POLITECNICO DI TORINO

Master of Science in Computer Engineering

Master Degree Thesis

NEUROLOGICAL CONSEQUENCES OF COVID-19

Development of a web application for executing an online test battery

Supervisors

Ch.ma Prof.ssa Gabriella Olmo

Dr. Vito De Feo

Candidate

Iacopo Giuseppe Foderini

ACADEMIC YEAR 2021/2022

Abstract

The Covid-19 pandemic, caused by SARS-CoV-2, emerged in Wuhan, China, between December 2019 and January 2020. At the time of writing this paper, it still represents a significant issue for global health and economics, as well as for the livelihood and for physical and mental well-being of people worldwide. A range of persistent symptoms can linger long after the acute infection as a part of a condition called Post-Covid-19 Syndrome or Long Covid. Long Covid can affect many organs and systems including the brain, causing fatigue, headaches, tinnitus and earaches, memory and concentration issues ('brain fog'), dizziness, sleep difficulties, anxiety and depression. Prevalence, type, and severity of such long-lasting effects and how they may link to other factors such as the severity of initial disease and comorbidities, is still unclear. 'Neurological integrity and cognitive functioning test battery', a project from the University of Essex, aims to investigate the behavioural and brain markers indicative of neural integrity (sensory integration) and cognitive functioning (attention and memory) as biomarkers of neurological damage from Covid-19. We are designing a web application that allows participants to perform an online battery of tests, consisting of a series of sensory and cognitive tasks including visual and auditory stimuli. Behavioural markers are then collected from participants' performance on these tasks as well as from their answers to Covid-19 symptom and general health surveys, preserved and made available to researchers through the system itself. The use of a web application enables the centralisation of the processes of information gathering and retrieval, and greatly simplifies the recruitment of participants that can take part in the experiment from their homes, with the only requirement that they have a computer with Internet access. The results of this study will help to understand which markers are affected by Covid-19 and how they are related, in order to support researchers, clinicians and rehabilitation therapists to better assess risk, chart recovery, and restore optimal mental and physical functioning after Covid-19. The purpose of this paper is to document the activities carried out as part of above-described research study, providing details on the system design, architecture and implementation.

Contents

List of Tables

List of Figures

Introduction

At the end of December 2019, Chinese health authorities notified the *World Health Organization* (*WHO*) of a cluster of pneumonia with unknown aetiology in the city of Wuhan, in the Hubei province. On the 9th of January 2020, *China CDC* (*Chinese Center for Disease Control and Prevention*) identified a new coronavirus, temporarily named *2019-nCov*, as the aetiologic cause of these diseases; the resulting respiratory illness was named *Covid-19* (Coronavirus Disease) by the WHO. Later, the *International Committee on Taxonomy of Viruses* classified the virus as *SARS-CoV-2*, as it formally associated it with the coronavirus that causes the severe acute respiratory syndrome (*SARS-CoVs*, Severe Acute Respiratory Syndrome coronaviruses). This newly discovered virus turned out to be highly contagious and rapidly evolved from being a serious problem confined to China, to a worldwide health emergency. This led the WHO to declare COVID-19 as a pandemic, on the 11th of March 2020 [1]. As of September 2022, there have been more than 600 million confirmed cases of COVID-19, including about 6 million deaths [2].

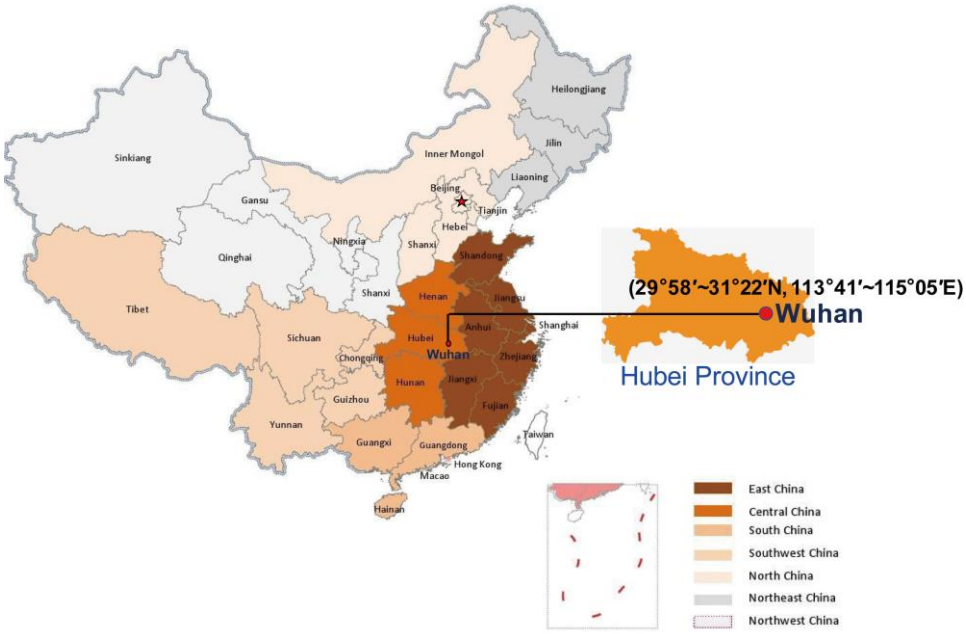

Figure 1: Geographic location of Wuhan, Hubei Province in China [3].

COVID-19 can lead to diverse clinical manifestations, ranging from an asymptomatic infection to an acute respiratory distress syndrome and multiorgan failure with high risk of mortality. However, most people with COVID-19 get better within a few days to a few weeks after infection, regardless of the severity of symptoms experienced. On the other hand, some people can experience long-term effects which can last weeks, months, or longer, as part of a condition named *Post-COVID-19 Syndrome* or *Long COVID*. Patients affected by this syndrome can experience a variety of symptoms, which can persist from the acute infection or can be newly developed,

and can affect different organs and systems. Though statistics for how many people suffer from Post-COVID condition can be quite different, depending on the population analysed, a recent study claims that estimated global incidence of Long COVID is of 43% [4]. It can be deduced, consequently, that Post-COVID-19 condition has a substantial prevalence, and that this can put a significant strain on countries' healthcare systems.

Considering the above, it is clearly understandable why scientific community, which was already putting a great effort into the research on COVID-19, has been recently focusing on Post-COVID-19 Syndrome. In particular, even though Long COVID impact on population is evident, as well as it is recognized as a clinical entity, there are many ongoing studies that aim to better define its characteristics, especially prevalence, type, and severity of long-lasting effects, and how they may link to other factors such as the severity of initial disease and comorbidities.

In this context, University of Essex has started a research project called "Neurological integrity and cognitive functioning test battery", in order to investigate Post-COVID Syndrome, focusing principally on its neurological effects. People affected by the above-mentioned disease, in fact, can suffer from a series of neurological conditions, such as fatigue, headaches, tinnitus and earaches, memory and concentration issues ("brain fog"), dizziness, sleep difficulties, anxiety and depression. The project's objective is to investigate the fundamental behavioural markers indicative of neural integrity (sensory integration) and cognitive functioning (attention and memory) in order to understand if they can act as biomarkers of neurological damage from COVID-19. Specifically, the research study aims to find out if these markers of cognitive functioning and neural integrity are affected by COVID-19 and if such effects systematically relate to the experience and severity of persistent symptoms. Through addressing these issues, it will be possible to support researchers, clinicians and rehabilitation therapists in better understanding which aspects of neural functioning to target for assessing risk and recovery, in order to restore optimal mental and physical functioning after COVID-19. Going specifically, the behavioural markers being studied are:

- response speed and accuracy to visual and auditory stimuli
- the "temporal binding window" of sensory integration
- sensory adaptation responses
- retention of verbal information.

The project involves the creation of a web application through which participants will be asked to execute several tasks and to answer some surveys, with the final purpose of collecting the aforementioned biomarkers. The primary idea is to deliver a web application that is completely developed, tested, deployed for production, in ways that benefit participants, that can execute tasks from the comfort of their homes and

from all over the world, as well as researchers, that can access to a fully operating tool, which could also be used for future studies.

The present paper is intended to outline the work carried during my participation in the University of Essex' project described so far. All the related activities were executed in a team composed of me and other two Essex' students, under the supervision of doctors Vito De Feo, Helge Gillmeister and Loes Van Dam. In particular, the team has been working on an expansion of an existing project which was inherited by previous groups shortly after the first launch stage. The inherited application necessitated, first of all, various bugfixes and, in second place, the addition of new features. The adopted approach involved iterative improvements agreed with the whole research team, made up of computer scientists and psychologists, on a weekly basis. Moreover, the required improvements were implemented according to a priority, defined by the aforesaid team. At the time of writing this paper, the web application supports all required features and improvements and has been deployed on a server accessible from the Internet, ready for final testing before its final use in the ongoing research.

Bearing in mind what has been told so far, this thesis is organized as follows:

- **Chapter I** provides the justification of the research, by means of an overview of COVID-19 disease, with an emphasis on its neurological effects. The tests designed to collect the behavioural markers under investigation are described in this chapter too.
- In **Chapter II**, the web application project is described in depth, highlighting its purposes and listing its functional and non-functional requirements.
- In **Chapter III** the designed solution is reported and detailed, characterizing the different layers that make up the application.
- **Chapter IV** lists the actual improvements and fixes that needed to be performed on the existing application, and the results obtained in implementing them.
- Lastly, in **Chapter V** the thesis' conclusions are presented, together with some possible future developments.

Chapter I

Justification

1.1 Definition

The coronaviruses fall under the order *Nidovirales* and are members of the *Coronaviridae* family, which is further divided into four primary genera: α, β, γ, δ. The latest coronavirus (classified as 2019-nCoV or SARS-CoV-2) is the seventh member of the big β -coronavirus (CoV) family [5]. β-coronavirus can be described as an enveloped, non-segmented, positive-sense single-stranded RNA virus.

The name "coronavirus" comes from Latin corona, meaning "crown" and it refers to the characteristic appearance of virions which in the photographs taken with the electron microscope form an image that resembles the appearance of a halo or a crown. The nucleocapsid of the virion, which is buried inside phospholipid bilayers and covered by the spike (S) glycoprotein coat, is made of genomic RNA and phosphorylated nucleocapsid (N) protein. In the virus envelope, the S proteins are home to the membrane (M) protein hemagglutinin-esterase (HE) and the envelope (E) protein [3].

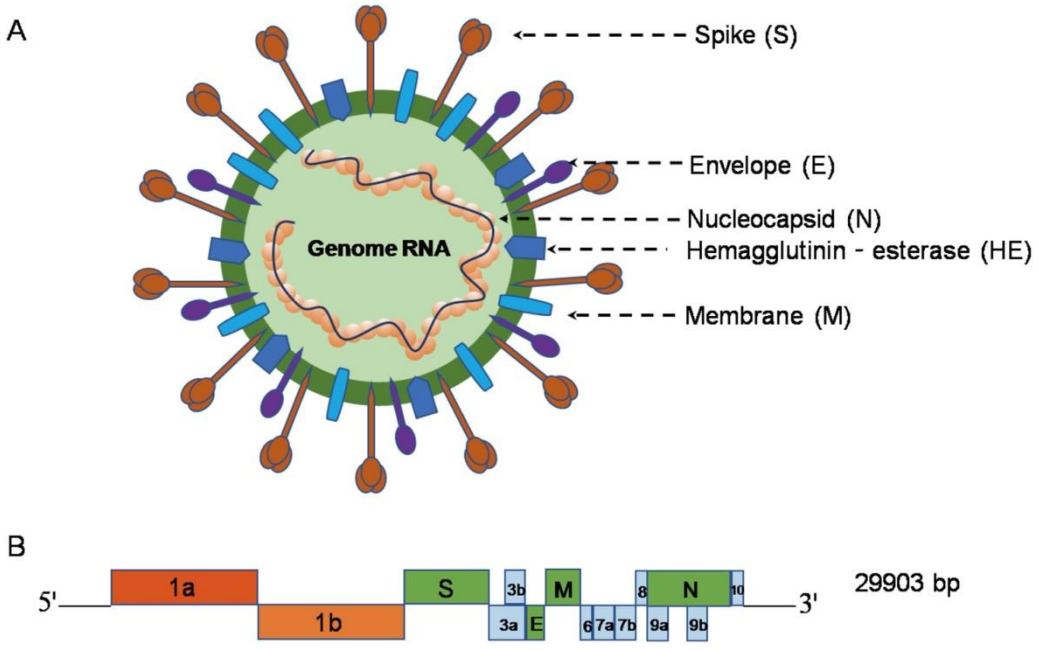

Figure 2: β-coronavirus particle and genome [3].

1.2 Viral lifecycle and host cell invasion

People can spread the virus to one another by exchanging respiratory droplets and aerosols. After entering the body, the virus connects to host receptors and enters host cells via membrane fusion or endocytosis.

The S protein, which is the most crucial for host attachment and penetration, is visible sticking out from the viral surface. This protein is made up of two functional subunits (S1 and S2), of which S1 is in charge of attaching to the receptor on the host cell, while S2 is involved in the fusing of the membranes of the viral and host cells.

As a functional SARS-CoV receptor that is significantly expressed in pulmonary epithelial cells, ACE-2 has been found.

The S protein first attaches to this host receptor to initiate the virus's invasion of the host cell. The S protein is activated by a two-step protease cleavage process after SARS-CoV-2 binds to ACE-2: the first cleavage is for priming at the S1/S2 cleavage site, and the second is for activation at a region close to a fusion peptide inside the S2 subunit. The first cleavage stabilizes the S2 subunit at the attachment point, and the second cleavage probably activates the S protein and causes conformational changes that fuse the membranes of the viral and host cells.

The viral contents are released inside the pulmonary alveolar epithelial cells after membrane fusion by the virus. Inside the host cell the virus replicates itself thanks to an active RNA polymerase. This enzyme synthesizes a negative-sense antigenome that is then used as a template to create a new positive-sense viral genome, which in turn cause the cell cytoplasm to synthesize new proteins.

The M protein aids integration to the cellular endoplasmatic reticulum while the viral N protein binds the new genomic RNA. These new nucleocapsids are subsequently released in the ER membrane and moved to the lumen. Finally, using Golgi vesicles, they are moved to the cell membrane and released into the extracellular environment by exocytosis.

The newly formed virus particles are now able to attack the nearby epithelial cells and to spread among different people as part of respiratory droplets [6].

1.3 Pathophysiology

Although a lot has been learned about COVID-19's transmission and clinical presentation, less is known about its pathogenesis.

1.3.1 Asymptomatic phase

In the upper respiratory tract, SARS-CoV-2 binds to nasal epithelial cells after being acquired through respiratory aerosols. The most important receptor for viral entry into cells is ACE-2, which is seen to be highly expressed in adult nasal epithelial cells.

The second phase consist of local replication and propagation of the virus in addition to infecting the ciliated cells in the conducting airways. This stage lasts for a few days and the immune response is minimal. The patients are highly contagious despite having a low viral load at this time, and nasal swab tests can already identify the virus.

1.3.2 Invasion and infection of the upper respiratory tract

Thanks to the conducting airways, the virus moves from the nasal epithelium to the upper respiratory tract. The clinical outcome is strictly connected to the upper airway involvement, so the main symptoms are fever, lethargy and dry cough. The release of interferons (IFN- β and IFN- λ) and C-X-C motif chemokine ligand 10 (CXCL-10) from virus-infected cells contributes to a stronger immune response at this phase. In most people the strong immune response is adequate to stop the spread of infection, therefore the infection does not get past this stage.

1.3.3 Involvement of the lower respiratory tract and progression to acute respiratory distress syndrome (ARDS)

The disease aggravates in around one-fifth of all infected patients, who then experience severe symptoms. Viral replication begins once the virus reaches type 2 alveolar epithelial cells via the host receptor ACE-2 and begins to create new viral nucleocapsids. As a result, virus-laden pneumocytes release interleukins (IL-1, IL-6, IL-8, IL-120, and IL-12), tumour necrosis factor (TNF), IFN, and IFN-, CXCL-10, monocyte chemoattractant protein-1 (MCP-1) and macrophage inflammatory protein-1α (MIP-1α). The neutrophils, CD4 helper T cells and CD8 cytotoxic T cells are chemoattracted by this "cytokine storm" leading to an infiltration of mononuclear cells in the lung tissue. These cells are in charge of warding off the virus, but in doing so they also cause inflammation and lung damage. This injury in lung tissue can cause apoptosis of the host cells with the release of new virus particles that were stored into the cells. The injury caused by the trapped inflammatory cells and the viral replication terminate in the loss of both type 1 and type 2 pneumocytes. This condition shows up as a diffuse alveolar damage which ultimately leads to an acute respiratory distress syndrome [6].

1.4 Viral transmission and clinical features

The respiratory droplet transmission of the COVID-19 virus happens when a person comes into close contact with someone infected who is vigorously coughing or sneezing. If mucosal surfaces—like eyes, nose, and mouth—are exposed to incoming infectious respiratory droplets the disease can spread among people. The virus can also spread through objects like bedsheets, blankets, culinary utensils, thermometers, and stethoscopes that have been touched or used by an infected person. Except for certain procedures that produce aerosols (endotracheal intubation, bronchoscopy, open suctioning, nebulization with oxygen,

bronchodilators, or steroids, bag and mask ventilation prior to intubation, tracheostomy, and cardiopulmonary resuscitation) there's no evidence of airborne transmission of COVID-19.

COVID-19 incubation period usually lasts 5-6 days and is defined like the time between virus exposure and the beginning of symptoms. In a few cases this temporal gap can extend up to 14 days. During this phase, named "pre-symptomatic" stage, infected people are already contagious and can spread the virus. COVID-19 patients are mainly in the 40-to-70-year age range and the most common symptoms are fever, body pains, dyspnoea, malaise, and dry cough, though they may also show up asymptomatic or get a mild, moderate, or severe disease [6].

Table 1: Clinical spectrum of COVID-19 disease.

Severity of disease	**Presentation**
Asymptomatic	• No clinical symptoms • Positive nasal swab test • Normal chest X-ray
Mild illness	• Fever, sore throat, dry cough, malaise and body aches or • Nausea, vomiting, abdominal pain, loose stools
Moderate illness	• Symptoms of pneumonia (persistent fever and cough) • without hypoxemia • Significant lesions on high-resolution CT chest
Severe illness	• Pneumonia with hypoxemia (SpO2<92%)
Critical state	• Acute respiratory distress syndrome, along with shock, coagulation defects, encephalopathy, heart failure and acute kidney injury

1.5 Diagnosis and imaging

1.5.1 Molecular tests (RT-PCR)

Naso-pharyngeal and oropharyngeal swabs are the most common tests to collect samples from the upper respiratory tract, whereas expectorated sputum and bronchoalveolar lavage (only for mechanically ventilated patients) are used for the lower respiratory tract. Samples are kept at 4°C before being sent to the lab, where the viral genetic material is amplified using a reverse-transcription procedure. This technology, using existing viral RNA, creates a double-stranded DNA molecule, either through reverse-transcription PCR (RT-PCR) or a real-time (quantitative PCR). The amplified genetic material is then used to identify SARS-CoV-2 genetic code. Both in

the event of a positive test result and in order to confirm the viral clearance in COVID-19 previous positive cases, it is indicated to repeat the procedure. These tests have a not very high sensitivity: only 53.3% of COVID-19-confirmed patients had positive oropharyngeal swabs and roughly 71% of patients tested positive for RT-PCR using expectorated sputum. Usually the RT-PCR tests are able to identify a positive case after 2 to 8 days.

1.5.2 Serology

Despite all efforts, no reliable antibody test has been developed. The US Vaccine Study Centre at the National Institutes of Health are working on a new kind of test, which appears to be excellent in both specificity (higher than 99%) and a sensitivity (about 96%), according to continuing research by the centres for disease control and prevention (CDC).

1.5.3 Blood tests

In addition to specific diagnostic tests, there are also many typical modifications in the blood tests. In many instances, a normal or decreased white blood cell count (also known as lymphopenia) can be seen, which is also thought to be a sign of worse prognosis. Increased levels of aspartate aminotransferase, alanine amino-transferase, creatine kinase (CK MB and CK MM), lactate dehydrogenase, and C reactive protein are very likely to be shown. Some patients may present higher neutrophil-to-lymphocyte ratio and D-dimer levels. In extreme situations, abnormalities in coagulation might be seen, as shown by an increase in prothrombin time and INR.

1.5.4 Chest X-ray

In the early stages of the disease, a chest X-ray is not useful and may not reveal any major alteration. In later stages the damage starts to expand and can be seen like bilateral multifocal alveolar opacities or linked to pleural effusion.

1.5.5 CT

Thanks to its high sensitivity, starting from the beginning of the disease, high-resolution CT (HRCT) is considered the gold standard for the diagnosis of COVID-19 pneumonia. The lower lobes are more frequently involved, and the most common features include multifocal bilateral "ground-glass" patches together with consolidation and a patchy peripheral dispersion. It is possible to find the so called "reversed halo sign" (RHS) which is defined as a central area of patchy opacities within a peripheral rim of consolidation. Pleural effusion, cavitation, calcification, and lymphadenopathy can be found either alternatively or together in the same scenario [6].

Table 2: Investigations for COVID-19.

Investigation	Remarks
Basic blood work	• Decreased WBC count as well as lymphopenia • Increased levels of AST and ALT, LDH and CRP • Increased D-dimer • Increased PT/INR
Molecular testing via RT-PCR	• Techniques employed are RT-PCR and rRT-PCR which amplify viral genetic material obtained via nasal swab • Poor sensitivity • Repeat testing required for verification of viral clearance
Chest X-ray	• No significant findings early in the disease • Bilateral patchy opacities in advanced disease
HRCT chest	• Multifocal bilateral 'ground or ground-glass' areas associated with consolidation areas with patchy distribution • 'Reverse halo' sign • Cavitation, calcification and lymphadenopathy • High sensitivity for COVID-19 diagnosis
Serology/antibody testing	• Further research still required for a proper/sensitive antibody test

1.6 Neurological manifestations of long COVID syndrome

COVID-19 survivors appear to have a very high prevalence of protracted neurological problems, nevertheless no certain criteria for the diagnosis of "long-COVID" have been accessed.

As a multi-organ disorder "long-COVID" implies a broad range of clinical symptoms, such as pulmonary, cardiovascular, immunological, endocrine, renal, hematologic, gastrointestinal, dermatologic, mental or neurological clinical manifestations. Although about one-third of patients with antecedent COVID-19 severe acute respiratory syndrome show an involvement of the central or peripheral nervous system, observational studies based on patient-reported data reveal a roughly threefold higher incidence of neurological symptoms.

Headaches, "brain fog," myalgias, cognitive impairment, fatigue, abnormalities in sleep, mood, smell or taste, sensorimotor impairments, and dysautonomia are often involved in "long-COVID" as neurological symptoms. Neuroinflammation and oxidative stress are the main mechanisms in the propagation of neurological "long-

COVID" sequelae, but there isn't enough information on the pathophysiological patterns yet.

The "long-COVID" syndrome has been linked to a wide range of neurological symptoms that affect both the central nervous system (CNS) and peripheral nervous system (PNS). However, it's important to note that non-specific symptoms like fatigue, "brain fog", post-exercise malaise and sleep disorders may be epiphenomena of underlying respiratory, cardiovascular, endocrine, renal, hematologic, autoimmune, or psychiatric damages.

Similarly, neurological symptoms are deeply connected to "long-COVID" manifestations due to other systems.

Table 3: Neurological manifestations of 'long-COVID' syndrome, according to the localization in the nervous system.

Localization in the nervous system	Neurological symptoms
Central nervous system	• Fatigue • 'Brain fog' • Headache • Sleep disorders • Cognitive impairment • Emotional/mood disorders • Dizziness • Dysautonomia
Peripheral nervous system	• Muscle weakness • Myalgias • Hyposmia • Hypogeusia • Hearing loss/tinnitus • Sensorimotor deficits (hypoesthesia, dysesthesia, tremor)

1.6.1 Pathophysiological mechanisms underlying neurological manifestations of 'long-COVID'

COVID-19 can lead to both acute and chronic damage in the neurological system.

As regards acute COVID-19 manifestations, they seem to be linked to several overlapping pathogenetic mechanisms: viral neuroinvasion accompanied by aberrant neuroimmunological responses, blood-brain barrier dysfunction due to increasing endotheliopathy, coagulopathies that cause hypoxic-ischemic neuronal injury, metabolic imbalances, oxidative stress cascades and cellular apoptosis [7].

In order to understand the molecular basis for neurological "long-COVID" sequelae further studies needs to be done in the future.

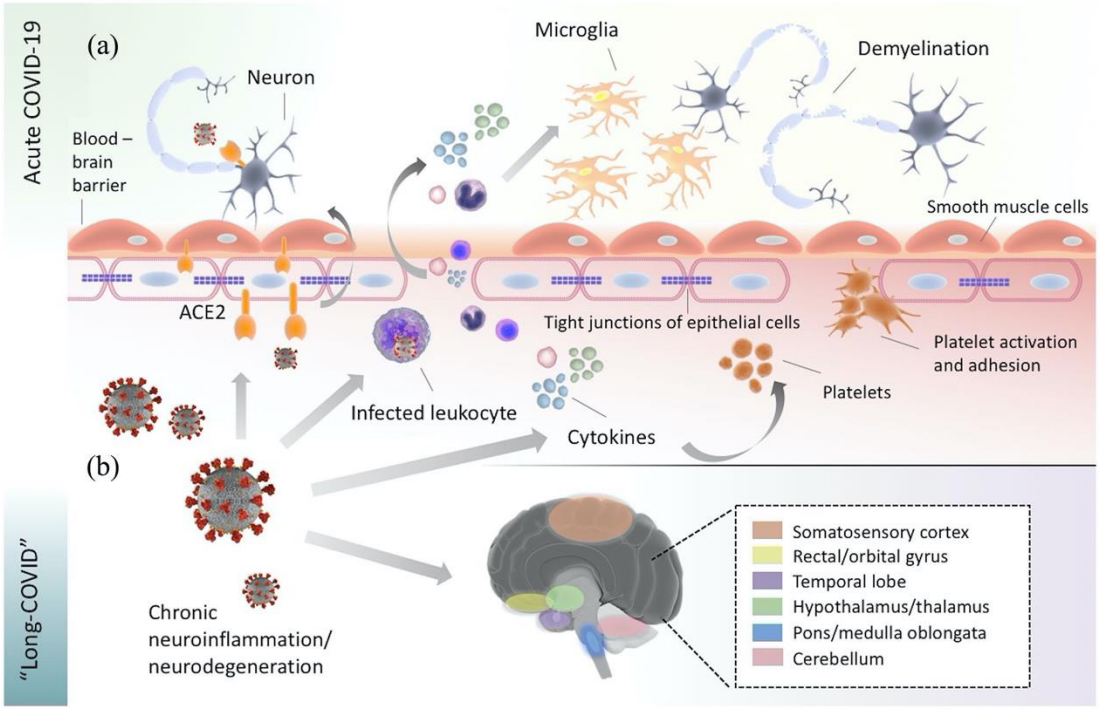

Figure 3: Potential pathophysiological mechanisms implicated in the manifestation of acute and 'long-COVID' [7].

1.7 Behavioural markers

As previously stated, it has been noted that COVID-19 patients can have neurological and cognitive repercussions. As a result, it is now necessary to use behavioural markers to gather information about them that is both objective and methodical.

Specifically, the behavioural markers being studied are:

- response speed and accuracy to visual and auditory stimuli
- the "temporal binding window" of sensory integration
- sensory adaptation responses
- retention of verbal information.

Each behavioural test that is used in the application, in order to obtain the required markers, is described in full in the following paragraphs.

1.7.1 Memory Experiment

The main objective of this experiment is the observation of the implicit memory of the Event-Related Potentials (ERPs) on the dataset of item recognition confidence answers that Covid-19 patients provide [8].

The experiment is divided into 2 distinct tasks:

- Word categorisation task
- Word recognition task

In both cases the stimuli are words selected from the *Medical Research Council Psycholinguistics Database*, presented in uppercase letters and with an average number of 4.89 letters in each word (min=3, max=8).

Word categorisation task is further divided into 2 blocks, each of which composed of 65 words; the participant must respond with a yes or no to indicate whether the word represents a living thing or not, or an artificial thing or not.

In *word recognition task* the 130 words that were displayed in the first challenge are combined with 70 completely new ones; these are taken from the same database as well. There are 4 blocks in which these words are given, with 50 of them in each block. For each word, participants are firstly asked to indicate if it comes from the ones used in the word categorisation task on a 5-point scale:

1. I'm sure it's new
2. Probably new
3. Not sure
4. Probably old
5. I'm sure it's old

Next, if they answered 4 or 5 (i.e., they believe they have already seen that word) they are also asked to indicate from which of the two previous sub tasks (alive vs not alive, manmade vs not manmade) it comes from, on a similar 5 points scale:

1. I'm sure it's from alive list
2. Probably from alive list
3. Guessing
4. Probably from manmade list
5. I'm sure it's from manmade list

1.7.2 Flash beep task

This experiment corresponds to an audio-visual temporal order judgement (TOJ) task, whose main objective is to assess multisensory integrity. The participant is asked to choose which of two events occurred first after being exposed to an auditory stimulus (a beep) and a visual stimulus (a flash). In particular, there is a *stimulus onset asynchrony* (SOA) between the flash and the beep which can assume one of the following 15 values in ms: -400, -240, -200, -160, -120, -80, -40, 0, 40, 80, 120, 160, 200, 240, 400. Here, the positive levels show instances when the visual flash was delivered first, while the negative levels show instances where the auditory beep was provided first. The total number of trials is 120.

1.7.3 Loudness perception task

The loudness perception task seeks to determine the extent to which Covid-19 patients' sensory adaptation has declined. Loudness adaptation is the subjective

decrease in loudness that occurs over time when a quiet and continuous pure tone, typically below 40dB SPL, is played monaurally [9]. In practice, a high-pitched beep embedded in white noise is played for around 3 minutes. The user is then asked, every 20 seconds for a total of 9 times, to indicate whether the high-pitched beep he is hearing is as loud as or quieter than the reference beep.

1.7.4 Target detection task

This is a *Posner Task* which is a common paradigm for studying visual attention. The user has detected when a target stimulus is presented and respond as quickly as possible. Specifically, the stimulus is a circle which can be darker on top or on bottom; participant, indeed, has to indicate where the displayed circle was darker.

The stimulus appears either on the left side or the right of the screen, and participants are cued before the stimulus appears. Cues can be valid or invalid: an invalid cue is "incorrect" if it, for instance, precedes a target on the right side of the screen while pointing to the left; vice versa, an arrow pointing to the left side of the screen that comes before a target on the left is a valid cue. Occasionally, the target stimulus comes with a white noise sound that acts as a distractor stimulus.

The stimuli are presented in 2 blocks, each composed ideally by 128 trials; however, at the end of each block the wrongly answered trials are presented again, until all user's answers are correct.

1.7.5 Movement perception task

The movement perception test, like the target detection experiment, tries to measure multisensory awareness. Movement perception, in particular, is the process of inferring the speed and direction of elements in a scene based on visual, vestibular and proprioceptive inputs. In order to test it, the experiment is designed in the following way:

- At the beginning, the participant is asked to stare at a fixation cross represented for a short period of time.
- Thereafter, a moving pattern will appear; however, after about 3 seconds a brief test stimulus will come out with motion in a different direction.

User is asked to indicate the motion direction that he perceived. The total number of trials that made up the experiment is 10.

Chapter II

Project description

2.1 Introduction

The goal of this study is to provide a method for assessing COVID-19's effects on several aspects of cognitive function and neural integrity in order to ascertain whether or not these changes are connected to the persistence of symptoms. We want to create a system that will enable everyone over the age of 18 to take part in a number of activities, the outcomes of which will be evaluated by researchers and medical specialists. It should be also noted that the final data will be more reflective of the world as a whole if people from many countries are able to participate than if it came from just one location.

2.1.1 Purpose of the system

Users who register for the service will have access to a cloud-based platform where they may perform seven different tasks in any order. The normative sample's participants will only be tested once; thereafter, their resulting answers will be utilized to create reliable norms for each activity. Researchers and lab assistants will have access to data from all lab activities in such a way that they will be able to make decisions and develop conclusions based on the information at hand. We employed common perceptual and cognitive research tasks as the foundation for the experimental tasks in order to examine both single and multimodal processing. Examples include the capacity to assess the flow of events or spot mistakes.

2.1.2 Scope of the project

The web-based application has to work with many other current browser versions (e.g., Chrome, Firefox, and Edge). The required features for the participants, the administrator, lab assistants, and researchers must be included. An important requirement is that each user will only ever need to finish a task once. The administrator is in charge of users' management while the lab assistant is able to access to the answers; a researcher can also act as a lab assistant and administrator. Since it is anticipated that in the future tests would be done from anywhere in the world and in more remote locations, it is imperative that the app continues to operate in regions with poor internet connections.

2.2 Observations and feedback

As mentioned earlier, when I started working on the application, it had already been completed and deployed in its *Alpha* version. This made it possible for users to test it

and pass their feedback to the technical team; moreover, it allowed a comparison against the functional and non-functional requirements listed in the previous paragraphs. As a result, a document containing all required interventions was produced by the psychologists (which designed the implemented tests, and therefore were in charge of testing) and researchers belonging to the team. Bugs, requested modifications to the current interface with a more visually beautiful and straightforward design, an additional task, and more features for each experiment were among the observations and improvements noted.

2.2.1 Improvements

The above-named document, hence, was filled with a list of modifications and improvements, which could be created thanks to the testing of the application, the comments from the psychology department at the University of Essex, and the individual research on the neuropsychological tests used to assess cognitive impairment. It is worth noting that this process was cyclic, since fresh comments were periodically received while some other were removed (i.e., they were solved or simply not wanted anymore), consequently, the document was constantly updated. However, the main things that needed to be fixed have been grouped in the following tables, according to their priority. Psychologists, in fact, divided required updates according to a priority, which followed the following scheme:

- High priority work included most of the changes for the cognitive tasks and those that were supposed to take more time to implement;
- Medium priority involved the required modifications to the questionnaires and registration page;
- Low-priority tasks referred to minor visual alterations.

Table 4: High priority issues classified according to task.

Task	To fix
General	• Allow multiple attempts for every task, participants must do them in their own time and order. • Allow a practice trial for every task. • All tasks must end with a screen that says “You have successfully completed this task” and redirect participants back to the Home page. • The full application should be translated when choosing a different language in the dedicate language switch. • Remove the different types of users and focus only on the participants of the study.

Memory Experiment	• Add a general overview of the task before the practice attempt for both tasks, word categorisation and recognition. • Word lists must be changed to a list provided by the psychology team. These should include the correct responses. Categorisation experiment: • Add a break time after the 22nd and 44th question. • Add "correct" or "incorrect" answer, after each practice block and at the end of the whole task. Recognition experiment: • Reduce blocks to 4 rounds each and it must be of 50 words. • Add a source recognition (alive vs. manmade) task after each item recognition (old vs. new) task. • Source memory screen option, where participants indicate which list it was from, should only come up if they indicated "definitely old", "probably old" or "not sure". If they have indicated "probably" or "definitely new", they should go straight to the next word.
Flash beep task	• Add information about the number of trials in each round. • Add a back button option to be able to return to the instructions when doing the practice trial. • Match instructions of indicating left arrow for sound and right arrow for light with the control options. • Control options should be used either through the keyboard or click the appropriate response box. • The participant should be able to see their performance (correct • and incorrect counts) after finishing the practice attempt. • Add a break after every 40 trials with feedback on the participant's performance. This should be based on the accuracy, and it should be given as a percentage. • The data that needs to be collected should be the responses from the start of the first stimulus.

	• Change instructions to make sure the participant understands that what is important is their accuracy not their speed.
Target detection task	• Remove the "Oddball task" and change in order to fit the guidelines of a Posner cueing task. • Allow participants to select one audio device, either headphones or external speakers and present them one below the other. • Add extra trials, when the participant makes a mistake, at the end of each block. • Limit the movement of the elements on the screen during practice. • Add a start button for the participant to take their own time to read the full instructions.
Loudness Perception	• Allow participants to select one audio device, either headphones or external speakers and present them one below the other. • If the participant selects external speakers, show another screen in between for further instructions. • Add a flash effect once the response buttons (10%, 20%, etc.) come up on the screen so that participants notice them. • The sound stimulus must be constant during time, so replace current audio file (which decreases its intensity periodically).
Movement Perception	• Create a new task to analyse the visual adaptation of participants based on their ability to judge the direction of moving dots. • Add feedback on the practice attempt, to indicate if the arrows went up or down. • Add feedback after the 10th trial.

Table 5: Medium priority issues classified according to task.

Task	**To fix**
Registration page	• **Change the participant information and consent sheet.** • **Send to participants the new information with their registration confirmation email.**
Personality survey	• Change the answer choices from "Strongly Agree" and "Strongly Disagree" to "Yes" and "No". If the participant clicks "Yes", it scores 1, if they click "No", it scores 0.

COVID survey	• Participants should only see the next question in the survey after answering the previous one. • If in the first question regarding a participant's experience with COVID, they answer "No", add a pop-up window that asks to confirm this, in case someone clicked accidentally. If they confirm access to other tasks should be prevented.
Demographics survey	• Participants should only see the next question in the survey after answering the previous one. • Add a "Prefer not to say" option for the question "Current country of residence" and "Ethnic background". • Add more educational background options for participants who are not from the United Kingdom.

Table 6: Low priority issues classified according to task.

Task	**To fix**
General	• Change the "Welcome to the study!" formatting so that it is all one line and in black on white. • Change text size in response options and instructions to make sure older people, impairment vision or fatigue is not an issue when reading.
Home page	• Add the time each task takes to its respective button. • Add the "Happy Again" logo to replace the progress circle. • Add an overview text regarding the place and conditions that are optimal to take the test, as well as a recommendation on where to start the experiment. • Remove the "Go to current experiment" button.
Memory experiment	• Change spacing between each instruction to make it even. • Capitalise all words that are titles. • Remove unnecessary instructions, make them shorter, and easy to follow. Recognition experiment:

	• Change instructions for animate trials, it should ask if the word is “alive” rather than a “living thing”. • Change “Guessing” to “Guessing / Neither list” in the source recognition (alive vs manmade) task to help participants identify words that are judged as new.
Target detection	• Round up decimal places of breaktime feedback.
Loudness perception	• Change name to “Loudness Perception” instead of “Sensory Adaptation”. • Make instructions narrower in width to discourage head movements and keep them through all the duration of the task.
Covid survey	• Add “Covid survey” to the main screen of this questionnaire.

2.3 Functional Requirements

Broadly speaking, it can be stated that functional requirements answer the question: *"What should the software system do?"*. In technical terms, they are defined as:

- Statement of services and functionalities that the system should provide.
- Definition of the behaviour of the system in response to particular situations.
- Definition of the reaction of the system to particular inputs.
- Statement of what the system should not do, in relevant cases.

The application's main features identified in the preliminary study phase have been merged with the updated needs, obtained by the analysis of the above-described feedback received, and have been summarized in the following table.

Table 7: Functional requirements.

Identifier	Description
FR1	The application must handle all participants' registration.
FR2	The system must provide all participants with information on the study and a consent form before registering.
FR3	The system should ask participants to accept the consent form to allow their registration.
FR4	The system must email the participant to confirm their email address and validate their account.
FR5	The system must handle resending the confirmation email if it is not received or participant cannot find it.
FR6	The application must allow all registered participants to sign in.
FR7	The application must allow participants to reset their password if they have forgotten it.
FR8	The application allows participants to log in and off at any time.
FR9	The system must detect and record the metadata of each participant's device.
FR10	The application should present participants a battery of tests, including questionnaires and tasks, and allow them to undergo each one.
FR11	The application allows each participant to return to the home page at any time.
FR12	The application must present the estimated time for each experiment.
FR13	The application must let participants undertake the" Demographics" questionnaire and record the results.
FR14	The application will only allow participants to go to the next question of the" Demographics" questionnaire after answering the current one.

FR15	The application must let participants undertake the” Covid” survey and record the results.
FR16	The application will only allow participants to go to the next question of the” Covid” survey after answering the current one.
FR17	The system must block the participant if they declare that they have not had Covid-19.
FR18	The application must let participants undertake the” Personality” survey and record the results.
FR19	The application will show a detailed instruction description for the first part of the” Word Categorisation” task before allowing participants to start.
FR20	The application will grant participants a practising trial for the first part of the” Word Categorisation” task.
FR21	The application will allow participants to undertake the first part of the” Word Categorisation” task after the demo and save the results.
FR22	The application will show a detailed instruction description for the second part of the” Word Categorisation” task before allowing participants to start.
FR23	The application will allow participants to undertake the second part of the” Word Categorisation” task and save the results.
FR24	The application will show a detailed instruction description for the” Flash Beep” task and will ask to select an audio device before allowing participants to start.
FR25	The application will grant participants a practising trial for the” Flash Beep” task.
FR26	The application will allow participants to undertake the” Flash Beep” task after the demo and save the results.
FR27	The application will show a detailed instruction description for the” Loudness Perception” task and will ask to select an audio device before allowing participants to start.
FR28	The application will grant participants a practising trial for the” Loudness Perception” task.
FR29	The application will allow participants to undertake the” Loudness Perception” task after the demo and save the results.
FR30	The application will show a detailed instruction description for the” Target Detection” task and will ask to select an audio device before allowing participants to start.
FR31	The application will grant participants a practising trial for the” Target Detection” task.
FR32	The application will allow participants to undertake the” Target Detection” task after the demo and save the results.

FR33	The application will show a detailed instruction description for the" Movement Perception" task before allowing participants to start.
FR34	The application will ask participants to configure the screen through a size matching exercise before allowing participants to start the" Movement Perception" task.
FR35	The application will grant participants a practising trial for the" Movement Perception" task.
FR36	The application will allow participants to undertake the" Movement Perception" task after the demo and save the results.
FR37	The application must permit participants to take the battery of tests as often as they want.
FR38	The application presents a screen informing the participant when they have completed a task.
FR39	The application should have an introductory window presenting the study information to the participants.

2.4 Non-functional requirements

The features, constraints, and characteristics of the program are referred to as non-functional requirements. They define standards that can be used to evaluate a system's performance, in contrast to the functional ones which specify particular requirements that a web application must follow in order to provide a positive user experience and be simple to use. The majority of the criteria were left unchanged with respect to their first version, because little to no justification was provided for their modification by the project's analysis and feedback. The current schedule for the system's operation, revision, and transition is shown in the tables below.

2.4.1 Performance

Performance requirement refers to how quickly a software system, or a certain component of it, reacts to particular user actions while handling a specific workload.

Table 8: NFR01.

Attribute	**Description**
Identifier	NFR01
Name	Throughput
Description	The number of requests the system is able to process in a time window.
Metric target	30,000 requests/minute
Priority	Medium

Table 9: NFR02.

Attribute	Description
Identifier	NFR02
Name	Front-end response time
Description	The time it takes to perform an action that does not involve a connection to the back end.
Metric target	1 second
Priority	Medium

Table 10: NFR03.

Attribute	Description
Identifier	NFR03
Name	Connection response time
Description	The timeout time when performing an action involving connection to the back end independent of load.
Metric target	5 seconds
Priority	Medium

2.4.2 Scalability

Scalability measures the highest workloads that the system can handle while still delivering the required levels of performance.

Table 11: NFR04.

Attribute	Description
Identifier	NFR04
Name	Auto scaling
Description	The server must scale dynamically as client-side requests increase.
Metric target	Support 1000 requests/minute and scale up to 30,000 requests per minute.
Priority	High

2.4.3 Resource Management

Resource management describes the needs related to the memory that the program requires.

Table 12: NFR05.

Attribute	Description
Identifier	NFR05
Name	Memory limits
Description	The maximum amount of memory that should be used by the web application on the client side.
Metric target	2 GB of RAM
Priority	High

2.4.4 Security

Security is a non-functional requirement that guarantees that every piece of data inside the system will be safe from malware assaults and unauthorized access.

Table 13: NFR06.

Attribute	Description
Identifier	NFR06
Name	Secure connection to backend
Description	Requests to the backend must use HTTPS security protocol and HTTP requests must be redirected to use HTTPS.
Metric target	None
Priority	High

Table 14: NFR07.

Attribute	Description
Identifier	NFR07
Name	Secure authentication token
Description	The token used to authenticate users must be generated with a secure encryption algorithm.
Metric target	None
Priority	High

2.4.5 Availability

Availability refers to the likelihood that a user will be able to use the system at a specific time.

Table 15: NFR08.

Attribute	Description
Identifier	NFR08
Name	Server availability
Description	The server must be able to be available most of the time, with almost no downtimes.
Metric target	Availability percentage of 99.5%
Priority	High

2.4.6 Reliability

The reliability of a system or component indicates the likelihood that it will function well for a predetermined amount of time under specific circumstances.

Table 16: NFR09.

Attribute	Description
Identifier	NFR09
Name	Internet Connection Interruption Tolerance
Description	The system continues to work with local interactions even after a momentary internet connection interruption.
Metric target	None
Priority	Medium

2.4.7 Maintainability

Maintainability is the length of time needed to fix a problem, make changes to a solution or its component to improve performance or other characteristics, or adapt to a changing environment.

Table 17: NFR10.

Attribute	Description
Identifier	NFR10
Name	Logs
Description	The system must store the history of the actions and operations being performed so that they are sent in case of error.
Metric target	None
Priority	High

Table 18: NFR11.

Attribute	Description
Identifier	NFR11
Name	Version support
Description	The system must not stop working when upgrading the system (making changes).
Metric target	None
Priority	High

2.4.8 Usability

Usability indicates how effectively and easily users can learn and use a system.

Table 19: NFR12.

Attribute	Description
Identifier	NFR12
Name	User-friendly interface for participant
Description	All interfaces and activities that can be performed by the participant in the application should be intuitive and easy to use.
Metric target	None
Priority	High

Table 20: NFR13.

Attribute	Description
Identifier	NFR13
Name	User-friendly interface for the administrator, researcher and lab assistant
Description	All the interfaces and activities that the administrative user can perform in your application should be intuitive and easy to use.
Metric target	None
Priority	Medium

2.4.9 Portability

Portability determines how a system, or its element can be launched within one environment or another.

Table 21: NFR14.

Attribute	Description
Identifier	NFR14
Name	Timestamps in international format
Description	All timestamps collected in the application must be in UTC format when stored in the database.
Metric target	None
Priority	High

Table 22: NFR15.

Attribute	Description
Identifier	NFR15
Name	Multi-platform support
Description	The system must work independently of the operating system and on Chrome, Firefox, MS Edge and Safari browsers.
Metric target	None
Priority	High

2.4.10 Interoperability

Interoperability is the capacity of a system to communicate data and exchange information with other systems and external hardware with ease.

Table 23: NFR16.

Attribute	Description
Identifier	NFR16
Name	Connection to backend server API
Description	The server must be able to query the backend server and have the permissions to modify the existing database.
Metric target	None
Priority	High

Chapter III

Design of the solution

This chapter describes in detail the design steps executed for the implementation of the desired web application. Hence, in first place it defines the adopted methodology; in second place it presents the use cases, which determined the main success scenarios, and the layering analysis, which allowed the evaluation, and then the selection, of the appropriate components of the system architecture.

3.1 Methodology

The software release life cycle (SRLC) is a series of benchmarks that outline the many phases of a piece of software's life cycle or sequential release timetable, from conception through its ultimate fully baked release. The team I was part of started its work on the web application right after the Alpha stage was completed, thus it had been handed out to the psychology team for the sake of being thoroughly tested. This project's aim was to develop, prototype, and test remedies, in order to address the problems caused by the previous version of the app, obtaining a product that is a candidate for release and will eventually become the stable version.

The *Design Thinking* methodology is used in this work to accomplish the above-mentioned goals. Understanding users' needs and concerns is the main focus of this procedure. The goal of using this kind of process is to identify the best method for collecting the necessary data on brain markers. Therefore, the primary job was to devise solutions able to address the issues and bugs the web application presented, as well as the input the psychology department supplied.

There are five steps of the Design Thinking process to take into consideration in order to develop and deepen a relationship between the web application and the users:

1. Empathising with the problem
2. Defining changes and problems to improve
3. Ideating creative solutions and action plan
4. Prototyping the proposed solutions
5. Testing prototype and validate

Figure 4 depicts the thesis's step-by-step methodology. The graphic illustrates a top-down analysis, which includes breaking issues down into smaller procedures and activities in order to solve them. Each mission prioritizes adding value, enhancing user experience overall, and gathering pertinent data to research event-related potentials (ERPs).

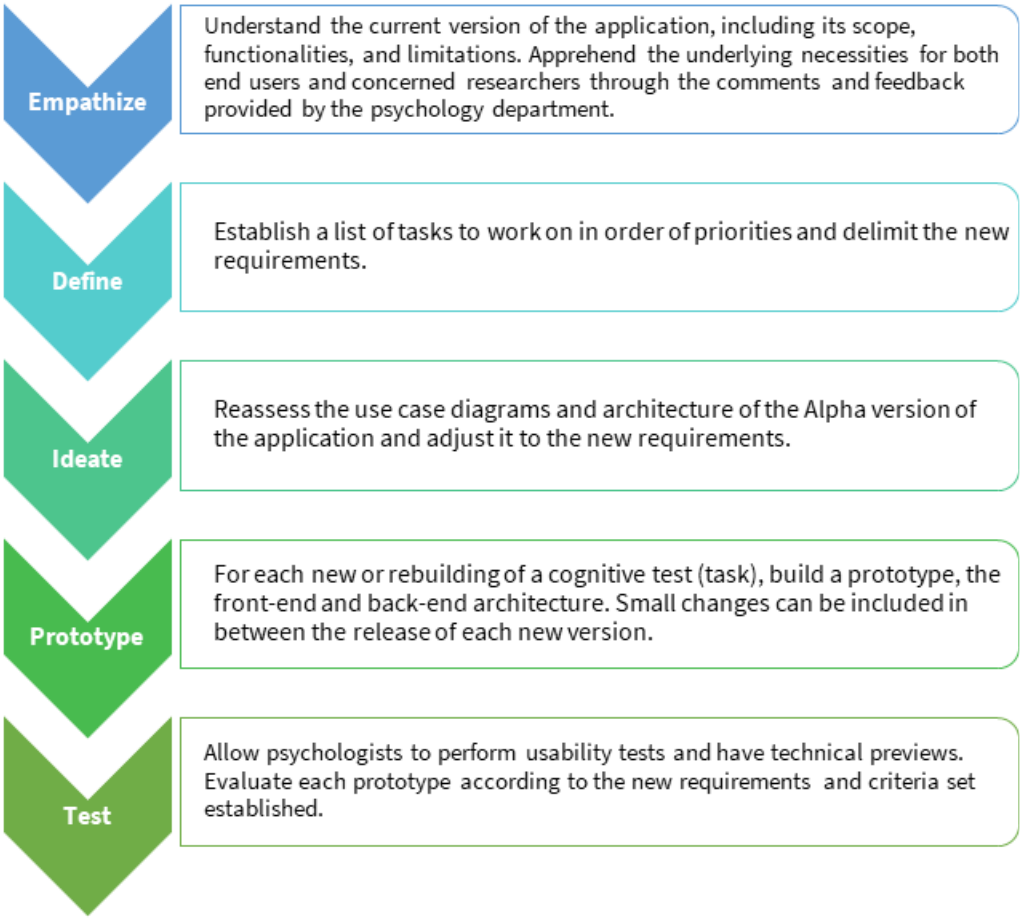

Figure 4: Design Thinking steps for this project [10].

3.2 Use cases

A use case is a series of actions or event steps that often describes how a role - referred to as an actor in the Unified Modelling Language (UML) - interacts with a system to accomplish a task, and which produces an observable result that contributes to its goals. Actors represent the role that human users or other systems have in the interaction [11]. The use cases outline the platform's functional requirements and serve as the foundation for gradual increment of the system supported features. Furthermore, they are often graphically depicted as use case diagrams; this may aid in giving a higher-level perspective on the considered system. The tables below give the comprehensive description of the majority of the identified use cases, together with the corresponding functional requirements, while figure 5 depicts the resulting diagram. It should be pointed out that some of these use cases are the product of a review of previously identified ones (during study phase executed by previous teams), which were updated to support the changed requirements, while some others have been designed from scratch, to include the newly requested features.

Table 24: UC1 - Register.

ID	**UC1**
Name	Register
Functional requirement	FR1, FR2, FR3, FR4, and FR5.
Goal	The participant is able to register for the study.
Pre-condition	The participant was shown a Welcome page.
Post-condition	The system records the participant's data in a secure stand-alone database under a pseudonym.
Main success scenario	1. The participant clicks on the "Register" button. 2. The information sheet on the study is presented. 3. The participant reads the consent form and agrees to participate in the study. 4. The participant clicks again on the "Register" button. 5. The registration screen is shown. 6. The participant fills all the entries, name, email, and password, with valid data clicks on the "Confirm" button. 7. The system stores the entered user data into a secure database. 8. The application informs the participant that an email was sent to the provided address to verify their account. 9. The participant clicks on the link and validates their account. 10. The redirected page informs the participant about their successful registration.
Includes/Extends	Validation: 6.a.1. The participant's email has already been used and it's recorded in the database. The application informs the participant about their existing account. 8.a.1. The participant is unable to find the email sent. The application offers to click on "Resend email".

Table 25: UC2A - Log in.

ID	**UC2A**
Name	Log in
Functional requirement	FR5, FR6, FR7, and FR8.
Goal	The participant is able to log into the application.
Pre-condition	The participant is successfully registered.
Post-condition	The system records the active session of the participant and redirects them to the Home page.
Main success scenario	1. The participant clicks on the "Log On" button. 2. The log on screen is shown. 3. The participant fills all the entries, email and password. 4. The system authenticates the participant's entered data. 5. The participant is logged in and presented the Home page.
Includes/Extends	Password recovery: 3.a.1. The participant does not remember their password. The application shows a "Forgot password" button to click on. 3.a.2 The application presents a screen where the participant can enter their email and establish a new password in case they need to. Authentication: 4.a.1. The system is unable to find the participant's data. The application displays a "No user created with the inserted email" message. 4.a.2. The entered password does not match the one registered. The application displays a "No user created with the inserted email" message.

Table 26: UC2B – Log Out.

ID	**UC2B**
Name	Log Out
Functional requirement	FR8.
Goal	The participant is able to log out from the application.
Pre-condition	The participant is logged in.
Post-condition	The system records the ending of the session, and the participant is redirected to the Log on page.
Main success scenario	1. The participant clicks on the “Log Out” button. 2. The system stores the ending session time. 3. The application shows the “Log On” screen.
Includes/Extends	None

Table 27: UC3 - Home.

ID	**UC3**
Name	Home
Functional requirement	FR9, FR10, FR11, FR12, and FR39.
Goal	The participant is able to see the initial instructions, a full menu of tasks and questionnaires to work on.
Pre-condition	The participant is logged in.
Post-condition	None
Main success scenario	1. The application presents the first instructions of the study, including recommendations for completing tasks. 2. The application displays 8 task options in the form of buttons, which each show the estimated time it requires to complete them.
Includes/Extends	None

Table 28: UC4 – Demographics.

ID	**UC4**
Name	Demographics
Functional requirement	FR13, FR14, FR37, and FR38.
Goal	The participant is able to accomplish the Demographics questionnaire.
Pre-condition	The participant is logged in.
Post-condition	The application stores the participant's demographics data in the database and will allow them to answer the questions again if they want to.
Main success scenario	1. The participant clicks on the "Demographics" button. 2. The application displays the first question and waits for the participant's response. 3. The application continues to the next question once the user has answered the previous, this is repeated for 8 questions. 4. The participant finishes answering all questions and clicks on the "Submit" button. 5. The system stores the responses. 6. The application displays a "You have completed the survey" message. 7. The participant is redirected to the Home page.
Includes/Extends	None

Table 29: UC5 - COVID.

ID	**UC5**
Name	COVID
Functional requirement	FR15, FR16, FR17, FR37, and FR38.
Goal	The participant is able to accomplish the COVID questionnaire.
Pre-condition	The participant is logged in.
Post-condition	The application stores the participant's history of COVID data in the database and will allow them to answer the questions again if they want to.
Main success scenario	1. The participant clicks on the "Covid Survey" button. 2. The application displays the first question "Have you had Covid-19?" and waits for the participant's response. 3. The application continues to the next question once the user has answered the

	previous, some of them will unchain more questions depending on the response. 4. The participant finishes answering all questions and clicks on the “Submit” button. 5. The system stores the responses. 6. The application displays a “You have completed the survey” message. 7. The participant is redirected to the Home page.
Includes/Extends	2.a.1. The participant answers negatively about ever having COVID-19. The application displays a warning screen to confirm this. 2.a.2. If the participant reconfirms that they have not had the illness, the application displays a message stating that they’re only looking for people who have had Covid-19 and blocks their account. 2.a.3. If the participant corrects their response and clicks on the “No” button, the application closes the warning message and allows the user to select another response.

Table 30: UC6 - Personality.

ID	**UC6**
Name	Personality
Functional requirement	FR18, FR37, and FR38.
Goal	The participant is able to accomplish the personality questionnaire.
Pre-condition	The participant is logged in.
Post-condition	The application stores the participant’s personality data in the database and will allow them to answer the questions again if they want to.
Main success scenario	1. The participant clicks on the “Personality Survey” button. 2. The application displays 33 yes or no questions. 3. The participant finishes answering all questions and clicks on the “Submit” button. 4. The system stores the responses. 5. The application displays a “You have completed the survey” message.

	6. The participant is redirected to the Home page.
Includes/Extends	None

Table 31: UC7 - Practice word categorisation.

ID	**UC7**
Name	Practice word categorisation
Functional requirement	FR19, FR20, and FR37.
Goal	The participant is able to have a practice trial before going into the real categorisation task.
Pre-condition	The participant is logged in.
Post-condition	None
Main success scenario	1. The participant clicks on the "Word Categorisation" button. 2. The application displays a screen with an overview description of the task. 3. The participant clicks on the "Start" button. 4. The application displays a screen with the instructions of the practice task. 5. The participant clicks on the "Start" button. 6. The system decides randomly with which block to start, "Alive" or "Manmade". 7. The application presents a screen with a word selected randomly from either the alive or manmade word bank, depending on which one was chosen in the step before. 8. For the alive block, the participant selects whether the word is a living thing or not by pressing the left arrow key or the right arrow one, respectively. For the manmade block the participant selects whether the word is a thing created by humans or not, using the same keys. 9. The system records the practice word, reaction timestamp, and participant's response. 10. The system checks if the response was correct or not. *Steps 7 through 10 are repeated for the 15 practice words.*

	11. The participant proceeds to the real task (UC8).
Includes/Extends	11.a.1 The participant proceeds to the word categorisation task.

Table 32: UC8 - Word categorisation.

ID	**UC8**
Name	Word categorisation
Functional requirement	FR19, FR21, FR37, and FR38.
Goal	The participant is able to accomplish the categorisation task.
Pre-condition	The participant is logged in and has gone through the practice trial.
Post-condition	The application stores the results, word, response time and decision, in the database.
Main success scenario	1. The application displays a screen with the instructions of the task. 2. The participant clicks on the “Start” button. 3. The system decides with which block to start, “Alive” or “Manmade”, depending on the one that previously was set for practice (UC7). 4. The application presents a screen with a word selected randomly from either the alive or manmade word bank, depending on the one set for practice (UC7). 5. For the alive block, the participant selects whether the word is a living thing or not by pressing the left arrow key or the right arrow one, respectively. For the manmade block, the participant selects whether the word is a thing created by humans or not, using the same keys. 6. The system records the word, reaction timestamp, and participant’s response. 7. Steps 4 through 7 are repeated for each 22 words, after a break time, until the 65th round. 8. The application grants the participant a break time which can be terminated whenever it’s clicked on the “Continue the experiment” button.

	9. The application displays a window with the results obtained, the count of correct responses. 10. The participant decides to continue the experiment. 11. After finishing the first block, the application returns to the practice trial (UC7) and switches to either alive or man-made to complete the second block.
Includes/Extends	11.a.1. The participant proceeds to the word recognition task.

Table 33: UC9 - Word recognition.

ID	**UC9**
Name	Word recognition
Functional requirement	FR22, FR23, FR37, and FR38.
Goal	The participant is able to accomplish the recognition task.
Pre-condition	The participant is logged in and has gone through the categorisation task.
Post-condition	The application stores the results, word, response time and decision, in the database.
Main success scenario	1. The participant clicks on the “Continue the experiment” button after finishing categorising. 2. The application displays a screen with an overview description of the task. 3. The participant clicks on the “Start” button. 4. The application presents a screen with a source recognition (alive vs. manmade) or item recognition (old vs. new) exercise. 5. The participant selects one of the options by pressing either 1, 2, 3, 4, or 5 (keyboard). 6. The application will return to step 4 and toggle between source or item recognition. This is repeated for 50 words of the words previously categorised. 7. The application informs the participant that they have finished a block and asks to continue when they feel ready. 8. The application repeats steps 3 through 7 until the 4 blocks are completed.

	9. The system stores the responses. 10. The application displays a "You have completed this task" message. 11. The participant is redirected to the Home page.
Includes/Extends	None

Table 34: UC10 - Practice flash beep.

ID	**UC10**
Name	Practice flash beep
Functional requirement	FR24, FR25, and FR37.
Goal	The participant is able to have a practice trial before going into the real flash beep task.
Pre-condition	The participant is logged in.
Post-condition	None
Main success scenario	1. The participant clicks on the "Flash Beep Task" button. 2. The application displays a screen with an overview description of the task. 3. The participant chooses an external audio device, either headphones or external speakers. 4. The participant clicks on the "Start" button. 5. The application displays a screen with the instructions of the practice task. 6. The participant can click the "Instructions" button to open a recapitulation window of the task overview or on the "Start" button. 7. The application displays a screen with summary instructions on top and white rectangle background on the bottom. 8. The application shows a flash and a beep, in random order. 9. The participant chooses which came first, flash or beep, by either clink on the respective buttons or using the keyboard arrows. *Steps 8 and 9 are repeated for 8 rounds.* 10. The participant clicks on "Terminate the practice" to continue.
Includes/Extends	None

Table 35: UC11 - Flash beep.

ID	**UC11**
Name	Flash beep
Functional requirement	FR24, F26, F37, and FR38.
Goal	The participant is able to accomplish the flash beep task.
Pre-condition	The participant is logged in and has gone through the practice trial.
Post-condition	The application stores the results, response time and decision in the database.
Main success scenario	1. The application displays a screen informing the participant that they have finished the practice trial and can continue when they are ready. 2. The participant clicks on the “Start” button. 3. The application displays a screen with summary instructions on top and white rectangle background on the bottom. 4. The participant clicks on the “Start” button. 5. The application shows a flash and a beep, in random order. 6. The participant chooses which came first, flash or beep, by either clicking on the respective buttons or using the keyboard arrows. 7. Steps 3 and 4 are repeated for 40 rounds. 8. The application grants the participant a break which can be terminated 9. once they click on “Continue the experiment”. 10. Steps 3 through 6 are repeated until the 120 rounds are completed. 11. The system stores the responses. 12. The application displays a window with the results obtained, the count of correct responses. 13. The participant clicks on “Terminate the experiment”. 14. The application displays a “You have completed this task” message. 15. The participant is redirected to the Home page.
Includes/Extends	None

Table 36: UC12 - Practice loudness perception.

ID	**UC12**
Name	Practice loudness perception
Functional requirement	FR27, F28, and FR37.
Goal	The participant is able to have a practice trial before going into the real loudness perception task.
Pre-condition	The participant is logged in.
Post-condition	None
Main success scenario	1. The participant clicks on the “Loudness Perception” button. 2. The application displays a screen with an overview description of the task. 3. The participant chooses an external audio device, specifically the headphones. 4. The application shows a screen with some instructions and a “configure volume” button. 5. The participant clicks on the “configure volume button” 6. The application plays a 10 s beep, while the user can properly set the volume of his device. 7. A screen with other instructions and a “start test” button is displayed.
Includes/Extends	3a.1 The participant chooses an external audio device, specifically the external audio. 3a.2 A screen with additional instructions and a “next” button is displayed 3a.3 The participant clicks on “next” button.

Table 37: UC13 - Loudness perception.

ID	**UC13**
Name	Loudness perception
Functional requirement	FR27, F29, F37, and FR38.
Goal	The participant is able to accomplish the loudness perception task.
Pre-condition	The participant is logged in and has gone through the practice trial.
Post-condition	The application stores the results, response time and decision in the database.

Main success scenario	1. The participant clicks on the “start test” button. 2. A screen with a summary of instructions and a response scale are displayed. 3. The application plays a beep for 180 seconds. 4. After 10 seconds the response scale flashes and activates 5. The user answers by clicking a button of the response scale. 6. The response scale disables. 7. Steps 11-13 are repeated 9 times. 8. The application displays a “You have completed this task” message. 9. The participant is redirected to the Home page.
Includes/Extends	5a.1 The user does not answer for the 10 subsequent seconds.

Table 38: UC14 - Practice target detection.

ID	**UC14**
Name	Practice target detection
Functional requirement	FR30, F31, and FR37.
Goal	The participant is able to have a practice trial before going into the target detection task.
Pre-condition	The participant is logged in.
Post-condition	None
Main success scenario	1. The participant clicks on the “Target detection” button. 2. The application displays a screen with an overview description of the task. 3. The participant chooses an external audio device, either headphones or external speakers. 4. The participant clicks on the “Start” button. 5. The application displays a screen where instructions are reported, together with a grey box and a “Start” button. 6. The user clicks on the “Start” button. 7. A “Get ready” write appears in the grey box for 5 seconds.

	8. The user is presented the trial: a circle, darker on top or bottom, appears on the left or right side of the grey box, preceded by a fixation cross and by an arrow, which may or may not indicate the screen portion where the circle was going to be displayed. The trial may also play a beep while showing the circle. 9. The user answers by clicking either the right or the left arrow key, according to the instructions. 10. Feedback is displayed on screen, indicating whether the provided answer was correct or wrong. 11. Steps 8-10 are repeated 10 times. 12. The application shows a screen indicating that practice was over, and containing 3 buttons: “Practice slow”, “Practice normal”, “Start”.
Includes/Extends	9.a.1 User does not answer by pressing one of the arrow keys. 9.a.2 A “too slow!” message is displayed on the screen.

Table 39: UC25 - Target detection.

ID	**UC15**
Name	Target detection
Functional requirement	FR30, F32, F37, and FR38.
Goal	The participant is able to accomplish the target detection task.
Pre-condition	The participant is logged in and has gone through the practice trial.
Post-condition	The application stores the results, response time and decision in the database.
Main success scenario	1. The application shows a screen indicating that practice was over, and containing 3 buttons: “Practice slow”, “Practice normal”, “Start”. 2. The user clicks “Start”. 3. A “Get ready” write appears in the grey box for 5 seconds. 4. The user is presented the trial: a circle, darker on top or bottom, appears on the

	left or right side of the grey box, preceded by a fixation cross and by an arrow, which may or may not indicate the screen portion where the circle was going to be displayed. The trial may also play a beep while showing the circle. 5. The user answers by clicking either the right or the left arrow key, according to the instructions. 6. Steps 4-5 are repeated for minimum 128 trials: every wrongly answered trial will be repeated at the end of the block, possibly increasing the total number of trials. 7. Feedback indicating the number of correct answers is displayed at the end of the block. 8. The user clicks on "Continue the experiment". 9. A screen indicating that the first block ended and containing a "Start" button is displayed. 10. The user clicks "Start". 11. Steps 2-8 are repeated for the 2nd block. 12. The application displays a "You have completed this task" message. 13. The participant is redirected to the Home page.
Includes/Extends	None

Table 40: UC16 - Practice movement perception.

ID	**UC16**
Name	Practice movement perception
Functional requirement	FR33, F34, FR35 and FR37.
Goal	The participant is able to have a practice trial before going into the real movement perception task.
Pre-condition	The participant is logged in.
Post-condition	None
Main success scenario	1. The participant clicks on the "Movement perception" button. 2. The application displays a screen with a description of the next task (screen

	resolution measuring) and a “Start” button. 3. Participant clicks “Start”. 4. The application shows a white screen with some instructions and a red rectangle whose size can be modified. 5. The user draws a red rectangle as big as a standard bank card. 6. User confirms by pressing ‘Y’ on the keyboard. 7. The application displays a screen with an overview of the task and a “Start Practice” button. 8. Participant clicks “Start Practice”. 9. The application displays a screen with the instructions of the movement perception task and a “Start Practice” button. 10. Participant clicks “Start Practice”. 11. Instructions are reported on a grey background in full screen mode. 12. User presses ‘ENTER’ 13. A trial is presented: white moving dots on a grey background are displayed in full screen mode. After some seconds the dots change direction. 14. The user answers, indicating the direction where dots were moving, by using the mouse and clicking when satisfied. 15. The user confirms his answer by pressing ‘Y’ on the keyboard. 16. The application displays feedback on screen about the correctness of the answer. 17. Steps 11-13 are repeated again for another trial. 18. The application displays a screen indicating practice has been completed and containing a “Start” and a “Practice” button.
Includes/Extends	6.a.1 User presses ‘N’ on the keyboard. 6.a.2 Go back to step 2. 15.a.1 User presses ‘N’ on the keyboard. 15.a.2 Go back to step 11.

Table 41: UC17 - Movement perception.

ID	**UC17**
Name	Movement perception
Functional requirement	FR33, F36, F37, and FR38.
Goal	The participant is able to accomplish the movement perception task.
Pre-condition	The participant is logged in and has gone through the practice trial.
Post-condition	The application stores the results, response time and decision in the database.
Main success scenario	1. The application displays a screen indicating practice has been completed and containing a “Start” and a “Practice” button. 2. A trial is presented: white moving dots on a grey background are displayed in full screen mode. After some seconds the dots change direction. 3. The user answers, indicating the direction where dots were moving, by using the mouse and clicking when satisfied. 4. Steps 2-3 are repeated for 10 rounds. 5. Feedback indicating the number of correct answers is displayed on screen. 6. The application displays a “You have completed this task” message. 7. The participant is redirected to the Home page.
Includes/Extends	None

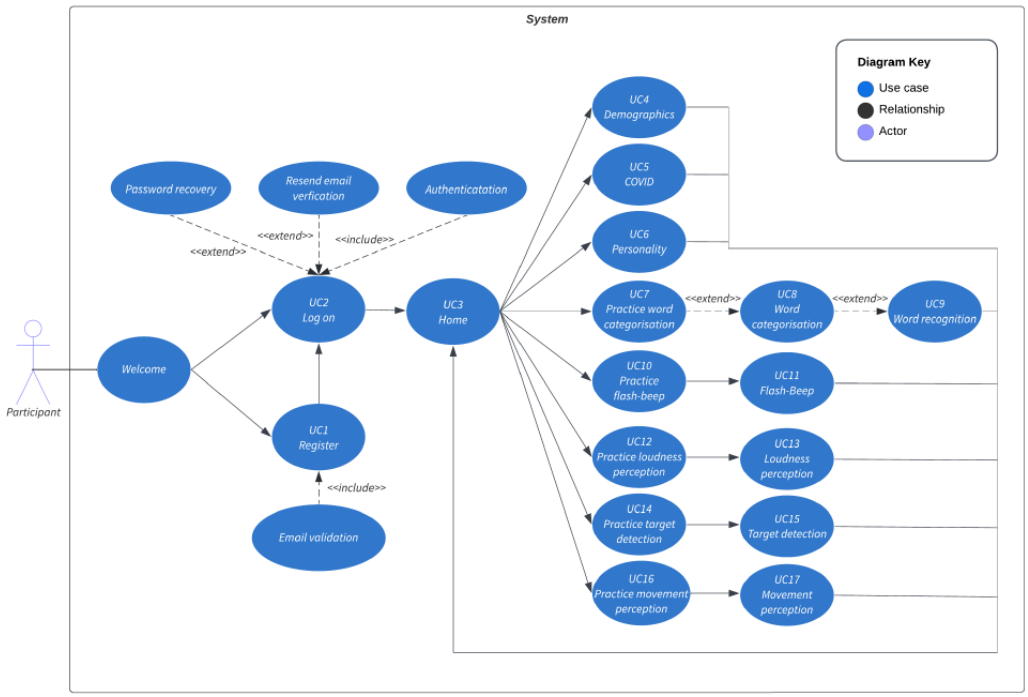

Figure 5: Participant interactions as use cases [10].

3.3 Architecture layers

The design process of a software is essential to generate a precise specification of the information system, with the final goal of getting the solution to be realized. This is achieved through reorganizing the project's characteristics identified in the conceptual modelling phase. Then, it should also be highlighted that this procedure is critical to meeting both the functional and non-functional requirements.

From the software specifications, the majority of which were maintained in their original form, it can be inferred that the program uses a client-server architecture. As a participant, the client submits requests to the centralized web server. The latter supplies to the client the required data, after authenticating the user's credentials.

As a result, the three-tier architecture is the multitier structure that has been chosen; as its name suggests, it consists of three levels of computer logic:

- the *presentation tier*, in charge of the user interface,
- the *application tier*, which handles the real business logic,
- the *data tier*, which stores and maintains the data.

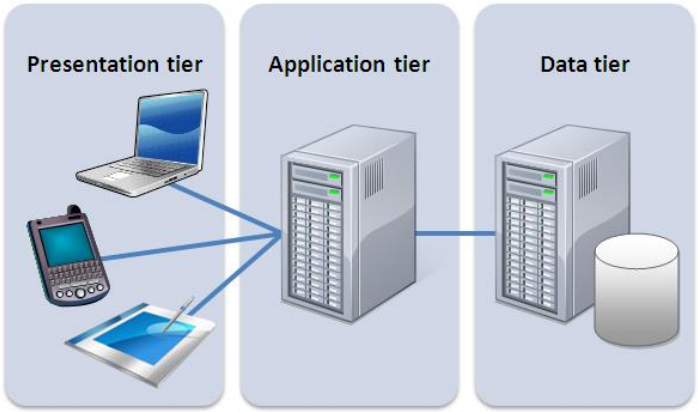

Figure 6: Three-tier architecture [12].

The choice of using a three-tier architecture provides several benefits. Thanks to this approach, in fact, each layer is independent of the other layers. This, first of all, significantly improves two high priority non-functional requirements, that is, availability and reliability, by hosting distinct software layers on distinct servers and caching replies. Secondly, it guarantees a great scalability, since the layers may all be individually scaled to meet the needs of the application and enhances the global system performance, balancing the workload on each tier. Ultimately, it significantly speeds up the development of code: different teams, indeed, can work on frontend and backend separately, without affecting each other.

3.3.1 Presentation layer

The frontend portion of the project represents the presentation layer, sometimes referred to as the user layer. The user interacts with and sees this layer, which conveys all the information and tracks the participant's inputs. It is what enables the users to carry out each experiment since it immediately relates with them. This tier enables also the meeting of some non-functional requirements. For example, it guarantees that the user's graphical interface is clear and simple to use, in order to satisfy, the usability one. There are currently two defined interfaces for the web application which are dependent on the user's actual state within the system:

- Non-logged: The user is not currently signed into the application. The only information on the interface is the study's goal and there are two buttons: one to log in, the other to register.
- Logged-in: The participant has successfully registered and is logged into the application. Along with a menu of eight alternatives for tasks to begin carrying out, the interface also includes further study-related information. Additionally, a home button is available for refreshing and another one for logging out.

3.3.2 Application layer

The core of the system is the application layer, commonly referred to also as the logical tier or business logic. The application tier manages all program capabilities based on the requirements. To accept all requests from the client, this intermediary layer interacts with presentation one. Through APIs (Application Programming Interface), it also interacts with the data layer to return results, extracting or saving data in accordance with the request. Because of this, the code is organized so that it can be found in one of this two major directories (inside backend code directory):

- Models: It contains all the data models for each one of the experiments and questionnaires.
- APIs: It manages and contains all the connections for each one of the experiments and questionnaires.

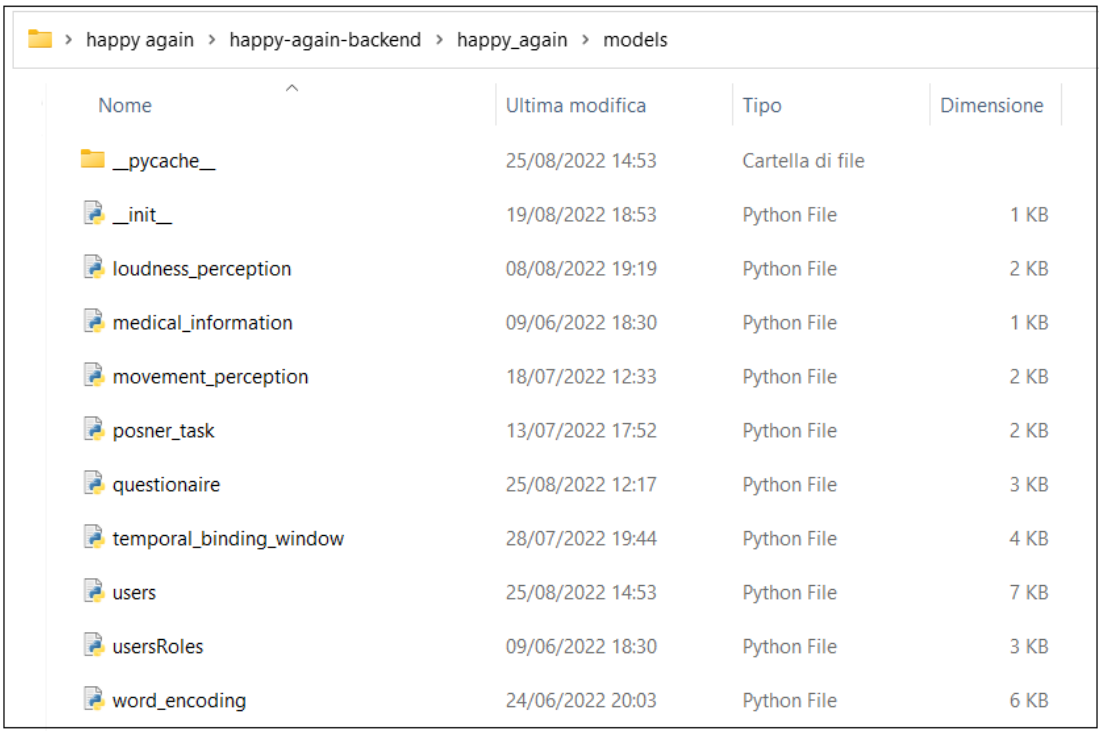

Figure 7: Models directory.

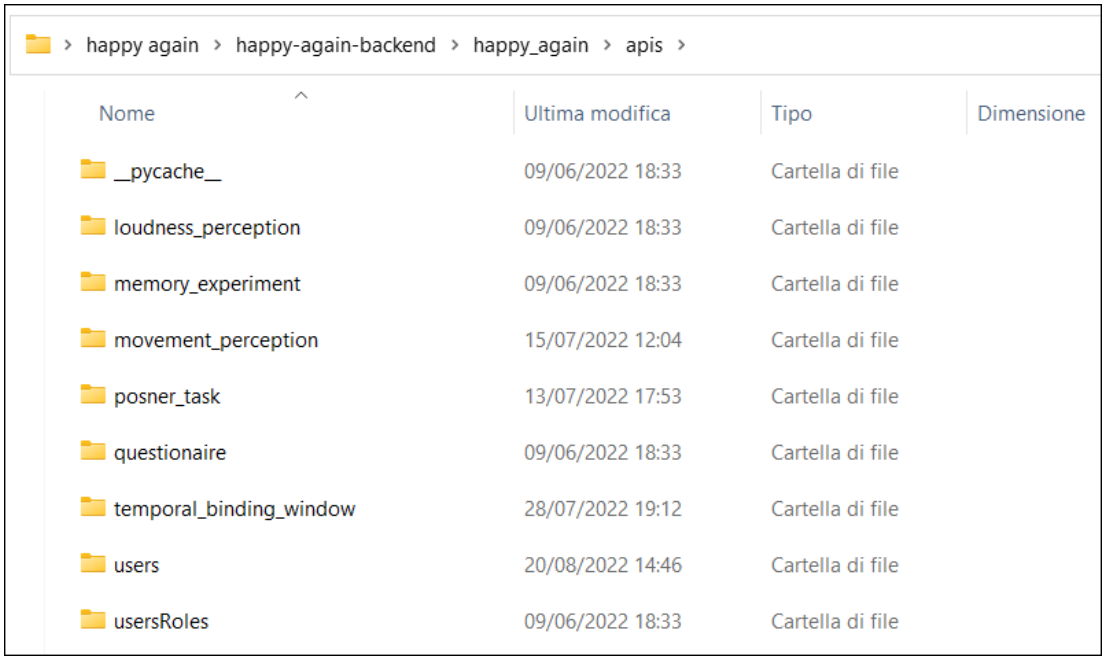

Figure 8: APIs directory.

3.3.3 Data layer

All of the data handled by the program is saved and maintained in the data layer, often known as the database. Since there are predetermined relationships between the columns and rows, the databases that have been developed for this project are relational.

The two basic types of information that must be saved are user and experiment data. Separating the two groups is crucial because researchers will have access to the experimental data that is gathered. This satisfies some of the previously defined non-functional requirements by guaranteeing the security and integrity of the user's personal data. Consequently, two databases have been set up:

- *Userdb*: It keeps track of a user's registration information, including their email address, hashed password, date and time of registration, confirmation time and date, and whether or not they have been excluded from the research (i.e.).
- *Happyagaindb*: It saves surveys and experiments results in accordance with the user ID. It also controls the user's session and metadata data, such as the operating system and browser that were utilized. Finally, this is where the various language options are kept.

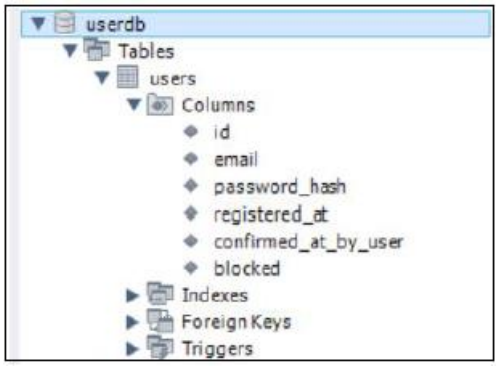

Figure 9: Users' database.

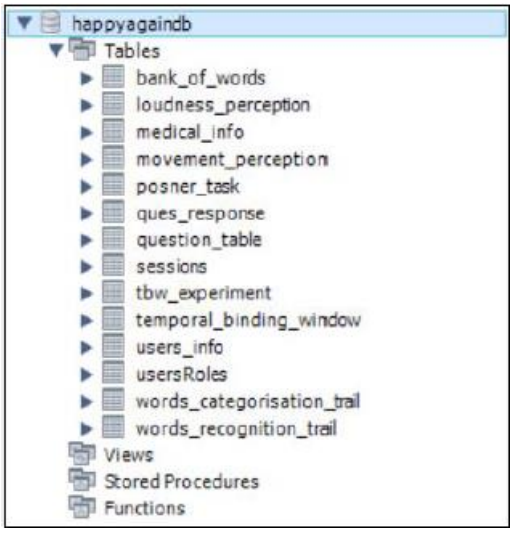

Figure 10: Experiments' data database.

Chapter IV

Results obtained

Building and delivering weekly updated versions resulted from defining use cases and defining the type of architecture. These prototypes were made available for the respective verifiers to evaluate and test. Different programming languages, frameworks, libraries, and other technologies are included into each application's component (i.e., frontend, backend, database); these are further explored in this chapter. The system's results also feature a graphical user interface and a specified deployment mechanism, whose description is provided in this chapter as well.

4.1 Technologies used

With a few exceptions that will be discussed further below, the program's components are organized in accordance with the standard web application architecture.

4.1.1 Backend

The *Python Flask* framework was used to create the project's backend. It is a compact framework that offers a core set of fundamental functions for creating back-end web applications. For any additional services necessary, Flask may import other extension packages. Specifically, the included extensions are:

- *Flask-MySQLdb*: It provides MySQL with the proper connection for Flask.
- *Flask-SQLalchemy*: It permits generating data models; creates an application that is able to connect with the two required MySQL databases.
- *Flask-Mail*: It provides a simple interface to set up the Simple Mail Transfer Protocol (SMTP), which sends the registration email to the participant.
- *Flask-Cors*: CORS stands for Cross-Origin Resource Sharing. It enables setting up the server and acquiring access rights to just certain resources on that server. Other domain requests are denied by it.
- *Flask-JWT-Extended*: It includes features to authenticate users and store and retrieve the security tokens for each user.
- *Passlib* library that supports a large number of hash algorithms. The application uses the *pbkdf2 sha256* library class to hash user passwords.

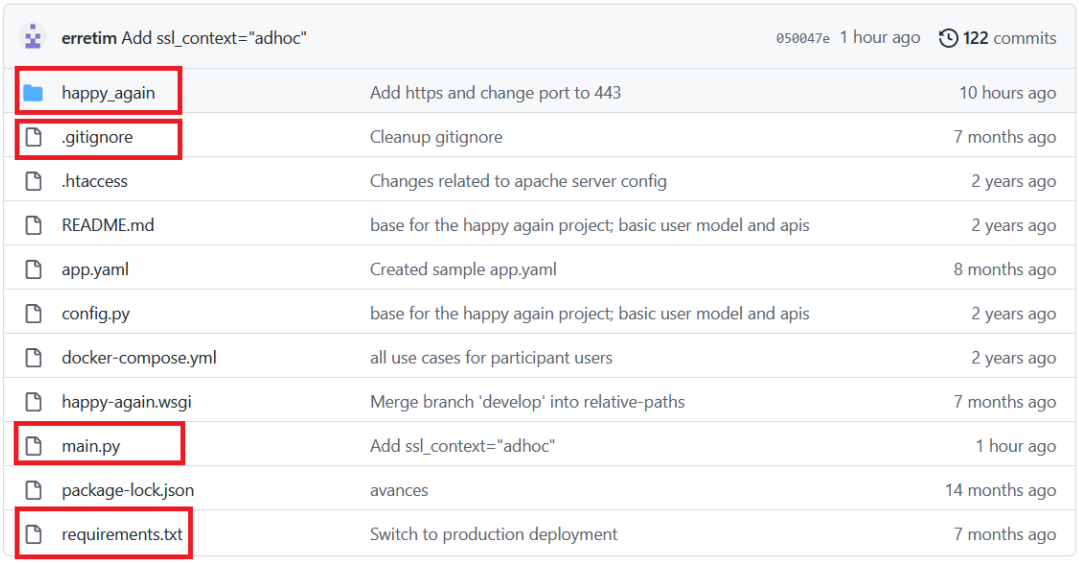

Figure 11: *happy_again_backend* directory.

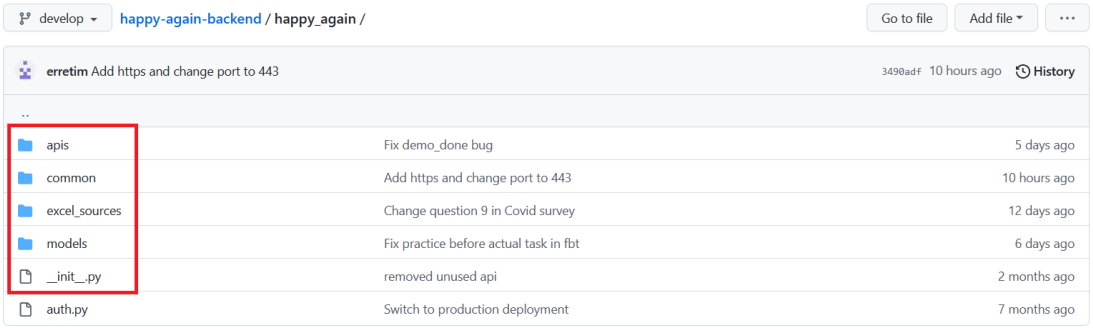

Figure 12: *happy_again_backend/happy_again* directory.

4.1.2 Frontend

The *Angular* framework, developed and maintained by Google, was used in the development of the frontend, the portion of the program that manages all client interactions. It's typically employed to create robust front-end web apps. It functions admirably with RESTful APIs and may be used with any back-end service. Due to its suitability for fast front-end development, Angular is a highly well-known framework. In comparison to writing complicated HTML and JavaScript components from scratch, it produces these components, along with other essential services, significantly more quickly. It is beneficial for creating dynamic content since templates that employ variables, expressions, and other dynamic programming principles are available in place of static HTML components.

Angular is easily portable to many platforms, and it functions well with various web browsers and operating systems. Ultimately, unit testing and end-to-end testing are both well-suited to this technology.

The programming language Angular is built on (and optimized for) is TypeScript; moreover, also the full documentation is written using it. This consideration, along with the fact that TypeScript is a superset of JavaScript, led to the choice of TypeScript as the main frontend programming language.

The major feature of this framework, however, is that it is founded upon the *Model-View-ViewModel* pattern, schematically depicted in Figure 13.

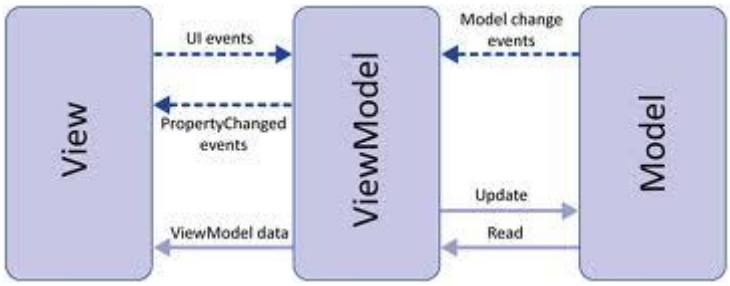

Figure 13: Model-View-ViewModel pattern [13].

In particular, this architecture is composed of the following elements:

- Model: it consists of the classes that are not visually displayed and that represent the content or data layer.
- View: it is the structure, layout, and appearance of what a user sees on the screen.
- ViewModel: the logic, instructions, and attributes necessary for the view to link data and alert any status changes are implemented in this component.

4.1.3 Database

When my team started working on the project, the database in use was *SQLite*, an open-source database engine built in the C programming language. It is an "embedded database" i.e., it is a library that software developers can easily include into other programs. Due to its integration into the most common web browsers, operating systems, mobile phones, and other embedded devices, it is the database engine that is used by most applications. The issue with SQLite is that it creates a "*.db*" file on the local machine on which it is run. This makes it a poor choice for applications that are deployed because it is very difficult to synchronize the local database file with the deployed database file, which may lead to problems in the future. For this reason, we decided to switch to MySQL, which is one of the most widely used relational databases, and is open source and freely downloadable. MySQL cooperates with the operating system to implement a relational database in the storage system of the machine it runs upon. Thus, it was installed on a university server (see deployment paragraph for more details) that acted as a MySQL server, after being properly configured.

4.1.4 Data exchange

REST (Representational State Transfer) APIs is used for communication between the frontend and backend architecture.

First, the user submits a request using a RESTful API, a type of application program interface that enables communication between several clients and the server. As a representation of the resource's current status, this request is delivered to the endpoint. The system uses HTTP as the delivery mechanism for information. *HTTPRequest* is used to send requests, while *HTTPResponse* is used by the server to answer requests. Both of them must be in a machine-readable format that is supported, in this example, JSON (JavaScript Object Notation) objects.

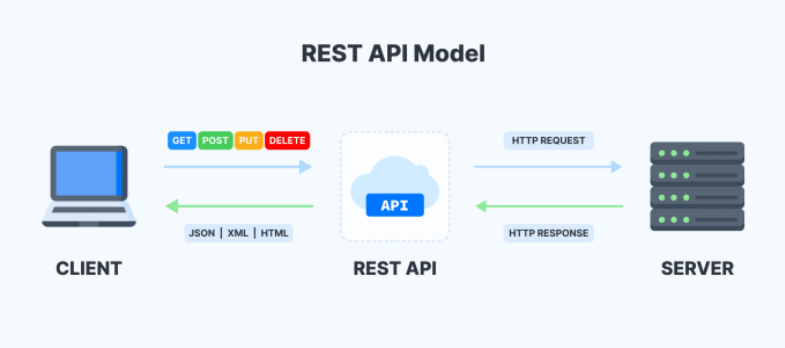

Figure 14: REST API model [10].

4.1.5 Deployment

The application is hosted (both frontend and backend) on a University of Essex' server, which, during development stage, was accessible only via VPN, while, at the time of writing this thesis, it is being opened to Internet access. Going specifically, the used HTTP server is an *Apache2* server, installed on a Linux machine. Though it may seem unpractical (also due to the configuration efforts required), it was good enough to allow the testing by the psychologist team from their own machines, at a reasonable cost. The previous deployment scheme, in fact, involved the use of *Firebase* for frontend and of *Google Cloud Platform* for backend. However, the cost was getting increased day-by-day by Google therefore this solution was not feasible for the long term. Lastly, it should also be noted that the utilized server hosts the database server described above (*MySQL* server).

4.2 Code description

The front-end and back-end code, as well as the file structure within the context of realizing the system architecture and fulfilling the project criteria, are explained in the following paragraphs. Version control and teamwork were conducted using two dedicated GitHub repositories (i.e., one for frontend, another one for backend).

4.2.1 Backend

The whole backend directory is made up of several files and folders, among which:

- *.gitignore*: to prevent additional files used in IDEs from being posted to GitHub, we utilize this file to indicate their extensions.
- *requirements.txt*: Indicates which packages must be installed in order for the Flask app to function. Every time a new dependency is added to the project, this document has to be updated. Looking at the requirements.txt file will allow everyone who downloads and uses the project in the future to find anything they require.
- *main.py*: it is the entry point to the server-side application. In order for the program to start, in fact, this file must be executed, by means of the command "*python main.py*". It contains a *deployment* variable which determines whether the backend is run locally (on port 1234 of *localhost*) or on the deployment server.
- *happy-again-backend/happy again* directory: it includes the *__init__.py* file which initialises the application in such a way that this directory can be treated by main.py file as a package. It includes also the following subdirectories:
    - *happy-again-backend/happy again/models*: in this directory classes that represent data are reported (e.g., class *LoudnessPerception* defined in *loudness_perception.py* file which represents the data collected in this experiment)
    - *happy-again-backend/happy again/apis*: includes all the Application programming interface calls (HTTP get, put, post) which are handling all the back-end operations.
    - *happy-again-backend/happy again/common*: it contains the definition of some common constants and utility functions imported and used in other files. For instance, the *deployment* variable used in *main.py* is defined in the *consts.py* file inside this directory.
    - *happy-again-backend/happy again/excel_sources:* it contains the excel sheets from which database tables are initialised, e.g. the words used in the memory experiment.

4.2.2 Frontend

As already stated before, Angular framework follows the Model-View-ViewModel architecture, hence, newly built Angular components are made up of the following three elements:

- *Template*, which represents the visual element. It describes, indeed, the view (HTML and CSS files).

- *Class*, which comprises methods and properties that regulate the logic of the view. It is a TypeScript file.
- *Metadata*, which are the data that tells Angular whether a certain Class is a component (defined by a decorator).

As a confirmation of the above-described structure, Angular CLI generates at least three files for each new component, which can be edited to produce the intended visual and dynamic behaviour. They are displayed in the following figure.

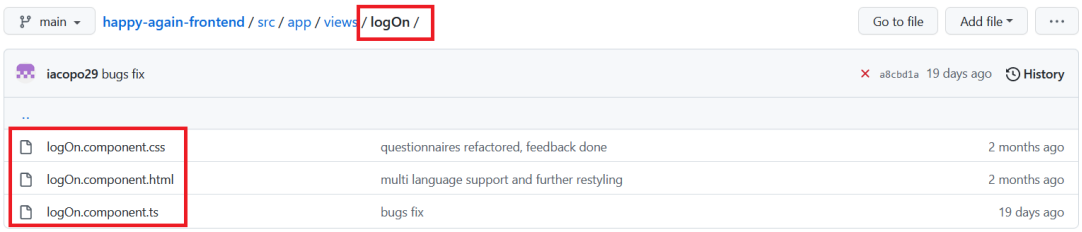

Figure 15: log on component directory.

The folder happy-again-frontend/src/app/views contains all the components' directories required by the application.

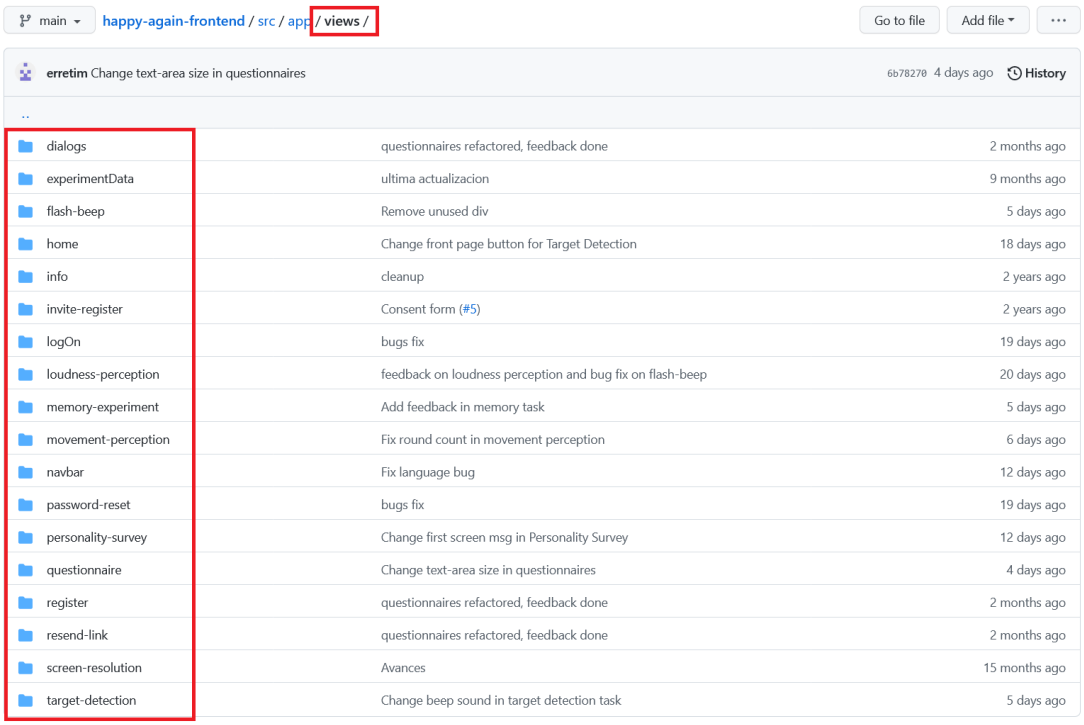

Figure 16: Views directory.

The necessary APIs are instead contained in appropriate TypeScript files inside the *happy-again-frontend/src/app/service* directory.

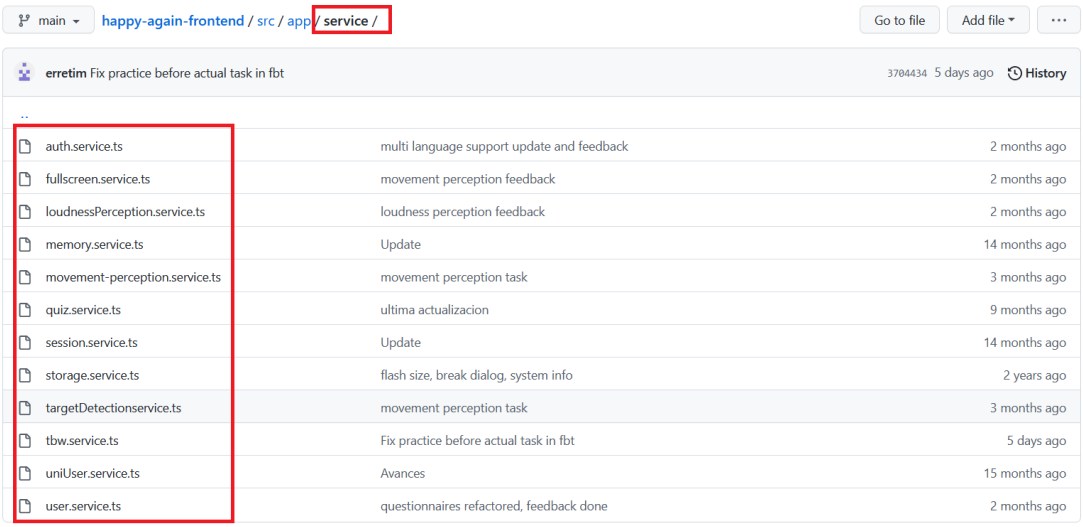

Figure 17: Service directory.

4.3 User Interface

The personal contributions to the project are presented in the paragraphs that follow. These comprise screenshots and an explanation of the user interface of the application, with a comparison of the version that was handed out to my team and the most recent one. Please consider that, despite the interface being the visible result, some major work has been carried out also on the backend and database, with the aim of supporting new experiments (i.e., movement perception task, target detection task) and of meeting the required changes for some other tasks (e.g. memory experiment).

4.3.1 Welcome page

This is the screen shown when first reaching the application URL. Only the study's purpose and an invitation to participate are displayed on this view. There is also a navbar that is part of every application page with some minor variations, i.e. “Log on” and “Register” buttons are replaced by “Log out” and “Home” when user is already logged in. In addition, the navbar contains a language switch that wasn’t actually working, even though some of its logic was partially implemented in back end.

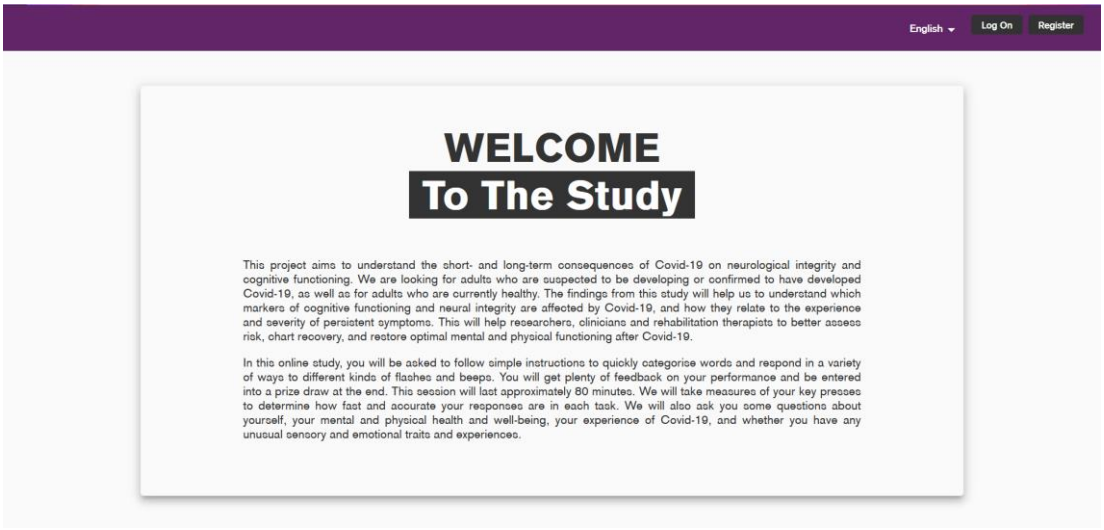

Figure 18: Welcome page - old.

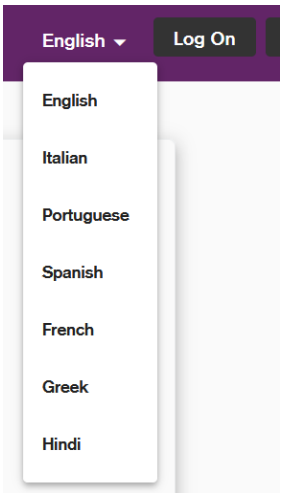

Figure 19: Language switch detail.

The new version presents an updated text, the project logo and a register button which links to the register page. The language switch is fully functional even if currently only the Italian version is available (other translations are not completed yet or need revision). Please note that UI elements and text size has been increased in order to be more readable, and this has also been applied in all other pages.

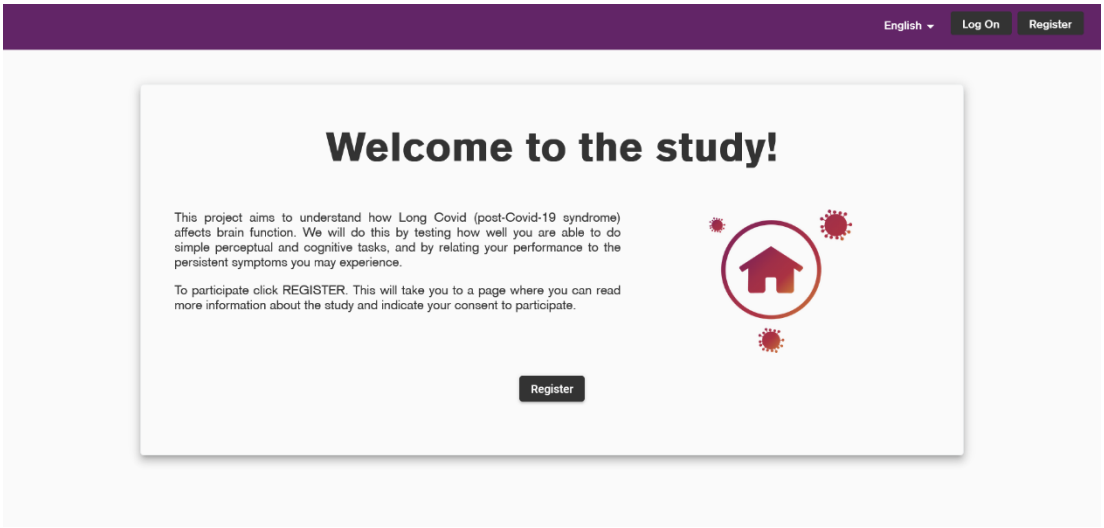

Figure 20: Welcome page - new.

Figure 21: Welcome page - Italian translation.

4.3.2 Register

This page allows a participant to register in order to take part in the experiment. Consequently, there is a form to be filled out (Fig. 22), and then the consent form (Fig. 23). Once the user ticks the checkbox to accept the consent form, he receives an email to confirm the registration(Fig. 24). The form also contains a “resend link” button, to be used in case the email was not received.

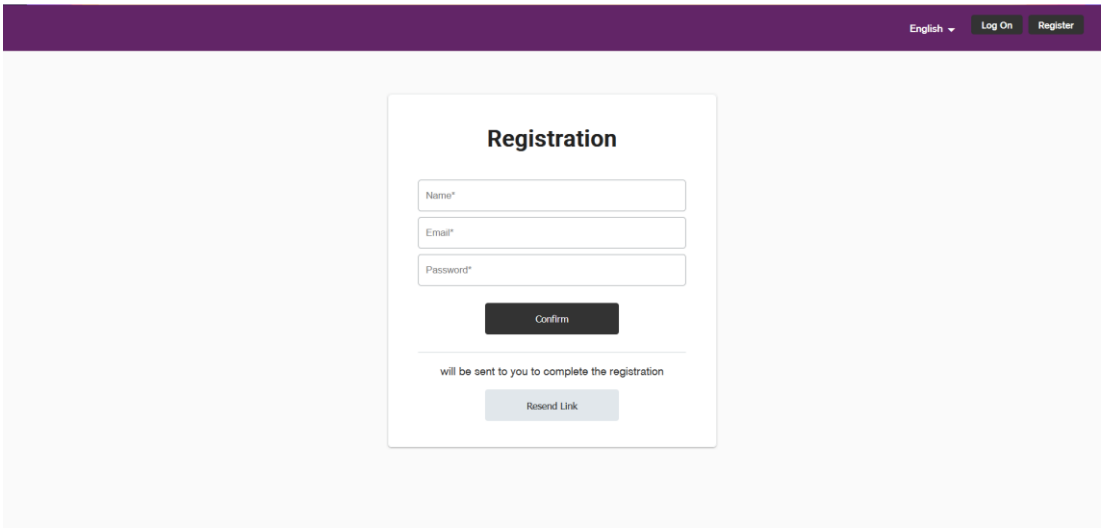

Figure 22: Register - old.

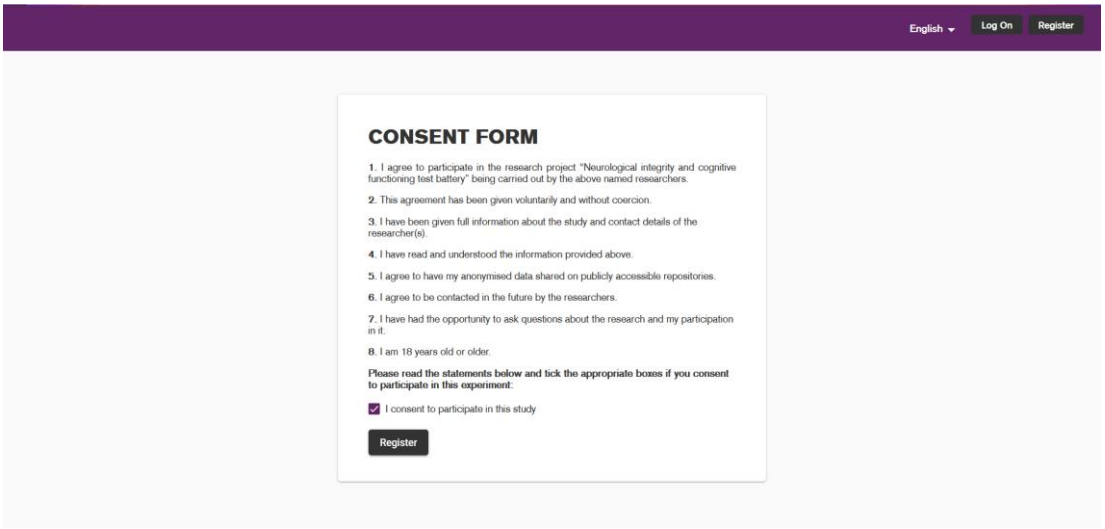

Figure 23: Consent form - old.

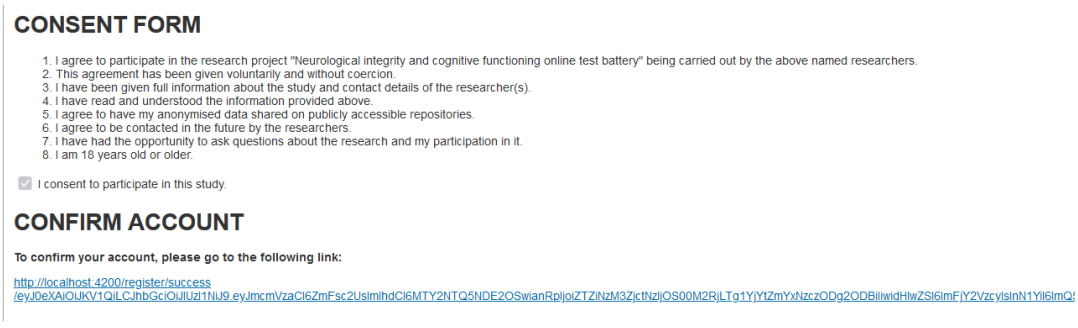

Figure 24: Registration email.

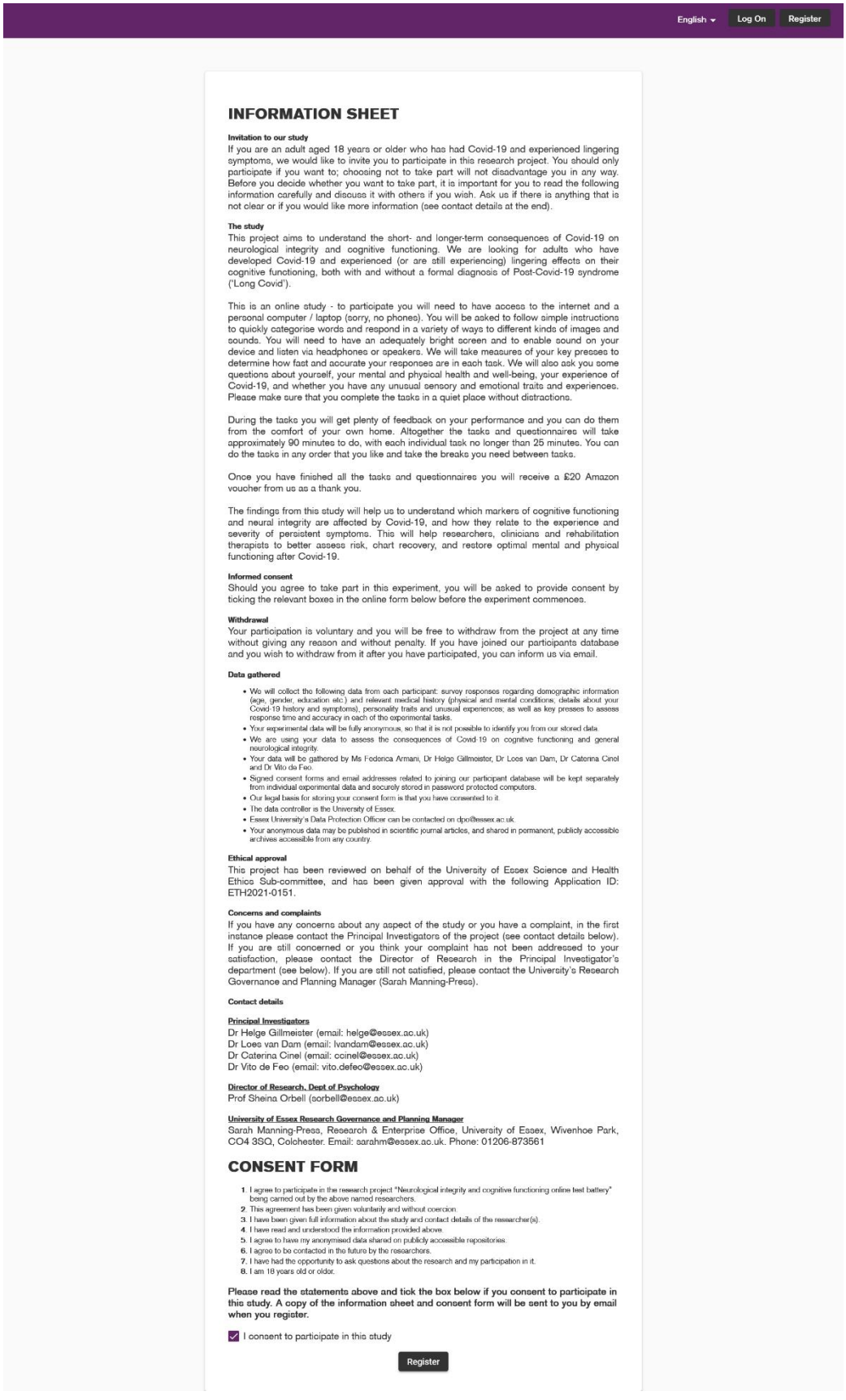

Figure 25: Consent form – new.

The new version of the registration has been refactored in order to be more user-friendly and simpler to use. It begins with the consent form screen (Fig. 25), that now includes the information sheet. Once the user gives his consent by ticking the checkbox the registration form is displayed. This has been improved by adding a clearer and more intuitive error signalling (Fig. 26-27), since in the previous version errors were reported in a *snackbar* just for a few seconds, not highlighting the wrong fields. The “resend link” button has been removed and its functionality is embedded in the newly added subsequent screen (Fig. 28). If the user clicks on it, another new screen is displayed where the user can fill a text field to insert his email (Fig. 29). The received email now supports multi language, thanks to additional work performed on the backend of the application.

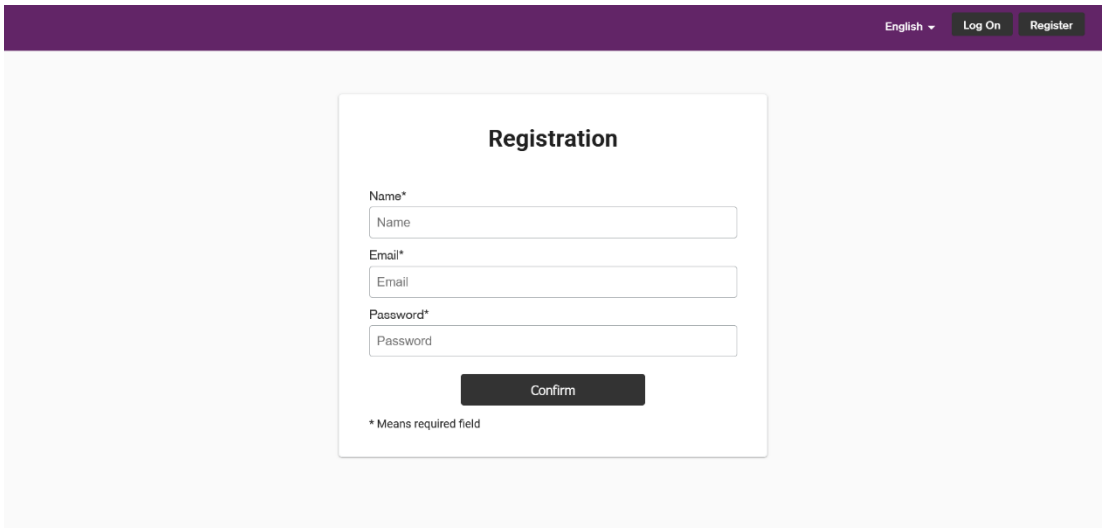

Figure 26: Register form 1 - new.

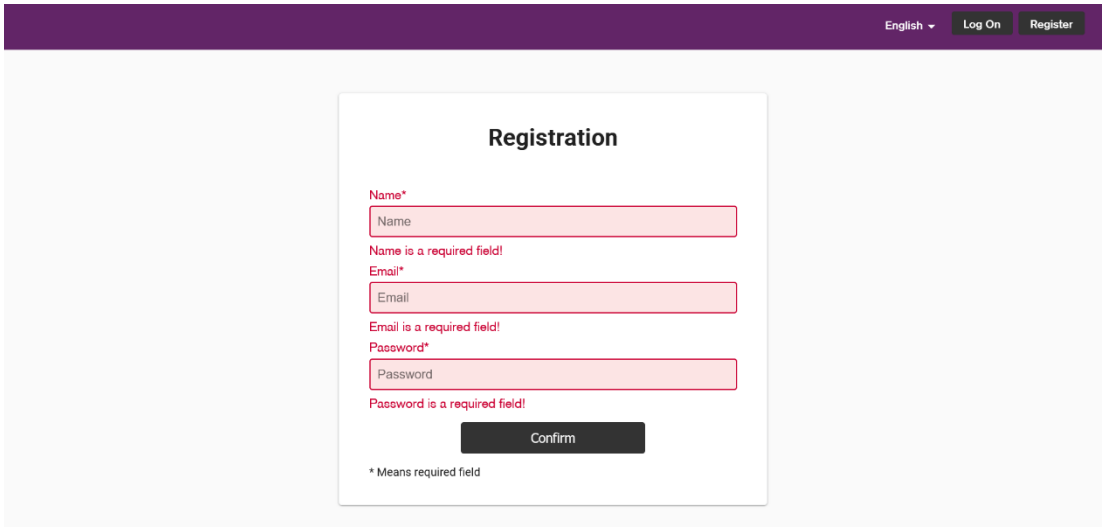

Figure 27: Register form 2 - new.

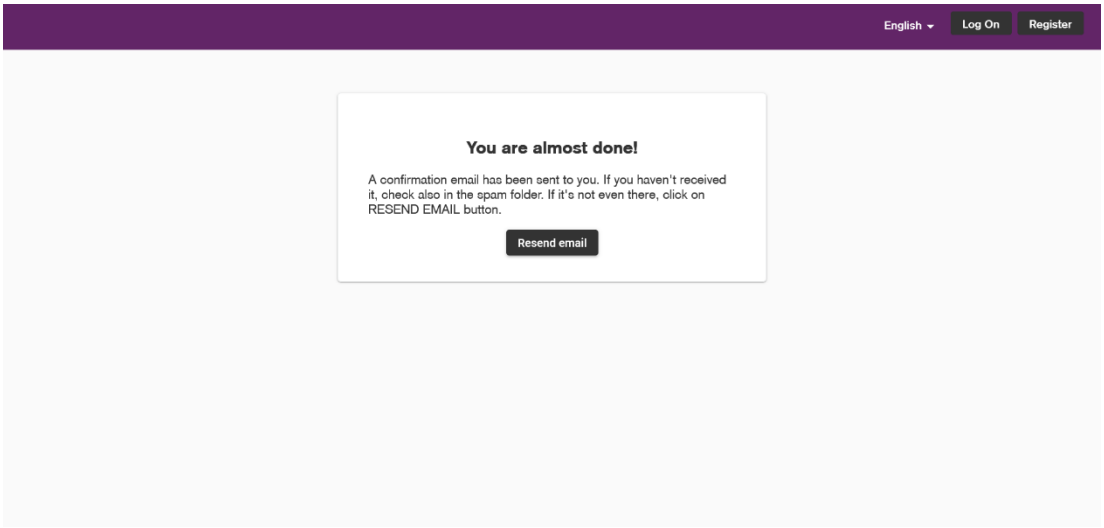

Figure 28: Register confirmation - new.

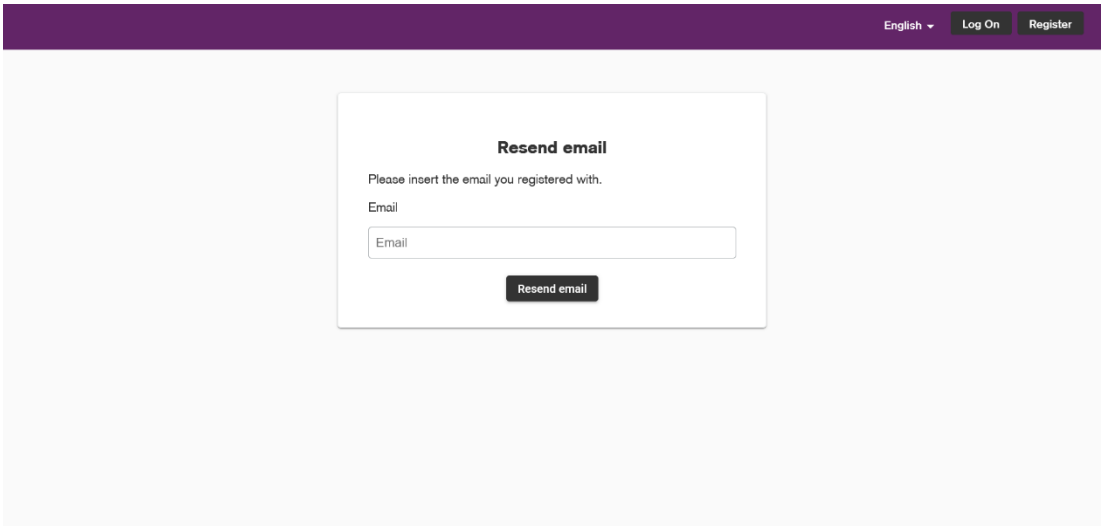

Figure 29: Resend email - new.

4.3.3 Log on

After the participant has successfully registered he is presented with a login screen where his email address and password must be entered to access the application (Fig.30). The platform offers a “forgot password” option in case the user cannot remember the specified password. The user can then enter the email they used to register on a new screen that appears after the original one has been expanded (Fig. 31).

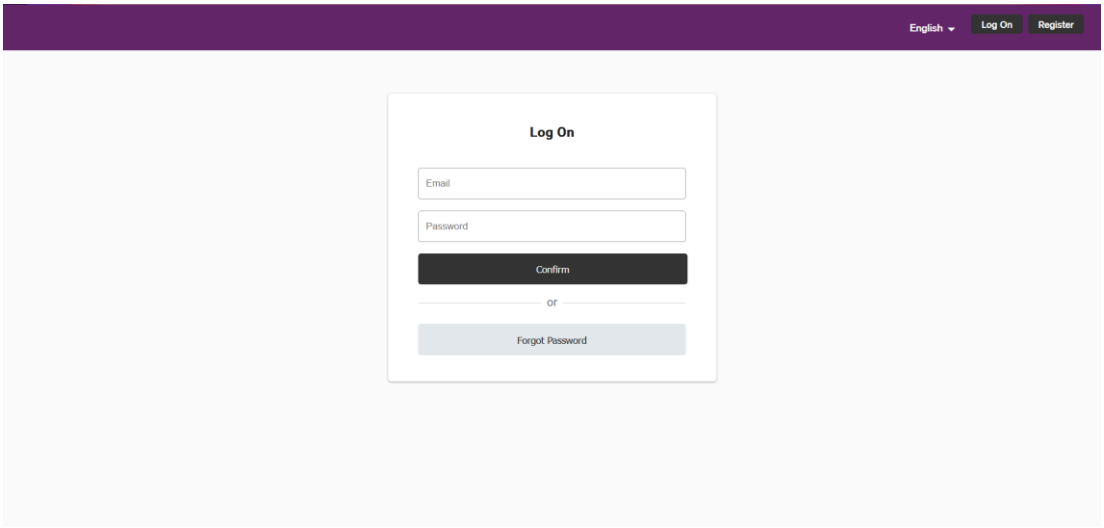

Figure 30: Log on - old.

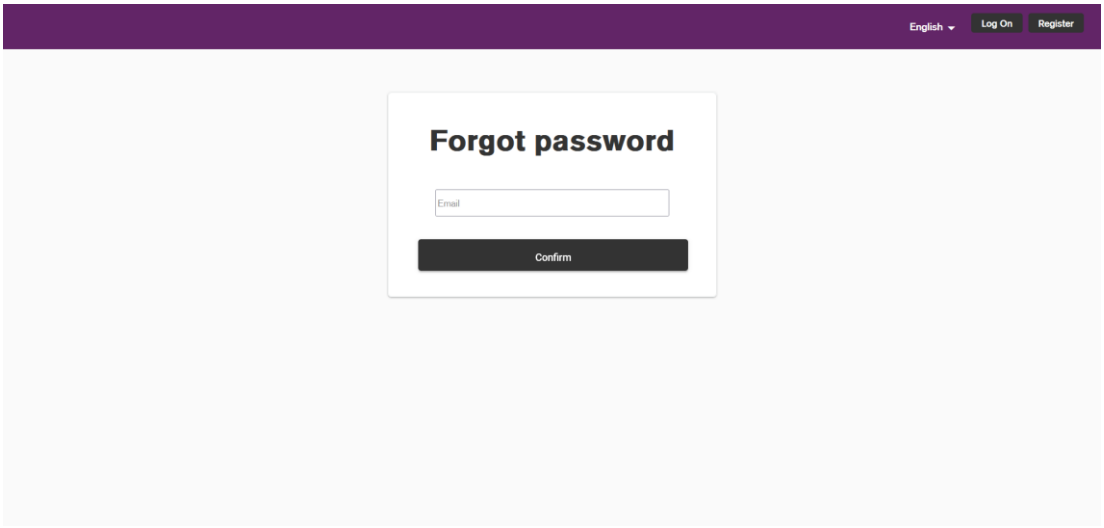

Figure 31: Forgot password - old.

The new version of the interface (Fig. 32) has an additional “email not confirmed” button that links to the same screen described in figure 29. The forgot password screen has been updated to keep consistency with the resend confirmation email UI (Fig. 33). Once the confirm button has been pressed, an email with the link to reset the password is sent to the inserted address. Some work has been carried out also on the backend part, to fix some issues related to the email contained links, which did not function properly on both the local and deployed versions.

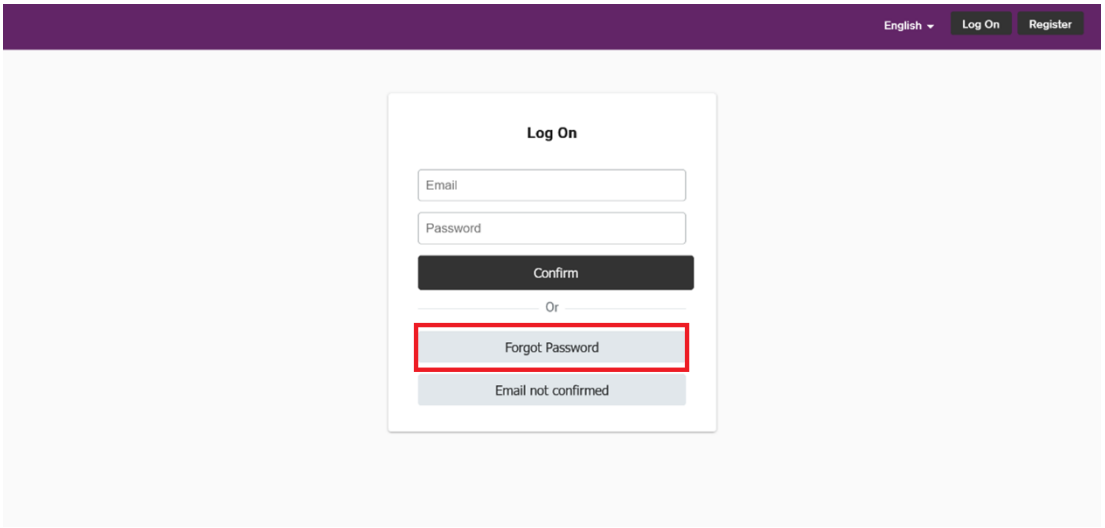

Figure 32: Log on - new.

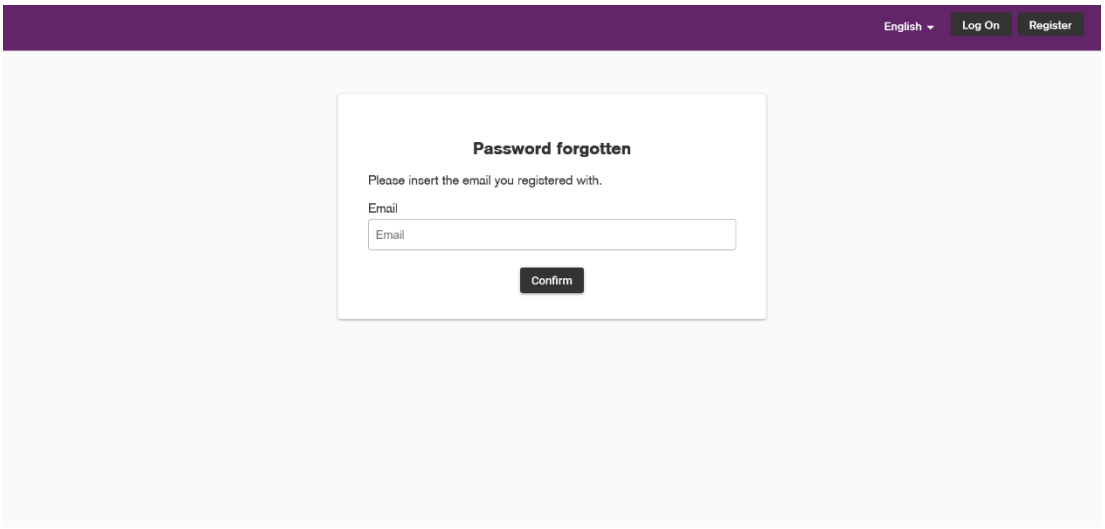

Figure 33: Forgot password - new.

4.3.4 Home logged

This screen is displayed as soon as the log in successfully completed (Fig.34). It contains a text with a study overview and some instructions, and the links to the different tasks to complete. In addition there is a progress circle that indicates the percentage of completed tests and the next one to execute. The new version (Fig. 35) does not contain the progress circle, replaced by the project logo and the links to the different experiments are now buttons, which also indicate its expected duration.

Figure 34: Home - old.

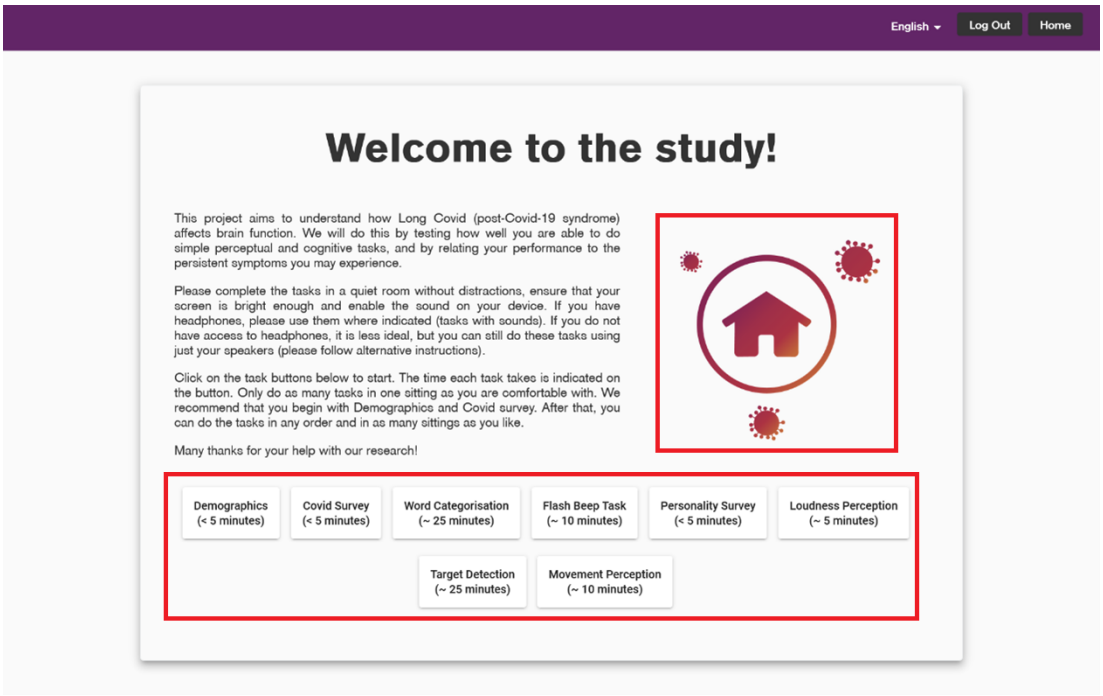

Figure 35: Home - new.

4.3.5 Demographics

Depending on the provided answers, the demographics survey asks the participant 8 to 9 questions (Fig. 36). The goal of this evaluation is to provide context for the participant's qualitative and quantitative attributes, including age, work position, and medical issues, among others.

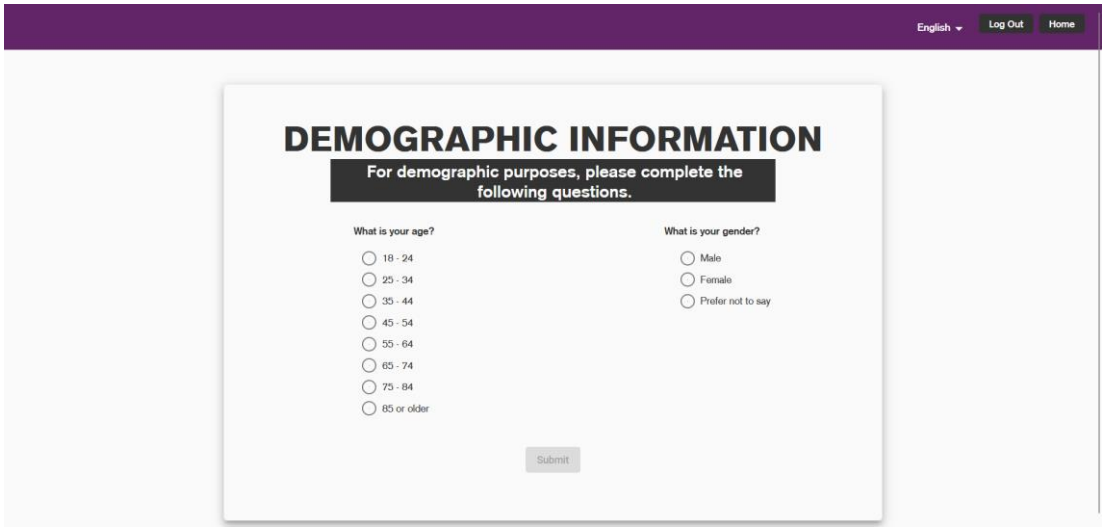

Figure 36: Demographics - old.

This interface was modified, so the new questions appear only after responding to the previous one (Fig. 37). Finally, a screen notifying the user that the survey has been finished displays (Fig. 38), and after a little delay, the user is sent back to the home page. This feature has been added in all other tasks as well.

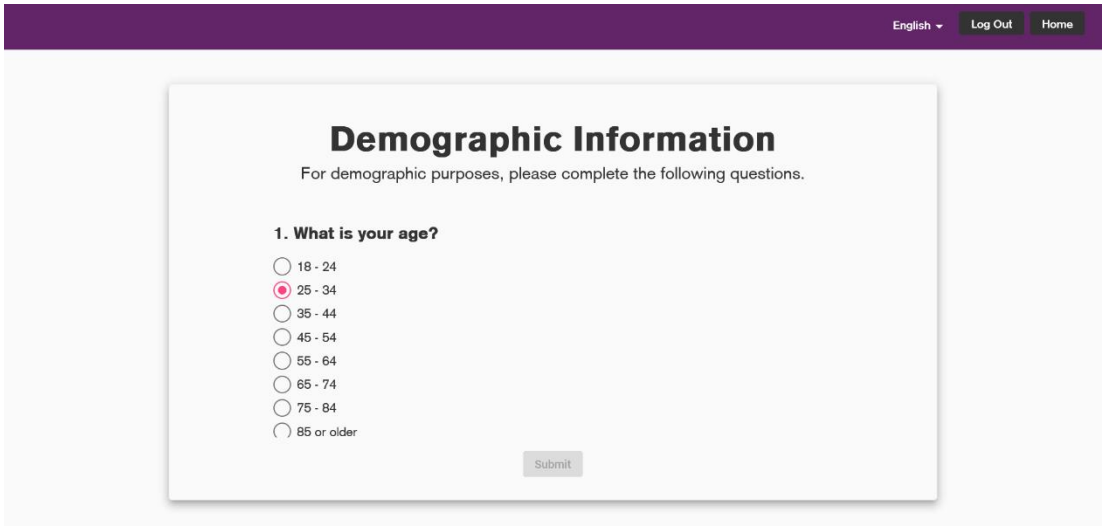

Figure 37: Demographics - new.

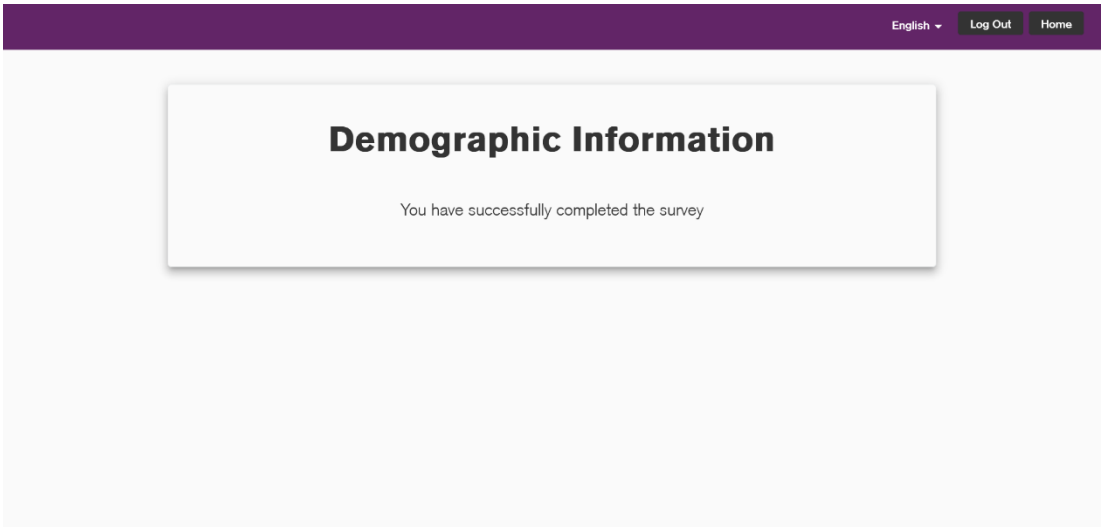

Figure 38: Demographics success message - new.

4.3.6 COVID Survey

The COVID survey is a crucial step in learning more about the patient's medical history. If the participant has had the virus, it is posed seven to twelve questions that are shown in accordance with the person's prior responses (Fig 39). In the new version (Fig. 39) there is a new feature that implied frontend and backend work: the system will remove the user's account from the application if he answered in the first question that he had not had Covid-19 (Fig. 42). The program first displays a warning message asking whether he is certain of his response before proceeding (Fig 41). Since every participant is required to have had Covid-19, the survey is crucial to preserving the study's objectives.

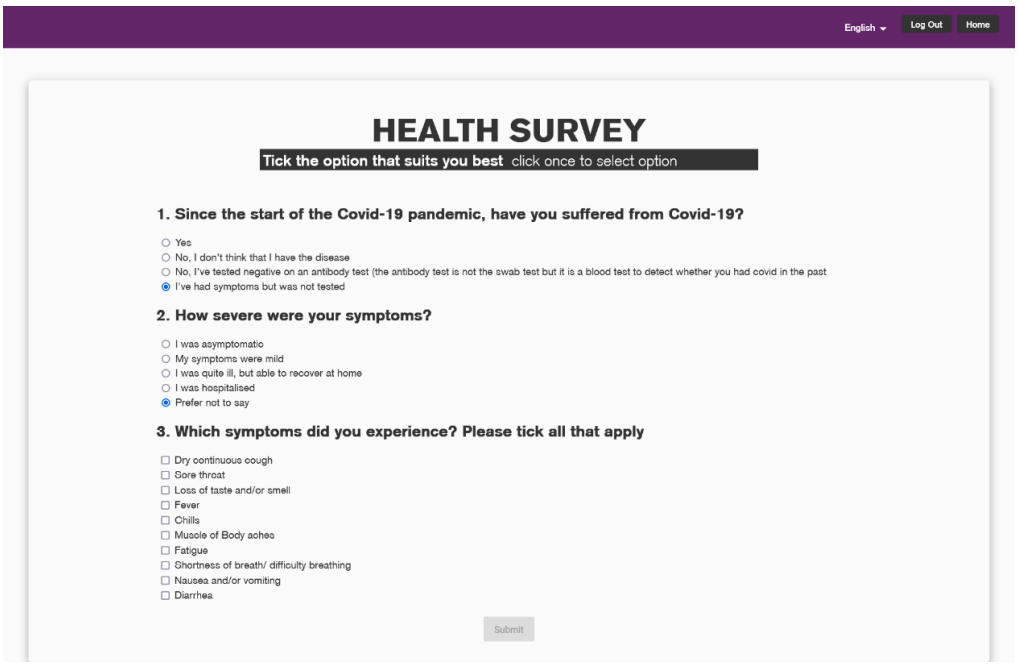

Figure 39: COVID survey - old.

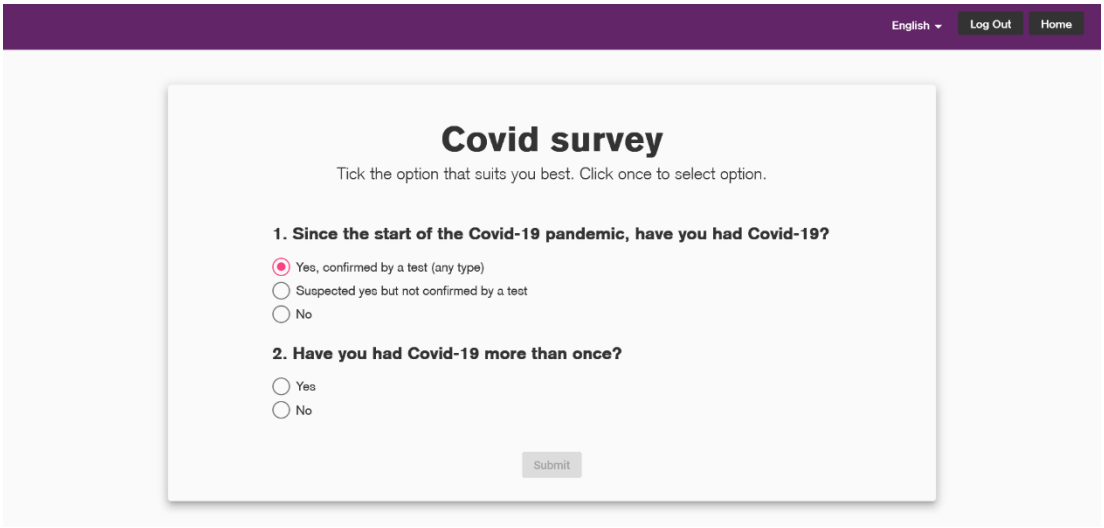

Figure 40: COVID survey - new.

Figure 41: COVID survey warning 1.

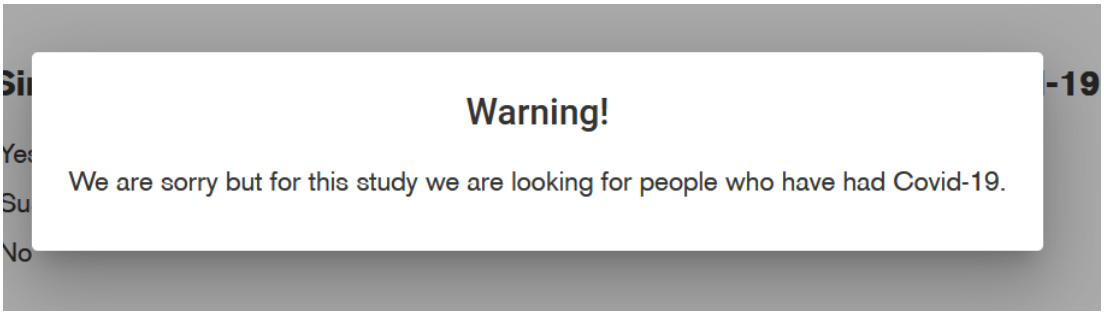

Figure 42: COVID survey warning 2.

4.3.7 Personality survey

The purpose of the personality survey is to comprehend and organize the participant's sensory experiences. It was made of 74 statements on which the participant could agree on a 4-point scale (Fig. 43). In the new version (Fig. 44) the questions have been reduced to 33 as well as the response options, which are now only 2, represented by buttons and not checkboxes. These changes required backend and frontend work.

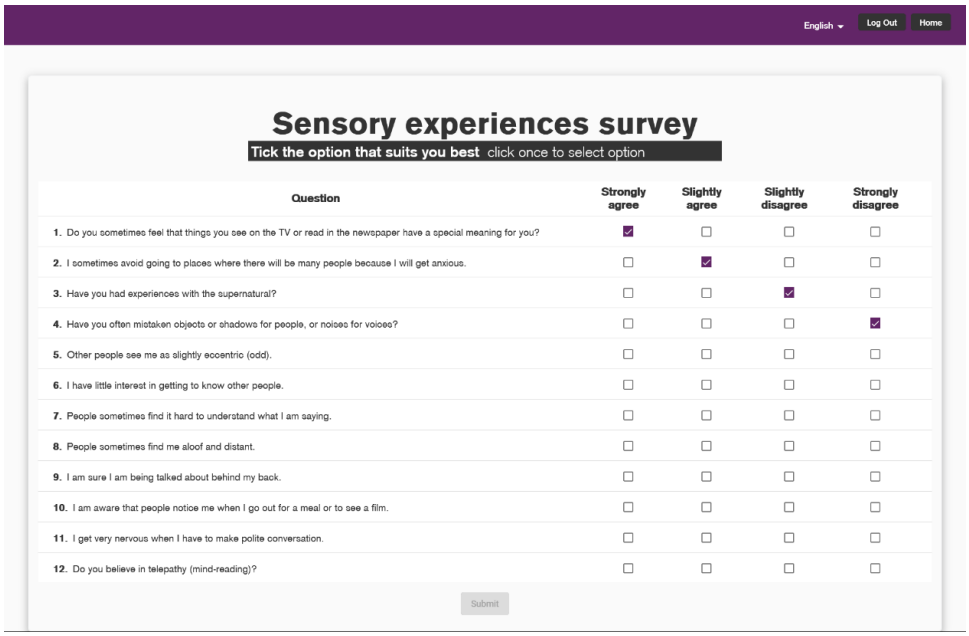

Figure 43: Personality survey - old.

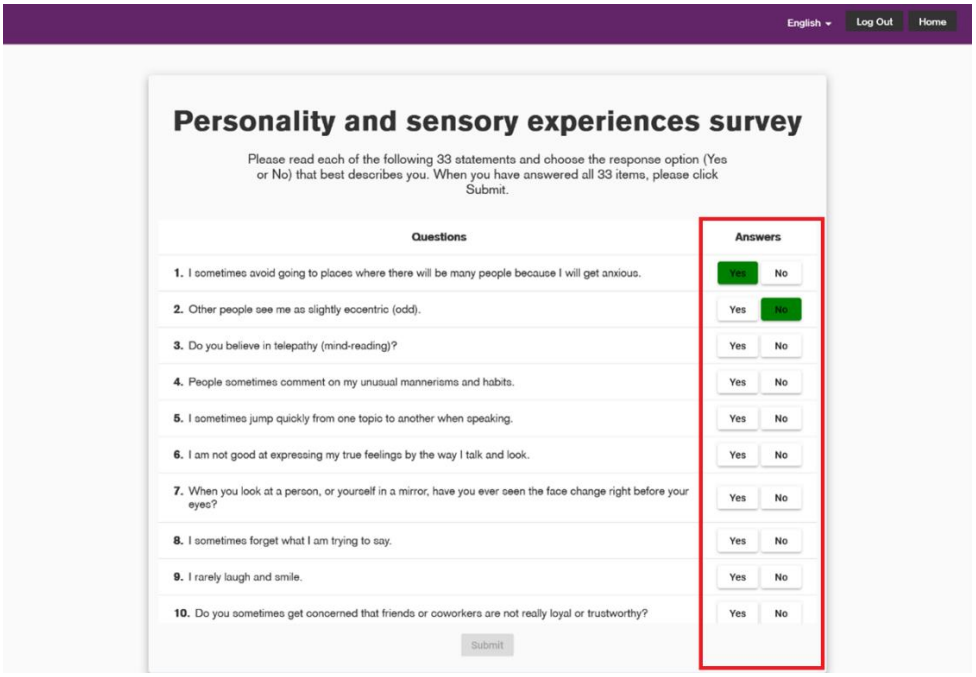

Figure 44: Personality survey - new.

4.3.8 Memory experiment

As already discussed in chapter 1, this test is divided into two distinct ones:

- Word categorisation,
- Word recognition.

The initial version (Fig. 45 to Fig. 50) has been consistently updated, by working both on front end and back end part. First of all the database of words has been modified, changing the words and their number. Now the word categorisation tasks are executed in two blocks of 65 trials (Fig. 54), with breaks every 22, containing feedback. They are now preceded by an overview of the experiment and a get ready screen (Fig. 51 to Fig. 53).

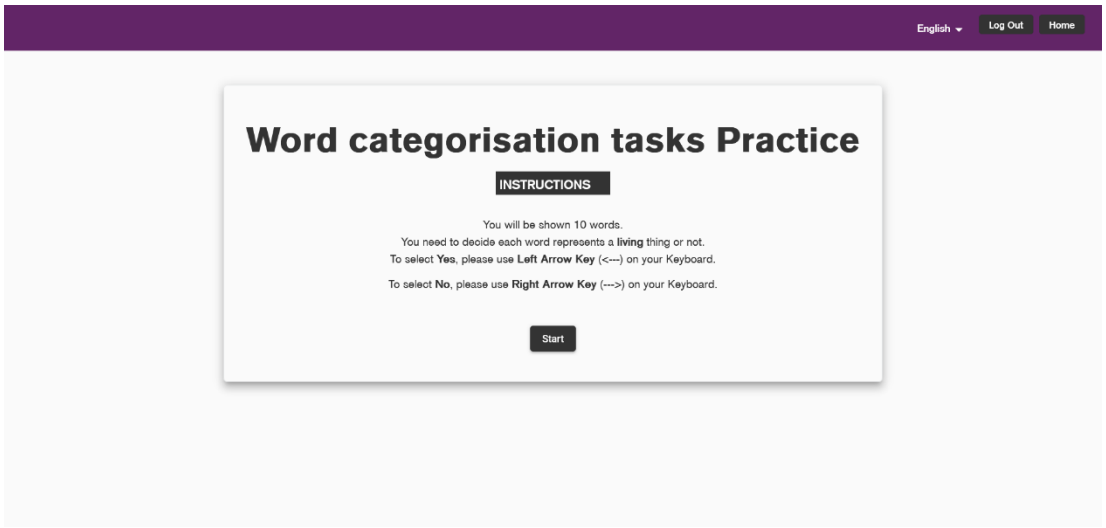

Figure 45: Word categorisation instructions – old.

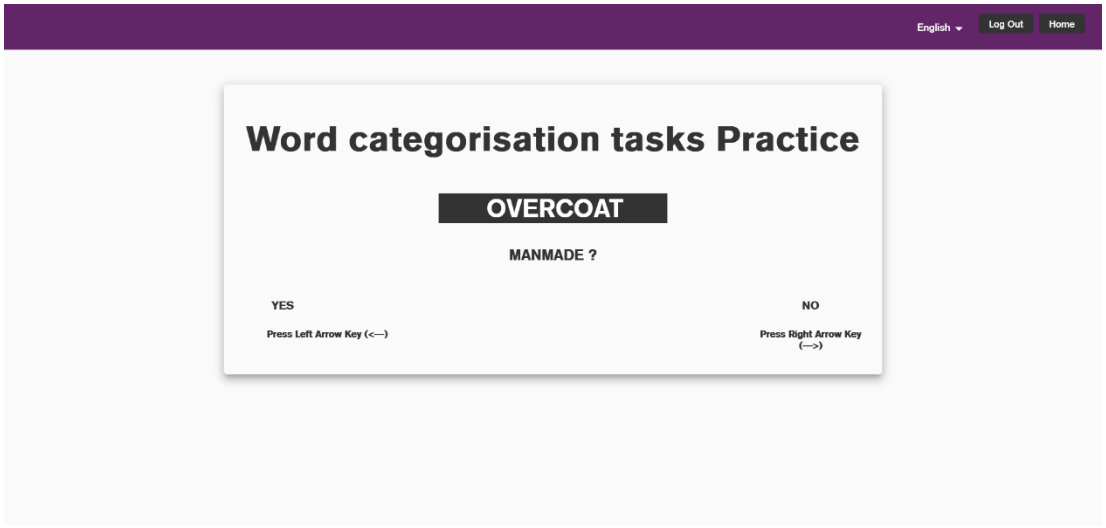

Figure 46: Word categorisation trial - old.

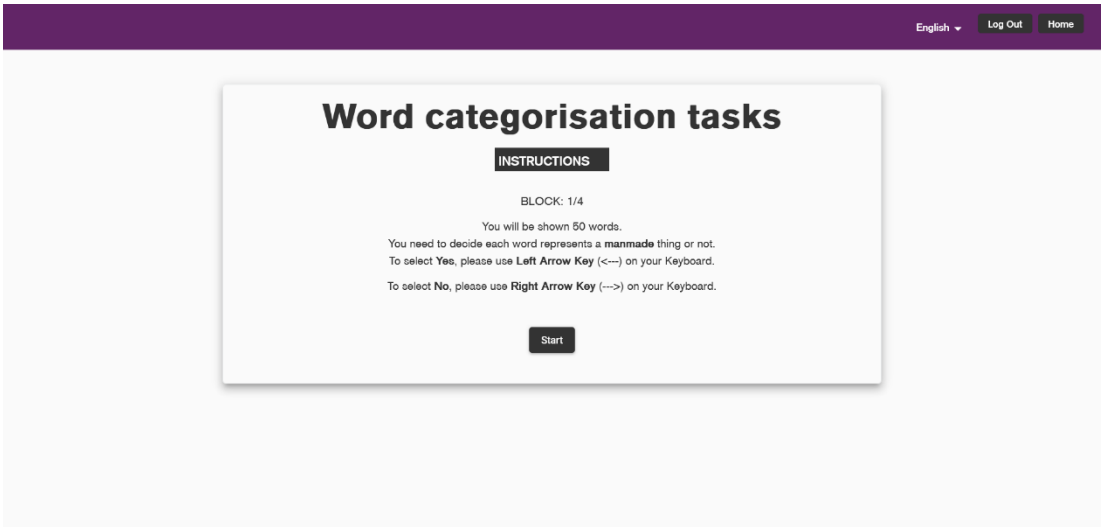

Figure 47: Word categorisation instructions 2 - old.

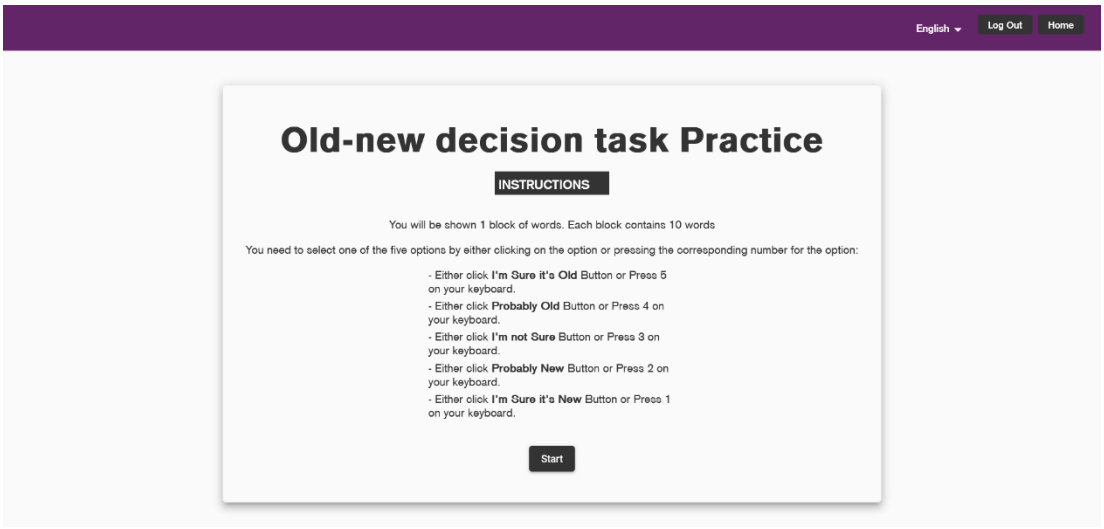

Figure 48: Word recognition instructions - old.

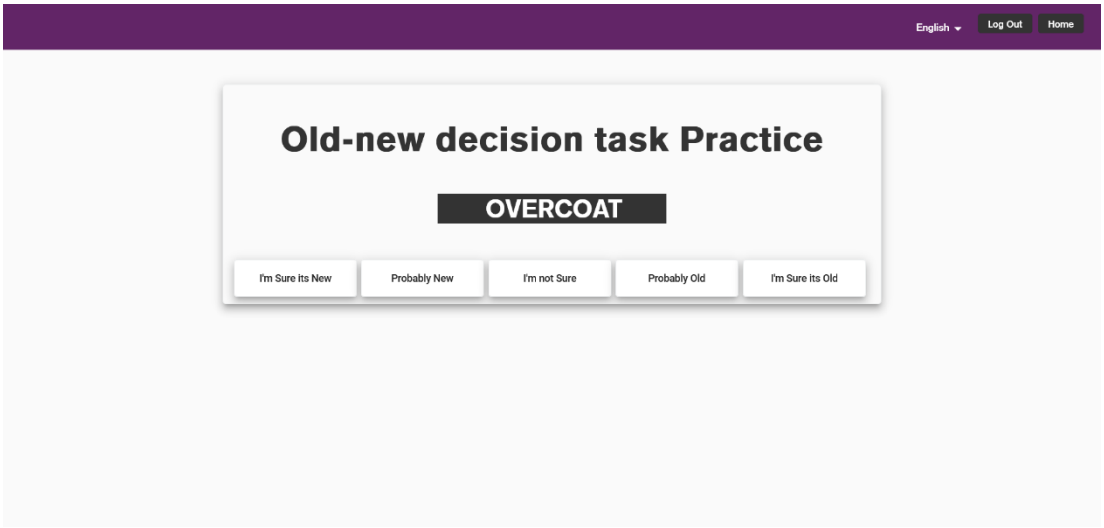

Figure 49: Word recognition trial - old.

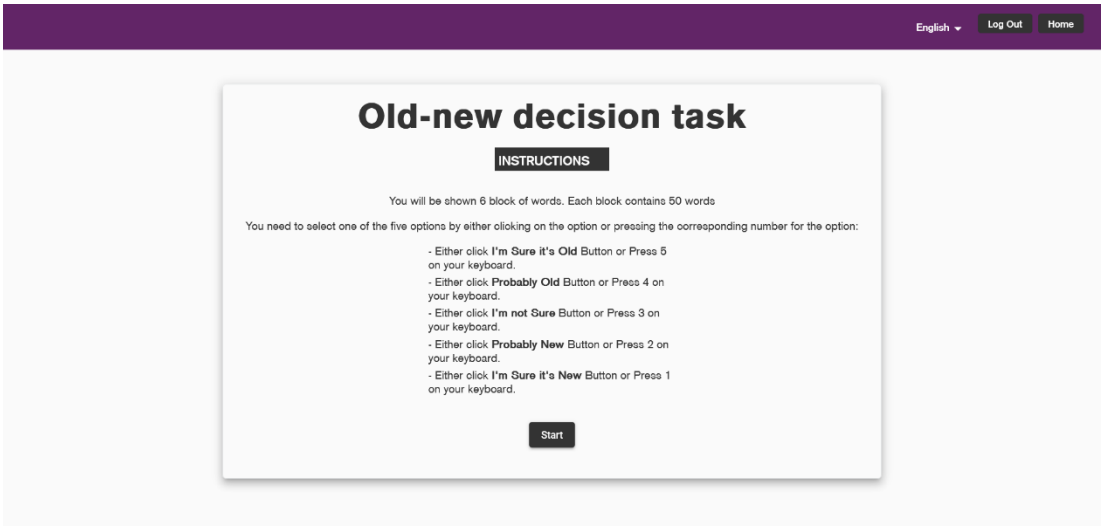

Figure 50: Word recognition instructions 2 - old.

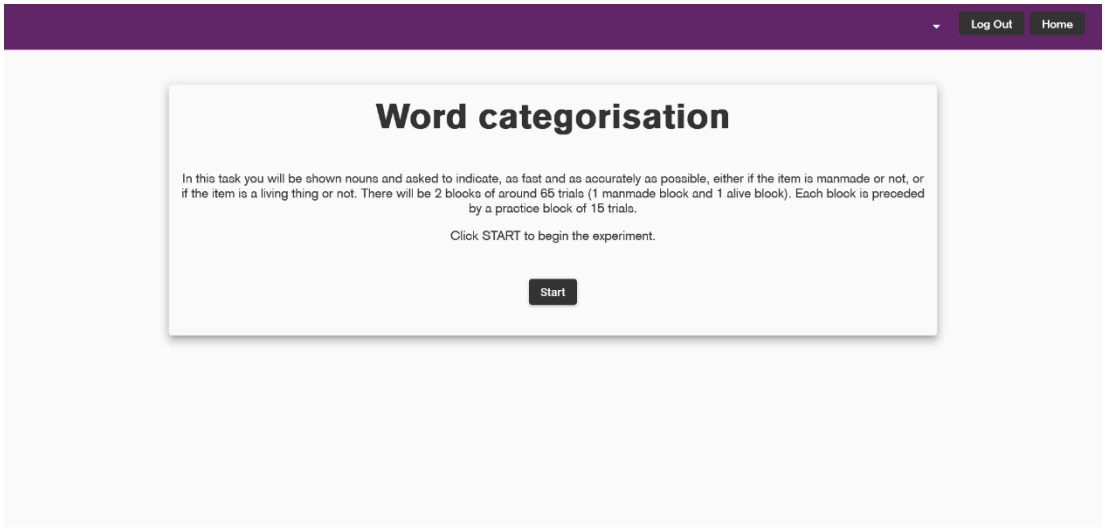

Figure 51: Word categorisation overview - new.

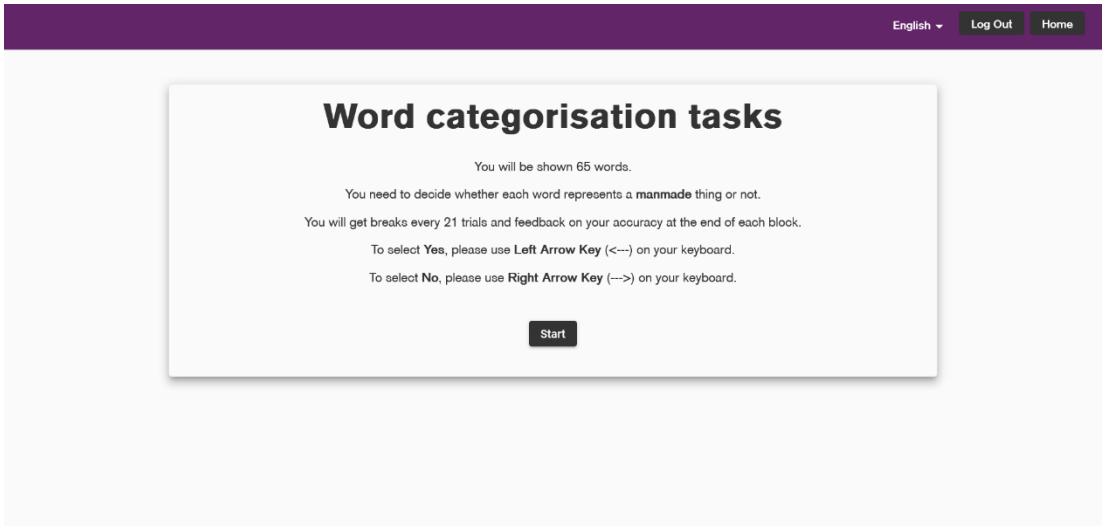

Figure 52: Word categorisation instructions 1 - new.

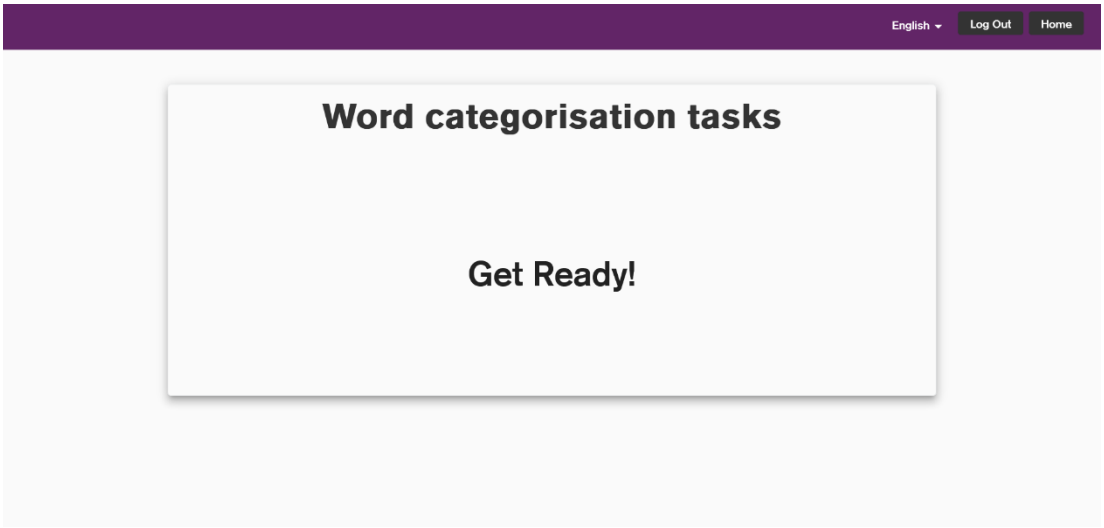

Figure 53: Word categorisation “get ready” - new.

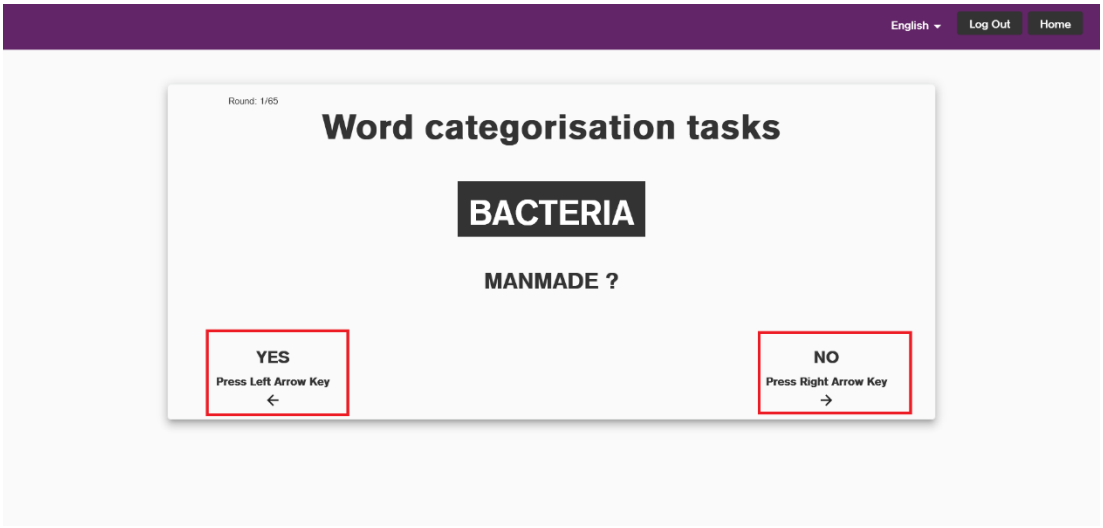

Figure 54: Word categorisation trial - new.

As regards the recognition task (Fig. 55), this has been extended to include a “source recognition” task: for words indicated by participants as “old” or “probably old”, they are furtherly asked to state whether they are from alive or manmade list (Fig. 57). Practice for this task has been removed, while feedback about performance has been added in the between blocks screens (Fig. 58).

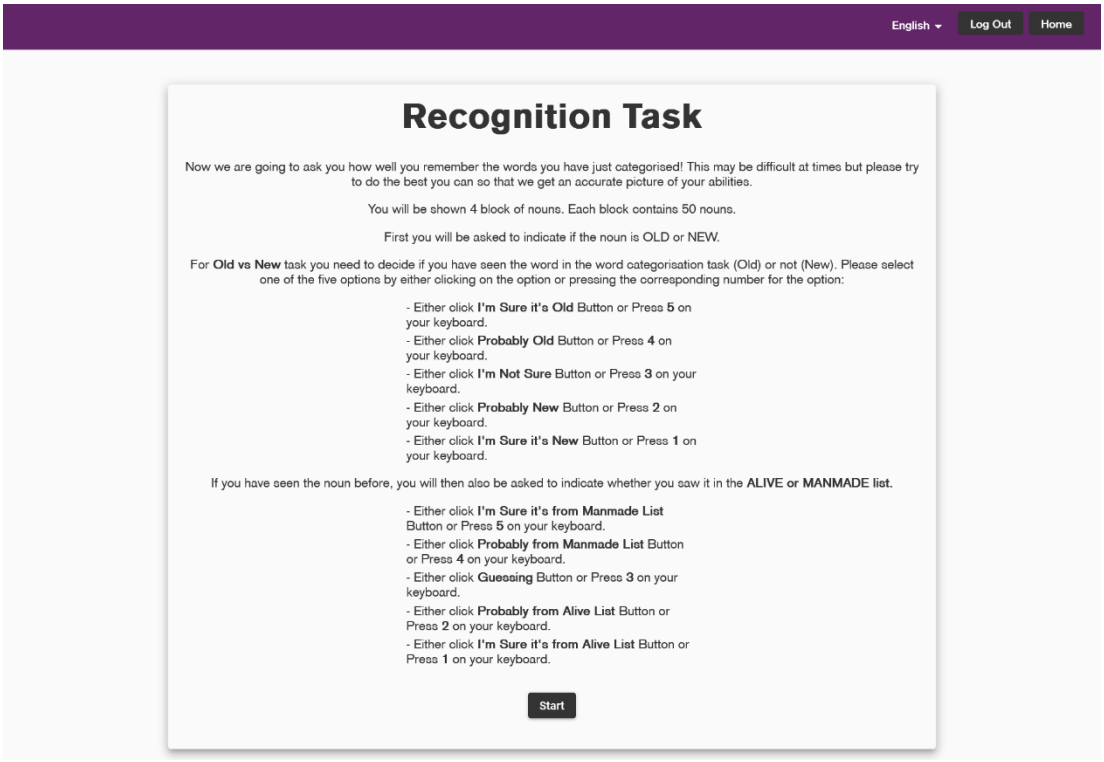

Figure 55: Word recognition instructions - new.

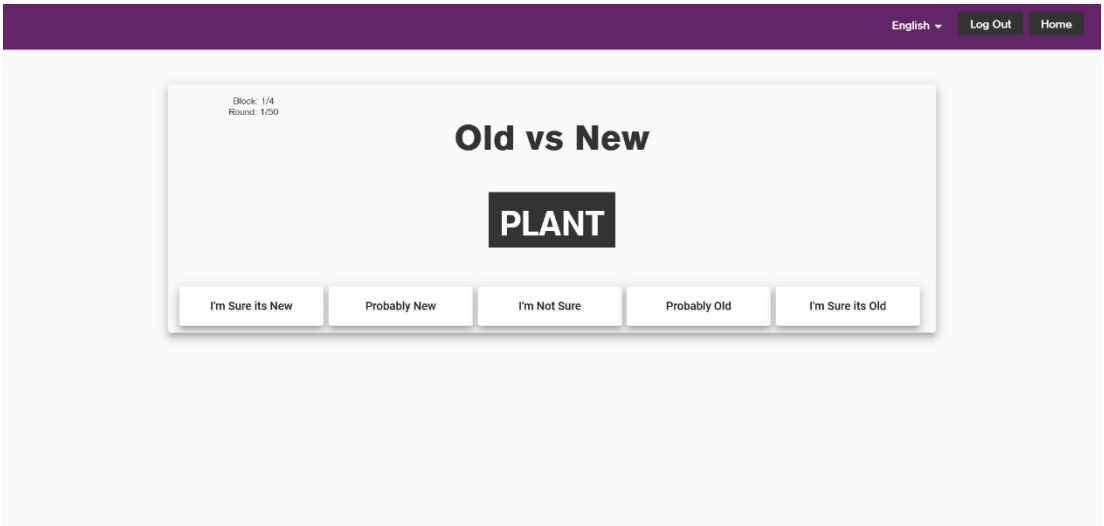

Figure 56: Word trial "old vs new" - new.

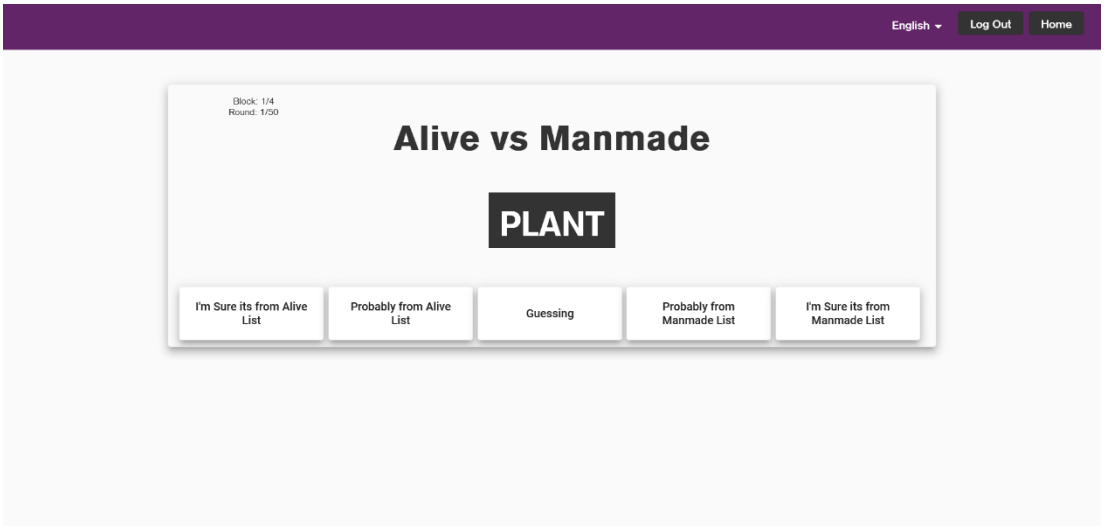

Figure 57: word recognition trial “alive vs manmade” - new.

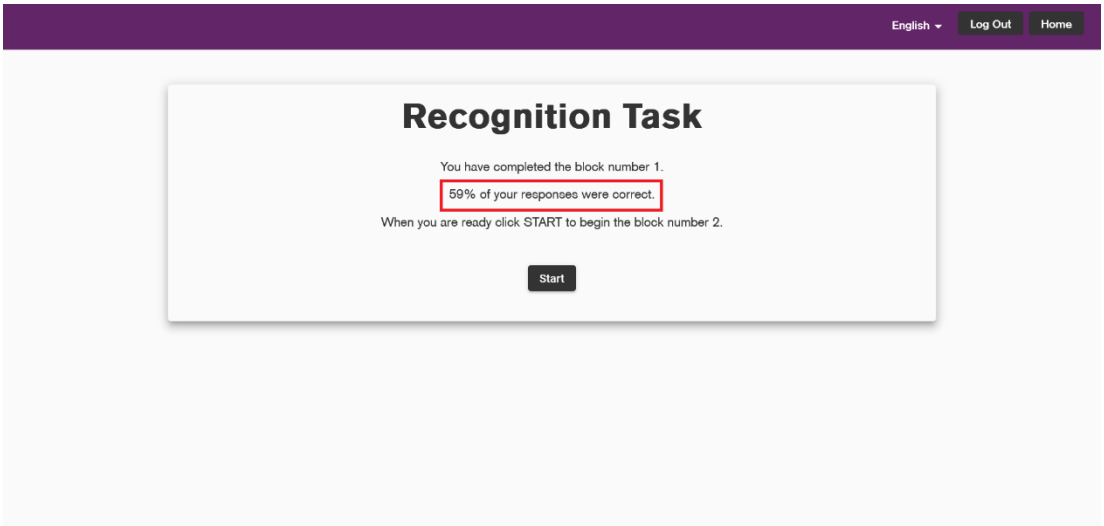

Figure 58: Word recognition between blocks screen - new.

4.3.9 Flash beep task

The task consists of some instructions followed by 8 practice trials and 120 experimental trials, with breaks every 40 (Fig. 59 – Fig. 61). As it can be noted from figure 61, the circle drawn on the screen was not displayed properly since it was cut, and some other backend related bugs were present as well. Moreover, in the instructions screen there is a little form that asks users to indicate the used audio device. The old version allowed to start the experiment without answering the form or ticking both options. This problem has been fixed in the new version (Fig. 62) and also in target detection and loudness perception tasks, which presented the same issue. The circle in trial view is now fully visible (Fig. 65) and an “instructions” button has been added in practice screen to access an instructions recap (Fig. 63 – Fig. 64).

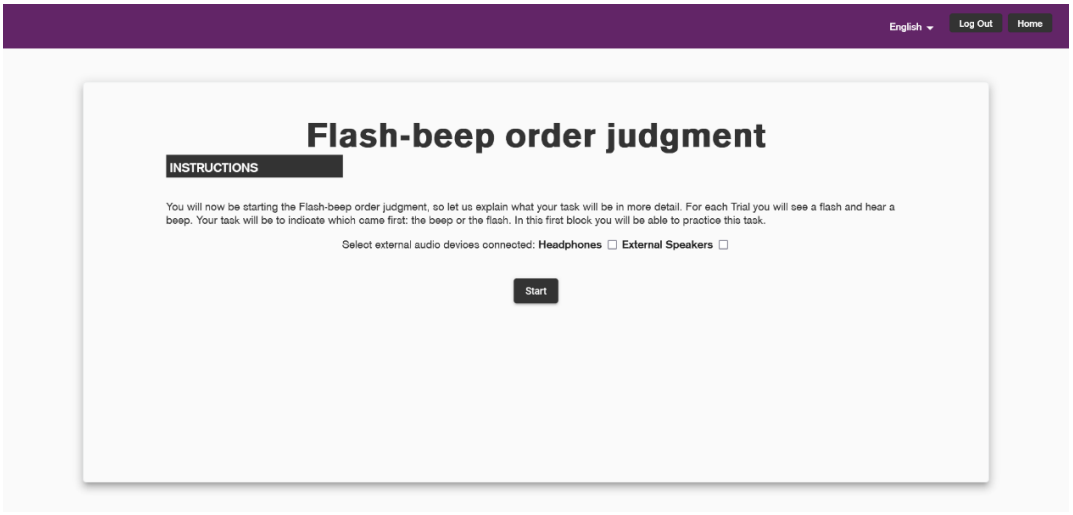

Figure 59: Flash beep instructions - old.

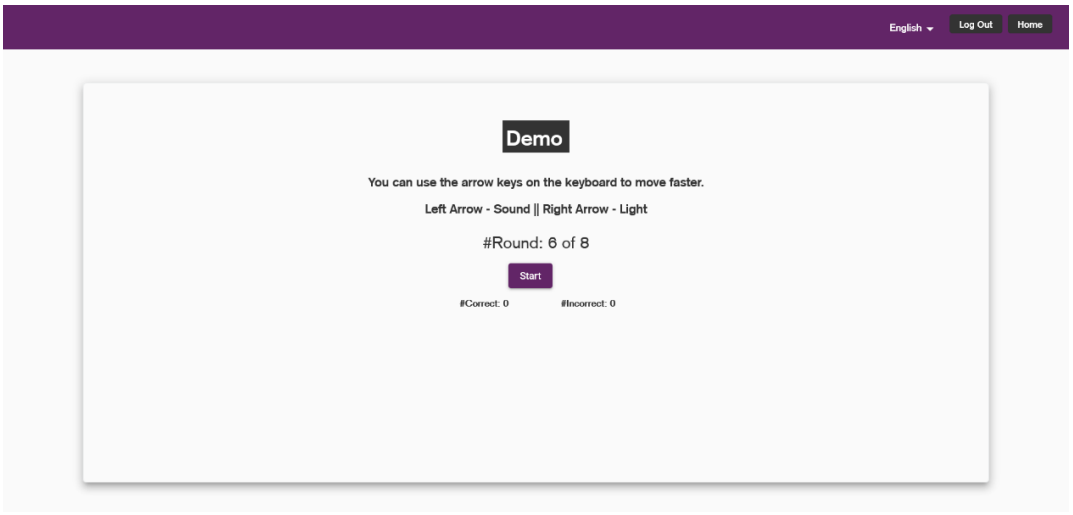

Figure 60: Flash beep trial 1 - old.

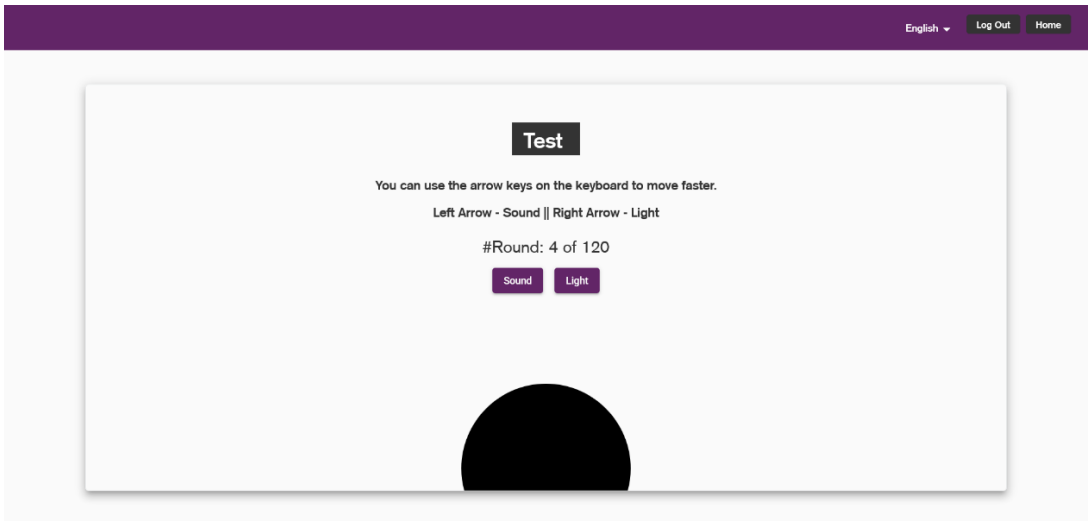

Figure 61: Flash beep trial 2 - old.

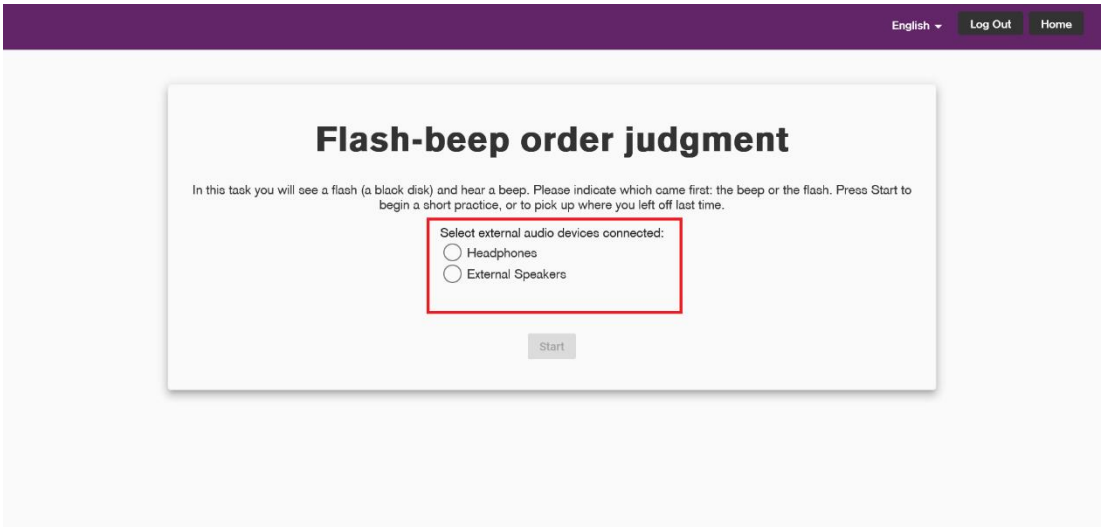

Figure 62: Flash beep instructions - new.

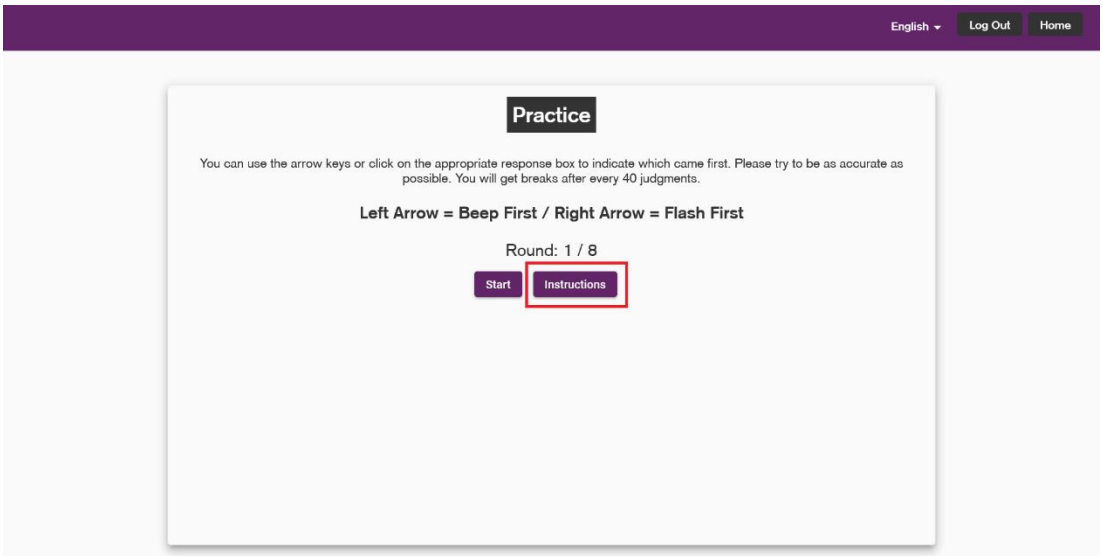

Figure 63: Flash beep trial 1 - new.

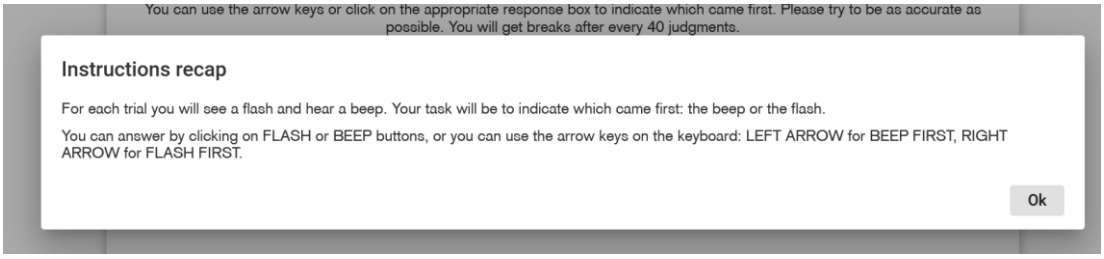

Figure 64: Flash beep instructions recap - new.

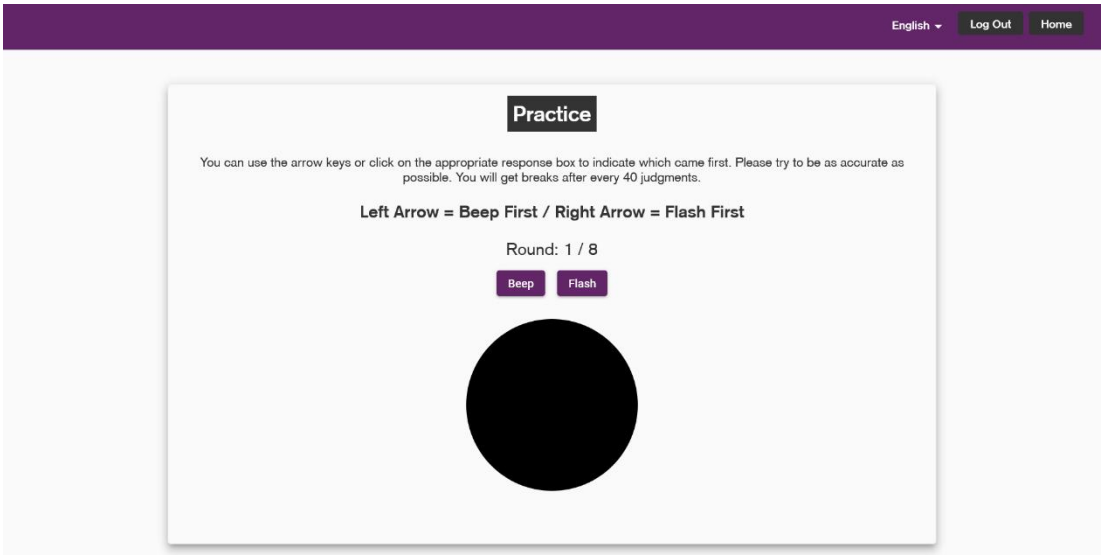

Figure 65: Flash beep trial 2 – new.

4.3.10 Loudness perception

This experiment, in its initial version, consisted of an instructions screen (Fig. 66) with a form to ask the audio device connected (presenting the same bug described in flash beep task) and several additional interfaces to properly configure volume, hear a reference sound and further information (Fig. 67 to Fig. 69). The trial consists of a sound played for 180 seconds and a response scale, which activates every 20 seconds, used to indicate the perceived intensity of the sound with respect to reference one (Fig. 70).

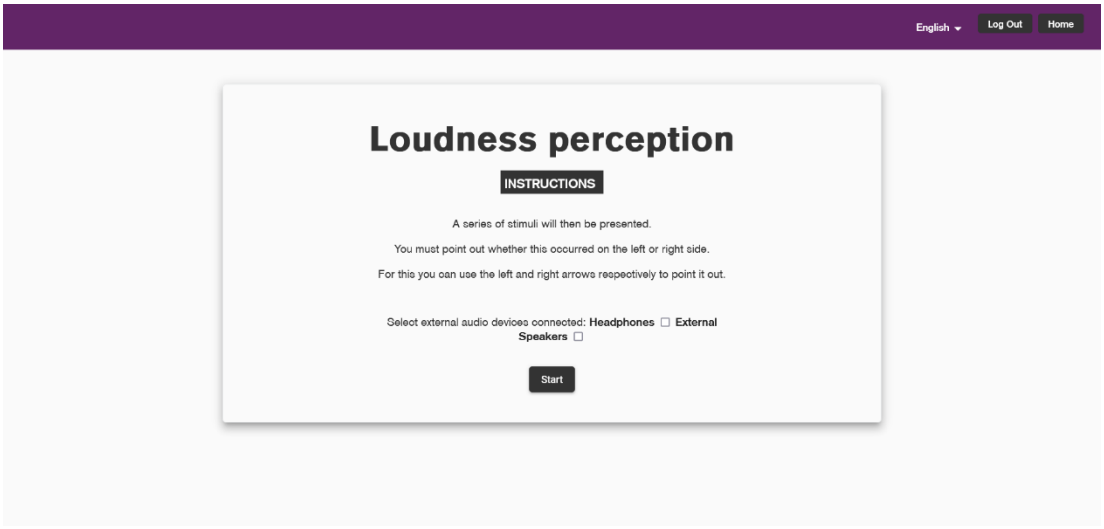

Figure 66: Loudness perception instructions 1 - old.

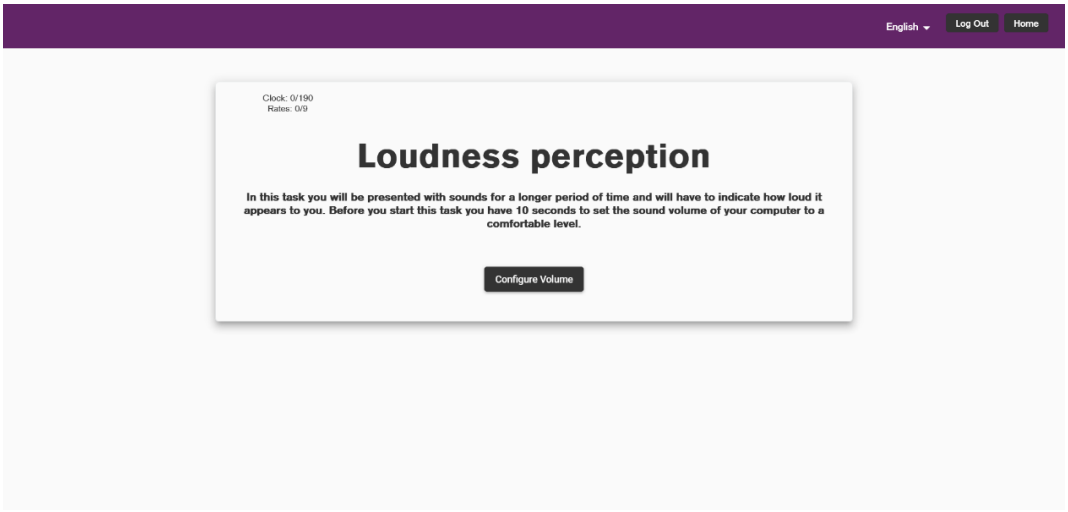

Figure 67: Loudness perception instructions 2 - old.

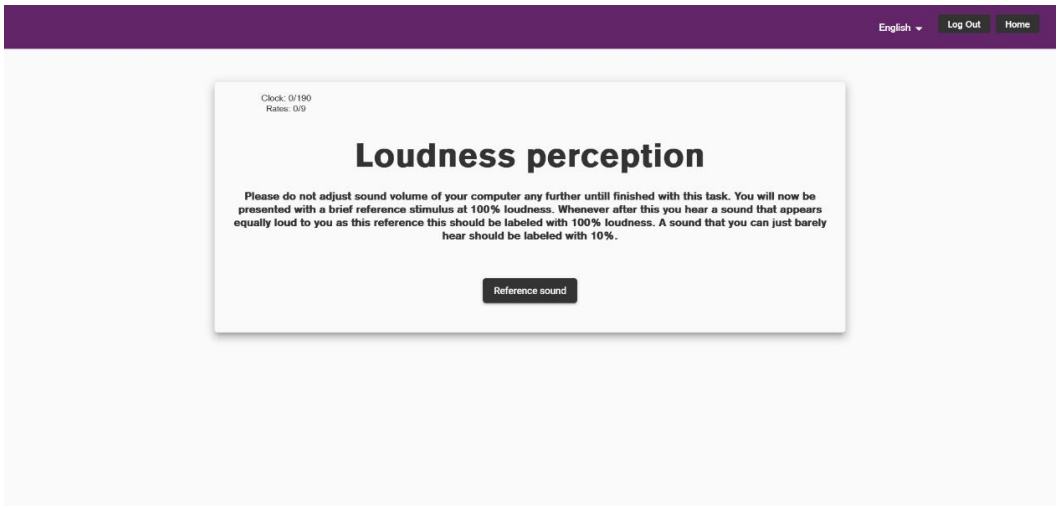

Figure 68: Loudness perception instructions 3 - old.

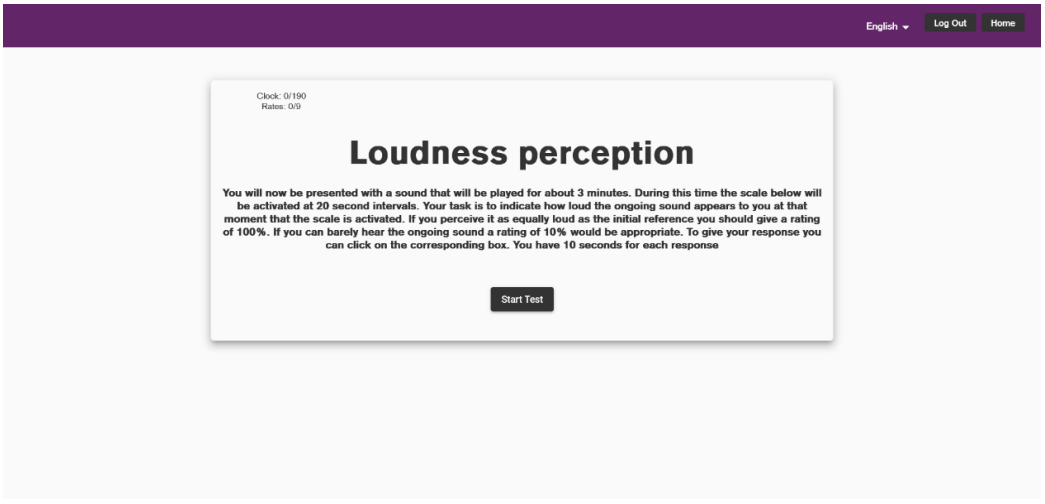

Figure 69: Loudness perception instructions 4 - old.

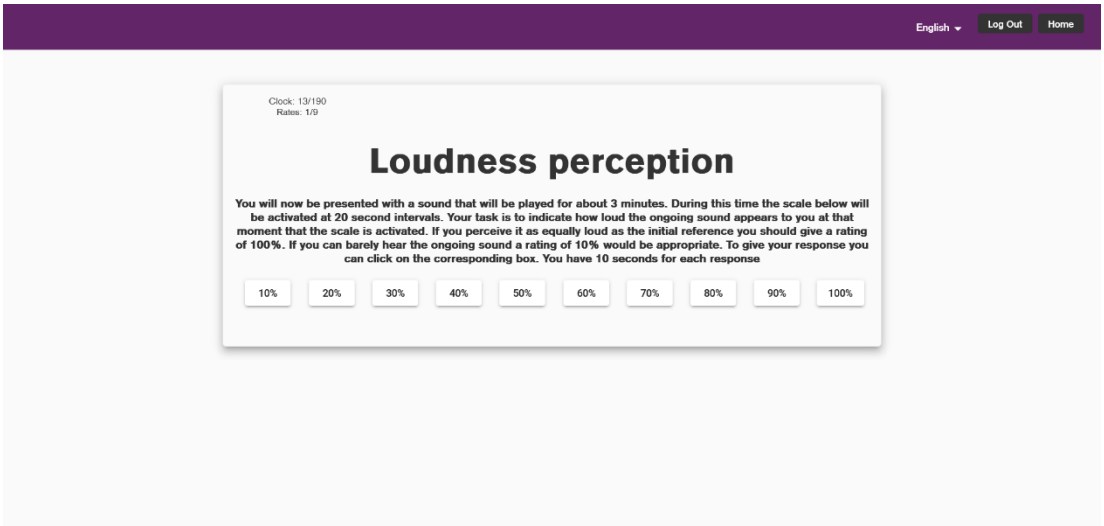

Figure 70: Loudness perception trial - old.

The new version presents updated instructions in all screens and an additional screen if the user selects external speakers as an audio device (Fig. 71 to Fig. 74). The reference sound is not played separately from the trial sound, but it now corresponds to the first 10 seconds of the audio played, as indicated by the novel "100%" label, observable in Fig. 75.

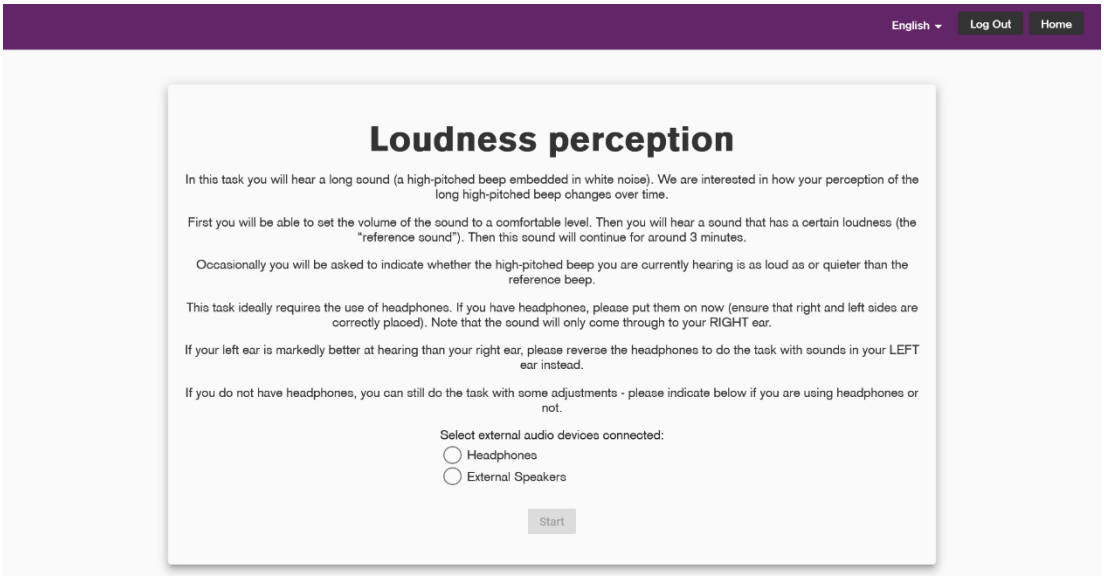

Figure 71: Loudness perception instructions 1 - new.

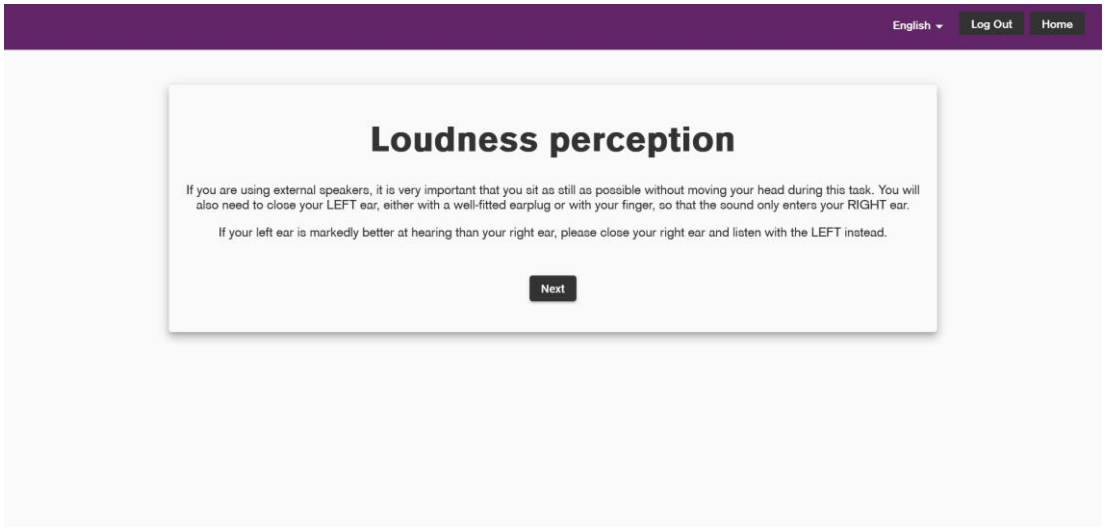

Figure 72: Loudness perception instructions 2 - new.

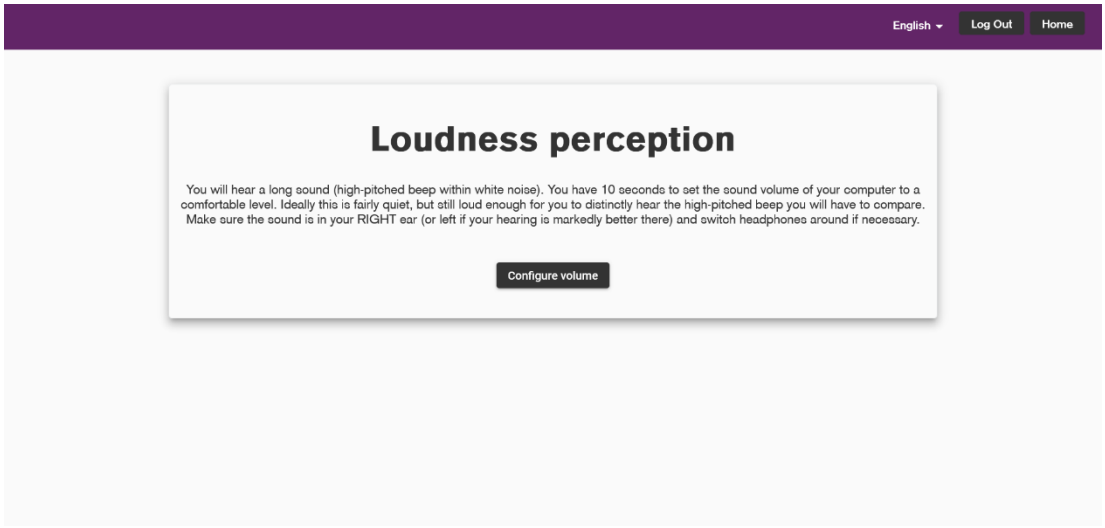

Figure 73: Loudness perception instructions 3 - new.

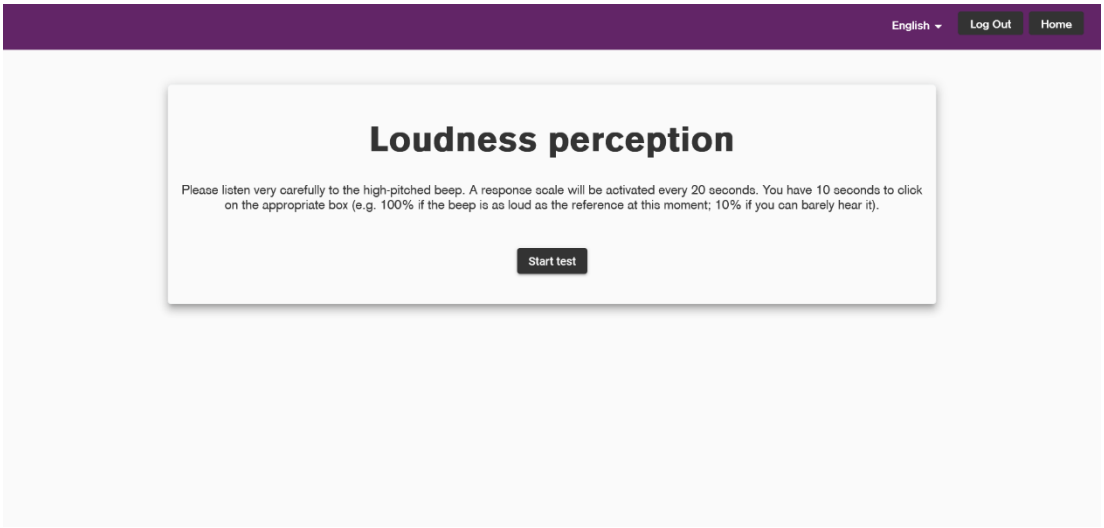

Figure 74: Loudness perception instructions 3 - new.

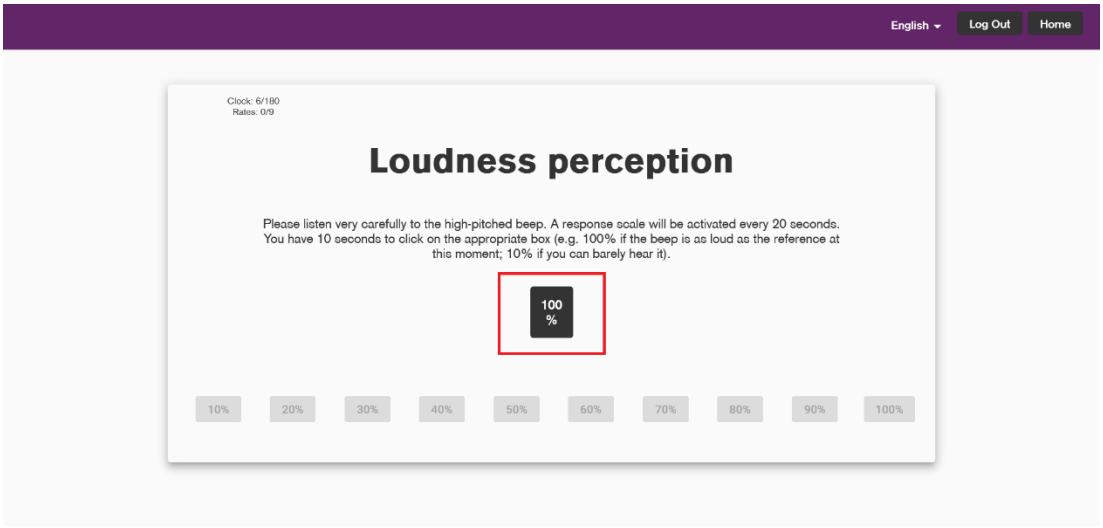

Figure 75: Loudness perception trial - new.

4.3.11 Target detection

This task has been completely rebuilt from scratch since its old version was an "oddball" task (Fig. 76 – Fig. 77) and now is a Posner task. After the instructions screen (Fig. 78), 10 practice trials are prompted on the screen, preceded by a "Get ready" message (Fig. 79 to Fig. 81). For every trial the user has to indicate whether the stimulus is darker on top or on bottom. During practice, in addition, after every answer feedback is displayed (Fig. 82). At the end of the demo it is possible to choose to redo it, at normal or reduced speed (Fig. 83). The actual test is composed of 2 blocks of 128 trials (possibly more, since wrongly answered ones are repeated until correct), with regular breaks every 36 trials, with feedback about performance (Fig. 84 – Fig. 85).

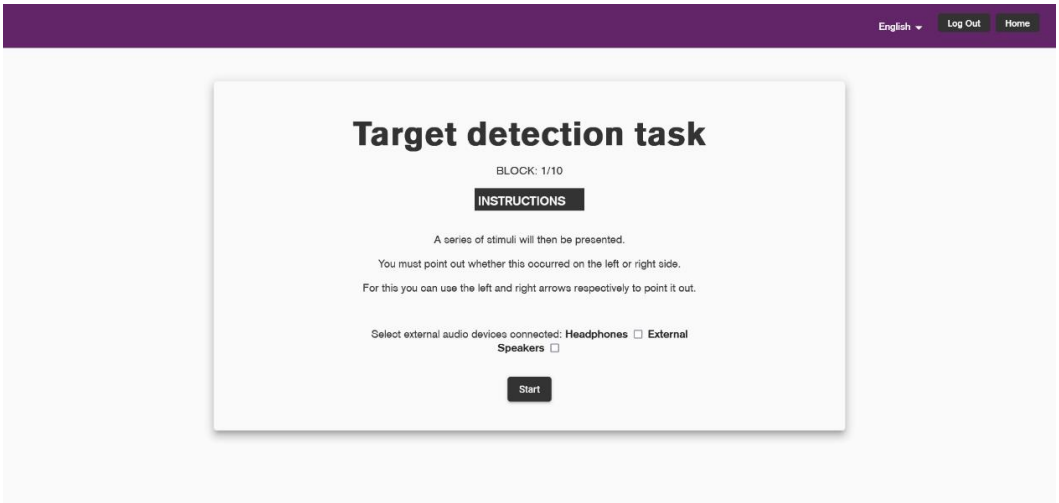

Figure 76: Target detection 1 - old.

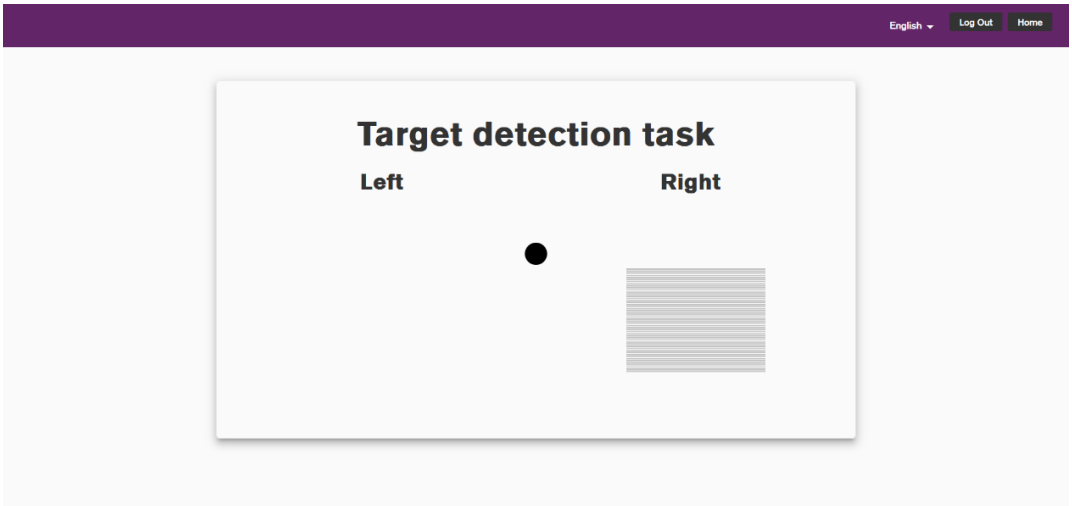

Figure 77: Target detection 2 - old.

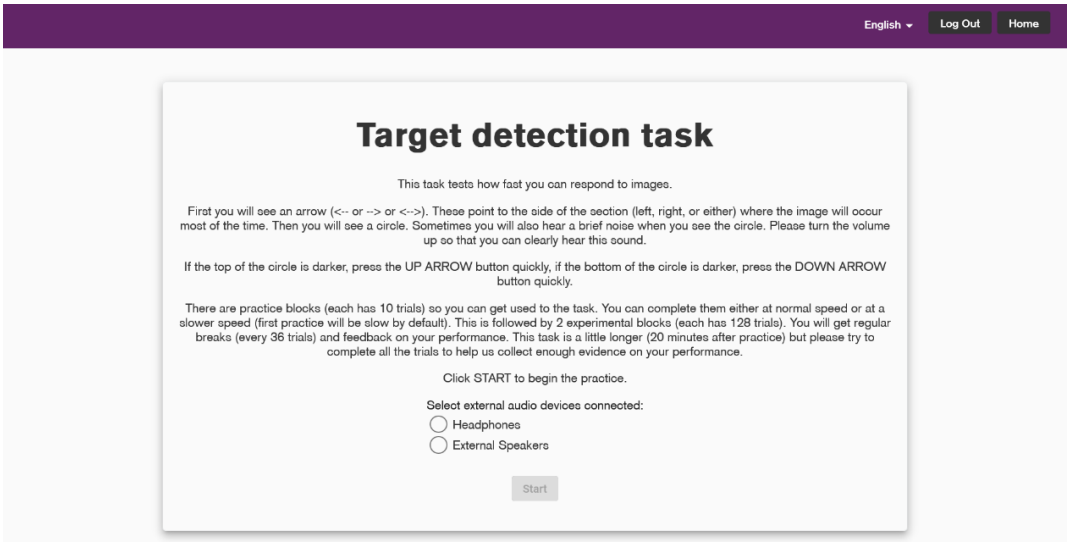

Figure 78: Target detection instructions - new.

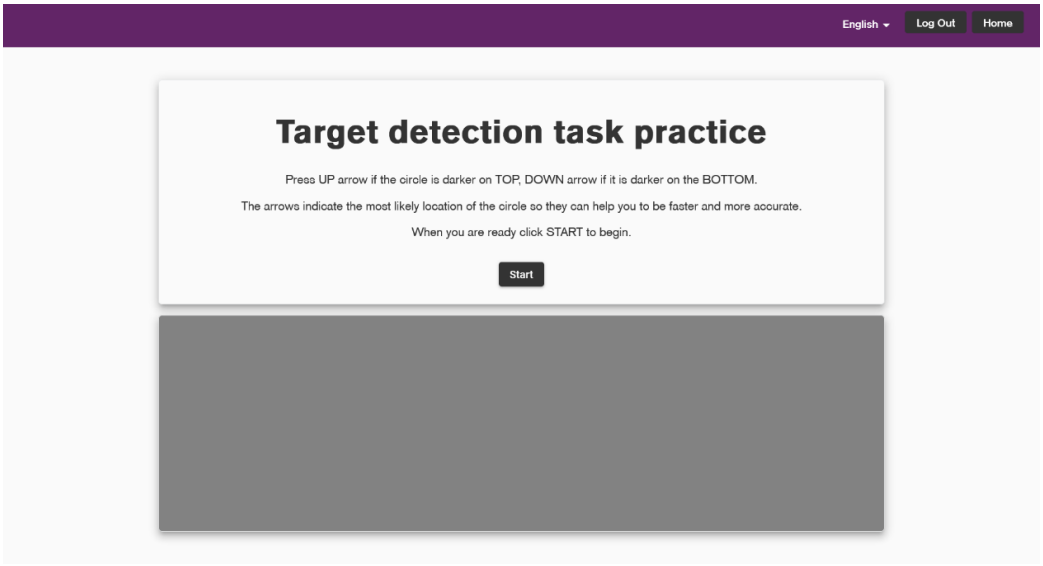

Figure 79: Target detection test view - new

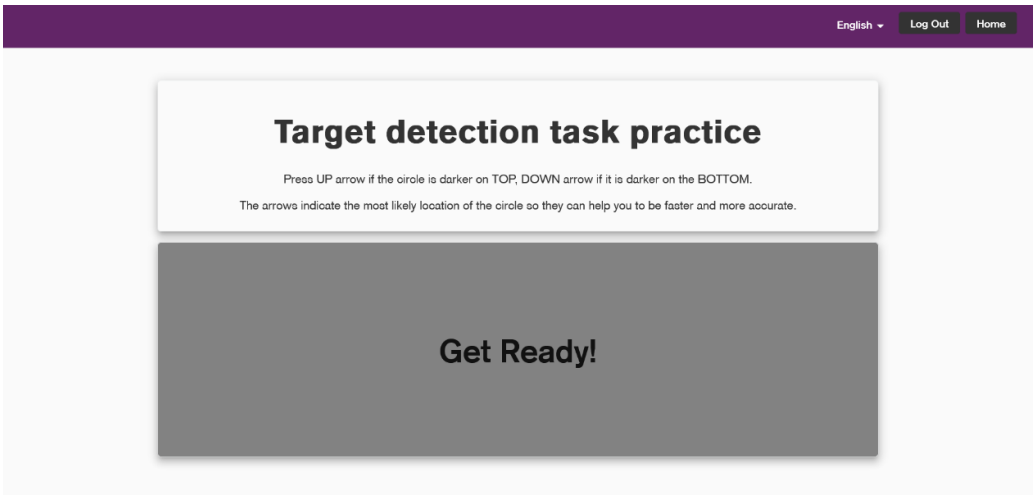

Figure 80: Target detection “get ready” - new.

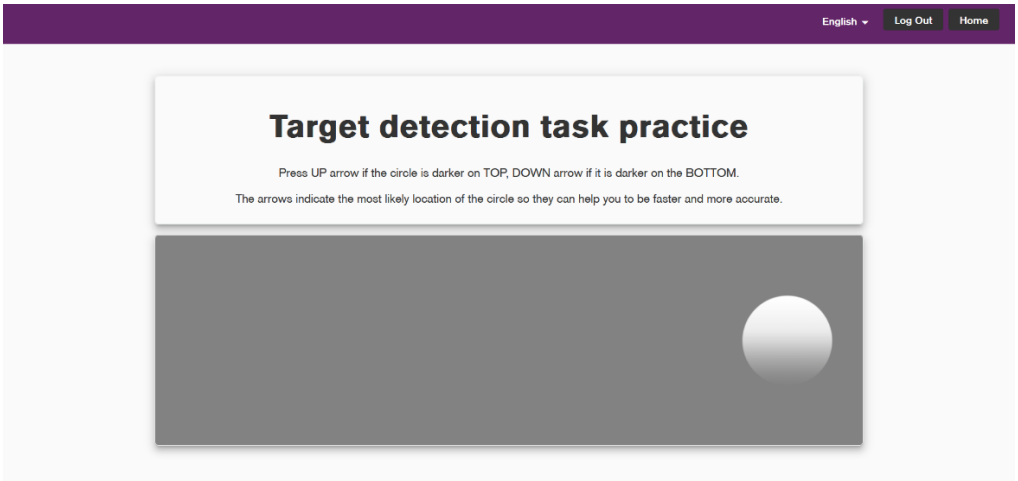

Figure 81: Target detection trial - new.

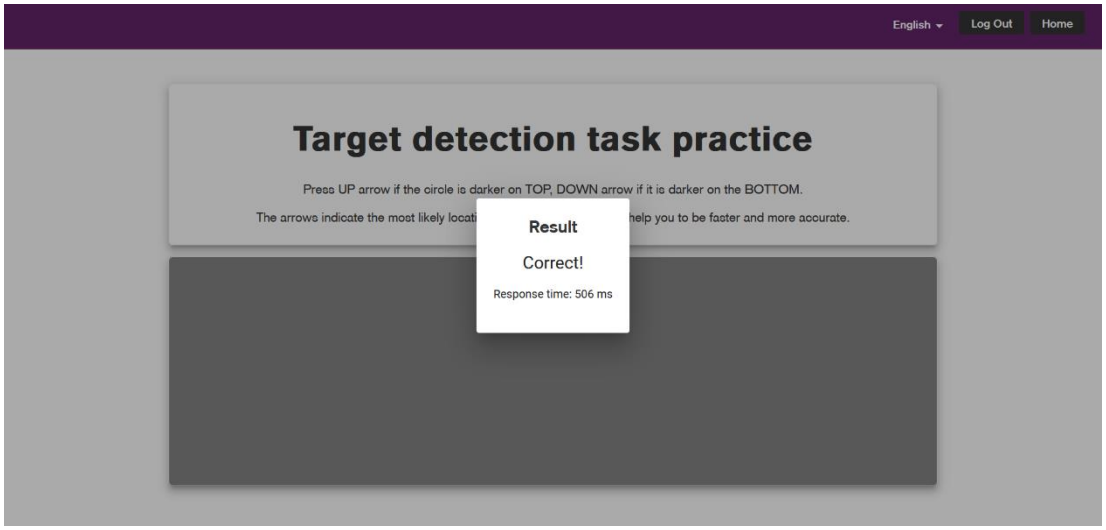

Figure 82: Target detection break - new.

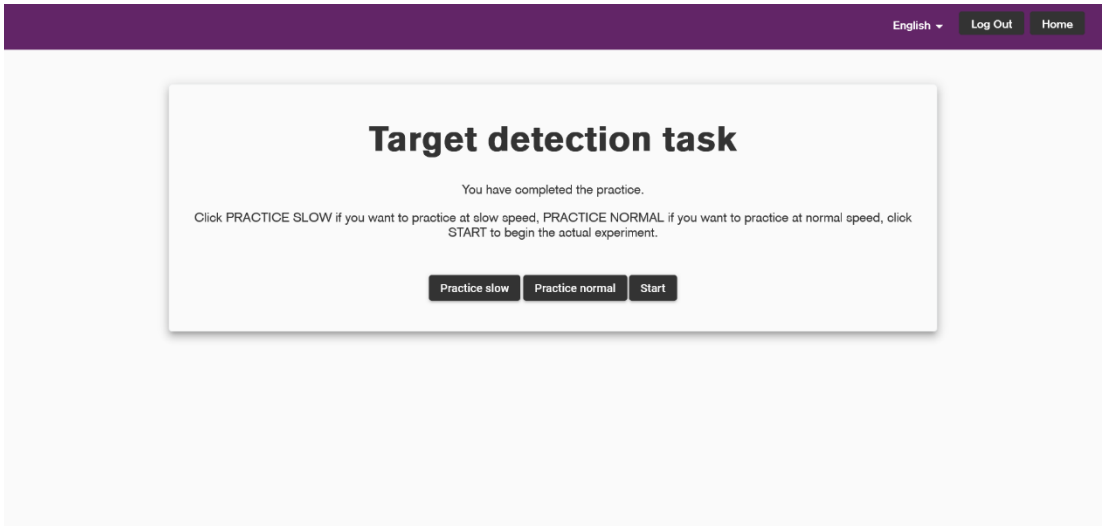

Figure 83: Target end practice screen - new.

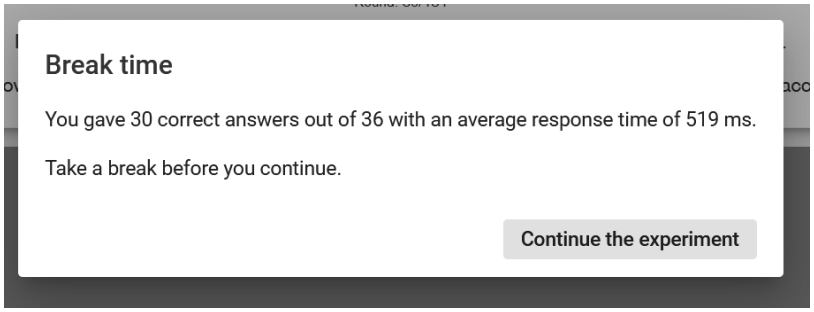

Figure 84: Target detection break feedback - new.

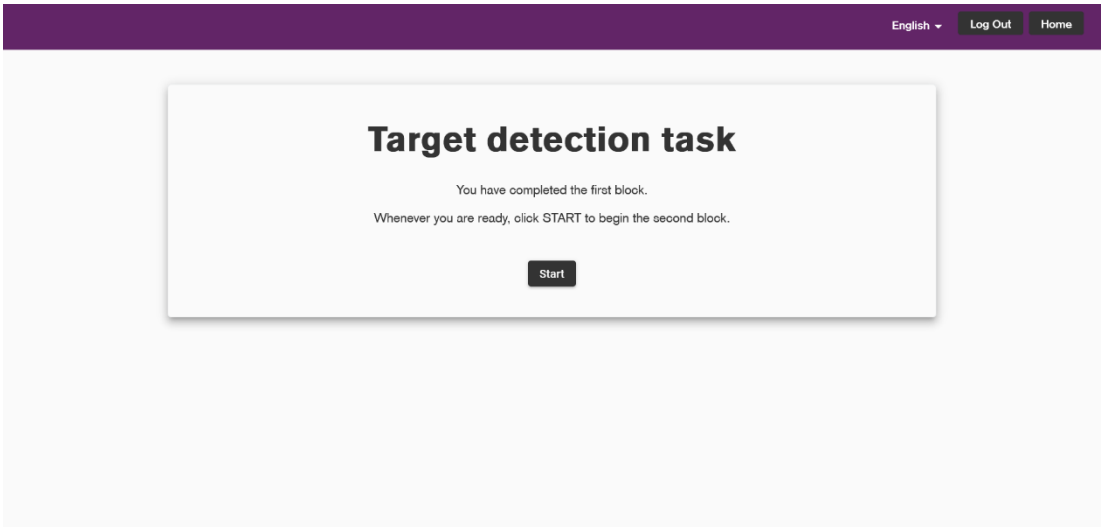

Figure 85: Target detection between blocks screen - new.

4.3.12 Movement perception

This experiment has been newly designed and was not available in the initial version of the application. It first requires the participant to draw a red rectangle as big as a standard bankcard, in order to compute the screen resolution of the device and present the test stimuli in the proper size (Fig. 86 – Fig. 87). Once this step has been completed the instructions of the task are presented (Fig. 88 – Fig. 89), and followed by 2 practice trials, which include step by step instructions (Fig. 90 to Fig. 94). At the end of the practice it is possible to redo it, or move to the actual test (Fig. 95).

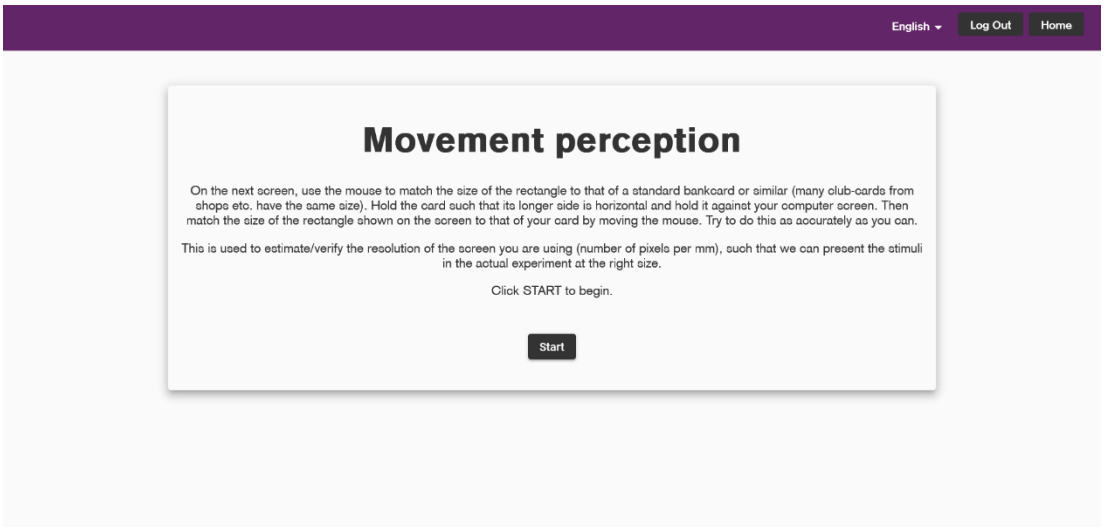

Figure 86: Movement perception instructions 1 - new.

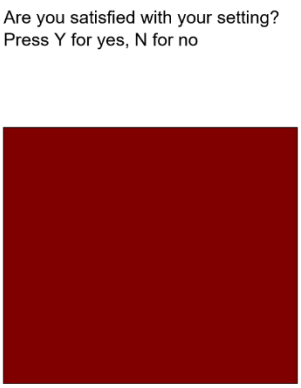

Figure 87: Get resolution screen - new.

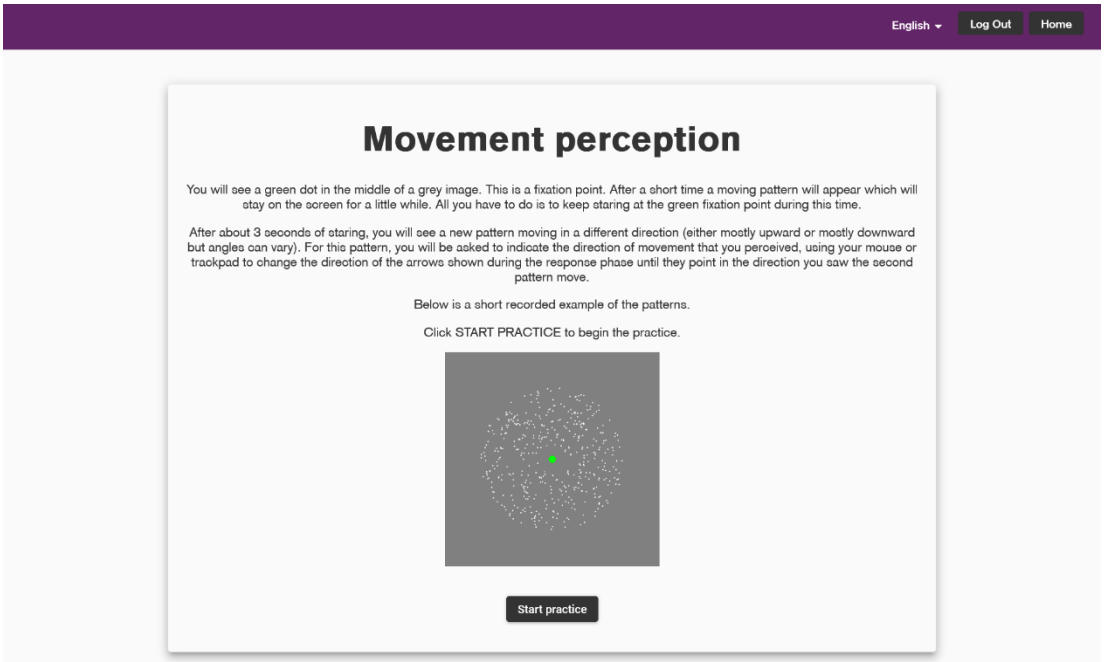

Figure 88: Movement perception instructions 2 - new.

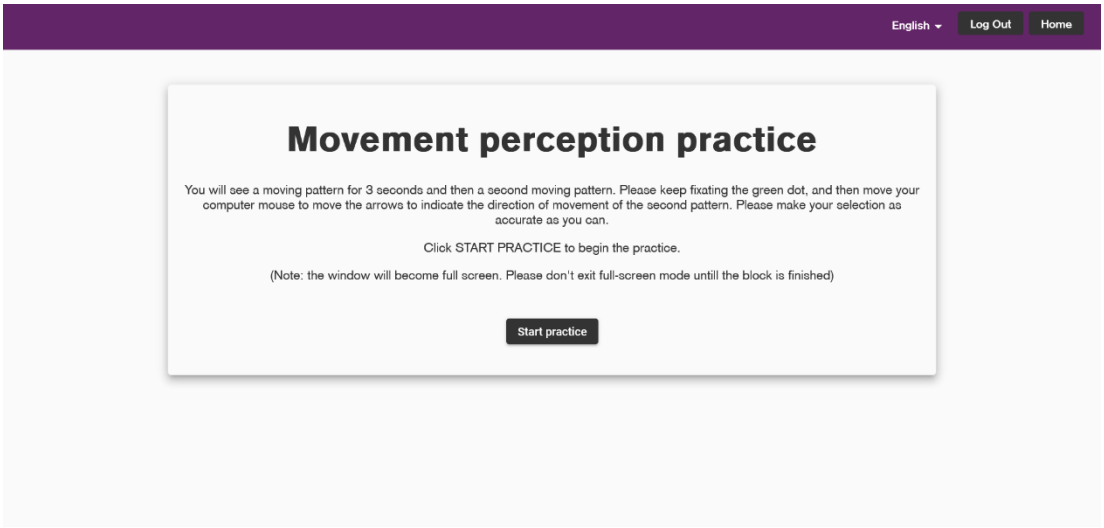

Figure 89: Movement perception instructions 3 - new.

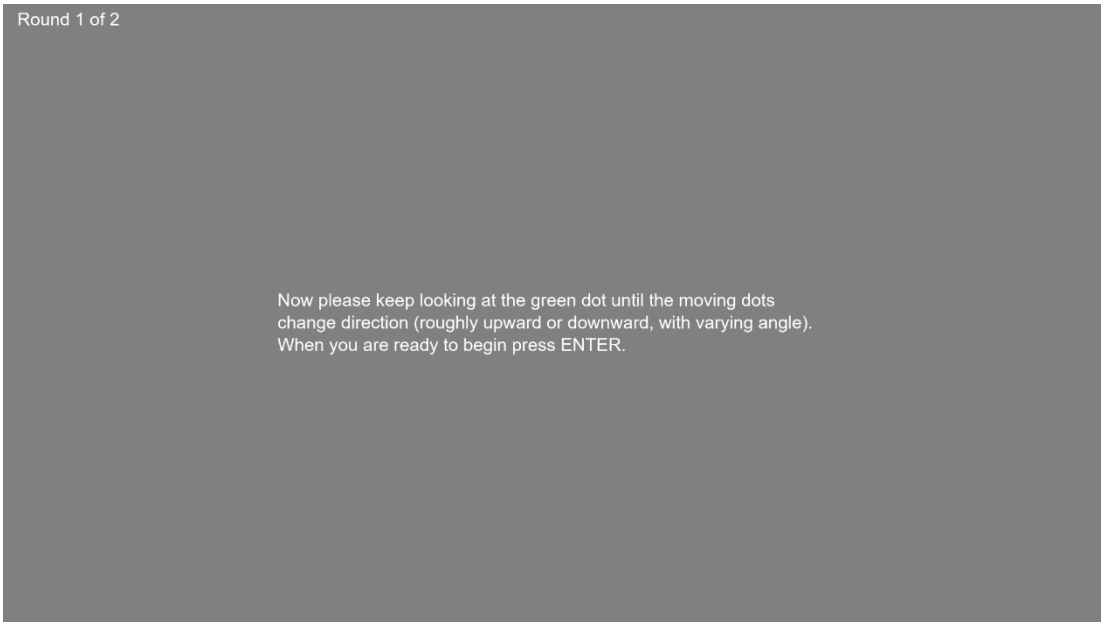

Figure 90: Movement perception trial 1 - new.

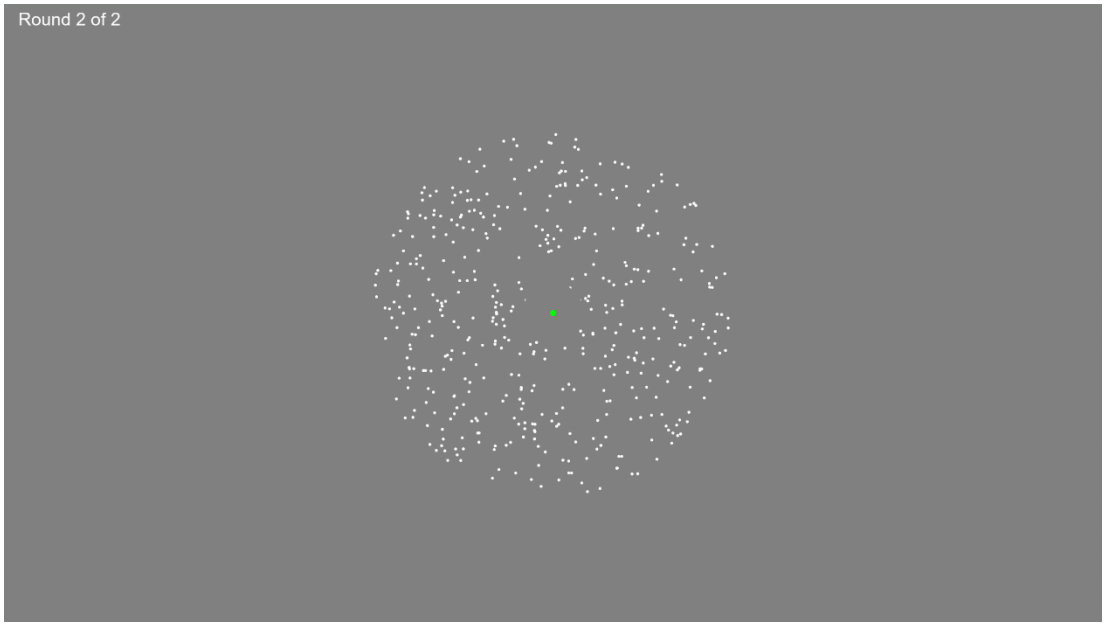

Figure 91: Movement perception trial 2 - new.

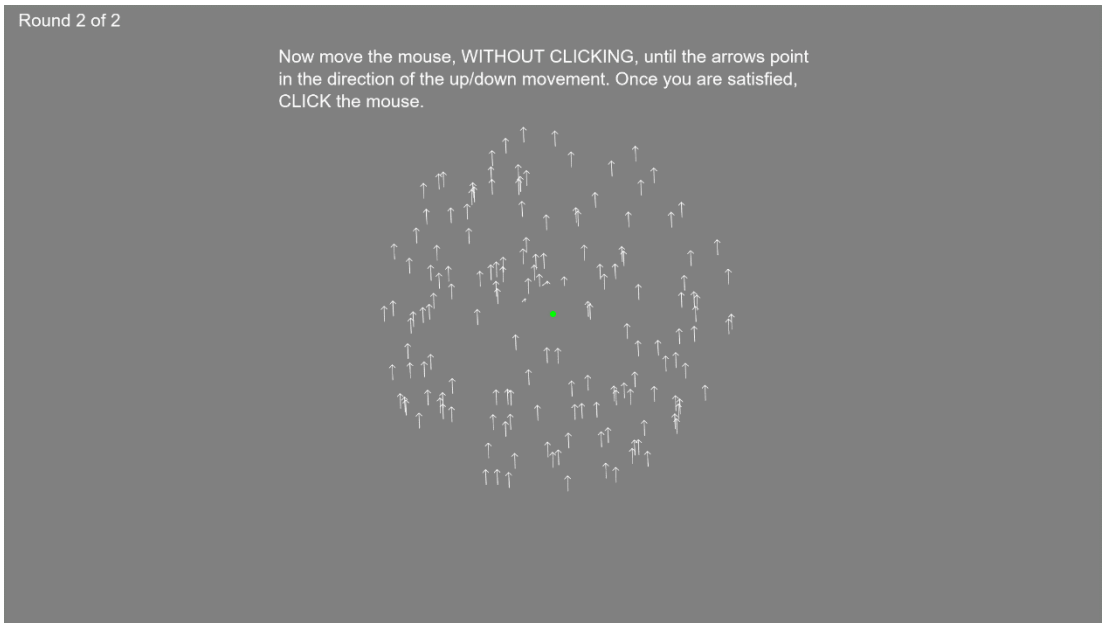

Figure 92: Movement perception trial 3 - new.

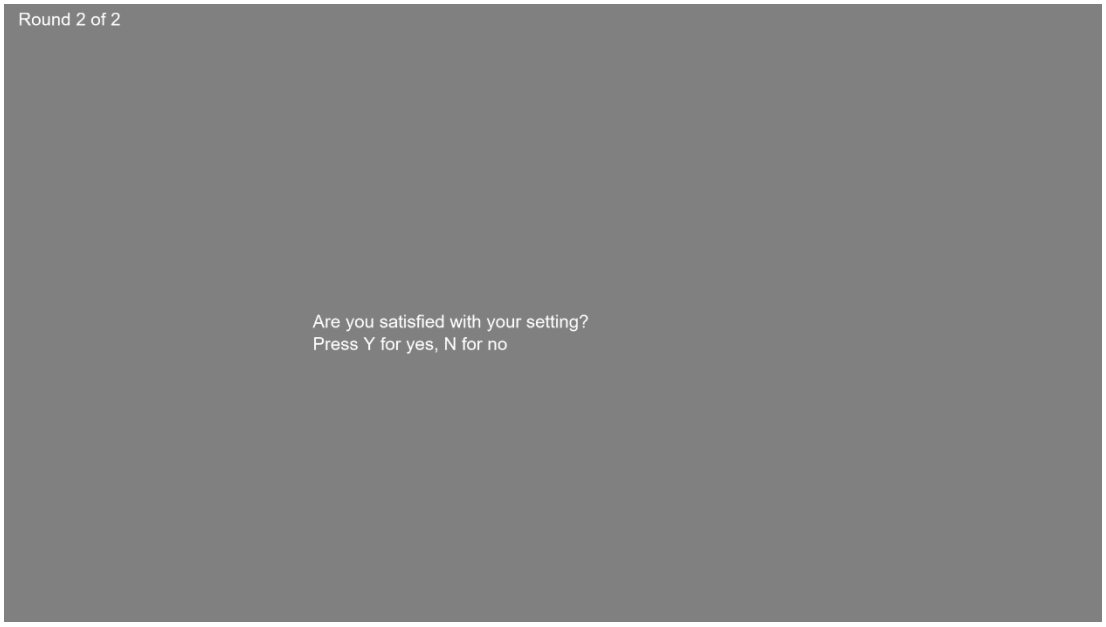

Figure 93: Movement perception trial 4 - new.

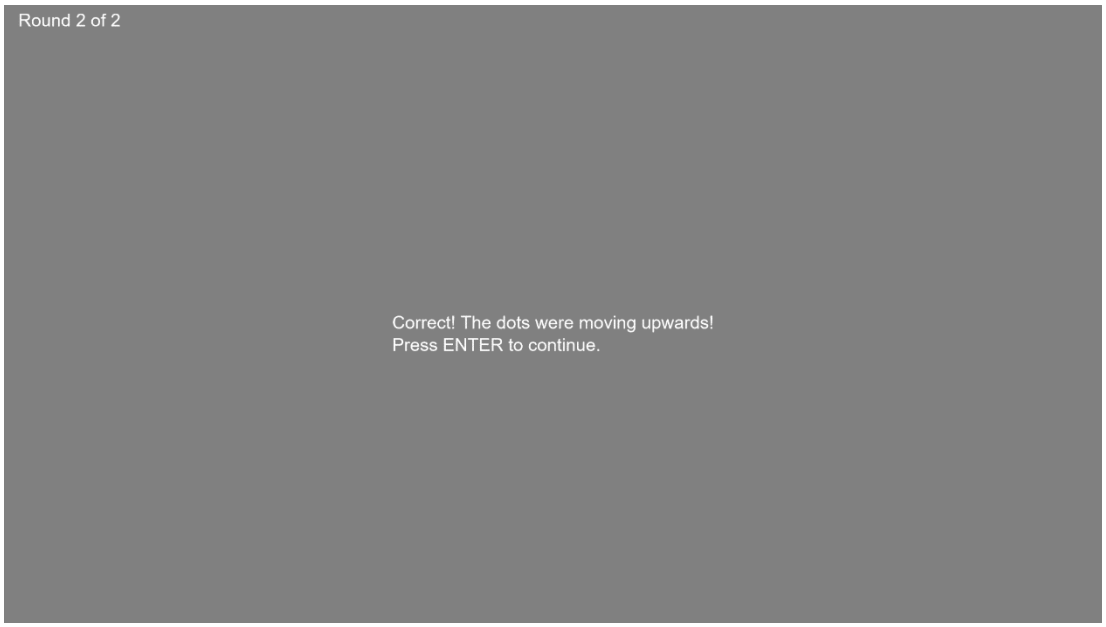

Figure 94: Movement perception trial 5 - new.

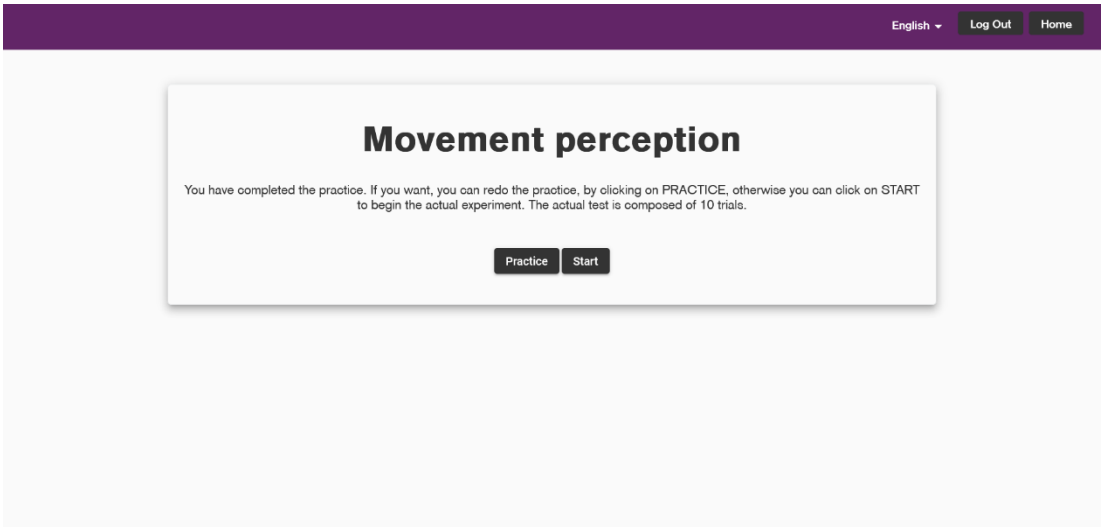

Figure 95: Movement perception end practice - new.

Chapter V

Future steps and Conclusion

5.1 Future steps

At the time of writing this thesis, even though the application is at an advanced stage, some additional steps need to be taken before it can be effectively used to gather the data necessary to perform the intended study.

First of all, some remaining refinements, even if of a minor entity, must be completed. Thereafter, the application needs to undergo an extensive testing, both by computer scientists, looking for any code-related problem, and by psychologists, who will act as the end-users, assessing the overall user experience, as well as the correctness of the proposed tests. It should be pointed out that, as a consequence of this deep testing, some additional requirements may come up. Since the code is organized in a modular structure, the required modifications are expected to be applied with no significant difficulty. Lastly, the necessary configurations to make the application accessible from the Internet must be done: this includes the proper setting of both the *Apache2* server and the *MySql* server, paying particular attention to security-related issues.

Once the application has been thoroughly tested and definitely terminated, the next big challenge for the future work will be creating a framework for analysing the data obtained from the actions carried out by the participants. This could entail creating a prediction model that, based on the answers gathered, provides details on behavioural markers, in order to determine how COVID-19 has affected the subjects' cognitive abilities.

It should be also noted that, although independent, the current application still significantly relies on researchers' monitoring of participants, since they might need assistance while being assessed. Future development must ensure that the software can be utilized without any additional assistance. This might be accomplished by developing an integrated platform that offers real-time support and monitoring access for researchers.

According to these statements, it is simple to understand how the hardest part of the development has already been carried out, leaving only some details and final testing to be completed before the application can be considered effectively finished. It will be possible, then, to focus on the implementation of complementary features, such as the above-described framework of analysis.

5.2 Conclusion

The web application described in the present thesis is a key component in the research conducted by University of Essex. It constitutes, in fact, a centralised data retrieval tool, through which it will be possible to collect all information necessary to carry out the intended research. The realized software, moreover, has been conceived in order to support future studies that aim to further investigate the long-term neurological consequences of COVID-19.

The developed application, in addition, allows researchers to gather data in a quick, reliable and automated way, and to recruit a significant number of participants, from all over the world. This makes possible a considerable extension of the pool of people who can take part in the study, strengthening the statistical basis on which the potential findings are built on.

The overall project in which the development of the web application object of this piece of work is inserted enables the coexistence and cooperation of neuroscience, psychology and software engineering, with the ultimate goal of making progress in our understanding of COVID-19 and its neurological effects, in order to develop new therapies, aiming at improving significantly patients healing process.

Bibliography

[1] "WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020." https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020 (accessed Sep. 20, 2022).

[2] "WHO Coronavirus (COVID-19) Dashboard | WHO Coronavirus (COVID-19) Dashboard With Vaccination Data." https://covid19.who.int/ (accessed Sep. 20, 2022).

[3] Y. Jin *et al.*, "Virology, epidemiology, pathogenesis, and control of covid-19," *Viruses*, vol. 12, no. 4. MDPI AG, 2020. doi: 10.3390/v12040372.

[4] C. Chen, S. R. Haupert, L. Zimmermann, X. Shi, L. G. Fritsche, and B. Mukherjee, "Global Prevalence of Post COVID-19 Condition or Long COVID: A Meta-Analysis and Systematic Review," *J Infect Dis*, Apr. 2022, doi: 10.1093/INFDIS/JIAC136.

[5] S. P. Singh, M. Pritam, B. Pandey, and T. P. Yadav, "Microstructure, pathophysiology, and potential therapeutics of COVID-19: A comprehensive review," *Journal of Medical Virology*, vol. 93, no. 1. John Wiley and Sons Inc, pp. 275–299, Jan. 01, 2021. doi: 10.1002/jmv.26254.

[6] A. Parasher, "COVID-19: Current understanding of its pathophysiology, clinical presentation and treatment", doi: 10.1136/postgradmedj-2020.

[7] M. I. Stefanou *et al.*, "Neurological manifestations of long-COVID syndrome: a narrative review," *Therapeutic Advances in Chronic Disease*, vol. 13. SAGE Publications Ltd, Feb. 01, 2022. doi: 10.1177/20406223221076890.

[8] R. J. Addante, "A critical role of the human hippocampus in an electrophysiological measure of implicit memory," *Neuroimage*, vol. 109, pp. 515–528, Apr. 2015, doi: 10.1016/j.neuroimage.2014.12.069.

[9] R. P. Lawson, J. Aylward, S. White, and G. Rees, "A striking reduction of simple loudness adaptation in autism," *Sci Rep*, vol. 5, Nov. 2015, doi: 10.1038/srep16157.

[10] P. Gòmez Reyna, "An extension on the cognitive effects of COVID-19 through web app development," 2022.

[11] K. Bittner and I. Spence, *Use case modeling*. Addison-Wesley, 2003.

[12] "Three-tier architecture - ManagementMania.com." https://managementmania.com/en/three-tier-architecture (accessed Oct. 10, 2022).

[13] “MVVM: in pratica, breve introduzione alla teoria - VIVIDO.” https://www.vivido.it/blog/2013/06/25/mvvm-in-pratica-breve-introduzione-alla-teoria/ (accessed Oct. 10, 2022).