**VIEWALITY: Project Proposal and Description**

**Table of contents**

1. Executive summary
2. Competitive analysis
3. Overview and scenarios
4. Initial list of high-level functional requirements
5. List of non-functional requirements
6. System development infrastructure
7. Deliverables

**Executive Summary**

Change Counter calculates the value of your change. With Change Counter, tallying your loose coins is as simple as arranging them on a flat surface then snapping a photo with any device that has access to the web. With the only alternative being visiting a local grocery store and using a machine that imposes a fee of up to 11.9%, Change Counter not only saves a measurable amount of time, but also money.

**Competitive Analysis**

Currently, there are very few similar apps to ours on the market.  When searching “coin counter app” or “change counter app” on Google, most results are educational tools.  Most of these apps are games for children to learn how to count coins or make change.  The two apps we found that are similar to ours and specifically made for counting change are “Scybot Coin Counter” by Scybot Technologies and “Coin Counter” by Smarticle Particle.

Scybot uses a mobile device’s video camera to identify how many coins (circles) are on a flat, well-lit surface and then asks the user to identify what type of coins they are (penny, nickel, etc.) in order to calculate the total.  The main drawback with this app is that the user has to separate the coins by type before capturing the video.  Reviewers on iTunes mentioned that they would rather count it themselves than go through the hassle of using the app.

Coin Counter does not use a mobile device’s camera and is truly more like a calculator.  There are four fields (for quarters, dimes, nickels and pennies) where users enter the quantity of each coin and the app calculates the total by multiplying the number of coins by its denomination.  Currently there are no reviews on the Google Play store, but it is safe to assume that this is almost as cumbersome as counting the coins yourself.

Our app concept seems to be unique in that it will allow the user to simply take a photo of an unorganized group of coins and the total will be calculated.

|  |  |  |  |
| --- | --- | --- | --- |
| ***Key Features*** | **Change Counter** | **Coin Counter**  **by Scybot** | **Coin Counter**  **by Smarticle Particle** |
| *Camera usage* | ***X*** | ***X*** |  |
| *Calculates Total* | ***X*** |  | ***X*** |
| *Eliminates need to sort coins by type* | ***X*** |  |  |
| *Identifies coin by denomination* | ***X*** |  |  |

**Overview and scenarios**

The typical user may find him or herself using Change Counter to calculate the total value of an unorganized pile of coins in order to save time, effort, and money, in lieu of visiting a local storefront to use a coin counting machine that imposes hefty fees or do so entirely by hand.

**Initial List of High-Level Requirements**

1. Have the user be able to take and/or submit a picture of change on a flat surface.
   1. The user will take a photo and submit it through the web where it will be analyzed in order to count the amount of change in the photo.
2. Analyze the photo submitted and compare it to detail photos of coins to calculate a dollar amount of how much change is in the photo.
   1. The analyzer will take the photo the user submitted and compare it against a database of photos of different coins to distinguish each individual coin, and be placed in a group  (ie penny, dime, nickel, etc.).
   2. Once each individual coin has been distinguished separated into the appropriate groups the total amount will be tallied into a dollar amount.
3. Display user the total and a detailed summary of change that has been counted.
   1. The user will be displayed the total dollar amount and a detailed summary of how much of each type of coin is in the photo and the subtotal towards the full total amount.

**List of Non-functional Requirements**

1. To capture adequate coin images for optimal image processing, the following guidelines will be required:
   1. The environment in which the coins are placed must be unobscured, but not excessively illuminated, as to avoid highly glossy, unclear images.
   2. Images must be captured directly above the array of coins.  Photos taken at side angles will hinder the image comparison process from functioning appropriately.
2. A low amount of storage will be needed to keep track of coin types and quantities before and after calculations are completed.
3. There will be no functionality to count the monetary value of any form of paper currency, no matter if it is US or foreign.
4. Depending on the number of coins counted, the processing speed for image comparison and calculation will vary by a miniscule percentage.
5. When using either an Android or iOS device, the most sufficient distance required to acquire photos will range from 3 inches to 17 inches from the face of the coins to the lens of the camera.

**System Development Infrastructure**

1. Front end: HTML5, CSS
2. Backend: MySQL, Javascript
3. Source control: GitHub
4. Framework: Bootstrap
5. Browsers: Chrome, Firefox, IE, Opera, Safari
6. Mobile Application Development: Android Studio, as well as both SDK and XCode for iOS
7. Image Comparison: OpenCV (Computer Vision) API for Windows, Linux, Mac OS, iOS, and Android

**Git Hub Respository URL**

<https://github.com/dtaylo61/CEN4010-Project>

**Task Management Tool**

SomeOne URL - <https://prprogrammers.someone.io/>

**Server URL**

<http://lamp.cse.fau.edu/~dtaylo61/ChangeCounter/>