Bachelor Thesis

Computer Engineering

Are-U-Drunk?

Measuring alcohol intoxication via smart mobile sensing

Author
Cristina Díaz García

Supervisors
Oresti Baños Legrán
Claudia Villalonga Palliser

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

Granada, academic year 2020-2021

Are-U-Drunk? Measuring alcohol intoxication via smart mobile sensing

Cristina Díaz García

Palabras clave: alcohol, desarrollo móvil, procesamiento de video

Resumen

El alcohol es una droga legal de la que a veces se hace un uso excesivo y/o inadecuado. Esto no solo puede afectar a quien la usa, sino a quien le rodea, ya que el comportamiendo del sujeto puede llevar a situaciones como una pelea o un accidente de tráfico. El objetivo de este proyecto es la detección de un posible abuso del alcohol para así poder informar al usuario y facilitar una toma de mejores decisiones. Es por ello que se ha diseñado un sistema digital de detección de movimiento ocular con el que detectar movimientos involuntarios provocados por el efecto del alcohol en el sistema nervioso. La usabilidad y el rendimiento de la aplicación desarrollada han sido evaluados mediante una escala de usabilidad de sistemas, System Usability Scale en inglés, en la que se puntúan diferentes aspectos del sistema. Tras esta evaluación, la aplicación ha demostrado ser útil, usable y práctica.

Are-U-Drunk? Measuring alcohol intoxication via smart mobile sensing

Cristina Díaz García

Keywords: alcohol, mobile, video processing

Abstract

Alcohol is a legal drug that is sometimes overused and/or misused. This can affect not only who uses it, but who surrounds them, since the behavior of the subject can lead to situations such as a fight or a traffic accident. The objective of this project is the detection of possible alcohol abuse in order to inform the user and facilitate better decision-making. For this purpose a digital eye movement detection system has been designed with which to detect involuntary movements caused by the effect of alcohol on the nervous system. The usability and performance of the developed application have been evaluated using a System Usability Scale, in which different aspects of the system are scored. After this evaluation, the application has proven to be useful, usable and practical.

Yo, **Cristina Díaz García**, estudiante de la titulación Graduado en Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 53742687J, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Cristina Díaz García

Granada a 06 de septiembre de 2021

D. **Oresti Baños Legrán** y Dª. **Claudia Villalonga Palliser**, ambos profesores del Departamento Arquitectura de Computadores de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado ***Are-U-Drunk? Measuring alcohol intoxication via smart mobile sensing***, ha sido realizado bajo su supervisión por **Cristina Díaz García**, y autorizo la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expide y firma el presente informe en Granada a 06 de septiembre de 2021

El supervisor:

Oresti Baños Legrán

La supervisora:

Claudia Villalonga Palliser

Agradecimientos

A mi padre, mi madre y mi hermano, que me han inculcado los valores que tengo y me han hecho llegar hasta donde estoy.
A Carlos, por estar siempre ahí.
A mis amigos y amigas, especialmente Casandra y Francisco, por ser puntos de apoyo incondicionales cuando más falta hace.

Contents

List of Figures

List of Tables

Listings

1 Introduction

1.1 Context

Alcohol is a legal drug in most countries worldwide, which makes it very easy to access for anyone who is over the legal age to buy it in their country.

According to organizations like the World Health Organization [1], the harmful use of alcohol is a causal factor in more than 200 disease and injury conditions and in the age group 20–39 years approximately 13.5% of the total deaths are alcohol-attributable. Worldwide, 3 million deaths every year result from the harmful use of alcohol, a 5.3% of all deaths [6].

As it is shown in Figure 1, there are individual and societal vulnerability factors [6] that affect alcohol consumption and alcohol-related harm. In the context of the occidental world, alcohol consumption is totally accepted and most of the socialization among individuals is made by eating and drinking, being most of the beverages alcoholic ones. This could lead sometimes to a misuse and abuse of alcoholic beverages, taking bad decisions, some more naive like texting an ex-partner, some more hazardous like driving.

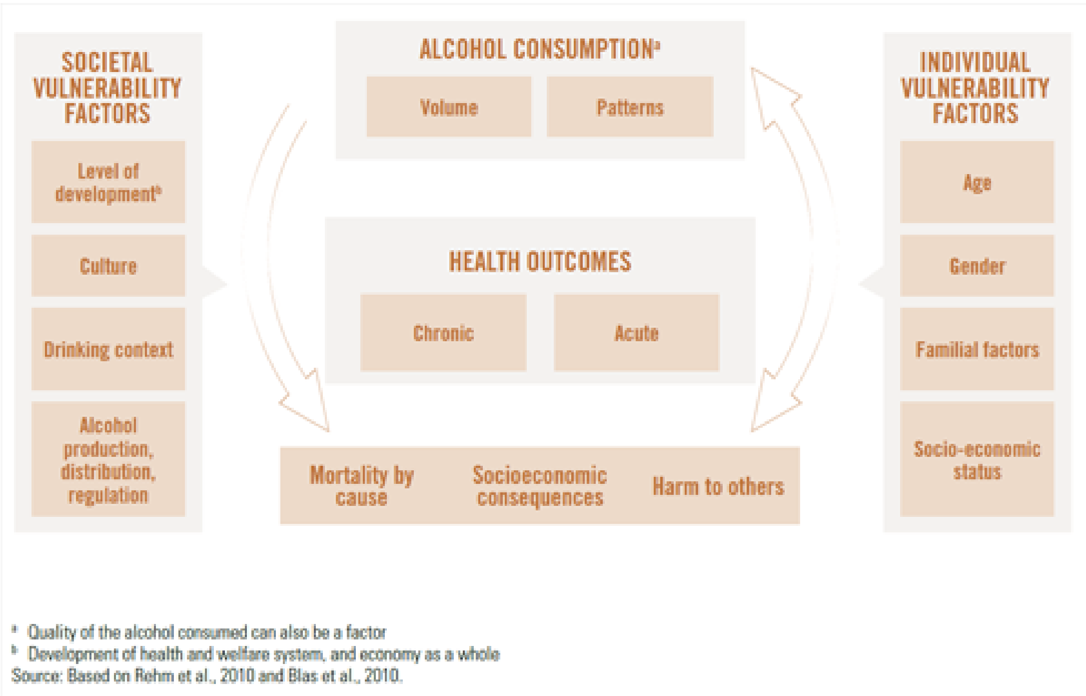


Figure 1: Vulnerabilities declared by the WHO. Reprinted from [1].

1.2 Motivation

Nowadays, most people, at least in the developed countries, own one or more smartphones. This makes the monitoring of people easier than ever. Adding to this fact how science has advanced, it makes up one of the best possible scenarios for studying the interaction between society and alcohol intake, both amount of alcohol consumed and behavior changes.

Having in mind what has been previously mentioned and with the exhaustive use of smartphones that is being done, the sensing of people's behaviors has been made easier, and as an example, we could use the keyboard to check whether the writing gets worse or the predictive words are used more frequently after consuming alcohol or taking a selfie to estimate the blood alcohol concentration.

According to Medical News Today [7] and Healthline [8], amoung the numerous effects on the consumer's body and behaviour, the consumption of alcohol may lead to reduced reaction and movement. This will be key aspect to be considered for the system we want to develop. As a consequence of the alcohol intoxication, an abnormal movement is shown by the eyes, known as Nystagmus.

Nystagmus is, according to MedlinePlus [9], 'a term to describe fast, uncontrollable movements of the eyes that may be side to side (horizontal nystagmus), up and down (vertical nystagmus) and rotary (rotary or torsional nystagmus)'. This effect is the one described in Figure 2. Nystagmus can affect vision, balance, and coordination.

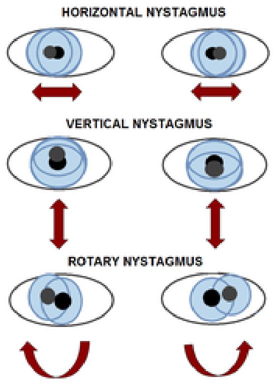


Figure 2: Horizontal, vertical and rotary Nystagmus

1.3 Approach and planning

1.4 Objectives

- **Main goal**: to develop a system to automatically assess a person's drunkness based on the analysis of their eye(s) movement.
- **Secondary goals**:
  - To design a system that detects and tracks the user's eyes and analyzes them in search of horizontal nystagmus movements.
  - To develop a system that detects and tracks the user's eyes and analyzes them in search of horizontal nystagmus movements.
  - To evaluate and test the previously mentioned system.

1.5 Structure

The first chapter of the project, *State of the art*, contains a wider perspective of the current state of tools to detect and monitorize alcohol consumption through a research about wearables, applications and scientific articles. The second chapter, *Design*, shows a more exhaustive list of the requisites of the system and the design followed: architecture, programming languages and frameworks used. The third chapter, *Implementation*, offers a detailed explanation of the development and deployment of the system. The fourth chapter, *Evaluation*, analyses the system by means of System Usability Scale tests [10] and analysis of the data obtained from the use of the application. The fifth and last chapter, *Conclusions*, analyses the initial objectives of the project and whether they were achieved or not, as well as possible future work.

2 State of the art

To get some perspective of the current state of the art with respect to applications related to alcohol intake, three different approaches were taken.

2.1 Search methods

2.1.1 Handsearch

Some exploration on Google has been done on the field of wearables, such as BACtrack Skyn [11] and Copilot [12]. A later section will deepen on these two wearables.

2.1.2 Commercial applications

Firstly, a search through the Apple Store [13] was done with three different queries: alcohol, driving, and intake. No applications related to alcohol consumption were found with any of the queries, only driving games and applications to consume more water or measure the intake of calories. Secondly, a search through the Google Play Store [3] was done with the query alcohol, which threw very interesting results. These results will be reviewed deeper in a later section.

2.1.3 Scientific publications

When searching, the terms taken into account were smartphone or similar devices, alcohol or drunkenness and monitor/monitoring and similar. The fields of research were medicine, engineering, psychology and computation. The query used is the one Figure 3 shows, which threw 231 results on 8th March 2021.

TITLE-ABS-KEY ((smartphone OR wearable OR smartwatch) AND (alcohol OR drunkenness) AND (monitor* OR sens* OR detect*)) AND (LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017)) AND (LIMIT-TO (SUBJAREA , "MEDI") OR LIMIT-TO (SUBJAREA , "ENGI") OR LIMIT-TO (SUBJAREA , "COMP") OR LIMIT-TO (SUBJAREA , "PSYC"))

Figure 3: Query used in the search through Scopus [2]

Those 231 results were filtered, firstly by the abstract, depending on how interesting the abstract was. This resulted in 63 results after the first filter. These 63 results were once again filtered depending on the accuracy of the abstract according to the subject of our work. Finally, 22 final results were obtained, which were read and a 'State of Art Matrix' was built. This matrix can be found in Appendix A. After reading the 22 papers, as of 8th March 2021, some very interesting approaches can be analyzed. This proccess is what Figure 4 shows.

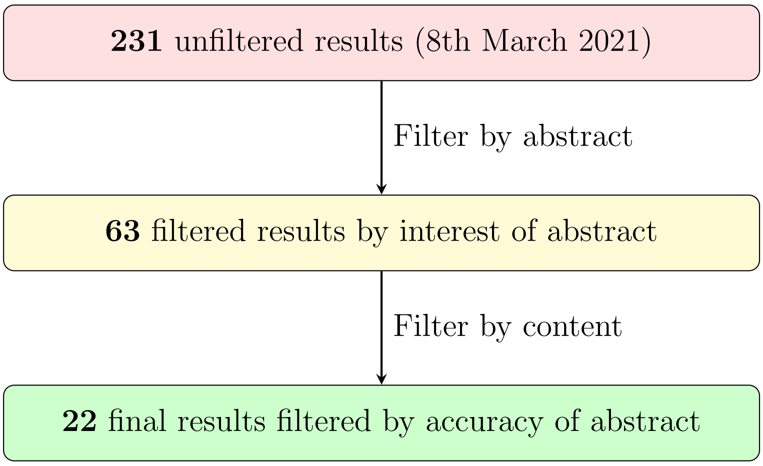


Figure 4: State of the art filtering steps

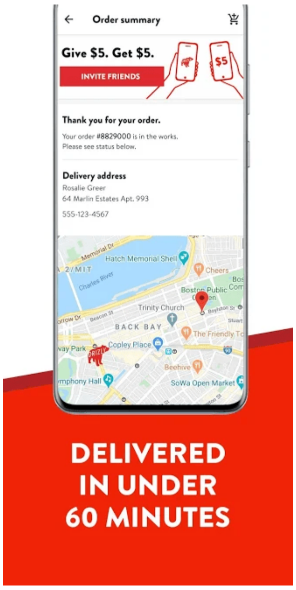
2.2 Wearables

BACtrack Skyn [11] is a wearable, a smart bracelet, only available for research, that tracks Transdermal Alcohol Content (TAC) in real time. It can be used on its own or integrated with Apple Watch [14]. It has a cloud-based web portal used to manage, view and export the data obtained with the device. More information can be requested to the BACtrack team on their webpage.

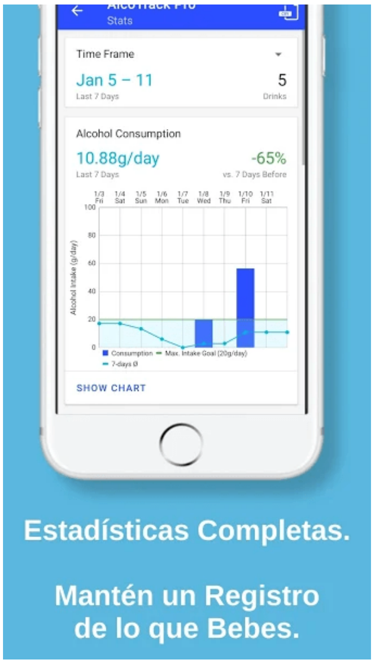
Copilot [12] is a watch and application tandem that allows the user to monitor and learn their driving patterns so it can detect when the pattern changes because of alcohol intake thanks to the watch's gyroscope, accelerometer and heartbeat monitor.

2.3 Applications

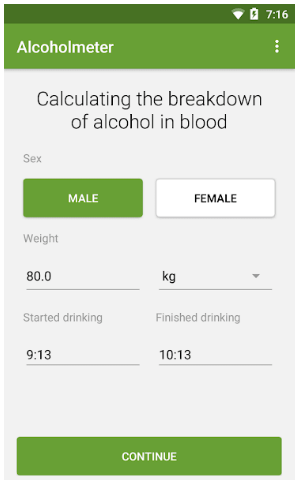
As previously mentioned, none of the results when searching through the Google Play [3] were interesting. On the other hand, it is interesting to deepen on the results of the search through the Google Play [3]. The next paragraphs deepen on these results.



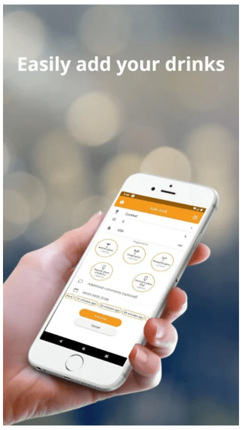
(a) Drizly [15] application



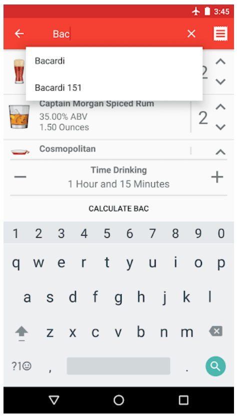
(b) Alcotrack [16] application



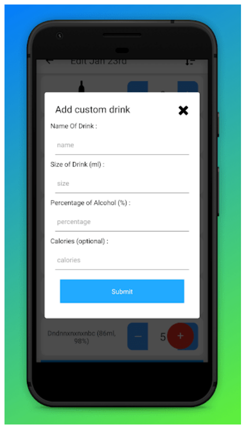
(c) Alcoholcheck [17] application



(d) Alcofy [18] application



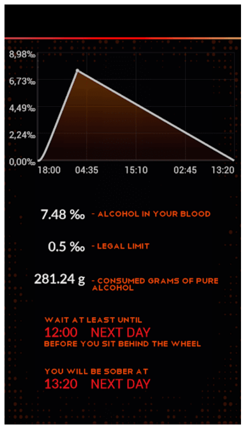
(e) Blood Alcohol Calculator [19] application



(f) Simple Alcohol Unit Tracker [20] application

(g) Blood Alcohol Calculator [21] application

(h) Alco Calculator [22] application



(i) Alcohol test (for fun) [23] application

Figure 5: Screenshots of the different applications found in the Google Play [3]

Drizly [15] is an application where the user can select different alcoholic beverages and get them delivered home. Alcotrack [16] tracks alcohol consumption through self-report of alcoholic drinks. Alcoholcheck [17] calculates blood alcohol content and an estimate of how long it will take you to reach zero level of alcohol in blood again after inserting gender, age, weight and consumed beverages. Alcofy [18] is an application to add drinks in real time to see how drunk the user is and will be the next hours until they become sober. A limit can be set and the application will send a warning if the user is about to cross that limit. The estimated blood alcohol concentration is calculated based on a formula that has been studied and praised in scientific papers. Blood Alcohol Calculator [19] is an application to add weight, gender, drinks selected from a list of hundreds of built-in drinks or created/customized by the user and how long the user has been drinking to calculate the blood alcohol content of you and your friends. This shows the estimation of blood alcohol concentration and some possible side effects related to that blood alcohol range, as well as a detailed graph letting you know what your blood alcohol content will be over time. Simple Alcohol Unit Tracker [20] calculates the approximate blood alcohol concentration level based on the information about the user and their consumed alcoholic drinks. It has several options: calculation of blood alcohol concentration, reflex test, addition of new alcohol types and the setting of alcohol limits. Blood Alcohol Calculator [21] calculates blood alcohol content and an estimate of how long it will take you to reach zero level of alcohol in blood again after inserting gender and consumed beverages. Alco Calculator [22] calculates the concentration of alcohol in a disolution (beverage or not). Alcohol test (for fun) [23] is an application to add drinks and understand their alcohol consumption/drinking habits. A screenshot of each of these applications can be found in Figure 5.

2.4 Scientific articles

With respect to scientific articles found in Scopus [2] and filtered as previously mentioned, different aspects have been taken into account:

2.4.1 Countries from the affiliations

Most of the affiliations are exclusively from USA [5, 24–32], but others were a colaboration of one or more countries out of the USA and a country from USA [33–35]. Only three articles have affiliations exclusively from UK and those are [36–38]. Australia participated in [39] and with Switzerland in [40]. Japan was the country of the affiliations of [41], India was the country of affiliation of [42] and Denmark of [43]. Table 1 summarizes the relation between countries and affiliations.

2.4.2 Sample size

Most of the studies had a sample of less than 100 people summing all of the mentioned phases of the study, specifically 13 out of the 22 studies previously mentioned [24–27, 29–31, 33, 35, 36, 38, 43, 44]. Four of them [32, 37, 41, 42] did not mention anything about the sample used in the study and the remaining five had samples between 120 and 671 subjects [5, 28, 34, 39, 40]. More information can be found in the Appendix A.

Country	Affiliations (out of 22)
USA	13
UK	4
Autralia	3
Switzerland	2
Brazil	1
Canada	1
Denmark	1
Japan	1
India	1
South Korea	1
Spain	1
Thailand	1

Table 1: Affiliations' countries of the different studies reviewed

2.4.3 Diversity

According to diversity, three different aspects were taken into account: age diversity, gender diversity and ethnic diversity.

Regarding age, eight of them [5, 28, 33, 35, 37, 41–43] did not mention anything about the ages of the samples. Most of the rest have a wide age range, but four of them [27, 30, 31, 40] were specifically for young people therefore the age range was more limited. Regarding gender, eight of them [5, 33, 35–37, 41–43] did not mention anything about the gender and one mentioned diversity but did not mention ratios nor percentages. One had a 100% of males with no specific reason [44] and another one had 100% of females because the study wanted to study the impact of problematic alcohol drinking in female college students [31]. The rest of the studies were approximately half male and half female (some had up to 70% one gender), except one [38], that has almost 90% male and only almost 10% female. Regarding ethnic, more than half of the studies, 15 out of 22 [24, 25, 28, 33, 36–42, 44], did not mention anythihng about ethnic, four of them [5, 26, 29, 34] mentioned diversity but did not specified proportions nor percentages and the rest were diverse. More information can be found in the Appendix A.

2.4.4 Applications

Less than a fourth of the studies, 5 out of 22 [26, 32, 33, 35, 42], did not use any smartphone application. One of the remaining studies [44] used pre-existing applications: Tetris, Fruit Ninja, and Unblock Puzzle, all three of them with some modifications to track important information; another 5 studies [5, 27, 40, 41, 43] used an application but did not mention the name and the rest used the following applications: Alcogait [24], Ria Treatment Platform app [25], Drinks:Ration [36], DrunkSelfie [28], Drink Less [37], InDEx (Information about Drinking for Ex-serving personnel) [38], CNLab-A [39], DUI (Drunk user interfaces) [29], DrinkTRAC [30], Empatica [31] and Alcooquizz [34].

2.4.5 Device(s)

Regarding devices, some of the studies used only smartphones [25, 26, 28–30, 34, 37–44] and some others used smartphones and some kind of wearable [5, 24, 27, 31, 36]. Only [35], [33] and [32] used only wearables (smart shoes, smart eyeglasses and electronic bracelet, respectively). Table 2 summarizes the devices and sensors used in the reviewed studies. More information can be found in the matrix attached in the Appendix A.

Table 2: Smartphone specifications and measurements from smartphones used in the reviewed studies

Study	**Device(s)**	**Sensors**	**Measurements**
Li et al. (2021) [24]	Smartphone (Google Pixel XL) + wearable (LG Watch Sport)	Accelerometer and gyroscope	Gait changes
Intarasirisawat et al. (2020) [44]	Smartphone (Samsung S6)		Device acceleration, rotational motion and touch-based features
Phan et al. (2020) [40]	Smartphone (Model not specified, Android OS)	Accelerometer, Bluetooth, WiFi, GPS, apps logs, camera	Location accuracy, speed, and GPS coordinates, network percentage; and mobility. human mobility, social context and person-person proximity
Mitchell et al. (2020) [25]	Smartphone (Model not specified and OS not specified)	Breathalyzer (External)	Alcohol in breath
Businelle et al. (2020) [5]	Smartphone (Samsung Galaxy J3 smartphone (or equivalent)) + wearable (SCRAM bracelet)	Geolocation	Longitude + latitude (geolocation), BAC

Study	Device(s)	Sensors	Measurements
Suffoletto et al. (2020) [26]	Smartphone (Model not specified and OS not specified)	Accelerometer	Mean of acceleration signal, Variance of acceleration signal, Correlation of pairwise acceleration signals, Covariance of acceleration signal, Maximum difference of acceleration signal, Maximum difference of pairwise acceleration signals, Mean trend of acceleration signal of 0.1 second windows within the window, Windowed mean trend of acceleration signal of 0.1 second windows within the window, Variance trend of acceleration signal, Windowed variance trend of acceleration signal
Leightley et al. (2020) [36]	Smartphone (Model not specified, Android and iOS OS) + wearable (not specified)	GPS, push notifications	Location, heart rate, distance travelled, activities, height, and weight
Wakana & Yamada (2019) [41]	Smartphone (Model not specified and OS not specified)	Breathalyzer (External)	Breath Alcohol Concentration (BrAC)
Killian et al. (2019) [27]	Smartphone (Model not specified, Android and iOS OS) wearable (SCRAM bracelet)	Accelerometer	Acceleration/movement and TAC (transdermal alcohol content)
Sempionatto et al. (2019) [33]	Wearable (smart glasses)	Alcohol biosensor and amperometric sensor	Alcohol and Glucose in tears

Study	Device(s)	Sensors	Measurements
Willoughby et al. (2019) [28]	Smartphone (Model not specified, Android OS)	Camera	Face Detection, Locating Facial Landmarks, Aligning Faces using Landmarks, Landmark positions, Landmark vectors, Landmark lines, Detecting facial lines and wrinkles using Canny Edge detection and the Hough transform, Forehead redness, Smiles, Lips and Eyes
Garnett et al. (2019) [37]	Smartphone (Model not specified, iOS OS)	None	ABV (alcohol by volume), size, quantity and price. Mood, productivity, clarity and sleep quality. Calorie intake, spend on alcohol, and how mood, productivity, clarity and sleep quality are affected by heavy drinking.
Chatterjee, Isha and Sharma (2018) [42]	Smartphone (Model not specified and OS not specified)	Camera	Video of the person
Leightley et al. (2018) [38]	Smartphone (Model not specified, Android and iOS OS)	None	Amount of alcohol consumed, how and when
Poulton et al. (2018) [39]			Quantity, frequency, type of alcohol, start and finish times
Mariakakis et al. (2018) [29]	Smartphone (Third-generation Moto G smartphone)	Touchscreen, accelerometer, and gyroscope, flash and camera	How correctly they write and if they correct or not the mistakes, also how the subject maintain the balance, heart rate
Suffoletto et al. (2018) [30]	Smartphone (Model not specified, iOS OS)	Accelerometer, gyroscope and magnetometer	Real, dynamic and static acceleration, angular velocity, attitude of the device

Study	Device(s)	Sensors	Measurements
Leonard et al. (2017) [31]	Smartphone (Model not specified, Android OS) wearable (Empatica E4 wrist band)	Electrodermal activity (EDA), accelerometer and temperature sensor	Real-time EDA, movement, temperature, heart rate
Bertholet et al. (2017) [34]	Smartphone (Model not specified, Android and iOS OS)	None	Quantity and frequency of alcohol consumption
Kinnamon et al. (2017) [32]	Wearable (Electronic Bracelet for Monitoring of Alcohol Lifestyle)	Immunoassay based EtG biosensor	Ethyl glucuronide (EtG) in sweat
Mellentin et al. (2017) [43]	Smartphone(Model not specified, Android OS)	App logs	Real-time measures of cue-induced cravings.
Park et al. (2017) [35]	Wearable (smart shoes)	Array of pressure sensors	Pressure of the feet when walking

As shown in Table 2, more than a fifth are multiplatform, available for both Android and iOS, almost a fifth works for Android and almost a fifth does not specify the operative system. Some of them have specific models, like Samsung S6, Samsung Galaxy J3 (or equivalent), Third-generation Moto G and Google Pixel XL. Table 3 sums up the different devices used along the studies. Not a lot of wearables are used. Some of them are SCRAM bracelet, Empatica E4 wrist band and LG Watch Sport. Table 4 summarizes the different wearables used in the reviewed studies.

Smartphone specifications	Studies (out of 22)
Model not specified, Android and iOS	5
Model and OS not specified	4
Model not specified, Android OS	4
Model not specified, iOS OS	2
Samsung S6	1
Samsung Galaxy J3 (or equivalent)	1
Third-generation Moto G	1
Google Pixel XL	1

Table 3: Smartphone specifications from smartphones used in the reviewed studies

2.4.6 Sensors

The most used sensors are accelerometer and gyroscope, but also geolocation, camera and temperature sensors. Some of the studies do not mention the use of any kind of sensors [34, 37–39]. Table 5 summarizes the use of sensors. More information can be found in table 2 and in the Appendix A.

Wearables	Studies (out of 22)
SCRAM bracelet	2
LG Watch Sport	1
Not specified	1
Empatica E4 wrist band	1
Electronic Bracelet for Monitoring of Alcohol Lifestyle	1

Table 4: Wearables used in the reviewed studies

Sensors	Studies (out of 22)
Accelerometer	8
Gyroscope	4
Camera	3
Application logs	2
Geolocation	2
Breathalyzer	2
Bluetooh	1
WiFi	1
Flash	1
Push notifications	1
Alcohol biosensor	1
Amperometric sensor	1
Touchscreen	1
Magnetometer	1
Electrodermal activity (EDA)	1
Temperature	1
Immunoassay based EtG biosensor	1
Pressure	1
None	4

Table 5: Sensors [5] used in the reviewed studies

2.4.7 Mobile self-report

Almost half of the studies (10 out of 22) use mobile self-report as an information collector. Of those 10 studies, in 8 of them [5, 30, 34, 36–40] the user added information about alcohol consumption: type of alcohol, amount of beverage, time elapsed when drinking...; one [31] asked for information about feelings when a notification popped up and the last one [43] was about cravings when staring at alcoholic beverages. More information can be found in the Appendix A.

2.4.8 Measurements

Most of the measurements taken are physical ones, related to location and motion like acceleration and movement, but also measurements related to drinking like quantity, frequency, type of alcohol and start and finish times when drinking. Also alcohol in blood, breath and transdermal are meassured. Some studies use videos [42] or selfies [28] of the user. More information can be found in table 2 and in the Appendix A.

2.4.9 Methods

Table 6 summarizes the methods to proccess the previously mentioned data gathered from the sensors and the output obtained with this proccesses.

Table 6: Methods to proccess measurements and outputs from them in the reviewed studies

Study	**Methods**	**Output**	**Drawbacks**
Li et al. (2021) [24]	Machine Learning	Warning while drinking/on the way to the car	Possible bias in layer normalization and methods towards certain subjects
Intarasirisawat et al. (2020) [44]	Machine Learning	Visualization	No gender diversity. Exclusion of people with different conditions. Small sample of study/control.
(Phan, Labhart, Muralidhar & Gatica-Perez, 2020) [40]	Machine learning	Classifiication of the consumption: heavy or non-heavy drinking	No assurance data is correctly inserted, bias in self-report
Mitchell et al. (2020) [25]	Statistics	Models for the relationship between BAC and time	No control group. Patients were highly motivated. Approximately 46% of data for the BAC readings were missing.
Businelle et al. (2020) [5]	Others	Text messages to in-the-moment distraction, reframing, immediate help-seeking, planning, and other tools to reduce craving.	The intervention app may not be applicable to those with low literacy and cognitive impairment. EMA or SCRAM monitoring may have independent effects on drinking. Self-monitoring can lead to changes in drinking, even without an “intended” intervention.

Study	Methods	Output	Drawbacks
Suffoletto et al. (2020) [26]	Machine Learning	Models, regressions and predictions of excessive alcohol consumption	Small sample size, the use of a cohort that largely drinks below risky levels, and controlled setting of data measurement. It did not examine whether gait-related features discriminate lower levels of drinking. It placed the smartphone on the lower back, which may not represent where individuals keep their phones in natural environments. Did not find that population-based models were accurate in predicting intoxication.
Leightley et al. (2020) [36]	Statistics	Push notifications/ SMSs to warn/inform the subject	-
Wakana & Yamada (2019) [41]	Others	Estimated BrAC and detection of saturated water vapor and the metabolites	
Killian et al. (2019) [27]	Machine Learning	Clasification of sobriety every 10 seconds.	Low control of the phone placement, low accuracy of accelerometer, classifiers had a higher accuracy for sober data than intoxicated data and that the variance of the best classifier was high for intoxicated subjects.

Study	Methods	Output	Drawbacks
Sempionatto et al. (2019) [33]	Others	Estimation of glucose and alcohol in blood	Small sample size. small sample volumes of the extracted tears, ease of sample evaporation during collection (particularly in sampling ethanol), potential composition variations between individuals and dependence on the sampling method.
Willoughby et al. (2019) [28]	Machine Learning	Classification of images: drunk or sober	Limited Dataset, Coarse Classification, Smartphone Internet Access Requirement and Comprehensive evaluation of DrunkSelfie app by users
Garnett et al. (2019) [37]	Basic statistics	Feedback and graphics about goals.	-
Chatterjee, Isha and Sharma (2018) [42]	Computer vision techniques of facial landmark detection and motion detection	Alert sound and location notification if the subject is drunk or drowsy	
Leightley et al. (2018) [38]	Basic statistics	SMS with information and warnings, statistics of usage of the application and visualization of data from the reports	No assurance data is correctly inserted, not a long-term analysis, sample is very small, possible selection bias of the participants
Poulton et al. (2018) [39]	Statistics	Average daily and weekly consumption, hourly rate of consumption each day, days of consumption, total drinks and days in which 4 or 6 or 8 or (and so forth) more drinks were drunk	Limited alcohol options, no assurance data is correctly added
Mariakakis et al. (2018) [29]	Machine learning	Estimation of BAL (Blood alcohol level)	It doesn't meassure BAL directly

Study	Methods	Output	Drawbacks
Suffoletto et al. (2018) [30]	Machine Learning	Accurate prediction of blood alcohol concentration during drinking episodes	Sample is very small. Missing gait task data. Participants were paid to complete tasks, which likely artificially inflated completion rates. App was made only for iOS devices. Possible wrong insertion of alcohol drinks.
Leonard et al. (2017) [31]	Psycological	Analysis of the emotion + context, sum of use of screen with the app, amount of reports	Sample is very small. The form of the band was not accurate and the participants often forgot to charge it. The connection was not great and some alerts were inopportune
Bertholet et al. (2017) [34]	Statistics	Statistics about alcohol consumption habits	The uncontrolled design does not allow us to infer causation, further study is needed to evaluate the application's efficacy. Assessment of alcohol consumption relied on self-report, therefore social desirability bias or recall bias are possible. Assessment of app use also relied on self-report.

Study	Methods	Output	Drawbacks
Kinnamon et al. (2017) [32]	Electrochemical	Visualization of graphs with information about voltage, intensity and EtG concentration	-
Mellentin et al. (2017) [43]	Basic statistics	Measures and graphs to keep track of their training activities and potential advances in controlling cue reactivity	
Park et al. (2017) [35]	Machine Learning	A pressure distribution map and classification of the gait	

Most of the studies use machile learning techniques like Logistic Regression (LR), Linear Support Vector Machine (LSVM), Random Forest (RF), Bi-directional Long Short Term Memory (Bi-LSTM) and Convolutional Neural Network (CNN), in addition to Cross-Validation (CV). Statistics like average, median, summations, t-tests and chi-square tests are also used by some studies. [32] uses square wave voltammetry (SWV), an electrochemical methods, [42] uses computer vision techniques of facial landmark detection and motion detection. Others use techniques such as tears estimulation and alcohol and glucose detection in tears like [33], questionnaires for developing + 'real time' drinking algorithm for testing like [5] or a calculation algorithm based on a differential evolutional method like [41]. The summary of the usage of these methods can be seen in table 7.

Method	Studies (out of 22)
Machine Learning	9
Statistics	7
Computer vision techniques	1
Psycological	1
Electrochemical	1
Others	3

Table 7: Methods used in the studies

2.4.10 Output

There is a wide range of outputs, from visualization to classification, but also notifications, models for the relationship between drinking and time estimations of alcohol in blood. More information can be found in Table 6 and in the Appendix A.

2.4.11 Drawbacks

The most common drawbacks are the small sample size and the lack of control of some variables, like low control of the phone placement, low accuracy of accelerometer or no assurance data is correctly inserted. More information can be found in Table 6 and in the Appendix A.

2.4.12 Conflicts of interest

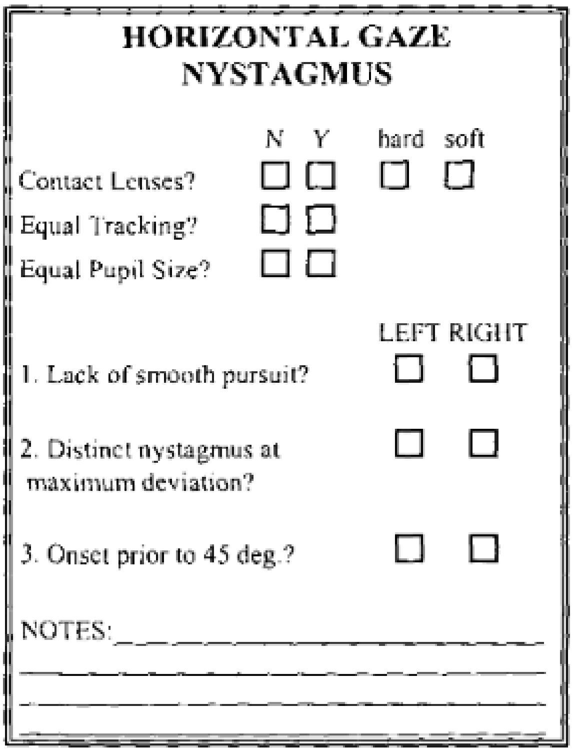
None of the above mentioned articles had any conflicts of interest but these four:

- John Mendelson is the owner of Ria Health. The remaining authors have nothing to disclose [25].
- MB is an inventor of the Insight mHealth Platform and receives royalties related to use of this platform [5].
- NTF sits on the Independent Group Advising on the Release of Data at NHS Digital. NTF is also a trustee of a military-related charity. AS is a full-time member of the Armed Forces seconded to King's College London. DM and CW are employed by Combat Stress, a national charity in the United Kingdom that provides clinical mental health services to veterans [36].
- J.B. has received unrestricted research funding from Pfizer related to smoking cessation. R.W. has received research funding and undertaken consultancy for companies that manufacture smoking cessation medications. R.W. and S.M. are advisers to the National Centre for Smoking Cessation and Training. S.M. is Director of the UCL Centre for Behaviour Change [37].

3 Design

3.1 Horizontal Gaze Nystagmus

Due to the effects of alcohol consumption in the Nystagmus movements, the US police use horizontal gaze Nystagmus as a test, additionally to two others standarized tests (walk and turn and one-leg-stand), to check whether to arrest someone (if they are drunk) or to let them free (if they are sober).The National Highway Traffic Safety Administration of United States wrote a complete guide about the science and usage of Horizontal Gaze Nystagmus for judges, prosecutors and law enforcement [4] in which can be found a deep explanation of the test's performance. Figure 6 summarizes the parameters the police officer would check if it was an in-person test.



HORIZONTAL GAZE NYSTAGMUS

	N	Y	hard	soft
Contact Lenses?	☐	☐	☐	☐
Equal Tracking?	☐	☐		
Equal Pupil Size?	☐	☐		

	LEFT	RIGHT
1. Lack of smooth pursuit?	☐	☐
2. Distinct nystagmus at maximum deviation?	☐	☐
3. Onset prior to 45 deg.?	☐	☐

NOTES:

Figure 6: Parameters check by the police officers. Reprinted from [4].

To perform this test, the police officer will firstly ask the subject whether they wear lenses and if the answer is positive, whether they are hard or soft. Secondly, they will check that both eyes have an equal tracking of the object, such as a pen or the tip of a penlight, used for the test and the equality of both pupils' size. The use or not of contact lenses, hard or soft, do not affect the test in any way, contrary to some concerns. The lack of equal tracking or equal pupil size may indicate blindness in one eye, a glass eye, a medical disorder or an injury. If the subject exhibits these characteristics, the officer should discontinue the test and may need to seek medical assistance for the individual if a medical disorder or injury appears to exist. After these checks, the test starts with three different clues (each of the subtests performed to check the gaze): lack of smooth pursuit, distinct nystagmus at maximum deviation and onset prior to 45 degrees. These three clues are performed asking the suspect to follow the object, placed 12 to 15 inches away (approximately 30-40cm), with only their eyes, both to the left and the right sides, summing up six clues in total. Failing at least two out of this six clues is enough to fail the test.

The first two clues, left and right lack of smooth pursuit, is obtained by moving the object slowly but steadily from the center of the subject's face towards the left and the right ears respectively. The left and right distinct nystagmus at maximum deviation is obtained by, starting again from the center of the suspect's face, moving the object toward the left and right ears respectively, bringing the eye as far over as possible, and holding the object there for four seconds to ensure that quick movement of the object did not possibly cause the nystagmus. The left and right onset prior to 45 degrees is done by moving the object at a speed that would take about four seconds for the object to reach the edge of the suspect's left and right shoulder respectively. The officer notes this clue if the point or angle at which the eye begins to display nystagmus is before the object reaches forty-five degrees from the center of the suspect's face. This test can be seen in a YouTube [45] video called 'Horizontal Gaze Nystagmus: The Truth is in the Eyes' [46].

3.2 Requirements

The MoSCoW prioritization method [47] is used to classify the requirements of this project. MoSCoW is an acronym for "Must have, Should have, Could have and Won't have", categories in which requirements are divided.

- **Must have requirements:** They are critical for the success of the project.
  - The mobile application must be cross-platform.
  - The user must be able to record videos.
  - The application must determine whether it is safe for the user to drive or not.
  - The application must show instructions to the user to perform the different clues.
  - The system must support modularity and scalability.
- **Should have requirements:** These are important requirements, but not necessary for the system release.
  - The system should store the tests and their clues for further research.
  - The application must have a 'Terms and Conditions' section explaining the usage and treatment of their data.
  - The application should show a preview of the recorded video.
  - The application should have a 'Help' section explaining how the test works.
- **Could have requirements:** Desirable requirements that could improve user's experience or satisfaction. They will be included if there is time at the end of the development.
  - The application could show a loading screen when the videos are being proccessed.
  - The user could give feedback about their experience and accordance with the result of the test.

- The system could provide a dashboard to analyze test results statistics.
  - The application could provide information about alcohol consumption and the hazzards of driving after drinking.
  - The application could allow the user to order a taxi if the test fails.

- **Won't have requirements:** They are inappropiate or the least important ones, so they are not included in the project.
  - The user won't be able to log into the application with any social sign-in (i.e. Google, Facebook).
  - The application won't check the pupil size nor the smooth gaze.
  - The stored user data won't be sensitive.

3.3 Architecture

In the designed system there are two subsystems: the mobile application and a server to host a database and a the backend to proccess data sent from the application. The user interacts with the smarphone sending videos and receiving instructions and feedback from it. The smartphone firstly sends a request to the endpoint to initialize the test and the first clue. Then, it calls the backend to finish the current clue, update the test's end timestamp and create the next clue database entry. For this purpose, it calls each of the endpoints named after each of the clues. The API can perform the create and update operations to the tests and clues on the database, and the database returns an ACK. Figure 7 shows the previously explained architecture of the system.

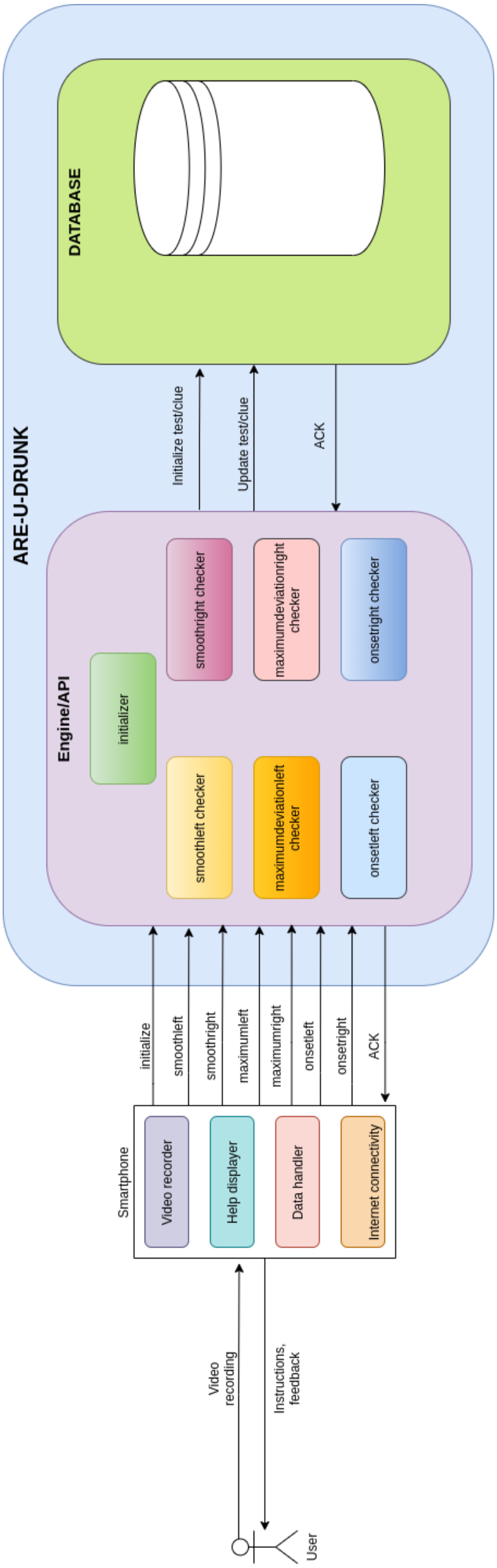


Figure 7: Architecture of the system

3.3.1 Mobile application

When designing the mobile application, the test prerequisites (pupil size and smooth gaze) are not considered because of the limitations of the current technology: the pupil size is very difficult to calculate because it depends on the distance to the camera and the angle of the face with respect to the phone. For this reason, the prerequisites are assumed to be correct and the mobile application is desinged as figure 8 shows.

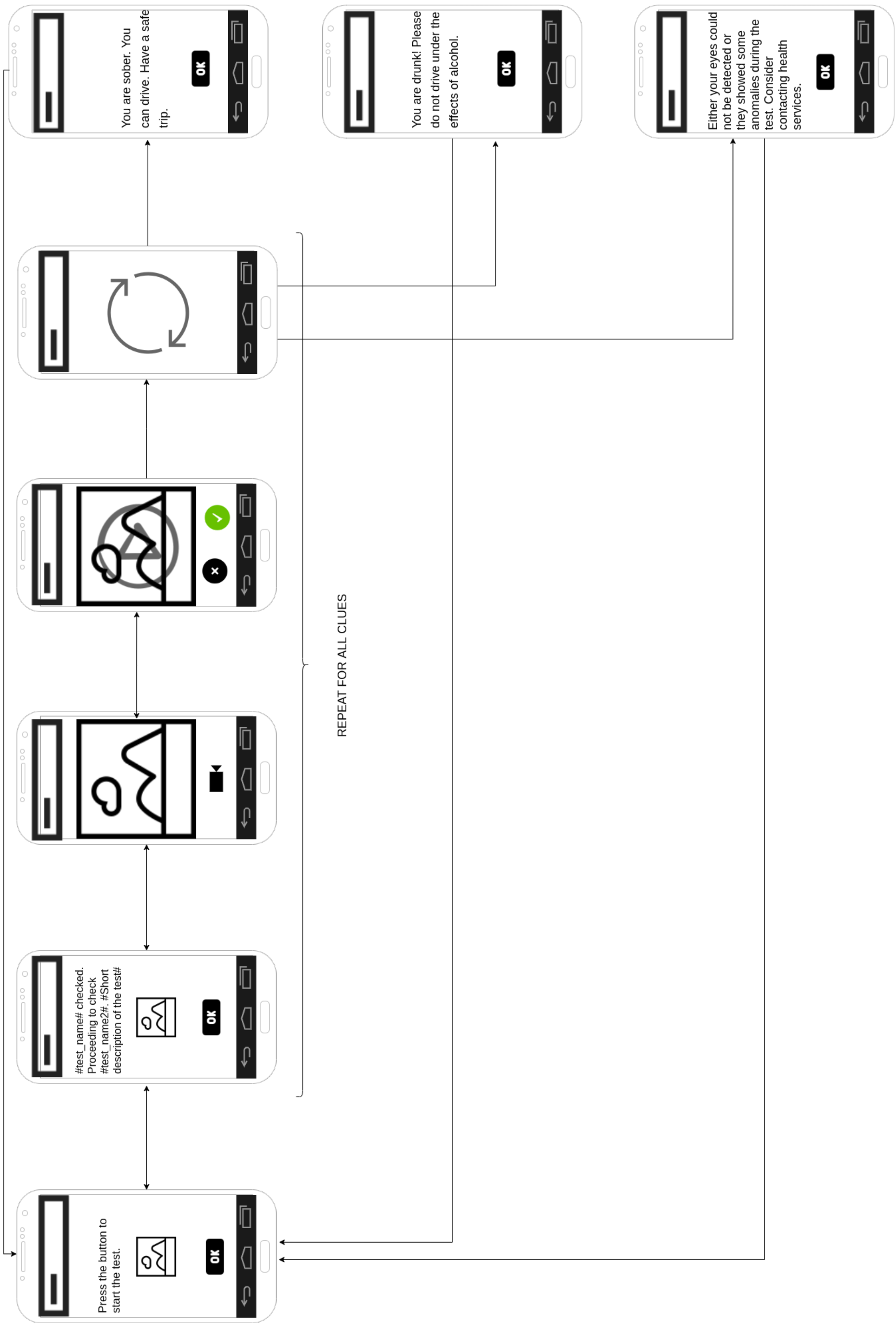


Figure 8: View flow of the application

The application starts with a screen in which the user is informed that the test will start as soon as they click the button. Then, the user is redirected to a screen with the instructions for the clue. When pressing the camera button, the application displays a camera preview in which the user can record a video and then is redirected to a video preview where the user can confirm or discard the recorded video. In case the video is discarded, the user can record another video until the user confirms the video. This is repeated for all of the clues while less than two clues fail. If five or more clues are correct, the user is finally redirected to a screen with a success message. If two clues fail, the test stops and the user is redirected to a screen with a message encouraging the user not to drive. If the eyes are not detected in any of the clues, the test stops and the user is redirected to a screen with a message encouraging the user to reach for professional help.

Figure 9 shows how the user interacts with the application and the application with the server.

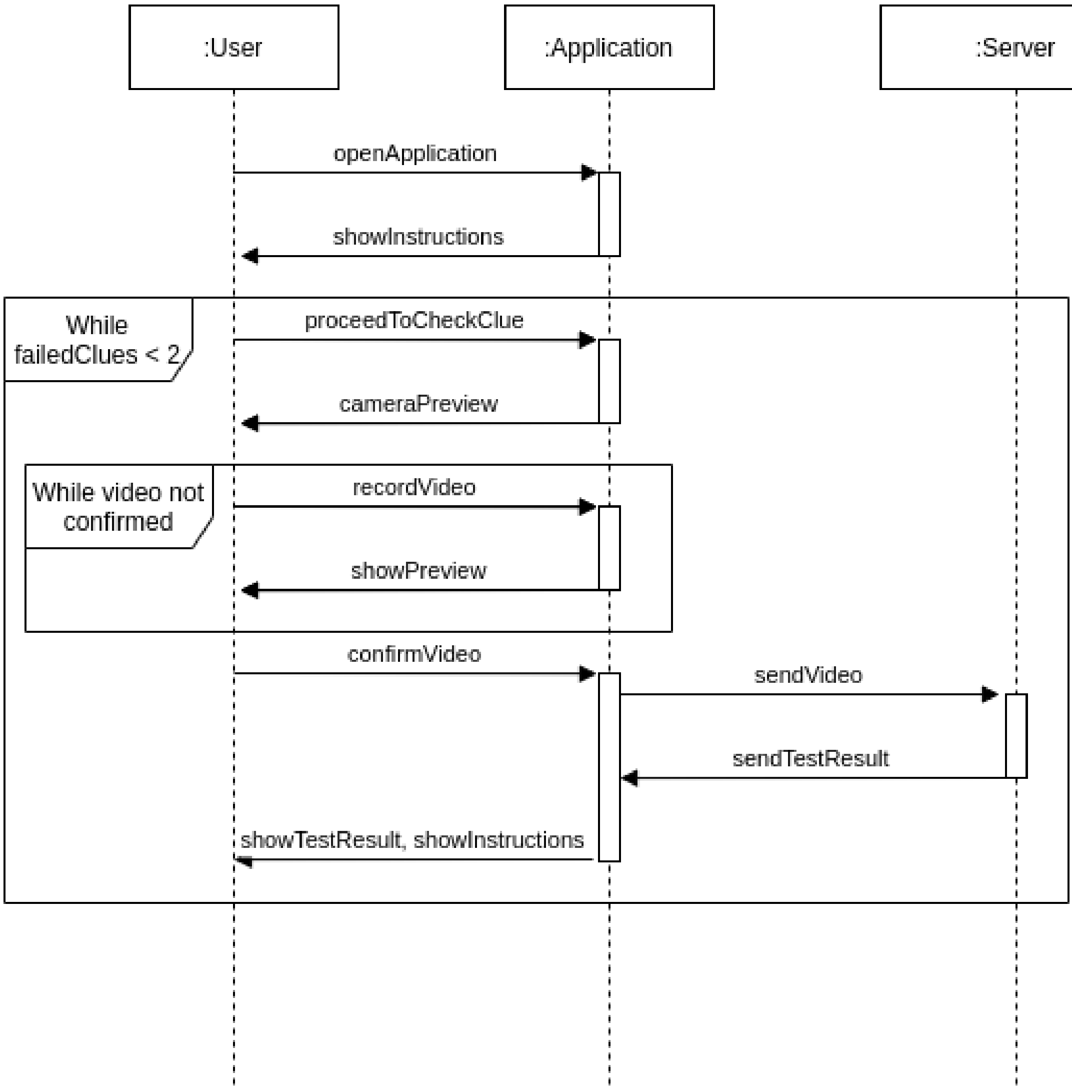


Figure 9: View flow of the application

3.3.2 Server

The server consists on two modules: the database and an REST API that interacts with the database and the application.

The database contains relevant data about the tests and the clues, like:

- Start and end timestamps of the tests.
- An unique user identifier.
- The URL where the video is stored.
- The type of clue.
- The result of the test and each of the clues.

The API receives the clues' data (videos) sent from the application. Each of the clues call their own endpoint that proccesses them to determine whether the clue is passed or failed, returns the results to the mobile application and finally creates the corresponding entries in the database, storing the video in a folder in the server. This interaction is shown in Figure 7.

4 Implementation

When implementing this mobile application, the following technologies are being used:

- **Flutter 2.2.1** [48]: Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications and the web from a single codebase. Flutter was used instead of previously settled technologies such as Java [49], Swift [50] or Kotlin [51] due to the first funtional requirement: cross platform compatibility.

- **Android Studio 4.2** [52]: Android Studio is the official integrated development environment (IDE) for Google's Android [53] operating system. Android Studio is used due to the great compatibility with Flutter.

- **Python 3.8.5** [54]: Python is an interpreted high-level general-purpose programming language. Python is used due to its community and wide documentation about iamge and video documentation.

  - **Flask 2.0.1** [55]: Flask is a micro web framework written in Python. Flask is used due to its ease of use to create APIs. This API is used to integrate the application and Python code.
  - **Gunicorn 20.1.0** [56]: Gunicorn is a Python WSGI HTTP Server for UNIX. It is used to deploy the API.
  - **sqlalchemy 1.4.20** [57]: SQLAlchemy is the Python SQL toolkit and Object Relational Mapper. It is used to design and manage the database.
  - **psycopg2 2.9.1** [58]: Psycopg is the most popular PostgreSQL database adapter for the Python programming language.

- **PostgreSQL 12.7** [59]: PostgreSQL is a powerful, open source object-relational database. It is the chosen database to store the data from the analyzed tests and clues.

- **Docker 20.10.7** [60]: Docker is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers. Docker is used due to its portability and scalability.

- **pgAdmin 5.4** [61]: pgAdmin is the most popular and feature rich Open Source administration and development platform for PostgreSQL, the most advanced Open Source database in the world.

When integrating Python and Flutter, two Flutter plugins were considered:

- **starflut** [62]: This plugin was discarded due to the lack of good documentation, since this would make development, maintenance and addition of functionality difficult.

- **chaquopy** [63]: This plugin is only available for Android [53], discarding it with no further research due to the first funtional requirement: cross platform compatibility.

4.1 Mobile application

For the implementation of the application, four different views have been developed: an info page, a camera preview page, a video preview page and a result page. All of these base pages are parents to the different application screens. This avoids repeating code for similar views. Figure 10 shows the Start Page of the application.

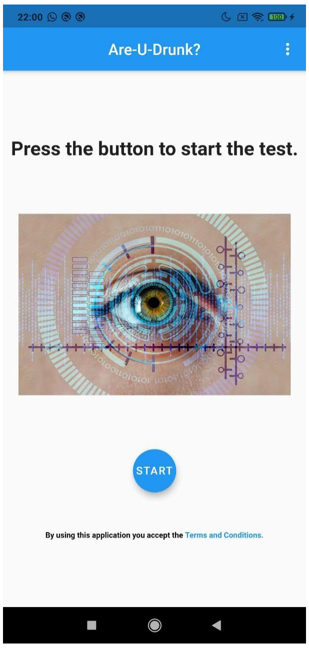


Figure 10: Start Page

Code listing 1 shows how the Popup menu button works and listing 2 how the link to the Terms and Conditions document is created. Listing 3 shows how the button is created and how it calls the backend to initialize the test and the first clue when it is pressed.

```
PopupMenuButton(
      itemBuilder: (BuildContext bc) => [
       PopupMenuItem(child: Text("Help"), value: "/help"),
      ],
      onSelected: (route) {
       Navigator.push(
         context,
         MaterialPageRoute(
         builder: (context) => HelpPage()));
      },
     )
```

Listing 1: Popup menu button code

```
RichText(
 text: new TextSpan(
  children:[
  new TextSpan(
```

```
  text:"By using this application you accept the ",
  style: TextStyle(color: Colors.black, fontSize: 10, fontWeight: FontWeight.bold),
 ),
  TextSpan(
   text: "Terms and Conditions.",
   style: new TextStyle(color: Colors.blue, fontSize: 10,fontWeight: FontWeight.bold),
   recognizer: new TapGestureRecognizer()
   ..onTap = () async {
    final url = '<TCs URL>';
    if (await canLaunch(url)) {
     await launch(url, forceSafariVC: false, forceWebView: false);
    }
   },
  )
 ]
)
)
```

Listing 2: Terms and Conditions document link

```
FloatingActionButton(
        child: Text("START"),
        onPressed: () async {
         await initializeTest();
         Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) =>
              CheckSmoothPursuitLeftPage()),
         );
        },
       ),
...
Future<void> initializeTest() async {
  int fatherId = await initialize();
  SharedPreferences prefs = await SharedPreferences.getInstance();
  await prefs.setInt('fatherId', fatherId);
 }
```

Listing 3: Code to call backend to initialize test and first clue

The calls to the backend are shown in code listings 4 and 5, to call the endpoint to initialize the test and the first clue, and to call an enpoint to proccess each of the clues, respectively. Both calls to the endpoints are done through HTTPS requests.

```
Future<int> initialize() async{
 String identifier;
 try {
  identifier = (await UniqueIdentifier.serial)!;
```

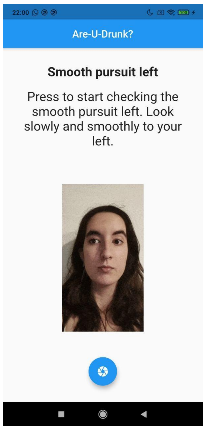
```
  } on PlatformException {
   identifier = 'Failed to get Unique Identifier';
  }
  Uri url = Uri.https(uri, path + "initialize");
  http.MultipartRequest request =
  new http.MultipartRequest("POST", url);
  request.fields['user'] = identifier;
  http.StreamedResponse response = await request.send();
  final body = await response.stream.bytesToString();
  final responseJson = jsonDecode(body);
  return responseJson['father_id'];
}
```

Listing 4: Code to call the initialize endpoint

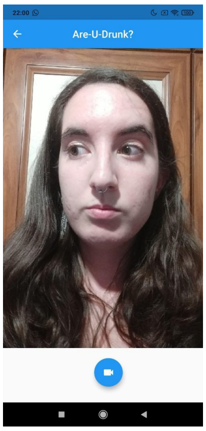
```
Future<String> callBackend(String endpoint, XFile file) async{
  String userIdentifier;
  SharedPreferences prefs = await SharedPreferences.getInstance();
  int fatherId = prefs.getInt('fatherId') ?? 0;
  try {
   userIdentifier = (await UniqueIdentifier.serial)!;
  } on PlatformException {
   userIdentifier = 'Failed to get Unique Identifier';
  }
  Uri url = Uri.https(uri, path + endpoint);
  http.MultipartRequest request =
  new http.MultipartRequest("POST", url);
  http.MultipartFile multipartFile = http.MultipartFile.fromBytes(
    'media', await file.readAsBytes(),
    filename: file.path);
  request.fields['user'] = userIdentifier;
  request.fields['father_test_id'] = fatherId.toString();
  request.files.add(multipartFile);
  http.StreamedResponse response = await request.send();
  final body = await response.stream.bytesToString();
  final responseJson = jsonDecode(body);
  return responseJson['result'];
}
```

Listing 5: Code to call the endpoints to proccess each of the clues

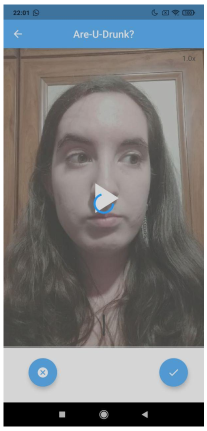
Figure 11a shows the page with information about the first clue. It shows a text and a GIF image explaining how to perform the text. Each of the clue has a similar info page.



(a) Page with information.



(b) Page to record the video.



(c) Page to confirm video.

Code listing 11b shows the page to record a clue. The user clicks the button to record and the phone starts recording and automatically stops after four seconds and redirects the user to the video preview screen, shown in listing 11c. Listing 6 shows the code to record the video and stop it automatically, and code listing 7 shows the call to the backend when the video is confirmed and the code to show a loading animation while the backend is proccessing the request.

```
await _initializeControllerFuture;
await _controller.startVideoRecording();
await Future.delayed(const Duration(seconds: 4), () {});
final image = await _controller.stopVideoRecording();
await Navigator.of(context).push(
MaterialPageRoute(builder: (context) => getNextPage(image.path)));
```

Listing 6: Code to record video automatically

```
context.loaderOverlay.show();
XFile video = XFile(widget.path);
String result = await callBackend(widget.endpoint, video);
context.loaderOverlay.hide();
```

Listing 7: Code to preview the recorded video

The management of the response is shown in code listing 8.

```
if (result == "fail") {
 failedClues += 1;
}

if (result == "undetected") {
 await Navigator.of(context).push(
  MaterialPageRoute(
    builder: (context) => EyeDetectionFailPage()),
 );
```

```
} else {
 if (failedClues < 2) {
  // If the picture was taken, display it on a new screen.
  await Navigator.of(context).push(
   MaterialPageRoute(
     builder: (context) => widget.getNextPage()),
  );
 } else {
  failedClues = 0;
  await Navigator.of(context).push(
   MaterialPageRoute(
     builder: (context) => DrunkPage()),
  );
 }
}
```

Listing 8: Code to manage the response of the backend

When the final result is obtained (eyes not detected, fail or pass), a similar screen to figure 12 is shown.

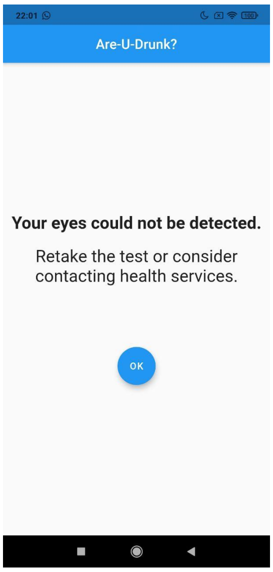


Figure 12: Eyes not detected screen

4.2 Server

The server side uses Docker to host both the database and API. The interaction between the API and the database can be seen in Figure 7. This server is composed by an Application Programming Interface developed with Python and Gunicorn to receive data from the smartphone application and query the database, and a PostgreSQL database to store the videos recorded by de application. As previously mentioned, sqlalchemy is the toolkit used to query the database from the API.

Code listing 9 shows the docker-compose.

```yaml
version: '3'
services:
  database:
    image: "postgres:9.6.22-buster" # use 9.6.22-buster postgres version
    env_file:
      - database.env # configure postgres
    volumes:
      - database-data:/var/lib/postgresql/data/ # persist data even if
        ↪ container shuts down
    ports:
      - "60123:5432"
  backend:
    image: are_u_drunk_rest_api
    build: .
    ports:
      - "60321:5000"
    volumes:
      - clue-videos:/app/videos
volumes:
  database-data:
  clue-videos:
```

Listing 9: docker-compose

With this docker-compose, the database and backend services are created. The database service uses a volume to persist the data inserted when the service stops and starts again. The backend service uses a volume to store the videos recorded by the application and referenced in the database entries.

```dockerfile
FROM python:3.8.11-slim-buster
RUN apt-get update && apt-get install -y libpq-dev gcc python3-dev musl-
   ↪ dev g++ libglib2.0-dev python3-opencv libopencv-dev
COPY requirements.txt /
RUN pip3 install -r /requirements.txt
RUN pip3 install ipython
COPY ./src /app
COPY ./run.sh /app
COPY ./gunicorn.conf.py /app
WORKDIR /app
ENV PYTHONPATH /app
RUN mkdir videos
ENTRYPOINT ["./run.sh"]
```

Listing 10: Dockerfile

Code listing 10 shows how the container is instantiated: a Python Docker image is used, the dependencies and requirements are installed, and finally, the code is copied to the container's root and *run.sh* is set as the entry point. *run.sh* initializes the app with the specified configuration in the Gunicorn config file. Also a directory named "videos" is created to store the videos recorded by the application.

As previously mentioned, Docker is used due to its ease to scale and deploy, and using the command shown in Figure 11 the API and the backend can be set up in the environment (development and production).

```
$> docker-compose up --build d
```

Listing 11: Commands to set the environment up

4.2.1 Database

The database has been implemented following the schema shown in figure 13.

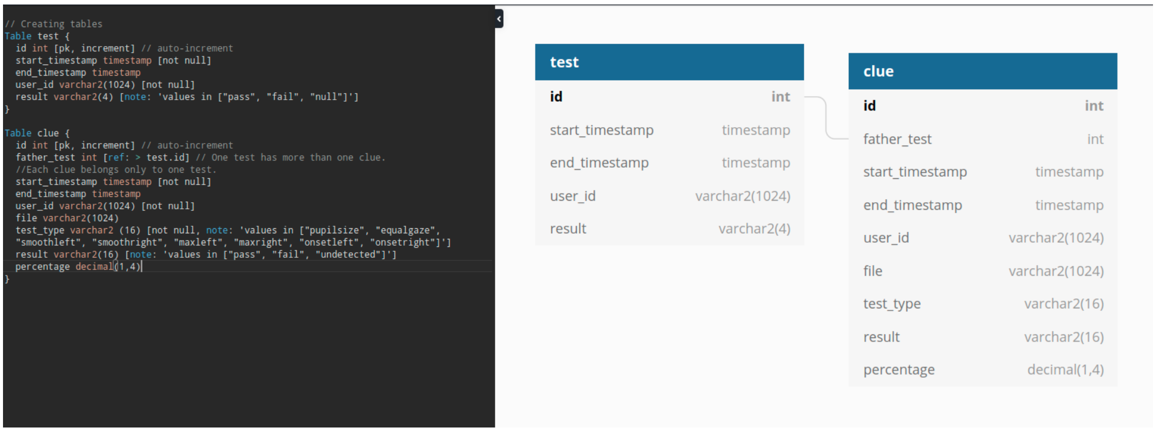


Figure 13: Database schema

The entity *Test* has an id (auto-incremental), a start timestamp (not nullable), an end timestamp (nullable), an unique identifier of the user who took the test (not nullable) and the result (nullable). The start timestamp is inserted as soon as the test begins and the end timestamp is updated as soon as the user ends each of the clue, so we could be able to trace if the user quits before ending the test and where. The result is inserted when the user finishes the last clue, either the 6th clue, the second failed clue or one in which the eyes are not recognized. With this implementation, if the user doesn't finish the test, the result is null and we could query the end timestamp to know in which clue the user quitted.

The entity *Clue* has an id (auto-incremental), the id of the father test (not nullable), a start timestamp (not nullable), an end timestamp (nullable), an unique identifier of the user who took the test (not nullable), the path to the video file (nullable), the test type (not nullable), the result (nullable) and the percentage of success. The start timestamp is inserted as soon as the clue begins and the end timestamp is updated as soon as the user ends each of the clue, so we could be able to trace if the user quits while taking that clue. Therefore, the video file path is nullable since the user could quit and no video would be recorded. The result is inserted when the user finishes the clue. With this implementation, we could query either the result or the end timestamp to know if the user finished the clue.

4.2.2 Python API

The API consists of two classes to integrate with the database, seven different endpoints and seven different methods to analyze the video. The model definition for 'Clue' entity is shown in figure 12.

```
class Clue(Base):
  __tablename__ = 'clue'
  id = Column(Integer, primary_key=True)
  father_test = Column(Integer, ForeignKey('test.id'))
  start_timestamp = Column(DateTime, nullable=False)
  end_timestamp = Column(DateTime, nullable=True)
  user_id = Column(String, nullable=False)
  file = Column(String, nullable=True)
  test_type = Column(String, nullable=False)
  result = Column(String, nullable=True)
  def __init__(self, father_test, start_timestamp, end_timestamp, user_id, file, test_type,
   ↪ result):
    self.father_test = father_test
    self.start_timestamp = start_timestamp
    self.end_timestamp = end_timestamp
    self.user_id = user_id
    self.file = file
    self.test_type = test_type
    self.result = result
```

Listing 12: Clue class in clue.py

Out of the seven different endpoints, one is to initialize the test and the first clue and the six remaining are called when a clue ends. Figure 13 shows the endpoint called when the maximum deviation right is checked.

```
@app.route('/maximumright', methods=['POST'])
def maximum_right():
 user = request.form['user']
 father_test_id = int(request.form['father_test_id'])
 time = datetime.utcnow()
 filename=f"{user}-{time.strftime('\%Y-\%m-\%d \%H:\%M:\%S')}-maximumright.mp4"
 request.files['media'].save(filename)
 perc = get_percentage(filename, 'right')
 if perc > LIMIT:
   return_result = "pass"
 elif perc == 0:
   return_result = "undetected"
 else:
   return_result = "fail"
 _insert_into_db(father_test_id=father_test_id, return_result=return_result, user=user,
  ↪ filename=filename,
```

```python
        current_clue_type="maximumright", time=time, percentage=perc)
    return {'result': return_result}
```

Listing 13: Enpoint to proccess the maximum deviation right

The six endpoints to proccess the clues follow the same structure: the request parameters are obtained (user who called the backend, the id assigned to the test that groups all the clues and the video recorded) and the video is saved, naming it after the user, the timestamp of the call and the type of clue. Then, the method to track and analyze the gaze and determine whether the clue is passed or not and depending on the result, the corresponding operations are performed in the database. Finally, the result is sent as a response. Figure 14 shows the method previously mentioned. This code is based on Proctoring-AI project [64].

```python
def get_percentage(video, gaze):
try:
  left_data_list = []
  right_data_list = []
  face_model = get_face_detector()
  landmark_model = get_landmark_model()
  left = [36, 37, 38, 39, 40, 41]
  right = [42, 43, 44, 45, 46, 47]
  cap = cv2.VideoCapture(video)
  kernel = np.ones((9, 9), np.uint8)
  while cap.isOpened():
    ret, img = cap.read()
    if ret:
      rects = find_faces(img, face_model)

      for rect in rects:
        try:
          shape = detect_marks(img, landmark_model, rect)
        except cv2.error:
          continue
        mask = np.zeros(img.shape[:2], dtype=np.uint8)
        mask, end_points_left = eye_on_mask(mask, left, shape)
        mask, end_points_right = eye_on_mask(mask, right, shape)
        mask = cv2.dilate(mask, kernel, 5)
        eyes = cv2.bitwise_and(img, img, mask=mask)
        mask = (eyes == [0, 0, 0]).all(axis=2)
        eyes[mask] = [255, 255, 255]
        mid = int((shape[42][0] + shape[39][0]) // 2)
        eyes_gray = cv2.cvtColor(eyes, cv2.COLOR_BGR2GRAY)
        threshold = 75
        _, thresh = cv2.threshold(eyes_gray, threshold, 255, cv2.THRESH_BINARY)
        thresh = process_thresh(thresh)
        contouring(thresh[:, 0:mid], mid, img, end_points_left, False, right_data_list, left_data_list)
        contouring(thresh[:, mid:], mid, img, end_points_right, True, right_data_list, left_data_list)
        for (x, y) in shape[36:48]:
          cv2.circle(img, (x, y), 2, (255, 0, 0), -1)
    else:
      break
  cap.release()
  cv2.destroyAllWindows()
  try:
```

```
    if gaze == 'right':
      result = count_wrong_meassurements(right_data_list, True) / len(right_data_list)
    else:
      result = count_wrong_meassurements(left_data_list) / len(left_data_list)
    return 1 - result
  except ZeroDivisionError:
    return 0
except cv2.error:
  logger.exception("Exception detecting eyes")
  return 0
```

Listing 14: Method to track and analyze the gaze and determine whether the clue is passed or not

In the get_percentage method, two arrays are firstly created to store the points obtained from the eyes, the face detector and landmark model are instantiated and the landmark coordinates for the left and right eyes are harcoded. These landmark coordinates are obtained from the Dlib facial keypoints. Then, the video is analyzed frame by frame, detecting firstly the eyes with the eye_on_mask method and then detecting the eye position with the contouring method. All of the positions are stored in the corresponding data list and when the video is completely analyzed, the percentage of positions that are invalid is calculated. A position is considered invalid when the previous one is greater in the case of a left gaze and smaller in the case of a right gaze. If no data is stored in the list, eyes have not been detected and the method returns zero.

4.2.3 Deployment

This project has been uploaded to a server provided by de University of Granada [65]. For this purpose, the requests have been proxied through Nginx [66]. This is also useful to isolate the Are-U-Drunk project from the rest of the projects hosted in the server. To deploy the system, Docker is required in the server. Once this requirement is met, the application has to be downloaded from the repository and copied to the server. Then, the Gunicorn config file has to be modified to set the new path for the requests and using the command shown in listing 15 the containers for the services start working.

```
docker-compose up
```

Listing 15: Command to start the containers

The database is created by means of two scripts, one for each table. Listing 16 shows the script used to create the table for the tests. A database.env file has to be created to configure secure credentials to access the database.

```
SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
```

```
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;
SET default_tablespace = '';
CREATE TABLE public.test (
    id integer NOT NULL,
    start_timestamp timestamp without time zone NOT NULL,
    end_timestamp timestamp without time zone,
    user_id character varying NOT NULL,
    result character varying
);
ALTER TABLE public.test OWNER TO cdg;
CREATE SEQUENCE public.test_id_seq
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
ALTER TABLE public.test_id_seq OWNER TO cdg;
ALTER SEQUENCE public.test_id_seq OWNED BY public.test.id;
ALTER TABLE ONLY public.test ALTER COLUMN id SET DEFAULT nextval('public.
    ↪ test_id_seq'::regclass);
ALTER TABLE ONLY public.test
    ADD CONSTRAINT test_pkey PRIMARY KEY (id);
```

Listing 16: Script to create the table for the tests

Finally, the path to the application is added to nginx to redirect the request and the service is restarted. Listing 17 shows this change.

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;
        server_name _;
        return 301 https://hostrequest_uri;

}

server {
        listen 443 ssl default_server;
        listen [::]:443 ssl default_server;
        ...
  location /are_u_drunk_backend {
    proxy_pass http://localhost:60321;
    proxy_set_header Host host;
    proxy_set_header X-Real-IP remote_addr;
    proxy_set_header X-Forwarded-For proxy_add_x_forwarded_for;
```

```
        }
  ...
}
```

Listing 17: Nginx configuration file

5 Evaluation

5.1 Methodology

The mobile application 'Are-U-Drunk?' has been tested with 8 people, aged from 18 to 58, 4 female and 8 male. The 8 participants were tested using the application once before drinking different amounts of alcohol and once after drinking it. If any inconvenience happpened (eyes could not be detected for any reason), the test was repeated. This mobile testing happened when and were the participant desired any time before drinking and 15-30 minutes after drinking. The day after the last mobile test, a form was provided to take a System Usability Scale (SUS) [10] test.

5.2 Usability

To test the usability of the system, SUS [10] has been used. This test is composed by 10 statements, in each of which the respondent states how they feel about the statement from 1 to 5, being 1 strongly disagree and 5 strongly agree. The 10 statements are the following.

- I think that I would like to use this system frequently.
- I found the system unnecessarily complex.
- I thought the system was easy to use.
- I think that I would need the support of a technical person to be able to use this system.
- I found the various functions in this system were well integrated.
- I thought there was too much inconsistency in this system.
- I would imagine that most people would learn to use this system very quickly.
- I found the system very cumbersome to use.
- I felt very confident using the system.
- I needed to learn a lot of things before I could get going with this system.

Figure 14 represents the results. When asking if the participant would like to use this application frequently, 25% strongly agreed, 37.5% agreed and 37.5% were neutral about the question. The system was not unnecessarily complex, 62.5% of the users strongly disagreed about the application being complex and 37.5% disagreed. 75% of the participants strongly agreed the system was wasy to use, 12.5% agreed and 12.5% were neutral. Most of the participants, 87.5%, strongly disagreed they would need the support of a technical person to be able to use the application and 12.5% were neutral. When asking if the user found the various functions in the system were well integrated, 75% strongly agreed, 12.5% agreed and 12.5% were neutral. The participants' opinions on whether there was too much inconsistency in this application are: 62.5% strongly disagreed, 12.5% disagreed, 12.5% were neutral and 12.5% agreed. 62.5% of the users strongly agreed that most people would learn to use this system very quickly, 25% agreed and 12.5% were neutral. 75%

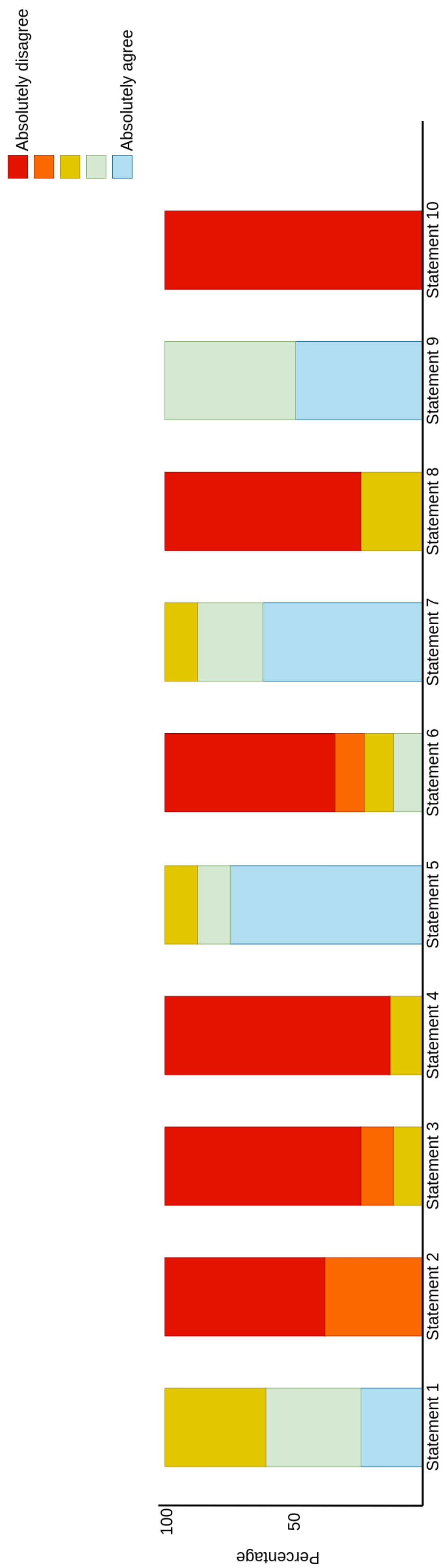


Figure 14: System Usability Scale responses' graph

strongly disagreed the system was very cumbersome to use while 25% was neutral about it. 50% agreed they felt very confident using the application and 50% strongly agreed. Finally, 100% strongly disagreed they needed to learn a lot of thing before get going with the tests.

5.3 Performance

Participant	Gender	Age	Amount drank	Samples	Before or after drinking	Result
Participant 1	Female	25	Half a glass of wine	Sample 1	Before	Null
				Sample 2	Before	Fail
				Sample 3	After	Undetected
				Sample 4	After	Fail
Participant 2	Male	23	Half a glass of wine	Sample 1	Before	Fail
				Sample 2	Before	Fail
				Sample 3	After	Fail
Participant 3	Male	54	Two beers and two glasses of wine	Sample 1	Before	Fail
				Sample 2	After	Fail
Participant 4	Male	18	Half a glass of wine	Sample 1	Before	Fail
				Sample 2	After	Undetected
				Sample 3	After	Fail
Participant 5	Female	53	A glass of wine	Sample 1	Before	Fail
				Sample 2	After	Fail
Participant 6	Male	58	One and a half rum and coke	Sample 1	Before	Fail
				Sample 2	After	Fail
Participant 7	Female	54	Half a rum and coke	Sample 1	Before	Fail
				Sample 2	After	Fail
Participant 8	Female	22	A beer	Sample 1	Before	Fail
				Sample 2	After	Undetected
				Sample 3	After	Fail

Table 8: Subjects and samples of the application

As Table 8 shows, gender and age have been considered in the study participants, having an equal percentage of male and female people and a wide age range. Most of the tests are failed, three did not detect the eyes and one was quitted. The results of the tests were mainly 'fail'. Querying the database with the query shown in Listing 18, we obtain the following data: 4 out of the 19 tests taken by all of the participants finished with only two clues, 6 finished with 3 clues, 5 finished with 4 clues and 6 finished in the 6th and last clue.

```
select acc, count(*) from (select count(*) as clues from public.clue
  ↪ group by father_test having count(*) > 1) as acc group by acc
```

Listing 18: SQL query used to group the tests by the number of clues taken

The percentages of accuracy of eye movement in the failed clues were between 41.91% and 56% when the first two clues are failed. When the tests stopped at the third clue, the percentages varied from 50.86% to 60%, when the fourth was the last clue, the percentages were between 42.86% and 60% and finally, if the test finished with a failed last clue, the percentages varied from 50.36% to 58.9%. Some of the failed clues' videos were too dark, which may have caused the algorithm not to detect and track the eyes properly. Glasses also appear to affect the detection since all of the clues with a result of 'undetected' were taken with glasses. This may also affect the tracking of the eyes, but further research has to be done to affirm this. Furtuhermore, some of the videos were recorded too fast, meaning this that the eyes were moving faster than expected. After drinking, then number of passed clues was lower than the previous test when the participant was sober.

6 Conclusions

In this section, the objectives previously set will be analyzed. Furthermore, possible future work will be defined.

6.1 Achieved objectives

- **Main goal: to develop a system to automatically assess a person's drunkness based on the analysis of their eye(s) movement.** A complete system, composed by a mobile phone (camera and internet connection included) and a back-end hosted in a server, has been developed and the functionality and usability of it has been tested by a small sample of people.

- **Secondary goals:**

  - **To design a system that detects and tracks the user's eyes and analyzes them in search of horizontal nystagmus movements.** The designed architecture is composed by the mobile application that, according to the proposed requirements, will record some videos to detect and track the eyes, send them to a server and proccess them.
  - **To develop a system that detects and tracks the user's eyes and analyzes them in search of horizontal nystagmus movements.** This goal has been achieved by means of analyzing the videos sent by the user and tracking the position of the eyes for every frame to check the movement looking for horizontal nystagmus movements.
  - **To evaluate and test the previously mentioned system.** A small sample of people used the designed and developed application before and after drinking. These people have also taken an usability test to check the ease of use of the application.

6.2 Future work

The mobile application 'Are-U-Drunk' could be improved in many ways. Some of the proposals for future work are:

- Improve the algorith to be resilient to different illumination levels. This will result in a more accurate test result since the eyes and the gaze could be detected in more complicated scenarios.

- Create and train a new model from scrath. This could help the algorithm detect the eyes faster and better.

- Improve the user experience by features like clinging or vibrating when the camera stops recording.

- Link a video with a deeper explanation about the test and the clues so the user knows exactly how the video must be recorded.

Appendices

A State of the Art Matrix

https://docs.google.com/spreadsheets/d/1mtM97CYdcI7DFjK2QBlT6idoHjkSppf8CLmN4FE_H24/edit?usp=sharing

Title	URL	Summary	Year	Countries from the affiliations	Sample	Average age
Estimation o	https://www.sco	Use of smartphone to detect alcohol when driving (among others uses)	2021/01	Rhode Island and Massachusetts	65, phase(s) not specified	30.79. SD 12.01
An Automate	https://www.sco	Alcohol use identification through mobile games	2020/09	UK and Thailand	40 for study, 40 for control	42.08. SD 10.29
Understandi	https://www.sco	Excessive alcohol consumption over-nights in young people	2020/05	Switzerland and Australia	241 participants and 847 user-nights from a dat	19.4. SD 2.5. Ag
A novel tele	https://www.sco	Study of a very complete app with several proffesionals to improve people's us	2019/10	Maryland and California, USA	77	49. SD 10. Ages
Reducing dr	https://www.sco	App for homeless with excessive alcohol consumption	2020/04	Oklahoma, Texas and New York,	80 for developing, 40 for testing	-
A preliminar	https://www.sco	Use of accelerometer to detect alcohol consumption's effects on people	2020/08	Pensilvania, USA	17	27.5. SD 5.5. Ag
Evaluating th	https://www.sco	App for veterans with excessive alcohol consumption	2020/10	London and Leatherhead, UK	37 for study, 37 for control	Over 18. SD and
Portable Alc	https://www.sco	Sensor and app to detect alcohol consumption in drivers	2019/10	Japan	-	-
Learning to	https://www.sco	Detection of alcohol through accelerometer	2019/08	Ohaio and California, USA	20 (1 participant didn't install the app properly a	22. Ages from 2
Eyeglasses-	https://www.sco	Estimation of BAC with a pair of glasses with several sensors	2019/07	USA, Brazil and Spain	3	-
Drunkselfie:	https://www.sco	Alcohol consumption detection and classification through selfies	2019/07	Massachusetts, USA	4 photos of each 53 people from a dataset	-
The develop	https://www.sco	App for people with excessive alcohol use	2018/05	London, UK	-	-
Driving Fitne	https://www.sco	Detección del efecto del alcohol por detección facial a través del móvil	2018/09	India	-	-
A smartphor	https://www.sco	App + methodology of the development	2018/09	London and Liverpool, UK	31	16% aged from
Assessment	https://www.sco	App to evaluate alcohol consumption	2018/08	Australia	671	23.12. Ages fron
Drunk user i	https://www.sco	App that simulates breathalyser	2018/04	Washington, USA	14	25.7. Ages from
Using phone	https://www.sco	App with sensors	2018/02	Pensilvania, USA	10	23.1. Ages from
Mobile healt	https://www.sco	App + physical monitoring	2017/07	New York, USA	10	20.7. Ages from
Smartphone	https://www.sco	App to drink less	2017/07	Australia, Switzerland, USA, Can	130	32.8. SD 10.0. A
Electronic br	https://www.sco	Electronical bracelet to measure alcohol in sweat	2017/01	Texas, USA	-	-
A smarter pa	https://www.sco	App to treat alcoholism	2017/01	Denmark	2 groups of 5 people each	-
Unobtrusive	https://www.sco	Monitoring of alcoholism thorugh smart shoes with sensors	2016/10	Seul and Massachussets, South	20	24.78. SD 4.97.

Figure 15: Preview of the State of the Art Matrix

Bibliography

[1] U. Nations, "World health organization," https://www.who.int/, 2021.

[2] Elsevier, "Scopus," https://www.scopus.com/freelookup/form/author.uri, 2004.

[3] Google, "Google play," https://play.google.com/store?hl=es&gl=US, 2008.

[4] U. S. N. H. T. S. Administration, "Horizontal gaze nystagmus : the science and the law : a resource guide for judges, prosecutors and law enforcement," https://rosap.ntl.bts.gov/view/dot/1647, 1999.

[5] M. Businelle, S. Walters, E.-Y. Mun, T. Kirchner, E. Hébert, and X. Li, "Reducing drinking among people experiencing homelessness: Protocol for the development and testing of a just-in-time adaptive intervention," *JMIR Research Protocols*, vol. 9, no. 4, 2020, cited By 2. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85083420810&doi=10.2196%2f15610&partnerID=40&md5=cffeb1ff1c9cdd80ccbea5b8c1059f2d

[6] U. Nations, "World health organization - alcohol," https://www.who.int/news-room/fact-sheets/detail/alcohol, 2021.

[7] M. N. Today, "Alcohol: Short-term and long-term effects," https://www.medicalnewstoday.com/articles/305062#short-term-effects-, 2021.

[8] A. Pietrangelo, "23 effects of alcohol on your body," https://www.healthline.com/health/alcohol/effects-on-body#Inflammatory-damage, 2021.

[9] U. N. L. of Medicine, "Medlineplus - nystagmus," https://medlineplus.gov/ency/article/003037.htm, 2021.

[10] J. Brooke, "System usability scale (sus)," https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html, 1986.

[11] BACtrack, "Bactrack skyn wearable alcohol monitor," https://skyn.bactrack.com/, 2016.

[12] Samsung, "Samsung copilot," https://www.samsung.com/es/tecnologiaconproposito/copiloto/en/, 2017.

[13] Apple, "App store," https://www.apple.com/es/ios/app-store/, 2008.

[14] ——, "Apple watch," https://www.apple.com/es/watch/, 2015.

[15] I. Drizly, "Drizly: Alcohol delivery. order wine beer & liquor (version 4.17.5)," https://play.google.com/store/apps/details?id=com.drizly.Drizly&hl=es&gl=US, 2021.

[16] F. Apps, "Alcotrack (version 3.3.5)," https://play.google.com/store/apps/details?id=com.felixheller.alcdroid, 2021.

[17] a. Dine4Fit, "Alcohol check - bac calculator (version 1.1.10)," https://play.google.com/store/apps/details?id=com.aspamobile.alkoholmetr, 2020.

[18] J. Dev, "Alcofy - alcohol tracker and bac drink calculator (version 3.0.0)," https://play.google.com/store/apps/details?id=com.jmstudio.alctrack, 2021.

[19] Crabtree, "Blood alcohol calculator (version 1.16)," https://play.google.com/store/apps/details?id=com.crabtree.bac, 2019.

[20] D. Jam, "Simple alcohol unit tracker (version 1.0)," https://play.google.com/store/apps/details?id=com.developerjam.drinklogger, 2021.

[21] Simopt, "Blood alcohol calculator (version 2.12)," https://play.google.com/store/apps/details?id=cz.simopt.besipalkoholkalkulacka, 2018.

[22] S. Aleksey, "Alco calculator (version 1.68.1)," https://play.google.com/store/apps/details?id=com.sag.AlcoCalculator, 2021.

[23] N. Utilities, "Alcohol test (for fun) (version 1.2)," https://play.google.com/store/apps/details?id=pl.netigen.new_alcohol_test, 2018.

[24] R. Li, G. Balakrishnan, J. Nie, Y. Li, E. Agu, K. Grimone, D. Herman, A. Abrantes, and M. Stein, "Estimation of blood alcohol concentration from smartphone gait data using neural networks," *IEEE Access*, 2021, cited By 0. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85100519004&doi=10.1109%2fACCESS.2021.3054515&partnerID=40&md5=26bbe0276d50721086f16145d8762422

[25] M. Mitchell, J. Mendelson, J. Gryczynski, S. Carswell, and R. Schwartz, "A novel telehealth platform for alcohol use disorder treatment: preliminary evidence of reductions in drinking," *American Journal of Drug and Alcohol Abuse*, vol. 46, no. 3, pp. 297–303, 2020, cited By 1. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85074047623&doi=10.1080%2f00952990.2019.1658197&partnerID=40&md5=3fe319b7c4635fef505ee4c3ccbcb354

[26] B. Suffoletto, P. Dasgupta, R. Uymatiao, J. Huber, K. Flickinger, and E. Sejdic, "A preliminary study using smartphone accelerometers to sense gait impairments due to alcohol intoxication," *Journal of Studies on Alcohol and Drugs*, vol. 81, no. 4, pp. 505–510, 2020, cited By 2. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85089708825&doi=10.15288%2fjsad.2020.81.505&partnerID=40&md5=e8295bd65723d0fbe22f6c565f0f1bfc

[27] J. Killian, K. Passino, A. Nandi, D. Madden, and J. Clapp, "Learning to detect heavy drinking episodes using smartphone accelerometer data," vol. 2429, 2019, pp. 35–42, cited By 1. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85071677609&partnerID=40&md5=2c3f2aed1f07cce8c441666c8672bc4d

[28] C. Willoughby, I. Banatoski, P. Roberts, and E. Agu, "Drunkselfie: Intoxication detection from smartphone facial images," vol. 2, 2019, pp. 496–501, cited By 1. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85072664870&doi=10.1109%2fCOMPSAC.2019.10255&partnerID=40&md5=56060b77a3485f27927e57f329c2628a

[29] A. Mariakakis, S. Parsi, S. Patel, and J. Wobbrock, "Drunk user interfaces: Determining blood alcohol level through everyday smartphone tasks," vol. 2018-April, 2018, cited By 8. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85046963575&doi=10.1145%2f3173574.3173808&partnerID=40&md5=9f891757aa85113fd5028567beb1021c

[30] B. Suffoletto, P. Gharani, T. Chung, and H. Karimi, "Using phone sensors and an artificial neural network to detect gait changes during drinking episodes in the natural environment," *Gait and Posture*, vol. 60, pp. 116–121, 2018, cited By 11. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85034813240&doi=10.1016%2fj.gaitpost.2017.11.019&partnerID=40&md5=1497731738c817654d9053585eee9c18

[31] N. Leonard, M. Silverman, D. Sherpa, M. Naegle, H. Kim, D. Coffman, and M. Ferdschneider, "Mobile health technology using a wearable sensorband for female college students with problem drinking: An acceptability and feasibility study," *JMIR mHealth and uHealth*, vol. 5, no. 7, 2017, cited By 10. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85047761057&doi=10.2196%2fmhealth.7399&partnerID=40&md5=37f8c669f9c7d47dae16aaf34bd409d7

[32] D. Kinnamon, A. Selvam, S. Prasad, and S. Muthukumar, "Electronic bracelet for monitoring of alcohol lifestyle," 2017, cited By 1. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85011006784&doi=10.1109%2fICSENS.2016.7808598&partnerID=40&md5=98048a57473bc7a66813beab5ad1bdde

[33] J. Sempionatto, L. Brazaca, L. García-Carmona, G. Bolat, A. Campbell, A. Martin, G. Tang, R. Shah, R. Mishra, J. Kim, V. Zucolotto, A. Escarpa, and J. Wang, "Eyeglasses-based tear biosensing system: Non-invasive detection of alcohol, vitamins and glucose," *Biosensors and Bioelectronics*, vol. 137, pp. 161–170, 2019, cited By 35. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85065477317&doi=10.1016%2fj.bios.2019.04.058&partnerID=40&md5=e87a68d7cf144d84ccfaa76e51089887

[34] N. Bertholet, J.-B. Daeppen, J. McNeely, V. Kushnir, and J. Cunningham, "Smartphone application for unhealthy alcohol use: A pilot study," *Substance Abuse*, vol. 38, no. 3, pp. 285–291, 2017, cited By 7. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85015693569&doi=10.1080%2f08897077.2017.1281860&partnerID=40&md5=26e7147aa096cb863ce6621796e67960

[35] E. Park, S. Lee, H. Nam, J. Garst, A. Huang, A. Campion, M. Arnell, N. Ghalehsariand, S. Park, H.-J. Chang, D. Lu, and M. Sarrafzadeh, "Unobtrusive and continuous monitoring of alcohol-impaired gait using smart shoes," *Methods of Information in Medicine*, vol. 56, no. 1, pp. 74–82, 2017, cited By 8. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85009415836&doi=10.3414%2fME15-02-0008&partnerID=40&md5=aa8370db66d56d7d02094616fac37a39

[36] D. Leightley, R. Rona, J. Shearer, C. Williamson, C. Gunasinghe, A. Simms, N. Fear, L. Goodwin, and D. Murphy, "Evaluating the efficacy of a mobile app (drinks:ration) and personalized text and push messaging to reduce alcohol consumption in a veteran population: Protocol for a randomized controlled trial," *JMIR Research Protocols*, vol. 9, no. 10, 2020, cited By 0. [Online].

Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85095776628&doi=10.2196%2f19720&partnerID=40&md5=2fd4680961b34fbe75796c3ab01dbd3a

[37] C. Garnett, D. Crane, R. West, J. Brown, and S. Michie, "The development of drink less: An alcohol reduction smartphone app for excessive drinkers," *Translational Behavioral Medicine*, vol. 9, no. 2, pp. 296–307, 2019, cited By 13. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85057619156&doi=10.1093%2ftbm%2fiby043&partnerID=40&md5=e431e1c2ebbf899107e043a73b171044

[38] D. Leightley, J.-A. Puddephatt, N. Jones, T. Mahmoodi, Z. Chui, M. Field, C. Drummond, R. Rona, N. Fear, and L. Goodwin, "A smartphone app and personalized text messaging framework (index) to monitor and reduce alcohol use in ex-serving personnel: Development and feasibility study," *JMIR mHealth and uHealth*, vol. 6, no. 9, 2018, cited By 14. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85060397231&doi=10.2196%2f10074&partnerID=40&md5=c8c152f5c8c084ec2aff17fe27b580ce

[39] A. Poulton, J. Pan, J. Bruns, L.R., R. Sinnott, and R. Hester, "Assessment of alcohol intake: Retrospective measures versus a smartphone application," *Addictive Behaviors*, vol. 83, pp. 35–41, 2018, cited By 11. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85034424784&doi=10.1016%2fj.addbeh.2017.11.003&partnerID=40&md5=8324ab9838b94b0938f67da428c870cc

[40] T.-T. Phan, F. Labhart, S. Muralidhar, and D. Gatica-Perez, "Understanding heavy drinking at night through smartphone sensing and active human engagement," 2020, pp. 211–222, cited By 1. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85097270846&doi=10.1145%2f3421937.3421992&partnerID=40&md5=01ec371fd543f40cf47840e53552bc70

[41] H. Wakana and M. Yamada, "Portable alcohol detection system for driver monitoring," vol. 2019-October, 2019, cited By 2. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85078702800&doi=10.1109%2fSENSORS43011.2019.8956885&partnerID=40&md5=2307489bb925509a66b5346ae4691913

[42] I. Chatterjee, Isha, and A. Sharma, "Driving fitness detection : A holistic approach for prevention of drowsy and drunk driving using computer vision techniques," 2018, cited By 3. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85059783100&doi=10.23919%2fSEEDA-CECNSM.2018.8544944&partnerID=40&md5=30af93d8fb8dc6a3c7689ee980ed7cf0

[43] A. Mellentin, E. Stenager, B. Nielsen, A. Nielsen, and F. Yu, "A smarter pathway for delivering cue exposure therapy? the design and development of a smartphone app targeting alcohol use disorder," *JMIR mHealth and uHealth*, vol. 5, no. 1, 2017, cited By 9. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85042122489&doi=10.2196%2fmhealth.6500&partnerID=40&md5=d9b3d391f5549ea1196df008966619a6

[44] J. Intarasirisawat, C. Ang, C. Efstratiou, L. Dickens, N. Sriburapar, D. Sharma, and B. Asawathaweeboon, "An automated mobile game-based screening tool for patients with alcohol dependence," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 3, 2020, cited By 0. [Online].

Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85092429721&doi=10.1145%2f3411837&partnerID=40&md5=21a8cd76072a85c88f3be68cda884f28

[45] Google, "Youtube," https://youtu.be/EAryg2X2J_4, 2005.

[46] D. A. Recognition, "Horizontal gaze nystagmus: The truth is in the eyes," https://youtu.be/EAryg2X2J_4, 2014.

[47] D. Clegg and R. Barker, *CASE Method Fast-track: A RAD Approach.* Addison-Wesley Publishing Company, 1994.

[48] Google, "Flutter," https://flutter.dev/, 2017.

[49] S. Microsystems, "Java," https://www.java.com/es/, 1996.

[50] Apple, "Swift," https://developer.apple.com/swift/, 2014.

[51] JetBrains, "Kotlin," https://kotlinlang.org/, 2016.

[52] Google, "Android studio," https://developer.android.com/studio?hl=es, 2014.

[53] ——, "Android," https://www.android.com/intl/es_es/, 2008.

[54] P. S. Foundation, "Python," https://www.python.org/, 1991.

[55] A. Ronacher, "Flask," https://flask.palletsprojects.com/en/2.0.x/, 2010.

[56] Google, "Gunicorn," https://gunicorn.org/, 2017.

[57] M. Bayer, "sqlalchemy," https://www.sqlalchemy.org/, 2021.

[58] T. P. Team, "psycopg2," https://pypi.org/project/psycopg2/, 2021.

[59] P. G. D. Group, "Postgresql," https://www.postgresql.org/, 1996.

[60] I. Docker, "Docker," https://www.docker.com/, 2013.

[61] T. pgAdmin Development Team, "pgadmin," https://www.pgadmin.org/, 2013.

[62] srplab, "starflut," https://pub.dev/packages/starflut, 2018.

[63] J. A. Dangar, "chaquopy," https://pub.dev/packages/chaquopy, 2021.

[64] vardanagarwal, "Proctoring-ai," https://github.com/vardanagarwal/Proctoring-AI, 2020.

[65] C. I. de España, "University of granada (ugr)," https://www.ugr.es/, 1531.

[66] I. Sysoev, "Nginx," https://www.nginx.com/, 2004.