# Overview (needed for design meeting)

## Constraints + Criteria

Note only relevant constraints + criteria are noted here (for example, height and weight of items are not included).

|  |  |  |
| --- | --- | --- |
| **Description** | **Design Choice** | **Explanation** |
| Must be >= 80% accurate at identifying/sorting paper, plastic, metal, glass, and “other | Two modes: default modes which guarantees at least 80% shipped accuracy + training mode where user can provide feedback and app suggests software update if available. | This guarantee is 3.2 times that of 25% of plastics which are collected and sent to recycling facilities in Canada. Also provides a ‘feedback’ system for user to understand when the system not up to guaranteed standards and allows for constant improvement of system. |
| Must identify/sort >= 6 items in a minute (every 10s) | Use ZEXMTE CRS4.0 Bluetooth adapter. | The adapters has 3 Mbps download speed which can download a webpage in 3 seconds. We only need one image and a couple of strings, which can be displayed on the app page nearly instantaneously. Therefore, information can be transferred well under every 3 seconds. |
| Must cost <= $250 | Use GitHub, Xcode (with Swift) and app packages. Use ZEXMTE CRS4.0 Bluetooth adapter. | The software platforms allow for free development. The Bluetooth adapter is on the low end of linux compatible Bluetooth adapters (~$13 as compared to ~$20) |
| Must not misidentify glass/Styrofoam/trash into other types more than 10% of the time | Use the same design choice as the first constraint, adapted to 10% of glass/Styrofoam/trash | See explanation for constraint #1 |
| Must meet all applicable Canadian safety standards | Use Swift accesibilty tags + Apple Developer colourblind guidelines | This will allow for an Apple app to comply with all visually impaired guidelines that is standard across all Apple apps. |
| Should be low power <= 50W | Use ZEXMTE CRS4.0 Bluetooth adapter | Bluetooth adapter uses low energy technology |
| Should have long lifespan >= 15Yr | Push current product as version 1.0, allowing for future updates and bug fixes. | A current App is theoretically useable indefinitely. Using a versioning system, users can download updates or bug fixes as they come available. |
| Should be open source | The app and Jetson logic will be publicly available on a GitHub repository. | Git is used for its popularity (the only version control that the whole team is familiar with), because it is free, and its accessibility to the public, and support for non-linear development. The git structure wherein there is a master, development and feature branch allows for a test harness that the code must pass through before being pushed out to the public. Feature branches also allow for parallel development. GitHub is going to be used because it is free for teams, the whole team has GitHub accounts, and it is the current market leader for Git |
| Should be modular | The app and Jetson logic will be publicly available on a GitHub repository. | Even without purchasing the system, users are able to use the logic behind the sorting software and AI for their own use. |

## Other Design Decisions

|  |  |
| --- | --- |
| **Design Decision** | **Reasoning** |
| Colour theme is blue | A team voted on the shade of blue and decided that the chosen base shade (Hex Colour #59BAFF) was the most aesthetic. Blue was chosen because it is associated with recycling |
| Product name is AmIGarbage? | The team voted from a pool of generated ideas for AmIGarbage based on its relevancy to the product and its humour |
| Show percentage certainty associated with each scan | This was based on a team discussion to ensure the user knows that there is some uncertainty with any Image Recognition tool, and the user should be confirming the tool is functioning correctly |
| Prototype with apple app for iPhone SE(for proof of concept), but design encourages Android app to be developed first. Note iPhone SE does not have soft navigation, so the prototype will not include soft navigation | Android systems are far more popular than Apple (with 87% of global market share in 2019). Additionally, the Android developer cost is cheaper (one time $25 fee as compared to yearly $99 fee). However, the prototype is an Apple App because the team has free access to iPhones. |
| Accuracy measurement includes preloaded, shipped accuracy. For example, if shipped accuracy is 95%, then current accuracy is equal to (95+numWrong)/(100+numScanned) | This avoids the edge case wherein any incorrectly scanned item within the first four scans of the system would warrant a software update. For example, we do not want the user to need a software update if they use the system for the first time, and the first scan is incorrect. The system must consider the shipped accuracy. |
| LOGO:Icon  Description automatically generated | Logo was determined by team to accurately depict the system. It involves the universally recognized logo of recycling with the technological addition of our electromechanical system. |
| Bluetooth is the mode of communication between app and controller | The team agreed that users would rather have the freedom to move around with their phone as opposed to being confined to a physical space with a physical connection. It is also a fairly standard practice in modern apps. |

# Team Meeting Review Notes On Initial Design –

* Blue / green theme
* Show % accuracy (aka I am 98% certain the item was paper)
* Make buttons bigger/clearer
* Look into colour blind things
* Look into general accessibility (text to speech)

# Mapping Constraints to Design Decisions Discussions / Research

1. Must be >= 80% accurate at identifying/sorting paper, plastic, metal, glass, and “other”:
   * Initial design (prior to detailed design):
     + Have two modes: training and default. Default mode is shipped with an accuracy of at least 80% (Controls currently is reporting around 95%, so it should be attainable), and have a ‘Training mode’ that incorporates feedback such that the model can adapt to the types of recycling the user has, and the system gets smarter
   * Problems with current design:
     + Controls + SW were discussing training mode, and it typically takes hours to retrain a model. So there are three options:
       - a. retrain the model as a scheduled daily or weekly job at a user designated time
       - b. let the user retrain the model whenever they want
       - c. only retrain the model when it reaches that 80% accuracy threshold
     + In all cases, to maintain the 80% accuracy guarantee, we will have to automatically shut down the system and retrain the model when (num\_wrong +1)/(total\_num) < 0.8
     + A current limitation is that we need hundreds to thousands of photos to retrain the model and that the model (with 2GB board) needs approximately 24 hours to retrain – so we can't retrain often and training probably shouldn't be done on an individual board basis, rather it should be a software update
     + Also, we shouldn't let the user dictate when to retrain, because they would have to know the implications, constraints, and process of retraining a model
     + Say we are at 80% accuracy. We retrain the model. What if the system gets it wrong right away after retraining? Then the system would be below 80%. I think the percentage should be determined *from the last time the model was retrained* instead of determined from all uses ever
     + For default mode, we have no feedback system. It is impossible to guarantee a percentage other than to say the model is 95% accurate, giving a factor of safety of 1.2
   * Updated design to account for issues:
     + Guarantee a **shipped accuracy** of at least 80%, with the higher the better. This guarantee is 3.2 times that of 25% of plastics which are collected and sent to recycling facilities in Canada as compared to the 100% target by 2030 ([Proposed Integrated Management Approach to Plastic Products](https://www.canada.ca/en/environment-climate-change/services/canadian-environmental-protection-act-registry/plastics-proposed-integrated-management-approach.html#toc2)). This is default mode (with no feedback).
     + Provide a training mode in which people can give feedback, and it is saved to the Jetson Nano for future upload and verification. If the accuracy falls below 80%, we merely suggest the user undergo a software update, **if available** (no guarantees of development though).
     + Accuracy is measured considering the shipped accuracy as well (to avoid someone failing constraint on their first scan, for example). So if shipped accuracy is 95%, then the current sample size is 100, and the amount of correct scans is 95.
   * Future design:
     + What is included in the SW update? Is it individuals' photos or general? Individual's seems better, but we would need a custom PCB or a new Jetson with higher CPU
2. Must identify/sort >= 6 items in a minute (every 10s)
   * We are maximally sending a couple of strings and a photo from the Jetson Nano to the app. If we want to make it modular (and to allow for more user freedom), we will use Bluetooth communication to control the system from the app. In terms of UDP vs TCP, we don't need either (the L2CAP and RFCOMM stack that Bluetooth provides should be enough considering the adapter has 3Mbps upload/download) <https://www.addictivetips.com/ubuntu-linux-tips/best-usb-bluetooth-adapters/#:~:text=%20Best%20USB%20Bluetooth%20Adapters%20that%20are%20Linux-compatible,The%20ZEXMTE%20Bluetooth%20adapter%20is%20a...%20More> )
   * With 3 Mbps, the app could theoretically upload an entire webpage in 3 seconds, which is much more than one image and a few strings <https://www.attinternetservice.com/resources/mbps-vs-mbps/>
3. Must cost <= $250
   * The software platforms (Git and Xcode) allow for free development. The Bluetooth adapter is on the low end of linux compatible Bluetooth adapters (~$13 as compared to ~$20) <https://www.addictivetips.com/ubuntu-linux-tips/best-usb-bluetooth-adapters/#:~:text=%20Best%20USB%20Bluetooth%20Adapters%20that%20are%20Linux-compatible,The%20ZEXMTE%20Bluetooth%20adapter%20is%20a...%20More>
4. Must not misidentify glass/Styrofoam/trash into other types more than 10% of the time
   * This calculation will fall under the exact same guidelines as overall accuracy
5. Must meet all applicable Canadian safety standards
   * Must use Swift’s built-in accessibility tags for all elements to allow for visually impaired users
   * Follow the apple developer’s colourblindness guidelines: <https://developer.apple.com/design/human-interface-guidelines/accessibility/overview/color-and-contrast/>
6. Should be low power <= 50W
   * App is not part of this power consumption. With Bluetooth adapter, the Jetson does not draw 50W. ZEXMTE USB 4.0 adapter employs low energy technology to lower energy consumption. See electrical prototyping for more info. <https://www.amazon.ca/dp/B0775YF36R/ref=as_li_ss_tl?keywords=ZEXMTE+Bluetooth+USB+Adapter&language=en_US&sr=8-3&linkCode=gs2&linkId=0844ab885ae485ed10280dabb38c4d3d&tag=techflow00-20>
7. Should have long lifespan >= 15Yr
   * This would likely require development for bug fixes and updates to include soft navigation and better AI. However, technically the app is sufficient enough to last 15 years. Constant development also allows for updates with respect to new phone features (such as soft navigation or other features).
8. Should be open source
   * Git is used for its popularity (the only version control that the whole team is familiar with + it is the most popular version control software in the world), because it is free, and its accessibility to the public, and support for non-linear development. <https://www.softwaretestinghelp.com/version-control-software/>
   * GitHub is going to be used because it is free for teams, the whole team has GitHub accounts, and it is the current market leader for Git
   * <https://www.upwork.com/resources/best-git-repository-google-github-bitbucket>
9. Should be modular
   * Code is published to a public GitHub repository because even without purchasing the system, users are able to use the logic behind the sorting software and AI for their own use.

# Other Design Decision Choices

1. Colour theme is blue
   * A team voted on the shade of blue and decided that the chosen base shade (Hex Colour #59BAFF) was the most aesthetic
   * According to Waste Management Canada, recycling is associated with a blue colour
   * <https://www.co.monterey.ca.us/home/showdocument?id=19702>
2. Product name is AmIGarbage?
   * The team voted from a pool of generated ideas for AmIGarbage based on its relevancy to the product and its humour
3. Show percentage certainty for each scan
   * This was based on a team discussion to ensure the user knows that there is some uncertainty with any Image Recognition tool, and the user should be confirming the tool is functioning correctly.
4. Prototype with Apple app, but design would prioritize Android development over Apple
   * Android dominated the Global market in 2019 with 87% of users using Android. This is why the technical design would priorities Android development over Apple development, because it’s available to more people. <https://www.pcmag.com/news/ios-more-popular-in-japan-and-us-android-dominates-in-china-and-india#:~:text=When%20it%20comes%20to%20the,over%20the%20next%20few%20years>.
   * Protoyping involves development of an Apple App based on the limitation that the team with access to the Jetson Nano only has access to iPhones. For proof of concept, this will suffice.
   * When comparing iOS and Android for developing an app, consider that both Google Play Store and Apple App Store have their publishing apps fees. In the case of Android apps, you will need to pay a one-time registration fee of $25. To publish your app in the Apple marketplace, you need to pay $99 annually. <https://theappsolutions.com/blog/development/ios-vs-android/#:~:text=When%20comparing%20iOS%20and%20Android,need%20to%20pay%20%2499%20annually>.
5. Accuracy measurement includes preloaded, shipped accuracy. For example, if shipped accuracy is 95%, then current accuracy is equal to (95+numWrong)/(100+numScanned)
   * This avoids the edge case wherein any incorrectly scanned item within the first four scans of the system would warrant a software update. For example, we do not want the user to need a software update if they use the system for the first time, and the first scan is incorrect. The system must consider the shipped accuracy.
6. Logo
   * Logo was determined by team to accurately depict the system. It involves the universally recognized logo of recycling with the technological addition of our electromechanical system; it has mechanical, electrical, and controls/software components embedded in the Logo.
7. Bluetooth
   * The team agreed that users would rather have the freedom to move around with their phone as opposed to being confined to a physical space with a physical connection. It is also a fairly standard practice in modern apps.