**DRUNK TEST**

**A Mini-Project Report**

**Under**

**Implementation of Technology**

***Submitted by***

**Pranav Joshi B046**

**Vedant Sharma B044**

**Pragti Kapoor B049**

**Abinav Jaiswal B043**

***in partial fulfillment for the award of the degree***

***of***

**B.Tech.**

**IN**

**Computer Science**

**At**



**Mumbai,**

**April 2015**

**CERTIFICATE**

This is to certify that the project entitled **DRUNK TEST** is the bonafide work carried out by **Pragti Kapoor, Vedant Sharma, Pranav Joshi, Abhinav Jaiswal** studying B.Tech (Computer Engineering), MPSTME (NMIMS), Mumbai, during the fourth semester of the academic year 2014-2015, in fulfilment of the requirements for the award of the Degree of Bechelors of Technology as per the norms prescribed by NMIMS. The project work has been assessed and found to be satisfactory.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Examiner 1 Examiner 2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dean

Dr. S.Y. Mhaiskar

**DECLARATION**

We, Pranav Joshi, Pragti Kapoor, Vedant Sharma, Abhinav Jaiswal B.Tech (Computer Engineering), semester- IV, understand that plagiarism is defined as anyone or combination of the following:

1. Un-credited verbatim copying of individual sentences, paragraphs or illustration (such as graphs, diagrams, etc.) from any source, published or unpublished, including the internet.

2. Un-credited improper paraphrasing of pages paragraphs (changing a few words phrases, or rearranging the original sentence order)

3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did wrote what.

4. We have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of our work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.

5. We affirm that no portion of my work can be considered as plagiarism and we take full responsibility if such a complaint occurs. We understand fully well that the guide of the seminar/ seminar report may not be in a position to check for the possibility of such incidences of plagiarism in this body of work.

Signature of the Students:

\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

Name: \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

Roll No. \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

Place: Mumbai

Date: April 2015

**Acknowledgement**

*We would like to thank our faculty mentor, Prof. Poonam Gupta for her guidance and assistance throughout the project. We would also like to thank Mr. Aftab Sheikh for providing us valuable insights into various aspects of Android,,XML programming. His helpful gestures were crucial in implementing core JAVA programming concepts to explore practical utilities.*

*Lastly we would also like to mention the contribution by our seniors who gave us tips which came in handy during the selection of our core ideas.*

Table of Contents

Sr. No. Topic Page No.

List of Figures i

Abbreviation ii

Abstract iii

1. Introduction 1

1.1 Problem Specification 1

1.2 Solution Outline 2

1.3 Application/Usage 2

2. System Analysis 3

2.1 Existing System (If any) 3

2.2 Software Requirement 4

2.3 Hardware Requirement 5

3. Design and Methodology 10

3.1 Flowchart of App 13

3.2 Output (Screenshot of the app) 15

3.3 Limitation of App 25

4. Conclusion and Future Scope 26

5. References 27

i

List of Figures

Name Page No.

Programming platform screenshots *9*

Flowchart *10*

Application output screenshots *11*

Peripherals *none*

ii

ABBREVIATIONS

Name Page No.

**SDK- Software Development Kit 3**

**JDK- Java Development Kit 3**

**API- Application Program Interface 4**

**JRE- Java Runtime Environment 4**

**AVD – Android Virtual Device 5**

**IDE – Integrated Development Environment 4**

**GUI – Graphical User Interface 7**

**XML – Extensible Markup Language 9**

**ADT – Android Developer Tools 9**

**iii**

**Abstract**

*This project is aimed at implementing existing technologies and create an application by pursuing self-learning.*

*The knowledge of JAVA programming, Android Studio, SDK manager, Eclipse, etc is learnt by assimilating information from various sources such as the Internet, books, journals, etc.*

*An attitude of accumulating knowledge by self-learning and implementing all possible resources is cultivated.*

*By choosing a particular idea, a group oriented approach is followed. Various group members have contributed different aspects of application development. The efforts of every individual of the team are combined and the final product is tested across various platforms.*

1

INTRODUCTION

* 1. **Problem Specification**

As consumption power increases and incomes rise, the issue of alcohol consumption bothers any civil society. The dangers of alcohol overconsumption are well known. Individuals consume more volumes than their body can bear, especially those who are binge drinkers. Hence wrong decisions are made when a person is under the influence resulting in a confrontation with the police. The drinker has to face a breath analyser test at the hands of the authorities. The dangers of drunk driving are life threatening. Hence a statement of caution and a comprehensive screening method to test drinking is lacking.

A daily increase in accidents and inconveniences resulting due to a person not willing to accept that he is drunk needs a cohesive and self-validating parameter which can never be inaccurate. To decrease the load of the traffic police and to reduce law breaking instances, a concrete solution is required.

**2**

* 1. **Solution Outline**

Rather than making the drinker face a breath analyzer test at the hands of the authorities, we have come up with a uniform reflex cum mental ability test which can be generalized to an extent where any person who in an inebriated state can take it. It involves a sequential structure which is navigationally simple to undertake. The application is capable of conducting its own test of differentiating between a drunk and a sober individual. This can help reduce improper conduct and decisions taken when in unsound mind.

Decrease in unwanted and uncomfortable situations for drinkers and those

around them can ensure a pleasant environment for all.

**1.3 Application/Usage**

The Drunk Test application can be utilized in many real world scenarios.

It can be given to all bartenders in pubs to assess their customers’ state. If a

person is found to be sloshed he can be directed to call for a cab to take him to

any destination he so pleases. This can reduce possible tussles with the

authorities and ensure a smooth experience for the inebriated individual.

Hence it can combat excessive usage of alcohol or at the very least alert

concerned people so that uninformed and rash decisions aren’t taken.

Further application can range from studying reflexes of those who have a high

alcohol content so as to facilitate neuro-cognitive scientific research.

3

2. System Analysis

This involves a complete assessment of inventory required and used for the development and successful implementation of the Android application technology. All software and hardware specifications are elaborated upon so as to ensure recreation of the environment used and the procedures followed for execution.

2.1 Existing System (If any)

The existing environment consists of JDK and JRE software which are already assumed to be pre-installed and pre-configured. The data file paths and resource allocations are calibrated and set so as to include an Android application supporting environment.

The Windows OS is used which belongs to the version 8.1. For Android emulators, coding software, importing of subroutines, etc. proper infrastructure should be existing in the system. Java Environment, if not existing, has to be set up. Resource allocation has to be judiciously done.

Any discrepancies in the system, if present, should be configured so as to facilitate the installation and operation of peripherals required for android application development.

A detailed examination of core programming implementation has to be conducted so as to ensure the capability of the existing system.

4

2.2 **Software Requirement**

ADT (Android Developer Tools) is a plugin for Eclipse that provides a suite of tools that are integrated with the Eclipse IDE. It offers you access to many features that help you develop Android applications. ADT provides GUI access to many of the command line SDK tools as well as a UI design tool for rapid prototyping, designing, and building of your application's user interface.

If there is usage of Eclipse with ADT, awareness is required that Android Studio is the official IDE for Android. But it can also be done using eclipse

In addition to Eclipse's standard editor features, ADT provision of custom XML editors to help create and edit Android manifests, resources, menus, and layouts in a form-based or graphical mode has to be installed . Double-clicking on an XML file in Eclipse's package explorer opens the appropriate XML editor.

In addition to the normal code editing features of Eclipse, ADT provides enhancements to the Android development experience that allow you to quickly jump to declarations of various types of resources such as strings or layout files. You can access these enhancements by holding down the control key and clicking on the following items:

A resource identifier, such as R.id.button1, jumps to the XML definition of the view.

A declaration in the R.java file, such as public static final int Button01=0x7f050000", jumps to the corresponding XML definition.

An activity or service definition in your manifest, such as <activity android:name=".TestActivity">, jumps to the corresponding Java class. You can jump from an activity definition (or service definition) into the corresponding Java class.

You can jump to any value definition (e.g. @string:foo), regardless of which XML file "foo" is defined in.

Any file-based declaration, such as @layout/bar, opens the file.

5

Non-XML resources, such as @drawable/icon, launches Eclipse's default application for the given file type, which in this case is an image.

@android namespace resources opens the resources found in the SDK install area.

Custom views in XML layouts, such as <foo.bar.MyView></foo.bar.MyView>, or <view class="foo.bar.MyView">) jump to the corresponding custom view classes.

An XML attribute such as @android:string/ok or android.R.string.id in Java code opens the file that declares the strings. The XML tab opens when doing this, not the form-based editor.

ADT provides many features to allow you to design and build your application's user interface. Many of these features are in the graphical layout editor, which you can access by opening one of your application's XML layout files in Eclipse.

The graphical layout editor is the main screen that you use to visually design and build your UI. It is split up into the following parts:

The Android SDK is composed of modular packages that you can download separately using the Android SDK Manager. For example, when the SDK Tools are updated or a new version of the Android platform is released, you can use the SDK Manager to quickly download them to your environment. Simply follow the procedures described in Adding Platforms and Packages.

There are several different packages available for the Android SDK. The table below describes most of the available packages and where they're located once you download them.

Available Packages:

SDK Tools

Contains tools for debugging and testing, plus other utilities that are required to develop an app. If you've just installed the SDK starter package, then you already have the latest version of this package. Make sure you keep this up to date.

SDK Platform-tools

Contains platform-dependent tools for developing and debugging your application. These tools support the latest features of the Android platform and are typically updated only when a new platform becomes available. These tools are always backward compatible with older platforms, but you must be sure that you have the latest version of these tools when you install a new SDK platform.

6

Documentation

An offline copy of the latest documentation for the Android platform APIs.

SDK Platform

There's one SDK Platform available for each version of Android. It includes an android.jar file with a fully compliant Android library. In order to build an Android app, you must specify an SDK platform as your build target.

System Images

Each platform version offers one or more different system images (such as for ARM and x86). The Android emulator requires a system image to operate. You should always test your app on the latest version of Android and using the emulator with the latest system image is a good way to do so.

Sources for Android SDK

A copy of the Android platform source code that's useful for stepping through the code while debugging your app.

Samples for SDK

A collection of sample apps that demonstrate a variety of the platform APIs. These are a great resource to browse Android app code. The API Demos app in particular provides a huge number of small demos you should explore.

Google APIs

An SDK add-on that provides both a platform you can use to develop an app using special Google APIs and a system image for the emulator so you can test your app using the Google APIs.

Android Support

A static library you can include in your app sources in order to use powerful APIs that aren't available in the standard platform. For example, the support library contains versions of the Fragment class that's compatible with Android 1.6 and higher (the class was originally introduced in Android 3.0) and the ViewPager APIs that allow you to easily build a side-swipeable UI.

Google Play Billing

Provides the static libraries and samples that allow you to integrate billing services in your app with Google Play.

7

Google Play Licensing

Provides the static libraries and samples that allow you to perform license verification for your app when distributing with Google Play.

Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. By means of various plug-ins, Eclipse may also be used to develop applications in other programming languages: Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Lua, Natural, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM VisualAge.[2] The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

Released under the terms of the Eclipse Public License, Eclipse SDK is free and open source software (although it is incompatible with the GNU General Public License). It was one of the first IDEs to run under GNU Classpath and it runs without problems under IcedTea.

All the software mentioned above can be downloaded from the internet. There are websites which are maintained by publishers themselves and indigenous sourcs can also be pursued.

8

2.3 **Hardware Requirement**

Windows:

Microsoft® Windows® 8/7/Vista/2003 (32 or 64-bit)

2 GB RAM minimum, 4 GB RAM recommended

400 MB hard disk space

At least 1 GB for Android SDK, emulator system images, and caches

1280 x 800 minimum screen resolution

Java Development Kit (JDK) 7

Optional for accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

FOR ECLIPSE

Recommend 2GB memory for IDE and 2GB if running server locally

Note: WebLogic Server defaults to a large heap, which may be inappropriate for development. Changing the server start script can easily remedy the server memory usage.

Component Support at a glance

Eclipse Version

Eclipse IDE for Java EE Developers 4.3 (Luna), 4.3.1 (Luna SR1), and 4.3.2 (Luna SR2)

Java Version for IDE

Sun JDK 1.8

Application Server

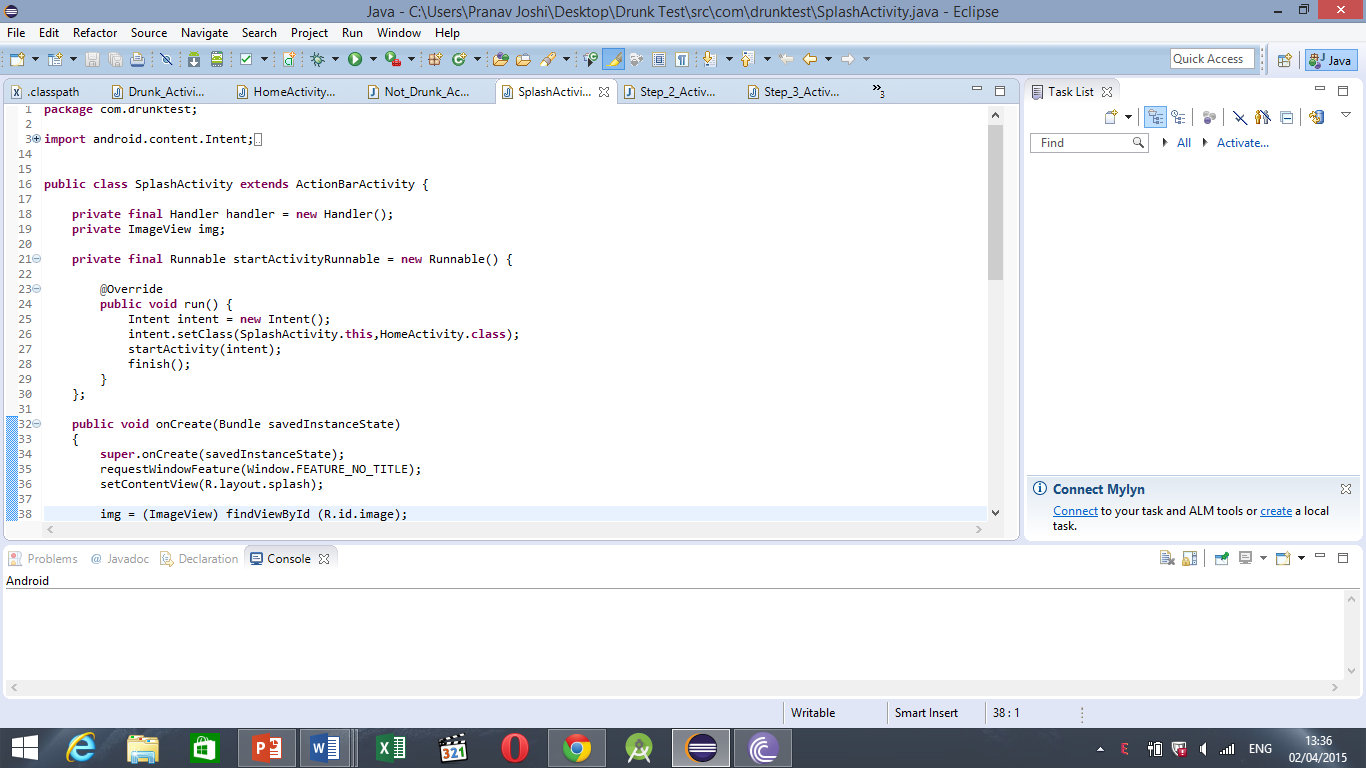
Oracle WebLogic Server® 12.1.x

Oracle WebLogic Server® 10.3.x

9

3. Design and Methodology

This phase of the application is most conceptual and important. Using available knowledge of JAVA programming concepts, ADTs , importing various classes and files and creating a concrete roadmap for complete realization of the idea is all what this part is about.



Coding for various functions such as page turning, input/output, data storage, GUI, transition, animation, etc is done on the eclipse platform by importing android sdk resources.

Coding is done in the language JAVA, interface is designed in XML.

In Java, all functions and subroutines of the applications, including timers, bubbles, MCQs, etc. are written as core programs. They are then linked with the overall application classes and resources. The application is later built and run in an android emulator.

10

3.1 Flowchart of App



START

STOPP

DIALER

0

YOU ARE DRUNK

APP IS OPENED

FADE-OUT PICTURE

STOPP

YOU ARE NOT DRUNK

F

T

SUCCESS

F

T

COLOR TEST

MCQ

SUCCESS

BUBBLE TEST(5 SEC)

BURST ALL BUBBLES

T

F

SUCCESS

REFLEX TEST

(2 SEC)

F

T

ALL ANS CORRECT

F

T

MCQ 1 (MATHS)

YES?

ARE YOU DRUNK?

F

VARIABLE QUESTION

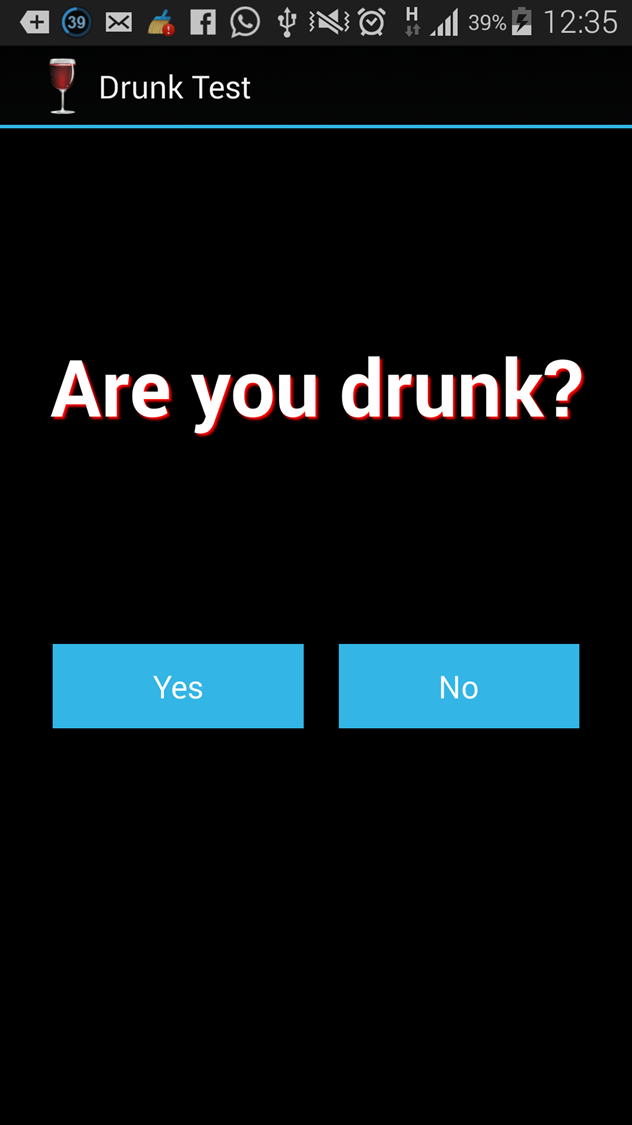
11

**SCREENSHOTS**



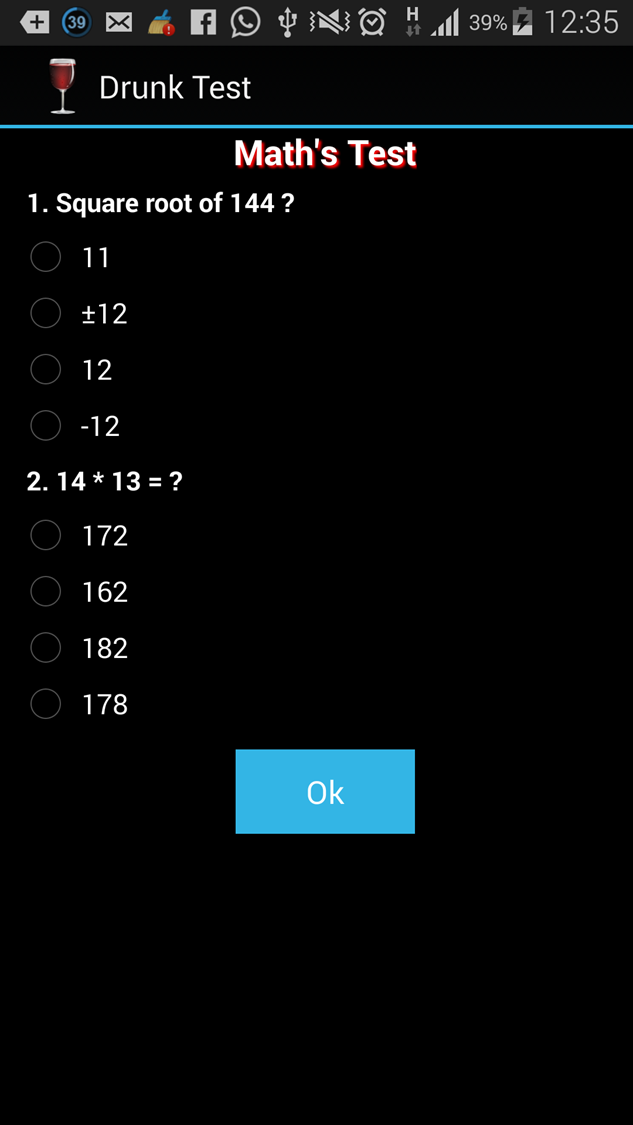
The sepia image is sourced from the web and a fade-in fade out feature is added in Image view .This improves the aesthetic appeal of the application.

12



Screen one  
ACTIVITY 1.Are you Drunk?   
(YES\NO)

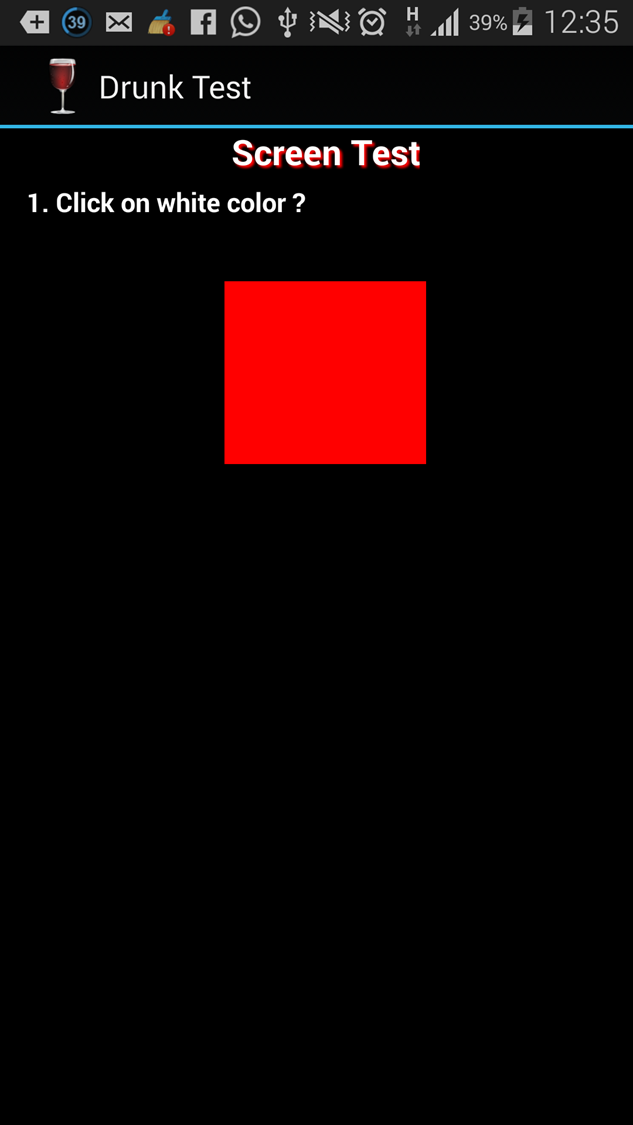
13



Answer both questions correctly.

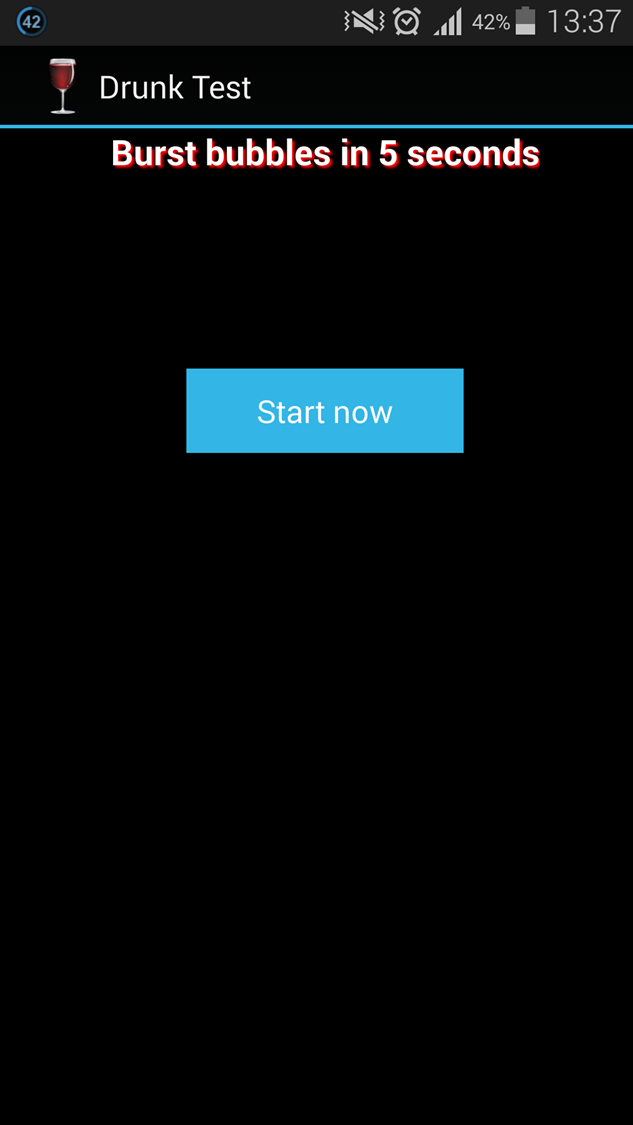
14

Screen 3 and test 2  
Activity 3, Testing color   
Identification and   
reflexes.

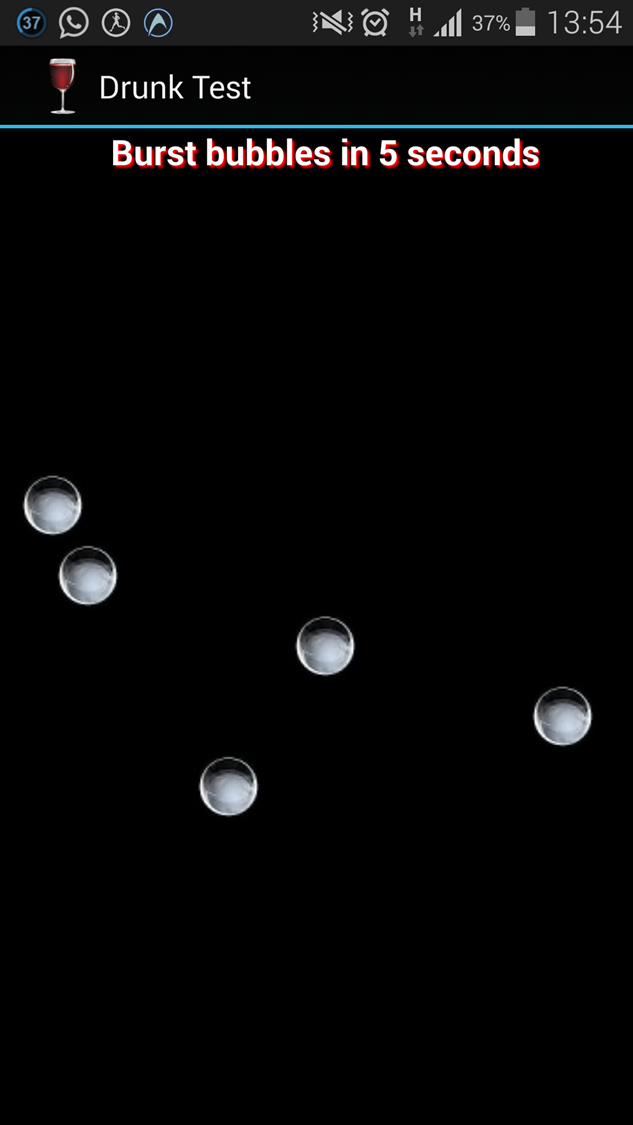


15

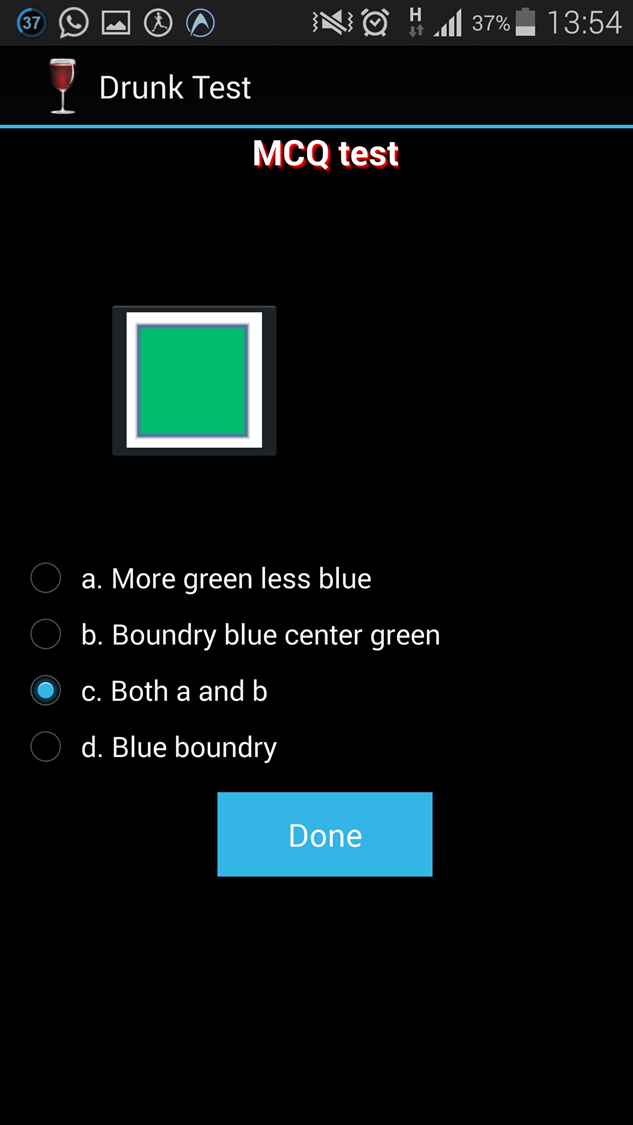
Screen 4, Activity 4,   
test 3. checking the speed Test.  
In 5 seconds , the said   
person is supposed to  
burst 5 bubbles.



16



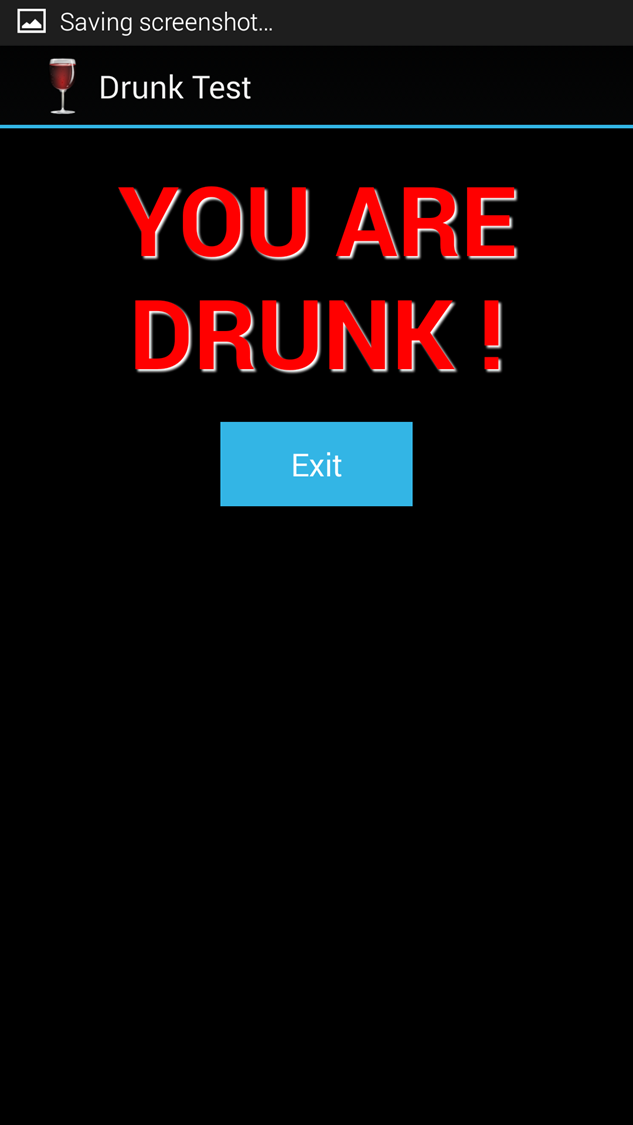
17



Screen 5, Activity 5,  
and test 4.  
IMAGE Visibility and  
colour identification  
test

18

RESULTS PAGE  
Negative Result



19

3.3 Limitations of App

1. The application lacks cloud accessibility. As it is with self-evolving and regular updating features, the application limits its frontiers in both.
2. The animation and graphics are somewhat primitive and do not encompass complex motion tracking software and 3D movements of any kind.
3. The application cannot be used for testing individuals who have exceeded the capacity to digest alcohol and are experiencing serious health hazards. They need to receive medical assistance. This application is only useful for binge drinkers and those who ae inebriated yet in a controllable state.

20

4. **Conclusion and Future Scope**

The application is a viable and concrete test for checking whether a person is in complete control of cognitive and reflexive abilities. It can redirect to the dialer so as to enable calling of any person or even calling for a taxi.

We have hence applied existing technologies and used knowledge of programming languages to create an application which can run in Android based devices ad emulators.

The future scope of this project is quite significant. The application can replace breath analysers to a considerable extent. It can be used by the Traffic Police Department to test whether drivers are drunk or not. It can also be expanded to include self-evolving questions.

21

5. **References**

**Websites:**

1. [*www.androiddevelopers.com*](http://www.androiddevelopers.com)
2. [*www.wikipedia.com*](http://www.wikipedia.com)
3. *www.Eclipse.org*
4. [*www.stackoverflow.com*](http://www.stackoverflow.com)
5. [*www.youtube/androidtechniques.com*](http://www.youtube/androidtechniques.com)

**Books:**

*The Complete Guide to Android Application Development*

* *Oxford University Press, 2009*

*Advanced Java – S.Balasubramanian, 2011*

*Introduction to XML - Tata McGraw Hill, 2011*

*Android SDK for Beginners* ***(e-book)*** *– Softonic Publications, 2014*