Politecnico di Torino

Master of science program in Computer Engineering

Project of Thesis

Neurological Consequences of Covid-19

Web app design for behavioral tests

Supervisors

Gabriella Olmo
Dr Vito De Feo

Candidate

Ignacio Jara Sáez

December, 2021

Abstract

Covid-19 infection can cause persistent symptoms, including fatigue, disorientation, concentration difficulties and memory problems, long after recovery. In the proposed project, an application was developed to obtain information about key behavioural markers indicative of neuronal integrity (sensory integration) and cognitive functioning (attention and memory) as biomarkers of Covid-19 neurological damage. This is done with an online study, in order to be able to relate the effects and severity in terms of persistent symptoms in patients who have already been infected.

The application allows participants to perform a battery of tests, consisting of a series of cognitive tasks with visual and auditory stimuli. From the answers given by users, behavioural markers will be obtained, which will be recorded in the application. The purpose of this is to centralise and streamline the process of obtaining information through the use of technology.

The behavioural markers to be assessed are: 1. Speed and accuracy of response to visual and auditory stimuli, 2. The "temporal binding window" of sensory integration, 3.

This paper describes the project, the methodology employed and provides details on the design. In addition, the architecture of the system and its individual parts are presented.

Dedication

I would like to begin by thanking my family, especially my parents, who instilled values in me, supported me and gave me the possibility to develop myself educationally in an optimal way, which allowed me to advance in a better way in this university process.

I thank both the Politecnico di Torino and the Pontificia Universidad de Chile, for allowing me to study in these establishments and to nourish myself with the knowledge provided by the different professors of excellence that both universities have.

I would like to thank the University of Eseex and my guiding professors, Dr Vito De Feo and Gabriela Olmo, for supporting me throughout the thesis process, especially in difficult moments, trusting me completely.

Finally, I would like to thank my friends and Maria Jose for their unconditional support, and for being there for me at all times to help me achieve my goals.

Contents

Chapter 1

Introduction

In late December 2019, a new variant of coronavirus emerged in Wuhan, China, known as SARS-CoV-2. This virus is highly contagious, as it is transmitted from person to person through respiratory secretions expelled by coughing, sneezing and talking. Given the rapid rate of transmission of this new strain, it quickly spread around the world, generating a high mortality rate in the population. This has led to a humanitarian crisis in several parts of the world, as well as an economic, social, political and public health crisis.

Faced with this health emergency, which continues to this day, the scientific community has been working intensively on research to contribute to the evidence associated with this virus, in order to understand in depth both the signs and symptoms and the pathogenesis of this virus and therefore the effects and comorbidities that it could cause in the short and long term in the future. This new strain has had a major impact worldwide as it affects different systems to varying degrees, depending on multiple factors such as viral load, immune response and associated risk factors such as hypertension, diabetes mellitus and obesity. While it is known that COVID-19 can cause lung scarring, heart and kidney damage, doctors and researchers are also beginning to measure the long-term neurological potential of the virus [1]. This ultimately impacts on people's ability to re-engage in the activities of daily living.

Preliminary studies have suggested that even the immune response triggered by the viral infection can lead to serious illness and complications, which may also damage the central nervous system [1]. Persistent symptoms following infection with the virus include fatigue, lack of concentration and memory problems, which persist for a long time after recovery. This is why more information is needed to make better clinical and national decisions to better control the pandemic and its consequences.

This project aims to create a technological tool that can determine, through different markers, the cognitive and neuronal functioning after infection by the SARS-CoV-2 virus, in order to obtain information on its effects that will be useful for both the scientific community and the population. Behavioural and brain markers indicative of neuronal integrity and cognitive functioning, i.e. attention and memory, will be used. These provide information about the speed and accuracy of response to visual and auditory stimuli. The behavioural markers to be assessed are:

- Speed and accuracy of visual and auditory stimuli.
- "Temporal Binding window" of sensory integration
- Sensory adaptive responses
- Retention of verbal information

Participants will undergo proven experimental tasks commonly used in perceptual and cognitive research to investigate unisensory and multisensory processing.

The tests used in this project include the Temporal order Judgement task, Memory task, Sensory adaptation task, Oddball detection task, Schizotypal personality questionnaire and covid-related questionnaire, with a total time of 80 minutes.

The first task consists of a temporal order judgement between two stimuli; a visual flash and an auditory beep. Participants must judge which one is presented first.

The second is divided into two assessments, one of which consists of word coding, i.e. discriminating whether the word is a living thing or not. The second part consists of recognising how old the word is on a scale from 1 to 5, where 1 is new and 5 is old.

The Sensory adaptation task allows to evaluate the judgement of volume as a percentage of the initial perception of volume.

Oddball detection task allows observing auditory, visual and audio-visual stimuli on the left and right sides of the screen, then responding to targets on one side, while ignoring distractors on the other side. Schizotypal personality questionnaire where the participant has to choose from four boxes how much he/she agrees with the options presented.

Finally, the covid-related questionnaire consists of the participant answering questions according to how he/she experienced the disease according to its symptoms and evolution. There are several types of answers, where the next question depends on the answer given in the previous question.

The data collected will provide information on speed and accuracy of response, temporal binding window, sensory adaptive responses and retention of verbal information. Participants will sign a consent form and will also be given a document with details of how the data will be used, how they will be contacted for follow-up and how they can withdraw from the study at any time they wish. At the end of the test, participants will be eligible to win a prize equivalent to £20 as a thank you for contributing to the research.

Given the health measures implemented by different countries to protect their citizens and thus avoid contagion and the collapse of public health, it is very useful to generate systems that allow this information to be obtained remotely. This facilitates the generation of information, as people who are willing to participate can take the tests from the comfort of their own home. Participants must complete a survey detailing their demographics (age, gender, education, etc.) and medical history (neurological conditions, sensory impairments, history of covid-19). In addition, they should complete information regarding their mental well-being (measured by the Warwick-Edinburgh Mental Well-being Scale), anxiety status in relation to Covid-19 infection, schizotypal traits (Schizotypal Personality Questionnaire, SPQ) and dissociative experiences (Dissociative Experiences Scale).

This project was carried out in conjunction with the Department of Psychology at the University of Essex, where researchers Dr. Helge Gillmeister, Dr. Loes van Dam, Dr. Caterina Cinel and Dr. Vito de Feo provided assistance with the requirements and design of the different tests. In addition, at an early stage of the project, a group of 12 students also supported in the development of requirements and structuring of the project.

Chapter 2

Justification

2.1 Context

In recent years, the world has experienced a strong impact on health caused by the Covid-19 viral infection (coronavirus disease 2019), caused by a new variant of the virus known as SARS-CoV-2 (severe acute respiratory syndrome coronavirus 2). This has had different repercussions in different countries, both developed and underdeveloped, putting sanitary measures and public health policies in check. This virus is highly contagious, mainly through the secretion of respiratory droplets (coughing, sneezing) and saliva, which has set off alarm bells about this new outbreak, causing a pandemic recognised by the World Health Organisation (WHO).

Given the sharp increase in contagion and the lack of preparedness for prevention in these public health situations, the limitations of countries' health systems have been exposed, with the least developed countries showing the greatest deficiencies and inequalities in implementing and controlling the pandemic. Hygiene measures such as disinfection of objects and spaces for public and daily use, hand washing and the use of masks have had to be implemented. This complex scenario has generated a political, economic, social and health crisis.

Governments have been overwhelmed and have had to restrict the free movement of their citizens in order to contain transmission while the scientific community works hard to obtain evidence to help control the pandemic and avoid further complications. At the economic level, there has been a stagnation in development, as companies have had to urgently implement health measures, many of them improvised, to allow their employees to work safely. This has thus affected different aspects of people's lives, leading to various collateral biological and mental health problems. Health care workers directly see how resources are stretched and are confronted with ethical dilemmas, such as which patient uses the last available bed. Thus, infection seems to be a cross-cutting issue for everyone, regardless of social strata.

The hope for a gradual return to normality is centred mainly on the vaccines that different laboratories have been developing in order to protect the population and avoid humanitarian losses. One of the great advances made by the scientific community is that the SARS-CoV-2 virus has been isolated, making it possible to obtain neutralising antibodies, which prepare the human body, reducing the symptoms and the serious and even fatal consequences of the disease. [2]

2.1.1 Pathogenesis

In late 2019, a new variant of the large coronavirus family was discovered in Wuhan city, Hubei province, China. It was first identified on 1 December of that year following a report of a cluster of people with pneumonia of unknown cause, linked to workers in a wholesale seafood market in southern China. The disease spread rapidly, generating high international impact, leading the World Health Organisation to declare a health emergency on 30 January 2020.

The source of infection is not yet clear and is still under investigation. The most likely cause is that the disease originated from an animal, as it is very similar to bat coronaviruses, making

the latter likely to be the primary reservoir of the virus, with research showing that it is 96% identical to the genome of the bat coronavirus. However, there are articles that rule it out as a possible transmitting agent. Due to the prevalence and wide distribution of coronaviruses in animal species, coupled with the wide genetic diversity and frequent recombination of their genomes, it is likely that new strains will be detected in human cases in the future.

Coronaviruses belong to a large family of viruses that cause disease in both animals and humans. There are six species of coronaviruses that cause disease in humans. Four of these (229E, OC43, NL63 and HKU1) cause common influenza symptoms in immunocompromised people and two species (SARS-CoV and MERS-CoV) cause severe acute respiratory syndrome associated with high mortality rates.

It is an RNA virus, which is round or oval in shape, sometimes even polymorphic. It is 60-140 nm in diameter and has a spike protein on its surface, which is used for typing. This nucleocapsid protein encapsulates the viral genome, allowing it to be used as a diagnostic antigen.

The mechanism of animal-to-human transmission is still unknown. It is thought to have occurred through direct contact with infected animals and their respiratory secretions and/or material from the digestive tract.

On the other hand, the mechanism of human-to-human transmission is mainly by direct contact with respiratory droplets (larger than 5 microns which are able to be transmitted over a distance of up to 2 metres), through hands or by fomites contaminated with secretions followed by contact with mucous membranes of the mouth, nose or eyes. Vertical mother-to-child transmission has even been suggested, but more evidence is needed.

Following infection, the incubation period has been estimated at 4-7 days with an average of 5 days. After infection, it takes about 14 days for symptoms to appear, so the WHO suggests isolation for that length of time. However, there are cases where the virus has been transmitted even after the first 14 days, extending up to 24 days according to a publication by Chinese researchers.[3]

2.1.2 Signs and symptoms

The clinical picture caused by infection with the virus varies from person to person. Most have mild to moderate symptoms, recovering without hospitalisation. These can be classified in different ways, which can be seen in table 2.1 and 2.2.

Cardinal Symptoms	Non-Cardinal Symptoms
• Fever (body temperature of 37.8°C or higher) • Anosmia (sudden and complete loss of sense of smell) • Ageusia (sudden or complete loss of taste).	• Cough • Nasal congestion • Dyspnoea (Shortness of breath) • Tachypnoea (Increased respiratory rate) • Odynophagia (sore throat when swallowing liquids or food) • Myalgia (Muscle pain) • General weakness or fatigue • Chest pain • Chills • Diarrhoea • Anorexia or nausea/vomiting • Headache (headache)

Table 2.1: Classification according to cardinality of symptoms. [4]

Most common symptoms	Less common symptoms	Severe symptoms
• Fever (body temperature of 37.8°C or higher) • Dry cough • Tiredness	• Aches and pains • Sore throat • Diarrhoea • Conjunctivitis • Headache • Loss of taste and/or smell • Skin rash or discolouration of fingers or toes	• Shortness of breath or breathlessness • Chest pain or tightness • Loss of speech or movement

Table 2.2: Classification according to symptom type. [4]

A retrospective cohort study involving 41 patients with confirmed cases of COVID-19 in Wuhan City, China, showed that the average age was 49 years, with a higher male prevalence. The most important signs and symptoms were fever (98%), dry cough (76%), dyspnoea (55%), myalgia or fatigue (44%) and lymphopenia (63%). Other very frequent symptoms as recorded by WHO are expectoration (33%), odynophagia (14%), headache (14%), myalgia or arthralgia (15%), nausea or vomiting (5%), nasal congestions (5%).

In general, the symptoms presented by the disease are usually mild, being mistaken for a common cold or flu. However, 15% of patients develop the most severe symptoms, which require

hospitalisation. About 5% develop very severe symptoms, which require constant monitoring and must be treated in Intensive Care Units. [3]

2.1.3 Clinical presentations

1. **Enfermedad no complicada (mínimamente sintomática)**: se presentan signos no específicos tales como fiebre, tos, dolor de garganta, congestión nasal, cefalea leve y malestar general. Ausencia de signos de deshidratación, disnea o sepsis. Pacientes adultos mayores e inmunodeprimidos pueden presentar síntomas atípicos. Además, pueden existir manifestaciones digestivas como nauseas, vomitos y diarreas. Este cuadro suele confundirse con otras afecciones virales respiratorias, siendo prácticamente indistinguible. [3]
2. **Infección no complicada de las vías respiratorias bajas no complicada (neumonía ligera)**: A los síntomas presentados anteriormente, los pacientes pueden presentar fiebre, tos productiva o crepito o presentarse como una neumonía atípica, pero sin signos de gravedad y con una saturación de oxígeno con aire ambiental ¿90%. Ausencia de signos de mayor gravedad o de insuficiencia respiratoria. [3]
3. **Neumonía (grave)**: Presencia de tos productiva con fiebre, aleteo nasal, taquipnea o presentarse como una neumonía atípica pero con signos de gravedad. Puede haber tiraje intercostal o supraesternal, cianosis central con saturación de oxígeno ¡90% y dolor pleurítico. Puede producir y asociarse a un síndrome de distrés respiratorio agudo. [3]

2.1.4 Diagnosis

For diagnosis, the most commonly used standard method is the detection of RNA by real-time polymerase chain reaction (RT-PCR), the sample of which in most cases is obtained from the nasopharyngeal mucosa. [5]

Several laboratories have been developing vaccines to help prepare the immune system and thus avoid complications such as hospitalisation and, in worst-case scenarios, death.

2.1.5 Radiological manifestations

When the diagnostic suspicion is high but RT-PCR is inconclusive or negative, radiological examinations are an elementary complement, with pulmonary findings being the main ones to guide the diagnosis.

Bilateral subpleural infiltrates can be seen on radiography in 75% of cases, while a quarter will have only unilateral involvement.

Consolidations or bilateral ground-glass areas are seen on computed axial tomography (CAT). In older adults, subpleural predominance with associated pleural thickening may be found. CAT findings may precede RT-PCR positivity and are therefore considered a predictor of severity in some cases.

Ultrasound has a similar sensitivity to CT for detecting pleural thickening, subpleural consolidations and ground glass. Among the benefits is that the exposure of hospital staff is lower compared to CAT. [5]

2.1.6 Prevention and care

Preventive measures to avoid transmission are as follows: [3, 6]

- Frequent hand washing with soap and water or cleaning with hydroalcoholic gel.
- Frequent use of masks covering nose, mouth and chin.

  - When removing the mask, keep it in a plastic bag.
  - If it is made of cloth, wash it every day.
  - If disposable, dispose of it in the rubbish bin.

- Keep a social distance of at least 1 metre between you and people coughing or sneezing.
- Avoid touching your face, especially the conjunctivae of the eye and mouth.
- Cover your mouth and nose when coughing or sneezing.
- Stay at home if you are not feeling well.
- Refrain from smoking and activities that weaken the lungs.

2.1.7 Complications

Complications from this disease, in addition to those generated by the virus itself, are in some cases fuelled by the uncontrolled immune response, also called a "cytokine storm", which can lead to multi-organ failure. This increase in cytokine concentration has been detected in critically ill patients infected with SARS-CoV-2 and has been associated with aggravation of the disease.

Pneumonia and multi-organ failure are the most frequent complications, even leading to death (paper on clinical and epidemiological features).[3]

Other complications include adult respiratory distress syndromes, renal failure, acute lung injury, septic shock, ventilator-associated pneumonia and severe acute respiratory infection.

2.1.8 Case criteria

1. **Close contact**

   A person who:

   (a) Has been exposed to a confirmed or probable case with Covid-19, from 2 days before and up to 10 days after the onset of symptoms of the case.

   (b) Person who has been in contact with an asymptomatic confirmed case, between 2 days before and 10 days after sampling.

2. **Probable case**

   (a) By laboratory result: a person who meets the Suspect Case definition and has an indeterminate or inconclusive PCR result or a person who is asymptomatic or has a non-cardinal symptom and who has a positive result for a rapid antigen detection test for SARS-CoV-2 taken in a licensed health care facility.

   (b) By imaging: a person who meets the definition of a Suspect Case, where the PCR result is negative or indeterminate or inconclusive, but who has a CT scan of the chest with images suggestive of COVID-19.

   (c) By epidemiological link: person who has been in close contact with someone diagnosed with Covid-19 and who has one of the cardinal symptoms or at least two of the remaining signs consistent with Covid-19 within 14 days of contact with the case.

   (d) For death: Deceased person who in the absence of a confirmatory PCR test result, his/her medical death certificate states SARS-CoV-2 infection as the basic cause of death or triggering factor.

   (e) In front of Delta Variant: Person who meets the above definitions and in whom point mutations associated with Delta variant are identified through PCR.

3. **Confirmed case**

(a) Living or deceased person with a positive PCR test for SARS-CoV2.

(b) A person who meets the definition of a suspect case (according to the official definition) and who presents a positive result for a rapid antigen test for SARS-CoV-2, taken in a health centre authorised by the health authority or delegated entity to perform this test. [4]

2.1.9 History of cognitive and neural effects

Possible effects of Covid-19 infection include lung scarring, heart and kidney damage. However, evidence has begun to show the impact it can even have on the brain. In healthy patients with Covid 19, blood clots and strokes have been seen to occur.[1]

The effects after recovery from Covid 19 can persist for months. Some of these include neurological problems such as confusion, disorientation, lack of concentration, headaches, extreme fatigue, mood swings, insomnia and loss of smell and/or taste. [1]

Evidence suggests that the most serious complications appear to stem from the host's own immune response to viral invasion rather than the damage that the virus itself can cause. The excessive and aggressive immune response generated by the immune system causes damage to the organism itself, which attacks itself without distinguishing between the infectious agent and its own cells. In addition, effects caused by fever or organ failure may also lead to brain dysfunction. [1]

Some theories show that the virus is capable of attacking the brain or spinal cord as genetic material has been found in cerebrospinal fluid and there is even a case from Florida where viral particles were found in brain cells. [1]

In previous epidemics it has been observed that the coronavirus is able to penetrate the brain and cerebrospinal fluid, permeabilising the CNS in less than a week. Autopsies of patients suffering from severe acute respiratory syndrome (SARS-CoV) after the 2003 epidemic showed that sequences of the SARS-CoV genome were present in the cortex of the hypothalamus. [7]

The hippocampus appears to be vulnerable to coronavirus infections. This has a direct impact on memory, which deteriorates after infection, accelerating and aggravating neurodegenerative disorders such as Alzheimer's and Parkinson's disease. [7]

Two main mechanisms of CNS invasion have been proposed:

1. Through the blood-brain barrier, which is the first line of defence against viral infection. It is composed in part of brain microvascular endothelial cells between which there are tight junctions that control the permeability of the barrier and which would be compromised in coronavirus infection.

2. Direct infection of the virus in peripheral neurons or olfactory sensory neurons, by which the virus uses axonal transport to gain access to the CNS. [7]

It should be considered that it is possible that following Covid-19 infection and damage to other organs in the body, pre-existing psychiatric disorders and cognitive difficulties may worsen. The magnitude of this effect at the population level is unknown, as are the protective factors that operate in cases with no cognitive changes after infection.[7]

On the other hand, acute respiratory distress syndrome generates hypoxia, which contributes to neuropsychological changes that are associated with brain atrophy and ventricular enlargement. In turn, the duration of hypoxia is related to attention, verbal memory and executive functioning scores at discharge. This syndrome is capable of generating inflammatory responses, anaemia and ischaemia leading to cardiovascular and hepatic failure. This inflammatory cascade generates a series of neurological and physiological events that further aggravate neurological injury in the acute phase, increasing chronic cognitive dysfunction. [7]

Further studies are needed to understand how Covid-19 affects the nervous system. There is little information on the incidence, duration, underlying neural basis, risk factors and protective factors involved in the short- and long-term consequences following COVID-19 infection, which is a challenge for the scientific community. [1]

2.2 Behavioural Measurements

As a result of damage to the various systems of the organism, it has been observed that there are neuronal and cognitive consequences in COVID-19 patients. Thus, the need has arisen to obtain information about them objectively and systematically through behavioural markers.

The behavioural markers are speed and accuracy of response to visual and auditory stimuli, the "temporal junction window" of sensory integration, sensory adaptive responses and retention of verbal information.

Below is the detail of each of the behavioural tests that were implemented in the application, which will help to obtain the necessary markers.

2.2.1 Temporal Binding Window

The task consists of an audio-visual temporal order judgement (TOJ) task in which participants were presented with a visual flash and an auditory beep, and then asked to judge which of the two was presented first. The experiment had a within-participant factor of stimulus onset asynchrony (SOA) between the flash and the beep. The SOA had 12 levels (-350, -216, -133, -88, -50, -50, -16, +16, +16, +50, +88, +133, +216 and +350 ms) where negative SOAs indicate that the beep was presented first and positive SOAs indicate that the flash was presented first. [8]

This task seeks to assess multisensory integrity. A patient who has had COVID-19 is expected to perform poorly in the long term.

2.2.2 Memory task

This experiment has 2 tasks. For both, words were selected from the Medical Research Council's psycholinguistic database with a mean concreteness score of 589.50 and a mean number of 4.89 letters in each word (min=3, max=8).

The first task called word encoding task, in which a series of words are presented in which the participant has to answer yes/no to indicate whether the item was alive, or to indicate whether the item was man-made. In total 200 words are shown divided in 4 blocks, where 2 of them correspond to the decision if the word represented something made by man and the other two, if the word represented a living element.

The second task called word recognition task. The 200 words previously shown are mixed with 100 new words (decoys), presented in 6 blocks of 50 words each. For the item recognition trial, participants are asked to respond on a 5-point confidence scale, with 5 indicating they were sure it was old, 4 indicating it was probably old, 3 indicating they could only guess whether it was old/new, 2 indicating it was probably new and 1 indicating they were sure it was new. The purpose of this task is to examine implicit memory ERPs on an existing dataset of item recognition confidence responses given by patients with Covid 19. [9]

2.2.3 Sensory Adaptation Task

This task consists of judging the loudness of a beep as a percentage of an initial sound. For this, a sound of 70dB SPL was presented continuously for 190 seconds, where the participant had to indicate the percentage of loudness perceived with respect to the beginning of the task. These judgements had to be made every 20 seconds (i.e. 9 judgements in total). A lower sensory adaptation is expected in Covid patients in the long term. [10]

2.2.4 Odd Ball experiment

In this task, a series of unisensory auditory, unisensory visual and multisensory (audio-visual) stimuli are presented in random order to the left and right hemispheres, while the participant

attends to different paths, to one or the other of these sides. The participant must detect the occasional target stimuli, selecting whether they occurred on the right or left side of the screen.

The duration of all stimuli was 105 ms, and trials were presented with an inter-stimulus interval (ISI) that varied randomly between 350 and 650 ms (mean 500 ms). Spatial attention and multisensory awareness are expected to be impaired in Covid-19 patients.[11]

Chapter 3

Project description

3.1 Introduction

The aim of this project is to create a tool to determine whether different markers of cognitive functioning and neuronal integrity are affected by Covid-19 and whether these effects are related to the experience and severity of persistent symptoms. Specifically, to create a tool where people over the age of 18 can perform a set series of exercises, generating data that can be interpreted by doctors and researchers. The people who participate can come from anywhere in the world, which will provide global information and not segmented to a specific country.

The project aims to centralise and speed up the process of obtaining information, taking advantage of the use of technology, without losing the quality of the data. In this way, it will be possible to accelerate ongoing research into the effects caused by COVID-19 .

3.1.1 Purpose of the System

The system consists of a cloud platform to which a registered user will be able to perform 7 different tasks set in a specific order. The participants of the normative sample will be tested only once to establish optimal baselines for each of the tasks, and lab assistants and researchers will be able to download information from each task to draw conclusions about relevant data and make decisions.

The experimental tasks are based on proven tasks commonly used in perceptual and cognitive research (e.g., temporal order judgement or bug detection) to investigate unisensory and multisensory processing.

3.1.2 Scope of the project

The web platform must be compatible with modern versions of different web browsers (Chrome, Firefox and Edge). It must include the necessary functionalities for the participants, the administrator and lab assistants and researchers. Among the most important functionalities are that participants can perform tasks only once. The administrator can manage the users and the lab assistant can access the results. Finally, the researcher can perform the task of lab assistant and administrator.

It is necessary that the applications can continue to function in areas with poor internet connection, as in the future it is expected that the test will be conducted from anywhere in the world and in more remote areas.

3.2 Specific requirements

3.2.1 Functional requirements

The functional requirements of a system are those that describe any activity that the system must perform, in other words, the particular behaviour or function of a system or software when certain conditions are satisfied.

The various functions offered by the web application are presented below:

- **FR1** The system must allow participants to log in.
- **FR2** The system must allow participants to log in and log out.
- **FR3** The system must allow participants to accept the consent form.
- **FR4** The system must send an email to the participant to validate his/her account
- **FR5** The system must allow the participant to enter their personal details to their account
- **FR6** The system must recognise the metadata of the device used by the participant and save it.
- **FR7** The system must allow the participant to initiate and perform a variety of tasks and questionnaires.
- **FR8** The system must allow the participant to perform a demo of the TBW "beep and flash" experiment and record the results
- **FR9** The system must allow the participant to perform the TBW "beep and flash" experiment and record the results.
- **FR10** The system must allow the participant to perform the SPQ questionnaire and record the results.
- **FR11** The system must allow the participant to perform the Covid-related questionnaire and record the results
- **FR12** The system must allow the participant to perform the geographic questionnaire and record the results
- **FR13** The system must allow the participant to perform a demo of the words encoding task (first part of the memory experiment)
- **FR14** The system shall allow the participant to perform the words encoding task and record the results.
- **FR15** The system shall allow the participant to perform a demo of the words recognition task (second part of the memory experiment).
- **FR16** The system shall allow the participant to perform the words recognition task and record the results.
- **FR17** The system shall allow the participant to perform the oddball detection task (attention task) and record the results.

- **FR18** The system shall allow the participant to perform the Sensory adaptation task and record the results.
- **FR19** The system shall present the participant with an introduction screen of the investigation.
- **FR20** The system must allow the lab assistant to log in and log off
- **FR21** The system must allow the lab assistant to access the results of the experiments and questionnaires.
- **FR22** The system must allow to register administrators
- **FR23** The system must allow the administrator to create different types of users.
- **FR24** The system must allow the researcher to log in and log out.
- **FR25** The system must allow the researcher to create different types of users.
- **FR26** The system must allow the researcher to access the results of the experiments and questionnaires.
- **FR27** The system must allow to change the language of the application.

3.2.2 Non-Functional requirements

Non-functional requirements represent general characteristics and constraints of the application or system being developed. Each of them is presented below, separated by the specific feature it targets:

1. **Performance**

   First we start with performance, which refers to the effectiveness of the site and the speed of the response time of any action taken. In the following two tables we see the two requirements related to this point:

Attribute	**Description**
Identifier	RNF-001
Name	Throughput
Description	Describes the number of requests the system manages to process in a time window.
Metric Target	30,000 requests/minute
Priority	Medium

Attribute	Description
Identifier	RNF-002
Name	Front-end response time
Description	Describes the time it takes to perform an action that does not involve a connection to the back-end.
Metric Target	1 second
Priority	Medium

2. **Resource management**

   Resource management refers to the requirements associated with the memory demanded by the application. The associated requirements are presented in the following table:

Attribute	Description
Identifier	RNF-003
Name	Memory limits
Description	Describes the maximum amount of memory that should be used by the web application on the client side.
Metric Target	2GB of RAM
Priority	High

3. **Efficiency**

   La eficiencia se refiere a la capacidad que posee el software de ser accequible para las personas que lo utilizan. El requerimiento asociado a esta caracteristica es:

Attribute	Description
Identifier	RNF-004
Name	Connection response time
Description	Describes the timeout time when performing an action involving connection to the back end independent of load
Metric Target	5 seconds
Priority	Medium

4. **Security**

   Security seeks to guarantee the protection of data, and to achieve this objective, 2 requirements were established, which are as follows:

Attribute	**Description**
Identifier	RNF-005
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

Attribute	**Description**
Identifier	RNF-006
Name	Secure authentication token
Description	The token used to authenticate users must be generated with a secure encryption algorithm.
Metric Target	None
Priority	High

5. **Scalability**

   Scalability refers to the ability of the application to adapt and be able to respond satisfactorily as the number of users increases significantly. For the application to be scalable, it must meet the following requirements:

Attribute	**Description**
Identifier	RNF-007
Name	Auto scaling
Description	The server must scale dynamically as client-side requests increase.
Metric Target	Configure to support 1000 requests/minute and scale up to 30,000 requests per minute.
Priority	High

6. **Fault Tolerance**

   This characteristic means that in the event of a component failure, the system is capable of continuing to function. To achieve this, it must comply with the following table:

Attribute	**Description**
Identifier	RNF-008
Name	Internet Connection Interruption Tolerance
Description	Momentary interruption of internet connection should not affect the usability of the application to any major extent in local interactions.
Metric Target	None
Priority	Medium

7. **Usability**

   This refers to the user-friendliness of the application. Therefore, the following tables must be complied with:

Attribute	**Description**
Identifier	RNF-009
Name	Participant-friendly interface
Description	All interface and activities that can be performed by the participant in your application should be intuitive and easy to use.
Metric Target	None
Priority	High

Attribute	**Description**
Identifier	RNF-010
Name	User-friendly interface for the administrator, researcher and lab assistant
Description	All the interface and activities that the administrative user can perform in your application should be intuitive and easy to use..
Metric Target	None
Priority	Medium

8. **Maintainability**

   This represents the ease faced when correcting, or implementing updates to the application, after moving it to production. For this, it must be fulfilled that:

Attribute	**Description**
Identifier	RNF-011
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

Attribute	**Description**
Identifier	RNF-012
Name	Versions support
Description	The system must not stop working when upgrading the system (making changes).
Metric Target	None
Priority	High

9. **Portability**

   This represents the ability to be able to run on different platforms and to be adapted to each region. This is addressed in the following requirements:

Attribute	**Description**
Identifier	RNF-013
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

Attribute	Description
Identifier	RNF-014
Name	Multi-platform
Description	The system must work independently of the operating system and on Chrome, Firefox, MS Edge and Safari browsers.
Metric Target	None
Priority	High

10. **Interoperability**

    This is the ability of the different components to exchange information and to be able to use or modify it. This is represented in:

Attribute	Description
Identifier	RNF-015
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

11. **Reliability**

    This refers to the accuracy with which the software delivers the expected services. This is represented by the following requirement:

Attribute	Description
Identifier	RNF-016
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

Chapter 4

Analysis

4.1 Introduction

To analyse the web application, we chose to follow the suggestions defined by UML (Unified Modelling Language). This model has a wide range of different diagrams that facilitate the understanding of the system, allowing us to represent in a recognised language, the functionalities, requirements and characteristics of the system. [12]

In our case, we have focused on two types of diagrams: the class diagram and the use case diagram.

4.2 Class diagram

The class diagram represents the structure of the system, showing how the different elements that interact in the application are related. Three types of relationships are presented [12]:

- Composition: high degree of dependency between the containing class and its constituent classes.
- Aggregation: low degree of dependency.
- Association: less dependent on relationship and time.

The following are the classes involved in the application:

- **ParticipantId:** Represents the participant user which is stored in a separate database for security purposes.
- **Participant:** Corresponds to the general information of the participant.
- **Session:** Corresponds to the information of a session, and is used to control the active time of a session and store all the metadata.
- **TBW Trial:** Represents a flash and beep with all the information associated to it and the answer given by the participant.
- **TBW Experiment:** It has the information of the complete test, with its definition.
- **SPQ Trial:** It has the information of the answer given by a participant to a question.

- **SPQ Question Bank:** The question bank of the SPQ experiment.
- **Covid Trial:** It has the information of the answer given by a participant to a question.
- **Covid Question Bank:** It contains the question bank from the Covid Questionare
- **Oddball detection trial:** It has the information of the answer given by a participant in the Oddball experiment.
- **Oddball detection experiment:** It has the information of the complete test, with its definition.
- **Sensory adaptation trial:** It has the information of the response given by a participant in the Sensory adaptation experiment.
- **Sensory adaptation experiment:** It has the information of the complete test, with its definition.
- **Word encoding trial:** It has the information of the answer given by a participant in the Word encoding experiment.
- **Word recognition trial:** It has the information of the answer given by a participant in the Word recognition experiment.
- **Word bank:** Holds all the words to be used in both the Word encoding experiment and the word recognition experiment.
- **memoryTaskExperiment:** It has all the information of the memory experiment and its definition. It includes the 2 tests, Word encoding experiment and word recognition experiment.
- **UniUser:** It has the information of the user with special permissions.
- **Role:** Defines the role of a user with special permissions. This can be (researcher, lab assistant or admin assistant)
- **Permission:** Determines the permissions of a specific role. These permissions give the user the ability to create other administrator users or download the data of the different experiments.

4.3 Use case diagram

Use case diagrams allow us to identify the different actors that participate in the platform, how they interact with each other and the limitations that each one has with the system. In general, these diagrams are self-explanatory for both the client and the user. The use cases present the functional requirements of the platform and are the basis for incremental development.

Within these, 3 elements are presented:

- Actors: These represent a role, which can be performed by someone or something that needs to communicate with the system and exchange information. Therefore, actors can be from one person or several, as well as a software component.
- Use Cases: This is the technical description of the functionality, which is represented in concrete form. By gathering all the use cases, we obtain the functional requirements of the whole application.

- Communication between actors and use cases: Each of the actors can execute a different number of use cases, so we say that there is a communication between them.

The use cases of the project are presented below:

Identifier	**UC1**
Name	Participant registers with the app
Functional Requirement	FR1 The app will handle participant's registration
Initiator	Participant
Goal	Participant will register the app
PreCondition	Participant was presented with the Welcome page (Fig. UC3.1), Information page (Fig. UC3.2), the Consent Form (Fig. UC3.3) and Invitation to Register page (Fig. UC3.4). The app remembers the date consent was given and the preferred language.
PostCondition	The app creates a pseudoname for the participant and records the pseudoname together with participant's email address and a password into a secure stand-alone database (which is separate from the database for the experimental data).
Assumption	Participant does not use a mobile device
Main success scenario	1. Participants clicks on 'I want to register' button 2. The app presents the Registration screen with one text box: 'Enter Email Address' 3. Participant enters his/hers email, password, name and clicks on 'Confirm'. 4. The participant must fill in the consent form by clicking on the 4 checkboxes and clicks on Complete Registration. 5. The app records the email address, pseudoname, password and the date consent was given into a secure database. 6. The app informs the participant that an email was sent to the email address and asks participant to follow the link provided in the email to confirm the account. 7. Participant checks his/hers emails and follows the link. 8. The app updates the user with the date of confirmation. 9. The app informs user about successful registration

Extensions	3.a.1 Participant's email is already in the database. The app informs participant about the existing registration record and offers the participant to login 5.a.1 Participant does not receive an email with the link. Participant clicks on button 'Resent' the link. The app sends another link.

Identifier	**UC2A**
Name	The app handles participant's logging in
Functional Requirement	FR2 The app will handle participant's logging in and logging off
Initiator	Participant
Goal	Participant logs into the app
PreCondition	None
PostCondition	The app records the session and the participant lands on the correct page to start or to resume the study.
Assumption	Participant is not logged in. Participant does not use a mobile device.
Main success scenario	1. Participant clicks on 'Log on' button 2. The app displays 'Log on' screen with two text boxes: 1. Email and 2. Password 3. Participant enters his/hers email and password. 4. The app identifies the participant. 5. The app checks whether the participant's demographic data is already stored (if not, do UC4). 6. The app records the sessions metadata (do UC5) 7. The app performs the Screen Resolution exercise (do UC6) 8. The app identifies the experiment and the trial which participant left in his/hers last session. 9. The app takes participant to the next page as identified in point 8.

Extensions	4.a.1 The app cannot find the participant's email address. The app informs the participant and keeps displaying the 'Log on', which also provides a link to the 'Information on Study' page (before registration participant needs to read Info on Study and to give Consent). 4.b.1 Password provided by the participant does not match the password stored in the app. The app informs participant and keeps displaying the 'Log on' screen, which also provides a link to the 'Forgot password' page. 4.b.2 If participant clicks on the 'Forgot password' then the app asks the participant to enter his/hers email again and an email will be sent to the participant with a link to a page where participant can enter 1) new password 2) confirm password.

Identifier	**UC2B**
Name	The app handles participant's logging off
Functional Requirement	FR2 The app will handle participant's logging in and logging off
Initiator	Participant
Goal	Participant logs off the app and the app remembers the participant's experiment and the trial number.
PreCondition	Participant is logged in
PostCondition	Participant logs off the app and the app remembers the participant's experiment and the trial number.
Assumption	Participant is logged in
Main success scenario	1. Participant click on 'Log off' button 2. The app confirms with participant if he/she wants to log off and warns the participant that there are further experiments to be completed. 3. The participant confirms that he/she wants to log off. 4. The app remembers the ending of session time, the latest experiment and the trial number. 5. The app notifies the participant when the next session should take place (i.e. soon). 6. The app logs the participant off.
Extensions	3.a.1 Participant informs that he/she does not want to log off. The flow of the study continues.

Identifier	UC3
Name	The app presents the participant with Welcome page, Info on Study.
Functional Requirement	FR3 The app will present the participant with Info on Study
Initiator	The app
Goal	The app presents the participant with relevant information.
PreCondition	None
PostCondition	Participant read the Info gave consent for his/hers participation in the study. Participants lands on the 'Registration' page.
Assumption	Participant is not logged in. Participant does not use a mobile device.
Main success scenario	1. The app presents the Welcome page to the participant. 2. The app presents the Information page (Fig UC3.2) to the participant. 3. The app suggest to the participant to register and presents the 'Registration' screen.
Extensions	None

Identifier	UC4
Name	The app asks participant to enter their personal details and records them
Functional Requirement	FR4 The app will ask participant to enter their personal details and will record them.
Initiator	The app
Goal	Participant's personal details get recorded in the app
PreCondition	Participant is registered. This action is performed as a part of the Login workflow.
PostCondition	Participant's personal details recorded.
Assumption	Participant does not use a mobile device.

Main success scenario	1. Participant is presented with the demographic questions screen 2. Participant chooses one of the options for the age question and one of the options on gender question 3. Participant confirms his/hers choices by pressing 'Confirm' button. 4. The app records the age and gender of the participant.
Extensions	3.a.1 Participant did not make the age or gender choice. The app will inform the participant.

Identifier	**UC5**
Name	The app detects and records the device's metadata
Functional Requirement	FR5 The app will detect participant's device's metadata and will records them
Initiator	The app
Goal	Device's details for participant's session get recorded in the app
PreCondition	This action is performed as a part of Login workflow
PostCondition	Device's details for participant's session recorded in the app.
Assumption	None
Main success scenario	1. The app gets the details from the participant's browser 2. The details are recorded in the app. 3. Timestamp of the start of the session is recorded in the app.
Extensions	2.a.1 The details indicate that participant uses a mobile device. The app informs the participant "Thank you for considering to participate in this study. It seems however that you are trying to access this experiment from a mobile device. Note however that mobile devices are not suitable for this experiment as their screens are too small and don't have a keyboard or mouse. Please use a regular computer to participate in this experiment"

Identifier	**UC6**

Name	Demo for Temporal Binding Window experiment
Functional Requirement	The app will allow participant to undertake a demo of the TBW "beep and flash" experiment
Initiator	The app
Goal	Participants will accomplish one session of N number of trials (instances) of identifying what appears first: a beep or a flash.
PreCondition	The app has a record of participant's screen resolution and PPI.
PostCondition	Participant completed the demo, results of the trials were recorded and the participant was informed about results.
Assumption	Participant has given consent and does not use a mobile device.

Main success scenario	1. The app determines the number of trails N (from TBW demo experiment info stored in the app. The number of trials N is calculated by multiplying the number of SOA elements by the number of repeats for each SOA). 2. The app determines the demo "trial SOA range" (The "trial SOA range" will have N number of SOA elements; if number of repeats is 2, then each SOA element will be included 2 times in the "trial SOA range".) 3. The app determines whether the experiment has a break during the trials (from TBW demo experiment info stored in the app). 4. Participant is presented with the description of the experiment. 5. Participant clicks on the 'Continue' button. 6. Participant is presented with information on Practice sessio 7. Participant clicks on the 'Continue' button. 8. The participant is presented with a white background screen with (full screen) word 'Practice' and button 'Start' 9. The participant is presented with a white background screen with word 'Practice', button 'Close Fullscreen', 'Left Arrow – Sound First; Right Arrow – Light First', 'Round 1'. 10. The flash and a beep appear and with trial SOA range as per point 2. 11. Participant presses left arrow key on the keyboard if beep first or right arrow key if flash first. 12. The counter of incorrect or correct answers is updated according to the answer obtained and is displayed in the upper corner. 13. The app records for each trial: intendedSOA, flash timestamp, beep timestamp, response timestamp, actual SOA and responded flashfirst (true of false). 14. The app displays the next trial number out of N. Steps 9 – 13 are repeated. 15. After completing all N trials, participant is presented with "results" screen which displays a proportion number of correct trials (Fig UC8.5). The participant can press on 'Continue' button to continue the live TBW experiment.

Extensions	9.a.1 The participant escapes from the full screen. Message appears "If you want to continue, make the full screen again". The experiment pauses until the participant makes it full screen again, so the experiment resumes from there the participant stopped. 9.b.1 The app allows the participant to log off (if participant wants to) and remembers the position where the participant was at the time. When the participant is logged in next time the app will resume from where participant stopped.

Identifier	**UC7**
Name	Live Temporal Binding Window experiment
Functional Requirement	The app will allow participant to undertake the TBW "beep and flash" experiment
Initiator	The app
Goal	Participants will accomplish one session of N number of trials (instances) of identifying what appears first: a beep or a flash.
PreCondition	The app has a record of participant's screen resolution and PPI. Participant already performed the demo TBW experiment and clicked on 'Continue' button from the Demo results screen.
PostCondition	Participant completed the TBW experiment, results of the trials were recorded.
Assumption	Participant has given consent and does not use a mobile device.

Main success scenario	1. The app determines the number of trails N (from TBW live experiment info stored in the app. The number of trials N is calculated by multiplying the number of SOA elements by the number of repeats for each SOA). 2. The app determines the SOA range (from TBW live experiment info stored in the app). 3. The app determines whether the experiment has a break during the trials (from TBW live experiment info stored in the app). 4. Participant is presented with information on the session. 5. Participant clicks on the 'Continue' button. 6. The participant is presented with a white background screen with (full screen) and button 'Start' 7. The participant is presented with a white background screen, button 'Close Fullscreen', 'Left Arrow – Sound First; Right Arrow – Light First', 'Round 1' 8. The flash and a beep appear and with trial SOA range as per point 2. 9. Participant press the left arrow key on the keyboard if beep first or right arrow key if flash first. 10. The trial number displayed on the screen is updated. 11. The app records for each trial: intendedSOA, flash timestamp, beep timestamp, response timestamp, actual SOA and responded flashfirst (true or false). 12. The app displays the next trial number out of N. Steps 7 – 11 are repeated until N/2 trials have been completed. 13. If the experiment has a break (as per point 3), the app presents the 'break' page , otherwise go to point 15. 14. Participant press 'Start' on the 'break' page. 15. The app displays the next trial number out of N. Steps 7 – 11 are repeated until N trials have been completed. 16. After completing all N trials, participant is presented with "end of TBW experiment" screen and is redirected to the next experiment.

Extensions	7.a.1 The participant escapes from the full screen. Message appears "If you want to continue make the full screen again". The experiment pauses until the participant makes it full screen again, so the experiment resumes from there the participant stopped. 7.b.1 Nice to have: The app allows the participant to log off (if participant wants to) and remembers the position where the participant was at the time. When participant is logged in next time the app will resume from where participant stopped.

Identifier	**UC8**
Name	Word encoding task
Functional Requirement	FR19 The app will allow participant to undertake the words encoding task (first part of the memory task)
Initiator	The app
Goal	Participant will accomplish a task of encoding of a number of words in 4 blocks of practice and live sessions (4 blocks x N words in live session; each live block is preceded by a practice session with K number of words)
PreCondition	None
PostCondition	Participant completed the task of encoding 4xN live words and the app records the results of encoding (each word, response time and the decision "alive" ['yes' or 'no'] or "manmade" ['yes' or 'no'])
Assumption	Participant has given consent and does not use a mobile device

Main success scenario	1. The app retrieves number K (number of words in each practice block). 2. The app retrieves number N (number of words in each live block) 3. The app randomly decides which block to start with ('ALIVE' or 'MANMADE'). Practice ALIVE block: 4. If the first block is 'ALIVE' then the app presents to the participant the Encoding Task Instructions Practice page for the 'ALIVE' block 5. Participant clicks on 'Continue' button. 6. The participant is presented with a cross in the middle of the screen. The Participant is presented with a practice word (in uppercase letters in a white font, size 36, centred on a black background screen). The word is taken randomly from one of the practice words which are stored in a bank of words within the app. 7. Participant makes a choice (whether the word represents a living thing or not) and presses either 'S' (for 'yes') or 'L' (for 'no'). 8. The app records the presented practice word, reaction timestamp and the participant's choice (thus building the experiment's dataset). 9. The participant is presented with a cross in the middle of the screen. The participant is presented with another practice word from the bank of words (words which were already presented to the participant are not shown again). The points 7., 8. and 9. continue until all K practice words were presented to the participant. Live ALIVE block: 10. The participant is presented with the Live Encoding Task instructions for the 'ALIVE' block. 11. Participant clicks on 'Continue' button. 12. The participant is presented with a cross in the middle of the screen. The app presents to the participant a live word from the bank of words (the words previously presented to the participant are not shown again). 13. Participant makes a choice (whether the word represents a living thing or not) and presses either 'S' (for 'yes') or 'L' (for 'no').

	14. The app records the presented word, reaction timestamp and the participant's choice (thus building the experiment's dataset). 15. The points 12, 13, 14 continue until all N live words were presented to the participant. Practice MANMADE block: 16. Participant is presented with Encoding Task Instructions Practice page for the 'MANMADE' block 17. Participant clicks on 'Continue' button. 18. The participant is presented with a cross in the middle of the screen. The app presents the participant with a practice word from the bank of words (the words previously presented to the participant are not shown again). 19. Participant makes a choice (whether the word represents a manmade thing or not) and presses either 'S' (for 'yes') or 'L' (for 'no'). 20. The app records the presented practice word, reaction timestamp and the participant's choice (this building the experiment's dataset). 21. The participant is presented with a cross in the middle of the screen. The participant is presented with another practice word from the bank of words (the words already presented to the participant are not shown again). The points 18, 19. and 20. continue until all K practice words were presented to the participant. Live MANMADE block: 22. The participant is presented with the Encoding Task instructions for the 'MANMADE' block 23. Participant clicks on 'Continue' button. 24. The participant is presented with a cross in the middle of the screen. The app presents with a live word from the bank of words (the words previously presented to the participant are not shown again). 25. Participant makes a choice (whether the word represents a manmade thing or not) and presses either 'S' (for 'yes') or 'L' (for 'no'). 26. The app records the presented word, reaction timestamp, and the participant's choice (this building the experimental dataset). 27. The points 24., 25., 26. continue until all N live block words were presented to the participant. 28. Participant is again presented with Encoding Task Instructions Practice for the 'MANMADE' block

	29. Points 19 to 27 are repeated. 30. Participant is presented with Encoding Task Instructions Practice page for the 'ALIVE' block 31. Points 5 to 15 are repeated.
Extensions	4.a.1 If the first block is 'MANMADE' then the app presents the participant with the Encoding Task Instructions Practice for the 'MANMADE' block. The process is carried out in similar manner as in the Main Scenario but according to the overall flow

Identifier	**UC9**
Name	Word recognition task
Functional Requirement	The app will allow participant to undertake the words recognition task (second part of the memory task)
Initiator	The app
Goal	Participant will accomplish a task of recognising 6xK practice words and 6xN words (6 blocks x N words). Participant will make a decision for each word: whether the word was old (i.e. previously seen in the encoding task) or new. The decision is expressed on the scale from 1 (new) to 5 (old).
PreCondition	Participant completed the task of encoding Kx4 practice and Nx4 live words and the app has a record of the results of encoding.
PostCondition	Participant completed the task of recognising words and the app records the results of the recognition (each word, grading number from 1 to 5 and response time)
Assumption	Participant has given consent and does not use a mobile device

Main success scenario	1. The app retrieves 4xK words encoded by the participant during the practice encoding task and randomly selects additional 2xK words from practice words which are stored in a bank of words within the app. 2. The app retrieves 4xN words encoded by the participant during the live encoding task and randomly selects additional 2xN words from live words which are stored in a bank of words within the app. These 6xN words are randomly divided into 6 blocks of N words each. Practice Block: 3. The app presents to the participant the Practice Recognition Task Instructions page. 4. Participant presses 'Continue' button. 5. The participant is presented with a cross in the middle of the screen. The app presents the participant with a word from a practice block 6. Participant presses on one of these keys: 1, 2, 3, 4 or 5. 7. The app records the presented word, reaction timestamp and the participant's choice (this building the experiment's dataset). 8. Participant is presented with results (number of correctly recognised words from 6xK words) Live Block: 9. The app presents to the participant the Practice Recognition Task Instructions page 10. The participant is presented with a cross in the middle of the screen. The app presents to the participant a word from the live block. 11. Participant presses on one of these keys: 1, 2, 3, 4 or 5 12. The app records the presented word, reaction time stamp and the participant's choice(this building the experiment's dataset). 13. Points 9 – 12 continue until all N words from the block were presented. 14. Break (a page with counter) 15. Same as points 9 to 14 for the next block until all words were presented. In total there are 6 live blocks.
Extensions	None

Identifier	**UC10**
Name	SPQ questionnaire
Functional Requirement	The app will allow participant to undertake the SPQ questionnaire
Initiator	The App
Goal	Participants will accomplish the SPQ questionnaire.
PreCondition	None
PostCondition	The participant will have completed the spq questionnaire and will be saved in the application.
Assumption	Participant has given consent and does not use a mobile device
Main success scenario	1. The app checks if the test has been taken before. 2. The app retrieves the questions associated with the test 3. The participant is presented with the questionnaire instructions. 4. The participant must select the answer to the questions without leaving any blank. The possible alternatives are Strongly agree, Somewhat agree, Somewhat disagree and Strongly disagree. 5. Points 4 continue until all questions are answered. 6. User presses submit button 7. The application records each answer given by the participant. 8. The application redirects the participant to the next experiment.
Extensions	1.a.1 If the experiment has already been performed, the application will display the message: "You have already taken this test".

Identifier	**UC11**
Name	Covid questionnaire The app will allow participant to undertake the Covid related questionnaire App Participants will accomplish the Covid questionnaire. None The participant will have completed the Covid questionnaire and will be saved in the application. Participant has given consent and does not use a mobile device

Functional Requirement	The app will allow participant to undertake the Covid related questionnaire
Initiator	The App
Goal	Participants will accomplish the Covid questionnaire.
PreCondition	None
PostCondition	The participant will have completed the Covid questionnaire and will be saved in the application.
Assumption	Participant has given consent and does not use a mobile device
Main success scenario	1. The app checks if the test has been taken before. 2. The app retrieves the questions associated with the test 3. The participant is presented with the questionnaire instructions. 4. The participant must select the answer to the questions without leaving any blank. 5. The application depending on the answer will show another question. In case a new question is displayed, we go back to point 4. 6. User presses submit button 7. The application records each answer given by the participant. 8. The application redirects the participant to the next experiment.
Extensions	1.a.1 If the experiment has already been performed, the application will display the message: "You have already taken this test".

Identifier	**UC12**
Name	Lab assistant is registered with the app.
Functional Requirement	The app will handle lab assistant's registration.
Initiator	Researcher
Goal	Lab assistant registers with the app
PreCondition	Researcher has 'register new user' permission
PostCondition	Lab assistant is registered with the app

Assumption	Researcher has permission to setup other users. Lab Assistant role exists in the app with only permission to access experimental data database.
Main success scenario	1. Researcher initiates the registration of a new user by clicking on 'register a new user' button. 2. The app presents the page with text box 'enter new user email' and options to choose a role from one of three (researcher, lab assistant or admin assistant). 3. Researcher enters the new user's email address and chooses 'lab assistant' role. 4. The app checks that there is no another user in the app with this email 5. The app sends the email to the new user with a link to 'complete your registration' page. 6. The user receives the email and completes their registration by entering and confirming their password 7. The app checks the strength of the password, if the password is strong enough, presents the new user with the 'Successful registration' page and Logon option.
Extensions	None

Identifier	**UC13**
Name	The app handles lab assistant's logging in
Functional Requirement	The app will handle lab assistant's logging in and logging off
Initiator	Lab assistant registers with the app
Goal	Lab assistant logs in
PreCondition	None
PostCondition	Lab assistant is logged in and lands on the Uni User portal page with options appropriate to the lab assistant's permissions
Assumption	Lab assistant is not currently logged in.

Main success scenario	1. Lab assistant clicks on 'Admin Log on' button 2. The app displays 'Log on' screen with two text boxes: 1. Email and 2. Password 3. Lab assistant enters his/hers email and password. 4. The app identifies the UniUser and their role (lab assistant). 5. The app checks the lab assistant's permissions: permission to access only the experimental data. 6. The app presents the lab assistant page with an option to access a set of reports from the experimental data database.
Extensions	4.a.1 The app cannot find the lab assistant's email address. The app informs the user and keeps displaying the 'Log on' page. 4.b.1 Password provided by the lab assistant does not match the password stored in the app. The app informs the lab assistant and keeps displaying the 'Log on' screen, which also provides a link to the 'Forgot password' page. 4.b.2 If the lab assistant clicks on the 'Forgot password' then the app asks the lab assistant to enter his/hers email again and an email will be sent with a link to a page where the lab assistant can enter 1) new password 2) confirm password.

Identifier	**UC14**
Name	Admin assistant is registered with the app.
Functional Requirement	The app will handle administrator's registration
Initiator	Researcher
Goal	Administrator registers with the app
PreCondition	Researcher has 'register new user' permission
PostCondition	Administrator is registered with the app
Assumption	Researcher has permission to setup other users. Administrator role exists in the app with only permission to access participant's private data database.

Main success scenario	1. Researcher initiates the registration of a new user by clicking on 'register a new user' button 2. The app presents the page with text box 'enter new user email' and options to choose a role from one of three (researcher, lab assistant or admin assistant) 3. Researcher enters the new user's email address and chooses 'administrator' role. 4. The app checks that The app checks that there is no another user in the app with this email. 5. The app sends the email to the new user with a link to 'complete your registration' page. 6. The user receives the email and completes their registration by entering and confirming their password 7. The app checks the strength of the password, if the password is strong enough, presents the new user with the 'Successful registration' page and Logon option.
Extensions	None

Identifier	**UC15**
Name	The app handles administrator's logging in.
Functional Requirement	The app will handle administrator's logging in and logging off.
Initiator	Administrator
Goal	Administrator logs in
PreCondition	None
PostCondition	Administrator is logged in and lands on the Uni User portal page with options appropriate to administrator's permissions
Assumption	Lab assistant is not currently logged in.

Main success scenario	1. Administrator clicks on 'Admin Log on' button 2. The app displays 'Log on' screen with two text boxes: 1. Email and 2. Password 3. Administrator enters his/hers email and password. 4. The app identifies the UniUser and their role (administrator). 5. The app checks the administrator's permissions: permission to access only personal data of participants. 6. The app presents the administrator page with an option to access a set of reports from participants personal data database
Extensions	4.a.1 The app cannot find the administrator's email address. The app informs the user and keeps displaying the 'Log on' page. 4.b.1 Password provided by the administrator does not match the password stored in the app. The app informs the administrator and keeps displaying the 'Log on' screen, which also provides a link to the 'Forgot password' page. 4.b.2 If administrator clicks on the 'Forgot password' then the app asks administrator to enter his/hers email again and an email will be sent with a link to a page where administrator can enter 1) new password 2) confirm password.

Identifier	**UC16**
Name	Oddball detection task (attention task)
Functional Requirement	The app will allow participant to undertake the oddball detection task (attention task)
Initiator	The application
Goal	Participant completed the task of determine the presence of any stimulus and indicate whether it occurred on the right or left side.
PreCondition	User is logged in
PostCondition	The participant completed the task of determining the presence of visual and auditory stimuli and the application records the answers given.
Assumption	Participant has given consent and does not use a mobile device and is wearing headphones

Main success scenario	1. The app checks if the test has been taken before. 2. The application searches for what has already been done and loads it to start from there. 3. The application sets the different stimuli to be presented. 4. The app presents to the participant the Odd Ball Task Instructions page 5. Participant press the 'Continue' button. 6. The participant is presented with an auditory or visual stimulus to the left or right. In addition, the block that the participant is performing is indicated (there are 10 in total). 7. The participant presses one of these keys: right or left arrow, when the participant perceives a stimulus. 8. The application records the stimulus that was performed, the time stamp of the reaction and the participant's choice (thus building the experimental data set). 9. Points 6 – 8 continue until all 10 blocks were presented. 10. The application redirects the participant to the next experiment.
Extensions	1.a.1 If the experiment has already been performed, the application will display the message: "You have already taken this test".

Identifier	**UC17**
Name	Sensory adaptation task
Functional Requirement	The app will allow participant to undertake the sensory adaptation task
Initiator	The application
Goal	Participant completed the task of pointing out the perceived volume in relation to the initial volume.
PreCondition	User is logged in
PostCondition	The participant completes the task of pointing out the perceived volume in relation to the initial volume and the application records the answers given.
Assumption	Participant has given consent and does not use a mobile device and is wearing headphones

Main success scenario	1. The app checks if the test has been taken before. 2. The app presents to the participant the Sensory adaptation Task Instructions page 3. Participant press the 'Continue' button. 4. The participant is presented with an auditory stimulus for 190 seconds. 5. Every 20 seconds the application will enable buttons to judge the perceived sound. 6. The participant has 10 seconds to press one of these buttons: 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 100%. 7. The application locks the buttons after 10 seconds or when the participant presses it. 8. Points 5 - 7 continue until 190 seconds have elapsed 9. The application redirects the participant to the next experiment.
Extensions	1.a.1 If the experiment has already been performed, the application will display the message: "You have already taken this test".

Figure 4.1: Use cases diagram.

Chapter 5

Design of the solution

5.1 Introduction

The design process of an application is fundamental to be able to meet both functional and non-functional requirements, and to make the work process scalable and flexible. This process consists of reorganising the aspects raised in the conceptual modelling of the project, with the aim of developing a detailed specification of the information system, focused on obtaining the software solution to be realised.

A Three-Tier Architecture was chosen for the software design. This is a type of architecture that is composed of 3 "layer" of computing logic, the Presentation Tier (user interface), Application Tier (the functional business logic) and Data Tier (includes the databases and the access layer to these). This architecture was chosen because it provides 3 fundamental characteristics for this project. [13]

The first is the speed of development, given that it provides the possibility of modularising the layers, being able to modify them without influencing the others, focusing the work team on their core competencies, achieving a backend and frontend team, completely independent at the code level.

Secondly, it provides great scalability. This is because each of the layers can be scaled independently according to the requirements of the application, allowing each layer to be balanced, increasing overall performance. It also gives the possibility for the different layers to be hosted on different servers.

Finally, it increases reliability and availability by hosting different layers of the software on different servers and by caching responses, which is a high priority non-functional requirement of this application.

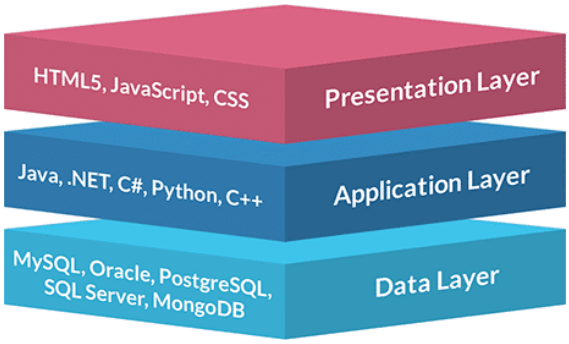

Figure 5.1: 3 tier architecture diagram (source: jinfonet).

5.2 Presentation layer

The presentation layer corresponds to the top level of the architecture and is represented by the front-end. It is the layer that interacts directly with the user, which will allow the user to perform each of the experiments, obtain the data or simply obtain information from the research. [13]

There are 3 main interfaces depending on the user:

1. The first one corresponds to the non-logged in user, where only the information of the study will be shown together with the buttons to log in or register.
2. The second one corresponds to the logged in participant, where the information of the study will also be shown, but in addition there will be a graph, which will represent the progress of the experiments carried out, together with a button to go to the next experiment. Depending on the experiments performed previously, the software automatically determines the next exercise to be performed. When all experiments are finished, the main page is displayed, showing that you have completed all experiments.
3. Finally there will be the page of the user with administrative privileges, where depending on their permissions, buttons will be presented to access the information of experiments already performed, or to add new users with special privileges.

5.3 Application Layer

This corresponds to the intermediate layer, which controls the functionalities of the application by processing the information, based on the requirements. The code was structured in two main directories, one where all the data models were found, and the other where we can find the connections and management of the desired functionalities. In the following, the development of the model associated to the Odd ball experiment will be presented, in order to better reflect how the code was structured (a description will be presented before each of the parts).

As mentioned above, first the model was created, which represents the data structure associated with the experiment.

```
1  class OddBall(db.Model):
2     INDEX = 'index'
3     USER_ID = 'user_id'
4     STIMULUS_ORDER = 'stimulus_order'
5     BLOCK = 'block'
6     SIDE = 'side'
7     STIMULUS = 'stimulus'
8     STIMULUS_TIMESTAMP = 'stimulus_timeStamp'
9     RESPONSE_TIMESTAMP = 'response_timeStamp'
10    RESPONSE_TARGET = 'response_target'
11
12    index = Column(Integer, primary_key=True, autoincrement=True)
13    user_id = Column(String(20), nullable=False)
14    stimulus_order = Column(Integer, nullable=False)
15    block = Column(Integer, nullable=False)
16    side = Column(String, nullable=False)
17    stimulus = Column(String, nullable=False)
18    stimulus_timeStamp = Column(DateTime, default=None)
19    response_timeStamp = Column(DateTime, default=None)
20    response_target = Column(String, default=None)
21
22    __table_name = 'odd_ball'
23    __bind_key__ = 'happyagaindb'
24
25    def __init__(self, user_id, stimulus_order, block, side, stimulus,
          stimulus_timeStamp, response_timeStamp, response_target):
26        self.user_id = user_id
27        self.stimulus_order = stimulus_order
28        self.block = block
29        self.side = side
30        self.stimulus = stimulus
31        self.stimulus_timeStamp = datetime.strptime(stimulus_timeStamp,
              '%Y-%m-%d %H:%M:%S.%f')
32        if (response_timeStamp != None):
33            self.response_timeStamp = datetime.strptime(response_timeStamp,
                  '%Y-%m-%d %H:%M:%S.%f')
34        self.response_target = response_target
```

Figure 5.2: Creation of the odd ball model.

We can see in figure 5.2 that for each attribute the expected data type and the different constraints have been established.

After having created the structure, the management of this entity was developed, and accessible routes were made available from the frontend, in order to process the information correctly.

```
1  @odd_ball_api.route('/odd_ball', methods=['POST'])
2  @authenticate
3  def post_responses(user_id, user):
4     request_json = request.get_json(force=True)
5     data = request_json.get('data')
6     if data:
7        for block in data:
8           for responses in block["responses"]:
9              resTime = None
10             resVal = None
11             if ("value" in responses.keys()):
12                resTime = responses["response_timeStamp"]
13                resVal = responses["value"]
14
15             response = OddBall(user_id, responses['order'], block["block"],
                   responses["side"], responses["stimulus"],
                   responses["stimulus_timeStamp"], resTime, resVal)
16             db.session.add(response)
17             db.session.commit()
18       return Response(200, mimetype='application/json')
19    else:
20       return Response(401, mimetype='application/json')
```

Figure 5.3: Path that allows to make a post of an object of class Odd Ball.

In figure 5.3 we can see the POST route in charge of creating an object of the Odd Ball class, which has the restriction of being generated by authenticated users. We also have 2 other GET routes, the first one that delivers all the responses generated by users and finally, one that responds if a particular user has already performed and completed the experiment.

5.4 Data Layer

This corresponds to the last layer, and involves the different data persistence mechanisms, such as the servers that host the databases. It should be noted that the project established the requirement to have two databases for security reasons and to ensure data integrity. One exclusively for the storage of the users, and the other to store the data of the different experiments and users with privileges.

The first database called userdb has the following table:

- users: stores all the users that have registered.

The second database called happyagaindb has the following tables:

- **bank of words**: stores the words to be used by the word encoding and recognition experiment.
- **words encoding trial**: Stores the answers given by the user of the word encoding experiment.
- **word recognition trial**: Stores the answers given by the user of the word recognition experiment.

- **user roles**: Stores all privileged users, specifying the type of user they are.
- **tbw experiment**: Stores the base parameters for the creation of the temporary binding window experiment.
- **tbw test**: Stores a specific test for a user. As this experiment can be paused and continued in a future session, the complete experiment should have been stored for each user, at the moment of starting it.
- **temporal binding window**: Stores the answers given by the user of the temporal binding window experiment.
- **sessions**: Stores the sessions of each user, saving the metadata of the device used when logging in.
- **sensory adaptation**: Stores the answers given by the user of the sensory adaptation task.
- **question table**: Stores all possible questions. These questions are used for the SPQ, demographic and covid questionnaire.
- **question response**: Stores the answers given by the users. In addition, the type of experiment to which each answer is related is indicated.
- **odd ball**: Stores the answers given by the users of the odd ball experiment.

Chapter 6

Results obtained

6.1 Technologies used

The components of the web application are organised according to the classical web application architecture (Fig. of the architecture) with some particular details discussed below.

At the beginning of the project, the following constraints on the architecture and deployment environment were defined:

- The back-end of the application is developed on a web server using the Flask framework in Python with connection to two SQL databases (private database and experimental database to meet the database separation requirement);
- The back-end will be deployed on a secure Ubuntu server;
- The front-end of the application is developed in typescript with the Angular framework, which uses the Model-View-View-Model (MVVM) pattern. This pattern is characterised by decoupling the application logic with the user interface as much as possible.
- Communication between the front-end client and the back-end server takes place via web services in the form of RESTful APIs, which use .json files over HTTPS to exchange data between the client and the server.

In the following, details about the parts that underpin the system architecture will be presented. Frameworks, libraries and programming languages are listed and briefly explained.

As recently described the web application is composed of the Client (graphical user interface, collecting data from the user, passing the data to the Server) and the Server (serving the application data to the Client, receiving the data from the client and storing it, providing the business logic to the application). APIs are used for communication between the Client and the Server. The team made the following decisions for the implementation of the Client-Server architecture:

- Angular Framework with Typescript for the Client software.
- Flask Framework with Python for the Server software
- SQLite library for the SQL database engine implementation
- SQLAlchemy which is Python's SQL toolkit and Object Relational Mapper
- Web services in the form of the popular architectural style REST API (REpresentational State Transfer Application Program Interface) using JSON (Java Scrip Object Notation) data files.

Angular is a front-end JavaScript framework, which was created and maintained by Google. It is used to build powerful front-end web applications. It can be used with any back-end service and works very well with RESTful APIs. Angular is a very popular framework because it is ideal for rapid front-end development. It creates complex HTML and JavaScript components and other necessary services in a much shorter period of time compared to if these components were written from scratch. It is good for dynamic content programming: instead of static HTML elements there are templates that use variables, expressions and other dynamic programming concepts. Angular is very adaptable to different platforms, it works well on different operating systems and with different web browsers. The Angular framework has good features for unit testing and end-to-end testing. [14]

This project was built with Angular CLI version 8.3.21. Angular CLI is a command line interface tool used to generate, develop and maintain Angular applications. The main features of the framework, utilities and development environment are automatically available through Angular CLI. Angular CLI is installed through NPM, which is Node's package manager for Node.js (JavaScript runtime environment). Details of the Angular CLI installation and Angular executable commands (ng commands) are available at https://github.com/happy-again/angular-cli and https://angular.io/cli.

The TypeScript language is not mandatory for creating an Angular application, but using it has a big time-saving advantage over using regular Java Script. It should be noted that it is a superset of Java Script and was created by Microsoft, which compiles to plain Java Script. [15]

Flask is a small framework that provides a core of basic services for back-end web application development. Other extension packages can be imported into Flask for any other services required. For example, there is no support in Flask for database access and web form validation, so other packages have to be integrated with the basic packages. [16]

The API is a service contract, which defines what should be done between the client and the server and what data should be sent or received. RESTful API is the industry standard architecture of a web API, which uses the standard HTTPS protocol. For example, the Client sends an HTTPS GET request to the Server URL and the Server responds with the data. The body of the response contains JSON (Java Scrip Object Notation) data, which provides a way to structure and nest the data.

Other libraries and extensions used in the application:

- JWT Manager library for storing and retrieving JWT tokens. JSON Web Tokens is an open, industry-standard (RFC 7519) method for securely transmitting data between parties as a JSON object. [17]
- The Flask-Mail extension to configure the SMTP (Simple Mail Transfer Protocol), in order to be able to send emails from the web application. Emails are handled through an instance of Mail.
- The built-in Python UUID library to generate unique identifiers, which are needed in some database tables (e.g. the user table).
- Passlib library that supports a large number of hash algorithms. The application uses the pbkdf2_sha256 library class to hash user passwords.
- Python datetime module for manipulating dates and times

6.2 Description of the project

6.2.1 Code

This section explains the front-end and back-end code: the structure of the files in the context of the realisation of the system architecture and in the context of meeting the project requirements.

The github repository was used for version control and collaboration. Below is the structure of the code, and important files of both the backend and the frontend.

```
from happy_again import create_app
from happy_again import db
from happy_again.common.consts import deployment
application = create_app()
print("app created")
db.create_all()
# Used only for develop environment
if not deployment:
  if __name__ == "__main__":
      application.run( port=1234, debug=True)
```

Figure 6.1: Main.py file.

1. **Backend**

   A github project called happy-again-backend was created. Inside the main directory we find the following important files:

   - **Requirements.txt** define los paquetes (dependencias) que se utilizan en la aplicación Flask. Cada vez que se añada una nueva dependencia al proyecto, deberá añadirse a este archivo. Cualquiera que quiera cargar y utilizar el proyecto en el futuro, puede instalar todas las dependencias necesarias haciendo referencia al archivo requirements.txt. Además ayuda el control de versiones, en caso de que se requiera una en especifico para que la aplicación funcione de manera correcta.

   - **Main.py** is the entry point to the server-side application (Fig 6.1). The application starts running when this file is executed. The deployment variable is intended to specify whether the project should be launched on a development machine or on the deployment server. If deployment is false, then the application will run on port '1234' on the localhost of the development machine.

   - The **.gitignore** file is used to define the extensions of files that we don't want to be pushed from the development environment to github, as they are supplementary files used in IDEs (e.g. in the PyCharm IDE tool).

   - **happy-again-backend/happy_again directory**: In the __init__ file the application is initialised, so happy_again can be treated as a package in the main.py file. We notice that when initialising the application, 2 important functions are called, the first one corresponds to create_app(), where an instance of the application is created (an object of the flask class). In addition, mail server configurations are specified, with which the mail object is created, which will allow mail to be sent. At runtime, the bindings 'sqlite:///userdb.db' and 'sqlite:///happyagaindb.db' will allow the automatic creation of database files on the local file system. Here we also initialise the JWT token library object, import the API packages created in the happy-again-backend/happy_again/apis directory and register the APIs in the app.

   - **happy-again-backend/happy_again/models**: In the __init__ file we import the models that were defined inside the happy-again-backend/happy_again/models directory. Here you will find the representation of the classes, which were described in the structural analysis diagram.

   - **happy-again-backend/happy_again/apis**: Here are written all the routes accessible from the frontend, which are presented in the following table.

2. **Frontend**

The Angular framework is based on the Model-View-View-Model (MVVM) pattern, which is schematically represented in Fig. 6.2.

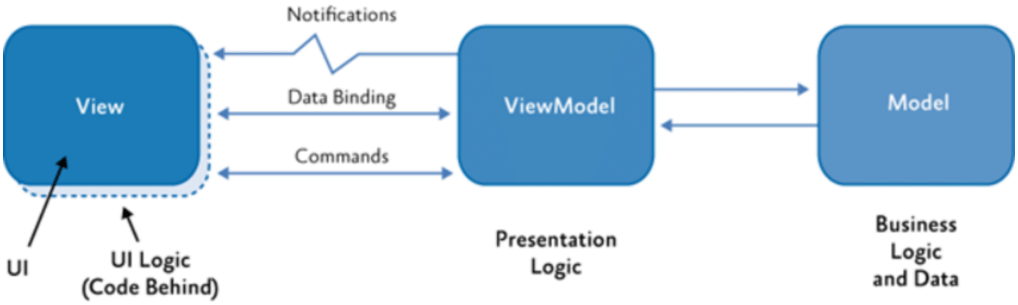

Figure 6.2: Model-View-View-Model (MVVM) pattern (source: researchgate).

According to the MVVM pattern, in the front-end of the application there are three main types of files responsible for the following: [18]

- View: Static visual representation (.html and .css files of a component structure).
- View model: Logic of the visual representation (.ts file of a component structure)
- Model: Client-side business logic of the application (.ts files of a service.ts)

Thus, the front-end work on the application consisted of the following main activities:

- Creation of components for each required feature (visual and dynamic elements of a feature).
- Creation of services to provide data to the components, for the implementation of the application logic and for communication with the back-end server.

Each newly created component in Angular consists of three parts:

- Template, which represents the view (HTML and CSS). It is the visual element.
- Class, which supports the view (Typescript). It contains properties and methods, which control the logic of the view.
- Metadata, i.e. the information that informs Angular if a particular Class is a component (defined by a decorator).

For each new component, Angular CLI creates at least three files, which are modified to achieve the visual behaviour and dynamic behaviour. For example, the 'home' component is represented by three files (.html, .css and .ts) as shown in Fig. 6.4.

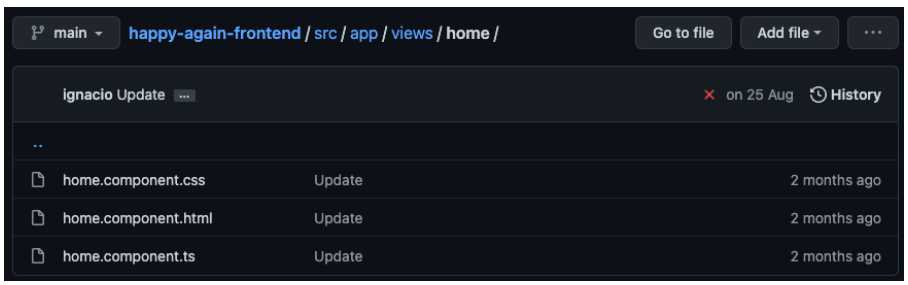

Figure 6.3: Home directory.

All components created for this project are located in the happy-again-frontend/src/app/views directory and are shown in Fig. 6.5.

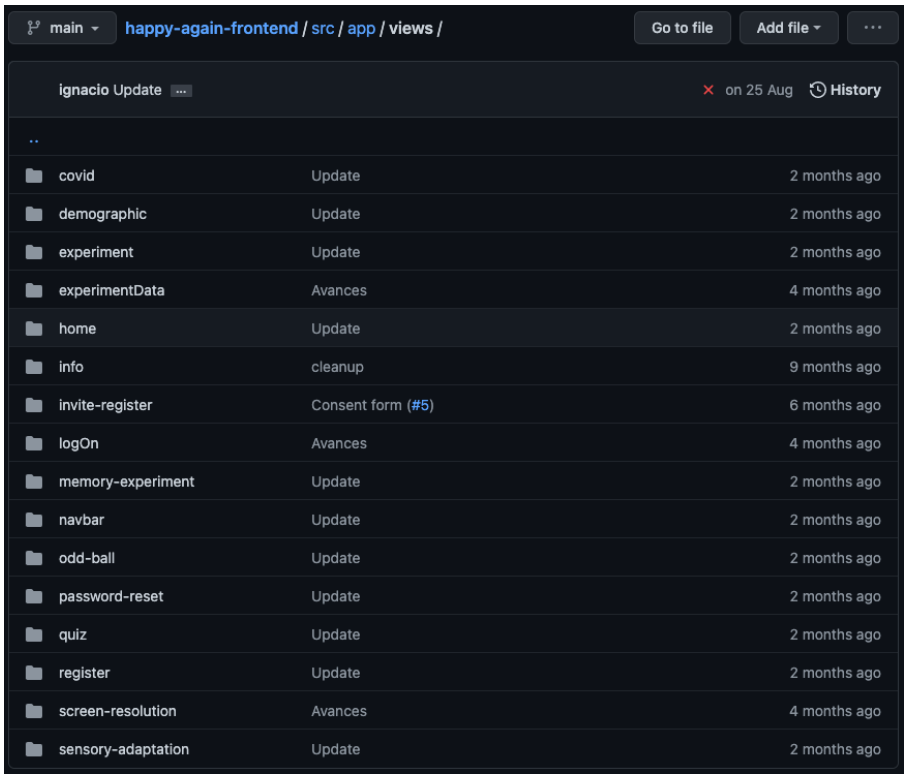

Figure 6.4: Views directory.

The APIs are implemented within the services. All services created for this project are located in the happy-again- frontend/src/app/service directory and are shown in Fig. 6.6.

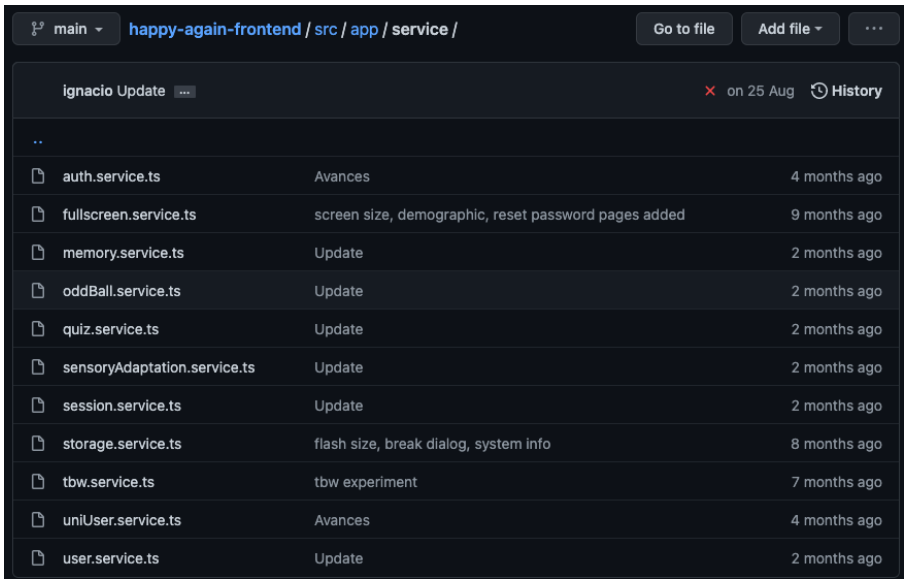

Figure 6.5: Service directory.

6.2.2 Interface

In the following, all the pages created will be presented and their role within the application will be indicated.

1. **Home Page**

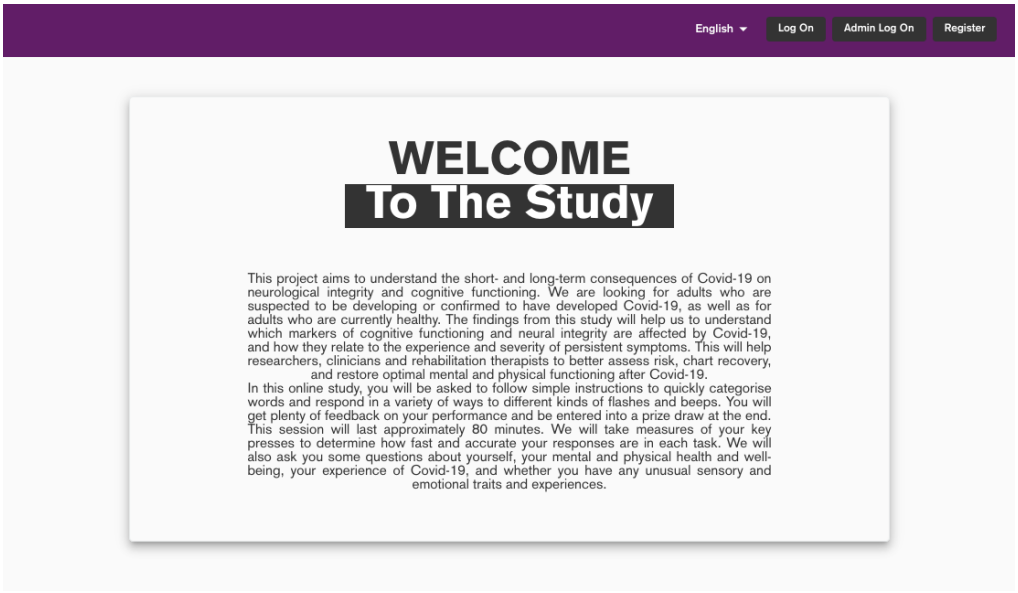

Figure 6.6: Home Page.

This corresponds to the home screen (Fig. 6.6), where the information of the study that is being carried out is detailed. We can also see that the top bar gives us the possibility to perform several actions. We can change the language (the application is available in English, Italian, Portuguese, Spanish, French and Greek), log in as administrator or as a normal user and there is also the option to register.

If you are already logged in, you can see a similar screen but with details of the progress of the tests carried out, which we can see below in fig 6.7.

2. **Registration and Log in**

Below is the registration flow. First you are asked for 3 mandatory fields, name, email and password. These have validator fields, so that no incorrect information is entered (e.g. entered with an email in the wrong format). After confirming, you are taken to the consent form which you must complete in order to successfully submit the registration application. When you send the request, an email will arrive at the email address you entered, to confirm your account and log in. In fig 6.8 and fig 6.9 we can see the flow just described.

After having confirmed the account, the user can log in, where the participant will have to enter his e-mail address and the chosen password (Fig. 6.10).

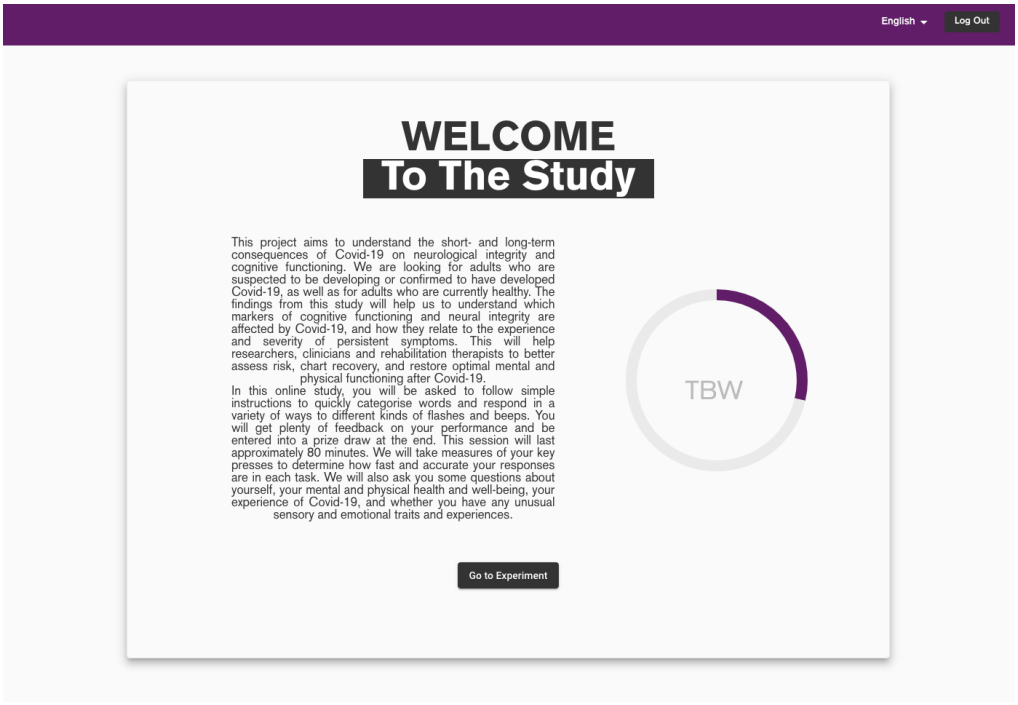

Figure 6.7: Home Page Logged in.

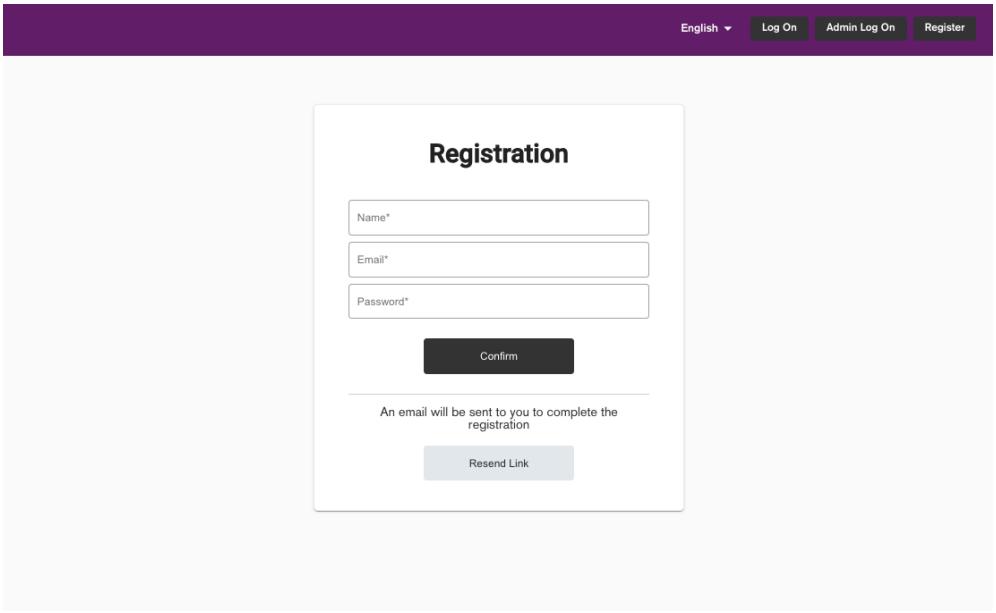

Figure 6.8: Registration of new account.

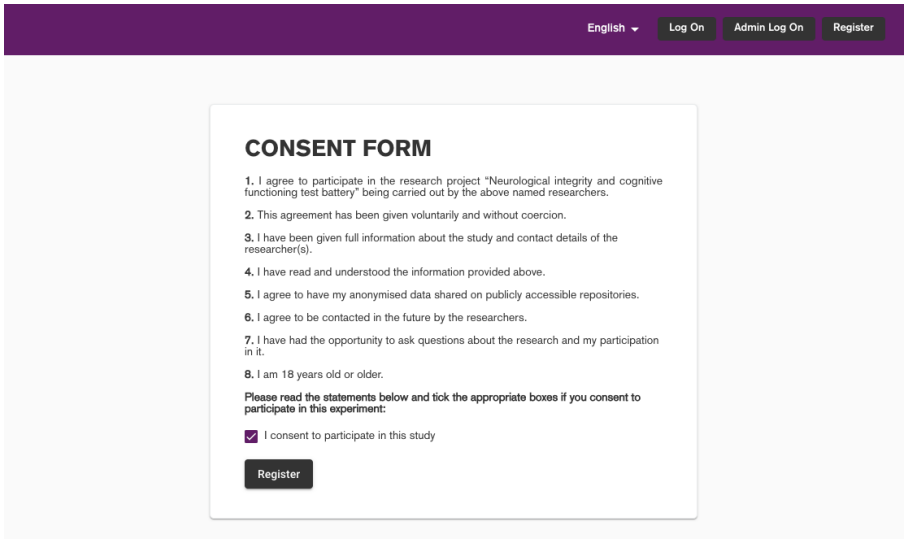

Figure 6.9: Consent Form.

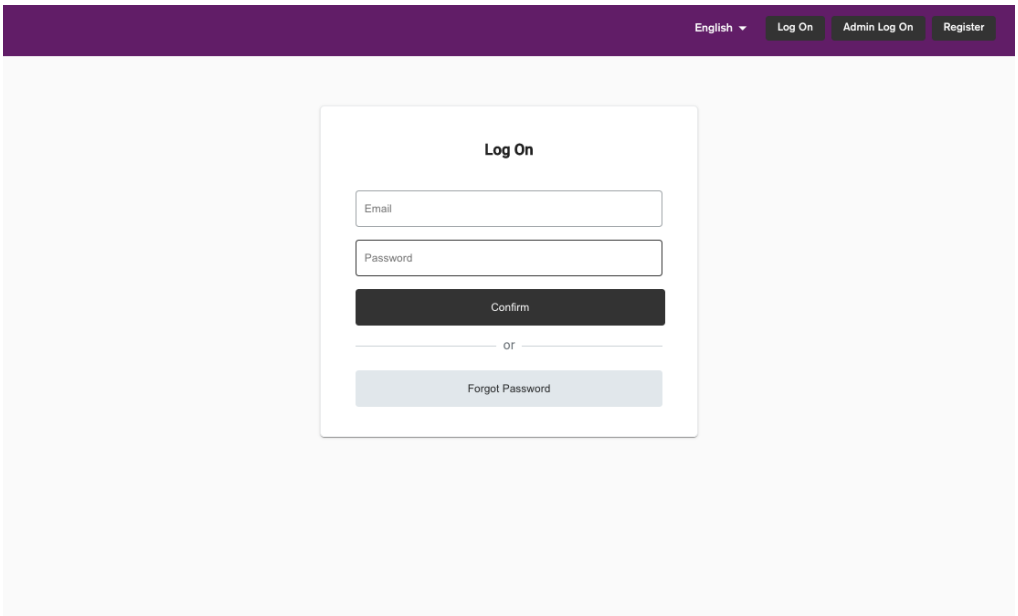

Figure 6.10: Log On page.

3. Covid questionnaire

The first experiment a user must perform is the covid questionnaire. It seeks to find out about the user's history with the virus, such as symptoms, duration of symptoms, severity, etc. It has an alternative question format, where, depending on the answer, more questions appear (Fig 6.11). The button to complete the questionnaire is enabled when you answer all the questions. When you complete the quiz, you are sent to the home page where you can see your progress and continue with the next experiments.

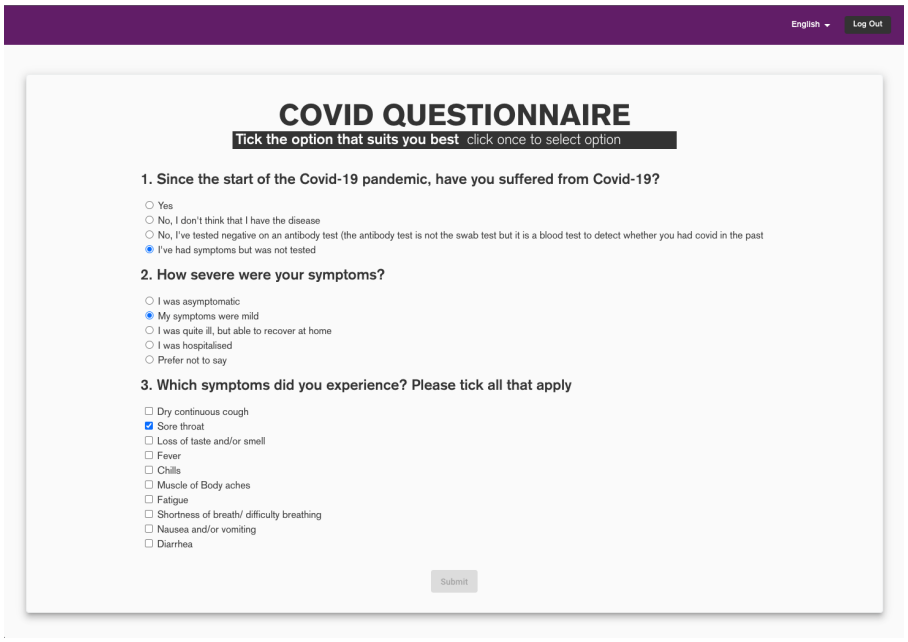

Figure 6.11: Covid Questionnaire.

4. **Memory Experiment**

The memory experiment is divided into two parts. The first part, called the encoding task, is about deciding whether the words displayed on the screen are man made or alive. This can be done by pressing the "S" key if the answer is "yes" and with the "L" key if the answer is "no".

The first thing that appears on the screen when entering the experiment are the instructions of the experiment, and when pressing the "Start" button, a demo will start, so that the user understands the experiment and performs it in a good way. In total, the first part consists of 4 blocks, each of which consists of 50 words. In the figure 6.12 and 6.13 we can see the interface of the explained.

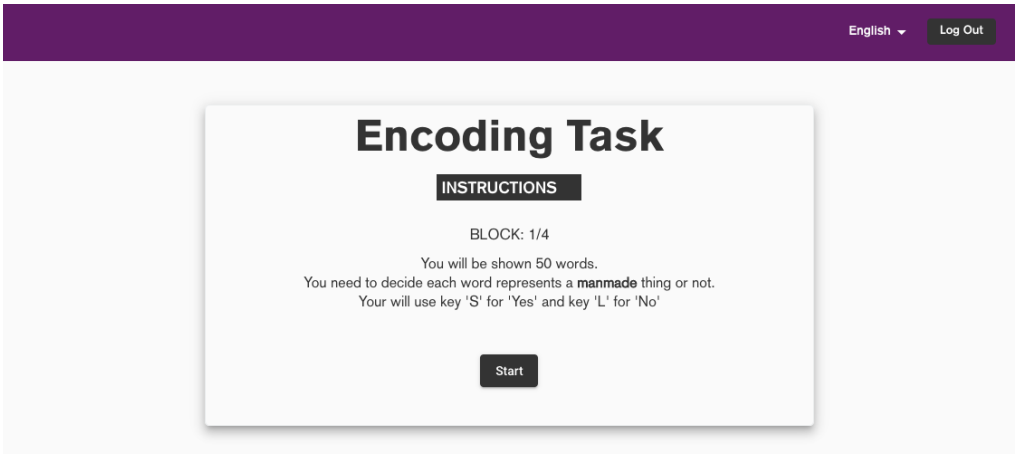

Figure 6.12: Encoding instruction.

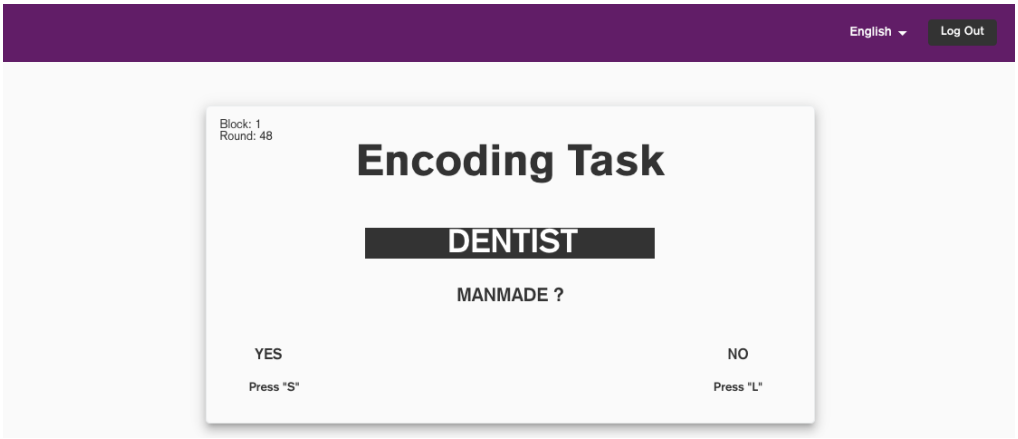

Figure 6.13: Encoding task.

At the end of the encoding task, the second part of the memory experiment begins, called the recognition task. Like the previous one, the associated instructions are presented, with the option of performing a demo (fig 6.14 and 6.15). This part consists of 6 blocks, each with 50 words. To answer, you can use the keys 1 to 5, or optionally you can press the buttons presented on the screen.

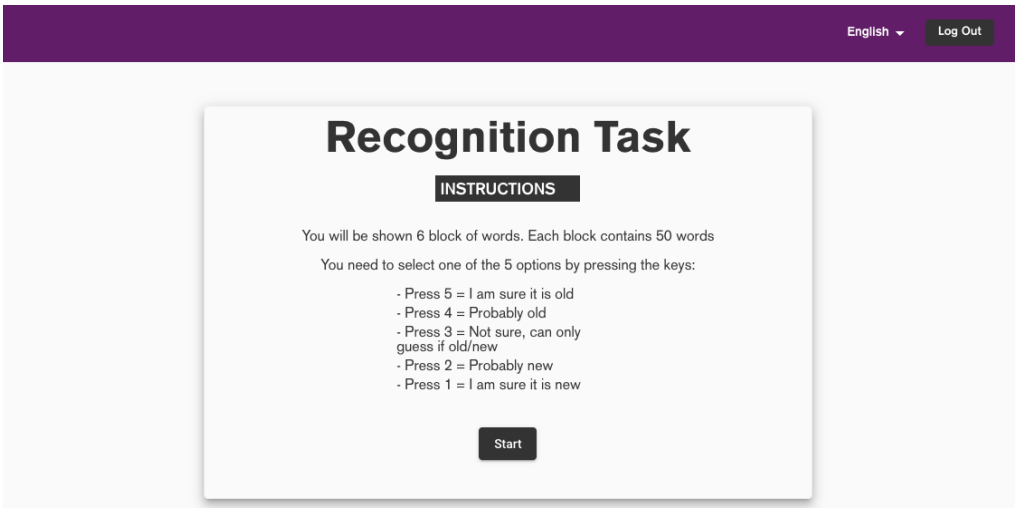

Figure 6.14: Recognition instruction.

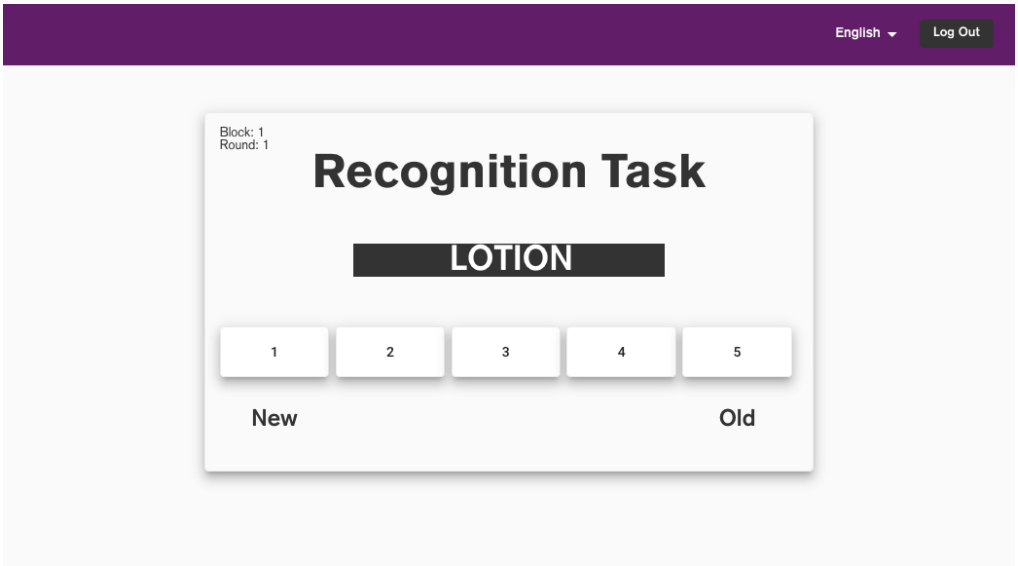

Figure 6.15: Recognition task.

5. **Temporal Binding Window**

   The next experiment is the TBW, where, like the previous ones, the instructions are shown, where the option is given to start with a demo of the exercise. This test requires a full screen, so if you minimise the page, the test is paused until it returns to full screen. Moreover, this test can be paused at any time, since each answer is sent in a POST to the backend, to be saved.

   At the start of the test, a black circle will appear and a beep will sound, where the user must choose which happened first (the sound or the appearance of the black circle). To make this decision, there are 2 options, press the left arrow (sound) or right arrow (sound). The second are buttons that represent the situation. In fig 6.16 and 6.17 we can see the situation.

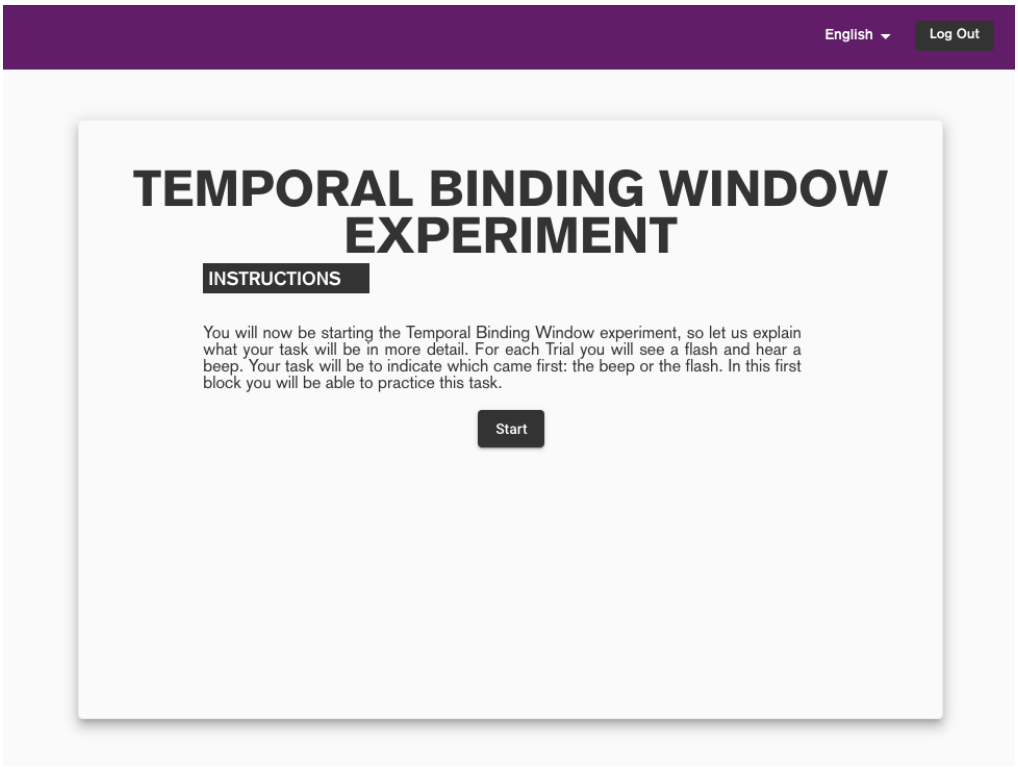

Figure 6.16: Temporal Binding Window Instruction.

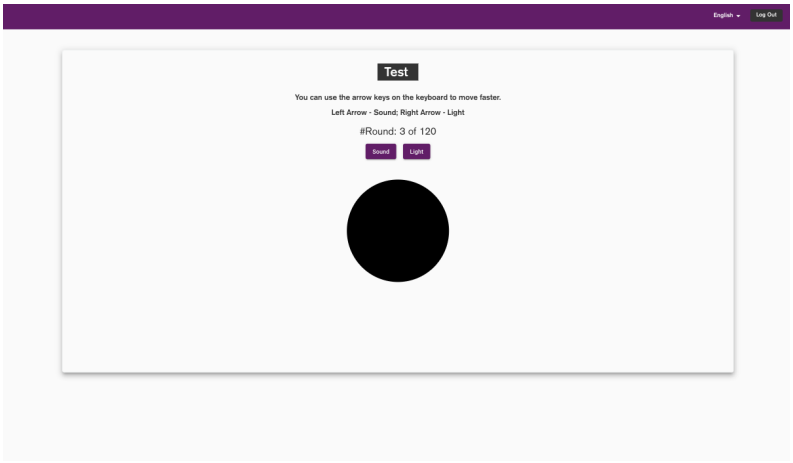

Figure 6.17: Temporal Binding Window Task.

6. SPQ Questionnaire

The next experiment is the Schizotypal Personality Questionnaire, which consists of 74 questions, all of which must be answered for the submit button to be activated (Fig. 6.18). In addition, for each question only one option can be selected.

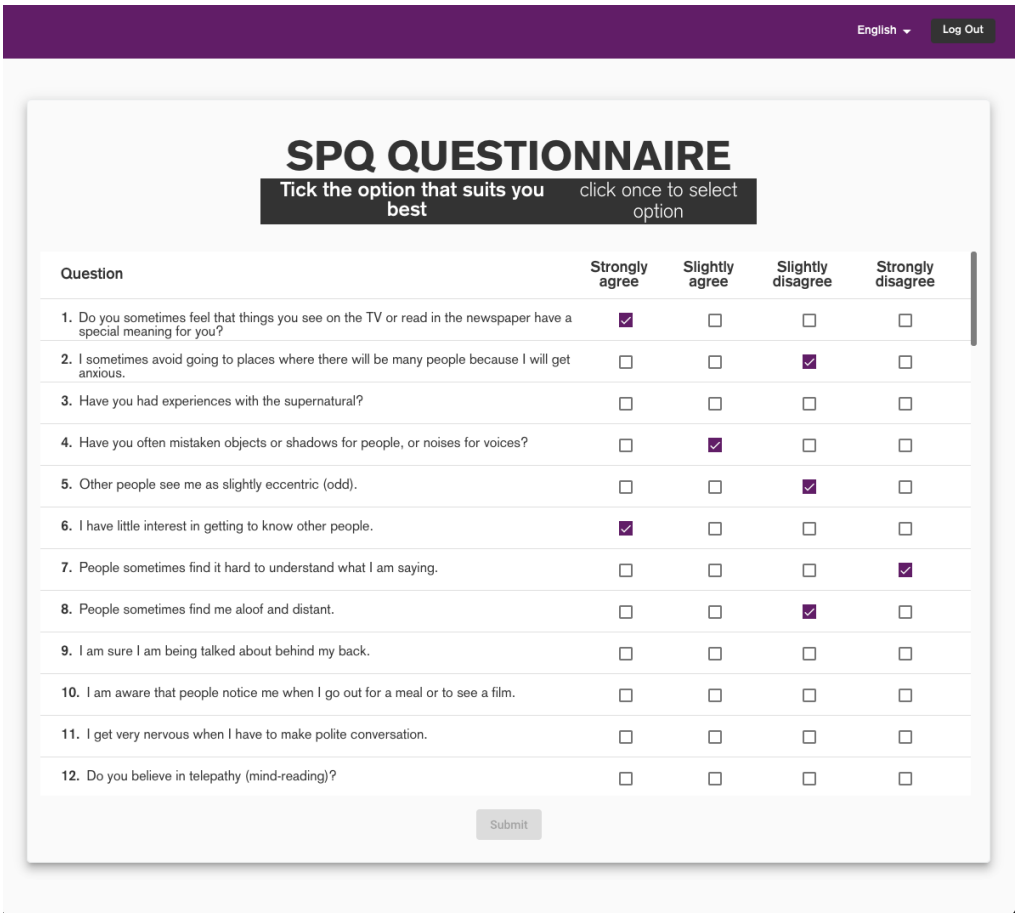

Figure 6.18: Schizotypal Personality Questionnaire.

7. **Demographic questionnaire**

This is followed by a demographic questionnaire, which seeks to obtain more information from the respective user (Fig. 6.19). These are alternative questions, where it is mandatory to answer all questions.

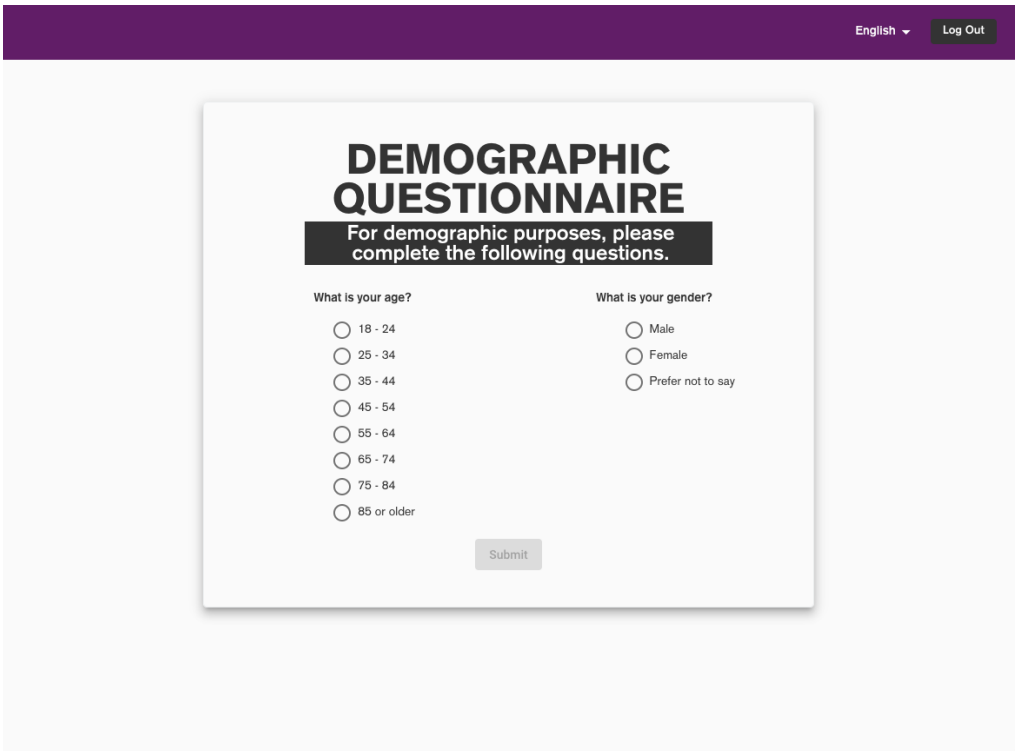

Figure 6.19: Demographic questionnaire.

8. Sensory Adaptation Experiment

The next experiment corresponds to the Sensory Adaptation, where, like most of the experiments with a more complicated interaction, it has the instructions (fig 6.20). Here an audio is presented that was generated with matlab, in the required frequency, which has a duration of 190 seconds. Every 20 seconds the user must indicate whether the sound presented has decreased in volume with respect to the initial sound. This decision to be made by the user is presented as a percentage, where the buttons are activated when the given time elapses and deactivated when the time elapses or the question is answered. This situation can be seen represented in fig 6.21 and 6.22. It is recommended to use headphones and to be looking straight ahead at all times to carry out the test.

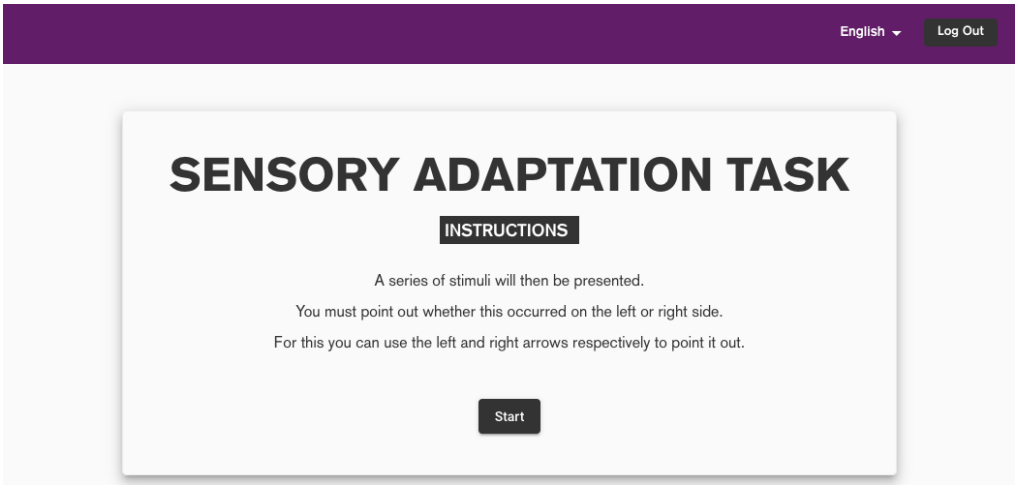

Figure 6.20: Sensory Adaptation Instruction.

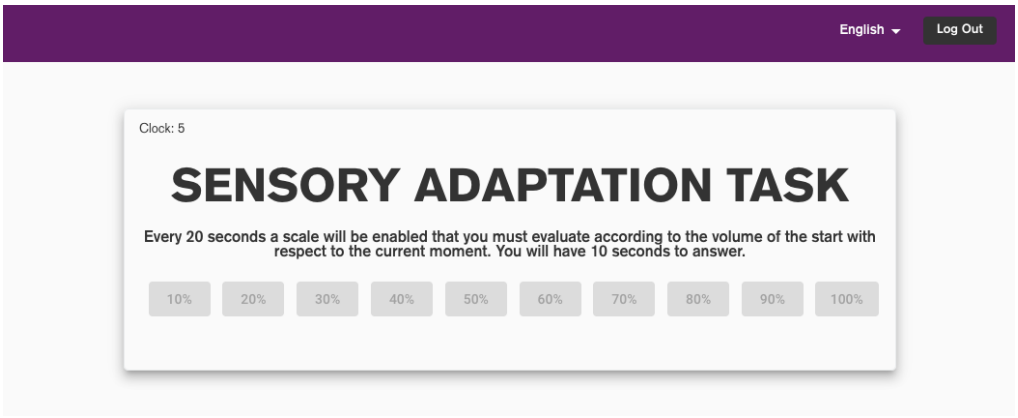

Figure 6.21: Sensory Adaptation Experiment.

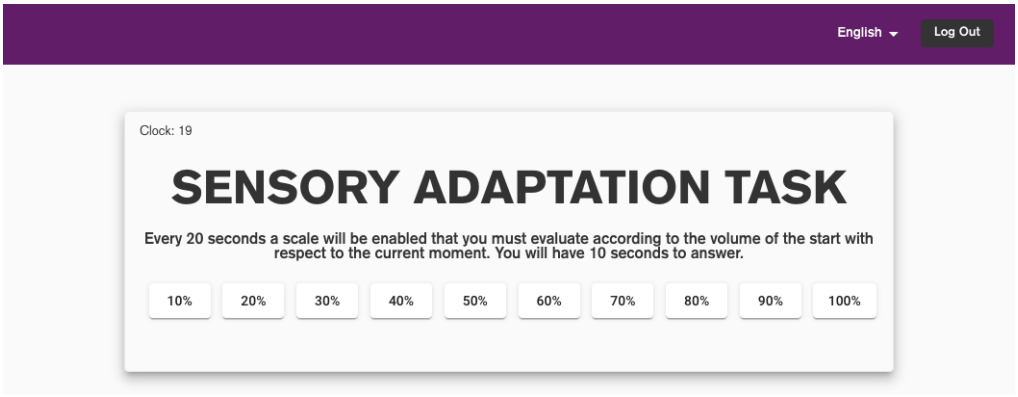

Figure 6.22: Sensory Adaptation Experiment active.

9. **Odd-ball experiment**

Finally, there is the odd ball experiment, which starts with the instructions of what the user has to do. This is represented by the figures 6.23 and 6.24, in which 10 blocks are evaluated, in which both visual and auditory stimuli are presented. With these stimuli, the user must determine whether the stimuli occurred on the left or right side. For this experiment, headphones must be used, since the sounds will also be localised to the left or right ear. In each block 105 stimuli will be presented.

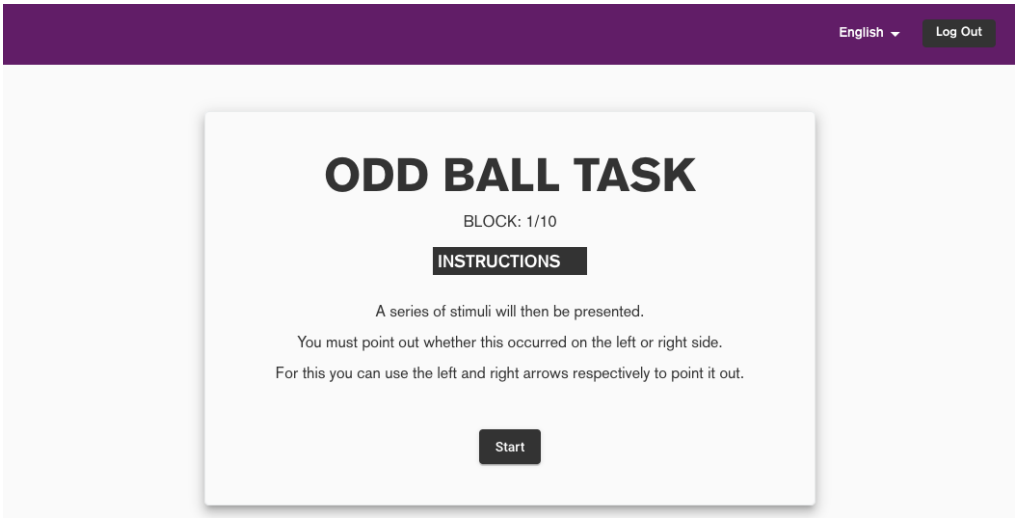

Figure 6.23: Odd ball instruction.

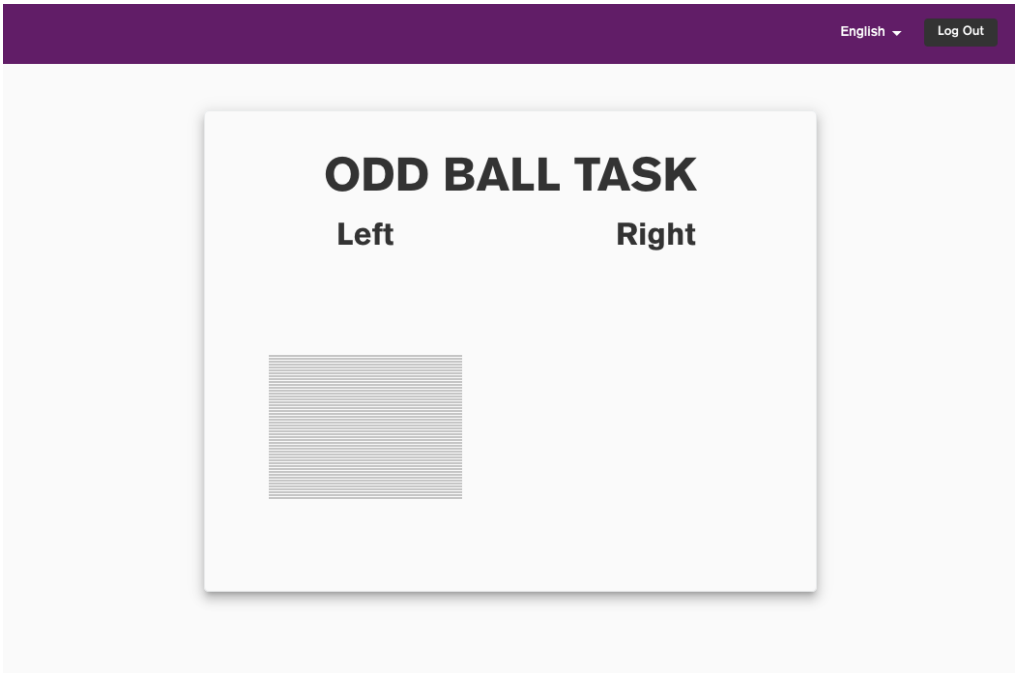

Figure 6.24: Odd ball task.

10. **Data download**

This interface is presented to users with privileges to download the data, and you must log in as an administrator. Here you can select any experiment, and the data is downloaded in .csv format (Fig 6.25).

Figure 6.25: Download Data interface.

11. **Registration of users with privileges**

As in the previous [**?**] interface, you must be logged in as a user with special privileges. Here you must select the email of the new administrator user you wish to create, and the respective role of that user (fig 6.26). When you confirm, the new user will receive an email to confirm his account and assign a password. After that, the user can log in as administrator.

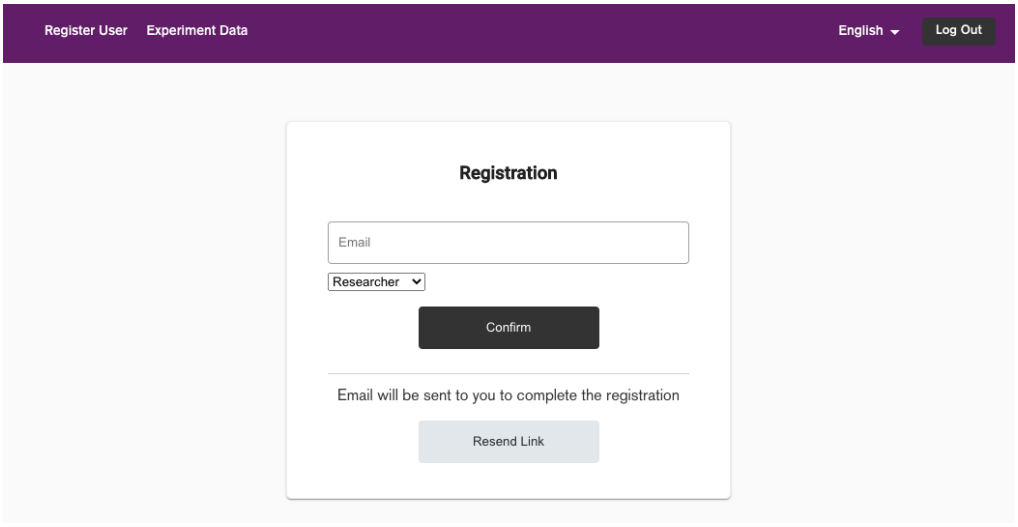

Figure 6.26: Registration of users with privileges.

Chapter 7

Conclusion

The developed programme aims to enable future research into the long-term cognitive effects, based on an analysis of the data obtained by the different participants performing the tasks. The system allows users to complete four tasks and three questionnaires. These were chosen by a group of doctors looking to better measure the four behavioural markers, speed and accuracy of response to visual and auditory stimuli, the "temporal binding window" of sensory integration, sensory adaptive responses and retention of verbal information.

A number of issues were faced during the thesis, which were addressed or pointed out as issues to be considered. Firstly, the establishment of a scalable architecture, because of possible changes in the software in the future. This is why the project is modularised in an optimal way, with documentation in code and using the PEP8 standard for Python and a typical JavaScript naming convention in order to make the code maintainable and scalable in the future.

A second problem encountered was the handling of portability. This is because different browsers had to be analysed and made to work in all browsers and the different resolutions that the participants might have. In addition, there is always the human factor, who may not comply with the instructions, such as using headphones or answering the experiments incorrectly. This is why the instructions are detailed for each experiment, and why some experiments are restricted if they do not meet certain technical conditions.

The third problem was the constant change in the requirements, which generated certain architectural changes that had not been considered previously. A lot of iterations were carried out in order to achieve a good result and have a fully functional software.

The challenge for future work is to establish a methodology for analysing the data obtained from the tasks carried out by the participants. This could mean the creation of a predictive model that provides information about behavioural markers based on the responses collected. In addition, conclusions can be drawn about the cognitive effects of COVID-19.

It should be noted that based on the information collected, new requirements of the experiments already developed will be raised, so it is also expected that these will undergo changes at the code level and since the project follows a modular structure, there would be no major problem.

From this system, it is expected to obtain relevant information, which will be made public, so that various researchers around the world can analyse it in order to make further progress in relation to the effects of COVID-19 and its possible cures.

Bibliography

[1] N. Lyn, 55% of coronavirus patients still have neurological problems three months later: study, August 2020, https://www.marketwatch.com/story/

[2] C. Cabezas, Pandemia De La COVID-19: Tormentas y retos, Rev Peru Med Exp Salud Publica. October 2020, pp. 603-604, https://scielosp.org/article/rpmesp/2020.v37n4/603-604/

[3] M. Pérez, J. Gómez & R. Dieguez, Características clínico-epidemiológicas de la COVID-19, Revista habanera ciencia médica, March-April 2020, http://www.revhabanera.sld.cu/index.php/rhab/article/view/3254/2505

[4] Minsal, Tipos de Caso y Contacto COVID19, November 2020, https://saludresponde.minsal.cl/tipos-de-caso-y-contacto-covid19

[5] J. Madrigal, M. Quesada, M. Garcia & A. Solano SARS CoV-2, manifestaciones clínicas y consideraciones en el abordaje diagnóstico de COVID19, Revista Médica de Costa Rica, Vol. 85, No. 629, January-June 2020 https://www.medigraphic.com/pdfs/revmedcoscen

[6] Organizacion Mundial de la Salud, Coronavirus, 2021, https://www.who.int/es/health-topics/coronavirus#tab=tab_2

[7] K. Ritchie, D. Chan & T. Watermeyer, The cognitive consequences of the COVID-19 epidemic: collateral damage?, Brain Communications, Vol. 2, No. 2, May 2020, pp. 1-4, DOI 10.1093/braincomms/fcaa069

[8] A.Wise & M. Barnett-Cowan,Perceived Simultaneity and Temporal Order of Audiovisual Events Following Concussion, Front. Hum. Neurosci, 12:139, 2018, DOI 10.3389/fnhum.2018.00139

[9] Neuroimage, A critical role of the human hippocampus in an electrophysiological measure of implicit memory, April 2015, 109: 515–528, DOI 10.1016/j.neuroimage.2014.12.069.

[10] R. Lawson, J. Aylward, S. White & G. Rees, A striking reduction of simple loudness adaptation in autism, November 2015, DOI 10.1038/srep16157.

[11] D. Senkowski, D. Talsma, C. Herrmann & M. Woldorff, Multisensory processing and oscillatory gamma responses: effects of spatial selective attention, September 2005, DOI 10.1007/s00221-005-2381-z.

[12] R. Rodriguez & M. Goncalvez, Perfil UML para el modelado visual de requisitos difusos, September 2009, http://ve.scielo.org/scielo.php?script

[13] J. Fong & R. Hui , Application of middleware in the three tier client/server database design methodology, July 1999, DOI 10.1590/S0104-65001999000200005.

[14] Angular, Features & Benefits, 2021, https://angular.io/features .

[15] Mozilla, JavaScript, December 2021, https://developer.mozilla.org/es/docs/Web/JavaScript .

[16] Flask, User's Guide, 2021, https://flask.palletsprojects.com/en/2.0.x/

[17] M. Jones, J. Bradley & N. Sakimura, "JSON Web Token (JWT)", RFC 7519, May 2015, DOI 10.17487/RFC7519

[18] Microsoft, The Model-View-ViewModel Pattern, September 2021, https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm