



**Faculty of Information Technology**

**University of Palestine**

## **Healthy Guide**

Auto-detection prototype of a healthy  
sitting for computer users

### **Submitted by :**

Baraa Omar Doghmosh

Mohammad Ali Tayeh

Ibrahim Ahmad AlTana

### **Supervised by :**

Dr. Mohammad Awad

**UNIVERSITY OF PALESTINE**

**(UP)**

## إهداء

إلى من علمني النجاح والصبر

إلى من افتقده في مواجهة الصعاب

ولم تمهله الدنيا لأرتوي من حنانه.. أبي

وإلى من تتسابق الكلمات لتخرج معبرة عن مكنون ذاتها

من علمتني وعانت الصعاب لأصل إلى ما أنا فيه

وعندما تكسوني الهموم أسبح في بحر حنانها ليخفف من آلامي.. أمي

## Abstract

The 'Healthy Guide' application is a software that maintain the eye health at the first class by providing two major features, first is the eye break that help user to take a break every ten minutes and take an eye exercises, the second feature is the healthy sitting that take a picture every hour and calculate the distance between user and computer screen to help him to be careful.

Additional features included in the 'Healthy Guide' is improve focus that provide six sounds to improve user focus, sleep and relax. And the advice center to notify user with some healthy tips every day.

The application developed using java language it's the most language used today, java help us to implement powerful user interface using JavaFx and to implement application on multiples platforms.

The application was tested on Windows, Linux and Mac os, all functions is working properly, also we test the application performances and the application installer to make sure it is working without any problem or exemption and all result it is expected to.

The evaluation of the application was done with wide range of selected users that categorized on three group, all of them test the application and evaluate the application functions and features and all results it is expected to our vision.

## Table of Contents

Acknowledgments.....	II
Abstract .....	III
Table of Contents.....	IV
List of Figures .....	VII
List of Tables .....	VIII
1 Chapter 1 Introduction .....	1
1.1 Project Background .....	2
1.2 Project Motivation.....	2
1.3 Problem Statement .....	3
1.4 Project Question .....	3
1.5 Project Objectives .....	3
1.6 Project Scope and limitation .....	3
1.6.1 Project Scope .....	3
1.6.2 Project Limitation .....	3
1.7 Conclusion.....	4
2 CHAPTER 2 LITERATURE REVIEW .....	5
2.1 Related Work.....	6
2.1.1 EyeLeo .....	6
2.1.2 SafeEyes.....	7
2.1.3 Stretchly .....	8
2.1.4 COMPARISON BETWEEN THE SYSTEMS .....	9
2.2 Choice of Technology .....	10
2.2.1 Desktop Application .....	10
2.2.2 Java (programming language).....	11
2.2.3 JavaFx .....	11
2.3 Google Cloud Vision API .....	12
2.4 Conclusion:.....	13
3 CHAPTER 3 RESEARCH METHODOLOGY .....	14
3.1 Project Methodology:.....	15
3.2 Extreme Programming Phases .....	16
3.2.1 Planning: .....	17
3.2.2 Design: .....	17
3.2.3 Coding:.....	17

3.2.4	Testing: .....	17
3.3	Agile methodology.....	18
3.3.1	The advantages of agile methodology: .....	18
3.3.2	The Limitations of agile methodology:.....	19
3.4	Conclusion:.....	19
4	Chapter 4 Prototype Design and Development.....	20
4.1	System Description .....	21
4.2	User Description.....	21
4.3	System Requirements .....	21
4.3.1	What is a Requirement? .....	21
4.3.2	Requirements Gathering .....	22
4.3.3	Functional Requirements .....	22
4.3.4	Non-Functional Requirements .....	22
4.4	System Analysis .....	23
4.5	Application Requirements:.....	23
4.5.1	Software Requirements .....	23
4.5.2	Hardware Requirements.....	23
4.6	Development Requirements: .....	23
4.6.1	Software Requirements .....	23
4.6.2	Hardware Requirements.....	23
4.7	Requirements Analysis.....	24
4.7.1	Use Case.....	24
4.7.2	Usecase List: .....	26
4.7.3	Usecases Details: .....	26
4.7.4	Sequence Diagram: .....	30
4.8	Design.....	33
4.8.1	User Interface Sketch.....	33
4.9	Storage Schema .....	36
4.10	Implementation: .....	38
4.10.1	Overview of implementation phase: .....	38
4.10.2	Graphical user interface (GUI) of the Desktop Application .....	38
4.11	Testing.....	44
4.11.1	Performance Testing .....	44
4.11.2	Functionality testing.....	45

4.11.3	Preferences Testing.....	45
4.11.4	Application Installer.....	45
4.11.5	Operating System Compatibility.....	46
4.12	Evaluation.....	47
5	Chapter 5 conclusion.....	48
5.1	Development process .....	49
5.2	Programming language .....	49
5.3	Problem we faced.....	49
5.4	what have been done .....	49
5.5	What technologies have we learned.....	49
5.6	Future work .....	50
	References.....	51
	Appendixes .....	52

## List of Figures

Figure 2-1 How Desktop Application Work.....	10
Figure 2-2 Pose angle, Coordinate system.....	13
Figure 3-1 Extreme Programming phases.....	16
Figure 3-2 Agile Planning Process .....	18
Figure 4-1 Use Case Diagram.....	25
Figure 4-2 Sequence Diagram for Eye Break.....	31
Figure 4-3 Sequence Diagram for Select Sound.....	31
Figure 4-4 Sequence Diagram for Calculation Distance .....	32
Figure 4-5 Sketch Main Screen.....	33
Figure 4-6 Sketch Eye Break Setting.....	34
Figure 4-7 Sketch Healthy Sitting.....	34
Figure 4-8 Sketch Improve Focus Screen.....	35
Figure 4-9 Sketch Advice Setting.....	35
Figure 4-10 Sketch Notification Before Break.....	36
Figure 4-11 Preferences Diagram .....	36
Figure 4-12 Main Screen GUI .....	38
Figure 4-13 Healthy Sitting GUI .....	39
Figure 4-14 Healthy Sitting on Training Dialog.....	39
Figure 4-15 Eye Break Setting.....	40
Figure 4-16 Improve Focus Screen of Birds Sound.....	41
Figure 4-17 Improve Focus Screen of Rain Sound.....	41
Figure 4-18 Sketch Improve Focus Screen of Waves Sound .....	42
Figure 4-19 Advice Setting.....	42
Figure 4-20 Eye Break Background .....	43
Figure 4-21 Notification Before Break.....	43
Figure 4-22 JProfiler Performance Testing.....	44
Figure 4-23 'Healthy Guide' on Mac OS .....	46
Figure 4-24 'Healthy Guide' on Fedora 25 (Linux) .....	46

## List of Tables

Table 2-1 COMPARISON BETWEEN THE SYSTEMS .....	9
Table 4-1 usecases list .....	26
Table 4-2 Usecase Details for Change Eye Break Preference .....	26
Table 4-3 Usecase Details for Change Healthy Sitting Preference .....	27
Table 4-4 Usecase Details Train ‘Healthy Guide’ to Measure Distance .....	27
Table 4-5 Usecase Details for Play Sounds to Improve Focus .....	28
Table 4-6 Usecase Details for Display Eye Break.....	28
Table 4-7 Usecase Details for Click Skip.....	29
Table 4-8 Usecase Details for Measure Distance between User and Screen.....	29
Table 4-9 Usecase Details for Display Healthy Advice .....	30
Table 4-10 Preferences key-value pair .....	37



# 1 Chapter 1 Introduction

This chapter displays the project idea, motivation, problem, objectives and scope on development of the application.

## **1.1 Project Background**

In the recent years, technology has become essential need in human being. Without it, multiple things would not be able to see to light excite. It is impossible to ignore the impact of technology in our daily lives, nowadays technology is being developed as a very fast pace in all activities of life such as health, education, health, business, industry and other aspects of life. Hence technology has presented dramatical change on help people and easier of life, due to its vital role in making people's life easy.

In fact, technology is at the very heart of human progress and it is interested of developing more and more application to help people solve problems in their way.

Software application development began with desktop applications, which could be used on standalone machines. However, with the appearance of web and online commerce, web application development gained importance. Word processors and media players can be considered to be typical desktop applications, while an online shopping cart on an ecommerce website can be considered as a web application.

Desktop application means “any software that can be installed on a single computer (laptop or a desktop) and used to perform specific tasks. Some desktop applications can also be used by multiple users in a networked environment”, they are security it provides total control over the standalone applications and protect it from various weakness, available anytime offline and provide you with faster response and rich user experience[1].

Known since ancient times that user need to taking small breaks based on medical research.

In fact, sitting in front of the computer for long hours strains to the eye because it make the person who's siting receive the radiation emotion and get his/her eyes strained, it strain the eye and affects the integrity of the spine. Researchers advice long-period-computer-user to point to user that the reason is when working on a computer [2]. This leads to eye strain and other physical problems. 'Healthy Guide' - the system that project team are going to build - ensures to maintain health the human.

## **1.2 Project Motivation**

Technology has changed the way we live that means working more hours. Every programmer starts work at least 8-10 hour daily working [2]. It is hard work because they spend a lot time of siting computer without moving him body. By using technology, we tried to help people to do eye exercise and make sure user was in safe siting position.

### **1.3 Problem Statement**

Many users suffer from sitting for long hours in front of computer screen, doctor's specialists have proved that sitting for hours in front of the computer screen cause strain the eye and dryness eye and in the long term can cause this weakness the vision this at the eye level, also sitting uncomfortable will affect the safety of the spine [2, 3].

In order to reduce the impact on users and based on medical information, there is an urgent need to apply screen breaks during the handling of the computer and make sure that the user in a safe sitting position [2, 3].

### **1.4 Project Question**

The focus of this project is on developing an application for maintaining human health. The question of the study is:

- How to develop an application that helps long-period computer user to maintain their eye and Spine based on medical researches?

### **1.5 Project Objectives**

This project aims to:

- analyzing the problems which that computer users face, to reach a solution that reduces the side effects of looking at computer screen and sitting at front of computer screen.
- design user interfaces that meet the application requirements.
- development a prototype that solve the mentioned problem.
- test and evaluate the prototype.

### **1.6 Project Scope and limitation**

#### **1.6.1 Project Scope**

The scope of the application encloses the following points:

- The application presents the desktop application for computer users in the world.
- The final version of application will be open source.
- The application development time from September 2016 to January 2017.

#### **1.6.2 Project Limitation**

- The application does not support Arabic language.
- The project does not support mobile application.

## **1.7 Conclusion**

This chapter discuss about background of the project, what motivate us to make this project and will make the technology less harm for the user in a simple way on other hand we talked about the scope of the project.

## 2 CHAPTER 2 LITERATURE REVIEW

This chapter reviews and identify the technologies which used to create the application, and clarify some concepts and related work and similar systems of the application.

## 2.1 Related Work

### 2.1.1 EyeLeo

EyeLeo is a handy PC application that regularly reminds you to take short breaks for your eyes. Specially designed to everyone who spends more than an hour per day looking at a display [4].

#### Features of EyeLeo:

- Short and long breaks with eye exercises.
- Notifications announcing a long break coming.
- Strict mode that does not allow skipping the breaks.
- Customizable parameters.

#### Limitation of EyeLeo:

- Eyeleo work only on windows operating system.
- Eyeleo supports only English language.
- Development stopped from Jan 2016 (last release).
- Eyeleo Features limited in Eye break, Notification and Eye Exercises.



*Figure 2-1 EyeLeo Screenshot*

## 2.1.2 SafeEyes

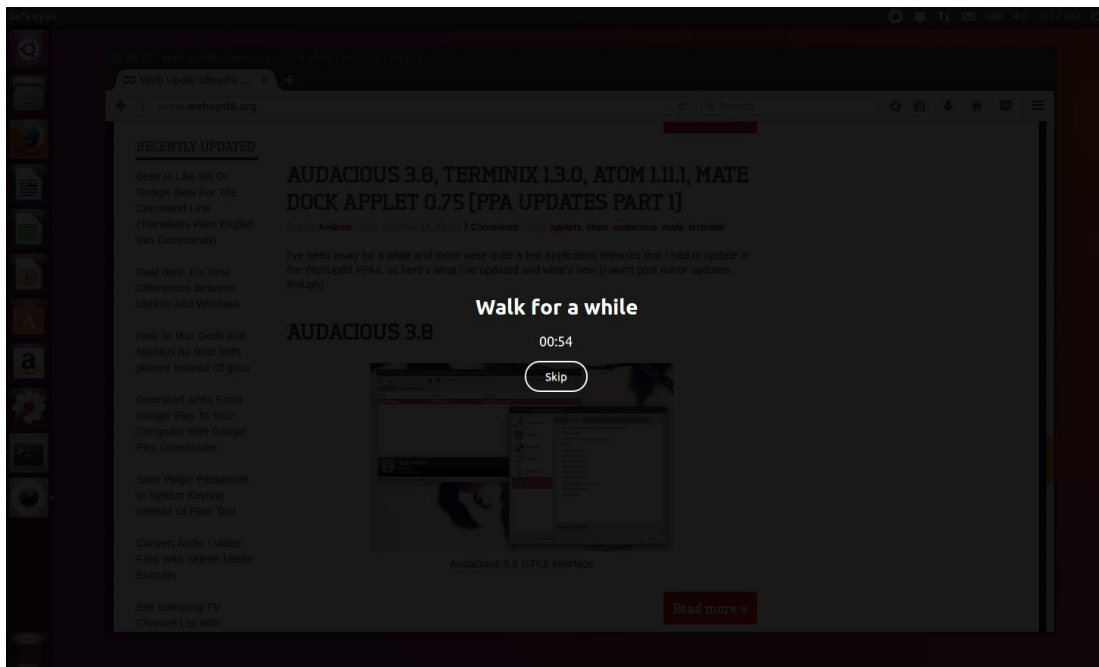
Safe Eyes was started as an open source alternative to Eye Leo and developed so far with numerous features of these features protect your eyes from eye strain using this simple and beautiful, yet extensible break reminder. A free and open source Linux alternative to EyeLeo [5].

### Features of SafeEyes:

- Short and Long breaks with eye exercises.
- Disable the keyboard during break.
- SafeEyes supports multi language.
- Strict break that does not allow skipping the breaks.
- Do not disturb when working with full-screen applications (Eg: Watching movies).

### Limitation of SafeEyes:

- SafeEyes work only on linux operating system.
- Not Work on Wayland Display Server
- Packages only Debian (.deb) and source code.
- SafeEyes features limited in Eye break, notification and eye exercises.



*Figure 2-2 SafeEyes screenshot*

### 2.1.3 Stretchly

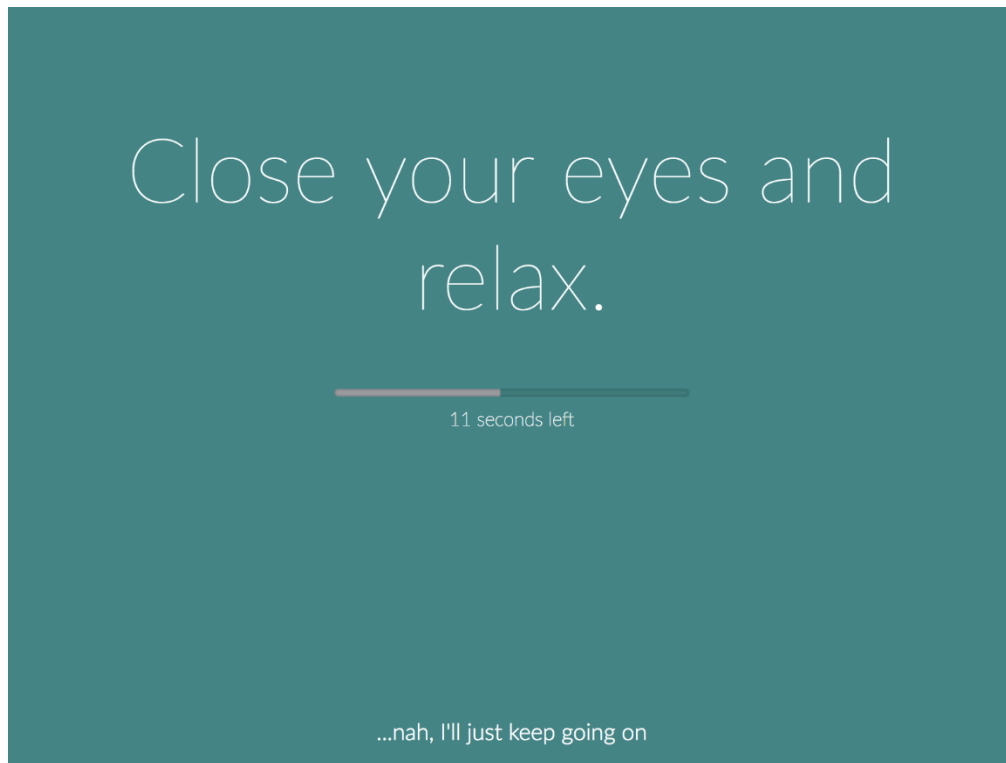
Stretchly is cross-platform electron app that reminds you to take breaks when working with computer[6].

#### Features of Stretchly:

- Short and long break
- Break time reminder app.
- Pause/resume reminding of breaks.
- Choose from different color schemes.
- Set break window or full screen.
- Cross-platform

#### Limitation of Stretchly:

- No localization supports.
- No Strict mode.



---

*Figure 2-3 Stretchly screenshot*

---



## 2.1.4 COMPARISON BETWEEN THE SYSTEMS

Features Comparison	EyeLeo	SafeEyes	stretchly	Healthy Guide
Short Break	Yes	Yes	Yes	Yes
Long Break	Yes	Yes	No	Yes
Eye exercises	Yes	Yes	No	Yes
notifications	No	Yes	Yes	Yes
Disables keyboard	No	Yes	No	No
Do not disturb	Yes	Yes	Yes	Yes
Strict mode	No	Yes	No	Yes
Change Break Color	No	No	Yes	Yes
Focus Sounds	No	No	No	Yes
Detect safety of sitting	No	No	No	Yes
Daily Health Tips	No	No	No	Yes
Technical Comparison				
Platform	Windows only	Linux only	Cross-Platform	Cross-Platform
Packages	EXE	Deb Only	All Type	All Type
LICENSE	freeware	Gnu	Open source	Open source
Programming language	C	Python	Electron (JS)	Java (JavaFX)

*Table 2-1 COMPARISON BETWEEN THE SYSTEMS*

In the above table, all the applications are common in eye break which the main subject of them. In our application the main advantage is calculate distance between user and computer, provide collect of relaxation sounds tracks suitable for user convenience and focus. Other advantage, our application t has provided set of daily medical advice-exercise. But our application does not have a mobile app which included in future work [4,5,6].

## 2.2 Choice of Technology

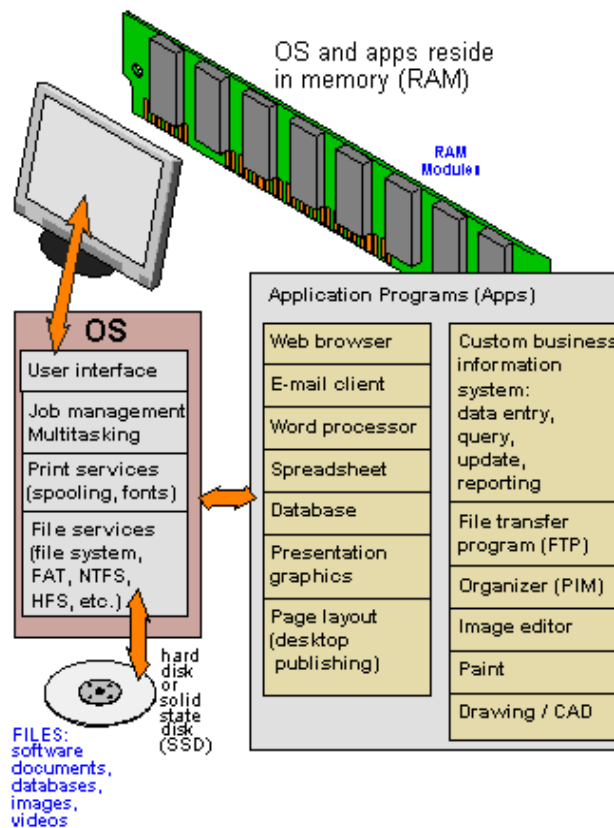
This section shows what are the technologies choices that are used to build the ‘Healthy Guide’ application and the reasons for those choices.

### 2.2.1 Desktop Application

An application program (app or application for short) is a computer program designed to perform a group of coordinated functions, tasks, or activities for the benefit of the user[7].

#### How desktop application work

This diagram shows the typical application programs that run in a desktop computer. It also shows the major components of the operating system.



*Figure 2-4 How Desktop Application Work*

In computers, application is installed and stored on the hard disk. As hard disk is a non-volatile memory, application does not lose on the turn off. But as the data access from the hard disk is very slow just after the computer is started OS is copied application into RAM from the hard disk. So instead of accessing OS from the hard disk, CPU directly access it from the RAM.

## Why Desktop Application

- Desktop applications are more security it provides total control over the standalone applications and protect it from various weakness.
- Desktop applications are available anytime offline and provide you with faster response and rich user experience.
- People feel more secure about sensitive data and the code is open source.

### 2.2.2 Java (programming language)

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation [8, 9].

The advantages of Java are as follows:

- Java is easy to learn.
- Java is platform-independent.
- Java is object-oriented.
- This allows you to create modular programs and reusable code.
- Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages.

### 2.2.3 JavaFx

JavaFX is a software platform for creating and delivering desktop applications, as well as rich internet applications (RIAs) that can run across a wide variety of devices. JavaFX is intended to replace Swing as the standard GUI (graphical user interface) library for Java SE, but both will be included for the foreseeable future. JavaFX has support for desktop computers and web browsers on Microsoft Windows, Linux, and macOS [10].

#### 2.2.3.1 FXML

FXML is an XML-based language that provides the structure for building a user interface separate from the application logic of your code. This separation of the presentation and application logic is attractive to web developers because they can assemble a user interface that leverages Java components without mastering the code for fetching and filling in the data.

From a Model View Controller (MVC) perspective, the FXML file that contains the description of the user interface is the view. The controller is a Java class, optionally implementing the Initializable class, which is declared as the controller for the FXML file. The model consists of domain objects, defined on the Java side, that you connect to the view through the controller.

While you can use FXML to create any user interface, FXML is particularly useful for user interfaces that have large, complex scene graphs, forms, data entry, or complex animation. FXML is also well-suited for defining static layouts such as forms, controls, and tables. In addition, you can use FXML to construct dynamic layouts by including scripts[11].

The advantages of FXML are as follows:

- The content of an FXML file can be localized as the file is read.
- It is easy for a development team to create and maintain a testable user interface.
- You can use FXML with any Java Virtual Machine (JVM) language.
- FXML is not a compiled language; you do not need to recompile the code to see the changes.
- FXML is not limited to the view portion of the MVC interface. You can construct services or tasks or domain objects, and you can use JavaScript or other scripting languages in FXML.

### **2.2.3.2 JavaFX and CSS**

JavaFX Cascading Style Sheets (CSS) is based on the W3C CSS version 2.1 with some additions from current work on version 3. JavaFX CSS also has some extensions to CSS in support of specific JavaFX features. The goal for JavaFX CSS is to allow web developers already familiar with CSS for HTML to use CSS to customize and develop themes for JavaFX controls and scene graph objects in a natural way [12, 13].

The advantages of CSS are as follows:

- Easier to maintain and update
- Greater consistency in design
- More formatting options
- Lightweight code
- Ease of presenting different styles to different viewers
- Greater accessibility

## **2.3 Google Cloud Vision API**

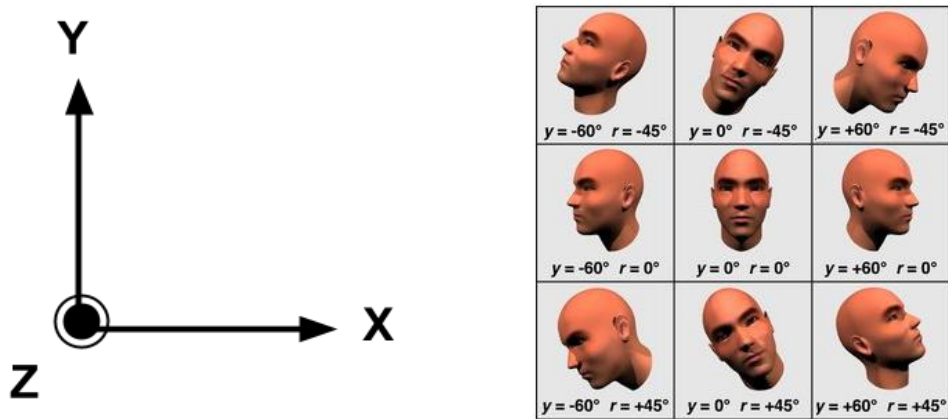
Google Cloud Vision API enables developers to understand the content of an image by encapsulating powerful machine learning models in an easy to use REST API. It quickly classifies images into thousands of categories (e.g., "sailboat", "lion", "Eiffel Tower"), detects individual objects and faces within images, and finds and reads printed words contained within images. You can build metadata on your image catalog, moderate offensive content, or enable new marketing scenarios through image sentiment analysis. Analyze images uploaded in the request or integrate with your image storage on Google Cloud Storage [14].

## Face Detection Concepts Overview

Face detection is the process of automatically locating human faces in visual media (digital images or video). A face that is detected is reported at a position with an associated size and orientation. Once a face is detected, it can be searched for landmarks such as the eyes and nose [15].

### Face Orientation:

The face API detects faces at a range of different angles, as illustrated below:



*Figure 2-5 Pose angle, Coordinate system*

Fig. 2.2. Pose angle estimation. The coordinate system with the image in the XY plane and the Z axis coming out of the figure 2.2, Pose angle examples where  $y = \text{Euler Y}$ ,  $r = \text{Euler Z}$ .

The **Euler X**, **Euler Y**, and **Euler Z** angles characterize a face's orientation as shown in Fig. 2.2. The Face API provides measurement of Euler Y and Euler Z (but not Euler X) for detected faces. The Euler Z angle of the face is always reported. The Euler Y angle is available only when using the "accurate" mode setting of the face detector (as opposed to the "fast" mode setting, which takes some shortcuts to make detection faster). The Euler X angle is currently not supported [15].

## 2.4 Conclusion:

In summary, this chapter of literature review compares between our project and related work to find gaps for these apps to avoid it and to create new features that meet the application goals. That has identified the approaches to be used for developing a desktop application.

# 3 CHAPTER 3 RESEARCH METHODOLOGY

This chapter introduces the selected methodologies, reasons for the selections, how it works, and explanation of the detailed phases of the project and its life cycle.

Software development life cycle (SDLC) can be defined as “is a splitting of software development work into distinct phases (or stages) containing activities with the intent of better planning and management” [16], these activities were built to provide a high quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates. Among the people concerned with agility in software development over the last decade, Agile Software Development story is talking about relatively light, joined efforts, effective, and human-powered software development techniques [17].

The study uses agile software development that uses iterative development as a basis but advocates a lighter and more people-centric viewpoint than traditional approaches. Agile processes fundamentally incorporate iteration and the continuous feedback that it provides to successively refine and deliver a software system during the Project Life Cycle (PLC).

### **3.1 Project Methodology:**

Extreme Programming (XP) is Agile-based methodologies and is a pragmatic approach to program development that emphasizes business results first and takes an incremental, get-something-started approach to building the application, using continual testing and revision.

#### **XP Methodology Reasons for Adoption and its Limitations:**

Extreme Programming (XP) methodology has many advantages to adopt the methodology for software development. Also there are some limitations for using it in some kind of software projects: [18]

#### **Reasons for Adopting XP Methodology:**

The advantages of adopting XP methodology are:

- XP is allow software development companies to save costs and time required for project realization.
- XP phases are carried out in extremely small (or "continuous"). The first pass through the steps might take a day or a week.
- XP is efficient, low-risk, flexible, predictable, scientific and fun way to develop software.
- XP is intended to improve software quality and responsiveness to changing customer requirements.
- XP is continuous testing and integration helps to increase the quality of your work
- XP improves a software project in five essential ways; communication, simplicity and feedback.

### XP Methodology Limitations:

The limitations of choosing XP methodology are:

- XP is hard to do, it's require get many professional developers.
- XP is code centric rather than design centric development.
- XP programming is not structured, it may be difficult to find significant number of defects for testers by just mere looking at the screen.

### 3.2 Extreme Programming Phases

Extreme Programming can be implemented by a disciplined methodology that focuses primarily on customer satisfaction. As all other software engineering methods, the process in XP is similar to the process used elsewhere. In fact, we can divide the above discussed practices into four clear progressions that are the planning, the design, the coding and the testing, as shown in Figure (3.1).

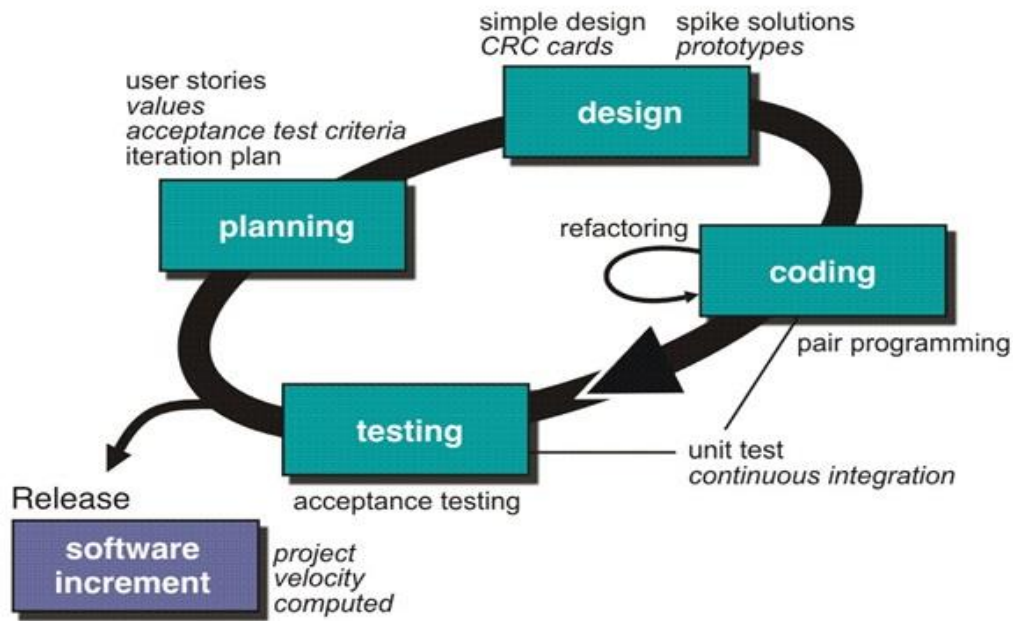


Figure 3-1 Extreme Programming phases



### **3.2.1 Planning:**

The team is gathering the requirements from users to identify about how the system works and what the gaps should the application fill to building an effective and useful application. Then, the information has been made available, but this information must be based on medical information. The team should interview specialist doctors to rely on our application for reliable information to find out what are the necessary guidelines that should be available to maintain eye and human spine integrity. See appendix a,b.

### **3.2.2 Design:**

We start development app by drawing an initial visualization for interface of the application, then design of Sketch (UI) and review of the Sketch get to the best result, finally design Sketch by FXML and CSS and choose the best suited for UX.

### **3.2.3 Coding:**

The next set of application is coding by converting analysis and design models to actual lines-of-code, the coding creates by modern techniques and programming language to achieve goals, after finishing designing phase. During this phase coding using JavaFx Programming language for backend, FXML and CSS for user Interface and maven for projects build, dependency by using project object model (POM).

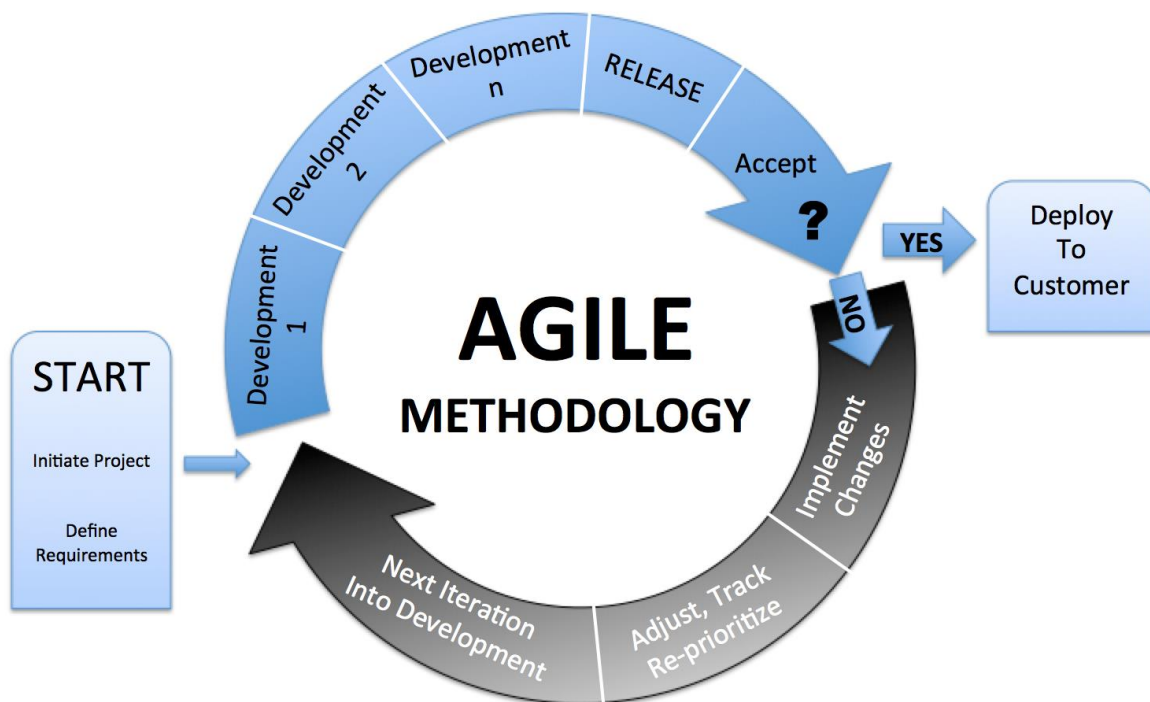
### **3.2.4 Testing:**

Test-first development is one of the most important innovations in XP. Instead of writing some code and then writing tests for that code, you write the tests before you write the code. This means that developer can run the test as the code is being written and discover problems during development.

Testing the application uses pair programming principle but in testing, that means each two developers in app form a set, each one in the set tests the other developer code and vice versa. the team test the app features stage by stage, like take a picture to calculate the distance, testing eye health break, focus sounds and daily tips.

### 3.3 Agile methodology

The Agile methodology is an iterative approach is performed in a collaborative environment by self-organizing teams. The methodology produces high-quality software in a cost-effective and timely manner to meet stakeholders' changing need that occur within the context of six framework activities: requirement, planning, design, developing, release and tracking & monitor software system



*Figure 3-2 Agile Planning Process*

#### 3.3.1 The advantages of agile methodology:

- ✓ Agile provides multiple opportunities for stakeholder and team by involving the client in every step of the project and delivering working software early.
- ✓ Agile approach provides a unique opportunity from prioritizing features to iteration planning and review sessions to frequent software builds containing new features.
- ✓ The agile process is broken into individual models that designers work on.
- ✓ The customer has early and frequent opportunities to look at the product and make decision and changes to the project.
- ✓ At the end of every sprint, user acceptance is performed.

### **3.3.2 The Limitations of agile methodology:**

- ✓ Agile model is considered unstructured compared to the waterfall model.
- ✓ Small projects can be implemented very quickly. for large projects, it is difficult to estimate the development time.
- ✓ Agile methodologies is that it is not suitable for maintenance, since there is not much documentation for the system.
- ✓ Agile methodologies depend heavily on the user involvement, and thus, the success of the project will depend on the cooperation and communication of the user.

### **3.4 Conclusion:**

This chapter choose carefully the methodology that will help us to develop the project perfectly and leading it to the success. According to the project objectives and the advantages and limitation of each methodology we decide to use the agile model which is the most appropriate methodology for our project because of the reasons we discussed in this chapter.

Development team attempt as much as possible to commit with design rules, standards of user Interfaces (UI) as well as code style, convention and readable code comments.

# 4 Chapter 4 Prototype Design and Development

This chapter demonstrates a full description of ‘Healthy Guide’ and its users, the functional and non-functional requirements, the requirements analysis tools, display the design stage activities like: user interfaces.

After all design stages complete implementation starting using development tools, how the testing and evaluation process will be implemented for ‘Healthy Guide’ application.

## 4.1 System Description

‘Healthy Guide’ is a desktop application that helps people to organize their eye health and backbone health.

The app consists of four basic functions including:

- Eye safe
- Health session.
- Focus sounds.
- Advise center.

**The Eye safe function:** It’s a function that stop the screen for 10 minutes by showing a timer notification, the user can change time of break and when its shown.

**The Health Sitting function:** It’s a function that reminder the user when his session is not in healthily way by count the distance between user face and the screen by using webcam.

**The Focus Sounds function:** It’s a function that let user being in relaxing mode

**The Advice Center function:** It’s a function that display an advice for the user when application is running.

## 4.2 User Description

‘Healthy Guide’ is an application that any user using PC or Laptop can use it.

## 4.3 System Requirements

For any system, there are functional and non-functional requirements to be considered while determining the requirements of the system, the functional requirements are user visible features that are typically initiated by stakeholders of the system, such as generate report, login, and signup. On the other hand, nonfunctional requirements are requirements that describe how the system will do what it is supposed to do, for example, security, reliability and maintainability.

### 4.3.1 What is a Requirement?

Requirements are the necessary attributes in the system, they are the statement that identifies a capability, characteristic or quality factor of the system in order to have value and utility to the users. Once the requirements are defined, then the developers can initiate the other technical work including system design, development, testing, implementation and operation.

### **4.3.2 Requirements Gathering**

In any technical project the requirements gathering is one of the important parts because it provides full understanding of what the project will do, so to gather requirements we have looked on other similar apps and decided what is the most important and interesting to put in ours, also to make new contribution we have done brain storming to add new features.

### **4.3.3 Functional Requirements**

The functions that the user can do to use the app are:

- User can customize eye break preferences (specify time between break, break duration, enable or disable short or long break and notification).
- The system will take a picture using webcam every 1 hour and determine if the user is far or close to screen and display a warning notification.
- Also user can customize the time when system take picture and enable or disable features.
- To improve and helps user to focus, relax, and sleep. User can select to listen to several sounds like rain, wind, shop and white noise.
- The application will run at background and user can view and customize setting using the application tray icon (view application windows, enable or disable EyeBreak, focus sound).

### **4.3.4 Non-Functional Requirements**

The non-functional requirements can be divided into several categories, the four basic categories are the operational, performance, security, and cultural requirements.

The operational requirements are the physical and technical environment in which the app will operate. The performance requirements are the speed, capacity, and reliability of the app.

The security requirements are the determining who has authorization to access the app and under what circumstances. The cultural requirements are the cultural factors that effect on the app.

#### **4.3.4.1 Operational Requirements**

The 'Healthy Guide' should work on any operating system.

#### **4.3.4.2 Performance Requirements**

The speed of doing operations must be in consideration, performance requirements include response time.

#### **4.3.4.3 Security Requirements**

It's important to insure the access of authorized users to the system and to insure that the data of the system are private can't be hacked or damaged. Security requirements include integrity, authorization, availability, confidentiality and authentication.

#### **4.3.4.4 Cultural Requirements**

The 'Healthy Guide' should serve all interested users, so it must be delivered in English language because it is a universal language.

### **4.4 System Analysis**

This part contains the analysis of the functional and non-functional requirements using

### **4.5 Application Requirements:**

All computer software needs certain hardware components or other software resources to be presented on the device. Most software defines two sets of system requirements: minimum and recommended it. With increasing demand for higher processing power and resources in newer versions of software and system requirements tend to increase over time.

#### **4.5.1 Software Requirements**

Software requirements, which were used during the development processing:

- ✓ Operating system (Windows, MacOS, Linux).
- ✓ Java 1.8 Installed

#### **4.5.2 Hardware Requirements**

Hardware requirements, which were used during the development processing:

- ✓ Webcam.
- ✓ Free memory 20 MB.
- ✓ Hard Disk 120 MB Max.
- ✓ Application is compatible with different screen sizes.

### **4.6 Development Requirements:**

Tools used to develop the system:

#### **4.6.1 Software Requirements**

- ✓ Java 1.8 Installed.
- ✓ Scene Builder.
- ✓ IntelliJ idea IDE Professional 2017
- ✓ Operating system (Windows, MacOS, Linux).

#### **4.6.2 Hardware Requirements**

- ✓ 4 GB of Ram.
- ✓ Intel Core i5 Cpu or equivalent.
- ✓ webcam.

## **4.7 Requirements Analysis**

Requirement analysis uses a combination of text and diagrammatic forms to depict requirement in a way that is relatively easy to understand.

### **4.7.1 Use Case**

#### **4.7.1.1 Main System Actors**

‘Healthy Guide’ application will have only one main actor (user) that will handle with the application to protect the eye and spine.

#### **4.7.1.2 Use case Diagrams**

A use case diagram is a static description of some way in which the application is used. This diagram shows how the application use cases are related to each other and how the users can get at them[19]. Each bubble represents a use case, and each actor person represents a user, system has one main actors which have some different processes, here we describing as the following UML description. Describes all the Process and relations with the users are using the project in the analysis stage until it is very clear image of the subject put before starting the design phase.

#### **4.7.1.3 User Use Case Diagram**

After running the ‘Healthy Guide’, the startup screen shows Four options first is Eye Break shows the Eye Break Preferences and allow the User to customize the Break, The second option is Healthy Sitting shows the Healthy Sitting Preferences and allow User to Train The Application to his face, third is Improve Focus Shows a list of nature Sounds that help user to Focus, Relax and sleep, finally Advice Center shows the Healthy advice included in application and allow user to customize advice Preferences.



Figure 4.1 show the Use Case Diagram.

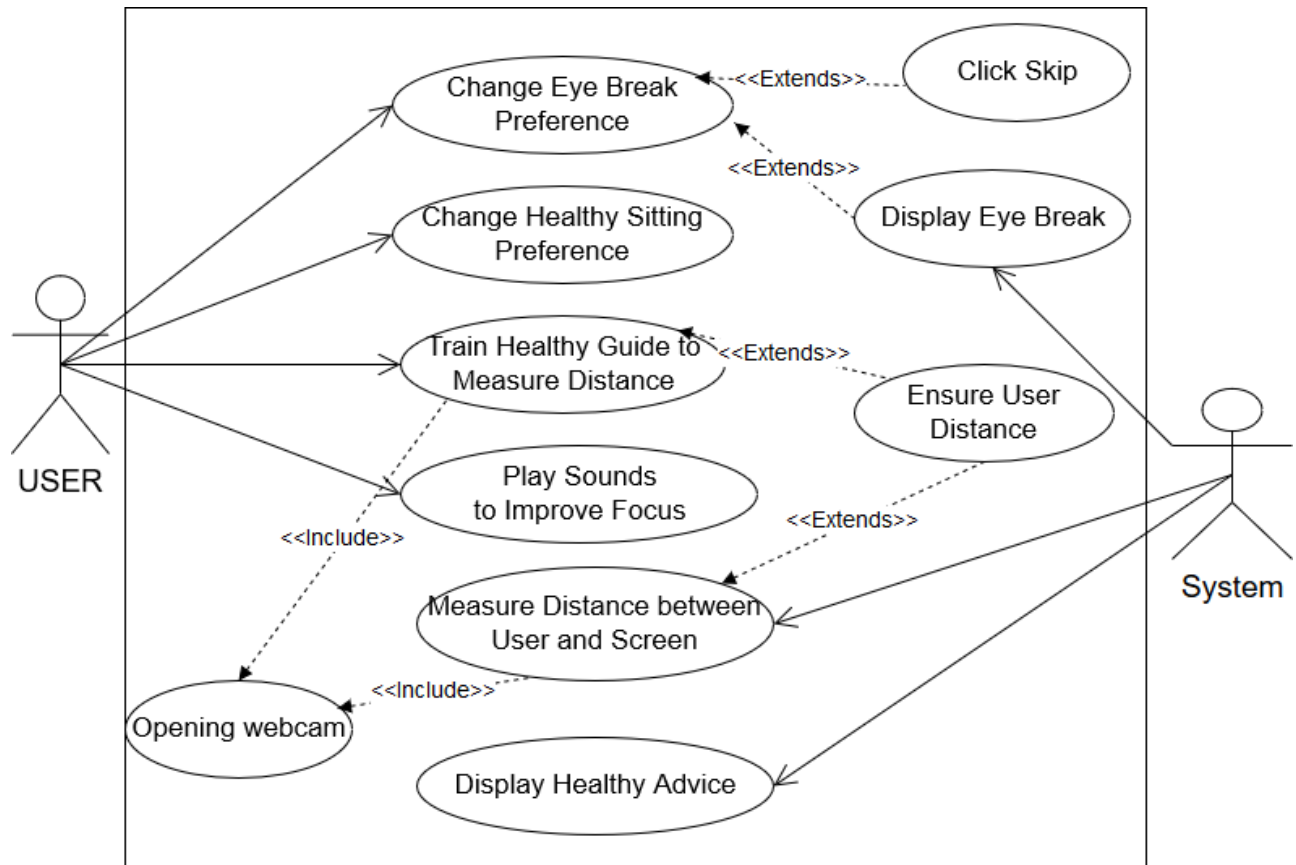


Figure 4-1 Use Case Diagram

### 4.7.2 Usecase List:

Table 4.2 usecases list, list the usecases that will use in the usecase diagrams and with its arrangement.

Use case List	
U1	Change eye break preference
U2	Change healthy sitting preference
U3	Train 'Healthy Guide' to measure distance
U4	Play sounds to improve focus
U5	Display eye break
U6	Click skip
U7	Measure distance between user and screen
U8	Display healthy advice

*Table 4-1 usecases list*

### 4.7.3 Usecases Details:

Usecases details is a detailed description for each use case, such as relations, actors of Usecase, precondition, post-condition, normal path and abnormal path and assumptions.

U1: Change Eye Break Preference	
<b>Actors</b>	User
<b>Description</b>	user want to change his preference of eye break UI
<b>Precondition</b>	user access the application
<b>Post-condition</b>	save new preference
<b>Normal Path</b>	<ol style="list-style-type: none"><li>1. User choose the option need to changed</li><li>2. User edit the options</li><li>3. User save the new preference</li><li>4. User can test break UI</li></ol>
<b>Abnormal Path</b>	User don't click on apply button to save preference
<b>Relations</b>	--

*Table 4-2 Usecase Details for Change Eye Break Preference*

<b>U2: Change Healthy Sitting Preference</b>	
<b>Actors</b>	User
<b>Description</b>	User want to change his preference for healthy sitting
<b>Precondition</b>	User access the application
<b>Post-condition</b>	Save new preference
<b>Normal Path</b>	<ol style="list-style-type: none"> <li>1. User choose the option need to changed</li> <li>2. User edit the options</li> <li>3. User save the new preference</li> </ol>
<b>Abnormal Path</b>	User don't click on apply button to save preference
<b>Relations</b>	--

*Table 4-3 Usecase Details for Change Healthy Sitting Preference*

<b>U3: Train 'Healthy Guide' to Measure Distance</b>	
<b>Actors</b>	User
<b>Description</b>	User want to train the application to his face to get the higher accuracy when measure distance
<b>Precondition</b>	User access the application internet connection webcam device
<b>Post-condition</b>	measure distance between user and screen then store it
<b>Normal Path</b>	<ol style="list-style-type: none"> <li>1. User click on the train button</li> <li>2. System opening webcam and take a picture</li> <li>3. System send the picture to Google Vision API and Detect the user face</li> <li>4. System calculate the face area</li> <li>5. User choose his position (Far, Normal, Close).</li> <li>6. System compare the area and save the new value to use it on next distance measurement</li> </ol>
<b>Abnormal Path</b>	<ul style="list-style-type: none"> <li>- error opening webcam</li> <li>- no internet connection to use Google Vison API</li> </ul>
<b>Relations</b>	--

*Table 4-4 Usecase Details Train 'Healthy Guide' to Measure Distance*

<b>U4: Play Sounds to Improve Focus</b>	
<b>Actors</b>	User
<b>Description</b>	user want to relax, sleep or improve focus by listening to natural sounds.
<b>Precondition</b>	user access the application
<b>Post-condition</b>	application playing sounds
<b>Normal Path</b>	<ol style="list-style-type: none"> <li>1. user choose the sounds.</li> <li>2. the sounds playing.</li> </ol>
<b>Abnormal Path</b>	--
<b>Relations</b>	--

---

*Table 4-5 Usecase Details for Play Sounds to Improve Focus*

---

<b>U5: Display Eye Break</b>	
<b>Actors</b>	System
<b>Description</b>	System display the eye break screen
<b>Precondition</b>	User access the application
<b>Post-condition</b>	Start eye break screen
<b>Normal Path</b>	<ol style="list-style-type: none"> <li>1. system check the break preferences</li> <li>2. the Eye Break Screen Start and appears</li> </ol>
<b>Abnormal Path</b>	<ul style="list-style-type: none"> <li>- feature is disabled by the user</li> <li>- eye break pause temporary</li> </ul>
<b>Relations</b>	extended by U1

---

*Table 4-6 Usecase Details for Display Eye Break*

---

<b>U6: Click Skip</b>	
<b>Actors</b>	User
<b>Description</b>	user want to skip the eye break screen and continues his work
<b>Precondition</b>	user access the application the eye break screen is appearing
<b>Post-condition</b>	close eye break screen
<b>Normal Path</b>	<ol style="list-style-type: none"> <li>1. user click on skip button</li> <li>2. the eye break screen disappears</li> </ol>
<b>Abnormal Path</b>	If strict mode is enable user can't use skip button
<b>Relations</b>	extended by U5

---

*Table 4-7 Usecase Details for Click Skip*

---

<b>U7: Measure Distance between User and Screen</b>	
<b>Actors</b>	System
<b>Description</b>	System measure distance between user and screen and displays a warning to the user
<b>Precondition</b>	User access the application internet connection webcam device
<b>Post-condition</b>	measure distance between user and screen
<b>Normal Path</b>	<ol style="list-style-type: none"> <li>1. System check the preferences</li> <li>2. System opening webcam and take a picture</li> <li>3. System send the picture to Google Vision API and detect the user face</li> <li>4. System calculate the face area</li> <li>5. Display alert for the user to improve his seat</li> </ol>
<b>Abnormal Path</b>	<ul style="list-style-type: none"> <li>- feature is disabled by the User</li> <li>- error opening webcam</li> <li>- no Internet connection to use Google Vison API</li> </ul>
<b>Relations</b>	--

---

*Table 4-8 Usecase Details for Measure Distance between User and Screen*

---

<b>U8: Display Healthy Advice</b>	
<b>Actors</b>	System
<b>Description</b>	System display daily healthy advice to the user
<b>Precondition</b>	User access the Application
<b>Post-condition</b>	Display healthy Advice
<b>Normal Path</b>	<ol style="list-style-type: none"> <li>1. System check the preferences</li> <li>2. Get random advice</li> <li>3. Display advice as information alert</li> </ol>
<b>Abnormal Path</b>	- Feature is disabled by the user
<b>Relations</b>	--

---

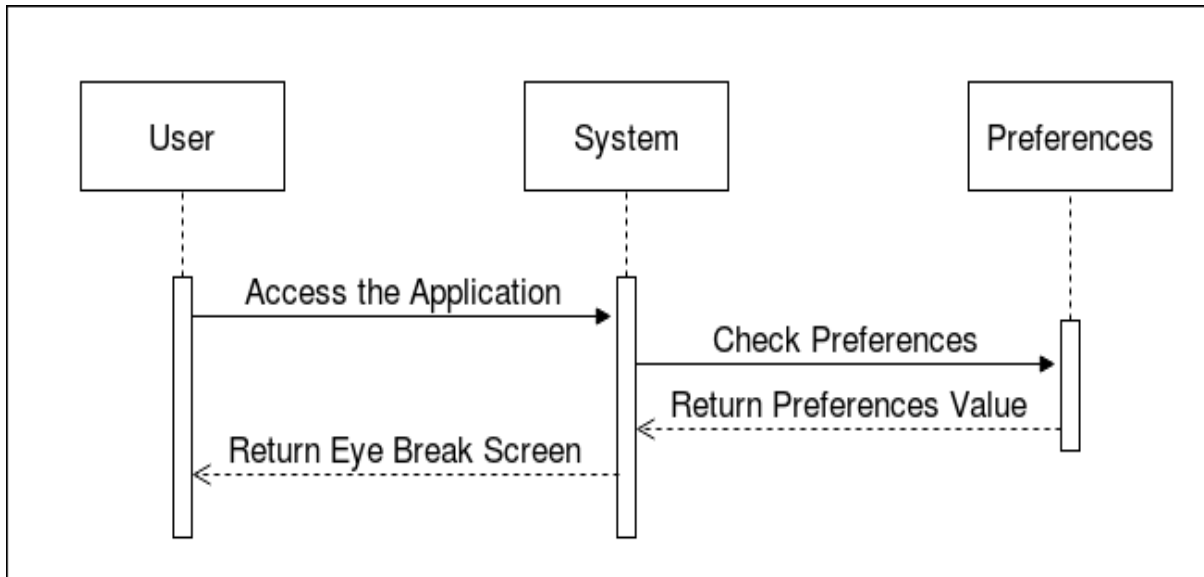
*Table 4-9 Usecase Details for Display Healthy Advice*

---

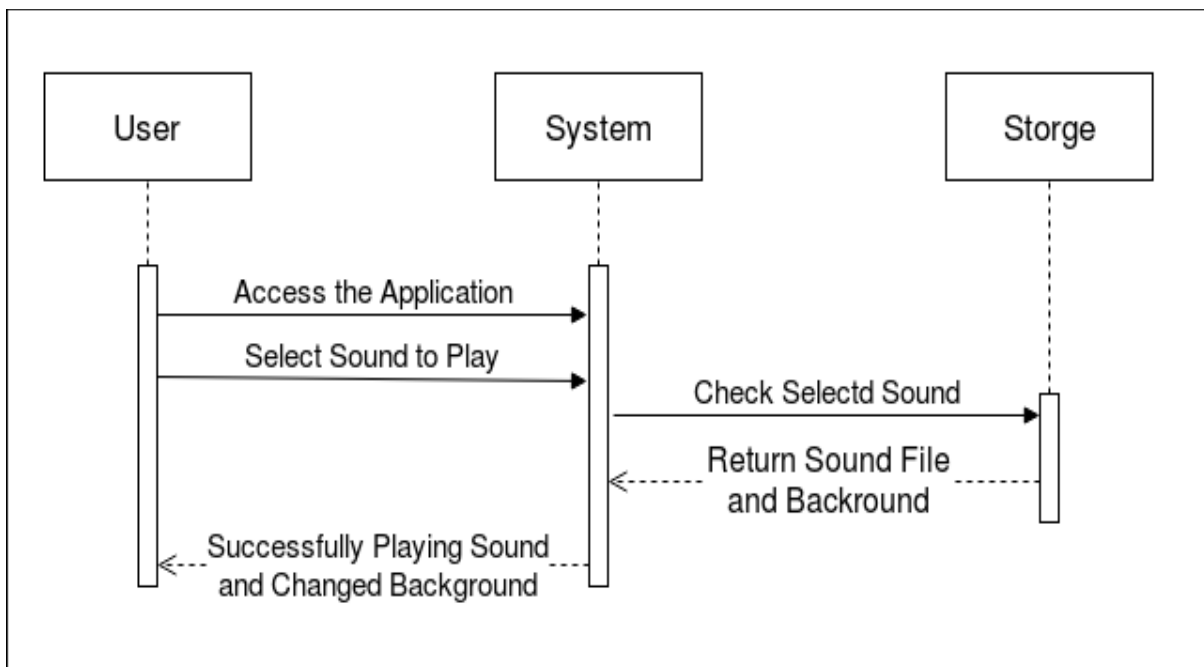
#### **4.7.4 Sequence Diagram:**

The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur. Much like the class diagram, developers typically think sequence diagrams were meant exclusively for them. However, an organization's business staff can find sequence diagrams useful to communicate how the business currently works by showing how various business objects interact. Besides documenting an organization's current affairs, a business-level sequence diagram can be used as a requirements document to communicate requirements for a future system implementation. During the requirements phase of a project, analysts can take use cases to the next level by providing a more formal level of refinement. When that occurs, use cases are often refined into one or more sequence diagrams[20].

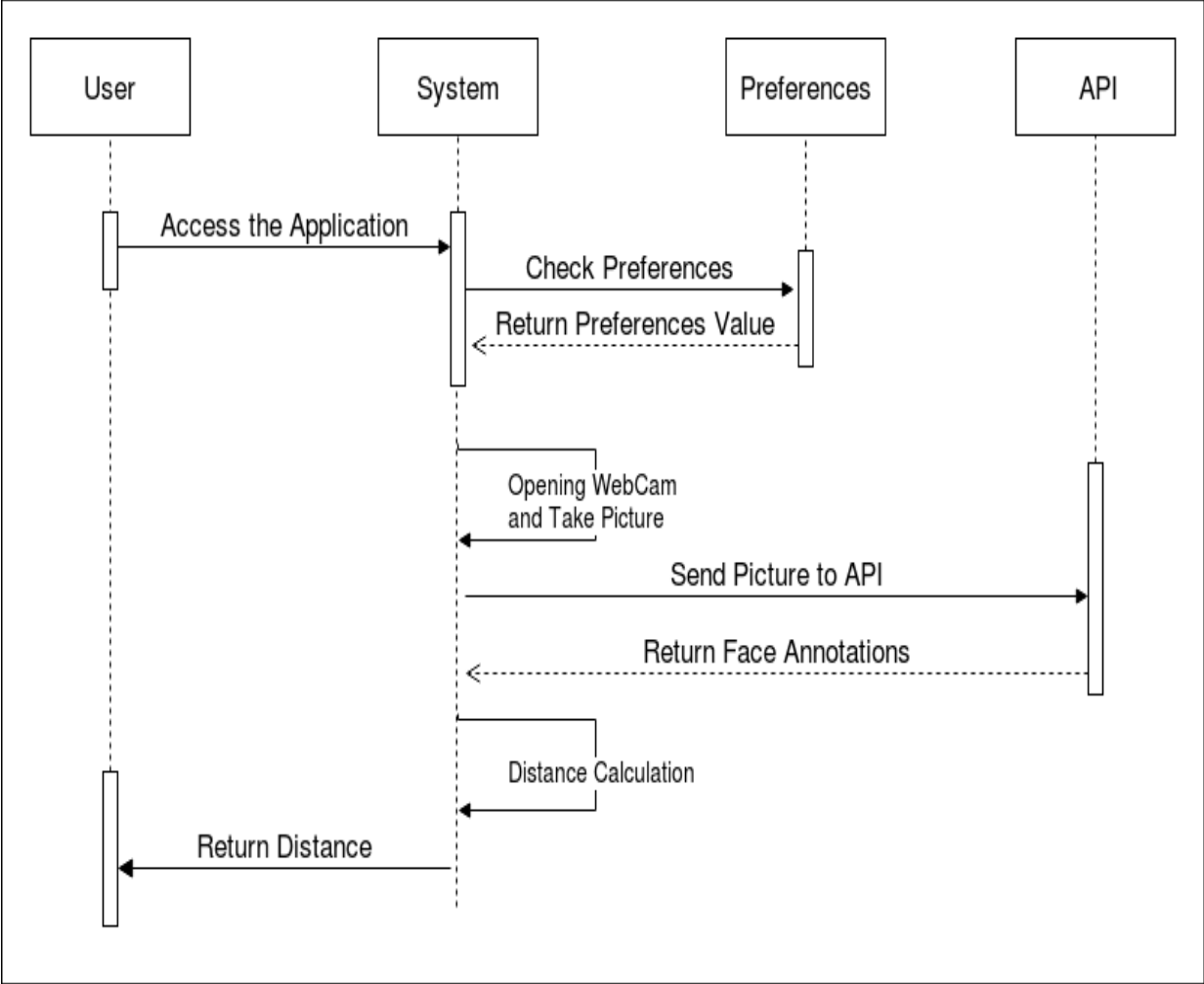
These are sample of sequence diagram for some interactions and operation:



*Figure 4-2 Sequence Diagram for Eye Break*



*Figure 4-3 Sequence Diagram for Select Sound*



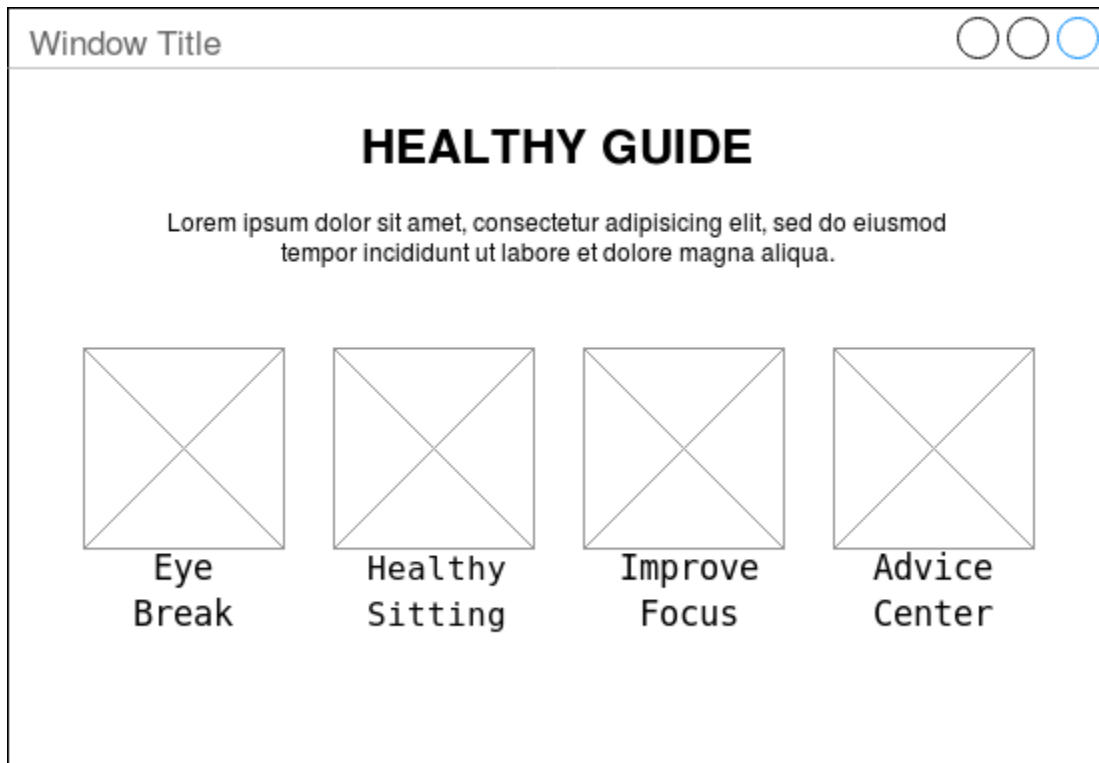
*Figure 4-4 Sequence Diagram for Calculation Distance*



## 4.8 Design

### 4.8.1 User Interface Sketch

Before start implementation and development phase we make sketch to user interface and these are some example of our project sketch's.



---

*Figure 4-5 Sketch Main Screen*

---

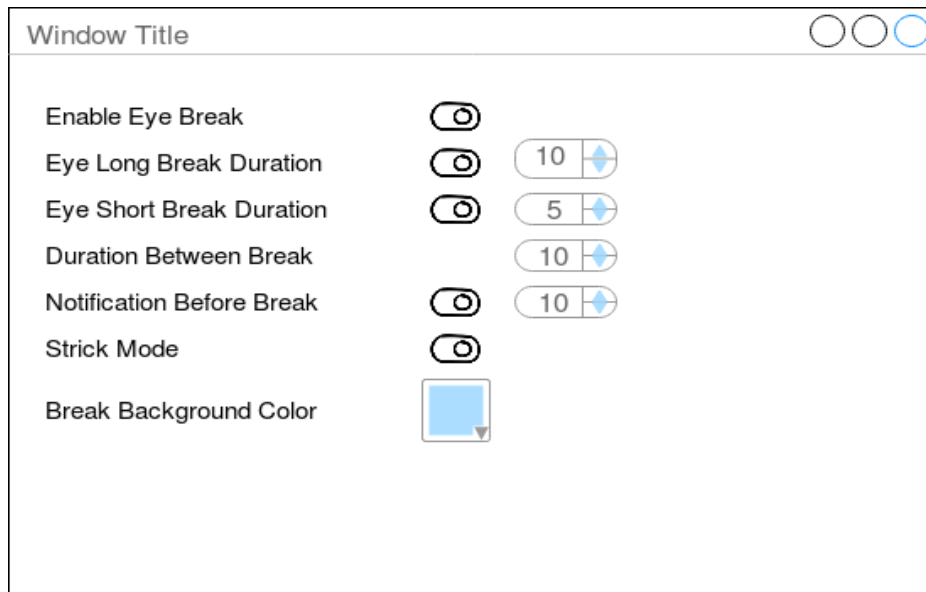


Figure 4-6 Sketch Eye Break Setting

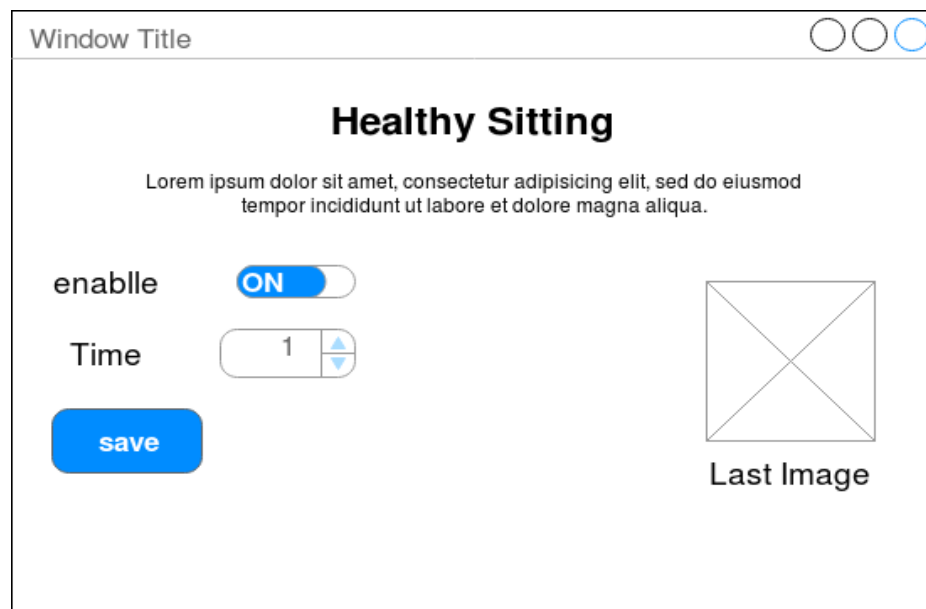
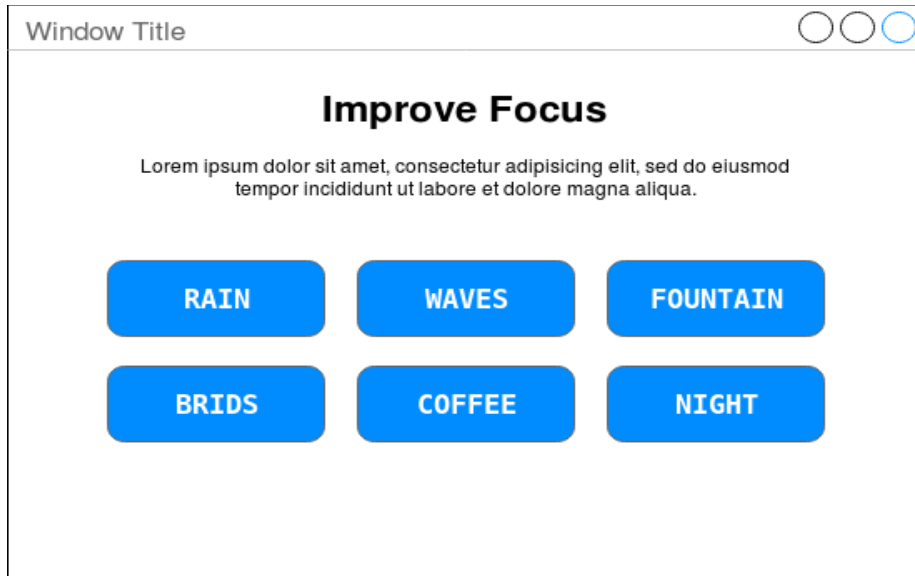
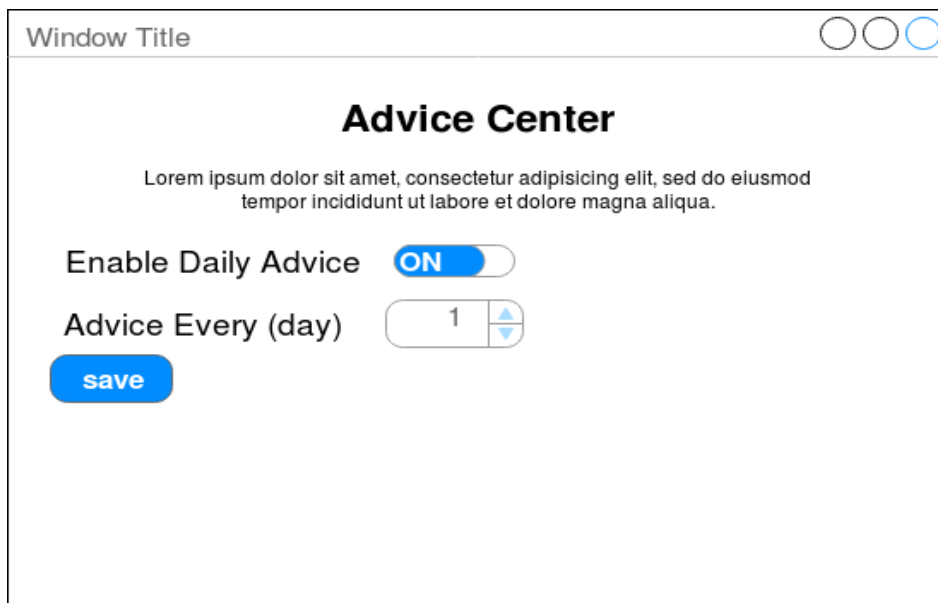


Figure 4-7 Sketch Healthy Sitting



*Figure 4-8 Sketch Improve Focus Screen*



*Figure 4-9 Sketch Advice Setting*

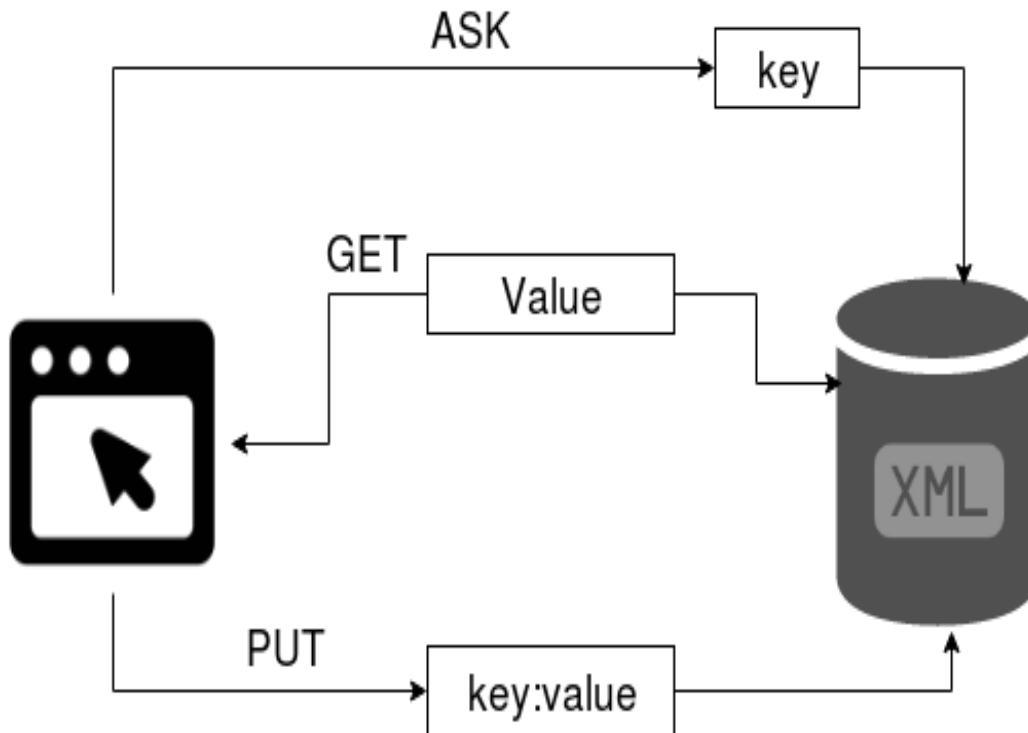


*Figure 4-10 Sketch Notification Before Break*

## 4.9 Storage Schema

'Healthy Guide' need to store user preferences like the break duration, color and enable or disable any features. Java allows us to save some application state using a class called preferences. Depending on the operating system, the Preferences are saved in different places (e.g. the registry file in Windows).

We use preferences to store application setting and user preferences as a key-value pair on a xml based file, preferences data persist on device reboot and are removed along with app uninstallation.



*Figure 4-11 Preferences Diagram*

### 5.3.1 Preferences key-value pair Description:

Key	Description	Type	Default Value
<b>isEnabled</b>	Enable Eye Break	Boolean	True
<b>isEnabledShort</b>	Enable Short Eye Break	Boolean	True
<b>isEnabledLong</b>	Enable Long Eye Break	Boolean	True
<b>NotificationIsEnabled</b>	Enable Notification	Boolean	True
<b>strictMode</b>	Enable strict Mode	Boolean	False
<b>shortBreak</b>	Duration of short break	Int	10 sec
<b>longBreak</b>	Duration of long break	Int	60 sec
<b>timeBetweenTwoBreak</b>	Time Between two short Break	Int	10 min
<b>showNotificationBefore</b>	Duration before break to get Notification	Int	10 sec
<b>BreakColor</b>	Eye Break Color	String	#000000
<b>isEnabledFace</b>	Enable take Picture to calculate the distance between Face and computer Screen	Boolean	True
<b>TimeToTakePic</b>	The Time Between two calculation	Int	1 hour
<b>isEnabledAdvice</b>	Enable advice	Boolean	True
<b>TimeToAdvice</b>	Time to get advice	Int	1 day

*Table 4-10 Preferences key-value pair*

## 4.10 Implementation

### 4.10.1 Overview of implementation phase

The implementation phase takes the requirements and design phase product and implements then using appropriate technologies. In the case of validation testing, it is during this phase that test cases are completed and automated in preparation for validation testing. Typically, a lot of testing on the early system versions are also performed during this phase, not only to validate the system, but to validate that there are no problems with the test cases themselves. A program will write based upon the algorithm designed in the last phase (design). A piece of code is written for every module and checked for the output. We will use the JavaFX to develop the software. At this phase must match between the requirements and the output of the system to determine the verification for it. As mentioned, the application language will be Java because mostly applications are written in it.

### 4.10.2 Graphical user interface (GUI) of the Desktop Application

#### 1. Main Screen



*Figure 4-12 Main Screen GUI*

## 2. Healthy Sitting Screen

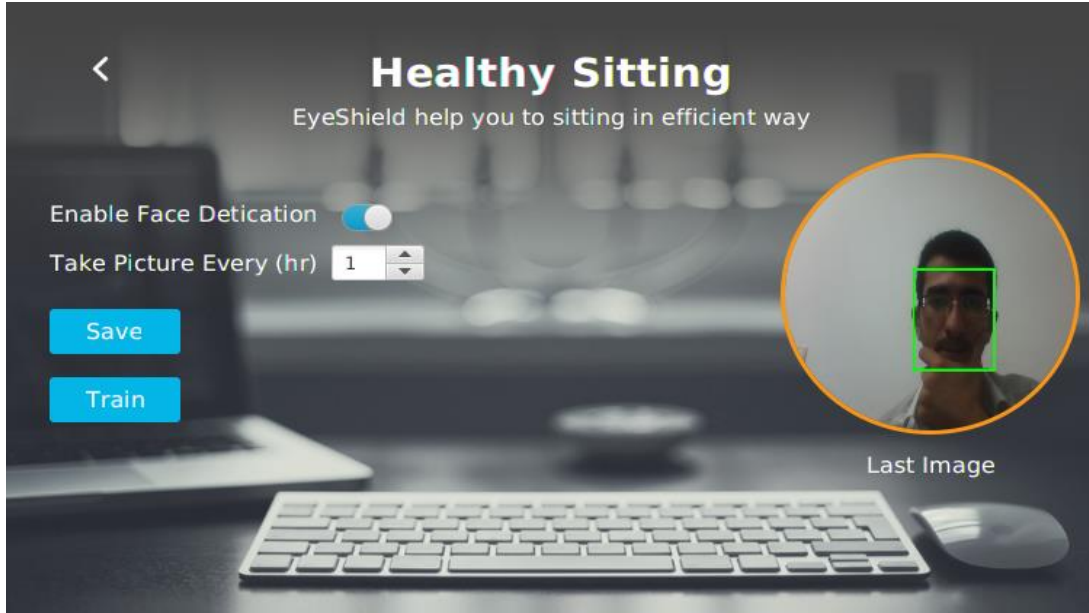


Figure 4-13 Healthy Sitting GUI

## 3. Healthy Sitting on training

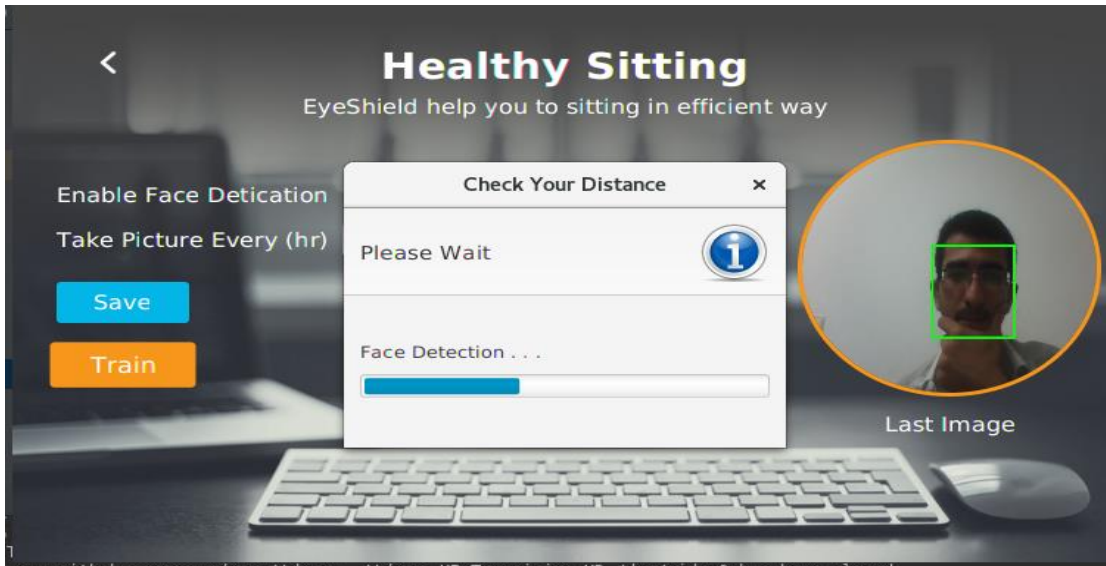


Figure 4-14 Healthy Sitting on Training Dialog

#### 4. Eye Break Setting Screen

Enable Eye Break

---

Eye Long Break Duration (Seconds)

Eye Short Break Duration (Seconds)

Duration Between Two Break (min)

---

Enable Notification Before Break

Show Notification Before (Seconds)

---

Strick Mode (Hide Skip Button)

---

Break Background Color

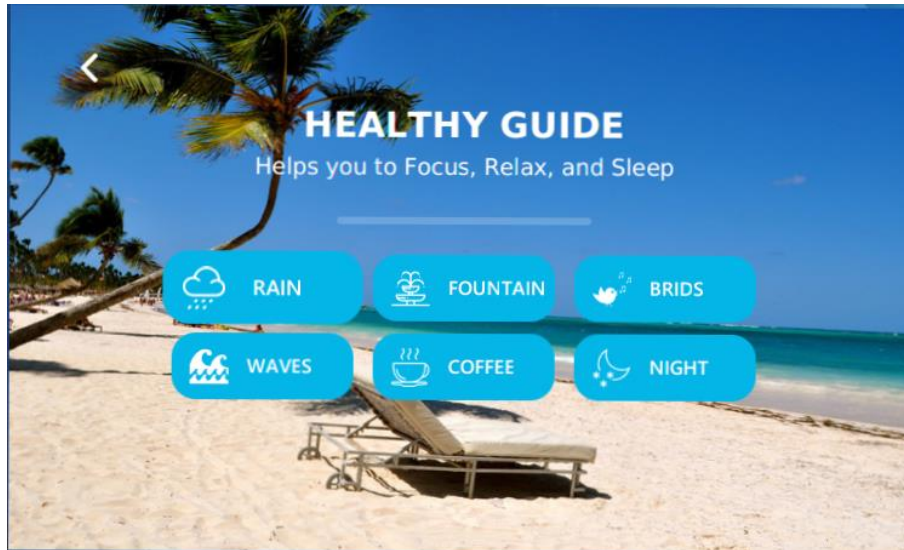
*Figure 4-15 Eye Break Setting*



## 5. Improve Focus Screen

The Background changed based on the sound playing

- **Waves sounds**

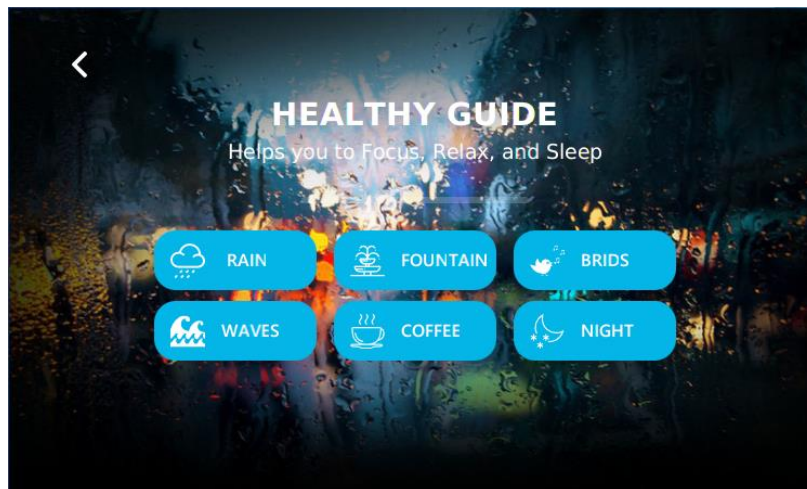


---

*Figure 4-16 Improve Focus Screen of Birds Sound*

---

- **Rain sound**

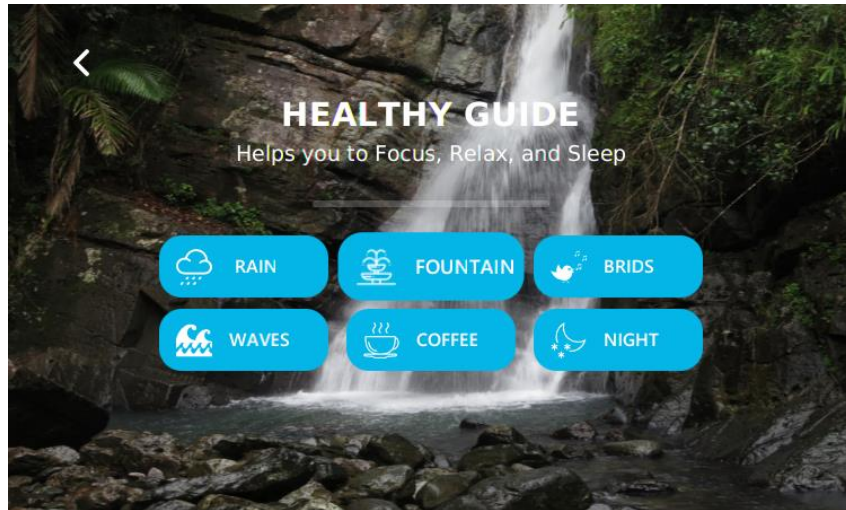


---

*Figure 4-17 Improve Focus Screen of Rain Sound*

---

- **Fountain**

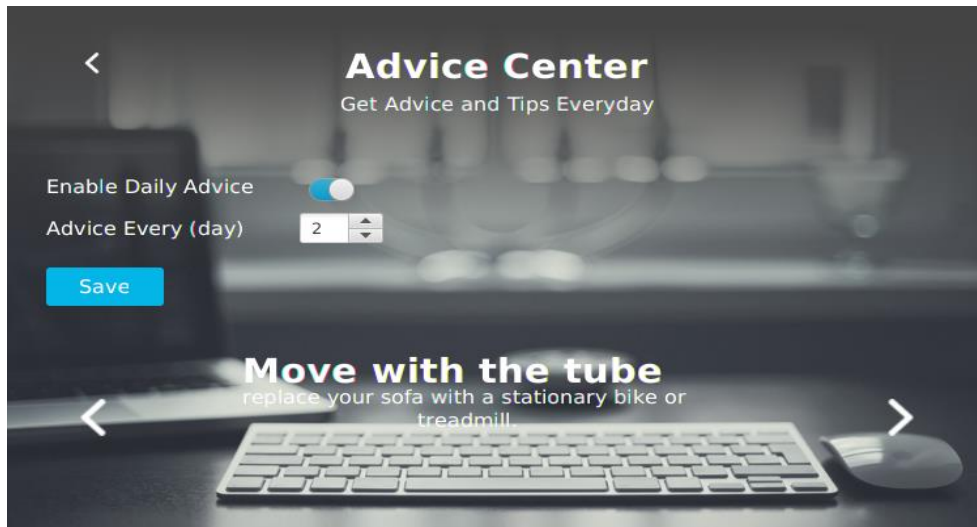


---

*Figure 4-18 Sketch Improve Focus Screen of Waves Sound*

---

**6. Advice Setting**



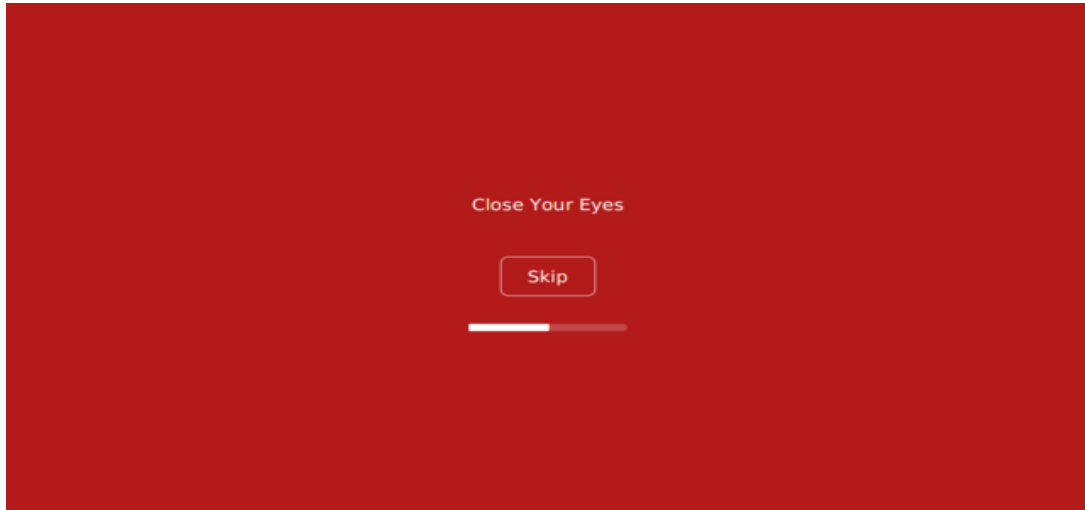
---

*Figure 4-19 Advice Setting*

---

## 6. Eye Break

The break color changed based on user preference

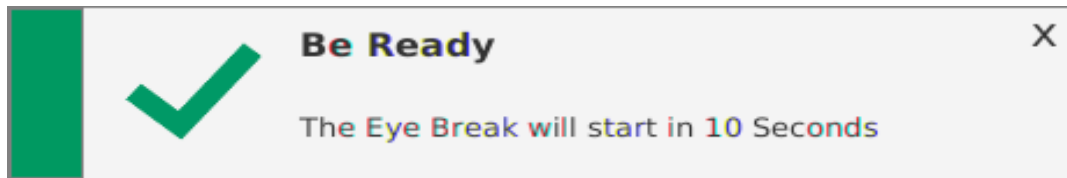


---

*Figure 4-20 Eye Break Background*

---

## 7. Notification



---

*Figure 4-21 Notification Before Break*

---

## 4.11 Testing

Software testing is the process evaluation of software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other word, software testing is a verification and validation process.

### Verification

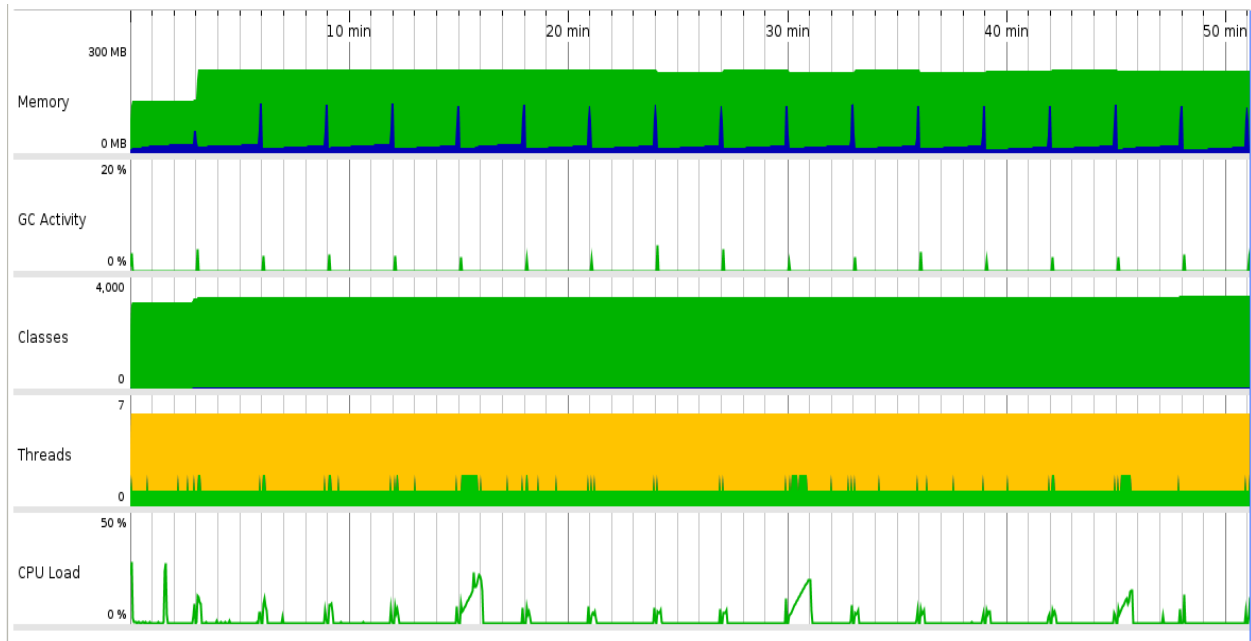
Verification is the process to make sure the system satisfies the condition imposed at the start of the development phase. In other word, to make sure the system behave' the way it is expected to.

### Validation

Validation is the process to make sure the system satisfies the specified requirements at the end of development phase. In other word, to make sure the system built as per user requirement.

### 4.11.1 Performance Testing

The performance was tested on Linux and Windows by using JProfiler analyzes CPU, Memory and Thread profiling and we got result that showed percentage of performance.



*Figure 4-22 JProfiler Performance Testing*

#### **4.11.1.1 Performance Testing Result**

JProfiler result show that The 'Healthy Guide' Using 12 to 20 MB of RAM and 0.75% to 1.25% of CPU load this result is the average at one hour.

#### **4.11.2 Functionality testing**

All functions in the application includes the eye break, distance calculation between user and screen, focus sounds and the advice center, were tested on multiple platforms (windows, Linux, Mac Os).

#### **4.11.3 Preferences Testing**

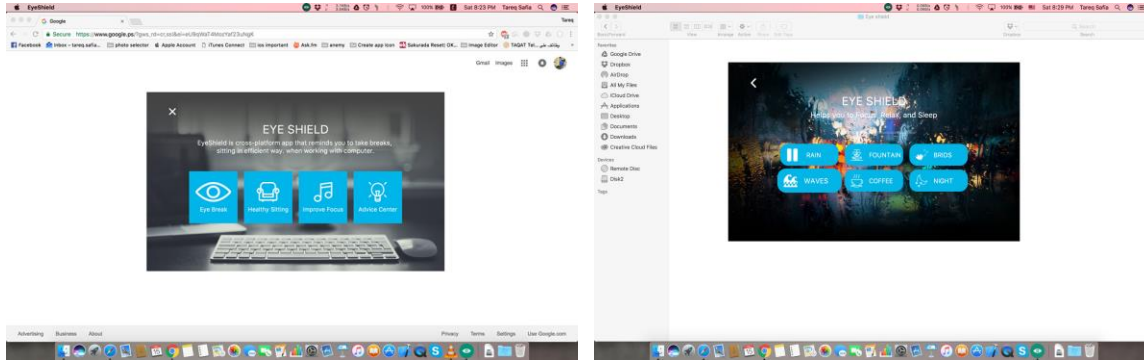
The preferences value is very important. We implement a default value for every Key on Preferences and if user don't modify their Preferences the application will auto save the default value and that prevent any exception or error while get the preferences value.

#### **4.11.4 Application Installer**

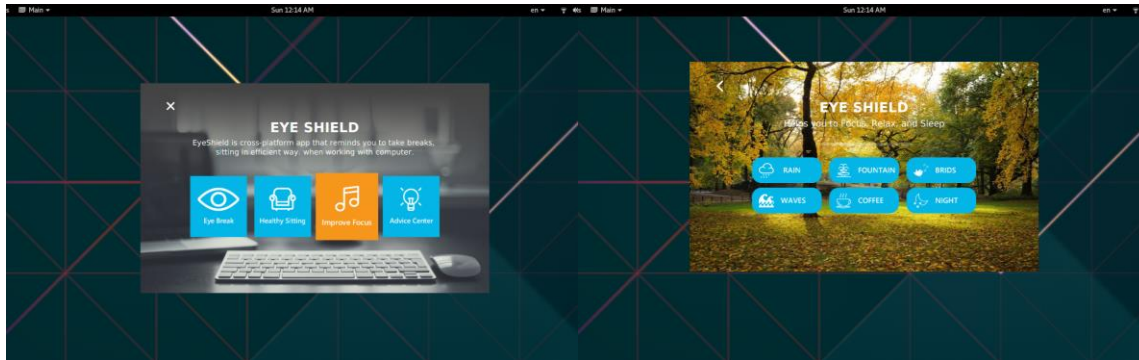
The installer was implemented using install4j (Powerful Multi-Platform Java Installer Builder by EJ-Technologies). Three Type of installer exe for windows, sh for Linux and dmg for Mac os. The installer contains the application files, libraries and pre-packaged JRE bundles. All installer tested and work properly.

### 4.11.5 Operating System Compatibility

The application were tested on MS Windows, MacOS and Linux. All functions work properly. As figure below show the application work on all platform were tested.



*Figure 4-23 'Healthy Guide' on Mac OS*



*Figure 4-24 'Healthy Guide' on Fedora 25 (Linux)*

## 4.12 Evaluation

The evaluation is done with the three samples based on what is explored in appendix (a).

The results at the evaluation have shown a high satisfaction level of the respondents. They have agreed that the prototype solve serious health problem the users could face.

We have selected a random sample of users. The results were as follows:

- The first sample of users was a sample of university students, has welcomed the idea and expressed their admiration which has also shown admiration for the design and application interfaces and easy to deal with.
- The second sample was group of technical experts, has shown welcomed the idea, and they add some feedback to improve the application and idea in the next version.
- The final sample was normal computer users, which found that the application of new and innovative idea and design wonderful and unique, they found the eye break and improve focus are the best features in the application and they ask to add more sounds to focus section.

## 5 Chapter 5 conclusion



The application report entitled "Healthy Guide Application" has come to its conclusion. The new application has been developed with so much care and at the same time efficient and less time consuming. Application is robust. Also provision is provided for future developments in the system. It provides new idea to improve eye health, by help user to take break and to make eye exercise.

## **5.1 Development process**

Following the Extreme programming development process was somehow good because it helped us in testing, retesting and correcting errors and defects.

The usage of Extreme programming methodology helped us to overcome the requirements changing.

## **5.2 Programming language**

The programming language we used JavaFX is a software platform for creating and delivering desktop applications, as well as rich internet applications (RIAs) that can run across a wide variety of devices. JavaFX is intended to replace Swing as the standard GUI library for Java SE, but both will be included for the foreseeable future. JavaFX has support for desktop computers and web browsers on Microsoft Windows, Linux, and macOS.

## **5.3 Problem we faced**

The problem we faced in our application is the measurement of distance between user and desktop screen.

## **5.4 what have been done**

1. Break screen for the notification that appear to user.
2. Measure the distance between user and desktop screen to make user sit in health session.
3. Playing a relax sounds in background that let user be in focus mode.
4. Advice center that appear to user in every screen break.

## **5.5 What technologies have we learned**

- Programming with JavaFx
- Design with FXML, CSS in JavaFx
- Using Google API

## **5.6 Future work**

1. Support multi language.
2. Support Mobile Device.
3. Improve Distance Calculation accuracy.
4. Kids' Corner (Kids has a Special case)
5. Make Website

## References

- [1] J. Smith. *Desktop Applications Vs. Web Applications*. Available: [http://www.streetdirectory.com/travel\\_guide/114448/programming/desktop\\_applications\\_vs\\_web\\_applications.html](http://www.streetdirectory.com/travel_guide/114448/programming/desktop_applications_vs_web_applications.html)
- [2] "د. خ. عوض،" صحة العين اثناء الجلوس امام الحاسوب ed.
- [3] "د. ع. البرش،" صحة الظهر والعمود الفقري اثناء الجلوس امام الحاسوب ed.
- [4] *EyeLeo* Available: <http://eyeleo.com/>
- [5] *SafeEyes*. Available: <http://slgobinath.github.io/SafeEyes>
- [6] J. Hovancik. *Stretchly*. Available: <https://hovancik.net/stretchly>
- [7] Wikipedia. *Application software*. Available: [https://en.wikipedia.org/wiki/Application\\_software](https://en.wikipedia.org/wiki/Application_software)
- [8] IBM. *Advantages of Java*. Available: [https://www.ibm.com/support/knowledgecenter/en/ssw\\_aix\\_61/com.ibm.aix.performance/advantages\\_java.htm](https://www.ibm.com/support/knowledgecenter/en/ssw_aix_61/com.ibm.aix.performance/advantages_java.htm)
- [9] Wikipedia. *Java (programming language)*. Available: [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
- [10] Wikipedia. *JavaFX*. Available: <https://en.wikipedia.org/wiki/JavaFX>
- [11] oracle.com. (2014). *Mastering FXML* Available: [http://docs.oracle.com/javafx/2/fxml\\_get\\_started/why\\_use\\_fxml.htm](http://docs.oracle.com/javafx/2/fxml_get_started/why_use_fxml.htm)
- [12] oracle.com. (2014). <https://docs.oracle.com/javafx/2/api/javafx/scene/doc-files/cssref.html>. Available: JavaFX CSS Reference Guide
- [13] w3.org. *CSS 2.1*. Available: <https://www.w3.org/TR/CSS21/>
- [14] *Google Vision API*. Available: <https://cloud.google.com/vision/>
- [15] *Face Detection Concepts*. Available: <https://developers.google.com/vision/face-detection-concepts>
- [16] M. Sami. (2012). *Software Development Life Cycle Models and Methodologies*. Available: <https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies>
- [17] Wikipedia. *Software development process*. Available: [https://en.wikipedia.org/wiki/Software\\_development\\_process](https://en.wikipedia.org/wiki/Software_development_process)
- [18] *Agile Model & Methodology*. Available: <https://www.guru99.com/agile-scrum-extreme-testing.html>
- [19] G. B. a. R. Maksimchuk, *Object-Oriented Analysis and Design 3rd*. Addison-Wesley Professiona, 2007.
- [20] D. Bell. (2004). *The sequence diagram*. Available: <http://www.ibm.com/developerworks/rational/library/3101.html>

## Appendixes

### Appendix (a)

#### ❖ مقابلة مع طبيب العيون د. خالد عوض – مستشفى سان جون للعيون

١. هل يؤثر الجلوس لفترات طويلة امام الحاسوب على صحة العين؟  
النظر لفترات طويلة الى شاشة الحاسوب يؤدي إلى تقليل عملة الترميش مما يؤدي جفاف العين
٢. ما هي الاحتياطات الواجب اتباعها لتجنب حدوث جفاف العين؟  
ينصح الشخص ان يقوم بعمل تمارين للعين متمثلة ب blanking (الترميش) من ١٠-٢٠ مرة في الدقيقة حتى يتم الحفاظ على راحة العين و سلامتها
٣. ما هي المدة الواجب اخذ قسط من الراحة بعد اتمامها امام شاشة الحاسوب؟  
يجب على الشخص الجالس امام شاشة الحاسوب لمدة نصف ساعة ان يحصل على قسط من الراحة لمدة لا تقل عن ٥ دقائق يقوم فيها بعملية الترميش او النظر لشيء بعيد
٤. ما هي التمارين اللازمة للحفاظ على سلامة العين؟  
تتمثل التمارين باغلاق العين و ارحتها و غسل الوجه بماء بارد
٥. ماذا يحدث عن استخدام الحاسوب بدون ارتداء النظارة لشخص يعاني من اضطرابات بصرية؟  
عدم استخدام النظارة للشخص الذي بحاجة لها سيؤدي لحدوث مضاعفات للمرض الذي يعاني منه كما سيؤدي لظهور عيوب بصرية جديدة
٦. ما الارشادات الواجب اتباعها للحفاظ على سلامة العين من الجلوس امام جهاز الحاسوب؟  
✓ الفحص الدائم و الاطمئنان على قوة الابصار  
✓ عدم الجلوس امام الكمبيوتر فترة طويلة (كل نص ساعة فترة استراحة)  
✓ الاسترخاء و اغلاق العين في فترة الراحة
٧. ما هو الوضع الطبيعي لراحة العين عند الاستخدام المفرط للحاسوب؟  
الوضع الطبيعي لاي شخص تتمثل في استراحة من ٥-٢٠ دقيقة لان الاستخدام المفرط و المتواصل للكمبيوتر يؤدي لعدم توزيع مرطبات العين مثل ( العين تفرز افرازات لغسل و ترطيب العين فكلما زاد تركيز العين على شيء معين يؤدي الى جفافها و الجفن اعلى العين يؤدي الى احتكاك في العين مما يسبب التهابها مما يؤدي الى صداع في الرأس)

## Appendix (b)

### ❖ مقابلة مع طبيب العظام د. عدنان البرش – مستشفى الخدمة العامة

١. ما هو تأثير الجلوس الغير صحيح امام جهاز الحاسوب على سلامة الظهر؟  
الجلوس الغير صحيح عامة له تأثير كبير على الظهر و ذلك بسبب ان اكبر ضغط يحدث على الفقرات و الغضاريف ما بين الفقرات يكون في فترة الجلوس اذا كانت الجلسة غير صحيحة ممكن ان تؤثر على المدى البعيد بحيث يحدث انزلاقات غضروفية في الظهر او انزلاق فقري في الظهر او مشاكل في الاربطة مما تؤدي لحدوث الام مزمنة في الظهر

### ٢. الارشادات الواجب اتباعها للحفاظ على سلامة الظهر؟

- ✓ تجنب الوقوف و الجلوس لفترات طويلة ما يزيد عن ساعة و ذلك عن طرق التحرك او بذل مجهود لمدة ٥ دقائق
- ✓ تجنب الاحمال الثقيلة
- ✓ حمل الاشياء عن الارض بشكل صحيح بان يجعل الظهر مستقيم حتى يتجنب حدوث اي ضغط مفاجئ على الظهر

### ٣. التمارين اللازمة في فترة الاستراحة للحفاظ على سلامة الظهر؟

- ✓ ممارسة تمارين خفيفة لتنشيط العضلات
- ✓ في حين ممارسة رياضة حمل الاثقال يجب على الانسان ان يبدأ بالثقل المعقول

### ٤. هل تختلف فترة الاستراحة ما بين انسان مصاب سابقا و انسان سليم؟

الانسان المصاب يجب ان يتجنب كافة الامور المضرة التي ذكرناها سابقا بشكل قطعي

### ٥. هل الجلسة الغير صحية تؤثر على اشياء اخرى غير الفقرات و العمود الفقري؟

نعم تؤثر على الاعصاب الطرفية المرتبطة بالعمود الفقري مما يؤدي لفقد الاحساس او فقد قوة عضلة الجسم.