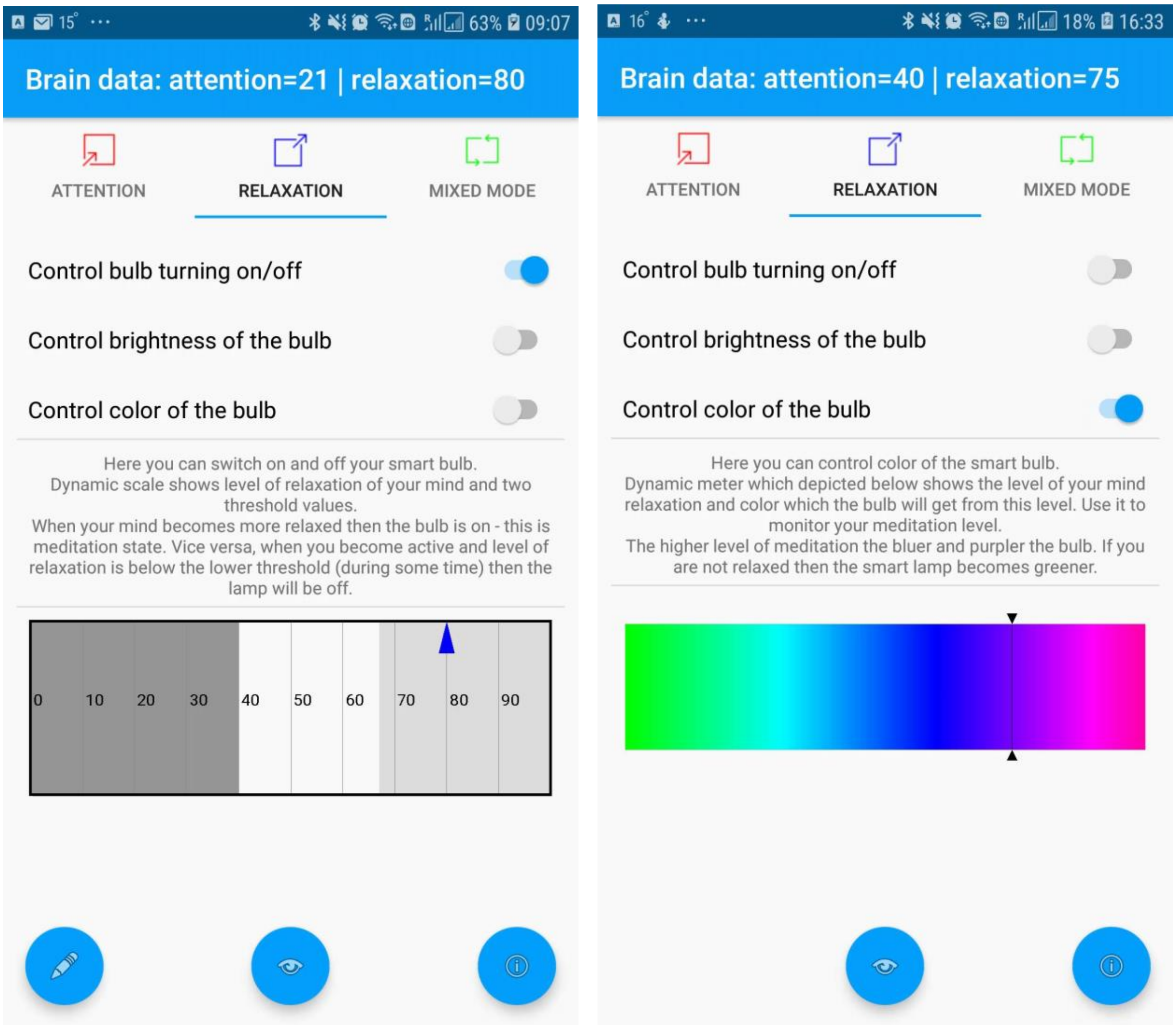


Figure 48: meditation mode GUI

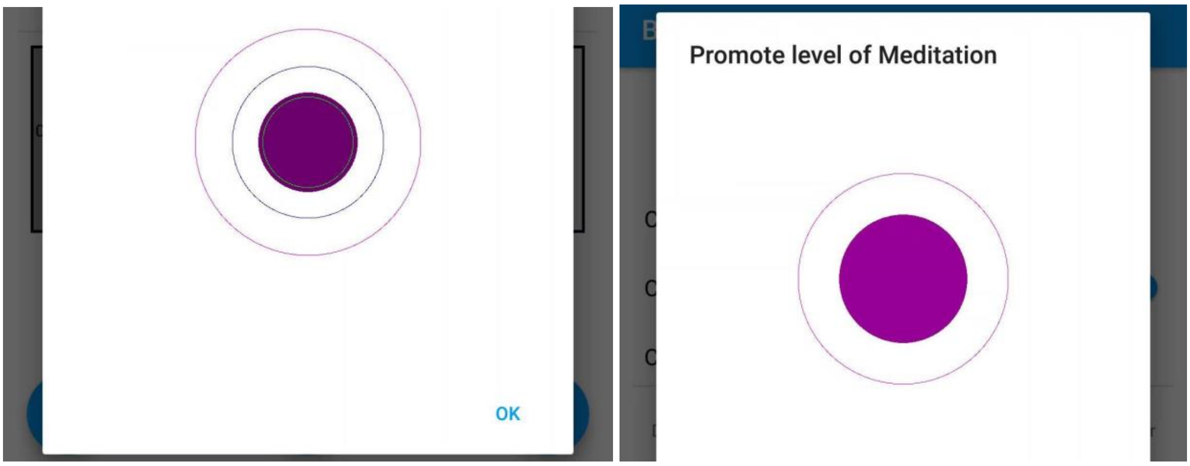


Figure 49: meditation mode GUI

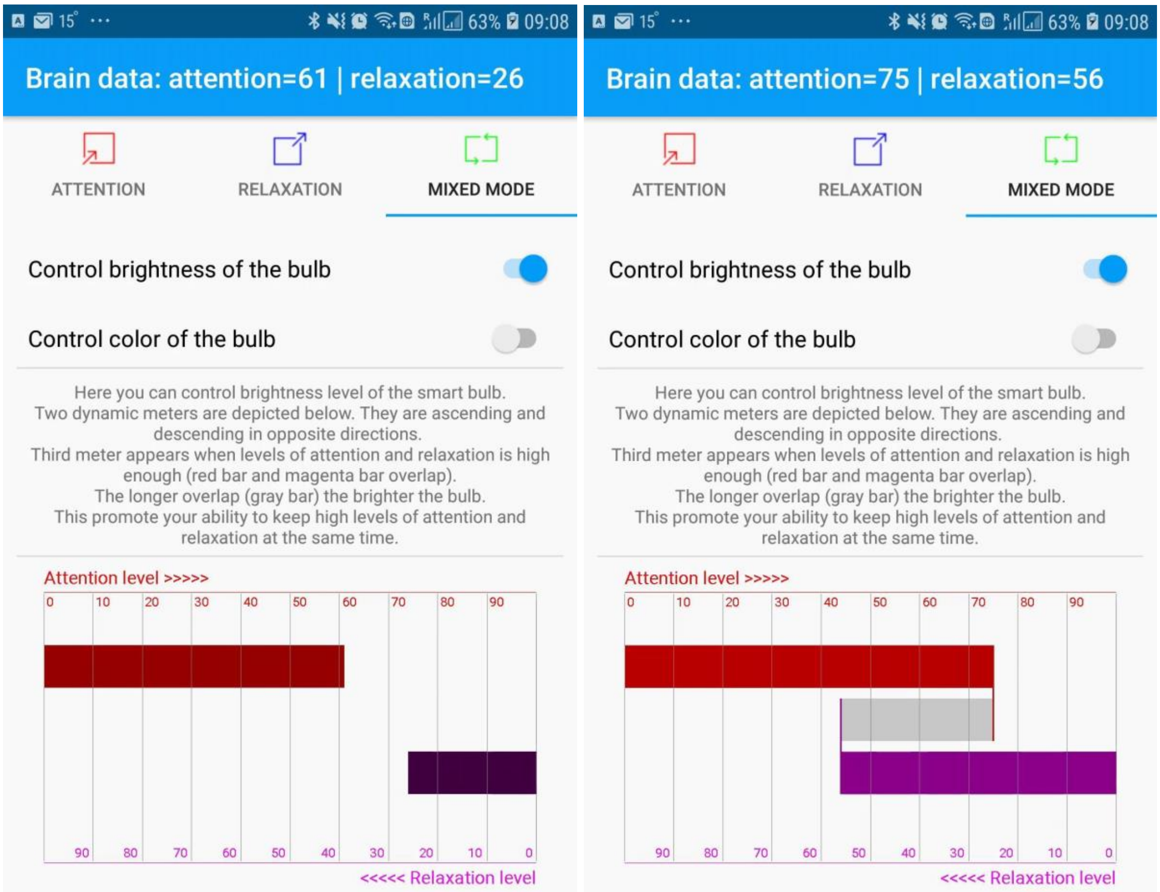


Figure 50: mixed mode GUI

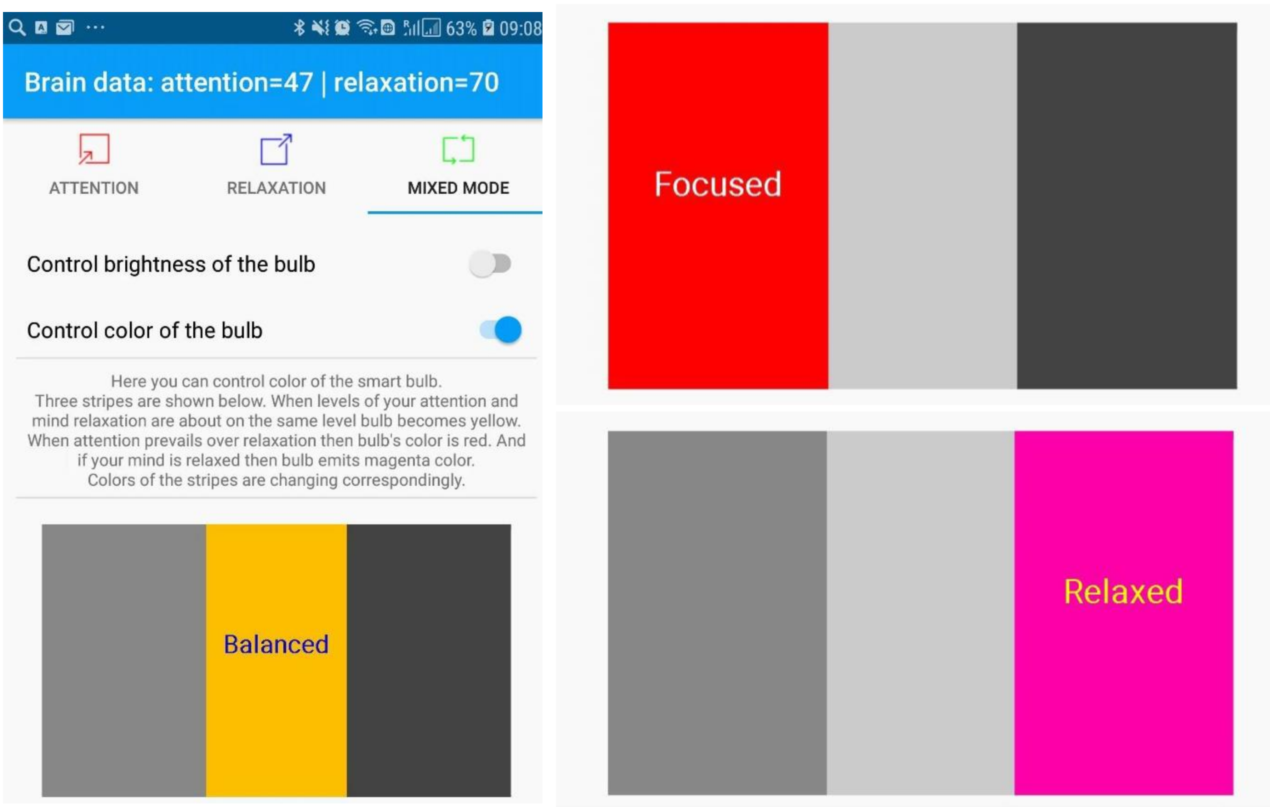


Figure 51: mixed mode GUI

8 INSTRUCTIONS ON USAGE OF THE SYSTEM

According to designed architecture is it needed to undertake some activities in order to use the system.

1. Needed equipment (full list of requirements is given in table 7).
    a. Smartphone working under Android operating system (version of OS is 6 or later).
    b. NeuroSky MindWave Mobile 2 device;
    c. Smart home hub – Raspberry PI computer (version 3) or later;
    d. USB Z-Wave dongle (with working frequency which is allowed by the regulative law of New Zealand – namely 921.42 MHz). In other countries the frequency may be different;
    e. USB Z-Wave dongle should be connected to the Raspberry PI.
    f. Z-Wave (or Z-Wave Plus) LED-Bulb of any formfactor (E-27 and so on).
2. Needed software.
    a. Mobile application “Control with brain waves” which should be installed on the used smartphone.
    b. “Domoticz” home automation server should be installed and run on the hub. Z-Wave and the smart bulb should be registered in the “Domoticz” GUI.
    c. Node.js executive system should be installed on Raspberry PI. Node.js should be configured for usage of Google Firebase.
    d. Script “StartFBZWBridge.js” should be configured to autorun on Raspberry PI.

Short preparation’s instructions.

1. Install the mobile application “Control with brain waves” on your smartphone.
2. Turn on the smart led-bulb according to the instructions from manufacturers and register the bulb in mesh network of the Z-Wave controller (according the instructions of Z-Wave dongle).
3. Connect USB Z-Wave dongle to Raspberry PI.
4. Turn on the Raspberry PI computer.
5. Install “Domotics”. CLI command: sudo curl -L install.domoticz.com | bash
6. Add and register USB dongle and the bulb.
7. Install “node.js”. CLI command:
   sudo wget https://nodejs.org/dist/v8.9.0/node-v8.9.0-linux-armv6l.tar.gz
8. Install Google Firebase. CLI command:
   npm init
   npm install --save firebase
9. Launch “StartFBZWBridge.js” script. CLI command: node StartFBZWBridge.js
10. After doing of these actions you are ready to use the system.

Usage instructions.

1. Turn on the Raspberry PI (with Z-Wave dongle) and smart led-bulb.
2. Launch the mobile application “Control with brain waves”.
3. Follow instructions on the screen of the device.

9 TESTING

9.1 Test planning

Table 12: Testing plan for "Brain.iot" project

Code	Name of testing	Expecting result
T-P	System performance testing.	The system should provide responsiveness of reaction on user's brain data changes. Time of reaction depends on the length of data transfer from mobile application to the home automation hub through Google Firebase real-time service.
T-FN	Functional testing	The program demonstrates predictable behavior which complies the requirements (table 7, MR) and doesn't have hangings and runtime errors.

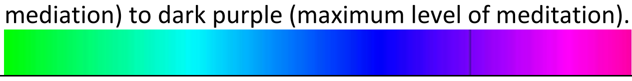
Table 13: performance testing plan

Test case code	Tested requirement	Expected result
T-P-1	PR-1	System should guaranty reaction on events of changing of the user's EEG data during not more than 1 second (i.e. the smart led-bulb should change its color or brightness immediately after corresponding EEG data was changed).

Table 14: functional testing plan

Test case code	Tested requirement	Expected result
T-F-1	MR-4, 5	The software should check availability of Internet connection prior to launching of its main functionality. In the case of absence of internet connection, the application should show appropriate message and wait until connection will be established.
T-F-2	MR-6, 7	The software should check availability of Bluetooth connection prior to launching of its main functionality. In the case of absence of Bluetooth connection, the application should show appropriate message and wait until connection will be enabled.
T-F-3	MR-8, 9	The software should check availability of NeuroSky MindWave device which is should be preliminary paired with mobile device which is used. In the case of absence of appropriate NeuroSky MindWave device, the application should stop its functioning and show error message to the user.
T-F-4	MR-10	The application should check the state of the detected NeuroSky MindWave device before starting of its main functionality.
T-F-5	MR-11, 12	In the case of erroneous state of the NeuroSky MindWave device (device is off) the application should show appropriate message and wait until the device will be turned on. The application should monitor the state of the NeuroSky MindWave device while functioning. In the case the device becomes unavailable the application should return user to the screen of connection establishing and show the appropriate message.
T-F-6	MR-13, 14	The application should constantly read data from NeuroSky MindWave device and show in the title of the main window. The application should monitor two characteristics of brain activity of the user: level of brain activity (concentration), level of brain relaxation (meditation).

T-F-7	MR-15	The application should provide user with ability to distantly control functioning of LED bulb (section 4.1.3 of the report).
T-F-8	MR-16	The application should provide the user with GUI for choosing of one of three modes of working: Mode 1) control of the LED-bulb by the value of the level of his brain's attention (concentration), Mode 2) control of the LED-bulb by the value of the level of his brain's relaxation (mediation), Mode 3) control of the LED-bulb by the value calculated on the base of combination of two mentioned levels.
T-F-9	MR-17	The application should provide visible neurofeedback by accepting data from the brain of the user (through BCI EEG device being used) and converting it to the commands directed to the home automation server which change behaviour of the bulb in each of modes mentioned in MR-16: Sub-mode (a) state of the smart LED-bulb (on/off) (not for mixed mode), Sub-mode (b) brightness, Sub-mode (c) colour.
T-F-10	MR-18-21	The application should provide visible neurofeedback by accepting data from the brain of the user (through BCI EEG device being used) and converting it to the demonstrative graphical representation in the main window of the application. Mode 1, sub-mode a. Level of user attention should be converted to two possible states of the smart LED lamp: on and off. Mode 1, sub-mode a. The smart LED lamp should be turned on (by the application) when the level of attention exceeds maximum threshold value during about 10 consecutive seconds. Mode 1, sub-mode a. The smart LED lamp should be turned off (by the application) when the level of attention becomes less then minimum threshold value during about 10 consecutive seconds.
T-F-11	MR-22 - 30	Mode 1, sub-mode a. Maximum threshold value should be stored in the cloud Google Firebase database (applies for attention and meditation levels). Mode 1, sub-mode a. Minimum threshold value should be stored in the cloud Google Firebase database (applies for attention and meditation levels). Mode 1, sub-mode a. The application should provide GUI for changing of minimum and maximum threshold values (applies for attention and meditation levels). Mode 1, sub-mode a. Changes of maximum threshold should be applied in the cloud database immediately after confirmation from the user (applies for attention and meditation levels). Mode 1, sub-mode a. Changes of minimum threshold should be applied in the cloud database immediately after confirmation from the user (applies for attention and meditation levels). Mode 1, sub-mode a. Changes of maximum threshold should be applied in the GUI immediately after user confirms them (applies for attention and meditation levels). Mode 1, sub-mode a. Changes of minimum threshold should be applied in the GUI immediately after confirmation from the user (applies for attention and meditation levels). Mode 1, sub-mode a. Changes of maximum threshold should immediately affect the behavior of the smart LED lamp after confirmation from the user was received (applies for attention and meditation levels). Mode 1, sub-mode a.

		Changes of minimum threshold should immediately affect the behavior of the smart LED lamp after confirmation from the user was received (applies for attention and meditation levels).
T-F-12	MR-31, 32	Mode 1, sub-mode b. Level of the user's attention should be converted to appropriate level of brightness of white color of the smart LED-lamp (see table 6). The dependence of the level of brightness of the bulb on the level of brain's concentration should be linear (level 0 should be ignored, because indicates improper functioning of the NeuroSky MindWave device). Mode 1, sub-mode b. Level of user attention should be correspondingly depicted in the graphical user interface of the application in order to allow the user to comprehend level of his mind activity. This should be the chart with visual meter which changes its position in accordance with the state of level of activity of the user.
T-F-13	MR-33, 34	Mode 1, sub-mode c. Level of the user's attention should be converted to appropriate color of the smart LED-lamp (see table 6). Color of the bulb should be in the range form green (minimum level of attention) to red (maximum level of attention).  The dependence of the color of the bulb on the attention level should be linear (level 0 should be ignored, because indicates improper functioning of the NeuroSky MindWave device). Mode 1, sub-mode c. Level of user attention should be correspondingly depicted in the graphical user interface of the application in order to allow the user to comprehend level of his mind activity. This should be the rectangle with pointer indicating the current color. Indicator should move along rectangle in accordance with the state of level of brain's activity of the user.
T-F-14	MR-35	Mode 1. The app should provide the user with additional graphical interface for training of user's ability to promote his level of attention by concentrating on specially organized animation.
T-F-15	MR-36	Mode 1. The app should provide concise information about concentration (attention) and how to develop this state of his mind.
T-F-16	MR-37, 38, 39	Mode 2, sub-mode a. Level of user's mind relaxation (meditation) should be converted to two possible states of the smart LED lamp: on and off. Mode 2, sub-mode a. The smart LED lamp should be turned on (by the application) when the level of user's mind relaxation exceeds maximum threshold value during about 10 consecutive seconds. Mode 2, sub-mode a. The smart LED lamp should be turned off (by the application) when the level of user's mind relaxation becomes less then minimum threshold value during about 10 consecutive seconds.
T-F-17	MR-40, 41	Mode 2, sub-mode b. Level of the user's mind relaxation should be converted to appropriate level of brightness of white color of the smart led lamp (see table 6). The dependence of the level of brightness of the bulb on the level of brain's relaxation should be inversely proportional, i.e. the more relaxed the subject's mind, the weaker the brightness (level 0 should be ignored, because indicates improper functioning of the NeuroSky MindWave device). Mode 2, sub-mode b. Level of user's mind relaxation should be correspondingly depicted in the graphical user interface of the application in order to allow the user to comprehend level of his mind relaxation (meditation). This should be the rectangular chart with visual meter inside which changes its position in accordance with the state of level of activity of the user.
T-F-18	MR-42, 43	Mode 2, sub-mode c. Level of the user's meditation should be converted to appropriate color of the smart LED-lamp (see table 6). Color of the bulb should be in the range form green (minimum level of mediation) to dark purple (maximum level of meditation). 

<table>
<tr><td></td><td></td><td>The dependence of the color of the bulb on the meditation level should be linear (level 0 should be ignored, because indicates improper functioning of the NeuroSky MindWave device (NeuroSky Inc., 2018)).
Mode 2, sub-mode c.
Level of user's mind meditation should be correspondingly depicted in the graphical user interface of the application in order to allow the user to comprehend level of his mind activity. This should be the rectangle with pointer indicating the current color. Indicator should move along rectangle in accordance with the state of level of brain's relaxation of the user.</td></tr>
<tr><td>T-F-19</td><td>MR-44</td><td>Mode 2.
The app should provide the user with additional graphical interface for training of user's ability to promote his level of relaxation by concentrating on specially organized animation. It is often recommended to close the eyes to bring up the level of meditation. Considering this circumstance, the app should provide user with audio neurofeedback. During this exercise the app should play the relaxing tune which volume of which depends on the level of user's brain meditation.</td></tr>
<tr><td>T-F-20</td><td>MR-45</td><td>Mode 2.
The app should provide concise information about mind relaxation (meditation) and how to develop this state of his mind.</td></tr>
<tr><td>T-F-21</td><td>MR-47</td><td>Mode 3, sub-mode b
The app should calculate the difference between the level of attention and the level of relaxation. On the base of this difference brightness of the smart LED lamp should be changed using the following scale:
<table>
<tr><td>Range of value of difference</td><td>Corresponding value of brightness of the led-bulb</td></tr>
<tr><td>0..20</td><td>10</td></tr>
<tr><td>21..40</td><td>31</td></tr>
<tr><td>41..60</td><td>51</td></tr>
<tr><td>61..80</td><td>71</td></tr>
<tr><td>81-100</td><td>90</td></tr>
</table>
Difference is should be calculated if values are overlapped. This means that sum of both values should be equal or greater than 100. The more difference between two levels the brighter the lamp.</td></tr>
<tr><td>T-F-22</td><td>MR-48</td><td>Mode 3, sub-mode c
The app should calculate the difference between the level of attention and the level of relaxation.
A) If difference is less or equal to 30, the app should report that user's mind is in balanced state and change color of the bulb to yellow.
B) If difference is more than 30 and level of attention prevails, the app should report that user's mind is highly focused and change color of the led-bulb to red.
If difference is more than 30 and level of relaxation value prevails, the app should report that user's mind is significantly relaxed and change color of the led-bulb to purple.</td></tr>
</table>

9.2 System performance test case T-P-1

The goal of this test case is to evaluate the performance of hardware and software which are used in the project. It is important to guarantee that the system provide responsive reaction during the process of its functioning. This aspect is important because the system should provide real-time neurofeedback when changes in user's mind should immediately change color or brightness of smart-bulb.

According to existing explorations in the area of usability and responsiveness (Nielsen, 1993) there are results which describe level of users' satisfaction by responsiveness of software depending on its reaction time on their actions. And here some threshold values were found (table 15). These numbers (first column) are due to the peculiarities of human perception.

Table 15: response time and its influence on users' experience

Time of reaction of the software (sec.)	Description of users' subjective perception	Recommended software behavior in the case of hangs when a user has to wait
about 0.1	Usually users feel satisfied and say that accept program behavior as smooth and instant.	No special actions needed.
about 1.0	Usually people feel some delay, but 1 second is the limit when users still fell themselves uninterrupted. If reaction is a bit longer, people experience some lack of control over software.	Usually nothing special is needed.
about 7-10	This is the maximum time which is normally can attract focus of human attention. Usually people show inclination to switch to other tasks and start feeling some kind of frustration.	In this case special continuous feedback from program is strictly required. Showing of progress bar or message with information about estimated time left highly recommended.

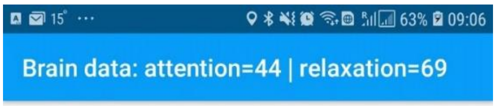
Performance testing was organized in the following way. Mobile application sends data of the user's brain activity to Google Firebase database over internet connection. Moment of sending was recorded to log file. After receiving of this data Google Firebase must resend it to the registered clients. Client is the server of home automation which is running script capable to receive notifications from Google Firebase. Moment of receiving of data was also written to log file. Then difference between these two moments was calculated. In this testing 2766 events were handled. Results of measuring is shown on fig. 52. Time of delivery of the event (i.e. changing of state of user's EEG data) from the smartphone to the hub of home automation system can be considered as a system response time. According to undertaken measurements the average time of response is about 482 milliseconds. According to table 15, this result shows that the overall level of the system responsiveness can be called satisfying because calculated average time is significantly less then threshold value of 1 second. Subjective visual observations also prove rather quick reaction of the system components on the events of EEG BCI device and immediate changes of the led-bulb state.

1	Time of receiving	Time of sending	Difference (ms)	Average
2	15:27:03:807	15:27:03.402	405.00	482.03
3	15:27:03:858	15:27:03.406	452.00	
4	15:27:03:862	15:27:04.402	460.00	
5	15:27:03:863	15:27:04.406	457.00	
6	15:27:03:864	15:27:05.402	462.00	
7	15:27:03:865	15:27:03.407	458.00	
8	15:27:03:866	15:27:03.460	406.00	
9	15:27:04:807	15:27:04.464	343.00	
10	15:27:04:850	15:27:04.400	450.00	
11	15:27:04:853	15:27:04.401	452.00	
12	15:27:04:854	15:27:04.399	455.00	
13	15:27:04:856	15:27:04.401	455.00	
14	15:27:04:857	15:27:04.450	407.00	
15	15:27:04:858	15:27:04.456	402.00	
16	15:27:05:810	15:27:05.402	408.00	
17	15:27:05:854	15:27:05.405	449.00	
2763	15:34:12:978	15:50:12.431	547.00	
2764	15:34:13:023	15:50:12.433	590.00	
2765	15:34:13:028	15:50:12.566	462.00	
2766	15:34:13:030	15:50:12.568	462.00	

Figure 52: results of measurement of events delivery time from smartphone to home automation hub

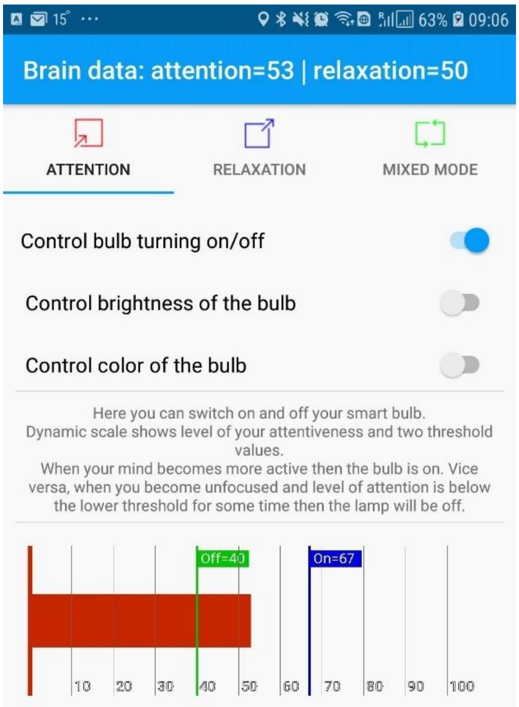
9.3 Functional test cases

<table>
<tr><th>Test case code</th><th>Tested requirement</th><th>Expected result</th><th>Test result</th></tr>
<tr><td>T-F-1</td><td>MR-4, 5</td><td>The software should check availability of Internet connection prior to launching of its main functionality. In the case of absence of internet connection, the application should show appropriate message and wait until connection will be established.</td><td>Passed</td></tr>
<tr><td colspan="4">1. Disable internet connection in Android control centre.
2. Launch the app.
3. Touch central round button to start connecting process.
4. Message about Internet connection’s absence appears. The software is waiting for internet connection.
5. Enable internet connection and return to the app.
6. Touch central round button to start connecting process.
7. Message about absence of Internet connection will not appear this time.
8. Stop the app.</td></tr>
<tr><td>T-F-2</td><td>MR-6, 7</td><td>The software should check availability of Bluetooth connection prior to launching of its main functionality. In the case of absence of Bluetooth connection, the application should show appropriate message and wait until connection will be enabled.</td><td>Passed</td></tr>
<tr><td colspan="4">1. Disable Bluetooth connection in Android control centre.
2. Launch the app.
3. Touch central round button to start connecting process.
4. Message about Bluetooth connection’s absence appears. The software is waiting for Bluetooth connection.
5. Enable Bluetooth connection in Android control centre. EEG BCI device should be paired preliminarily according to the user manual of this device.
6. Message about absence of Bluetooth connection will not appear this time.
7. Stop the app.</td></tr>
<tr><td>T-F-3</td><td>MR-8, 9</td><td>The software should check availability of NeuroSky MindWave device which is should be preliminary paired with mobile device which is used. In the case of absence of appropriate NeuroSky MindWave device, the application should stop its functioning and show error message to the user.</td><td>Passed</td></tr>
<tr><td colspan="4">1. Keep NeroSky MindWave device turned off.
2. Enable internet connection and Bluetooth in the Android control centre.
3. Launch the app.
4. Touch central round button to start connecting process.
5. Message about appropriate device’s absence appears. The software is waiting for device to appear.
6. Turn on the device, check battery level, put the device on (do everything according to instructions in the device’s manufacturer manual).
7. Return to the app and touch central button to start process of connection between the app and the device.
8. Start screen of attention mode will appear.
9. Stop the app.</td></tr>
<tr><td>T-F-4</td><td>MR-10</td><td>The application should check the state of the detected NeuroSky MindWave device before starting of its main functionality.</td><td>Passed</td></tr>
<tr><td colspan="4">1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device’s manufacturer manual).
3. Launch the app.</td></tr>
</table>

<table>
<tr><td colspan="4">
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear.
6. Disconnect device’s reference sensor from your earlobe.
7. Message about disappearing of connection appears.
8. The app stops functioning and shows start connection screen again.
9. Turn off the device.
10. Stop the app.
</td></tr>
<tr><td>T-F-5</td><td>MR-11, 12</td><td>In the case of erroneous state of the NeuroSky MindWave device (device is off) the application should show appropriate message and wait until the device will be turned on.
The application should monitor the state of the NeuroSky MindWave device while functioning. In the case the device becomes unavailable the application should return user to the screen of connection establishing and show the appropriate message.</td><td>Passed</td></tr>
<tr><td colspan="4">
1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device’s manufacturer manual).
3. Launch the app.
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear.
6. Turn the device off completely.
7. Message about disappearing of connection appears.
8. The app stops functioning and shows start connection screen again.
9. Stop the app.
</td></tr>
<tr><td>T-F-6</td><td>MR-13, 14</td><td>The application should constantly read data from NeuroSky MindWave device and show in the title of the main window.
The application should monitor two characteristics of brain activity of the user: level of brain activity (concentration), level of brain relaxation (meditation).</td><td>Passed</td></tr>
<tr><td colspan="4">
1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device’s manufacturer manual).
3. Launch the app.
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear.
6. Message in the title bar with two consciously changing numbers appears.


7. Stop the app.
</td></tr>
<tr><td>T-F-7</td><td>MR-15</td><td>The application should provide user with ability to distantly control functioning of LED bulb (section 4.1.3 of the report).</td><td>Passed</td></tr>
<tr><td colspan="4">
1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device’s manufacturer manual).
3. Launch the app.
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear.
6. Message in the title bar with two consciously changing numbers appears.
</td></tr>
</table>

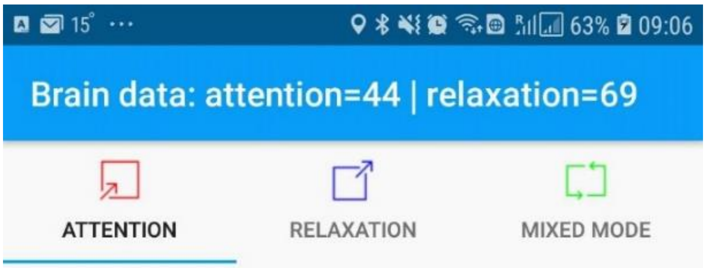
7. Touch first switch button (enabled state).



8. Follow the instructions on the screen. Regulate the level of your attention. The smart led-bulb will be turned on if level is greater than shown maximum threshold.
9. Touch first switch button again (disabled state). In this case the bulb will be turned off.
10. Stop the app.

T-F-8	MR-16	The application should provide the user with GUI for choosing of one of three modes of working: Mode 1) control of the LED-bulb by the value of the level of his brain's attention (concentration), Mode 2) control of the LED-bulb by the value of the level of his brain's relaxation (mediation), Mode 3) control of the LED-bulb by the value calculated on the base of combination of two mentioned levels.	Passed

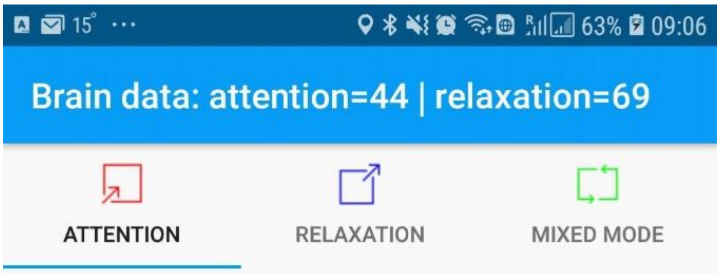
1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device's manufacturer manual).
3. Launch the app.
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear.
6. Message in the title bar with two consciously changing numbers appears.



7. The GUI consists of three tabs: attention (mode 1), relaxation (mode 2), Mixed mode (mode 3).
8. Stop the app.

<table>
<tr><td>T-F-9</td><td>MR-17</td><td>The application should provide visible neurofeedback by accepting data from the brain of the user (through BCI EEG device being used) and converting it to the commands directed to the home automation server which change behaviour of the bulb in each of modes mentioned in MR-16:
Sub-mode (a) state of the smart LED-bulb (on/off) (not for mixed mode),
Sub-mode (b) brightness,
Sub-mode (c) colour.</td><td>Passed</td></tr>
<tr><td colspan="4">

1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device’s manufacturer manual).
3. Launch the app.
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear.
6. Message in the title bar with two consciously changing numbers appears.

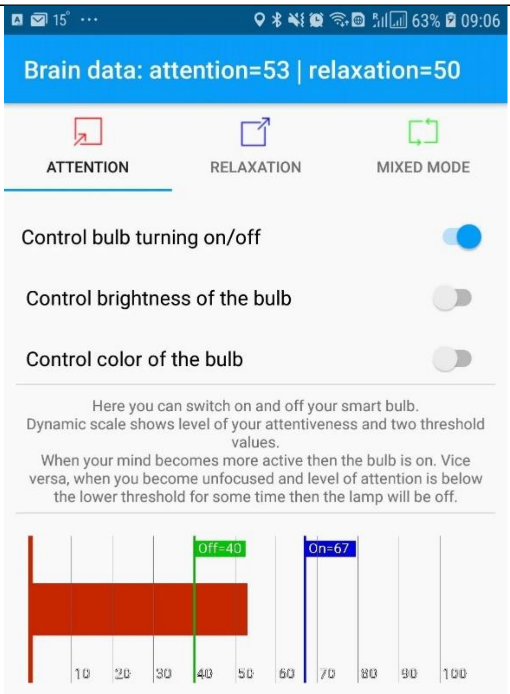


7. The GUI consists of three tabs: attention (mode 1), relaxation (mode 2), Mixed mode (mode 3).
8. Tab “ATTENTION” has three switch buttons for each of mention sub-modes.
9. Scroll to the “RELAXATION” tab. This tab has three switch buttons for each of mention sub-modes.
10. Scroll to the “MIXED MODE” tab. This tab has tow switch buttons for each of mention sub-modes.
11. Stop the app.

</td></tr>
<tr><td>T-F-10</td><td>MR-18-21</td><td>The application should provide visible neurofeedback by accepting data from the brain of the user (through BCI EEG device being used) and converting it to the demonstrative graphical representation in the main window of the application.
Mode 1, sub-mode a.
Level of user attention should be converted to two possible states of the smart LED lamp: on and off.
Mode 1, sub-mode a.
The smart LED lamp should be turned on (by the application) when the level of attention exceeds maximum threshold value during about 10 consecutive seconds.
Mode 1, sub-mode a.
The smart LED lamp should be turned off (by the application) when the level of attention becomes less then minimum threshold value during about 10 consecutive seconds.</td><td>Passed</td></tr>
<tr><td colspan="4">

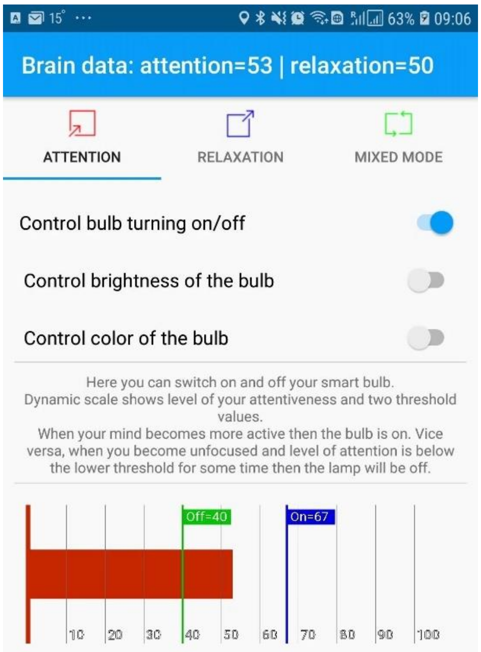
1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device’s manufacturer manual).
3. Launch the app.
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear.
6. Touch first switch button (enabled state).

</td></tr>
</table>

	 7. Follow the instructions on the screen. Regulate the level of your attention. The smart led-bulb will be turned on if level is greater than shown maximum threshold. 8. Follow the instructions on the screen. Regulate the level of your attention. The smart led-bulb will be turned off if level is less than shown minimum threshold. 9. Touch first switch button again (disabled state). In this case the bulb will be turned off. 10. Stop the app.		
T-F-11	MR-22 - 30	Mode 1, sub-mode a. Maximum threshold value should be stored in the cloud Google Firebase database (applies for attention and meditation levels). Mode 1, sub-mode a. Minimum threshold value should be stored in the cloud Google Firebase database (applies for attention and meditation levels). Mode 1, sub-mode a. The application should provide GUI for changing of minimum and maximum threshold values (applies for attention and meditation levels). Mode 1, sub-mode a. Changes of maximum threshold should be applied in the cloud database immediately after confirmation from the user (applies for attention and meditation levels). Mode 1, sub-mode a. Changes of minimum threshold should be applied in the cloud database immediately after confirmation from the user (applies for attention and meditation levels). Mode 1, sub-mode a. Changes of maximum threshold should be applied in the GUI immediately after user confirms them (applies for attention and meditation levels). Mode 1, sub-mode a. Changes of minimum threshold should be applied in the GUI immediately after confirmation from the user (applies for attention and meditation levels). Mode 1, sub-mode a.	Passed

		Changes of maximum threshold should immediately affect the behavior of the smart LED lamp after confirmation from the user was received (applies for attention and meditation levels). Mode 1, sub-mode a. Changes of minimum threshold should immediately affect the behavior of the smart LED lamp after confirmation from the user was received (applies for attention and meditation levels).	

1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device's manufacturer manual).
3. Launch the app.
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear.
6. Touch first switch button (enabled state).

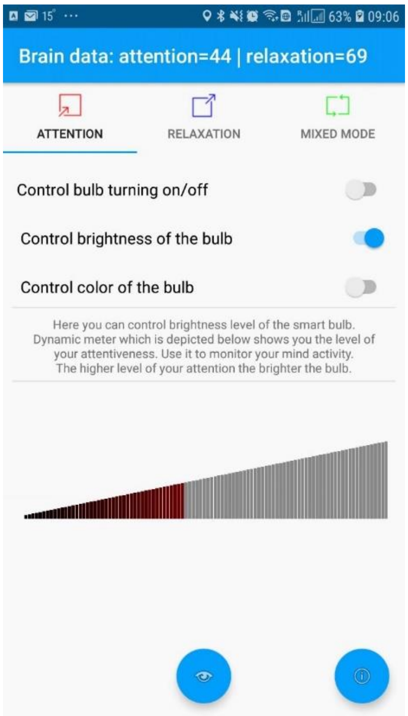


7. Round button with edit icon (pen) in the left corner appears.
8. Touch the button.
9. Setting window appears.
10. Change minimum value to 50.
11. Change maximum value to 60.
12. Touch Ok button and close the window.
13. Follow the instructions on the screen. Regulate the level of your attention. The smart led-bulb will be turned on if level is greater than shown maximum threshold. This means that new value is applied.
14. Follow the instructions on the screen. Regulate the level of your attention. The smart led-bulb will be turned off if level is less than shown minimum threshold. This means that new value is applied.
15. Touch first switch button again (disabled state). In this case the bulb will be turned off.
16. Stop the app.

T-F-12	MR-31, 32	Mode 1, sub-mode b. Level of the user's attention should be converted to appropriate level of brightness of white color of the smart LED-lamp (see table 6). The dependence of the level of brightness of the bulb on the level of brain's concentration should be linear (level 0 should be ignored,	Passed

<table>
<tr><td></td><td></td><td>because indicates improper functioning of the NeuroSky MindWave device).
Mode 1, sub-mode b.
Level of user attention should be correspondingly depicted in the graphical user interface of the application in order to allow the user to comprehend level of his mind activity. This should be the chart with visual meter which changes its position in accordance with the state of level of activity of the user.</td><td></td></tr>
<tr><td colspan="4">

1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device's manufacturer manual).
3. Launch the app.
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear.
6. Touch second switch button (enabled state).



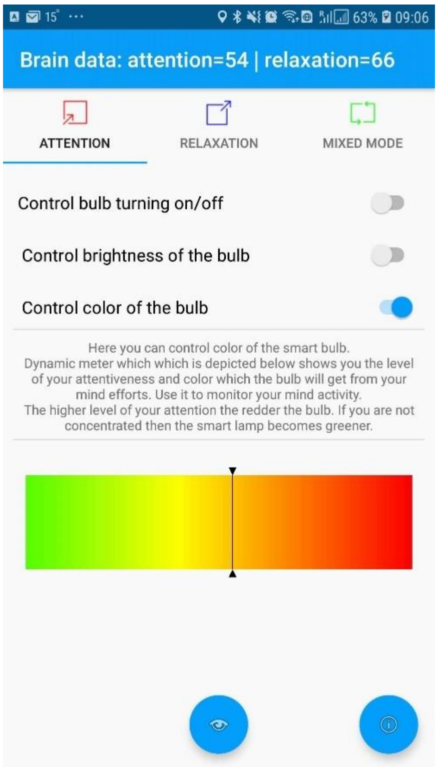
7. Follow the instructions on the screen. Regulate the level of your attention. The greater is the level, the brighter is the lamp (and vice versa).
8. Follow the instructions on the screen. Regulate the level of your attention. The greater is the level, the more intensive red color of the meter on the screen and higher its level (and vice versa).
9. Touch the second switch button again (disabled state). In this case the bulb will be turned off.
10. Stop the app.

</td></tr>
<tr><td>T-F-13</td><td>MR-33, 34</td><td>Mode 1, sub-mode c.
Level of the user's attention should be converted to appropriate color of the smart LED-lamp (see table 6). Color of the bulb should be in the range form green (minimum level of attention) to red (maximum level of attention).

The dependence of the color of the bulb on the attention level should be linear (level 0 should be ignored, because indicates improper functioning of the NeuroSky MindWave device).
Mode 1, sub-mode c.</td><td>Passed</td></tr>
</table>

<table>
<tr><td></td><td></td><td>Level of user attention should be correspondingly depicted in the graphical user interface of the application in order to allow the user to comprehend level of his mind activity. This should be the rectangle with pointer indicating the current color. Indicator should move along rectangle in accordance with the state of level of brain's activity of the user.</td><td></td></tr>
<tr><td colspan="4">

1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device's manufacturer manual).
3. Launch the app.
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear.
6. Touch third switch button (enabled state).



7. Follow the instructions on the screen. Regulate the level of your attention. The greater is the level, the redder is color of the lamp (and vice versa – yellow and green colours correspond to lesser values of attention).
8. Follow the instructions on the screen. Regulate the level of your attention. The greater is the level, the closer the is position of the pointer to red area of the meter on the screen (and vice versa – positions of pointer in the areas of yellow and green colours correspond to lesser values of attention).
9. Touch third switch button again (disabled state). In this case the bulb will be turned off.
10. Stop the app.

</td></tr>
<tr><td>T-F-14</td><td>MR-35</td><td>Mode 1.
The app should provide the user with additional graphical interface for training of user's ability to promote his level of attention by concentrating on specially organized animation.</td><td>Passed</td></tr>
<tr><td colspan="4">

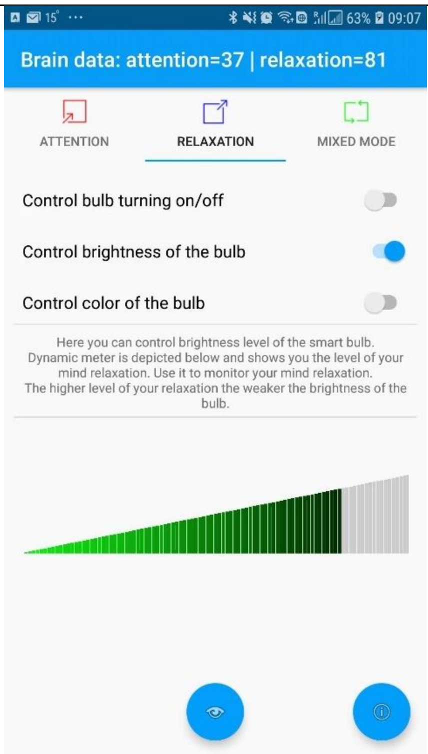
1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device's manufacturer manual).
3. Launch the app.

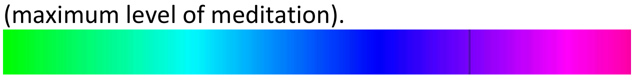
</td></tr>
</table>

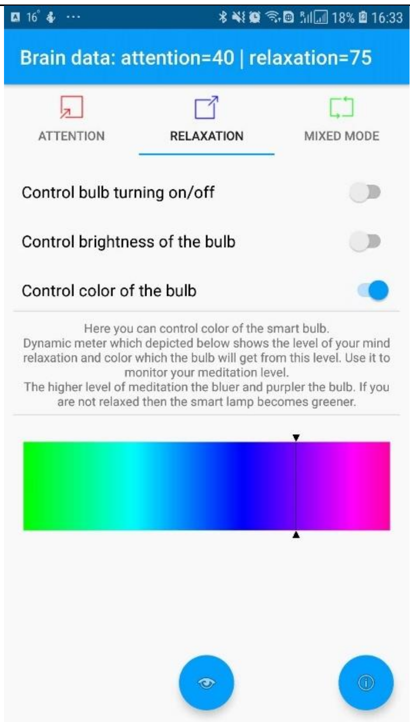
<table>
<tr><td colspan="4">4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear.
6. Touch any switch button (enabled state).
7. Round button with eye icon in the central bottom area appears.
8. Touch this button.
9. Popup window with animated graphical scene appears.
10. Follow the instruction in the title of the window and watch how circle follows the level of your attention.
11. Close the popup window by touching of Ok button.
12. Touch any switch button again (disabled state). In this case the bulb will be turned off.
13. Stop the app.</td></tr>
<tr><td>T-F-15</td><td>MR-36</td><td>Mode 1.
The app should provide concise information about concentration (attention) and how to develop this state of his mind.</td><td>Passed</td></tr>
<tr><td colspan="4">1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device’s manufacturer manual).
3. Launch the app.
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear.
6. Touch any switch button (enabled state).
7. Round button with “i” icon in the right bottom area appears.
8. Touch this button.
9. Popup window with help information appears.
10. Close the popup window by touching of Ok button.
11. Touch any switch button again (disabled state). In this case the bulb will be turned off.
12. Stop the app.</td></tr>
<tr><td>T-F-16</td><td>MR-37, 38, 39</td><td>Mode 2, sub-mode a.
Level of user’s mind relaxation (meditation) should be converted to two possible states of the smart LED lamp: on and off.
Mode 2, sub-mode a.
The smart LED lamp should be turned on (by the application) when the level of user’s mind relaxation exceeds maximum threshold value during about 10 consecutive seconds.
Mode 2, sub-mode a.
The smart LED lamp should be turned off (by the application) when the level of user’s mind relaxation becomes less then minimum threshold value during about 10 consecutive seconds.</td><td>Passed</td></tr>
<tr><td colspan="4">1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device’s manufacturer manual).
3. Launch the app.
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear. Scroll window to the tab “RELAXATION”.
6. Touch first switch button (enabled state).</td></tr>
</table>

<table>
<tr><td colspan="4">

7. Follow the instructions on the screen. Regulate the level of your mind's relaxation. The smart led-bulb will be turned on if level is greater than shown maximum threshold.
8. Follow the instructions on the screen. Regulate the level of your mind's relaxation. The smart led-bulb will be turned off if level is less than shown minimum threshold.
9. Touch first switch button again (disabled state). In this case the bulb will be turned off.
10. Stop the app.</td></tr>
<tr><td>T-F-17</td><td>MR-40, 41</td><td>Mode 2, sub-mode b.
Level of the user's mind relaxation should be converted to appropriate level of brightness of white color of the smart led lamp (see table 6). The dependence of the level of brightness of the bulb on the level of brain's relaxation should be inversely proportional, i.e. the more relaxed the subject's mind, the weaker the brightness (level 0 should be ignored, because indicates improper functioning of the NeuroSky MindWave device).
Mode 2, sub-mode b.
Level of user's mind relaxation should be correspondingly depicted in the graphical user interface of the application in order to allow the user to comprehend level of his mind relaxation (meditation). This should be the rectangular chart with visual meter inside which changes its position in accordance with the state of level of activity of the user.</td><td>Passed</td></tr>
<tr><td colspan="4">1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device's manufacturer manual).
3. Launch the app.
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear. Swipe to the "RELAXATION" tab.
6. Touch second switch button (enabled state).</td></tr>
</table>

<table>
<tr><td colspan="4">

7. Follow the instructions on the screen. Regulate the level of your mind's relaxation. The greater is the level, the weaker is the brightness of the lamp (and vice versa).
8. Follow the instructions on the screen. Regulate the level of your mind's relaxation. The greater is the level, the less intensive green color of the meter on the screen and lower its level (and vice versa).
9. Touch the second switch button again (disabled state). In this case the bulb will be turned off.
10. Stop the app.</td></tr>
<tr><td>T-F-18</td><td>MR-42, 43</td><td>Mode 2, sub-mode c.
Level of the user's meditation should be converted to appropriate color of the smart LED-lamp (see table 6). Color of the bulb should be in the range form green (minimum level of mediation) to dark purple (maximum level of meditation).

The dependence of the color of the bulb on the meditation level should be linear (level 0 should be ignored, because indicates improper functioning of the NeuroSky MindWave device (NeuroSky Inc., 2018)).
Mode 2, sub-mode c.
Level of user's mind meditation should be correspondingly depicted in the graphical user interface of the application in order to allow the user to comprehend level of his mind activity. This should be the rectangle with pointer indicating the current color. Indicator should move along rectangle in accordance with the state of level of brain's relaxation of the user.</td><td>Passed</td></tr>
<tr><td colspan="4">1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device's manufacturer manual).
3. Launch the app.
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear.
6. Touch third switch button (enabled state).</td></tr>
</table>

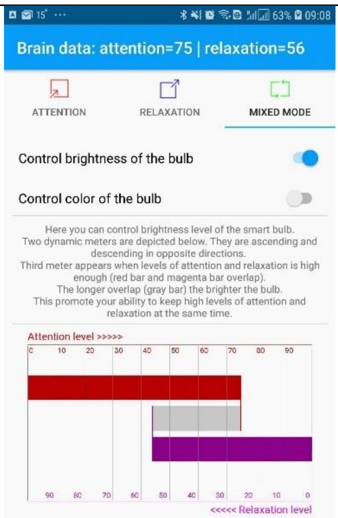


7. Follow the instructions on the screen. Regulate the level of your mind’s relaxation. The greater is the level, the more purple is color of the lamp (and vice versa – blue and green colours correspond to lesser values of mind’s relaxation).
8. Follow the instructions on the screen. Regulate the level of your mind’s relaxation. The greater is the level, the closer is the position of the pointer to purple area of the meter on the screen (and vice versa – positions of pointer in the areas of green and blue colours correspond to lesser values of relaxation).
9. Touch third switch button again (disabled state). In this case the bulb will be turned off.
10. Stop the app.

T-F-19	MR-44	Mode 2. The app should provide the user with additional graphical interface for training of user’s ability to promote his level of relaxation by concentrating on specially organized animation. It is often recommended to close the eyes to bring up the level of meditation. Considering this circumstance, the app should provide user with audio neurofeedback. During this exercise the app should play the relaxing tune which volume of which depends on the level of user’s brain meditation.	Passed

1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device’s manufacturer manual).
3. Set the sound level of the system to medium value (or other individually acceptable value).
4. Launch the app.
5. Touch central round button to start connecting process.
6. Start screen of attention mode will appear. Swipe screen to the left toward “RELAXATION” tab.
7. Touch any switch button (enabled state).
8. Round button with eye icon in the central bottom area appears.
9. Touch this button.
10. Popup window with animated graphical scene appears.
11. Follow the instruction in the title of the window and watch how circle follows the level of your relaxation.

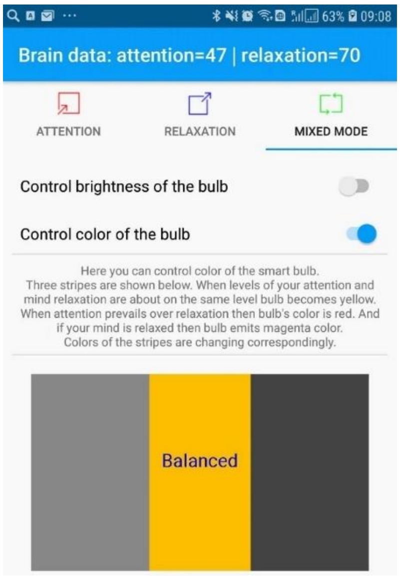
<table>
<tr><td colspan="4">12. Along with animation, sound track (ocean's waves and wind) will be playing in the background. Sound volume will follow the level of your mind's relaxation.
13. Close the popup window by touching of Ok button.
14. Touch any switch button again (disabled state). In this case the bulb will be turned off.
15. Stop the app.</td></tr>
<tr><td>T-F-20</td><td>MR-45</td><td>Mode 2.
The app should provide concise information about mind relaxation (meditation) and how to develop this state of his mind.</td><td>Passed</td></tr>
<tr><td colspan="4">1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device's manufacturer manual).
3. Launch the app.
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear. Swipe to "RELAXATION" tab.
6. Touch any switch button (enabled state).
7. Round button with "i" icon in the right bottom area appears.
8. Touch this button.
9. Popup window with help information appears.
10. Close the popup window by touching of Ok button.
11. Touch any switch button again (disabled state). In this case the bulb will be turned off.
12. Stop the app.</td></tr>
<tr><td>T-F-21</td><td>MR-47</td><td>Mode 3, sub-mode b
The app should calculate the difference between the level of attention and the level of relaxation. On the base of this difference brightness of the smart LED lamp should be changed using the following scale:
<table>
<tr><td>Range of value of difference</td><td>Corresponding value of brightness of the led-bulb</td></tr>
<tr><td>0..20</td><td>10</td></tr>
<tr><td>21..40</td><td>31</td></tr>
<tr><td>41..60</td><td>51</td></tr>
<tr><td>61..80</td><td>71</td></tr>
<tr><td>81-100</td><td>90</td></tr>
</table>
Difference is should be calculated if values are overlapped. This means that sum of both values should be equal or greater than 100. The more difference between two levels the brighter the lamp.</td><td>Passed</td></tr>
<tr><td colspan="4">1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device's manufacturer manual).
3. Launch the app.
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear. Scroll window to the tab "MIXED MODE".
6. Touch first switch button (enabled state).</td></tr>
</table>



7. Follow the instructions on the screen. Regulate the levels of your mind's relaxation and attention. The smart led-bulb will be tuned to the difference between these levels. The more each level is, the brighter is the bulb.
8. Follow the instructions on the screen. Regulate the levels of your mind's relaxation and attention. Graphical level's meters on the screen overlap if levels exceed the value of 50. The length of overlap area will be tuned to the difference between these levels. The more each level is, the longer is the area of overlap.
9. Touch first switch button again (disabled state). In this case the bulb will be turned off.
10. Stop the app.

T-F-22	MR-48	Mode 3, sub-mode c The app should calculate the difference between the level of attention and the level of relaxation. C) If difference is less or equal to 30, the app should report that user's mind is in balanced state and change color of the bulb to yellow. D) If difference is more than 30 and level of attention prevails, the app should report that user's mind is highly focused and change color of the led-bulb to red. If difference is more than 30 and level of relaxation value prevails, the app should report that user's mind is significantly relaxed and change color of the led-bulb to purple.	Passed

1. Enable internet connection and Bluetooth in the Android control centre.
2. Turn on the device, check battery level, put the device on (do everything according to instructions in the device's manufacturer manual).
3. Launch the app.
4. Touch central round button to start connecting process.
5. Start screen of attention mode will appear. Scroll window to the tab "MIXED MODE".
6. Touch second switch button (enabled state).
7. Follow the instructions on the screen. Regulate the levels of your mind's relaxation and attention. The smart led-bulb will be tuned to the difference between these levels. If two levels are close to each other the bulb emits yellow color. If difference is rather big then color of the bulb will be red (attention prevails) or purple (relaxation prevails).



8. Follow the instructions on the screen. Regulate the levels of your mind's relaxation and attention. The colours of rectangles on the screen will be changed according to the difference between these levels. If two levels are close to each other, then central rectangle has yellow color and shows "Balanced" text label. If difference is rather big then left rectangle is red and shows "Focused" text label (attention prevails) or right rectangle is purple and shows "Relaxed" text label (relaxation prevails).
9. Touch first switch button again (disabled state). In this case the bulb will be turned off.
10. Stop the app.

9.4 Reflection on testing

Testing process can be considered as some kind of challenge for developers of software systems. But anyway, this process is important for final desired outcomes. This time testing gave me ability to undertake evaluation of performance of the system. Before this testing was undertaken, time of reaction of Google Firebase was considered as possible issue. This database is the remote part of the system and works like a retranslation node which receives data and resend them to home automation system. Nevertheless, timing measurement showed acceptable result. Firebase resends data within time frame which is psychologically perceived by a human as immediate. This allows to change color and brightness of the smart led-bulb during the insensitive period of time. The most important part of functional testing was evaluation of implemented connectivity functions of mobile software. Functioning of the entire system depends on how the user's smartphone interacts with external EEG BCI device which can not be controlled. The system being developed acts as a client of EEG NeuroSky MindWave headset. Also, software should be connected to internet because EEG signals from the device transferred to the bulb through external Google Firebase service. This test case took fair long time and gave ability to expand my skills in the area of thorough documenting of this process.

10 Conclusion

10.1 Reflection

I was impressed by the topics of this project many years ago. But continuous business with everyday job duties did not allow to go deep inside it. Studying in EDENS colleges finally gave me ability to study and realize the system which combine such thrilling and perspective technologies like BCI (brain-computer interface), IoT (internet of things) and home automation (smart home or "domotics").

The entire process of researching and implementation took two months and involved following activities:

1. Planning and design of overall project's architecture (considering hardware and software components for future IoT infrastructure of the project). Choice of SDLC.
2. Studying of brain computer-interface.
    a. Analysis of hardware components, choice of the BCI-device. Evaluation of suitable technology, sensors, etc.
    b. Analysis of software components for the device (API, SDK, suitable programming languages).
    c. Creation and testing of mobile application's prototype for interaction with chosen BCI-device.
3. Studying of smart home technologies.
    a. Analysis of hardware of home automation systems with the respect to project's aims. Choice of equipment. Choice of home automation hub.
    b. Analysis of software components of home automation systems, APIs, SDKs, protocols.
    c. Creation and testing of prototype of software for intercommunication between mobile application and smart home device.
4. Studying of technologies for implementation of IoT.
    a. Choice of suitable technology for IoT implementation (considering performance, ethics issues, security issues, consistency of data being used). Design of database structure.
    b. Analysis of software components of IoT: APIs, SDKs, programming languages.
    c. Creation and testing of the prototype of the application for intercommunication between BCI and IoT, home automation system and IoT.
5. Overall project development according to proposed architecture.
    a. Assembling all the hardware components in one system. Testing of previously created prototypes for suitability for using in the final software being developed.
    b. Development and testing of mobile application for communication between EEG BCI-device and IoT component of the project.
    c. Development of application for home automation hub for implementation of interconnection between IoT component, home automation server and smart device.
6. Overall testing of all components working together.

These activities allowed me to accustom myself with modern and perspective technologies.

First of all, BCI should me mentioned. This not only theoretically interesting. When this technology will be developed to its full scale (along with robotized mechanical systems), people with physical disabilities can get very valuable and useful instruments to significantly improve quality of their lives. For me these instrument gives ability to improve my mental productivity during intellectual activities. Programming of BCI gave me skills of retrieving and handling of raw streams of data. Also, I gained experience in studying and applying of third-party's API (which is provided by NeuroSky). Source code of the API was not very well documented but general quality of provided classes can be called as rather high. In connection with BCI, another valuable experience should be called. I mean programming of connections over Bluetooth low energy interface. Android SDK provides straightforward (in the sense of instantiation and usage) and convenient classes for implementation of this task. Practical usage of the device shows some drawbacks. And the largest is that the device failed to properly detect eyes blinks. This obstacle did not allow unleash its full potential abilities.

Programming of mobile application for interaction with NeuroSky BCI-device allowed me to enrich and widen my knowledge of Kotlin programming language. I would rate this programming language as more flexible and expressive than Java (firstly syntactically).

Learning of IoT also became one another exiting part of the project. IoT can be implemented in different ways. This project utilizes real-time cloud service which is capable to collect and dispatch real-time data from EEG stream generated by the BCI device. I found that nowadays many leading IT companies (IBM, Google, Microsoft, Amazon and others) provide needed functionality. Comparing of their services can be useful for future projects during the rest of my studying. If to say about chosen service, I can call Google Firebase SDK rather impressive in the sense of its "transparency". Slight weak feature of this instrument is the lack of comprehensive and full description of the API (they provide step-by-step instructions and examples instead). This cannot be considered as good approach because does not give answers about non-standard use cases and sometimes I had to use the method of "trial and error".

Home automation and concomitant technologies gave me not only enrich my professional experience but also improve my living environment (I bought some devices for personal use). Remote wireless controlling of smart device gives new level of comfort. But programming of this technology gave ability to understand inner mechanisms of this technology and use them during the implementation of the project. Here I can mention home automation protocols (Z-Wave and ZigBee), home automation servers (such as Domoticz).

SUMMARY

The project “Brain.iot” was dedicated to the development and implementation of software and hardware system for creation of interface between a human brain, computer and devices of internet of things. The aim was achieved and resulting system allows the user to control external smart device of home automation system by data of his electroencephalogram. As was mentioned above, the created system involves both hardware and software part. Overall architecture of the system was designed from the scratch and can be characterized as responsive (on the base of data of undertaken performance testing) and corresponding to the requirements (on the base of undertaken functional testing).

References

Aeotec Inc. (2018, October 15). *Host Software.* Retrieved from Aeotec: https://aeotec.freshdesk.com/support/solutions/articles/6000042049-host-software

Aeotec Inc. (2018, October 10). *Z-Stick Gen5. Your gateway to building your own Z-Wave gateway.* Retrieved from Aeotec by Aeon Labs.: https://aeotec.com/z-wave-usb-stick

Aeotec Inc. (2018, October 10). *Z-Wave LED Bulbs.* Retrieved from Aeotec by Aeon Labs.: https://aeotec.com/z-wave-led-lightbulb

Amazon Corp. (2018, October 27). *AWS IoT*. Retrieved from Amazon Web Services - Cloud Computing Services: https://aws.amazon.com/iot/

Arnuphaptrairong, T. (2011). Top Ten Lists of Software Project Risks:Evidence from the Literature Survey. *Proceedings of the International MultiConference of Engineers and Computer Scientists.* Hong Kong: IMECS 2011.

Bijholt, A., Femke, N., Gerritsen, S., Poel, M., & Braam, L. (2015). Usability of Three Electroencephalogram Headsets for Brain–Computer Interfaces: A Within Subject Comparison. *Interacting with Computers*, 500-511.

Charara, S. (2018, October 15). *Zigbee vs Z-Wave: Two Big Smart Home Standards Explored*. Retrieved from The Ambient: https://www.the-ambient.com/guides/zigbee-vs-z-wave-298

CISCO. (2017). *IoT Fundamentals. Networking Technologies, Protocols, and Use Cases for the Internet of Things.* San Jose: Cisco Press.

Domoticz. (2018, October 11). *Control at Your Finger Tips.* Retrieved from Domoticz: https://domoticz.com/

Domoticz team. (2018, October 15). *Domoticz API/JSON URL's*. Retrieved from Domoticz: https://www.domoticz.com/wiki/Domoticz_API/JSON_URL's

Emotiv Inc. (2018, October 23). *Emotiv - Leaders in wireless EEG brain monitoring technology*. Retrieved from Homepage - Emotiv: https://www.emotiv.com/

Goodwin, S. (2013). *Smart Home Automation with Linux and Raspberry Pi.* Apress Media.

Google LLC. (2018, August 20). *Android Developers*. Retrieved from Build for Android: https://developer.android.com/design/

Google LLC. (2018, October 5). *A comprehensive mobile development platform.* Retrieved from Firebase: https://firebase.google.com/

Google LLC. (2018, October 25). *Google Cloud IoT - Fully Managed IoT Services*. Retrieved from Google cloud: https://cloud.google.com/solutions/iot/

Grousse, M. (2018, September 5). *10 Best IoT Platforms In 2018. IoT Technology Forecast.* Retrieved from DA-14: https://da-14.com/blog/10-best-iot-platforms-iot-technology-forecast

Hersent, O., & Boswarthick, D. (2012). *The Internet of Things: Key Applications and Protocols.* New York: Wiley.

Hosmer, C. (2018). *Defending IoT Infrastructures with the Raspberry Pi.* Apress Media.

IBM Corp. (2018, October 27). *Internet of Thing Platform*. Retrieved from IBM Cloud: https://console.bluemix.net/catalog/services/internet-of-things-platform

InteraXon Inc. (2018, October 23). *MUSE ™ | Meditation Made Easy*. Retrieved from MUSE ™ | Meditation Made Easy: http://www.choosemuse.com/

Kumar, D., & Arjunan, S. (2016). *Human-Computer Interface Technologies for the Motor Impaired.* CRC Press.

Langer, A. (2016). *Guide to Software Development. Designing and Managing the Life Cycle.* New York: Springer.

Microsoft Corp. (2018, October 26). *Azure IoT - Internet of Things Platform*. Retrieved from Microsoft Azure: https://azure.microsoft.com/en-us/overview/iot/

Moskala, M., & Wojda, I. (2017). *Android Development with Kotlin.* Birmingham-Mumbai: Packt Publishing.

NeuroSky Inc. (2018, October 25). *NeuroSky Developer Tools*. Retrieved from NeuroSky Developer - Docs: http://developer.neurosky.com/docs/doku.php?id=developer_tools

NeuroSky Inc. (2018, October 26). *Useful, Applicable Brain Activity Algorithms.* Retrieved from NeuroSky Inc. Body and Mind. Quantified.: http://neurosky.com/biosensors/eeg-sensor/algorithms/

NeuroSky, Inc. (2018, October 23). *EEG - ECG - Biosensors*. Retrieved from NeuroSky | Body and Mind. Quantified.: http://neurosky.com/

NewroSky Inc. (2018, October 25). *ThinkGear Serial Stream Guide*. Retrieved from NeuroSky Developer Program: http://developer.neurosky.com/docs/doku.php?id=thinkgear_communications_protocol

Nielsen, J. (1993). *Usability Engineering.* Morgan Kaufmann.

Pressman, R. (2001). *Software engineering.* New York: McGraw-Hill.

Raspberry Pi. (2016). *Documentation*. Retrieved from https://www.raspberrypi.org/documentation/

Samuel, S., & Bocutiu, S. (2017). *Programming Kotlin.* Birmingham: Packt Publishing.

Stoneburner, G., Goguen, A., & Feringa, A. (2002). *Risk Management Guide for Information Technology Systems.* Gaithersburg: National Institute of Standards and Technology.

Thalmic Labs. (2018, October 10). *Myo Gesture Control Armband*. Retrieved from Myo Gesture Control Armband: https://www.myo.com/

The Tobii Group. (2018, October 10). *Tobii.com - Tobii is the world leader in eye tracking*. Retrieved from Tobii Developer Zone - Eye Tracking SDK downloads, documentation and forum: https://developer.tobii.com/

Wallace, L., & Keil, M. (2004). Software Project Risk and their Effect on Outcomes. *Communication fo the ACM*, 68-73.

Zaphiris, P., & Ang, C. (2008). *Human Computer Interaction. Concepts, Methodologies, Tools, and Applications.* Citeseer.

ZigBee Alliance. (2018, October 19). *ZigBee Developers*. Retrieved from ZigBee Alliance: https://www.zigbee.org/zigbee-for-developers/zigbee-3-0/

Z-Wave Alliance. (2018, October 10). *About Z-Wave Technology*. Retrieved from The Internet of Things is powered by Z-Wave.: https://z-wavealliance.org/about_z-wave_technology/