ManDown - A Social Drinking Management App - Individual Component Report

Author: Michael Hart
Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ
E-mail: mh1613@imperial.ac.uk
Supervisor: Dr. Yiannis Demiris

***Abstract*— In our society, acute intoxication (drinking too much alcohol) is a common occurrence, as drinking is widely accepted and encouraged. ManDown [1] aims to aid users by preventing becoming too intoxicated, monitoring their intoxication level and advising when to stop drinking. Diagnostics for monitoring this level are investigated in this report, in particular slowed reaction time and quick repetitive movements. Reaction time will be tested by turning a complex test into a game and measuring response speed, and quick actions will be tested by checking passive data for groups of spikes. The current progress and planned implementation of these diagnostics are explored, with other parts of ManDown discussed by other team members.**

I. Introduction

Intoxication through the consumption of alcohol is very common, particularly in Western countries such as in the UK. The idea of pub culture is a clear example of this, as it is considered normal to drink socially in a pub or a club. However, the effects of acute intoxication can be dangerous, sometimes resulting in loss of motor control, passing out, and even alcohol poisoning. Worse than acute intoxication is chronic intoxication, where repeated intoxication has severe effects on both the brain and the rest of the body, with many cancer risks and brain diseases possible [2].

There is a lot of advice available for reducing the negative effects of acute intoxication, such as setting a maximum amount to drink in a night, and using non-alcoholic mixers [3]. These initiatives are in place to try and reduce the health risks and the impact on the National Health Service (NHS), which provides healthcare to UK residents. Ambulances were used to transport almost 3/4 of patients with alcohol poisoning in 2013 [4], and initiatives such as the Booze Bus [3] attempt to reduce the impact of alcohol on the NHS.

ManDown is a mobile application designed not to stop the user from drinking at all, but to advise the user on when to stop drinking to prevent some of the worst effects of acute intoxication [1]. It will use an Android phone and smart watch to track the user's intoxication level, providing alerts with advice to users, and using machine learning to tailor its detection model to each user. Data collection will take two forms; passive sensor data collection, and active data collection, where the user interacts with the app and even plays games to collect the information required. This report investigates user diagnostic collection using the data from the app.

II. Related Services

ManDown provides a personalised service to users by automatically tracking their intoxication level, and adjusting its model for intoxication per user for the best fit possible. Using machine learning and passive data collection is a relatively novel idea for a mobile healthcare application; most other apps require only active inputs, such as time of consumption and number of units. One example of a drink tracking app is *Drinkaware*, which is aimed at users trying to reduce long-term alcohol consumption. This is to reduce calorific intake as well as reducing intoxication itself. ManDown adds other data collection to notify the user before drinking too much, providing more short term health benefits.

Users attempting to track their intoxication level are likely to use a device such as a mobile breathalyser. These exist as add-ons to phones or as separate devices. The devices are generally used for drivers that wish to check that they are under the driving limit before driving, but are generally not accurate, and suffer from the fact that alcohol in the bloodstream has a time delay before being detectable in the breath [5].

Social aspects of ManDown may include the ability to form groups, so that the group can stick together while moving between locations and make sure its users don't get left behind. This idea is similar to social apps such as *GoodSAM* [6]. *GoodSAM* allows users to identify the location of someone who seems to need medical help, so that medical professionals with the app can arrive and try to help. ManDown will notify other members of the group if one member is in need of help.

III. System Design

The Design Report [1] gave an initial design for the entire system architecture of ManDown, found in Figure 1. This design uses an Android smartphone and Android Wear smartwatch to collect data from the user, transmit the data to a server in the cloud, and display the results to the user. Android and Android Wear were selected due to the project team being familiar with the systems and the Java programming language, required for Android applications. It was determined that further sensors would not be used, so that the equipment felt natural to the user, without the need of strapping on extra sensors.

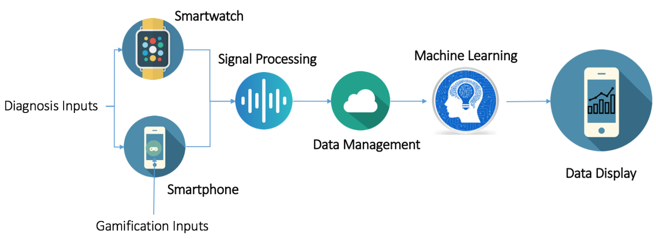

Fig. 1. Overall System Architecture from [1]

To implement the system, each team member was assigned one of the following categories to work on:

- Diagnostic Collection
- Machine Learning
- Data Storage
- User Interface/Experience

I was one of the three team members assigned to the Diagnostic Collection task.

A. Diagnostic Collection

Three team members were assigned to Diagnostic Collection due to the number of diagnostics to investigate. These diagnostics may be determined using either active data, where the user interacts with the app using a control or playing a game; or passive data, where the smartphone and smartwatch collect data from the accelerometer, gyroscope, and magnetometer to determine position and movement. The diagnostics chosen are given in the following list [1], ordered by how useful and easy to implement they are expected to be:

- Reaction Time
- Walking Pattern
- Loss of Motor Control
- Nystagmus
- Scanning Dysarthria
- Quick Repetitive Actions

I will be investigating Reaction Time and Quick Repetitive Actions.

Reaction Time refers to the slowed reaction time experienced by intoxicated people. This is one of the factors that make drink driving so dangerous, as the driver is unable to react to stimuli in time, such as stopping correctly for traffic lights or avoiding a person who has just stepped into the road.

Quick repetitive actions refer to the idea that an intoxicated person has fewer inhibitions, and is likely to move more energetically and erratically than a sober person. This is generally referred to as reduced inhibition, or disinhibition, and impaired judgment.

IV. Research

A. Diagnostics

To be able to successfully diagnose the intoxication level of the app user, the symptoms of intoxication were considered to see which could be measured using passive sensor collection or active app interaction. Some symptoms were considered too difficult to measure using the hardware on hand, such as actively measuring the Blood Alcohol Concentration (BAC); other symptoms discarded include nausea and vomiting, and increased heart rate [7], [2], due to the difficulty of measurement with the equipment available.

As stated in Section III-A, the diagnostics I am particularly considering are Reaction Time and Quick Repetitive Actions. The use of machine learning is not considered due to the novelty of the idea; a search online turned up no material relating machine learning to intoxication level.

1) Reaction Time: Reaction Time is used here to refer to visual reaction time, or the time it takes for a user to respond to a visual stimulus. Although other forms of reaction time, such as reacting to touch or sound, are useful for detecting intoxication, these are more difficult to include in ManDown. Using reaction time to detect intoxication is a challenge as there are different methods of measuring visual reaction time, and the reaction time of each person varies, as does the amount of alcohol required to become intoxicated. Furthermore, most methods of measuring reaction time give better scores with practice as the user becomes more used to them.

Alcohol intoxication generally increases reaction time, slowing the response of participants to a visual stimulus [8], [9], [10]; participants would react to a stimulus, such as a light flashing on, and the mean or median time of reaction were measured to find this result.

Maylor and her team released several papers, finding that mean reaction time increased with alcohol intake [11], [12], [13], but each study had a different focus. The effect of the alcohol depended both on the amount ingested and the task required [11], and the complexity of the task also affected the change in reaction time [13].

The effect of complexity on reaction time is shown to be a factor in other studies. Tzambozis [8] found that complex tasks gave an 18% increase in reaction time, whereas simple only gave a 9% increase, for the same BAC level of 0.05%. One method of studying a more complex task of reaction time is given by Jääskeläinen [14], whereby the user reacted to a digit being flashed up depending on whether it was odd or even. Therefore, a more complex task would give a better indication of intoxication level, and would also be more suitable for gamification.

2) Quick Repetitive Actions: Quick Repetitive Actions is a loose term, referring to more erratic gestures and jerky movements that can come with the loss of inhibition. This behaviour changes a lot from person to person [15], which is why the diagnostic is considered difficult to implement and containing low information content. The difficulty of implementation is a result of observing data over a long period of time, removing noise of everyday activities, and finding a pattern that indicates intoxication. In those people it does affect, quick repetitive actions are a result of lowered inhibitions and increasingly poor judgment [2]. This poor judgment is considered to be due to an increase of dopamine and the inhibition of GABAergic interneurons, increasing impulsiveness [16].

B. Android App Development

As other team members are considering the implementation of the overall application and the user experience components, research on app development as a whole is not required. Only the parts specific to passive sensor collection were considered. To be able to take sensor data, Android's SensorManager must be used; moreover, to be able to run a process in the background, a Service must be used [17]. This allows for accessing and storing sensor data, even while the app has been closed, assuming the user has given permission for this to happen.

C. Gamification

Gamification is the process of converting some data collection tool into a game to encourage the user to provide that data. In the case of ManDown, the games could be used to provide such data as reaction time and how well the user is retaining focus. The effectiveness of gamification depends on how it is used, with both positive and negative effects possible [18]. As a full investigation is part of another team member's remit, they will attempt to make games that have positive effects, with the only overlapping subject area being the data taken from the game, which is discussed later in the report.

V. Component Design

A. Android Application

Android applications are generally split into two main tasks; activities, which the user interacts with directly, or services, which perform background tasks requiring no user interaction [17]. ManDown has elements of both, which will require activities and services to interact with one another. It is planned that there will be two activities for presenting information to the user and allowing the user to play games, and a service for reading sensor data from both the smart phone and the smart watch, and transmitting this and any game data to the cloud. This is represented in Figure 2.

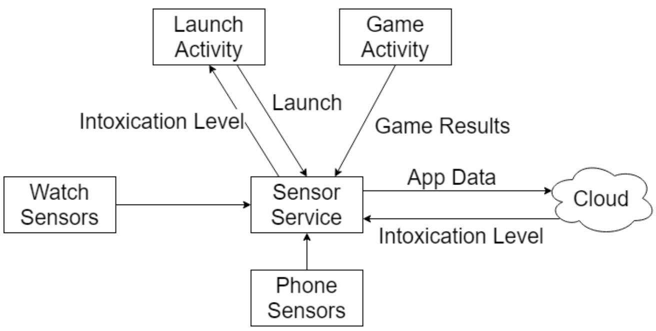

Fig. 2. Android Application Layout

The user interacts with the Launch Activity, the Game Activity, and another Activity running on the smart watch, not shown in Figure 2.

My efforts will be concentrated in the Sensor Service, which must use Android's SensorManager to collect sensor data from both the Water Sensors and Phone Sensors, collate and process the information, then transmit it to a server in the Cloud.

B. Detecting Diagnostics

1) Active App Interaction: Use of games within the application encourages users to interact more fully with the app, by entertaining the user. Furthermore, games naturally keep a score of how well the user is doing, which can be used to tell how the user has changed over the course of an evening of drinking.

One game which is particularly promising for detecting slowed reaction times is a similar game to Whack-a-mole, written by another team member. The scores of the user are expected to increase as they play the game and get naturally better at it, but as the user drinks, the scores may decrease due to slowed reaction time. In addition to the scoring of the game, the app can measure the time taken from the appearance of an object to the user tapping it, which is a complex test of the reaction time of the user. This time is expected to stay mostly constant as the user learns the game, and decrease with intoxication, giving a clear indication of intoxication level. These reaction times will be stored with the game scores and transmitted to the server in the Cloud for processing.

2) Passive Data: Detecting diagnostics from the raw sensor data is the next step in the application. The Sensor Service will retrieve readings from all sensors and after a suitable number of readings, attempt to process the data. The result of this process will be stored in a database table, to be transmitted with the sensor data.

As quick repetitive actions are long-term actions in the context of sensor, the processing can average several frames together, or simply reduce the sample rate. This reduction of sample rate is preferable as battery consumption from constant sensor readings is likely to be high; reducing the sample rate removes the need for an averaging operation, and lowers the amount of data to be stored and transmitted to the Cloud.

An early estimate for sample rate is around 50ms, as this is likely to detect sharp motions without taking a large number of readings. However, even this is a large amount of data to be taking constantly over the course of several hours, the likely use time of the application. Hence, the application will take data for several minutes, then wait a longer period before taking another sample.

This data will be processed after many sensor readings to detect for jolting movements, indicative of quick repetitive movements, by considering the size and frequency of spikes in the data. These results will be stored locally in the same database as the sensor data to be transmitted to the Cloud.

C. Testing Phase

Henceforth, the frequency of sensor polling will be referred to as the sample frequency, and the inverse of the time between the start of each sample set will be referred to as the sample rate.

Initial estimates for the sample frequency and rate are 50ms and five minutes respectively. These are simply educated guesses; to be able to determine the most effective compromise between sample rate and battery consumption, the testing phase will collect a much larger data set, sampling constantly at around 10ms. This large data set will then be processed to determine the minimum frequency and sample rate for effective diagnostics.

During the collection of the data set, some users will be encouraged to use the app, depending on which experimental group they are a part of [1]. The users that play the game will provide reaction time data which can be correlated with the breath alcohol content to determine intoxication level.

D. Current Progress

At this stage, an Android activity has been written to poll for accelerometer sensor data. This must be extended to include other sensor data, as well as converted into an Android Service, including a launcher to launch the service. Furthermore, the data is not yet stored in any way, as this will require the database system to be well defined. Once the database system is defined, and the transmission protocol has been selected, data storage and transmission will begin.

Apart from these tasks, the data from the sensors must be analysed. This requires first that the application can take all sensor data; the gyroscope and magnetometer must be polled. The raw data can then be collated and processed.

An additional activity has been written with the Whack-a-mole style game, called Whack-a-beer. This must be integrated with the rest of the application, and have the total scores and reaction times recorded and transmitted with the sensor data.

VI. Conclusion

Alcohol consumption is a serious issue in the United Kingdom, with a large expenditure of public funds going on healthcare and ambulance services. Drinking too much at once leads to acute intoxication, and depending on the amount consumed can lead to loss of control, passing out, and poisoning. ManDown is a mobile healthcare app that aims to help users stop short of becoming too intoxicated by using sensors and games to collect data on the current intoxication level, and advising when to stop drinking.

Research shows that reaction time increases with alcohol, especially for complex tasks, which makes it an ideal diagnostic. A game will be made to measure reaction time for a more complex task to help identify intoxication level. Quick Repetitive Actions, resulting from a lowering of inhibition and inferior decision making from alcohol, may also provide information about intoxication level using only passive sensor data from a smart phone and smart watch.

Currently, two Android applications have been written, containing a game and an example of retrieving sensor data. These must be integrated together, then extended to store and transmit data to a server in the Cloud to identify intoxication level. This may then be displayed to the user. A testing phase will determine the most useful information and the optimal sample frequency and rate by comparing app data to a breathalyser result from study participants.

References

[1] D. Alyasiri, M. Hart, D. Ma, M. Poynton, S. Rubio, A. Shafiei, and W. Cong Te, "Mandown - a social drinking management app," 2017.

[2] Alcohol Advisory Council of New Zealand, "Alcohol - the body & health effects," [Online; accessed February 20, 2017]. [Online]. Available: www.hpa.org.nz/sites/default/files/documents/HealthEffects.pdf

[3] London Ambulance Service, "Alcohol-related 999 incidents," [Online; accessed February 18, 2017]. [Online]. Available: http://www.londonambulance.nhs.uk/news/alcohol-related_999_incidents.aspx

[4] Nuffield Trust, "The sobering burden of alcohol on the nhs," [Online; accessed February 18, 2017]. [Online]. Available: https://www.nuffieldtrust.org.uk/news-item/the-sobering-burden-of-alcohol-on-the-nhs

[5] M. Fransson, A. W. Jones, and L. Andersson, "Laboratory evaluation of a new evidential breath-alcohol analyser designed for mobile testing–the evidenzer," *Medicine, science and the law*, vol. 45, no. 1, pp. 61–70, 2005.

[6] GoodSAM, "Goodsam website," [Online; accessed February 22, 2017]. [Online]. Available: https://www.goodsamapp.org/

[7] L. Vonghia, L. Leggio, A. Ferrulli, M. Bertini, G. Gasbarrini, G. Addolorato, A. T. S. Group *et al.*, "Acute alcohol intoxication," *European Journal of Internal Medicine*, vol. 19, no. 8, pp. 561–567, 2008.

[8] K. Tzambazis and C. Stough, "Alcohol impairs speed of information processing and simple and choice reaction time and differentially impairs higher-order cognitive abilities," *Alcohol and Alcoholism*, vol. 35, no. 2, pp. 197–201, 2000.

[9] R. Gustafson, "Effect of moderate doses of alcohol on simple auditory reaction time in a vigilance setting," *Perceptual and Motor Skills*, vol. 62, no. 3, pp. 683–690, 1986.

[10] D. Sutton and J. Kimm, "Alcohol effects on human motor unit reaction time," *Physiology & behavior*, vol. 5, no. 8, pp. 889–892, 1970.

[11] E. A. Maylor, P. Rabbitt, A. Sahgal, and C. Wright, "Effects of alcohol on speed and accuracy in choice reaction time and visual search," *Acta psychologica*, vol. 65, no. 2, pp. 147–163, 1987.

[12] E. A. Maylor, P. M. Rabbitt, G. James, and S. Kerr, "Effects of alcohol and extended practice on divided-attention performance," *Attention, Perception, & Psychophysics*, vol. 48, no. 5, pp. 445–452, 1990.

[13] E. A. Maylor, P. Rabbitt, G. James, and S. Kerr, "Effects of alcohol, practice, and task complexity on reaction time distributions," *The Quarterly Journal of Experimental Psychology*, vol. 44, no. 1, pp. 119–139, 1992.

[14] I. P. Jääskeläinen, K. Alho, C. Escera, I. Winkler, P. Sillanaukee, and R. Näätänen, "Effects of ethanol and auditory distraction on forced choice reaction time," *Alcohol*, vol. 13, no. 2, pp. 153–156, 1996.

[15] M. T. Fillmore and M. Vogel-Sprott, "Response inhibition under alcohol: effects of cognitive and motivational conflict." *Journal of studies on alcohol*, vol. 61, no. 2, pp. 239–246, 2000.

[16] I. Boileau, J.-M. Assaad, R. O. Pihl, C. Benkelfat, M. Leyton, M. Diksic, R. E. Tremblay, and A. Dagher, "Alcohol promotes dopamine release in the human nucleus accumbens," *Synapse*, vol. 49, no. 4, pp. 226–231, 2003.

[17] Developers - Android, "App components," [Online; accessed February 20, 2017]. [Online]. Available: https://developer.android.com/guide/components/index.html

[18] J. Hamari, J. Koivisto, and H. Sarsa, "Does gamification work?– a literature review of empirical studies on gamification," in *System Sciences (HICSS), 2014 47th Hawaii International Conference on*. IEEE, 2014, pp. 3025–3034.