**Product Requirements**

*CurrenSee*

| **Team** | 9 |
| --- | --- |

# 

# **What is the problem to be solved?**

Around 6% of the U.S. population is at least legally blind. Modern medicine and technology has made it possible to continue a normal life even while unable to see clearly. Thankfully cards have diminished the need to carry cash around, but it still exists and can be troublesome to use if you are blind. Our goal is to contribute ease of life to those who still have cash in their wallet and have a hard time distinguishing between the bills.

The goal of our app, *CurrenSee*, is to be able to take a picture of currency and be able to tell the user what type of bill they have taken a picture of. As well as be able to tell the user how much the total bills equal to if multiple bills are in the picture. The first mentioned feature is to allow the user to mark their bills for future use. The second feature is to check how much cash you have on hand for those in a hurry.

# **System requirements** What system configuration needs to run your proposed system (including anything third party that is needed to run your system).

Our goal is to have the application run on at least iOS. The phone should allow permission for camera access. The phone should have access to WiFi in order to download the app.

# **Users profile**

Who is the system intended for? What characteristics should the users have (this can also be a range of things such as reading level, etc.).

This system is intended for blind, low vision, visually impaired

# **List of Features**

# Provide a numbered list of features (F1, F2, etc.) that concisely, clearly, and accurately describe that which constitutes your project.

1. Detecting a form of currency and capturing a photo.
2. Classifying seven denominations of the US money paper currency. (One, Two, Five, Ten, Twenty, Fifty, One-Hundred)
3. Recognizing various money bills in one photo.
4. Calculating the total value of money bills which is in the photo.
5. Converting the calculated value to any currency.
6. Reading out loud the result.
7. Managing the app by voice.

# **Functional requirements (user stories)**

List the Priority as 1 (High Priority - Critical) to 3 (Low Priority – Would be nice if we have time)

| **No.** | **User Story Name** | **Description** | **Priority** |
| --- | --- | --- | --- |
| R1 | Startup | App should be able to be opened by voice assistant ie. Siri | 1 |
| R2 | Object detection | App should use the camera to detect whether or not a US note/coin is in the frame. | 1 |
| R3 | Currency interpretation | Once currency is recognized, the app should read out which coin/note it is. | 1 |
| R4 | World currencies | App should do R2 & R3 for Major Non-US currencies ie. €, £, ¥ | 3 |
| R5 |  | App should allow User to set ‘home currency’ | 2 |
| R6 | Currency Conversion | If currency that is not ‘home currency’ is detected, the app should let User know the value of the note/coin in ‘home currency’. | 2 |
| R7 | Currency collection | App should be able to recognize currency when there are multiple bills or coins in the frame. | 3 |
| R8 | Currency total | If multiple notes/coins are detected, the app should tell User which notes were detected (from left to right) and the total amount. | 3 |

# **Non-Functional Requirements**

*Describe any constraints or cross-cutting characteristics of the system in a manner that is clear, specific, and testable. Each requirement should have a unique identifier (e.g. NF1, NF2,..). Only present those which are applicable to your system. Categories include but are not limited to:*

Security

Reliability

Usability

Cross-Platform Compatibility

Accuracy

List the Priority as 1 (High Priority - Critical) to 3 (Low Priority – Would be nice if we have time)

| **No.** | **User Story Name** | **Description** | **Priority** |
| --- | --- | --- | --- |
| NF1 | Useability | It is imperative that this app is as accessible as possible due to its nature of being an app addressing a disability. The app should work seamlessly with existing iOS accessibility features. | 1 |
| NF2 | Accuracy | The app’s response to image data from the camera must be accurate as the point of the app is to accurately detect currency and what specific amount it is. | 1 |
| NF3 | Performance | The app should interpret currency in 1-6 seconds, especially if it is to be used in a scenario like shopping in which efficiency will be beneficial to the experience. Likewise, the app should also not be a battery drainer if it is to be used in everyday scenarios. | 2 |
| NF4 | Cross-Platform Compatibility | The app should be accessible to both Apple App Store users and Android Google Play users. | 3 |
| NF5 | Reliability |  |  |
|  |  |  |  |
|  |  |  |  |

Sponsor Requirements

I have read and approve the material in this document. If there is no external sponsor, the TA or instructor will sign it for accuracy/scope.

