

---

# **Development of an Android application as a personalised visual rehabilitator for patients with Macular Degeneration**

Desarrollo de una aplicación Android como rehabilitador visual personalizado para pacientes con Degeneración Macular

---



Degree in Computer Science 2021-2022

Faculty of Computer Science

Authors:

**Escobar Suarez, Daniela Alejandra**

**Martín Puebla, Alba María**

**Núñez Conde, María**

**Rodríguez Pereira, Diego Alejandro**

Supervised by:

**Guijarro Mata-García, María**

**Recas Piorno, Joaquín**

Collaborators:

**González Vides, Leonela**

**González Montero, María Guadalupe**

# **Development of an Android Application as a Personalised Visual Rehabilitator for Patients with Macular Degeneration**

Desarrollo de una Aplicación Android como Rehabilitador Visual Personalizado para Pacientes con Degeneración Macular

Computer Science's Final Project By:

**Escobar Suarez, Daniela Alejandra**

**Martín Puebla, Alba María**

**Núñez Conde, María**

**Rodríguez Pereira, Diego Alejandro**

Supervised by:

**Guijarro Mata-García, María**

**Recas Piorno, Joaquín**

Collaborators:

**González Vides, Leonela**

**González Montero, María Guadalupe**

Final Project 2021-2022

Faculty of Computer Science

Complutense University of Madrid

Madrid, 2022



## **Dedication**

We dedicate this app to the Macular Degeneration patients. We hope that this project improves their lives.

We also want to dedicate this project to all the teachers who have taught us throughout our university education.

Finally, to all our loved ones.

## **Acknowledgment**

We express our deepest gratitude to Leonela González Vides and María Guadalupe González Montero for their support, advisement, and input on the matter.

We want to thank María Guijarro Mata-García and Joaquín Recas Piorno for making this project possible.

# Abstract

The Age-related Macular Degeneration (AMD) is an illness that creates one or more stains in the vision that may affect significantly in the everyday life of whoever suffers it. It is a disease that afflicts mostly to people of advanced age and, nowadays, their treatments are not modernized to the digital era. In the present day, it is mainly tested and exercised through paper, as there is not enough investment to digitalize, which would make easier the treatment of the patient, therefore, make easier the professional work.

The objective of this project is to develop an application that helps both patients with age-related macular degeneration and professionals who work with them. This application allows the professionals to keep tracking of the progress of their patients, in this way they can follow up on the evolution of the macular degeneration and the progress of the exercises that the patient is doing.

The application has evaluations, which can be used by professionals to properly monitor the patients. Thanks to them it is established the location of the reference dot of the patient, to help them see correctly, therefore, the app allows a personalised usage. This dot is the place where the patient has to look so that the stain goes to that location, allowing them to see what they want with the peripheric vision.

Besides, the app has 17 exercises to help the patient's visual rehabilitation. These exercises have different categories, from identifying letters and figures to recognizing parts of the face and everyday objects. Said exercises can be recommended by the professional through a personalized analysis using patient's clinic data.

The aid of the project is that the patient, besides being able to exercise with the professional, also may do it on their own tablet from the comfort of their house without losing their follow-up. Giving the opportunity of doing exercises more frequently than the traditional way where the rehabilitation is only made in person.

## Keywords:

Macular degeneration, rehabilitator, application, technologies, Android, AMD.

## Repository access:

Drive: [https://drive.google.com/file/d/1BMZCqhv4lfnnvFVDEH-m7-uvX\\_YG9nJo/view?usp=sharing](https://drive.google.com/file/d/1BMZCqhv4lfnnvFVDEH-m7-uvX_YG9nJo/view?usp=sharing)

GitHub: <https://github.com/DaniEscUCM/VisualRehabilitator.git>

## Resumen

La Degeneración Macular Asociada a la Edad (DMAE) es una enfermedad que produce una o varias manchas en la visión que puede afectar significativamente en el día a día de las personas que lo padecen. Este trastorno ocular es diagnosticado mayoritariamente a las personas de edad avanzada, y actualmente, los tratamientos no se han actualizado a la era digital. Hoy en día se utilizan principalmente pruebas y ejercicios en papel debido a que no hay suficiente inversión en digitalizarlo, lo cual facilitaría el tratamiento a los pacientes, y por consiguiente, el trabajo de los profesionales.

El objetivo de este proyecto es desarrollar una aplicación que ayude tanto a las personas que sufren de la degeneración macular asociada a la edad como a los profesionales de la salud que trabajan con este tipo de pacientes. Esta aplicación permite llevar un seguimiento de los pacientes con los que el profesional trabaja, en el que podrán seguir la evolución de la degeneración macular y el progreso de los ejercicios que va realizando el paciente.

La aplicación cuenta con evaluaciones, las cuales pueden ser utilizadas por los profesionales para realizar un seguimiento adecuado de los pacientes. Gracias a esto se puede determinar dónde ubicar el punto de referencia del paciente, el cual es diferente según la persona, para ayudarlo a ver correctamente, por lo que la app permite un uso personalizado. Dicho punto indica el lugar donde tiene que fijar la vista para que la mancha quede en esa ubicación, permitiendo así ver su objetivo con la vista periférica.

A su vez, la aplicación cuenta con 17 ejercicios para ayudar a los pacientes en su rehabilitación visual. Estos ejercicios son de distintas categorías, desde identificar letras y figuras hasta reconocer partes de una cara y objetos de la vida cotidiana. Dichos ejercicios pueden ser recomendados por el profesional haciendo un análisis personalizado con el uso de los datos clínicos del paciente.

La intención del proyecto es que el paciente, además de poder realizar los ejercicios junto con el profesional, también pueda practicar con su tableta personal desde el confort de su casa sin perder seguimiento alguno del progreso y de los resultados. Dando la oportunidad de hacer ejercicios más frecuentemente que por la vía tradicional donde la rehabilitación es efectuada únicamente de manera presencial.

### Palabras clave

Degeneración macular, rehabilitador, aplicación, Android, tecnologías, DMAE.

### Acceso al repositorio:

Drive: [https://drive.google.com/file/d/1BMZCqhv4lfnnvFVDEH-m7-uvX\\_YG9nJo/view?usp=sharing](https://drive.google.com/file/d/1BMZCqhv4lfnnvFVDEH-m7-uvX_YG9nJo/view?usp=sharing)

GitHub: <https://github.com/DaniEscUCM/VisualRehabilitator.git>

# Index

<b>Dedication</b> .....	<b>IV</b>
<b>Acknowledgment</b> .....	<b>IV</b>
<b>Abstract</b> .....	<b>V</b>
<b>Resumen</b> .....	<b>VI</b>
<b>Figures index</b> .....	<b>IX</b>
<b>Chapter 1 – Introduction</b> .....	<b>10</b>
1.1. State of the art .....	10
1.2. Objective .....	16
<b>Chapter 2 - Workplan</b> .....	<b>18</b>
2.1 Requirements.....	18
2.1.1 Requirements for improving usability .....	19
2.2 Software modelling .....	19
2.3 Tasks distribution.....	20
<b>Chapter 3 - App design</b> .....	<b>23</b>
3.1 ¿Why Android? .....	24
3.2 Use Cases .....	24
3.2.1. Patient .....	24
3.2.1.1 Patient Main Menu .....	25
3.2.2 Professional .....	25
3.2.2.1 Professional Main Menu .....	26
3.2.2.2 Professional Patient.....	26
3.2.2.3 Professional New Patient.....	26
3.2.2.4 Professional Profile .....	28
3.2.3 Error proofing .....	29
3.3 Database .....	31
3.3.1 Authentication Database.....	31
3.3.2 Real-Time Database .....	32
<b>Chapter 4 - Patient evaluation</b> .....	<b>37</b>
4.1 Drawing Dots in the correct coordinates.....	37
4.2 Manual input from professional.....	38
4.3 First Test .....	39
4.4 Second Test.....	40
4.5 Results from the tests .....	40
<b>Chapter 5 - Exercises</b> .....	<b>42</b>

5.1 First exercise .....	43
5.2 Second exercise .....	43
5.3 Third exercise.....	44
5.4 Fourth exercise .....	44
5.5 Fifth exercise .....	44
5.6 Sixth exercise.....	44
5.7 Seventh exercise .....	44
5.8 Eighth exercise .....	45
5.9 Ninth exercise.....	45
5.10 Tenth exercise .....	45
5.11 Eleventh exercise .....	45
5.12 Twelfth exercise .....	46
5.13 Thirteenth exercise .....	46
5.14 Fourteenth exercise .....	47
5.15 Fifteenth exercise .....	47
5.16 Sixteenth exercise .....	47
5.17 Seventeenth exercise .....	48
5.18 Pause.....	48
5.19 Focus dot settings.....	49
5.20 Charts information.....	49
<b>Chapter 6 - Results.....</b>	<b>51</b>
6.1 Technologies used .....	52
6.2 API Version Adaptations .....	54
6.3 Screen Size Adaptations.....	55
<b>Chapter 7 - Conclusions .....</b>	<b>57</b>
<b>Chapter 8 - Contributions.....</b>	<b>58</b>
8.1 Daniela Alejandra Escobar Suarez.....	58
8.2 Diego Alejandro Rodríguez Pereira .....	59
8.3 Alba María Martín Puebla.....	62
8.4 María Núñez Conde.....	64
<b>Bibliography .....</b>	<b>66</b>
<b>Appendix .....</b>	<b>69</b>
A. Figures .....	69

# Figures index

Figure 1 Central Vision Test: Amsler grid test.....	11
Figure 2 Eyecare: Amsler grid test .....	12
Figure 3 Amsler Grids: Amsler grid test .....	13
Figure 4 Verana Vision Test .....	14
Figure 5 MaculaTester .....	15
Figure 6 Gantt Chart.....	22
Figure 7 First design .....	23
Figure 8 Balsamiq design.....	23
Figure 9 App design.....	24
Figure 10 Material Design recommendations .....	29
Figure 11 Confirm message .....	30
Figure 12 Unfilled fields .....	30
Figure 13 Error message.....	30
Figure 14 Error in field .....	31
Figure 15 Firebase Authentication.....	32
Figure 16 Relative coordinates .....	38
Figure 17 Grid example .....	38
Figure 18 Calculated Focus .....	40
Figure 19 Weighted calculated focus.....	40
Figure 20 Dots setting.....	49
Figure 21 Exercise chart.....	49
Figure 22 Circle Image View sample .....	53
Figure 23 Compatible Android devices.....	54

# Chapter 1 – Introduction

Visually impaired people are a small but significant portion of the population, and they are generally left behind from a technological standpoint, especially with the introduction of mobile devices that do not incorporate any physical buttons. In addition to this, there is also a significant percentage of visually impaired people that are elderly, that do not receive encouragement and assistance in the process of learning to use technology.

In recent years, there has been an increasing development of mobile applications that uses a lot of amazing and cutting-edge technology, and with them a significant increase in the development of accessible mobile applications that help people with visual difficulties. Nevertheless, the percentage of the applications that incorporate any form of accessibility does not correlate with the percentage of the visually impaired individuals amongst the population. Besides this, there are even fewer applications that help to diagnose macular degeneration, especially for elderly people that are discouraged to use mobile devices. It was found that there are no applications available for free that allow for effective follow-up of the patient's progress and help them improve their vision. As said in section 1.1, most of the applications that are specialized in macular degeneration only have an evaluation of the stain, but they do not have any exercises to help them improve and they do not have a database that connects the patient to their doctor so, they cannot trace the patient's progress

At the moment of writing, most applications do not count with rehabilitating exercises for patients with Age-related Macular Degeneration (AMD). The few that have some, are paid.

Using modern technologies, patients with macular degeneration will be able to practice not only during the scheduled rehabilitation hours but also in any place without professional supervision. This will give them the opportunity to improve more as they will be able to be recurrently tested, practice exercises frequently, and have their progress followed up. This will be more manageable with this application.

This application will not be like any other that already exists. It will adapt to macular degeneration users, using large fonts and images to help the user see clearly. Also, people with this illness are generally old and not accustomed to modern technologies, so they need a simplified interface, with few and big buttons. Those are the reasons why only the essential buttons were included, with the text in large fonts and with substantial gaps between them.

## 1.1. State of the art

In the moment of writing, there is no other free application that treats AMD with both patient tests and exercises. There are only a few similar apps on the market:

- **Central Vision Test:** Play Store application developed by healthcare4mobile[1]. This app was made to carry out simple tests with the use of Amsler grid[2] to monitor the central visual field, as shown in Figure 1 Central Vision Test: Amsler . Central Vision Test was last updated in 2014, because of this, the app does not work with the latest Android versions. The app claims that the test is useful for patients with macular degeneration, house visits, checking if the user needs to visit the doctor, and nursing homes.

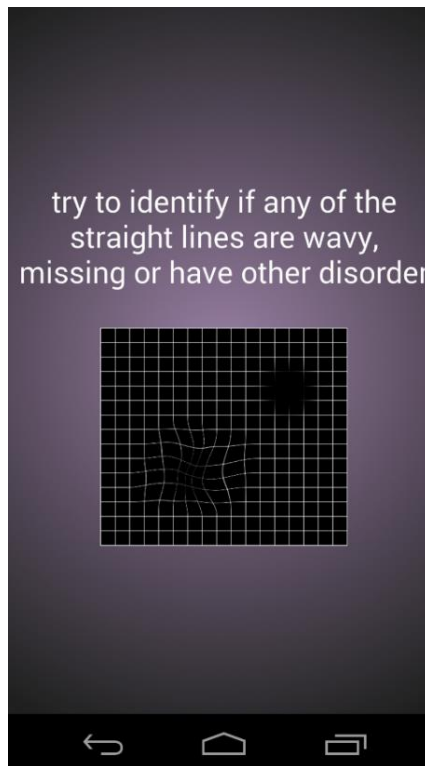


Figure 1 Central Vision Test: Amsler grid test

- **Eyecare - Amsler Grid Eye Test:** Appstore and Play Store application developed by Eyera Enterprises LLC[3]. It is an application designed for people with macular degeneration to monitor the development of their vision loss. This app lets the person perform the Amsler grid test for both eyes as show in Figure 2 Eyecare: Amsler grid test. Once the test is completed, the results are sent to the cloud to be stored, but the user can never access them. Moreover, professionals can register into the application so that patients can reach them. The team tried the application, and the conclusion reached was:
  - The application requires Android 4.4 and up.
  - The app has a problem with the Amsler grid size in landscape mode, where the grid gets cut off.
  - The general size of the buttons and text are too small, and it is difficult to see because they use low contrast.
  - The test can only be performed and sent to the cloud, and the user can never have access to the result of the recent test or the previous tests.

- Once the tests are performed, an email is sent to the email address provided when registering. This email contains images of the test results, comparing them and telling if there was an improvement or not. Doing this requires leaving the application and using an email app to read the email and see the results.
- Then, the test results are sent to the cloud, which takes a considerable amount of time.

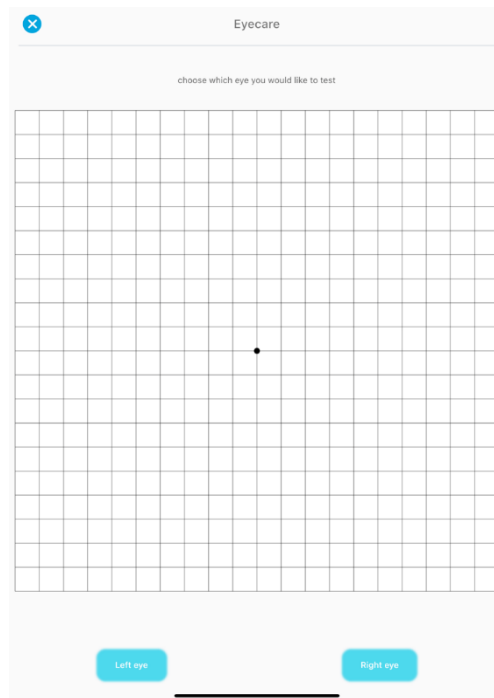


Figure 2 Eyecare: Amsler grid test

- **Amsler Grids:** AppStore application developed by Konan Medical USA [4]. This application deserves its name because it was one of the best apps implementing the Amsler Grid, as shown in Figure 3 Amsler Grids: Amsler grid test. The app's only function consists of having the Amsler Grid, which could later save in the device's internal storage.
  - It has three different test types: a white grid on black background, a red grid on a black background, and a black grid on a white background.
  - It also has three test options (each test): Blind Spot Amsler that monitors the test distance between tests, a fixation diagonal that shows two diagonals to improve guidance, and a high-resolution central area grid where the centre of the grid obtains smaller and precise grids.
  - Moreover, the app includes eye selection, erase a test, and an on-screen graphical test.

- Unfortunately, the “focus point” could only be changed from left to right and vice versa but could not be moved to another screen angle.

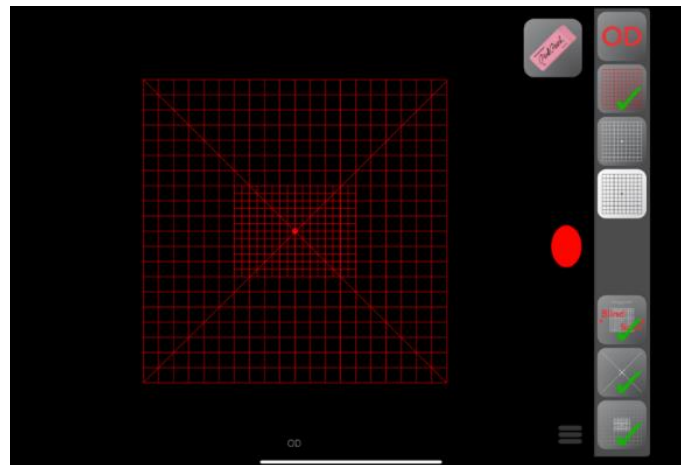


Figure 3 Amsler Grids: Amsler grid test

- **Verana Vision Test:** Appstore application developed by Verana Health, Inc with the contribution of specialised ophthalmologists [5]. The application detects the changes in a person's vision over the course of time and can perform two types of tests: a visual acuity test and an Amsler grid test. This app was one of the most highly valued apps available and the only one with a specialised professional contribution. The app is of a higher quality than the others, considering that it belongs to the free-app section; it has clear images and icons with a modest number of buttons and text, as shown in Figure 4 Verana Vision Test. Nonetheless, the Amsler grid is considerably smaller than the other apps tested and does not let the user change the orientation from vertical to landscape.
  - Key features:
    - Easy-to-use and understandable tests.
    - Clear instructions with audio to help guide the patient throughout the tests.
    - The app does not require the user to sign up.
    - Amsler grid test.

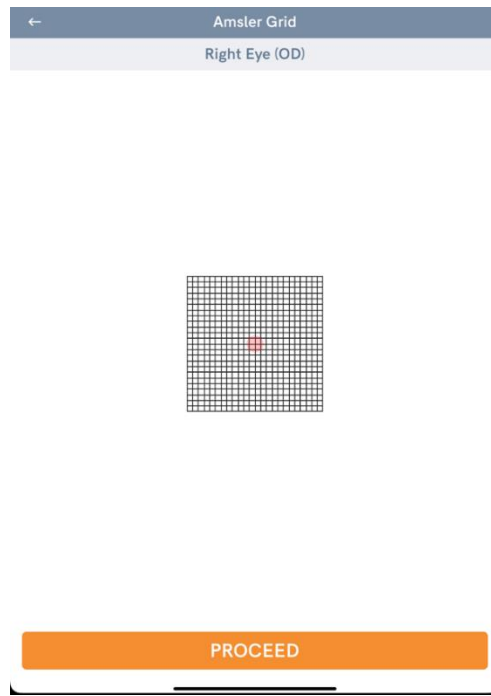


Figure 4 Verana Vision Test

- **MaculaTester:** AppStore and Play Store application developed by Sabina Technology, LLP, where vision tests with Amsler grid can be performed to verify the retina health and set reminders to conduct tests in the future. These tests can be stored to be later shown to the health professional in charge of the patient. Moreover, the tests can be compared with those already performed to analyse the development of the degeneration. The tests are filled out using the device's touch screen, where the person selects and marks the areas that they cannot perceive. This app is available on iPhone, iPad, and Android devices.
  - The app is available for 2.99\$.
  - It has a web [6].
  - The app has a video demonstration [7].
  - Figure 5 MaculaTester is shown the test that can be performed within the application. This Amsler grid test is considerably less visible than the other apps tested.

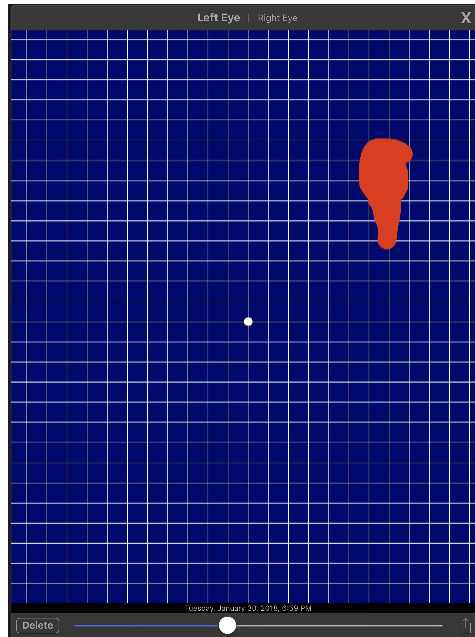


Figure 5 MaculaTester

- **Simply Visio:** Android and iOS application developed by Simply Deliver[8]. It can be found in the Play Store and AppStore, but it is only available in Spain on the Play Store. It is the first and only app that is available in different languages, including English and Spanish. The application offers four different exercises, and although the app is free to download, the access to these exercises requires payment, starting from 45 euros and ranging to 95 euros per exercise. Because of this, the exercises could not be tested. Despite the cost, from what could be seen on the web page, the app offers high-level and professional development exercises with a configurable text size and application theme that can be changed to the patient's preferences. The app also offers three apps within the application: a camera magnifier, a newspaper (that was not working at the time of the making), and a Sudoku. Nonetheless, in general, the app is not solid; it seems to crash often; the last update was in early 2019, and the login function does not remember users. Unlike the other applications mentioned before, this app does not provide any sort of evaluation to the user.
  - Web page: <https://simplyvisio.com>
  - Requires Android 5.0 and up.
  - Requires a tablet-size screen to see all functions correctly. Does not support full screen.
  
- **Eye Care Plus:** Android application (only available in Play Store) developed by healthcare4mobile [9]. The application is, along with Simply Vision, one of the most completed apps that were found. Nonetheless, the app lacks some important aspects. Unlike Simply Vision the app does not support other languages than English and does have tests that could be performed by the user, but these tests could not be performed by any doctor, nor have a way to introduce any data to personalize the exercises or know what is the patient suffering of. The test for macular degeneration, which they call the central

vision test, is similar to the Amsler grid, but unlike the other applications discussed above, it is not possible to select which areas the patient is unable to see, because the test only checks if the patient sees any blurry or missing lines, or that the lines appear wavy or bent, so it is only possible to suspect that the patient has a visual difficulty but is unable to do anything else to treat it or to at least tell which parts the patient is unable to see, unlike the other apps that the test could be evaluated by an ophthalmologist.

- The application does not support full screen, and it seems that is developed for only phones although in the Play Stores it says that is for all devices, because there was found that the grid layout for almost everything does not reach the total of the screen but only a part of it, and with that, all objects are partially seen. Also, the buttons and the images are small, which makes it difficult to see for even people that do not have any visual difficulties, and with that, there is no description for some of the buttons, which makes it difficult to know what a button does.
- Eye Care Plus is mostly focused on exercises, these exercises are for different types of eye problems, there are different categories that they offer, such as exercises for stimulation, lazy eye, dry eye, accommodation spasm, relaxation, and eye muscles. The application offers around 50 exercises at the time of writing, nevertheless, the application does not offer any exercise for macular degeneration. As said before, it can only suggest that a patient has macular degeneration but is unable to determine where could be the patient's focus point, for this reason, it is not possible to develop any exercise.
- Eye Care Plus offers some of the exercises with audio included, so the patient does not have to activate Talkback.
- The user is unable to interact in any form with the exercises. They were made to only follow or pursue figures, statics or with movement.
- The application offers tracing the patient progress, showing the progress of each day of the week, and trace the progress of each category of exercises.
- Eye Care Plus also offers 5 different training plans. Some of them are bedtime training, dry eye training, and lazy eye training. All of the plans are paid, so because of this cost, none of them could be tested.

## 1.2. Objective

**General Objective:** To develop an application that allows patients with Age-related Macular Degeneration to carry out personalised exercises for improving their sight. Making it possible for Professionals to keep track of their patient's progress and generating a focus dot that will guide the patients on where to position their eyes while doing the exercises for a faster rehabilitation.

**Specific Objectives:**

- Grant professionals and patients access to patient information. With the development of an interface where both users can see graphs of the progress made by the patient in each exercise and see if the patient is improving.
- Testing the patient's visual difficulties. An initiation test where the focus dot is calculated, an algorithm takes as input the coordinates of the blind spots and generates a focus dot.
- Adapting the focus dot and exercises to the patient
- Allow the patient to carry out eye's exercises. Seventeen different exercises to choose from all of them with an increasing level of difficulty. These exercises have evaluation mechanisms for making progress trackers.

## Chapter 2 - Workplan

This section will explain the steps that were taken to complete the project. These include defining the problem, planning the development to solve the problem and the development itself.

### 2.1 Requirements

The following requirements were given by Leonela González Vides, the optical technician who helped us with this dissertation's optical aspects.

Requirements for Tests:

- Several initial tests and calibration tests are needed to obtain visual data, all under the supervision of the professional user.
- Manual entry of the data by the professional, including personal data and the information on blurry or blank spots. Have the option to undo or start over again.
- Test the circular grill for the right eye, left eye and both, showing points to check whether the patient sees it, fixing their sight on a central point. The patient separated from the screen at 10cm. Each finding dot will be on screen for 3 seconds.
- With the circular grid, the patient can indicate the areas in which they cannot see, specifying whether it is the right eye, left eye or both. The patient should be 10cm away from the screen. Option to undo or start over again. The test may be repeated
- The patient's focus dot coordinate is calculated and may be changed by the professional during the test.
- The test's circular grid is 10 cm in diameter with horizontal and vertical lines separated by 0,5 cm.
- Tests may be repeated to evaluate the patient's performance by the professional in charge.
- The records of the tests, results and focus obtained will be saved in the database.
- Any test may be skipped if necessary.

Requirements for Accounts:

- Users will need an account to access the application's functionalities. The authentication will be done by email and a password. If the patient user does not have an email, it will be provided with a username, and they will use that username and a password to log in instead of the email and password.
- There are two main types of accounts: Professional and Patient. A Professional account has associated several patients, and a Patient account has just one associated Professional.
- A Patient can only be created by a professional, as it is needed to add technical information about the patient's sight.

- The database will save the patient's progress, with the exercises that have been done and if the focus is activated or not.
- A Professional has access to the data of his Patients, a Professional can also do exercises, and the statistics will be saved in the Patient that has been previously chosen.
- The professional may add or edit their patient's data.

#### Technical Requirements:

- Only for tablets with Android operating systems.
- There should not be any ads.

#### Requirements for Exercises:

- The patient statistics may be displayed using graphics that show successful and unsuccessful results for each exercise.
- Exercises vary in difficulty, the first ones being the easiest.
- The focus dot is used for helping the patients locate the images.
- If the focus dot is activated, it will appear in the exercises to provide visual aid.
- Static exercises: Recognise shapes, letters and objects that appear on the screen.

### **2.1.1 Requirements for improving usability**

The following requirements were made to improve the usability of the application, for people with low sight to use comfortably.

All elements in the app allow screen readers. The macular degeneration patients are not blind; however, they may want or need to activate tools such as talkback to help them with navigation. Also, the design of the patient view has been adapted for those with poor vision by having big and separated buttons, high contrast colours and large fonts.

For helping with the use of screen readers technologies such as Talkback, NVDA or JAWS, was used the attribute "content description" in all the elements that were images that did not contain text. This attribute is what is going to be read instead of an image if a talkback is used.

Out text sizes are configured using "sp" which stands for scalable pixels[10], this makes it possible for users to change the size of the texts in the settings of their android device. If they have previously configured their tablet to use big letters the app will adapt and will function with bigger size letters.

## **2.2 Software modelling**

The Scrum framework was used to develop this Project [11]. This framework aims to create a pattern of actions that will aid in the collaborative development of a project, especially a software project. The primary feature of this framework is that it needs regular submissions that progress over time until the project is completed.

This software development methodology is recommended for complex projects with a lack of well-defined tasks, such as this one. This technique was chosen since the tasks required for the project were unknown due to the team's lack of familiarity with technologies such as Android Studio and what responsibilities should be considered.

This implementation of Scrum started with the Product Backlog. In this phase, the team was documented in Android Studio [12] and in databases compatible with Android Studio so that there could be made a better planning of the project. Once this was over, the different tasks were distributed. The Product Backlog phase had a duration of 2 months.

The following step of the framework was the Sprints, which consist of regular submissions, within an interval of one month. These submissions were shown to the tutors, and feedback was given so that it could be added to the following iteration. In this phase, there were weekly meetings where all the members talked about their progress and helped each other if needed.

The last phase of software development process consisted in testing the app on optical technicians that work with the final users and understand their needs, suggestions of changes were given, and they were rapidly implemented and uploaded so that the final users could give us further feedback.

In addition, the team planned to create meetings every time that considerable changes in the application were made by any of the members. Before the branch was merged into the develop branch, each member, or at least two of them, inspected the code of the member that introduced the changes. With the inspection, there could be suggestions, recommendations, changes, disapproval, or approval. If the team approved of the changes, the changes were approved, and the specific branch was merged into develop. In this way, it is ensured that all the changes that go through development are functional and work with the rest of the project.

## **2.3 Tasks distribution**

For distributing the tasks, the team made use of the service that the tool ClickUp provides [13]. The following states were used to distinguish the tasks: Bugs to fix, To do, To do urgent, On Progress, and Complete. ClickUp has the option to add owners to the different tasks, which was useful, as the owner received an email if another person had changed the task's state. Pair programming was used for complex tasks, arranging meetings of two people and programming simultaneously.

At the beginning the main task was the design of the application, done by all the members of the project that included discussions with the professional collaborators. Followed by a learning process and versions adaptations. Then, the tasks were distributed mainly in three teams: database and login (María y Diego), patients' evaluation (Daniela), and exercises (Alba). However, these teams were flexible, especially in common work areas (such as the database is used in both the evaluations and exercises).

Since the beginning to the end of the project, there were monthly meetings with supervisors and professional collaborators to discuss requirements and questions, as seen in the documentation given "Actas.pdf". Also, it was a constant communication through email.

In the Figure 6 Gantt Chart that shows the distribution of the different tasks, as well as the period of time in which they were done, as explained before. The application versions shown to project supervisors and professionals are marked with red lines.



## Chapter 3 - App design

Leonela González Vides, the optical technician who assisted us with the optical parts of this research, gave us the first design of the app (Figure 7 First design). This first preview of what the app would look like, was made with a PowerPoint, and included a general overview of what buttons were required and how the exercises functioned. From this first prototype, it was decided to use a tablet in horizontal view, because it was needed for the activities to have a certain length that was not realistic with small portable devices. This design was used to create a more detailed and complex one using the Balsamiq wireframing[14] tool (Figure 8 Balsamiq design). This wireframe was used as a reference during the development phase; however, it was not fully adhered to, as shown in Figure 9 App design.

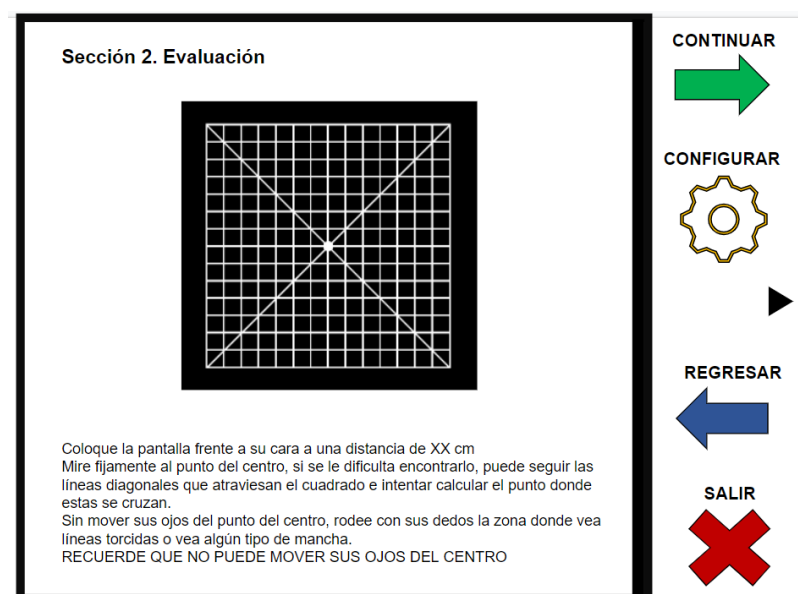


Figure 7 First design

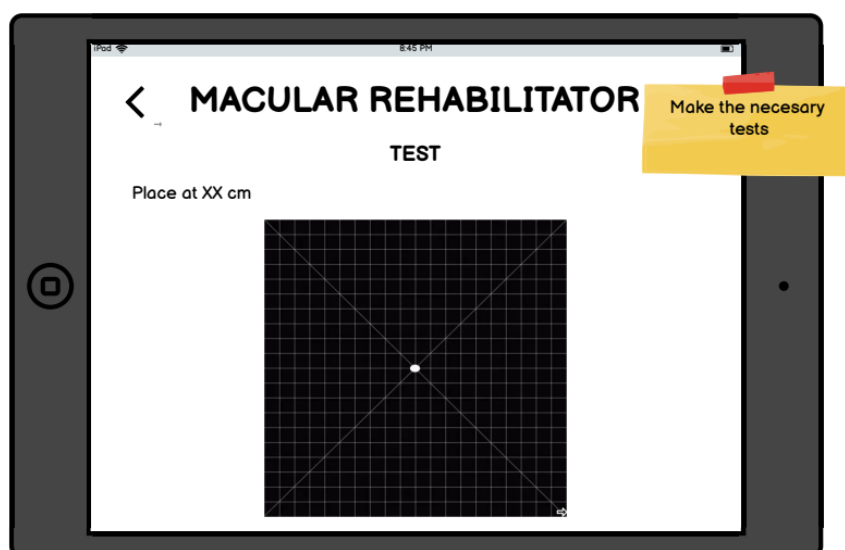


Figure 8 Balsamiq design

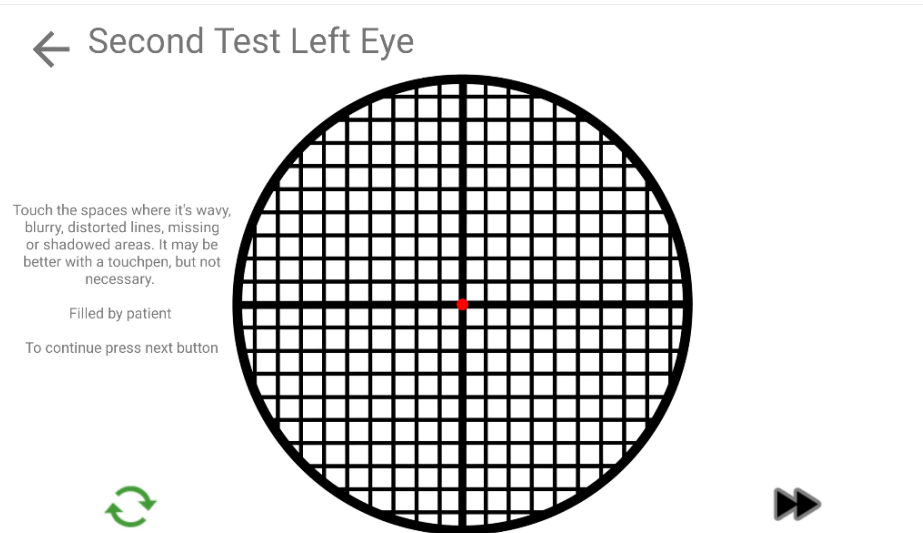


Figure 9 App design

### 3.1 ¿Why Android?

It was decided to develop the app on Android devices because they are more common than any other operating system. The data obtained by statcounter [15] shows that mobile devices with Android as the operating system in April 2022 make up 70,52% of the mobile operating system market share, and the percentage of tablets with Android as their operating system in April 2022 was 46,12%.

Breaking down the data, the application to be accessible to a wider range of users, it needs to be compatible with Android systems.

### 3.2 Use Cases

This chapter shows the differences between the two types of users. When the app is first initialised the first thing that appears is to answer if the person who uses the app is a Patient or a Professional as depending on these, the menus seen will be different as well as the authentication mechanism.

#### 3.2.1. Patient

Patients are supervised by a professional, and they can perform exercises or check their data, test history and exercises statistics. A Patient account can only be created by a professional, and they receive a numeric code to be able to sign up in the application. Because of this, a patient account cannot exist without professional supervision. The password for a Patient is the number given at the moment of registration. If a patient loses this number, he can ask the professional that created his account as he will have this number on his patient's list.

As a patient, you have two options for accessing the app. If it is the first time you enter the app, you must choose to register, where you will have to enter your email. Once you have registered, you can use the regular access which is to log in.

### **3.2.1.1 Patient Main Menu**

Follows the general design to improve the usability for people with visual impairments by using big size letters. At the top of the main menu, is the user's name. Beneath the user's name, are two main buttons. the first one is "data" to check the personal information, test history, and exercise results. The second button is to access the list of exercises. The menu has a smaller button to log out.

The colour selection was important. Most of the activities related to the patient have a distinctive separation between red and blue. If the patient is colour blind or has similar conditions, the colours will be overwritten by the ones chosen in device settings. This option is available in all androids since Android Lollipop[16].

This same menu is also displayed when accessed from the professional menu. The professional has an arrow to go back to their main menu. They can alter, delete, or create a new test in the data window, in addition to seeing the same information as the patient.

Following the same idea of big, separated buttons, the exercises menu selection enumerates seventeen exercises that the patient can make on their own or with their professional. Same does the test history and exercises history.

The main for the patient registration is big fonts and with less as possible text, maximizing the possibility of independence of the patient. It separates the process by beginning with asking for the numeric code given by the professional. This code, which is also the user password, is asked twice, to avoid mistakes. Then the patient will type their desired username. The signing in is similar to the registration, but only one page asking for the username and password. It will give messages in case of mistakes.

### **3.2.2 Professional**

Professionals supervise zero or more Patients. They can see their patient's information and their statistics as well as perform patient's exercises This last function was implemented in case a Professional want to practise some exercises with their patient in the clinic. Professionals can add new patients and add the information needed as well as performing the vision test. Once the vision test is done, the app will provide a position for the focus dot, and the professional can change its location.

*As a Professional, you have two options to start using the app registering if you lack an account or login in.*

### 3.2.2.1 Professional Main Menu

The professional main menu has access to a list of patients. The search engine situated right above can help filtering through them. As in the patient's main menu, it has a logout button. Additionally, a create patient, and a "my account" button.

### 3.2.2.2 Professional Patient

When a patient from the list is clicked, an activity identical to the patient main menu is shown. The only difference in the main menu is that the professional has a return button to go back to their home page.

Then, in the data section of the patient, the main difference is that the patient's data can be eliminated and edited. Also, in the test history, only the professional can add new tests, to ensure that the patient is tested under professional supervision.

### 3.2.2.3 Professional New Patient

The creation of a new patient consists of a form followed by a sight test that will be explained in chapter 4. The data submitted in the form will serve to the Professionals to keep track of their patients and have in hand all the data related to their vision that could be helpful to consider while exercising in the app. The specific fields were given by Leonela Gonzalez in the *informacionGeneral.pdf* attached with the dissertation.

The purpose of the form is to generate the relevant data of the patient. This data is later used by the professional to have the information needed to treat the patient. With this data, the professional can choose what exercises to recommend to the patient and make personalised decisions about the rehabilitation.

The form has two sections, the first one has the following fields:

- Date: This field is automatically filled with the actual date. The format is day/month/year hour: minutes: seconds.
- Name: This is the name that will appear at the top of the Patients menu and will be used followed by the surnames in the list of Patients of the Professional.
- Surname: Will be used by the Professional to distinguish between patients.
- Genre: The options are Masculine, Feminine and Other.
- Birthdate: The entry format does not have a specific format so that patients can just say their birth year if they struggle remembering dates.
- Ophthalmologic diagnosis: This field is for Professionals to add details about their diagnosis, the format is a string.
- Visual acuity: This is a technical field. The visual acuity (VA) is a number with decimals
- Visual field: This is a technical field. The Visual field (VF) is given in decimal numbers that represent degrees.

- Observations: A string that can be filled with extra annotation de Professional wants to remember about that patient. This is the only field that is not mandatory.

In the second section, the Professional can add extra information about the difficulties the patient may have. A checklist with all the different difficulties a Patient may have been displayed for the Professional to click if the difficulty is present:

Functional visual difficulties in kids and adults:

- Reeding
  - Signs
  - Books
  - Smartphone
  - Labels
  - Newspapers
  - Drug prescriptions
  - Faces
  - Magazines
- Writing
- Using electronic devices
- Seeing the school board (kids)
- Work difficulties
- Face recognition
- Watching TV
- Moving or playing indoors
- Moving or playing outdoors
- Moving from outdoors to indoors or vice versa
- Handling money
- Alimentation
- Housework
- Shopping
- Eating
- Personal hygiene

Visual habits:

- Makes head movements to see better
- Gets close to what they want to see, such as TV
- Prefers a specific type of lighting
- Gets bothered by the sun or the light

Use of optical and not optical help:

- Uses conventional glasses
- Uses sunglasses
- Finds the glasses helpful

Reading

- Used to read more before vision loss
- Would like to read more

Answering the last part of the form is optional and can be skipped. All the data introduced will be visible by both the Patient and the Professional that created the Patient.

### **3.2.2.4 Professional Profile**

The Professional's Profile is where all the information about the Professional account and profile is. Here the Professional can see all his data and additionally can make changes to his account and his data.

The professional's data is obtained when the professionals register into the application, where they provide their name and email, as well as set a password to their account. After registering, information can be added if the professional thinks it is necessary. The professionals can add a profile picture to identify themselves to their patients, which may result useful in cases where patients want to remember the face of their professional. Additionally, the professionals could add to their info a phone number that the patients could use to contact them.

Besides adding information to the professional's profile, the professionals can also change their data. There are implemented different buttons to change the data, where the professionals can change their name, which is going to be the name that will be shown in some of the activities like Professional's home page, in the welcome message, as well as the name that will be displayed to their patients. The professionals can also change their email address, in the case that they lost their last email, do not want to use the current email anymore or want to use another email instead. The profile picture can also be changed if there is already one in use. For the phone number the same happens, the professional can change their number if they lose the last one or want to use another phone number instead.

The professional's profile is also available to the patients, so they can access and see all the professional info, but they cannot make any changes to the data. In addition to looking at the professional's profiles, which include all the information discussed above, the patient can also contact the professional using the information left by the professional.

If a patient wants to write an email to their professional, they can use the application to automatically start an email application of their choice or one that is defined as default in the device. After the selection, the third-party email application is started with an email template which includes the professional's email and a generic subject that the patients can change if they want to. Because the patients have visual difficulties, this function was made in this way to facilitate the work of writing an email and contacting their professional. Otherwise, they would have to remember or write down the email address, then leave the application to go and find an email app, and last open the email app and write an email without a previous template.

There is also another way for a patient to contact their professional. With the professional's phone number, the patient can make a call to them from inside, without

having to write down the number, leave and search an app that they use to make calls, to finally be able to contact the professional.

The profile picture was implemented using the Java libraries mentioned in the technology used section: Image Cropper to crop an image selected, Circle Image View to convert the image cropped to a circular profile picture, Picasso to load the image from the database and Could Storage to store the image into the database.

### 3.2.3 Error proofing

In this project, there was followed some design patterns like material design, and in addition there was implemented some of Nielsen's usability heuristic principles [17].

Material Design was developed by Google as a design language that uses more layouts that are based on a grid, responsive animations, transitions, paddings, and depth effects. With Material Design, there were followed examples of how to design the interface in a way that makes it intuitive and would not become error promptly[18]. For example, the dialogues that were introduced in the application were made to reduce interruption but at the same time get the focus of the user into the error. At the same time, these dialogues may be dismissed either by tapping outside of the dialog or tapping the system back button.

Material Design also makes recommendations on how to design an alert dialog, where it does not have a title and where the answers are not only the answering the question but also it is related to the action that is going to do, as shown in the Figure 10 Material Design recommendations.

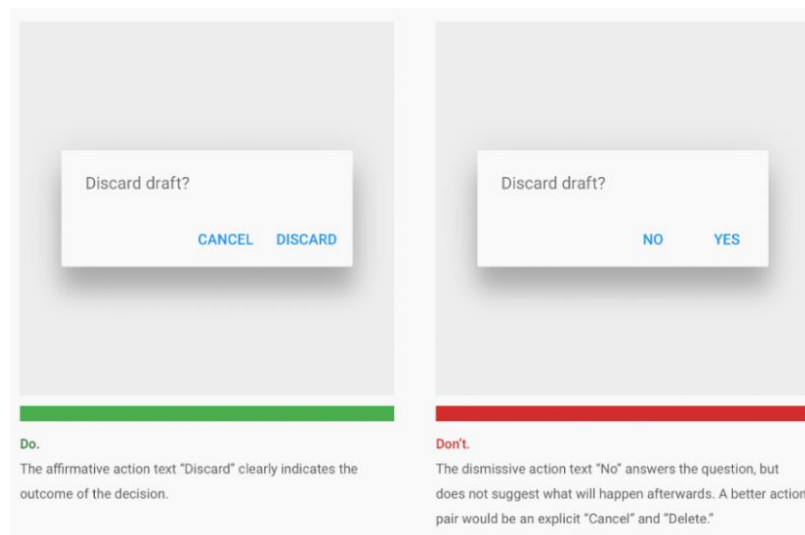


Figure 10 Material Design recommendations

The usability heuristic principles of Nielsen were as well implemented in this project. There was implemented the error prevention, so when the user is going to take any action that implies changing the data, prior to modifying the data, the application shows a confirmation message, informing the user to check the fields that

are filling up and correct because there are going to be changes. In this project, this principle could be most visible when the professional user fills in the patient data, where three activities are found that change the patient's data. When the professional ends filling up the data of one activity, prior to going on the next activity, a confirmation message is shown, here the professional has the option to not go to the next activity and check all the fields, and the option of confirming going to the next activity as shown in Figure 11 Confirm message.

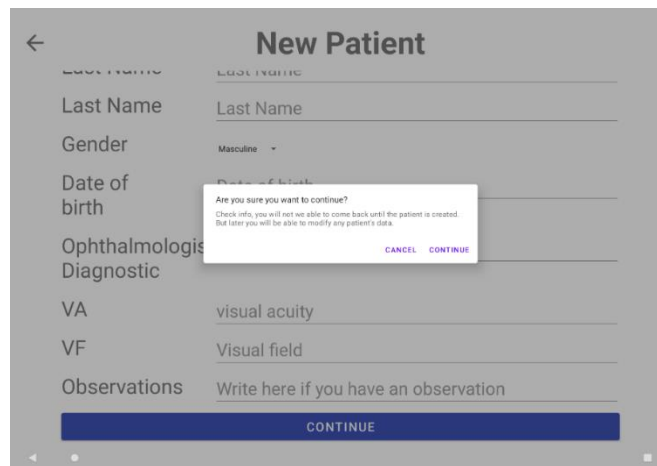


Figure 11 Confirm message

Following another Nielsen's usability heuristic principle, to help the user to recognise, diagnose, and recover from errors. The errors messages have to be explained in simple language, without mentioning error codes, and have to precisely indicate the problem, and also suggest a solution in a constructive way. Following the confirmation message discussed above: This Nielsen principle could be seen in the same activity discussed above. When a professional is done filling up the patient's data, if there is a field that was not completed by the professional, an error message appears saying that there is an empty field. Not only does this tells the user the problem, but it also gives them a constructive way of solving it, as shown in **¡Error! No se encuentra el origen de la referencia..** In addition to the error dialog, the field that has been not filled is displayed with a red icon, as well with another message alongside the icon, helping the user to find the error, in this case, the specific field that it needs to fill up, as shown in Figure Unfilled fields.

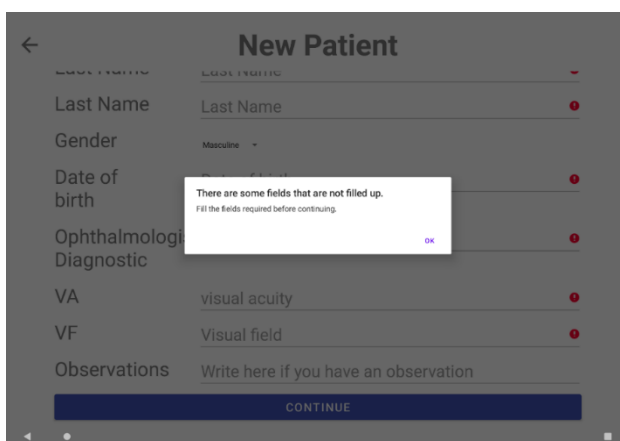


Figure 12 Error message

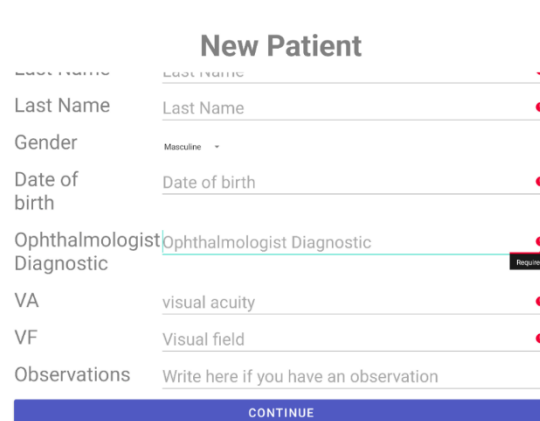


Figure 13 Unfilled fields

In the activities that are required to fill data, the app indicates with a focus the field that the user is filling up. And with the field, there is indicated with a text below if the field is required or not. For the errors, in addition to the dialog text, the square that wraps the field turns into red, the field shows a red icon, and changes the text below to a text indicating the specific error, as shown in Figure 14 Error in field.

← Change Password

Insert Old Password

Old Password !

Error: Fill Field

Insert New Password

New Password

123456 !

\* Required

Insert Again New Password

Repeat New Password

123 !

Error: passwords does not match

**CHANGE PASSWORD**

Figure 14 Error in field

### 3.3 Database

In this project, it was used for the Firebase Platform to manage, store, download and upload all the data and users' info used inside the application[19]. In total, this app uses three different but related Firebase products. Realtime Database stores all the main and most of the data essential for the app to run and work properly. Authentication, to manage users and to hold the user's metadata securely. And lastly and less critical is the Cloud Store, which is used to store larger files, especially the data that is not constantly accessed.

For managing all the data that the application needs to function, it was decided to use two different containers of information. One container is for authentication, and the other is used to store the rest of the data.

#### 3.3.1 Authentication Database

An advantage that Firebase has is the mechanism of authentication [20]. The Firebase Authentication container makes the task of implementing and managing users is made easy for the developers. This platform allows developers to implement a secure and easy to build sign-in and sign-up system for end-users. Authentication was chosen for the reason that no developer or any other person can have access to a user's password. This is because all of the systems are secured and managed internally

by Google [21]. This Authentication module allows different ways of authenticating to the application, which allows to create methods to authenticate with email-password, phone number, Google account, Facebook, Microsoft, GitHub, Apple, Twitter, Yahoo and others. This feature of having multiple ways to authenticate a user may be helpful in the future if the app is accepted by the public and grows in popularity.

The only data that is available to developers are: the email that was used by the user when registering, the session provider (email, phone, Google, Facebook, GitHub), the date in which the account was created, the date of the last time that the user had access to the database or used the application, and the user ID, as shown in Figure 15 Firebase Authentication.

identifier	providers	Creation date ↓	access date	User ID
new_patient@macularehab...	✉	11 Apr 2022	11 Apr 2022	JQsl2yOagiNBly7CLweMLmkDHiu2
pedrito@macularehabftg.c...	✉	11 Apr 2022	11 Apr 2022	Edkfh9TvtackZzX7yqsb2A3illf1
himaria@macularehabftg.c...	✉	9 apr 2022	9 apr 2022	a37MMRx1L4c1x3XahNLOirYwkfE3
pedro@macularehabftg.com	✉	9 apr 2022	9 apr 2022	eDNpqSUAr9bo4lx1NFxJANpA3Rx2

Figure 15 Firebase Authentication

The user ID attribute generated by the Authentication tool will be used to make connections between the Authentication container and both the Firebase's Realtime Database and Cloud Storage, allowing to identify which data belongs to a specific user. Also, because of Firebase's Authentication ID, not only the application can safely store the user's password but also secure Realtime and Cloud Storage databases by creating rules inside the database to restrict the data access to only those users that have logged in prior to the data request.

### 3.3.2 Real-Time Database

The database implemented in the app was built using Firebase "Realtime" Database[22]. The Firebase Realtime Database is a cloud-hosted NoSQL database with different options depending on the user's needs. In this case, it was decided on "Realtime" as it is specialised in synchronising data in real-time.

With Firebase Realtime Database, it can be stored all the data in a JSON format, which simplifies the task of adapting the data that is already managed or stored to a specific format. Before implementing the database, all the data collected inside the application was stored in JSON format in a file contained in the phone's internal storage; so, when the time comes to implement the database, the process of adapting to build and store in the database was a fluent task.

Thanks to the Realtime database, all the data can be synchronised between devices at a reasonable speed. Thanks to this, when a patient is doing exercises, the professional can watch the process in Realtime. For example, whenever a patient finished an exercise, the professional in charge could see the results in a manner of

little time. This feature proportionate that a person could change the device and get back all the information that has previously in the other device. This comes in handy when a patient is going to visit a professional. If the patient does not bring with him his tablet, he could use the professional's devices or any other device that the patient could reach for.

One of the most significant advantages of the Realtime database is that it is cloud-based. This is one feature that really helps people that are beginning a big project because, with the cloud-based database, all the data does not have to be stored on a server. This is thanks to the Realtime having an SDK built-in that it is not necessary to build or maintain a server. This is because Realtime is built with Firebase's Cloud Functions. Cloud Functions helps run back-end code without servers, such as testing and running database changes, registering a user with Authentication, and checking the security rules.

The Realtime database, although it seems like it can only work with good internet connections to enable the fast real-time that the product appears to offer, it also has an excellent optimisation for when the user is offline or with no internet. In this project it was used one primary strategy, first, when the data is modified, the changes are stored in the device's internal storage, to make sure the information is never lost anytime or under any conditions, and later, with the help of this feature, when the device comes online, it can upload all the data with the device's internal storage, or also with the use of local cache on the device. With this, when the device is with poor connection, or even without connection, the application can still run, store the data secured, and all these without losing any data or features.

With the Firebase tool, the data can be secured, using user-based security. Can work together with Firebase's Authentication tool, to secure all the data stored in the Realtime database, so no one with no proper authentication can access the information.

The Real-Time container is used for storing all the information about the patient's health and improvements, as well as being the connection between a professional and his patients. The data in this database is organised using JSON objects.

The database's primary key is "macularehab-default-rtdb", which contains the value of the other five keys: NumberOfProfessionals, Patient, PatientsNumericCodes, and Professional.

**NumberOfProfessionals:** This key-value pair represents the number of professionals that have made an account on the application. This number starts at 100.

**Patient:**

- **Key:** The authentication number given by Firebase when the patient account is created.
- **Value:**
  - **PatientNumericCode:** Is the code for the patient to get access to the application.

- **ProfessionalUID:** is the professional identification number that helps find the professional in charge of the patient.

**PatientsNumericCodes:** Here the numeric codes generated by the professional are stored. It only stores the codes that have been generated and that are not yet used by a patient to create an account. This helps to find if the code introduced by the patient during registration is correct, that this code was undoubtedly generated by a professional, and that it was already not being used by any other patient.

**Professional:** it lists the professional's unique identifier given by the authenticator when creating the account. Here all the information, either professional or patient info is stored. In each of these keys:

- **name:** Given name of the professional
- **numberOfPatients:** this number is used to create the patient's numeric code, which helps identify the professional's patients. This field is similar to NumberOfProfessionals and also starts in 100.
- **ProfessionalNumericCode:** is the value obtained from the NumberOfProfessionals value. This value is used to generate the patient's numeric code.
- **uid:** is the user's unique identifier generated by the Firebase Authenticator.
- **contact\_info:** professional's contact information.
  - **email\_address:** professional's email address.
  - **phone\_number:** professional's phone number.
- **patients:** List of professional's patients. Each patient is identified using the numeric code generated by the professional.
  - **Key:** The numeric code. This numeric code is constructed using: The first three numbers are the identification number of the professional. Because of this, each professional's patient starts with the same three numbers. The following three numbers are made by the number of patients a professional has. The last two numbers are generated randomly. These random numbers were created to reduce the probability of mistaking users' IDs and avoiding using another user's ID.
  - **Values:**
    - **tests:** list of the test realised to the patient with each key made from the date on which the test took place. Then for each test, there are: the focus calculated when the test was performed and the results of each test with the coordinates that fill the stain listed.
    - **av:** patient's visual acuity.
    - **cv:** patient's visual field.
    - **checkBox:** a list of Booleans values of each checkbox filled out in the patient's form by the professional. Each checkbox is a patient difficulty.
    - **date:** the date of when the patient's data and the account were created.
    - **date\_of\_birth:** patient's date of birth.
    - **diagnostic:** patient's diagnosis, which is most likely to be macular degeneration associated with age, but it could include other diagnoses if the patients suffer from something else.

- exercises:
  - **exercisesInfoList:** contains a list of exercises. Each exercise has:
    - **times\_completed:** the number of times an exercise was performed.
    - **resultsList:** contains a list of each exercise attempt. Each attempt contains:
      - **counterCorrect:** corrects obtained in the exercise.
      - **counterFailed:** failed results in the exercise.
  - **exercises\_acomplished:** the number of exercises that the patient has completed, namely, the number of exercises with more correct results than failed.
- **focus:** contains two values, "first" indicating the x coordinate of the focus dot and "second" indicating the y coordinate of the focus dot.
- **focusIsOn:** indicates if the focus was on or off last time.
- **focus\_size:** the size of the focus, is a decimal number between 1 and 2.
- **gender:** indicates masculine, feminine, or not indicated
- **hasAccount:** indicates if the patient created an account.
- **date\_last\_test:** date of the last performed test.
- **name:** patient's first name.
- **first\_lastName:** patient's first last name.
- **second\_lastName:** patient's second last name.
- **observations:** comments from the professional about the patient.
- **patient\_numeric\_code:** numeric id created to identify a patient.
- **patient\_uid:** unique authentication id of the patient's account. This field is generated when a patient account is created, and the number is generated automatically by the Firebase Authentication tool.
- **patient\_username:** The patient's account username or e-mail. This field is also created when a patient registers into the application.
- **professional\_name:** patient's professional name
- **professional\_uid:** professional's authentication id. This id is generated by the Firebase Authentication tool.

### 3.3 Cloud Storage

Last and least, Clouds Storage is mainly used to store larger files which are not always needed and are least important than the primary data of the patients and professionals.

Firebase's Cloud Storage is an object storage service with the capabilities and scale of Google's Cloud [23]. In this project, Cloud Storage was used to store heavy

user-generated data that exceeds the capabilities and functionality of Firebase's Realtime database that was discussed before.

Cloud Storage has the capability of robust operations, where it can perform uploads and downloads of objects regardless of the user's internet connection and network strength. These robust operations mean that both downloads and uploads can stop if the application loses internet connection and restart the process when the internet is re-established; also, it behaves depending on the network strength, bandwidth and type, saving the users time and cost.

Firebase also worries about the security of Cloud Storage. Because of this they integrated Cloud Storage with Firebase Authentication. With Authentication, the project was programmed and declared with the database's rules, so the data in Cloud Storage can only be accessed with the users that authenticated before. These rules can grow and reach other types of rules, to grant access not only by Authentication but also depending on the filename, the size of the file, the content type, and other types of metadata.

In this project, Firebase's Cloud Storage is used to store the user-generated profile's photos, because there are larger files, much more than the main data, and are not always accessed. These files are not always accessed or created because it is not required for a professional to upload a photo. Even if a professional uploads a photo to his profile, that photo is not frequently downloaded or needed. In conclusion, the files used are heavier and less critical than the other data managed in the application. For this reason, this data is kept in a different location and not in the Realtime Database.

## Chapter 4 - Patient evaluation

When a Professional adds a new patient, a first evaluation may be solved after indicating personal data. These tests are meant to be under professional supervision. And can only be done from the professional's account. Each test can be skipped if the professional considers it appropriate. There is previous information about the stain that can be uploaded and two different evaluations to determine the blind spot of the patient. Both the upload and the tests are repeated for the left eye, the right eye and both eyes.

Each array of coordinates is saved in the database by the date of creation, since the test can be repeated through the sessions to see the progress. The final coordinate to the focus dot is also saved and later used in the exercises. The focus test can be seen in the resume stain in the test's history.

There is a special process for smaller devices. If the tablet is under 10 cm, the tests won't be able to be taken. However, the manual input from the professional may be filled with a smaller circle that adapts to the screen size. As other activities have layouts to screen sizes under 700dp.

### 4.1 Drawing Dots in the correct coordinates

The circles are drawn using "metric units", which is the size of each square in the grid. Since objects on the screen are measured in pixels, and it is needed in centimetres, it is used a function of "pixels per inches", and from that it is transformed. Also, the measure must be a natural number, a pixel cannot be partially used. In a tablet with a width of 10 or more centimetres, the metric unit is half a centimetre, and the size of the circle is calculated as twenty times the metric unit: [24]

$$\text{metric unit} = [\text{pixels per inches} * 0.19685]$$

$$\text{size} = \text{metric unit} * 20$$

For devices with a width under 10 centimetres (only the manual input can be filled), the metric unit will be the width of the screen divided by 20, so the size of the circle is the width of the screen. The metric unit is floor rounded so the circle does not go off-screen. The size of the circle is calculated from the metric unit to avoid small variations from round numbers that may dealignment elements in the screen.

$$\text{metric unit} = \left\lfloor \frac{\text{screen size in pixels}}{20} \right\rfloor$$

$$\text{size} = \text{metric unit} * 20$$

For all the evaluations there are one or several dots in which the patient's vision is evaluated. Each dot must appear in the spaces between the lines in the grid. Knowing that the circle formula in Cartesian coordinates is  $x^2 + y^2 = r^2$  [25] and the grid is seen as a square grid and each space is a coordinate in relation to the centre, then if the previously mentioned coordinates of the square grid are limited by the

circle (Figure 17 Grid example), for each  $(x, y)$  coordinates and radius  $r$  (in this case 10)  $x^2 + y^2 \leq r^2$ . [26]

However, these coordinates would be in the line's intersections instead of the empty space. So, the positions within the screen are moved half a metric unit to the centre (if the coordinate is negative, it will be by addition, and if the coordinate is positive, it will be by subtraction) and neither  $x$  nor  $y$  can be zero.

The coordinates saved in the database are relative coordinates, with no relation to the screen size or pixels, as shown in Figure 16 Relative coordinates.

$$x \text{ coordinate} = \text{circle centre} + (\text{metric unit} * \text{relative } x \text{ coordinate}) \pm \frac{\text{metric unit}}{2}$$

**Observation:** In Android Studio, the coordinates system starts in the top left corner of images, so the coordinates in relation to the centre are opposites to a common Cartesian coordinate system for the  $y$ -axes. So,  $x$  negative values are on the left and  $y$  negative on the top. Same with coordinates from the screen. [27]

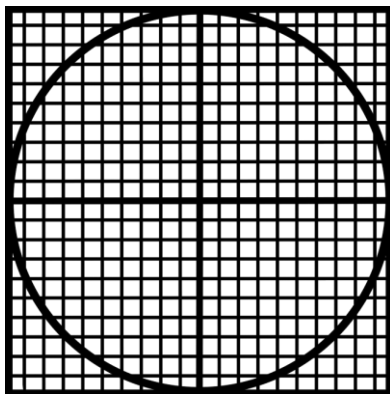


Figure 17 Grid example

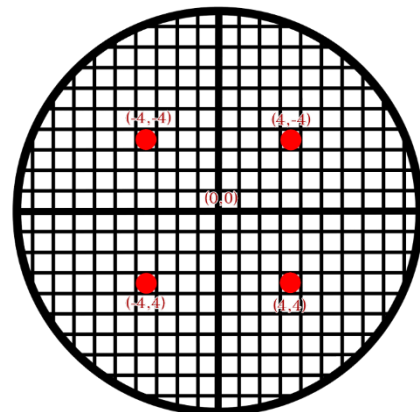


Figure 16 Relative coordinates

## 4.2 Manual input from professional

There are tools that professionals use to measure the blind spot of the Macular Degeneration patient. This information can be uploaded through the grided circle. When the screen is touched, an event is risen giving the coordinates within the screen. [28]

These coordinates are compared to possible coordinates in the grid. In the event that the coordinates are within a square in the grid, a dot will appear or disappear. Therefore, each valid coordinate is given a maximum and minimum  $x$  coordinate, the same happens with the  $y$  coordinate.

$x$  and  $y$ : coordinates from event.

$x_{min}$  and  $x_{max}$ : coordinates limiting valid coordinate in the  $x$ -axes to a specific valid point.

$y_{min}$  and  $y_{max}$ : coordinates limiting valid coordinates in the  $y$ -axes to a specific valid point.

$x_{min} \leq x \leq x_{max}$  and  $y_{min} \leq y \leq y_{max}$  then the dot related to  $x_{min}$ ,  $x_{max}$ ,  $y_{min}$  and  $y_{max}$  is activated or deactivated.

If the  $x$  and  $y$  coordinates do not match with any valid coordinates, then no dots are activated or deactivated.

The limiting value for each coordinate is calculated from where the dots are drawn, those are referenced to the centre of the square in the grid. Then the maximums and minimum are added or subtracted by half a metric unit:

$$x_{min} = x \text{ coordinate} - \frac{\text{metric unit}}{2}$$

$$x_{max} = x \text{ coordinate} + \frac{\text{metric unit}}{2}$$

There is a repeat button to start over with a clear grid.

### 4.3 First Test

Before each test comes an explanation obtained from the professional Leonela González:

1. Press anywhere on the screen when the dot can be seen. This helps the patient to concentrate on the activity without having to find a button to press and may be difficult to see.
2. Cover gently, without pressure, the left, right or non-eyes.
3. Place the screen at 15 cm.
4. Look at the centre dot and do not move the head during the test.

The test consists of a central dot that blinks three times then a second dot that also blinks three times. The blinking helps drag the attention back to the centre and to then find the dot. This will happen repeatedly until the second dot appears on all the coordinates. The centre dot is to catch the attention of the patient and the second one is to be found by the patient. The latest dot follows the valid coordinates within the circular grid, from left to right and up to down, the order does not matter. In the end, the test may be repeated.

The blinking program uses Java.lang library Timers, it is mainly used with a total time and tick interval [29]. Every 0.5 seconds the dot (either the centre or the finding one) will appear or disappear. Then, after 3 seconds (and after appearing and disappearing three times) it will start a certain action. When the centre dot has finished blinking it will stay visible, the coordinate of the finding dot will change and will start blinking. After finding dot time is up, it will disappear and initiate the centre dot blinking. This occurs unless it is the final dot, in that case the test will end and show the buttons to start over or continue.

This test takes between 17 and 28 minutes, depending on how fast the patient finds the dot.

## 4.4 Second Test

Identical to the manual input from the professional, the only difference is that it is filled by the patient to indicate distortions in the grid. Before each test it has instructions like the first test, also given by Leonela González (can be seen in document *IndicacionesActividadesEvaluacion.pdf*):

1. Touch the spaces where it is wavy, blurry, distorted lines, missing or shadowed areas.
2. Cover gently, without pressure, the left, right or non-eyes.
3. Place the screen at 15 cm.

Same as the manual input from the professional, the first touch within a valid coordinate activates the dot and a second touch deactivates it. There is a repeat button to start over.

This test was originally mended to be done with an Amsler Grid, but to help the data comparison with the first test and the professional input, it was changed to be also in the circular grid with Leonela González approval.

## 4.5 Results from the tests

After a debrief of the result of each test, it is calculated as a general stain, taking common and uncommon dots. The focus dot is obtained by the means of the x and y coordinates, respectively, from all the different tests, then the repeated coordinates weight more. The more a coordinate appears in different tests, the focus dot is moved more closer to that coordinate. The focus dot does not need to be a "valid coordinate", meaning it can go anywhere within the grided circle.

The focus is seen as in Figure 18 Calculated Focus, the blue dot (focus dot) is placed in the middle of the red dots (results from tests), which means that the dots appeared uniformly in several tests in those coordinates. However, if any coordinate appears more often than others, the focus dot will be moved to the more repeated dots, showed in Figure 19 Weighted calculated focus, in which the focus dot is at the intersection of two lines. Same resume coordinates from the tests, different focus dot.

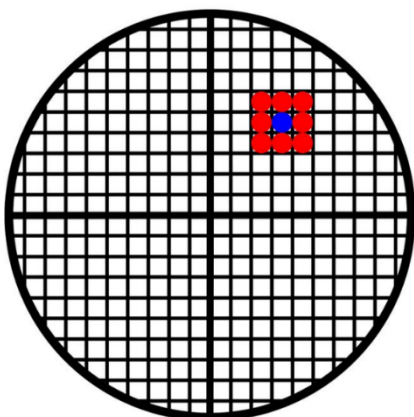


Figure 18 Calculated Focus

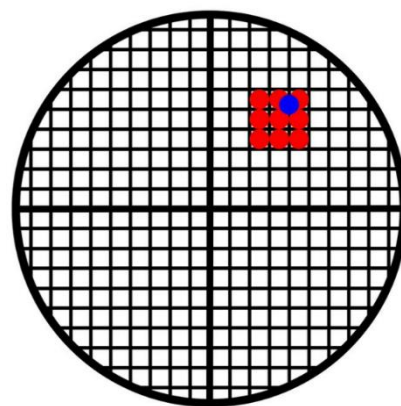


Figure 19 Weighted calculated focus

If the professional disagrees with the resulting focus dot, it may be moved wherever the professional considers appropriate (within the circle). The coordinates can be reset to the initial calculated value.

## Chapter 5 - Exercises

Patients and professionals will have seventeen exercises available. They are ordered in increasing levels of difficulty. Also, all of them are customisable: which will also increase or decrease the difficulty of a specific exercise depending on the patients' needs, which can change not only depending on the patient, but on the needs the same patient may have that day.

From exercises 2 to 13, the time between shapes, letters, images or instruction can be decided before the exercise starts. In the exercises 14 to 17, the time that the whole exercise lasts can be decided before the exercise starts. In all cases, the EditText is already filled with the recommended time, so the user does not need to fill anything, but they can if they prefer a shorter or longer period, to increase or decrease the difficulty of the exercise. It has error proofing: in case the user deletes the default time and does not fill the new time before playing the exercise, the default time will be set automatically.

For the exercises 2 to 10, if at the start of the exercise the focus is on, they will have 5 seconds to find the red dot before anything else appears on the screen. In exercises 11 to 13, the background image appears but the buttons and instructions are not touchable or invisible before the 5 seconds to find the focus point end. From the 14th to the 17th, the patient has all images/figures on the screen from the beginning and they must find specific objects within the selected time.

Each exercise can be paused, and it will display a menu where the patient could go back to the exercises list or turn on or off the focus.

Just after an exercise is completed, the result is written in the database (where the number of correct and wrong answers, and the attempt number were saved) and a view with the results comes up so the user can easily see the number of correct and wrong answers they had.

The red focus dots are used in the exercises with the purpose of teaching the patient where they should be looking at to find what is requested. In this way, the patient is taught where their reference point is, so that after practising enough they might be able to be looking in the correct position without the help of the red dot.

The focus dot maintains the proportions set in the tests. The conceptual grid of 10 cm of diameter is kept and the centre of the grid is considered as the centre of the figure, letter, or anything that the patient needs to find. The distance and location of the focus dot are set with coordinates in relation to the centre. If the device's width is lower than 10cm, the position of the focus point is adjusted proportionally to the screen size.

As some patients might not have a good enough visual capacity to read, or they may not be able to read that day because of other factors (tiredness, bad light, deterioration of their illness, among others), the talkback feature is available for all the exercises. The action of touching a button will require a double click if talkback is activated and a singular click otherwise. All texts are readable by touching them a single time (including the texts that changes in exercises 11, 12 and 13).

All the restrictions of the exercises have been decided in order to satisfy the requirements, design and instructions of the expert Leonela González Vides. The document used as a reference is attached with the name "Actividades\_solo\_VE.pdf".

## **5.1 First exercise**

Consists of demonstrating the general use of the exercises. Its only purpose is to show the functionality of the other exercises. It does not have a training goal like the following ones.

## **5.2 Second exercise**

It consists of finding the correct shape in a sequence of shapes. As they are static exercises and this is the first training exercise, all the shapes appear in the centre of the screen, they have a size of 3cm in diameter and they are filled in black. They appear following a specified order to make sure each shape appears the number of times and in the order desired. The patient is requested to touch the triangles.

The focus point, if it is on, will appear on its corresponding position taking into account that the centre is the shape. If the device's width is lower than 10cm, the size of the shapes will decrease proportionally to the screen size.

The user can change the default time of seconds in between shapes. In addition, in order to make the exercise more dynamic, once the button (which contains the shape) is pressed, it directly changes to the next shape without waiting for the time left to finish. This method is implemented with Timers (which have been mentioned in Chapter 4) and it is applied in all the following exercises. A fast response, as it is the instantly change of a shape when is touched, satisfies the Feedback Principle [30].

At the end of the exercise, the result is recorded in the database and a screen with the results appears. For this is needed to calculate the correct and wrong answers in the following way: whenever the correct shape appears and is pressed (in this case, the triangle), one correct point is counted and if it is not touched, one incorrect point is counted. When other shapes come out, if the user does not touch them is also counted as a correct point, and if they do, it is counted as a negative point. It is done this way in order to motivate the patient to improve, because as it is the beginning of the rehabilitation, the user may need some encouragement and if they fail all the exercises, they might stop practising. Despite this, the professional will know that if they have any wrong answers, it is because they have not touched a shape when they should have or they touched one when they should not have, so they will know that their patient still needs to improve.

This is also applied across all the exercises from the second to the tenth. From the eleventh onwards the counter of correct and wrong answers is not so generous because the user already has had enough practice and having harder levels might help them motivate to be more concentrated on the exercises.

### **5.3 Third exercise**

It is equivalent to the second exercise, but the shapes are half of the size of the previous exercise and they are not filled. Also, the total number of shapes and the number of different shapes they appear is greater in order to make this exercise harder.

### **5.4 Fourth exercise**

The fourth exercise consists of a series of letters in upper case and bold that appear in a specific order. The goal is to find the letter required (in this case, the letter E).

### **5.5 Fifth exercise**

This exercise follows the same structure as the previous exercise but is one step harder. The size of the letters is smaller and they are not bold.

### **5.6 Sixth exercise**

The sixth exercise has two shapes distanced from each other. Each shape has its own focus, and each focus is calculated having a reference to the centre of each shape. As in exercise one, they have 3cm in diameter and are filled in black, as the expert required. The user is requested to touch the circles.

This time they do not follow a specified sequence, the shapes appear in random order, but they have a condition that does not allow them to repeat the same shape on the same button consecutively. This logic is applied to all the exercises after this one, with the purpose of making the exercises more fun and less repetitive for the users, as every time they repeat the exercise, it will be different.

In addition, each shape has its own counter, so if they click (or double click in case they have the talkback on) before the time for that shape has expired, the shape touched will change into the next one, but the other shape will not. This is controlled with different Timers for each button, apart from the Timer for the focus point (in case it is on).

### **5.7 Seventh exercise**

It is equivalent to the sixth exercise, but the shapes are half the size of the ones in the previous exercise and they are not filled. Also, the total number of shapes and the number of different shapes they appear is greater in order to make this exercise harder.

## 5.8 Eighth exercise

The eighth exercise is a mix between the sixth and fourth. They appear two letters on the screen, that change independently (using distinct counters to control each letter). The user must find and touch the M's.

## 5.9 Ninth exercise

Same as the eighth exercise, but with a smaller font and not bold.

## 5.10 Tenth exercise

This exercise adds everyday objects to the set of exercises the patient can practice with. From a selection of different drinks, the patient is instructed to find the coffees. The number of times each drink appears is random but there is a condition put in place to ensure that the visual of a coffee appears at least once and that the same object cannot appear consecutively. As the total number of objects that appear is 12, the number of coffees can appear is between 1 and 6. Like what was done with the previous exercises that featured shapes and letters, not touching an incorrect object grants the patient one correct point. Therefore, the maximum number of correct answers the user can achieve is equal to the total number of objects, which is 12.

## 5.11 Eleventh exercise

In the eleventh exercise, the patient needs to find a given part of a woman's face. The possible parts are:

- Left eye.
- Right eye.
- Nose.
- Mouth.

A text appears at the top of the image indicating the part of the face that they should find, and, as recommended by the expert María Guadalupe González Montero, a recording saying what they must find out loud was included, because, even with the use of the talkback, it was too difficult for the patients to succeed in this exercise without this feature as they needed to look at the text and then find the red dot (if it was on) to finally find the specified part. With the audio saying out loud the part they should touch, the patient can keep the looking at the image all the time, without being distracted by having to look at the top of the screen to read the text. The recordings were included with the Java.lang library MediaPlayer [31].

In this exercise a total number of 10 instructions appear (consisting of the parts of the body to find), and the number of times they each appear is random, with the condition of not having the same instruction be displayed consecutively. If the red dot

is on, it moves every time the instruction changes so that it is easier for the patient to find the part of the body they are looking for. This exercise uses the centre of the corresponding part as the centre of the imaginary grid.

As this exercise is more complicated than the previous ones, the default time is longer than it was prior, and it can be easily adjusted to suit the patient's needs before the exercise starts.

In this exercise the difficulty is also increased in respect to how the result is calculated. A correct answer is only recorded when they correctly touch the specified part within the allocated time. A failure will be pressing the wrong part or not finding it in time. This method is the same for twelfth and thirteenth exercises.

The implementation of this exercise was challenging, as the user needs to find the parts of the body, and there is a button in each part. The buttons needed to be invisible but clickable, so the solution was setting the value alpha to 0. The problem with this strategy was that when the talkback on, the exercise was almost impossible to solve correctly. The solution to this was changing the value of alpha to 0.01, so they were still invisible but easily selected with the talkback on. This problem is solved using the same method in the next two exercises.

## **5.12 Twelfth exercise**

In the eleventh exercise, the patient needs to find a given part of a man's face, and it is equivalent to the eleventh exercise.

## **5.13 Thirteenth exercise**

The thirteenth consist in a picture of a living room and the patient needs to find a series of indicated objects, which can be any from:

- Left window.
- Right window.
- Lamp.
- Sofa.
- Carpet.
- Plant.
- Pot.

The patient has the given time to find each object. If focus is on, it will appear centred in the current finding object. A failure may be pressing the wrong object or not finding it on time.

The thirteenth exercise has a more heterogeneous difficulty than the eleventh and twelfth exercises which are similar exercises. The reason for this is that the size of the objects is not equal across the image, which makes the buttons that identify the object also not equal. In conclusion, this makes some objects easy to find and click, and some objects are the opposite, which are harder to find and to click.

## 5.14 Fourteenth exercise

In the fourteenth exercise, the patient needs to find the red crosses that are around six black unfilled circles of different sizes. Two circles are big, two are medium and two are small. Every time they click on a cross it turns green, to help them distinguish easily crosses that remain and to satisfy the Feedback Principle. The exercise will finish once the patient finds all the crosses or when the time given has expired.

The time from now on it is not set for each shape, letter or instruction, but rather for the whole exercise. The default time is set to 60 seconds, and it can be easily changed before the exercise starts, as with the previous exercises.

In the fourteenth exercise, the patient needs to find the red crosses that are around six black unfilled circles of different sizes. Two circles are big, two are medium and two are small. Every time they click on a cross it turns green, to help them distinguish easily crosses that remain and to satisfy the Feedback Principle. The exercise will finish once the patient finds all the crosses or when the time given has expired.

Correct answers are the number of crosses pressed (with a maximum of 24), and failures are a count of the remaining crosses. This exercise does not have a focus dot due to instruction from the expert Leonela González Vides.

## 5.15 Fifteenth exercise

The fifteenth exercise has the same layout as the fourteenth. By fixing the sight in one of the crosses, the patient has to press on the circles. To help the patient, the circles will be set to green after they have been pressed. It will end once all the circles have been found or the given time to complete the exercise is over.

The correct and incorrect answers is calculated analogously to the previous exercise but counting the circles that have or have not been touched, with the maximum number of correct and incorrect answers being 6.

## 5.16 Sixteenth exercise

The fifteenth exercise is based on finding certain letters in three sentences. In the first line, the patient must find the letter n in a sequence of m's and n's. The second part is finding o's between u's and c's. In the third they need to find v's in a sequence of v's, w's, and y's. Once a letter has been found, its colour will change to green, to fulfil the Feedback Principle. The correct letters can be pressed only once. However, the same wrong letter can be pressed an indefinite amount of times and each time it will be counted as a different mistake. Therefore, they may have a maximum of 22 correct answers and up to an infinite number of incorrect responses as it is calculated using the total of the correct letters left unfound and the total number of mistakes. In

this and the next exercise the points are calculated this way because as they are the last exercises, they are the most demanding, and patients that have made it to this far have already practiced enough in the previous exercises and might need a greater challenge.

The exercise will finish either when the patient finds all the specified letters or the given time is over. The default time is greater because, as explained before, this exercise is very demanding when it comes to calculating the final score.

This exercise does not have any focus dot. Instead, it has two red guiding lines on the top and bottom of each line which cannot be disabled, to fulfil the requirements of the expert Leonela González, the same situation occurs on the next exercise.

### **5.17 Seventeenth exercise**

The seventeenth exercise is similar to the sixteenth. The patient must find the same letters, but this time in larger lines. The way of giving feedback is different, instead of changing the colour of the letters, the whole rectangle around the letter changes to a translucent green, so that the user can still see the letter behind it, which keeps black.

In total they are 27 correct letters, therefore, the user can have up to 27 correct answers and up to an infinite number of incorrect answers.

The distance between letters and the size of the letters is very small, and this exercise contains more text than the sixteenth, making it the most difficult. Besides this, the calculation of correct and incorrect answer is very demanding, as explained on exercise sixteen, so the user has a challenge. To help them with this, the correct buttons are proportionally bigger than the wrong ones to give them an advantage and not make them feel discouraged.

### **5.18 Pause**

The pause functionality is implemented in all exercises. It gives the patient the opportunity to stop the exercise and then return to it. All exercises use timers, one way or another. The pause consists in taking the remaining time in the timer and stopping it. The remaining time is collected in every tick of the timer. Then when it is restarted the exercise, a new timer is set with the remaining time.

Also, when pause, the menu is displayed which includes the return to the exercise button. In this menu, it is contained a home button to return to the selection of exercises. There is also a settings buttons which sends the user to the settings windows where they may adjust the focus dot.

From exercise 14 to 17, there is no focus dot. However, the patient still has access to the pause and setting buttons.

## 5.19 Focus dot settings

The focus dot settings may be changed in the patient main menu (from patient or professional home page), in each exercise description and in the pause menu of the exercises. The view is as shown in Figure 20 Dots setting.

As the expert Guadalupe González requested, a setting to change dot size was added. The size is between the original size of the unit and the double it. It has a simplified interface of two buttons, one to increase the size and the other to decrease it. Using the same “draw dot” function used to represent the focus dot in the exercises to show the size, as it changes by pressing the buttons. From this function the size is determined and, instead of the database coordinate, it is used as the central coordinate (0,0).

Finally, there is the activate/deactivate focus switch. When activated it will appear in the exercises as visual reference as explained before.

The setting button icon was taken from Flaticon.

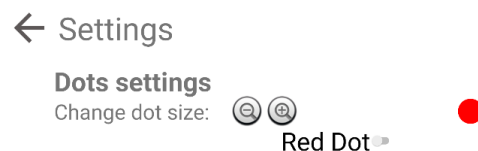


Figure 20 Dots setting

## 5.20 Charts information

The application stores all the information regarding the results of the activities that the patient performs over time in the database. Each time the patient finishes an exercise is saved the number of correct answers and the number of incorrect answers that the patient got. In this project, the data regarding the patient's results were chosen to be shown by a chart instead of showing a list of each result, as shown in Figure 21 Exercise chart. The reason for this choice of implementations was to improve the data visualisation, speeding the time for the professional to read the patient's result and facilitating the visualisation for the patients since the patients have more difficulty seeing text.

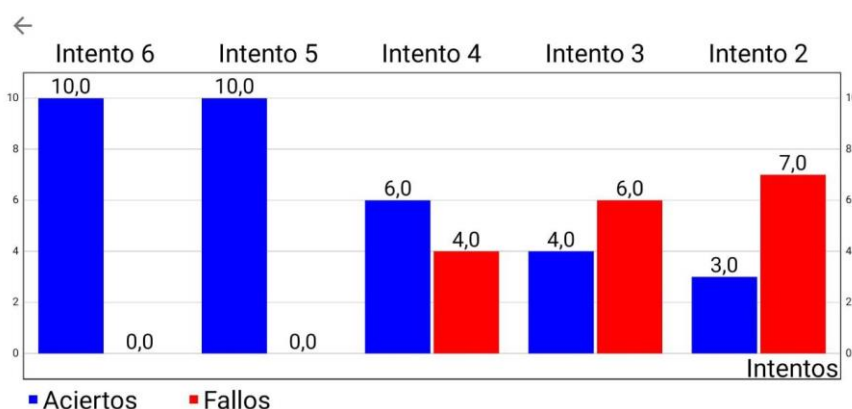


Figure 21 Exercise chart

To implement this feature into the application, there was used the Java library MPAndroidChart mentioned in the technology used section. First of all, is filling up the

data that is retrieved from the database. Then the data is separated into two datasets, the first one is the correct answers and the second one is the wrong answers. Later is to choose the colours of each bar representing the data, and the size of the bars and texts. For the colours, it was decided to follow the Nielsen usability principle [29], where the colours match the already ones that could be seen in the application, blue for correct answers and red for the incorrect ones. This colour distinction is told by a description that specifies the colour representation. With each bar, there is also a number indicating how many there are in the bar, so the user does not have to search what is the number that represents the bar. In addition, at the top of every group of two bars is located a label describing the number of attempts the group is. There is also a maximum number of results that are shown at a time, which was chosen to be five results, so the user is not overloaded with information, but at the same time, the user can navigate through the chart to search for more and older results. Also, the chart allows to zoom in and zoom out, and move horizontally or vertically; all these options were implemented in the case the user is more comfortable with a different size or view, and in this way be unable to see better the chart.

## Chapter 6 - Results

The first version of the application consisted of just the basic structure, user authentication and a straightforward exercise. The purpose of this version was to show the tutors of this project the main menus and the database, so they could check if the implementation was going in the right direction. This version was shown on the 21 of February 2022. The feedback received was positive, and the only suggestion made was to change the exercise because at the beginning of the implementation, a dynamic exercise was made, but these kinds of exercises fall outside the project's scope (static exercises).

The second version was shown to the project tutors and María Guadalupe Gonzalez. The first version was shown on the 24th of March 2022. The changes needed were the following:

- Graphs for showing the progress made by the Patient.
- Changing the exercise descriptions so that they're more understandable.
- Fixing the translation errors.

The third version was shown on the 21st of April 2022. To the tutors, María Guadalupe Gonzalez, and David Peña Quineche. This version had the previous changes implemented and the following new requirements were added:

- Fixing layout disarrangements.
- Making the app compatible with small tablets.
- The manual introduction test should adapt to the screen.
- Fixing exercises 16 and 17 so they fit into small tablets.
- Fixing the database to make it compatible with old Androids.
- Making the app work offline, with the option to upload progress.

The project was last tested with optical specialists on the 9th of May. This version had all the suggestions of the last meeting implemented. They gave us a list of recommendations and things that should be changed:

- The distance recommended to do the exercises should be said at the beginning of the exercises.
- The focus dot should be customisable in size.
- The default time for each exercise should be set to 5 seconds.
- The instructions to do the exercises should be said out loud automatically.
- Make cups of tea appear compulsory at least once in exercise 10.

These last changes were all implemented. A new executable was generated and given back to the app testers, they responded with positive feedback and no more suggestions.

## 6.1 Technologies used

**Android Studio:** Android Studio is an Integrated Development Environment (IDE) for building apps in Android's operating system apps. It was implemented for both the development and testing of the application.[12]

**Mobile tablets:** They were used to get a more realistic view of how the app would look like, as the Android Studio device emulator has a limited number of devices to try the app on. So, to get a better idea of how the app would look, it had to be tested the APK on several devices with different proportions and different APIs.

**Balsamiq:** Balsamiq is a wireframing tool that helped us create a visual structure for the app to use as a guideline during development.[14]

**Firebase:** Firebase is a platform that helps develop applications. Firebase works for Android but also for iOS and Web development. In this project, Firebase was used to build the database, authenticate users and store their data on Google's cloud server. [19]

**Gson:** Gson is a Java serialization/deserialization library that helps convert Java Objects into their JSON representation and from a JSON string into an equivalent Java Object [32]. In this project, Gson was used to simplify the task of passing objects information in the intents between activities, where the Java Object is converted into a JSON string from the source activity and then the JSON string is converted back into the Java Object in the destination activity. Gson was also used to store and retrieve data from the database, because the database used in this project works with JSON formatted objects. And last it was used to read and write into the files from the device's internal storage. The library is under the license: <http://www.apache.org/licenses/LICENSE-2.0>.

**Lottie:** Lottie is a Java library that works with Android, iOS, Web, and Windows to make and show animations in the application[33]. Lottie parses animations made from Adobe After Effects and exports the animation into a JSON and then renders natively the animation on the mobile. In this project, Lottie was used to import free-copyright animations made by the Lottie community from LottiesFiles into the application to make better-suited animations for people with macular degeneration, in order to show more visible effects of the application. LottiesFiles[8] is a web page where anyone can download animations, most of the animations are for free and without copyright, but there is also a marketplace to buy or sell animations. The library is under the license: <http://www.apache.org/licenses/LICENSE-2.0>.

**MPAndroidChart:** MPAndroidChart is a Java library that works with Android to make and integrate charts into the application [34]. There is also an iOS version of this library called **Charts** [35]. In this project, the MPAndroidChart library was used to make bar charts to display the patient's exercise results, in order to show a progress history of the patient. The library is under the license: <http://www.apache.org/licenses/LICENSE-2.0>.

**Android-Image-Cropper:** Android-Image-Cropper is a Java library for Android that helps to crop images selected by the user [36]. With this library, the user could zoom an image, rotate it, select it from multi-source, and customise the shape, limits and style of the image. In this project, Android-Image-Cropper was used to implement the

profile activity, where a user can select a photo from the device's gallery and perform any of the actions mentioned above. The library is under licence: <http://www.apache.org/licenses/LICENSE-2.0>.

**CircleImageView:** CircleImageView is a Java library for Android to show images with a circle shape from the original bitmap made from libraries like Picasso [37] (that also was used in this project). In this application, CircleImageView was used to show a profile picture with the shape of a circle as show in Figure 22 Circle Image View sample, which is a common shape that is seen in other apps that uses profile's photos like <https://www.linkedin.com>. The library is under the license: <http://www.apache.org/licenses/LICENSE-2.0>.

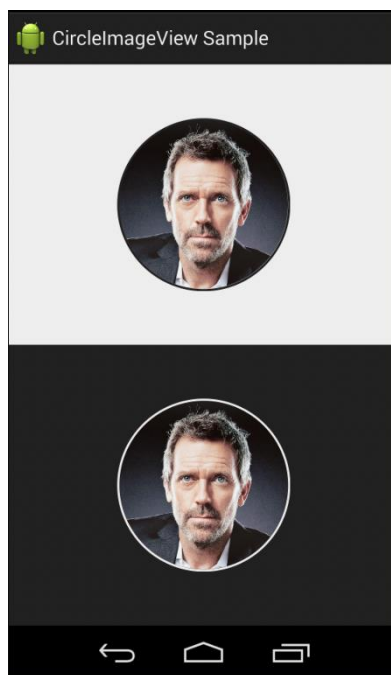


Figure 22 Circle Image View sample

**Picasso:** Picasso is a Java library for Android for image downloading and caching. Picasso handles imageView recycler and uses minimal memory resources with complex image transformation and automatic memory caching. In this project, it was used to download images from Firebase Cloud Storage [38]. The library is under the license: <http://www.apache.org/licenses/LICENSE-2.0>.

**ClickUp:** ClickUp is an App that helps organise tasks and keep track of their state[13]. The app provides a collaborative tool for generating tasks, give them priority and assign them to members of the team.

**GitHub:** GitHub is an online tool with version control that uses git [39]. In this project, it was used to host the code, share and work collaboratively with the team to track the changes made in any set of files.

## 6.2 API Version Adaptations

In this project, it was decided to implement an application that would be available to the most percentage of people possible while maintaining the application functionality. The objective was to make possible that the people that has a low income, who use old tablets and phones, to try the application and give them the opportunity to rehabilitate their sight. For this reason, the app had to go through adaptations to be functional to all the APIs that were intended to be available.

The Android minimum API version selected for this project was API 16: Android 4.1 (Jellybean). This API version makes the application be compatible with 100% of the devices available in May 2022. The percentages of the usage of the APIs can be seen in Figure 23 Compatible Android devices. At the same time, the maximum API version tested reached API 33, which is the highest API that could be tested from the virtual devices available in Android Studio.

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.1 Jelly Bean	16	100%
4.2 Jelly Bean	17	99,8%
4.3 Jelly Bean	18	99,5%
4.4 KitKat	19	99,4%
5.0 Lollipop	21	98,0%
5.1 Lollipop	22	97,3%
6.0 Marshmallow	23	94,1%
7.0 Nougat	24	89,0%
7.1 Nougat	25	85,6%
8.0 Oreo	26	82,7%
8.1 Oreo	27	78,7%
9.0 Pie	28	69,0%
10. Q	29	50,8%
11. R	30	24,3%

Figure 23 Compatible Android devices

The first obstacle was to find libraries that supported this range of APIs. So, every library chosen and used in the application works with every one of the APIs in the range mentioned above.

The first library imported was all the Firebase's tools. In these libraries, it was not until the update to version 30.0.0, that the firebase-core was removed from the dependencies [40]. With that dependency deleted, the Database and all the Firebase related tools can work with all the Android versions available.

The second obstacle was that from API 16 to API 18, the OutPutStramWriter Java class does not use StandardCharsets. For this reason, the users using API below 18, when writing data to a file in the device's internal storage. For example, when saving the patient's data, the application cannot correctly save the Latin Characters. Because of this, the users with API below 18 would not be able to correctly see Latin characters like 'ñ' or letters with accents. Nonetheless, they can use all the functions

regarding data storage and with the database, and if their name is written only in English characters, they would never notice it.

The third adaptation was made to include the Lottie library. The obstacle was that from API 25, Lottie animations that are imported to the layout are specified in the XML file, but in the APIs below 25, the View has to be created inside of the class that extends AppCompatActivity using the inflate function. Because of this, all the layouts that incorporated a Lottie View had to be moved to the Java class.

The fourth obstacle is regarding the language support of the application. The APIs below 23 manage the Locale class and the application's configuration differently. Because of this, all APIs cannot change the language freely. This problem was solved using the device's default language. For this reason, the users that execute the application in these versions of Android will have the device's default language and would not be able to change from that. Nonetheless, the app will be totally functional, and if the user changes the device language, the application will change to that language.

Last but not least, the fifth adaptation was regarding about the layouts. The first problem was that the layouts for API 16 were having problems correctly showing some of the activities and Views, such as ScrollViews and the Views that were using resources from the drawable folder. This problem was only in API 16, the rest of the versions worked perfectly with all the layouts and Views. To adapt the application to API 16, an XML layout was created exclusively for API 16 for each activity. The second problem was with the button's colours. The oldest APIs, from the API 16 to API 19, were having problems showing the properly colours of the buttons, where the colours were in purple or in grey instead of their original colour that was mainly white, red, and blue. The solution to this problem was to change the View attribute "android:backgroundTint" to the attribute "app:backgroundTint" on all the button's View attributes. In this way, the newest APIs can adapt to the oldest ones, and the button's colours show correctly.

### **6.3 Screen Size Adaptations**

Following the intention of making the application of this project to reach as many people as possible, there was made another adaptation. In this case, the intention is to make the app work with different screen sizes, ergo, to make the app work for most devices. To make this possible it was created layouts variants depending on the device's screen.

The project application is primarily targeted towards tablet devices because this type of device provides a larger screen that allows showing bigger and more visible layouts, helping those patients that have visual difficulties. But, at the same time, it was desired to reach most of the capable devices, like bigger phones, smaller tablets, or huge tablets. And because each one has a different screen size, and a different resolution, not all the layouts work with all devices. For this reason, it was necessary to create variants of the same layout, changing the size, the aspect, or the layout of the Views, to make all objects visible and correctly positioned.

Two different layouts were made, one for moderate/big tablets and one for smaller devices. Depending on the device's width, a layout would be chosen. Devices greater than 700dp which are normally tablets will choose the "tablet display", and smaller devices will choose the "default display". The differences between these two layouts are the text size and the percentages of the guidelines that were used as a guide for placing each element.

The Tablet display has an average font size of 40sp while the default display has an average font size of 20sp.

## Chapter 7 - Conclusions

The existence of an application specialized in macular degeneration, to practice and rehabilitate the vision with exercises, is very significant to patients with this disability, as they can now rehabilitate from the comfort of their own home and with a greater and higher frequency as they might not be able to have appointments with their health professional every day. And with an application like this they will not need to wait to continue with their rehabilitation process.

The app was developed in its totality to satisfy the patients necessities to progress in their rehabilitation, with or without their professional supervision. Helping their autonomy and letting the professional in charge be aware of their progress.

From the research documented in the "State of Art", it could be said with certainty that there is no other application that integrates a customisable focus dot, obtained from the evaluations. The focus dot feature helps the patients to progress quickly as they are constantly reminded of where to position their eyes. It is hoped that the focus dot feature, which was firstly used in the project, will be implemented in future visual rehabilitator apps. Another feature that makes the rehabilitator unique is the combination of eye-evaluations with a variety of free exercises.

The patient progress tracking is an essential part of their medical appointments because their professional can see the evolution of their patients just checking the application. The patients themselves can see the progress they are making which can encourage them to practice more and improve their sight to the point where they no longer need to use the app.

Evaluations made with experts in the matter were highly positive as they think this is a great tool to give to someone with AMD. It was well-received due to the advance made by the automatic calculated focus dot. Also, the exercises were said to be practical and functional.

In the matter of compatibility, Android is the most common operating system, and this application works with 100% of the Android devices, as it works with any API equal or greater than 16. The advantage of this extreme compatibility is its inclusivity: it allows people with lacking resources that they cannot afford buying a new tablet to use the application with an old tablet.

With the help of this application more patients can be reached. Additionally, they have a better rehabilitation because they can practise doing exercises more frequently, as they do not require an appointment with the professional, also in the comfort of doing it whenever they want and on their personal space.

## Chapter 8 - Contributions

The efforts of each team member will be highlighted in this chapter. Because it was chosen pair programming, some contributions may be shared among team members.

### 8.1 Daniela Alejandra Escobar Suarez

- Initial design in Balsamic, translating the consulting experts' design into a design closer to the final product.
- First version of exercises. Consisted in ha button with a figure that changed the position randomly every time that was clicked.
- Developing the patient's test:
  - First was figuring out how show the focus dot both in the test and finally in the exercises. By using a coordinate system withing a the grided circle and following the expert indication.
  - By using the information from the tablet such as pixels per inches to know the size in centimetres (through transformations).
  - Adapting the professional manual input if the device is smaller than 10 cm.
  - With the distances, draw dot function and timers the first test was developed.
  - Also, it was made another function to make an object blink that also uses timers to make it visible and invisible.
  - Then by detecting the touch coordinates and transforming it from coordinates in relation to the screen to an associated coordinate that uses the centre of the circle as reference to activate and deactivate dots. With this functionality the professional may enter the patient stain manually and the patient may test the parts that seem distorted. This are the professional manual input and patient second test.
  - All three identical version for left eye, right eye and both eyes.
  - From the tests it is shown a results resume. Then it is calculated a general stain taking common and uncommon coordinates from each test. Each individual coordinate is taken in consideration to calculate the mean coordinate that will be the suggested focus dot.
  - Another space was done so the professional may move freely the focus dot to wherever within the grid seems convenient for the patient.
  - To simplify the tests classes, there was created to classes to describe the two general behaviours.
  - Skipping the first and second test to devices smaller than 10 cm.
- From the test's development, it was created a function to draw the focus, with some necessary information.
- Helping to add the focus dot in the exercises and adapting for each one.
- Pause functionality:

- In each exercise by stopping the timers and adding a pause button it was added the pause functionality. This gives the patient the possibility to stop the test if necessary and return to the point where they left.
- The pause also displays a menu to go back home or the settings menu. On an original version, the settings menu was eliminated as it only contained turning up and off the focus dot, but it returned as the changing size of the focus was added.
- Developing the first demonstration exercise. Making sixteenth adaptation from the seventeenth exercise.
- Saving in database and local the information related to the focus, including the on/off status, size, and test history. In test history it is saved each test with all the coordinates, resume and resulting focus. They are saved by dated of solving the test.
- Settings were in main menu of the patient and in each exercise and its description:
  - First was the focus dot switch. This was to turn on or off the focus dot. It had to communicate with the database and/or local information to know previous value. Then depending in its location and next action, it would update or not the information. It had to hide the focus dot in the middle of the exercise if request and to appear if also requested in the middle of the exercise through the pause menu.
  - By professional recommendation from testing the app, it was added the changing size of the focus dot. It consisted in having the dot having a value between one metric and double the metric unit. It has a simple interface that shows the dot changing as pressing the buttons to make it bigger or smaller. This also had to be possible to be changed in the middle of an exercise.
- Test history display. Program and design the list of tests and layouts for patient and professional.
- Helping in the development and testing of the exercises. Adjusting layouts to different device sizes. Calculating the transformation of size for the object-focus relation in smaller devices.
- Bug fixing and adding changes indicated by experts.
- Visual feedback from fourteenth, fifteenth, sixteenth and seventeenth exercises.
- GitHub Management. Version control in branches, including develop. Also maintaining a copy of a working version in master branch.
- Memory drafting

## 8.2 Diego Alejandro Rodríguez Pereira

- Helped in the Balsamiq and first prototypes for the application.
- Database creation and connection to the Android Studio's project. Including Database data entry and rules. Developing the structure of the NoSQL database. Also includes the implementation of the patient and professional model class which are used to store the data of professionals and patients.

- User registration and login with the use of Firebase Authentication for improved security.
- Implementation of the internal storage system. Including the creation of the class to store data in the device's storage. With synchronisation and connection of internal storage with the Real-Time Database.
- Implemented the offline functionality in the app. This includes the database's adaptation to a device that does not have an internet connection.
- Implemented the download and upload functionality, so the user can work offline and download and upload when they want to update the information.
- Implemented change password and restore password. Including the creation of the template and the implementation of the feature that sends an email with a link to a web page where the users can restore their password.
- Implemented layouts and activities regarding the patients and professional interaction with the database and Authentication. Which includes the creation of the first prototype and the first layouts and activities.
- Implemented the base layout of the project with the different activities. Some of them are patient login and signup, professional login and signup, patient and professional home page, and patient info, exercise result, professional profile, patient result history, edit profile.
- Implemented patient's form. This includes the contact with the professional for the implementation of a form that would resume all the patient's information to be able to diagnose, treat, rehabilitate and recommend exercise for the patient.
- Creation of the application theme which includes the colour format of the buttons and the drawable design for the list of views. Also included the search and implementation of some of the Material Design Views such as dialogs and EditTextInput.
- Implemented the patient and professional formatting to store and retrieve data from the database, JSON strings and Java Objects.
- Patient Numeric Code format. Which helps identify where a patient belongs.
- Created a general class to store the exercise's results information.
- Helped and created the integration of the exercises and tests with the database and the device's storage. Including the creation of the exercise structure in the database and the creation of the classes to manage and store the data.
- Contact with the professional to implement a better-suited login and signup for the patient, the making of the base activities and the form implementation.
- Created a result activity and an activity to show the history of the exercise's results.
- Research and integration of the following project libraries: Gson, Lottie animations, MPAndroidChart, Image Cropper, Circle Image View, Picasso.
- Implemented the app's visual effects and animations.
- Implemented the two languages system for the application.
- Translation of the strings from English to Spanish and vice versa.
- Creation, implementation, and maintenance of error handlers.
- Construction of the view lists which use card views.
- Creation of the drawable layouts.
- Imported some of the copyright-free icons.

- Integrated the application icon.
- Implemented the adaptations necessary to make work the applications with all the Android APIs available at the moment of writing. Which includes:
  - Research of libraries that works with all APIs versions. Maintaining the desired functionality for the application.
  - Research for the API's releases notes to find which functions work which each API.
  - Layout adaptations to older APIs. Which includes the drawables and buttons adaptations. Also includes the Lottie animations adaptations.
  - Language adaptations for the APIs that do not have the configuration nor Locate parameters to change the language.
  - Firebase platform adaptation. Which includes: deletion of Analytics, adaptation of Authentication, Cloud Storage, removal of Firebase's Core, integration of Realtime Database.
  - Internal storage adaptation. This includes the system's adaptation that writes and reads the files of the device's internal storage. It also includes the adaptation of StandarCharsets to the APIs between 16 and 18.
- Implemented layout adaptations for the application to work with different screen sizes. Which includes the research of Layout adaptability.
- Recording of the voices for the exercises that required audio functionality. Where it is indicated by voice the object the patient has to search for.
- Implemented the functionality of reproducing with MediaPlayer the voices in each exercise that required the feature.
- Implemented the application's full-screen functionality. Adaptation of the application to hide the device's status and action bars to work in full screen.
- Adaptation of all the layouts to work in landscape mode. Includes fixing the problem of the layout not adapting properly to the device's landscape mode.
- Creation of the graphics which show the progress history of the patient's exercises results. Implemented using MPAndroidChart to create and deliver the data in a bar chart.
- Created professional's profile. Which includes:
  - Showing the professional profile picture. Which includes:
    - Integration of Picasso to download the image from the database.
    - Integration of Cloud Storage to store the profile picture.
    - Integration of AndroidImageCropper to crop, rotate, and zoom in and out the image.
    - Integration of CircleImageView to show the ImageView with the form of a circle.
  - Change of the data, which includes: change name, change email, change password, change phone number.
  - Contact professional. Which includes the integration of an intent to write emails, and the intent to make a call.
- Inspect, improved, and discussed the code and changes made by other partners in the project to make sure that the functionality implemented by them was working and finding possible malfunctions or bugs.
- GitHub management. Which includes managing the different branches.
- Memory drafting.

### 8.3 Alba María Martín Puebla

- Programming the rehabilitating exercises, which consists of the exercises from 2 to 17.
  - Controlling several Timers at a time with the purpose of making the exercises more dynamic and to give better feedback to the user.
  - Exercises 2 to 5 and 10 with one shape or letter in different sizes, with two Timers (one for the red dot and another for the shape/letter).
  - Exercises 6 to 9 with two shapes and letters of different sizes and two red dots, one for each, with three Timers (one for each shape or letter and one for both red dots) and several counters to control all.
  - Exercises 11 to 13, adjusting the alpha value so that all the buttons were invisible to users in a way they could only see the background image, but the buttons were still touchable, with and without the talkback. Handling lots of buttons of different sizes (each distinct and the same button has a different size for each layout for devices with different width) with a red dot each for each button. Managing Timers for the buttons and red dots, and making sure everything was synchronized: the voice with the text, red dots on their correct position (and only being visible during their corresponding moment) and the buttons with listeners to check if they are pressed in a correct or incorrect moment.
  - Exercises 14 to 17 handling lots of buttons and listeners and making the buttons as big as possible for each width size of the different devices. Giving colourful feedback using different functions defined in Android Studio and managing an immediate end when the exercise ends (all the correct buttons pressed or time expired).
- Designing all the exercises based on the instructions and documentation the expert gave us.
- Interpreting the expert's requirements and transforming the expert's design specifications into programming exercises.
- Drawable layouts of the rehabilitating exercises (exercises 2 to 17).
- Programming in java and making drawable layouts of the descriptions for the exercises, including the reading and possible modification of the desired seconds with EditTests, with error-proofing in case a user lets the space for the seconds empty, the application will not break, just set the default time.
- Writing all the descriptions for the exercises in English and Spanish.
- Exhaustive testing of all the exercises and exercises' descriptions with and without talkback.
- Testing the whole app with and without talkback.
- Studying the IDE of Android Studio to correctly understand the implementation of applications.
- Solving bugs that I found or that my partners found, especially those related to exercises and exercises descriptions.
- Doing layouts for devices with widths greater and smaller than 700dp.
- Doing layouts for devices with widths greater than 700dp, in between 700 and 600dp and lower than 600dp for exercises 11, 12 and 13, which were very specific cases that needed more layouts so they could adapt better as they have images, and it could not be just set the same layouts as before.

- Using ScrollViews for adaptative layouts.
- Testing the app in modern and old devices, mobile phones and tablets, some of them virtual and other physical.
- Testing the app in Android with API 16 with a virtual device.
- Collaborating with the coding and the testing of the app to make sure it error proof.
- Coding and checking the application to guarantee it is accessible and every button is readable with the talkback on.
- Drawing the focus point in the appropriate position in the exercises so that the specified user can complete them with support.
- Adding the calls to the generic classes of saving the exercises solutions in the database and showing the exercises results to the user in some classes and collaborating to add them in others.
- Helping to add the pause in some classes.
- Programming and designing the list of exercises menu.
- Deciding and applying the main colours of the application.
- Applying and checking that there is sufficient distance between the buttons of the application so the users with AMD can, touch the button they want with ease.
- Translations of texts and strings of the application from Spanish to English and the other way around.
- Searching for images without copyright to use on the exercises.
- Drawing images to use on the exercises and in the app in general (shapes for the exercises and some other shapes used in the app).
- Checking that, whilst the device used for testing has internet access, the data being generated at the moment is instantly updated in the database.
- Checking that when a patient presses "upload data" with the internet on, their data is updated into the data base.
- Helping with the translation to English of the memory.
- Revising and correcting the memory in English, to detect mistakes and solve them and rewrite them in a more formal way.
- Doing a Gantt chart that shows a summary of the timeline of the project and the people responsible for each task.
- Maintain an efficient conversation with teachers and the professional and organising meetings to talk about the app's progress
- GitHub management which included dealing with several branches (mainly the branch called develop and my own branch and sometimes the one's of the other members of the team).
- Continuously revise my team partners' code and test it, report their bugs and assist them.
- Memory drafting.

## 8.4 María Núñez Conde

- Database creation and connection with the android studio project: The configuration of a new database with Firebase and the adjustments needed on the project for it to have access to the database. Giving access to the database web page to all the team members for them to drop all data or to get an overview of the data managed.
- Database data entry: Deployment of CRUD instructions regarding the Realtime database in which user data is stored. The creation of two java classes: Professional and Patient, for an easy access and a data exchange mechanism.
- The study of the IDE Android Studio for a correct implementation of the application and the structure needed for the application to be scalable and easy to understand.
- Creation of the first version of user identification: Deploying the first version of user data and password management that was later updated to a google authentication mechanism for better security.
- Connection between users: Generating a method of associating patients to professionals, making use of their user ID.
- Creation of the base structure of the project: building of the base activities (user page, home, settings) and interaction between them.
- Adaptive layout: Making use of an adaptive layout that works with percentages rather than with pixel dimensions to provide a more heterogeneous display on different devices.
- Design of an App logo, using Adobe Illustrator.
- Implementation of a method to view passwords while they are being written, by adding an eye icon that when clicked shows the password in normal characters.
- Creating the patient-sign up form and connecting it to a database with the help of a transfer object named patient.
- Synchronizing the professional logging with firebase's auth technology, for adding security to the app.
- Modification of the seventeenth exercise, changing the structure and adding buttons for each letter instead of the old version where the text was all together in one text view.
- First prototype of the professional page, where a professional had a list of patients to choose from and analyse their data.
- Creating the Activity of settings, this activity allows users to deactivate the focus dot as well as changing the focus dot size.
- In the exercise seventeenth and sixteenth, adding an extra activity that shows the letters that need to be clicked in a big size, because the previous version had the letters that needed to be clicked in a small size. This change was a suggestion made in the last stages of testing with users.
- Taking notes during each meeting and writing a summary with the main topics discussed, as well as the decisions made.
- Doing two versions of each layout, one for displays wider than 700dp and other for displays shorter than 700dp. The main differences between these two

layouts were the text size and the percentages of the guidelines that was used as a guide.

- Following the recommendations of the final users and adding an activity before the Exercise's list activity where users are reminded of the distance they should keep between their eyes and the device.
- Adding some go back buttons, for an easy navigation as well as implementing the go back methods needed so that the user could not end up in an infinite loop.
- Translation of some strings from English to Spanish. As well as adding new strings to the project.
- Testing the Database structure, doing requests to the Firebase's database and trying to introduce unexpected data to see the database reaction and helping with the debugging process.
- Testing the Exercises that were done by other members and reporting the bugs found so that they could do the debugging.
- Testing the Layouts in different devices, themes, and sizes. Trying on several Android Studio emulators as well as on different devices such as tablets and smartphones. After the testing, correcting all the layout-related bugs encountered.
- Adjusting the text sizes for them to fit all devices.
- Doing a Gantt chart that shows the timeline that the project followed and the people responsible for each task.
- GitHub management. With the interaction of different branches and reverting changes.
- Memory drafting, formatting and revision of the parts made by other team members.

## Bibliography

- [1] "Central Vision Test - Aplicaciones en Google Play." <https://play.google.com/store/apps/details?id=com.eyexamtest.centralvision&hl=es&gl=US> (accessed May 08, 2022).
- [2] L. Segre, "Amsler Grid: A Test for Macular Degeneration and Other Vision Problems," *All About Vis.*, Accessed: May 26, 2022. [Online]. Available: <https://www.allaboutvision.com/conditions/amsler-grid.htm>.
- [3] "Eyecare- Amsler Grid Eye Test - Aplicaciones en Google Play." <https://play.google.com/store/apps/details?id=com.eyecareapp&hl=es&gl=US> (accessed May 08, 2022).
- [4] "Amsler Grids en App Store." <https://apps.apple.com/es/app/amsler-grids/id982604083> (accessed May 08, 2022).
- [5] "Verana Vision Test en App Store." <https://apps.apple.com/es/app/verana-vision-test/id1507459848> (accessed May 08, 2022).
- [6] "Home | Maculatester." <https://www.maculatester.com/> (accessed May 08, 2022).
- [7] "(632) MaculaTester App Demo (English) - YouTube." <https://www.youtube.com/watch?v=1iPMNHVGXHU> (accessed May 08, 2022).
- [8] "Simply Visio - Aplicaciones en Google Play." <https://play.google.com/store/apps/details?id=com.simply.simplyVisio&hl=es&gl=US> (accessed May 08, 2022).
- [9] "Eye Exercises & Training Plans - Apps on Google Play." [https://play.google.com/store/apps/details?id=com.eyexamtest.eyecareplus&hl=en\\_US&gl=US](https://play.google.com/store/apps/details?id=com.eyexamtest.eyecareplus&hl=en_US&gl=US) (accessed May 10, 2022).
- [10] "Diferencias entre PX, DP y SP en Android | OpenWebinars." <https://openwebinars.net/blog/diferencias-entre-px-dp-y-sp-en-android/> (accessed May 10, 2022).
- [11] "Metodología Scrum: cómo aplicar el método Scrum | APD." <https://www.apd.es/metodologia-scrum-que-es/> (accessed May 11, 2022).
- [12] "Download Android Studio and SDK tools | Android Developers." [https://developer.android.com/studio?hl=es&gclid=Cj0KCKjwsdiTBhD5ARIsAlpW8CJZoJyieCwkiY\\_ZYr\\_Kt6jX8SrMdmjrtejlHC4z2LxkmJOGO\\_JmbkAaAjzMEALw\\_wcB&gclsrc=aw.ds](https://developer.android.com/studio?hl=es&gclid=Cj0KCKjwsdiTBhD5ARIsAlpW8CJZoJyieCwkiY_ZYr_Kt6jX8SrMdmjrtejlHC4z2LxkmJOGO_JmbkAaAjzMEALw_wcB&gclsrc=aw.ds) (accessed May 08, 2022).
- [13] "Download ClickUp™ | Mobile & Desktop App, Chrome, Alexa, Google Home." [https://clickup.com/download?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=gs\\_cpc\\_t3\\_nnc\\_brand\\_trial\\_all-devices\\_tis\\_lp\\_x\\_all-departments\\_alpha\\_clickup&utm\\_content=all-countries\\_kw-target\\_text\\_all-industries\\_clickup-app&utm\\_term=e\\_clickup\\_app&gclid=Cj0KCKjwsdiTBhD5ARIsAlpW8CL0RxOsu2mp0eRRxWnC\\_dzT-czY2lOiXj-28agmQAL4sPFrp6gzg8kaAfirEALw\\_wcB](https://clickup.com/download?utm_source=google&utm_medium=cpc&utm_campaign=gs_cpc_t3_nnc_brand_trial_all-devices_tis_lp_x_all-departments_alpha_clickup&utm_content=all-countries_kw-target_text_all-industries_clickup-app&utm_term=e_clickup_app&gclid=Cj0KCKjwsdiTBhD5ARIsAlpW8CL0RxOsu2mp0eRRxWnC_dzT-czY2lOiXj-28agmQAL4sPFrp6gzg8kaAfirEALw_wcB) (accessed May 08, 2022).

- [14] "Balsamiq Company Info — We're Good People, and We Care! | Balsamiq." <https://balsamiq.com/company/#map> (accessed May 08, 2022).
- [15] "Tablet Operating System Market Share Worldwide | Statcounter Global Stats." <https://gs.statcounter.com/os-market-share/tablet/worldwide> (accessed May 08, 2022).
- [16] "How to Turn on Color Blind Mode in Android 5.0 | Tom's Guide." <https://www.tomsguide.com/us/color-blind-mode-android-5.0,news-20475.html> (accessed May 09, 2022).
- [17] J. Nielsen, *Usability Engineering*. 1994.
- [18] "Design - Material Design." <https://material.io/design> (accessed May 10, 2022).
- [19] "Firebase." <https://firebase.google.com/> (accessed May 08, 2022).
- [20] "Firebase Authentication | Acceso multiplataforma simple y sin costo." <https://firebase.google.com/products/auth?hl=es> (accessed May 10, 2022).
- [21] "Firebase Realtime Database | Firebase Documentation." <https://firebase.google.com/docs/database?hl=es> (accessed May 10, 2022).
- [22] "Firebase Realtime Database | Almacena y sincroniza datos en tiempo real." <https://firebase.google.com/products/realtime-database?hl=es> (accessed May 10, 2022).
- [23] "Cloud Storage para Firebase | Almacena y entrega contenido con facilidad." <https://firebase.google.com/products/storage?hl=es> (accessed May 10, 2022).
- [24] "Inches to cm converter." <https://www.inches-to-cm.com/> (accessed May 06, 2022).
- [25] "Circle - Wikipedia." <https://en.wikipedia.org/wiki/Circle> (accessed May 03, 2022).
- [26] "Midpoint circle algorithm - Wikipedia." [https://en.wikipedia.org/wiki/Midpoint\\_circle\\_algorithm](https://en.wikipedia.org/wiki/Midpoint_circle_algorithm) (accessed May 03, 2022).
- [27] "How do android screen coordinates work? - Stack Overflow." <https://stackoverflow.com/questions/11483345/how-do-android-screen-coordinates-work> (accessed May 07, 2022).
- [28] "MotionEvent | Android Developers." <https://developer.android.com/reference/android/view/MotionEvent> (accessed May 07, 2022).
- [29] "Timer | Android Developers." <https://developer.android.com/reference/java/util/Timer> (accessed May 10, 2022).
- [30] "Los cuatro principios del buen diseño de Donald Norman | | Webnova Diseño Web a Medida." <https://webnova.com.ar/los-cuatro-principios-del-buen-diseno-de-donald-norman/> (accessed May 11, 2022).
- [31] "MediaPlayer | Android Developers." <https://developer.android.com/reference/android/media/MediaPlayer>

(accessed May 10, 2022).

- [32] "google/gson: A Java serialization/deserialization library to convert Java Objects into JSON and back." <https://github.com/google/gson> (accessed May 08, 2022).
- [33] "airbnb/lottie-android: Render After Effects animations natively on Android and iOS, Web, and React Native." <https://github.com/airbnb/lottie-android> (accessed May 08, 2022).
- [34] "PhilJay/MPAndroidChart: A powerful 🚀 Android chart view / graph view library, supporting line- bar- pie- radar- bubble- and candlestick charts as well as scaling, panning and animations." <https://github.com/PhilJay/MPAndroidChart> (accessed May 08, 2022).
- [35] "danielgindi/Charts: Beautiful charts for iOS/tvOS/OSX! The Apple side of the crossplatform MPAndroidChart." <https://github.com/danielgindi/Charts> (accessed May 08, 2022).
- [36] "ArthurHub/Android-Image-Cropper: Image Cropping Library for Android, optimized for Camera / Gallery." <https://github.com/ArthurHub/Android-Image-Cropper> (accessed May 08, 2022).
- [37] "hdodenhof/CircleImageView: A circular ImageView for Android." <https://github.com/hdodenhof/CircleImageView> (accessed May 08, 2022).
- [38] "square/picasso: A powerful image downloading and caching library for Android." <https://github.com/square/picasso> (accessed May 08, 2022).
- [39] "GitHub." <https://github.com/> (accessed May 30, 2022).
- [40] "Firebase Android SDK Release Notes | Firebase Documentation." <https://firebase.google.com/support/release-notes/android#2022-03-24> (accessed May 10, 2022).

# Appendix

## A. Figures



A. I Second exercise



A. II Third exercise



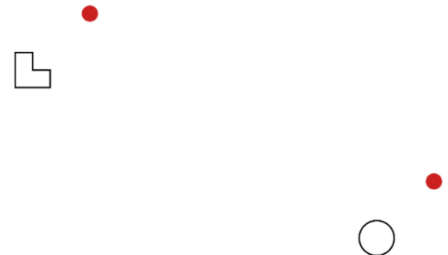
A. III Fourth exercise



A. IV Fifth exercise



A. V Sixth exercise



A. VI Seventh exercise



L



L



A. VII Eighth exercise



T



T



A. VIII Ninth exercise



A. IX Tenth exercise



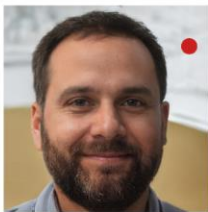
Find the right eye



A. X Eleventh exercise



Find the right eye



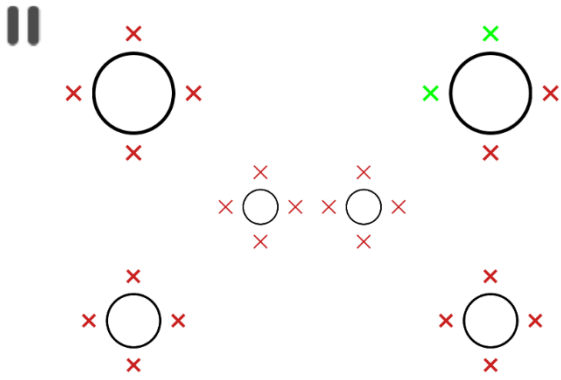
A. XI Twelfth exercise



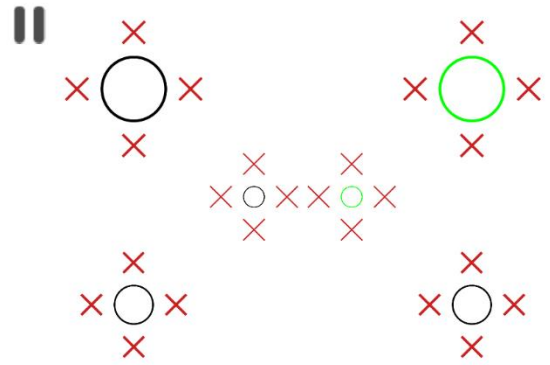
Find the plant



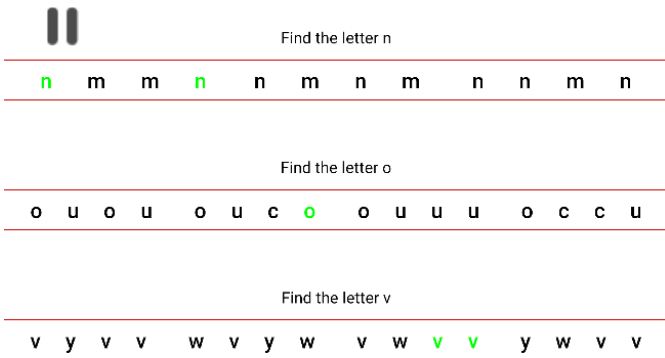
A. XII Thirteenth exercise



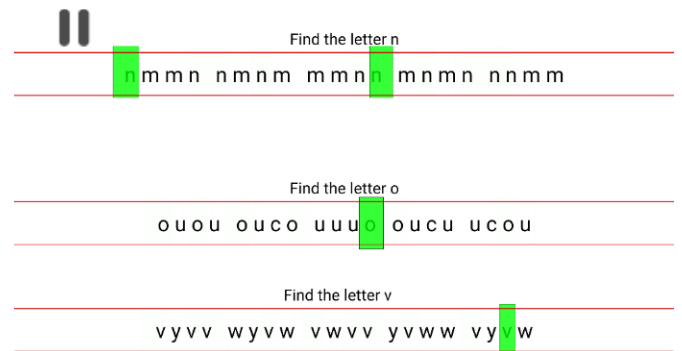
A. XIII Fourteenth exercise



A. XIV Fifteenth exercise



A. XV Sixteenth exercise



A. XVI Seventeenth exercise