**ACCAPTCHA**

Yul P. Puma

CSUN Student  
7360 Balboa Blvd.  
Lake Balboa, CA 91406  
818-267-9590

yul.puma.29

@my.csun.edu

Mohammad Laknahour

CSUN Student

18355 Collins ST.

Tarzana, CA 91356

513-551-9256

Mohammad.laknahoue.

115@my.csun.edu

Kyle Astudillo

CSUN Student  
22518 Criswell St

West Hills, CA 91307  
818-309-8496

kyle.astudillo.965

@my.csun.edu

Beulah Preethi Vallur

CSUN Student  
9500 Zelzah Ave,

Apt D126, CA 91325  
818-818-3052

beulahpreethi.vallur.865

@my.csun.edu

Carlos F. Sandoval

Team Member  
14502 Osborne St  
Panorama City, CA 91402  
818-624-8453

carlos.sandoval.714

@my.csun.edu

Ayrin Golestanian

CSUN Student

451 Hawthorne st. apt 208

Glendale, Ca 91204

818-669-3844

ayrin.golestanian.468

@my.csun.edu

**ABSTRACT**

In this paper, we describe the findings we found when developing a CAPTCHA alternative for elderly mobile users with disabilities.

**Keywords**

none

# **INTRODUCTION**

The problem we faced as a team was how to reinvent reCaptcha, a system designed to establish that a computer user is human, and to protect a websites from bots. The main issue, today, is that the more complex the more difficult it is to use, however, the simpler it is, the easier it is to replicate in a bot.

# Solution

When brainstorming ideas one must think of the demographic of the user. The objective is to develop a CAPTCHA alternative for elderly mobile users with disabilities. Although we were told to focus on one or two groups out of the elderly demographic, ideally, we want our application to be easily accessible to as many people as possible. We decided on a CAPTCHA that, regardless of disability or impairment, a user can interact with, exploiting the unique functionalities that a mobile device has, but also having a structure that is extensible to allow added accessibility for a wider audience. We thought about any people that we may know or imagine who have difficulties using CAPTCHA and ended up with a simple and effective security check that only a human would be able to replicate. Fundamentally, we wanted a CAPTCHA that is secure, easy to interact with, and above all else easy to understand.

Our application, in hindsight, will make use of the TalkBack feature, touch sensing, phone vibrations, and color/lighting. For deaf people, who cannot hear anything, it makes sense that these users will mostly rely on their sense of sight to use their mobile. We have text and visual feedback for them to understand the instructions to complete the CAPTCHA. For blind users who are able to use a screen reader, our app has taken advantage of the Talk Back functionality of Android. This feature allows the blind user to explore the screen with their hands, as it relays back to them audio that describes every part of the screen. Our application is also effective for the elderly demographic who rely on enhancements of visual interfaces. We took into account that this user may use magnification software, as well as requiring an easy-to-understand format. Our approach was in addition to the useful accessibility features for the blind users or deaf users, we implemented a structure that is effortless from which to obtain information, for instance extra large text for them to read, large elements, and a simple interface.

# Features

## Haptic Feedback

As a Team we did not want to rely on just audio and visual feedback alone because although audio is a strong tool it is only useful when you can hear it, so if you are in a loud setting or even sense random white noise in the background it renders audio useless. This also applies to visual feedback if you cannot see the text due to some visual disability. Therefore our team opted for including haptic feedback because there is no opportunity costs to including it along with adding one more way to communicating more information to the user. The way we included haptic feedback is when a button is pressed the button initiates a vibrate to let the user know that they have pressed a button on the ACCAPTCHA interface.

## Audio Feedback

When thinking of ways to improve a CAPTCHA’s usability we did not want to add too much functionality that it would increase its difficulty usage. Instead we decided to incorporate the TalkBack feature many Android phones already possess to assist our features. This way we are not necessarily adding new features that could possibly complicate the user experience. Beyond the simplification of the design, we took into account the integration of our CAPTCHA with the TalkBack feature. The point was to not reinvent the wheel, and use the tools already built into the phone to help in our decision of our overall system to distinguish between humans and bots.

## Visual Feedback

For those who cannot hear our audio feedback we wanted to include everything that was communicated through the audio medium through visuals. The way we applied this is through communicating text that can animate itself in order to communicate to the user if the correct button has been pressed along with making tactical toast messages to the user when certain actions are made.

## Easy Interface

ACCAPTCHA has a easy to use interface. We made use of the full screen of the phone so it can be easy to see and distinguishable. The page has a short phrase with instructions to pass the security check. The instructions will only involve touching one of the four large and highly-visible buttons on the page that take up most of the screen. If the wrong button is clicked, the phone vibrates, a pop-up message tells the user so, and the user is returned to the payment page to try again. The users are giving 3 total tries until they are locked out. When clicking on the button the instruction says to press, the page should only vibrate and allow the user to continue pressing the button until it meets the number of times the instructions told the user. Once the requirements are met the phone should take you to the payment page with the CAPTCHA button turned green and allowing the user to finish their transaction.

**Table 1. Comparison of Original Captcha with ACCAPTCHA**

|  |  |
| --- | --- |
| Original Captcha | Accaptcha |
| Problem:  -Very Complex  -Too much work and  -Hard read  -Buttons are too small  -Not people disability | Improvements:  -Modular Code  -Very Simple to use  -Easy to Read with animation features to zoom in for elderly people to read.  -Vibration and sound notification features to blind people.  --Simple buttons to press to people  -Future we can extend and add to web applications,Iphone APIS,Machine Learning,Haptic/Audio patterns recognition and feedback with Visual features,Color transition.  -Easy to use touch interface  -No typing required  -Enlarged easy to use buttons  -Followed web content accessibility guidelines.  -More User Friendly |

## No Typing Required

In the design of ACCAPTCHA we wanted no typing because that requires the user to make complex actions and although it would make it harder for a bot to bypass the security we as a team thought it was counter productive in what we wanted to accomplish which was to reinvent reCaptcha to be more human accessible without compromising the ability to protect applications from bots

## Following Web Content Accessibility Guidelines

# Reusability

We created ACCAPTCHA as a module to add anywhere. The idea is simply add a button that a user can press and then ACCAPTCHA handles the rest and simply return an easy to read True or False of if the user passed our tests.

# Road Map to the Future

In the Future we would like to make this prototype compatible with web applications and iphone applications. We would like to incorporate machine learning to detect which features the users are using and how successful each one is and how we can incorporate new and better features. Thirdly we would like to incorporate color transitions in order to help with people who have trouble differentiating different colors to help assist them within our system.In addition we would like our application to be more accessible to an even wider audience in the future, and one of our ideas is beat pattern recognition. We can have different modes for different disabilities, such as vibration for deaf-blind people, similar to morse code; audio patterns for blind people., 