Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project 2020 - Final Report

Project Title:	**A Persuasive Smartphone App for Improving Recycling Performance**
Student:	**Rebecca C. Hallam**
CID:	**01190898**
Course:	**MEng Electrical & Electronic Engineering (Technical Stream)**
Project Supervisor:	**Dr Thomas Clarke**
Second Marker:	**Dr Christos Bouganis**

Plagiarism Statement

The paper copy of my report has been deemed not necessary by my supervisor.

I affirm that I have submitted electronic copies of my final year project report to both Blackboard and the EEE coursework submission system.

I affirm that both copies of the report are identical.

I affirm that I have provided explicit references for all material in my Final Report which is not authored by me and represented as my own work.

Signed:

Abstract

This project concerns the development of a smartphone app, RecycleHelper, designed to educate and motivate users to improve their recycling behaviour. Emphasis is placed on the various psychological principles behind user interface, user experience and persuasive design. Multiple stages of development and testing were completed, to implement an iteratively improved design that involved the insights gathered from the target audience. Whilst some challenges existed in terms of correct classification of materials, it was observed that the simple act of having location-specific recycling information all in one easily accessible place such as an iPhone app, combined with various behaviour design principles, increased user's likeliness to recycle by an average of 15%. Furthermore, a statistically-significant positive correlation was found between a user's recycling knowledge, and their likeliness to recycle. Finally, continued use of the app was found to have improved a user's recycling knowledge by an average of 31%.

Acknowledgements

I would like to express my gratitude to Dr Thomas Clarke for his advice, support and belief in the concept during his supervision of this project. Without your faith in my idea when I self-proposed this project back in October 2019, RecycleHelper would likely not exist today.

I would also like to thank my boyfriend, Loic, for offering unlimited support and ideas throughout not only this project, but also the duration of my time at Imperial. Without your unwavering belief in me, I would probably not have proposed this project in the first place.

Thanks should also go to Anuja, for your continuous support, friendship and advice, especially during our final year at university. Completing coursework was always much more enjoyable when you were doing the same module as me - Thanks for making this year a great one, and encouraging me when motivation was low.

Next, I wish to express my thanks to Alexandra, Anuja, Charlotte G, Charlotte H, Constanca, Dori, Elaine, Grace, Guy, Hannah, Katie, Karishma, Kelly, Loic, Michele, Nia, Nina, Rachel, Rita, Rebecca, Vaikkun, Wes and Zoe for volunteering your time and patience for my rounds of user testing. You provided invaluable insights and opinions that helped me develop RecycleHelper into a (hopefully) successful app.

Finally, I want to thank my parents, Lindsay and Guy, and my sister, Charlotte, for your knowledge and insights throughout my 18 years of education. Without your sacrifices and support, I would not have achieved anywhere near as much as I have.

Contents

Chapter 1

Introduction

Over the past century, the volume of greenhouse gases that have been emitted into the Earth's atmosphere has increased by a factor of 10 [1]. This has been mostly due to the agriculture, industrial, transport and energy sectors. These emissions are a problem, as they cause an effect known as the greenhouse effect, which has been found to be directly linked to the rising of the Earth's temperature [2], also known as 'climate change'. This in turn has already had "observable effects on the environment" [3], such as "loss of sea ice, accelerated sea level rise and longer, more intense heat waves" [3].

To this effect, climate change has developed into one of the greatest issues that the world faces today. The world must come together and unite as one community to try and mitigate, if not reverse, these effects. Whilst this may seem like a large and rather ambitious goal, when broken down into steps that the individual can take, it starts to seem more manageable. That is, if everyone makes the effort to make small changes in their life to become more sustainable, economies of scale will take effect and cause the overall impact to be much larger.

An example of a small step that can be taken is improving recycling performance, as this, alongside composting and reducing waste altogether, has been shown to reliably cause less greenhouse gas emissions than waste just going to landfill. However, the UK's recycling rate is currently only at 45.2% [4].

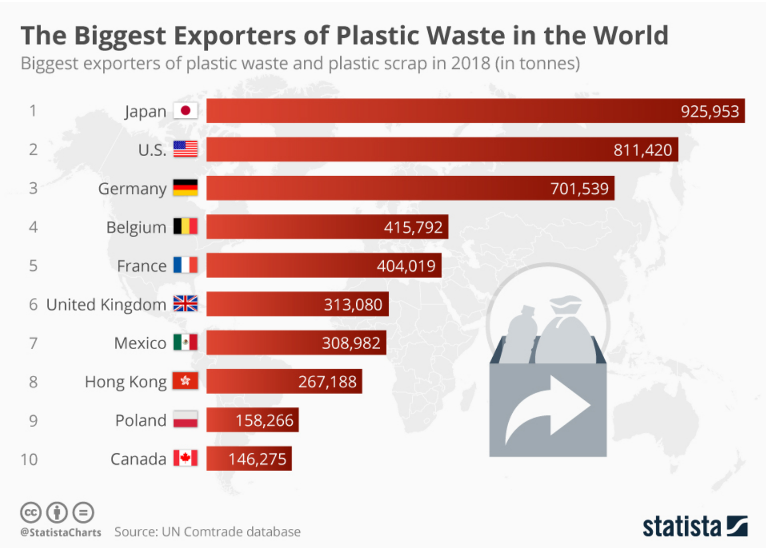


Figure 1.1: The Biggest Waste Exporters in the World [5]

Furthermore, Figure 1.1 shows that, when considering plastic waste alone, the UK is the 6th biggest waste exporter in the world [5]. Therefore, improving the recycling rate in the UK could have a notable positive contribution to reducing world waste. However, there is currently a significant lack of education and motivation surrounding correct recycling.

This project therefore aims to design, build and test a persuasive smartphone app capable of helping users to improve their recycling performance. The app will, at a minimum, allow the user to easily identify how to recycle their household waste, through use of their smartphone camera and an image classification machine learning model. This project will be further specified in Chapter 3.

The aim of this report is to introduce the reader to the relevant areas of background research and document the project development process. The report is structured as follows:

- **Background**: Sets the scene for the project, introducing and reviewing the areas surrounding the project. Existing work related to the project is also reviewed.
- **Project Specification**: Defines the scope and goals of the project. Further elaborates on initial design choices introduced in the Background Chapter and highlights the process and outcomes of Requirement Capture.
- **Analysis and Design**: Provides a high level overview of the design of the application, as well as justifying design choices, such as the various strategies and design principles that were used.
- **Implementation**: Details the process of developing the project deliverable. Due to the iterative nature of this project development strategy, this is split into three stages, each describing their respective designing, building, testing and analysis processes.
- **Evaluation**: Evaluates the outcome of the project and compares it against previously completed work in the same area. Contains a critical analysis of the work completed and evaluates whether the original project objectives were fulfilled.
- **Conclusions and Future Work**: Discusses the success of the project and respective achievements. Highlights any challenges or key findings discovered throughout completion of the project, and introduces areas of future work that could be developed past the project completion date.

Chapter 2

Background

2.1 Introduction

In all projects, it is important to fully understand the background in order to be able to make or understand design choices later on in the project. Furthermore, research allows one to understand what work has already been done in the area, so that any future work undertaken by a project can build upon it, rather than just replicate it.

Therefore, the aim of this background chapter is to provide context to the areas that this project is exploring, by introducing and discussing the relevant background themes. Previous work has been explored and explained relating to these themes. The background themes to be covered in this chapter, and also the relevant questions to be answered, are summarised in Table 2.1.

Theme	Questions
Recycling	What is recycling? How is something defined to be recyclable?
	How successful is recycling in the UK?
	At a high level, what is the process of recycling in the UK?
	How can consumers identify if an item is recyclable?
	What are the benefits of recycling? What are the costs? Is recycling worth it?
	What are the current attitudes towards recycling in the UK?
	What work has been done so far to try and improve the recycling rate?
	What could potentially prevent the recycling rate from being improved?
Platform Development	What are the main platform options for development?
	What are the advantages of each option?
	What are the disadvantages of each option?
	Are there any other factors that may influence the choice of platform?
Behaviour Design	What is behaviour design?
	How does motivation work?
	How can a system be designed to be persuasive?
	What psychological principles can be exploited to influence users?
	What are the potential ethical issues behind persuasive and influential design?
Machine Learning	What is machine learning?
	How does it work? What are the different methods?
	What are the advantages and disadvantages of different methods?
Related Work	What work has already been completed in this area?
	What observations has previous research made?
	What are the pros and cons of the work already completed?

Table 2.1: Project Background

2.2 Recycling

2.2.1 Definition

To understand the area of Recycling and its underlying issues, one must first understand what is meant by the term "recyclable". For the purpose of this project, this is defined as when a item is able to be "collected, sorted, reprocessed and manufactured back into a new product or packaging - at scale and economically" [6, pg. 4].

2.2.2 Recycling in the UK

In 2017, the average household recycling rate in the UK was 45.2% [4]. A further study that year also found that 66% of households were uncertain about what they could place in the recycling bin, and 76% of those surveyed added one or more item to their recycling collection that is not accepted locally [7]. It was then found in 2018 that this rate of incorrect recycling had further increased to 82% [8]. Furthermore, Figure 2.1 shows that, while the recycling rate for the UK may be 45.2%, this is only an average, and therefore there are areas in the UK where the level of performance is even lower. These are symbolised by the areas of shades of blue, and include places like Cornwall and most of London.

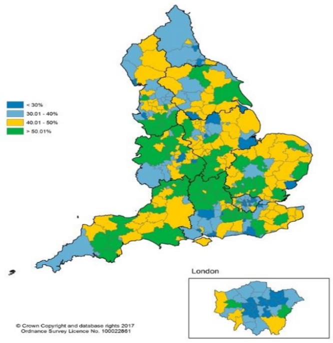


Figure 2.1: Percentage of household waste sent for recycling, preparation for reuse or composting, England, 2016/17 [7, fig. 3.7, pg. 47]

Increasing this rate of recycling has become not only a nationwide focus, but also a global one. This is because the current rates mean that a large proportion of waste ends up in landfill sites every year. Here, waste decomposition causes greenhouse gases to be released into the atmosphere, contributing to global warming. In 2016 alone, 24.4% (52.3 million tonnes) of UK waste was disposed of at landfill sites [4]. Figure 2.2 shows that, whilst certain materials produce more greenhouse gases than others, by recycling as much as possible, a significant percentage of greenhouse gas emissions could be prevented.

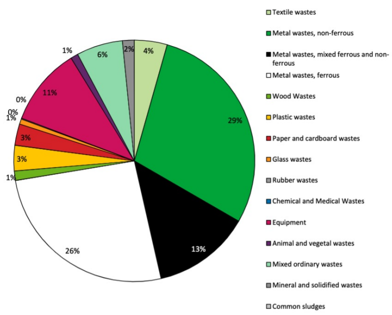


Figure 2.2: Material contribution GHG emissions avoided by recycling, 2014 [7, fig. 8.7, pg. 81]

Furthermore, a study has found that improvement of the UK's waste management would allow for a potential reduction of 4.5 million tonnes of CO_2 released into atmosphere [9]. The annual amount of CO_2 emitted by the UK was measured to be 373.2 million tonnes in 2017 [10, pg. 7]. Whilst this potential 1% overall improvement may seem insignificant, the decrease in emissions from 2016 to 2017 was 12.6 million tonnes [10, pg. 7] - therefore an improvement of 4.5 million tonnes could improve the rate of decrease of CO_2 emissions by 35%. Furthermore, it sets a precedent for further improvement in not only the waste sector, but also other sectors that are key contributors to carbon emissions.

2.2.3 Current Recycling Technology

Recycling methods have developed over the past decade, from initially requiring consumers to separate all materials at kerbside, to collecting mixed materials to be sorted at a central sorting facility [11]. There are now 5 standard types of materials accepted for recycling in the UK [12]; (*i*) Paper, (*ii*) Card, (*iii*) Glass, (*iv*) Metal and (*v*) Plastic - items composed of other materials, such as electronics, textiles and batteries generally need to be recycled using specialist services. Of the standard types, metal "provides one of the highest efficiency rates when recycled, as the quality of the resulting metals is almost as high as that of the initial ones" [12]. This means that it has a near-infinite recycling life cycle, whereas materials such as paper and cardboard can only be recycled a few times before the resulting quality is unusable. This suggests that, in addition to increasing the amount of waste recycled compared to that sent to landfill, a shift to producing less waste in general needs to occur.

Depending on location in the UK, recycling collections are generally one of three types:

1. All dry recyclables.
2. Paper and card collected separately.
3. Glass collected separately.

After collection, recycled materials are then sent to a Mixed Recycling Facility, or MRF, to be sorted further before being sent on to their next destination. MRF facilities therefore need to not only be able to cope with all types of collection individually, but also potentially simultaneously. Figure 2.3 shows that there are 3 main stages of processing at an MRF. The stages of recycling that follow after this depend on the material.

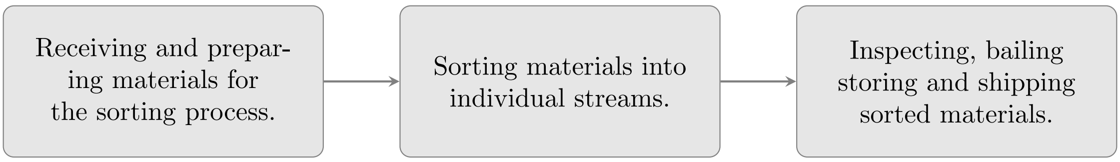


Figure 2.3: Stages at a Materials Recovery Facility [13]

2.2.4 Identification of Recyclable Items

It is all well and good having technology for recycling in place, but if the average consumer does not know whether to recycle their waste or not, then recycling facilities go to waste. Therefore, various labelling schemes have been brought into circulation on most packaging in the UK to inform consumers on what to do with their used packaging. However, there is a lack of standardisation when it comes to this, with different products using different labelling schemes, and some products including no label at all. There are also so many different labels that could be shown on packaging linked to recycling that it is difficult to know and keep track of what they all represent, and what that means about recycling specific to a local area. This section will discuss the most commonly seen labelling schemes in the UK.

2.2.4.1 On Pack Recycling Label

One of the most widely-used labelling system is the On-Pack Recycling Label, or OPRL, scheme. It defines recyclability as "the proportion of local authorities offering recycling services for that material and component" [14] and categorises materials according to this proportion.

(a) **Widely Recycled** 75%+ of councils collect.

(b) **Check Locally** 20-75% of councils collect.

(c) **Not Yet Recycled** <20% of councils collect.

(d) **Widely Recycled at Recycling Points**
>75% of councils do not collect this packaging, but it can be taken to local recycling points.

(e) **Plastic Packaging**
Some plastic packaging can now be recycled at supermarket carrier bag collection points.

(f) **Specialist Labels**
Specific items, such as metal paint cans are accepted for recycling at most local council recycling centres.

Figure 2.4: On Pack Recycling Labels [14]

Whilst these labels allow consumers to easily check how likely it is that a specific item will be accepted, they do not provide concrete information specific to location. This therefore requires the user to further research whether the item in question can be recycled or not. Furthermore, brands and companies must pay a £700 per annum fee [15] to use the labelling scheme on their packaging. Whilst this is a necessary charge to allow OPRL to cover their operating costs and ensure that the labelling scheme remains up to date, and may also be a negligible amount to a lot of companies, it does not exactly motivate companies to provide the recycling information to their consumers.

2.2.4.2 Plastics Labelling

In 2018, plastics production in Europe reached 61.8 million tonnes, with 39.9%, or 24.66 million tonnes of this being produced for packaging alone. However, only 9.4 million tonnes of the plastic post-consumer waste was recycled [16]. As a large group of packaging used in the UK is plastic, companies often make use of the 'Plastics Resin Code'. This is stamped on a significant portion of plastic packaging and explains what type of plastic each item is made of. From there, consumers can refer to the code to understand how best to recycle it.

Figure 2.5: Plastic Resin Identification Codes [17]

Resin Code	Material	Abbreviation	Ease of Recycling
1	Polyethylene Terephthalate	PET	Easy
2	High Density Polyethylene	HDPE	Easy
3	Polyvinyl Chloride	PVC	Difficult
4	Low Density Polyethylene	LDPE	Manageable
5	Polypropylene	PP	Easy
6	Polystyrene	PS	Difficult
7	Other	*n/a*	Very Difficult[1]

Table 2.2: Explanation of Plastic Resin Identification Codes [18]

The issue with this labelling scheme is that the average consumer will probably not bother to learn what these labels mean, and those that do will most likely find it hard to remember which number corresponds to what material and from there remember what this means about how to recycle it. Furthermore, a type of plastic packaging being defined as 'easy' to recycle also does not necessarily mean that it will be accepted for recycling in the consumer's local area.

[1] Generally is not recycled and goes straight to landfill.

2.2.4.3 Other Labels

In addition to the on-pack and plastic resin labelling schemes, there are several other labels linked to recycling and waste, but do not necessarily provide information about how to recycle an item:

(a) **Mobius Loop**
Represents an item that can be recycled.

(b) **Tidyman**
Aims to remind consumers not to litter.

(c) **Waste Electricals**
Electrical items with this symbol can be recycled.

(d) **Recyclable Steel**
The product is made out of recyclable steel.

(e) **Recyclable Aluminium**
The product is made out of recyclable aluminium.

(f) **The Green Dot**
The producer has made a financial contribution to packaging recovery and recycling.

(g) **Glass**
Reminder to recycle glass bottles and jars.

(h) **Seedling**
The registered trademark of European Bioplastics - means that the item is compostable.

(i) **Forest Stewardship Council Logo**
Identifies products which contain wood from well managed forests.

Figure 2.6: Other packaging labels [19]

In fact, it has been found that the presence of these labels further complicates the issue of consumers not being sure how to recycle their waste - there is a common misconception that these labels relate to *how* to recycle an item, when in fact they have other meanings. For example, in a survey, the Tidyman was understood correctly by only 15% of respondents, and the symbol for Recyclable Steel only 12% [20]. This overall lack of information on recycling labels, and a further lack of understanding of the other labels is perhaps a factor causing the UK's low recycling rate, with consumers thinking that they are recycling correctly based on their understanding of the symbols, when in fact they are not.

2.2.5 Benefit vs Cost of Recycling

Whilst recycling is generally seen as positive due to many factors, such as how it diverts waste from landfill sites, it has its own social, economic and environmental impact - each with positive and negative aspects. For example, whilst reducing the waste going to landfill will reduce the amount of greenhouse gases released at landfill sites, the recycling process will have its own carbon footprint, caused by factors such as transportation to recycling centres, as well as the requirements to operate these centres. Therefore, analysing these costs and benefits can shed some light on the real light on the impact of recycling. Examples of these impacts can be seen in Table 2.3.

Area	**Cost or Issue**	**Benefit**
Social	- Recycling is not yet commonly used on a large scale, such as in industry, where the potential positive impact is greater due to economies of scale. - Recycling increases the risk of contamination. For example, buildings in Taiwan made of recycled steel have been found to give residents and visitors gamma radiation poisoning [22]	- Increasing recycling eliminates the need for extra measures to secure government reduction targets [21] - Recycling centres provide jobs. - Encouraging consumers to recycle in turn encourages them to be more conscious about their waste and the environment.

Economic	- Waste going to landfill costs less per tonne than that going to recycling centres. - Separate waste collection has higher costs [21] - Reduced revenue for waste incinerators [21] - Recycling is not always cost-effective [24] - As the amount of recycling increases, investment into new recycling facilities will need to be made.	- Recycling generates revenue to pay for itself, whereas landfill and incineration do not [23] - Damage costs avoided [21] - Increased recycling creates more jobs [21] - Recycling firms generate more profit [21]
Environmental	- Transporting items to be recycled when they could be disposed of locally has an impact - Many materials degrade in quality with repetitive recycling and therefore ultimately end up in landfill. - A vast amount of 'Paper Sludge' produced by recycling paper ends up in landfill or contaminating groundwater [25]. - Recycling plants use a large amount of energy, which contributes to air pollution.	- Prevents waste from ending up in landfill sites or our oceans. - Recycling prevents or at least delays waste from ending up in landfill, thus delaying the impact of climate change. - Eliminates the need for manufacturers to source raw materials for their products [24] which in turn reduces the energy required - Recycling paper reduces the rate of deforestation [24] - Slows the rate of the Earth running out of resources [24] - CO_2 is saved by diverting materials to recycling.

Table 2.3: Impacts of Recycling

From Table 2.3, it can be seen that the recycling industry is complicated. So much so that it is often not clear whether recycling programs "save more than they cost" or even "save more energy than they consume" [26]. Therefore, multiple factors must be taken into account when considering the overall impact of recycling, and the result will often depend on which impact(s) take the highest priority. Unfortunately, the environment is often not the highest priority, as, at the end of the day, recycling is a business and not a charity. This means that the future of the industry relies heavily on the demands of the market. This in turn means that essentially, companies have no incentive to recycle items when there is more value placed in selling virgin material. An example of such a material is plastic - the price of recovered PET plastic is £222.50 per tonne, whereas virgin PET plastic is valued at £1084 per tonne [27].

However, as consumers grow more environmentally conscious, global demand for products made out of sustainable materials is growing. An example of a brand producing items from sustainable materials is active wear brand TALA [28], whose products are made out of materials like ocean plastic. An example of this is the fact that the percentage of "customers paying attention to sustainability of products in the Netherlands" was 34% in 2008, and grew to 53% in 2019 [29]. Therefore, as this demand grows, the value of recycled plastic will increase, and in turn the recycling market will develop.

However, still to be analysed is the cost vs benefit of recycling individual items. This is important as, in order to avoid contamination, items often need be cleaned before they are recycled. But does this necessary act consume more energy than is saved by recycling the item?

The UK standard for flow rates of kitchen taps is 4 to 6 litres/minute [30]. Under the assumption that washing an item, such as a plastic bottle, takes on average 1 minute, this uses around 5 litres, or around 80ml/s. Using the standard room temperature of 21 °C and the fact that the recommended safe hot water temperature for washbasins is 41°C [31], the desired temperature change to heat the water is therefore $\Delta T = 20°C$. From here, the equation [32] for the amount of thermal energy required to produce a temperature change ΔT can be used. This is equal to:

$$Q = mc\Delta T \tag{2.1}$$

Where m is the mass to be heated and c is the heat capacity of water $\approx 4.184J/g°C$. Using $m = 5000g$ and $\Delta T = 20°C$, this produces:

$$Q = 418,400J \approx 4.2kJ \tag{2.2}$$

1 Joule of energy is equivalent to $\approx 2.78x10^{-7}$kWh [33], and therefore

$$Q = 0.1163152kWh \approx 0.12kWh \tag{2.3}$$

But how does this compare to the amount of energy saved by recycling? - Analysis has found that recycling a single plastic bottle "can conserve enough energy to light a 60W light bulb for up to 6 hours" [34]. This is equivalent to $60 * 6 = 360Wh = 0.36kWh$ saved. Taking the 0.12kWh used to wash the bottle, this shows that recycling it would still *save* 0.24kWh, and therefore recycling the bottle is worth it in terms of energy saved.

2.2.6 Current Attitudes towards Recycling

Alongside surveys evaluating recycling performance, there are many surveys performed annually that look into how consumers' attitudes towards recycling are evolving. One such report, Viridor's Recycling Index [35] has analysed the UK's attitudes towards recycling over the past 4 years:

Figure 2.7: The UK's attitudes towards recycling [35, pg.4]

Figure 2.7 highlights a changing attitude and increased openness towards improving the country's recycling performance, but also an increased *expectation* of the government to take more responsibility. Interestingly, the report also found that "(consumers) need more information and support in order to feel reassured about how and what to recycle" and that "there is a demand for better recycling education for current and future generations"[35]. This was followed up with the following statistics [35]:

1. Only 1 in 3 (34%) are very confident they put different waste in the right bins.
2. Less than half (46%) say they are provided with enough information to know how and what to recycle.
3. 76% are frustrated about not having enough educational materials available on recycling.

Not only do these figures show that more information needs to be provided about how and what to recycle, but they also show a "significant drop in those who say they are provided with enough information" as well as a "rise in frustration in the lack of information and education materials on recycling" [35]. Another example of a survey analysing recycling behaviour is the Digest of Waste and Resource Statistics [7] completed in 2018. This found that 48% of those surveyed identified with the statement "I want to be a really good recycler and I take the trouble to ensure I'm doing everything right", compared to only 36% identifying with "Recycling is a good thing but I do not spend too much time worrying about it - the same things go in every week and I feel like I'm doing my bit". In addition to this, a survey was taken of 67 participants at the beginning of this project, assessing their current recycling behaviour and attitudes. When the participants were asked about how confident they were with respect to what can and cannot be recycled in their local area, only 4% responded saying they were very confident and knew how to recycle everything. The results can be seen in Figure 2.8:

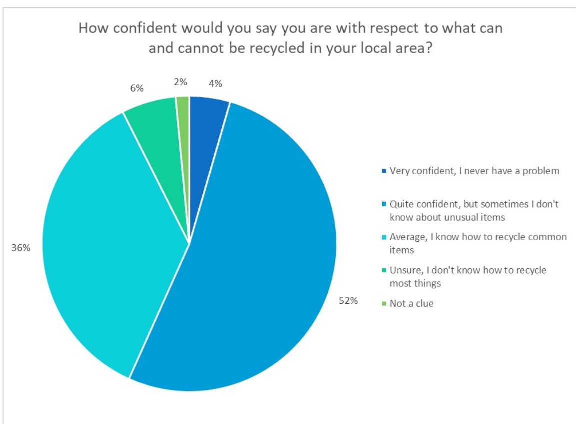


Figure 2.8: Initial Survey Results: Question 4

These results show that the majority of the participants of the survey have at least some uncertainty about what they can and cannot recycle. Furthermore, when asked about the availability and standard of information regarding how to recycle, 98.5% responded that they thought information about recycling should be more readily and easily available. Some of the further comments that were made can be seen below.

- "(I) would prefer if there was a more straightforward and clear way to find out (how to recycle)."
- "(Finding out how to recycle) is time consuming and I do not always manage to find the full information out."
- "The information is helpful for common items but often unclear/limited for unusual items."

The results of this survey are further discussed in Section 3.2.

2.2.7 Efforts to Improve the Recycling Rate

Many initiatives exist that are trying to improve the recycling rate in the UK. These vary from educational, where consumers are educated on how to correctly recycle, to motivational, where consumers are incentivised to recycle correctly, to a mixture of both solutions.

2.2.7.1 Online Resources

Plenty of resources are available online where the general public can research how, where and when items can be recycled. For example, RecycleNow offers a recycling locator [36], "helping (users) recycle and pass on (their) unwanted items for re-use". This offers information on (i) "where to Recycle a specific item", (ii) "what to put in your Recycling at home" and (iii) where to "find your nearest Recycling locations".

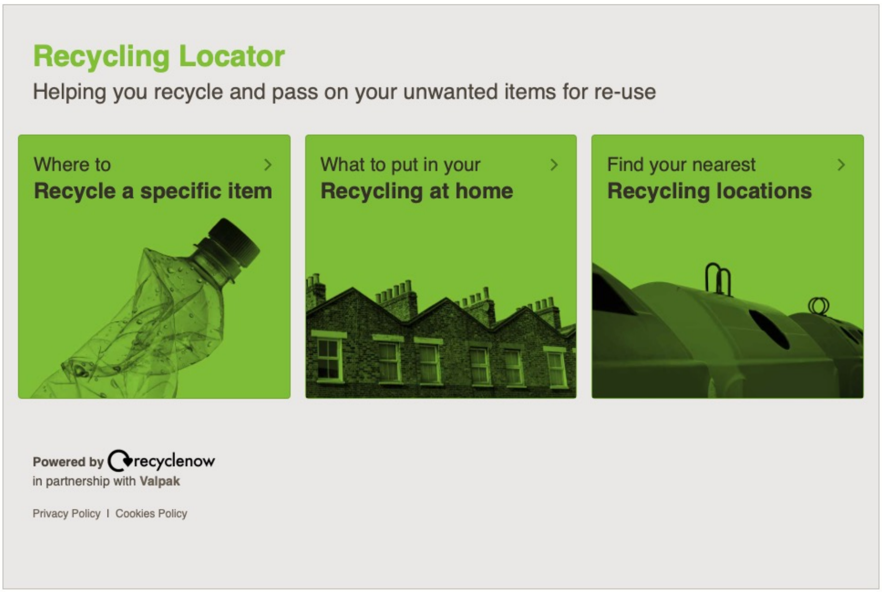


Figure 2.9: Example of Online Resources for Recycling Information - RecycleNow [36]

This is an extremely useful resource, as users can find out further recycling information about specific items *as well as* specific locations. Other online resources include local council or borough websites that provide location-specific information on recycling collections. Furthermore, Gov.uk has a resource where users can enter their local English or Welsh postcode - it will then direct them to the relevant council or borough website for their location [37].

2.2.7.2 Waste and Resources Action Programme

The Waste and Resources Action Programme, commonly known as WRAP, is a not-for-profit company that "works with governments, businesses and communities to deliver practical solutions to improve efficiency" [38] and "has in-depth experience of running national recycling campaigns that encourage consumers to take action" [39, pg. 33]. Their website offers many resources, guidelines and reports for consumers and companies alike.

An example of one of the campaigns that they run is RecycleNow, as mentioned in 2.2.7.1, used by over 90% of local authorities in the UK [40]. Furthermore, WRAP is working to produce a national standard for recycling. The aim of this is to produce "greater consistency in household recycling" [41]. They have produced an action plan to be carried out for 7 years, from 2018 to 2025[42, pg. 9] that can be summarised by the following points:

1. Packaging to be "sortable and recyclable".
2. A common set of materials to be recycled and householders to be able to "confidently and accurately" adhere to this.
3. Processing of materials to be "collected and sorted cost effectively" using "one of three systems."
4. "The domestic reprocessing sector to be sorted through the supply of materials of the required quality and quantity."
5. Barriers preventing improved consistency in recycling to be "identified and addressed."

This scheme would bring about standardisation throughout the UK in terms of what is recycled and how, making it easier for the average consumer to know how to recycle correctly, as there would be little to no variation between areas in the country. This in turn would allow for further consistency of the information provided on packaging by recycling labels.

2.2.7.3 Incentivisation

Alongside education about how to recycle correctly, research has found that users often need to also be motivated to recycle. One method of doing so is through incentivisation[2] schemes. A variety of such schemes have been tested by local councils and boroughs, examples of which include a direct cash reward scheme[3] and a chance to win a £50 voucher every time a resident put waste out for recycling[4]. However, incentivisation can also be used to motivate people in the opposite manner, i.e. persuading them *not* to do something. An example of how this can be achieved is through the introduction of a charge if they do what the scheme is trying to persuade them not to do. This approach has been implemented by local authorities in the context of recycling, through imposing levies on the "proportion of waste material going to landfill under landfill tax" [41], also known as "Pay As You Throw", or PAYT.

To assess their impact and levels of success, extensive research has been carried out on these schemes. For example, a 2014 study by Serco and Eunomia [43] of schemes at the time found that "voluntary incentive schemes appear to be a good technique for raising awareness of recycling activities, but do little in themselves to improve performance through the incentive offered." Furthermore, it found that "people seem to be more responsive to schemes where they strive to keep the money *in* their pocket rather than attempting to win some cash reward." This suggests that penalty schemes such as the landfill tax are likely to have a larger effect. The report went on to ask residents about their opinions of reward schemes.

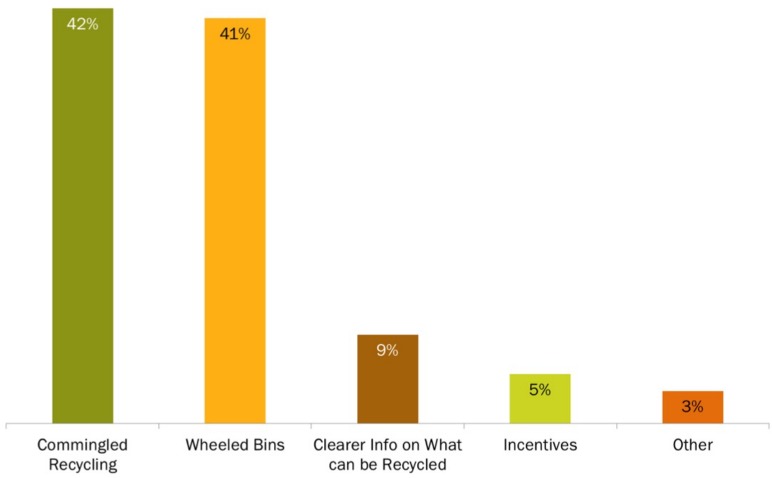


Figure 2.10: What would make residents recycle more? [43]

Figure 2.10 highlights that, according to residents, clearer information on what can be recycled would provide more motivation than incentive schemes, but that co-mingled recycling and wheeled bins would provide the

[2]Defined as providing incentives in order to motivate people to do something.
[3]Implemented by Wokingham Borough Council in April 2012 [43].
[4]Implemented in Bath from June 2015 to March 2016 [39].

most motivation. This result is perhaps due to the fact that co-mingled recycling and wheeled bins make the recycling process easier and simpler. This further supports the observation of the existence of a lack of motivation.

2.2.8 Potential Barriers to Improving the Recycling Rate

Whilst efforts are being made to improve the recycling rate, it is also important to consider potential barriers that may come in the way of any potential improvement. A report commissioned by SUEZ in 2015 [44] analysed factors that have “influence (over) England’s recycling performance”[44, pg. 7]. Inutuitively, any factors that may have the ability to influence the performance may have the ability to not improve it, and instead hinder it. These factors are discussed in the following sections.

2.2.8.1 Household Recycling Rates

In 2016, 27.3 million tonnes of household waste was generated, contributing to 12.25% of all waste generated in the UK [7, Fig 2.1, pg 30]. Household waste is the third biggest contributor to waste, behind both construction and commercial waste [45]. Recycling groups with the lowest recycling rate in households are food, plastics and textiles [7, Table 2.4, pg 35]. Of this 27.3 million tonnes, has already been mentioned that only 42% is recycled [44]. This is lower than a large portion of European countries, as seen in Figure 2.11.

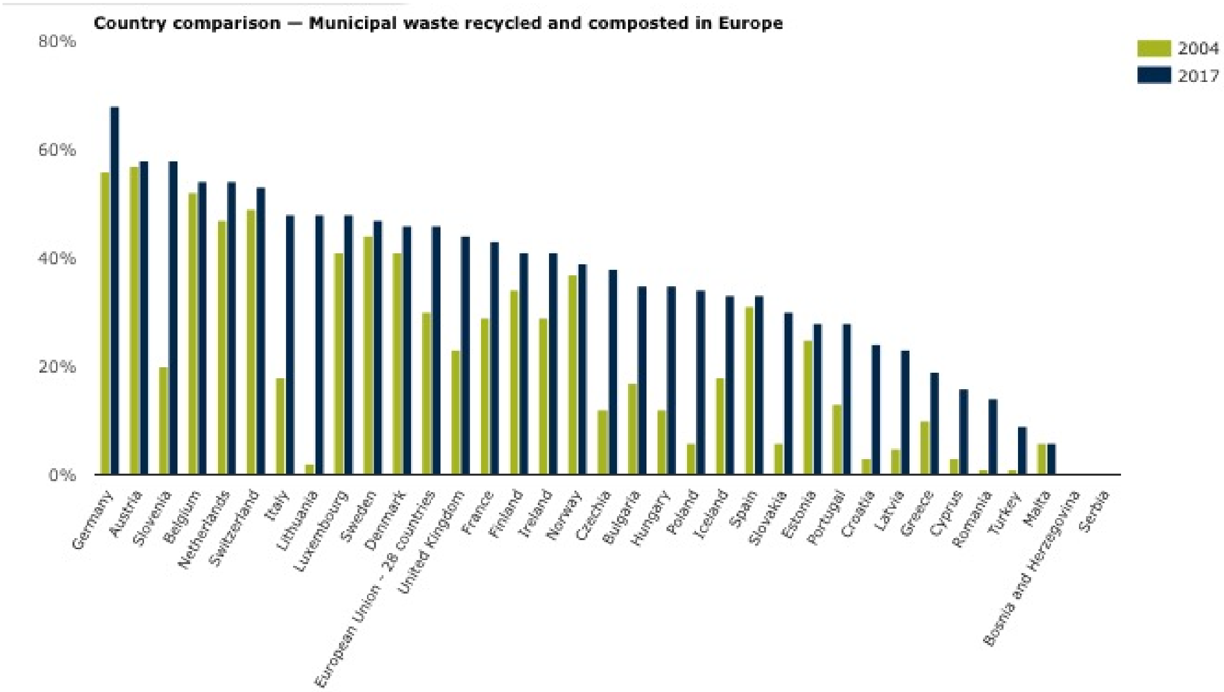


Figure 2.11: A Country Comparison of Municipal Waste Recycled and Composted in Europe in 2004 and 2017 [46, Fig. 2]

A recycling rate of 42% suggests a ’loss’ of 58%, which contributes almost 10% of total waste not recycled. Therefore, household recycling rates will have a considerable impact on the UK recycling rate as a whole.

2.2.8.2 Multi-occupancy Households

Research has found that recycling rates are much lower in households with multi-occupancy [44], such as student households or flat shares. This can arise due to factors such as increased waste from housemates sharing less items than families might, the frequent moving at the end of each academic year requiring packaging, increased breakages and loss, or tenants not being aware of how to correctly dispose of items due to frequent relocation. Within London, around 26% of households are in the private rented sector and 24% in the social housing sector [47]. This means that in London alone, up to 50% of housing could involve multiple-occupancy. Therefore, a reduced recycling rate in this type of housing can have a large impact on the overall recycling rate for the UK.

2.2.8.3 Policies and Incentives

The SUEZ report mentioned in 2.2.8 found that "England is lagging behind the European Union's high-performing recyclers who use stronger incentives, such as 'pay-as-you-throw' schemes" [44]. Furthermore, inconsistent policies throughout England cause a variety of recycling rates throughout the country[5], and hence the lower rates pull the overall average down. The report went on to calculate that, by implementing the best performing practices country-wide, the country's recycling rate would increase by 11%. This suggests that England's current recycling policies (or potentially lack thereof) is inhibiting its potential to improve.

2.2.8.4 Population Turnover

Population turnover can be defined as "the number of persons or households who have changed residence in a given time or area in relation to the total population or housing stock" [48, pg. 1]. This is caused by groups such as resident students, migrants or travellers [49, pg. 26]. Therefore, the higher the population turnover in a given area, the greater the required frequency of communications about how to correctly recycle, or about policies and incentives in place. Intuitively, the greater this required frequency of communications, the greater the difficulty it is to maintain such a rate and therefore the more people that will 'fall through the communication 'net'. This in turn may impact their ability and/or knowledge to recycle correctly and often.

2.2.8.5 Brexit

Leaving the European Union will inevitably have an effect on the percentage of waste recycled in the UK. This is because "3 million tonnes of UK domestic waste is exported to the EU annually for recycling or reuse" [50], as the UK "does not have the infrastructure in place to deal with the full recycling process" [51]. If countries decide to impose import taxes on waste coming from non-EU countries, the UK may be forced to pay up to £35 extra per tonne exported for incineration or recycling [51]. Furthermore, over half of the environmental law in the UK's comes from EU law, and only 66-75% of this "is easily transferable into national law", meaning that "transfer may have implications for the resource and waste management sector" [52]. Whilst hopefully these factors may not reduce the recycling rate, it will definitely make the process more complicated and expensive, and most likely infrastructure in the UK will need to be invested into in order to help counteract this change.

2.2.8.6 China's Waste Import Policy

Another factor that will affect the recycling rate in not only the UK, but also worldwide, is China's change in waste importing policy. Previously, China imported 39.5% of all plastic waste, pairings and scrap from the G7 countries, more than twice that of any other importer [53]. Considering more than just the G7 countries, the "sources of plastic waste imports into China in 2016" [54] can be seen in Figure 2.12.

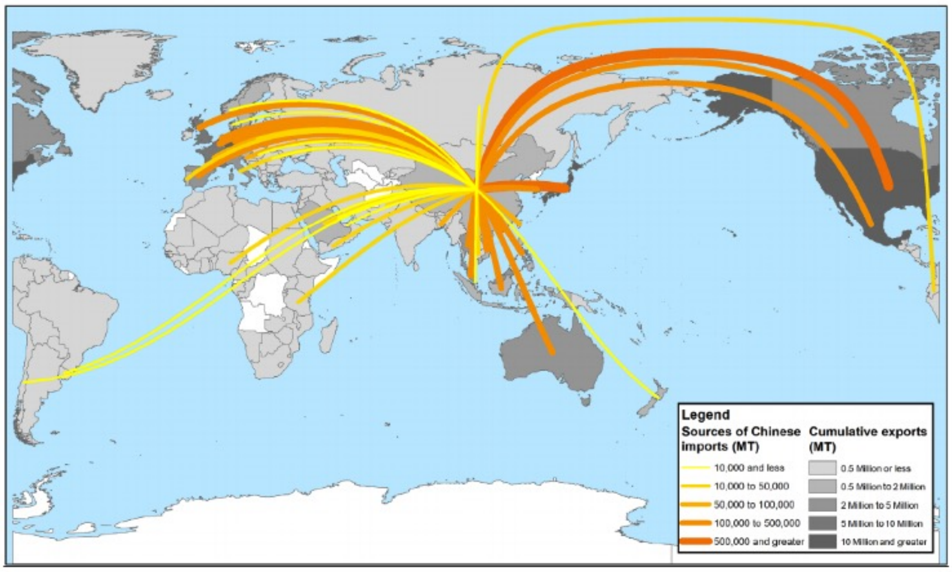


Figure 2.12: Sources of plastic waste imports into China in 2016 [54, Fig. 2, pg. 3]

[5]As seen in figure 2.1.

However, in December 2017, China changed their policy, banning certain imports[6][55], and severely restricting others to only those that meet Chinese control standard GB 16487.12 [54]. This policy change has meant that the majority of global waste has had to be diverted to other destinations. This can be seen in Figure 2.13. It shows that the amount imported to China has decreased by 95%. Therefore, as the amount of waste exported has not really decreased, the waste has ended up in places like Malaysia, Vietnam and Thailand.

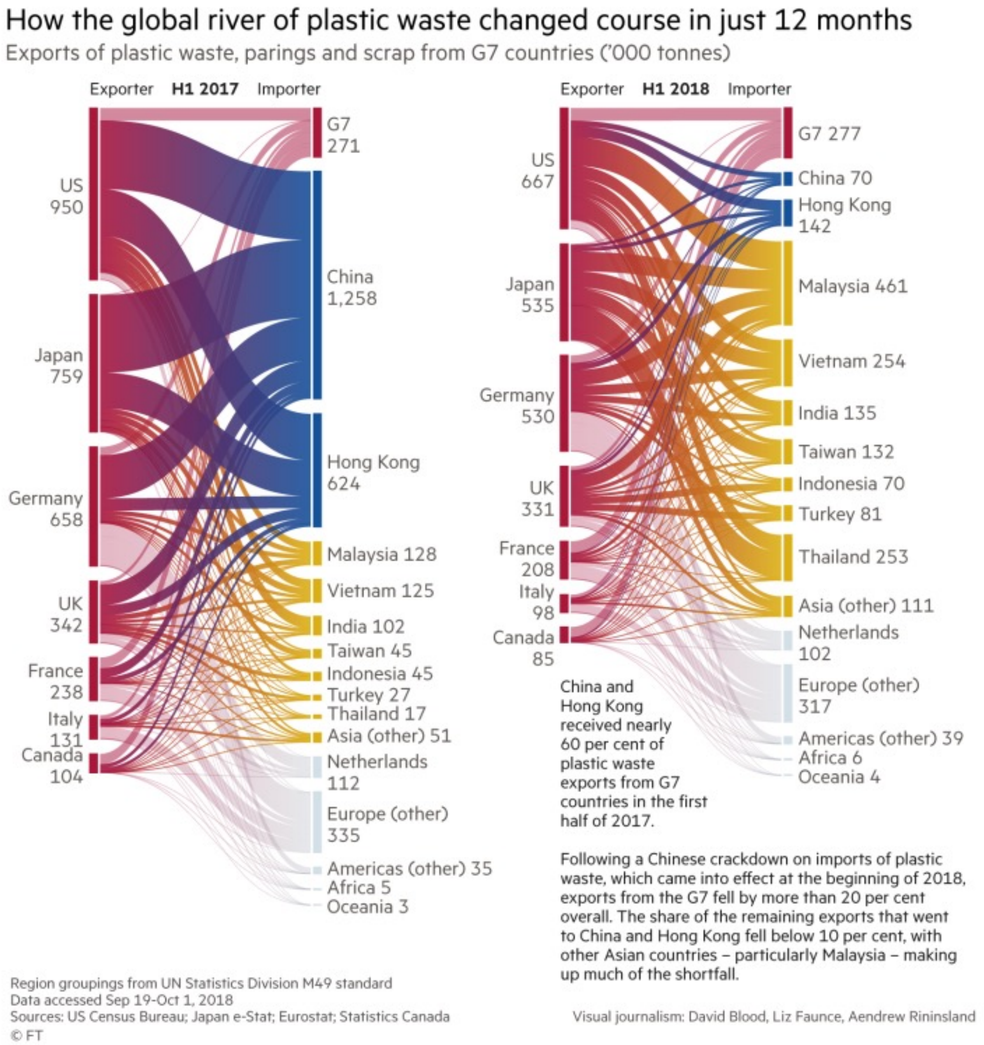


Figure 2.13: "How the global river of plastic waste changed course in just 12 months" [53]

With this change in waste flow, it is expected that all major waste exporting countries will have to process some of their waste locally. This will therefore have an impact on recycling rates as they will only be able to be improved if there is somewhere for the waste to go to be processed.

2.2.8.7 The COVID-19 Pandemic

A final factor preventing an improvement in recycling rates worldwide, let alone in the UK, is the on-going COVID-19 pandemic. This highly infectious virus has required countries throughout the world to impose strict lockdowns on its inhabitants, which has had a direct effect on waste and recycling, among countless other things.

For starters, the pandemic has caused a vast increase in the amount of medical waste used, which, for hygiene regions, must be disposed of in a sanitary manner, and therefore cannot be recycled. The extremely infectious nature of the virus has also caused an increase in the number of general items that must be disposed of in order to help prevent the spread, where they would normally be recycled, or sustainable alternatives used. An example of this is coffee cups. For instance, Starbucks, a multinational chain of coffee shops, banned the use of re-usable coffee cups and removed the 5p paper cup charge back in early March [56], in an effort to help

[6]Not only plastic imports.

minimise the number of infections. However, unfortunately, regular cardboard coffee cups are not recyclable, due to the material becoming saturated by the beverage that it holds. A further example of such a change is the fact that plastic bag charges and/or bans have also been lifted in many locations [57]. Despite these changes not necessarily being supported by government or scientific advice, many such choices are being made worldwide, as hygiene experts have “said (that) containing the virus should be a ‘greater priority’ than environmental concerns” [56], thus introducing a barrier against improving recycling rates.

Many changes have also occurred concerning recycling collections and centres themselves. For example, social distancing requirements have increased the complexity of operations, requiring the majority of centres to reduce the number of staff working at one time, if not closing centres altogether. These social distancing measures have also required tips to close, kerbside recycling collections to decrease in frequency and household recycling centres to not be able to accept individual drop offs. This has therefore reduced the ability of the general public to recycle everything that is concerned recyclable, and thus resulted in an increase of the waste just being disposed of. Furthermore, in countries such as the UK and Italy, those with suspected or confirmed cases of the virus have not be allowed to sort their waste [57], so as not to risk infecting others. This therefore also had an impact on the amount of household waste that is recycled.

2.2.9 Summary

In summary, whilst the rate of recycling has been improving over the years, there is still much work needed to be done to ensure that this continues to improve, rather than reaching a plateau. Key areas that should be focused on in order to achieve this are consumer attitudes (motivation) and correct classification (education). Classification is particularly important, as frequent mis-classification causes high contamination rates. This can, in turn, cause *entire recycling collections* to be rejected from recycling - in 2016 it was found that over 84% of recycling was rejected in this way [58]. Whilst in the long term this should be fixed by greater consistency across the country with respect to what can be recycled, as well as encouraging consumers to not generate as much waste in the first place, it can be seen that an effective short term solution is to provide some sort of tool or resource to help consumers correctly identify how to dispose of their waste.

A further benefit of increasing the amount recycled is that it will intuitively also increase the amount of recycled material available for use. This could then be used to meet the demand for products, as research has shown that if the UK invested in the recycling industry then “nearly 75% of domestic demand for products and packaging” [59] could be supplied from recycled materials. This could in turn decrease the amount of oil used to produce plastic products by more than 200 thousand metric tonnes [60]. Furthermore, increasing the local recycling capacity would allow the country to keep up with an improved recycling rate, without placing futher stress on external recycling locations. However, this is no small task, as currently only 9% is recycled on home turf, and more than two-thirds is exported to be recycled abroad [59].

However, such a change requires a shift in government policies and attitudes, as, without incentivisation or requirements imposed by the government, many companies lack motivation to source their products from recycled rather than virgin materials. This is because recycled materials have become more expensive than virgin ones due to factors like the Chinese import ban [61]. Whilst the UK government may be a world leader in their introduction of both the Net Zero Emissions Law [62] and the Plastic Ban [63], less progress has been made in the area of waste and recycling. However, this being said, efforts are starting to be made, with a ‘plastic packaging tax’, subject to consultation, set to be introduced in April 2022 on packaging “that does not include at least 30% recycled content, in a drive to reduce dependence on ‘virgin plastics’” [64].

2.3 Platform Development

It has been outlined in Chapter 1 that this project will be to produce an application to improve recycling performance through changing people's behaviour and attitude. Therefore, this section will highlight the different platforms available for such an application in order to make a decision about which would be the most appropriate to choose. The three main platforms available are:

1. Web
2. Native Mobile (iOS)
3. Native Mobile (Android)

2.3.1 User Base

2.3.1.1 Web Applications

As of April 2020, it was reported that there are over 4.5 billion active internet users [65]. This is over half of the global population, and also almost 10% more than the number of "unique mobile internet users" [65]. Of these 4.57 billion, the distribution of users according to location can be seen in Table 2.4.

Location	Percentage of Global Internet Users
Asia	50.3%
Europe	15.9%
Africa	11.5%
Latin America / Caribbean	10.1%
North America	7.6%
Middle East	3.9%
Oceania / Australia	0.6%

Table 2.4: Global Internet Users Distribution [66]

2.3.1.2 Mobile Applications

In contrast, as of September 2019, Android controlled 50.8% of the UK market share, and iOS only 48.71%[7] [67]. This means that in 2019, there was a roughly equal number of Android users compared to the number of iOS users. Therefore choosing one operating system over the other wouldn't drastically change the potential size of the user base in the UK. [68].

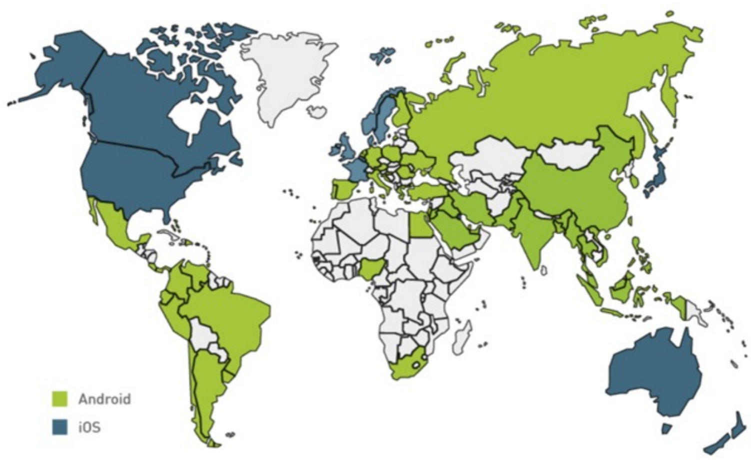


Figure 2.14: Average Global Preferences of iOS vs Android [69]

Figure 2.14 shows that overall, different countries have different preferences for mobile operating systems. This means that, depending on which country a developer may be producing their app for, a choice based on user base between iOS and Android may vary.

[7]The remaining 0.49% belongs to other market participants.

2.3.2 Development Complexity

2.3.2.1 Hardware

To develop a Web or an Android app, the developer simply needs a computer, of any brand, that runs any operating system. In contrast, to develop an iOS app, the developer must own or have access to a Mac that runs Xcode, Apple's app development software. This makes Android development more accessible in terms of range and cost of devices that have the required functionality. However, this project will be developed on an Apple MacBook Pro[8], meaning that neither operating system is particularly favourable over the other in this respect.

2.3.2.2 Device Fragmentation

Where web applications are not limited to a certain range of devices, iOS applications only runs on iPhones, and Android only on android-compatible devices. For iOS apps, there are 19 models that have been released since the launch of the original iPhone back in 2007 [70]. In stark contrast, there are now over 24,000 different Android models [71]. Such a large number makes it impossible to account for each individual device. This range of devices running the Android operating system is known as Device Fragmentation. The range of hardware for Android apps to be developed on means that an android developer would have to select a limited number of devices to support in order to make the development project feasible. This limit would influence the development complexity and also restrict the number of users that their app could reach.

2.3.2.3 Software

Web applications can be written in a whole range of languages, Android apps are predominantly Java based, and iOS is normally developed in Objective-C or Swift. Therefore, choosing an operating system based on this factor will come mainly down to preference. Considering the fact that I will be producing this app, and I have a background in C++, selecting iOS would prevent me from having to learn a new language before beginning development.

2.3.2.4 Software Fragmentation

Additional to device fragmentation, there is a larger variety of Android OS versions than iOS versions running on devices:

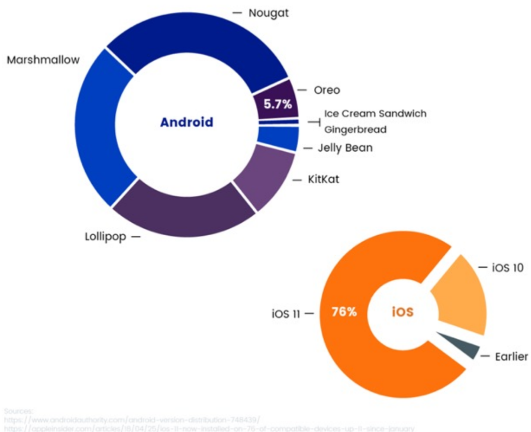


Figure 2.15: How many customers use the latest Mobile OS versions [72]

The rate of android operating system updates is also only slowing down, suggesting that this will become even more of a problem in later years – both Lollipop (2014) and Marshmallow (2015) were active on about 24% of Android devices a year from their respective releases, whereas Nougat (2017) was active on only about 21% of devices. [73] What's more, it's predicted that over half of all android devices are more than 2 years out of date [74]. Combining this with the open-source nature of the Google Play store causing distinct lack of standardisation, and it becomes complicated to produce an android app that will be functional on a significant portion of Android devices. In constrast, software fragmentation is a significantly smaller problem when it comes to iOS devices, as under one month after release, the latest version, iOS 13, was already on 50% of all devices [75].

[8]Apple and MacBook Pro are trademarks of Apple Inc., registered in the U.S. and other countries

2.3.2.5 Development Time and Cost

Assuming that a developer is equally proficient in any required programming language, the actual development time for a Web, iOS or Android App should be much the same. However, due to the device and software fragmentation mentioned above, unless the app in question is limited to a certain selection of devices, an Android app will take a much longer amount of time to test than its iOS or web counterpart would. As a result, this may cause a higher number of bugs, a longer further development time and increased cost for developing an app for the Android operating system.

2.3.3 Summary

This section has introduced the available platforms for development of an application, and highlighted the fact that there is no clear 'best choice' for every use case. Instead, each developer should carefully consider their target market, as well as system requirements, in order to make the optimal decision. However, it may also sometimes be the case that a developer is limited by resources, that may (at least partially) make this decision for them. An example of this is not owning or having access to a computer that runs an Apple operating system. This would mean that the developer would be unable to develop an iOS application, instead having to choose between Android and web applications. This actually highlights the exclusivity of Apple-related software and hardware products, which perhaps is the cause of the Google Play store hosting nearly 3 million apps [76], compared to the Apple store's 1.85 million [77]. This exclusivity has both positive and negative connotations - whilst it means that it prevents certain developers from being able to develop iPhone apps, and thus restricts consumers to the apps that they can use, it also perhaps guarantees a certain standard that an Apple customer can learn to expect.

2.4 Behaviour Design

Behaviour Design is the practice of designing to influence user behaviour. Since the advancement of technology during the 20th and 21st centuries, this field has evolved into the field of 'captology'. This was defined as "the study of how computer technology can be used to change people's opinions, or to persuade them to do something" by Dr. B. J. Fogg in 1996 [78]. Whilst this intuitively includes all technology, such as websites, games and mobile applications, this section will focus on mobile applications, due to their relevance to this project. An example of an app exploiting persuasive techniques to influence users to do certain things is Wish [79], an app that uses techniques such as countdowns, push notifications and extra discounts to persuade users to buy things.

To make the full use of persuasive technology, it is first important to understand *why* it works, as in what drives the behaviour of us as human beings. Whilst many frameworks exist, such as the Elaboration Likelihood Model [80] and the Influence Techniques Approach [81], an effective explanation is the Fogg Behaviour Model [82].

2.4.1 The Fogg Behaviour Model

This model states that "three elements must converge at the same moment for a behaviour to occur" [82] - Motivation, Ability and a Prompt.

The model first defines that all motivation is driven by either *physical factors*, in the form of a sensation, *emotional factors*, in the form of anticipation, or *social factors*, in the form of belonging and acceptance. For example, motivation caused by a physical factor could be how people are generally "motivated to (do something to) avoid pain" [83].

The model then goes on to link ability to motivation by the fact that "in order to perform a target behaviour, a person must have the ability to do so" [82]. Therefore, in order to become one step closer to influencing someone to do something, their ability must be increased. This can be achieved by one of three approaches:

1. Training to provide more skills
2. Providing a resource or tool to make the task easier
3. Making the task itself easier

The first option is generally the hardest, as humans are inherently lazy. Therefore, the most common approaches are options 2 or 3. This project is an example of option 2.

The third and final element of the Behaviour Model is a Prompt. This is what *causes* the target behaviour to happen. Again, it is split into 3 types. This is best explained by Figure 2.16.

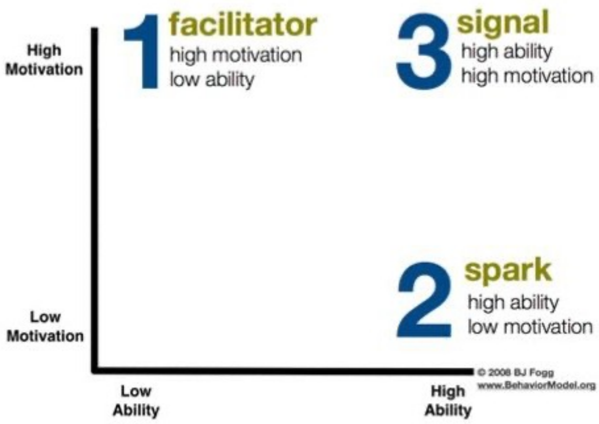


Figure 2.16: The Fogg Behaviour Model - Different Prompts [82]

The key idea behind a prompt is that influencing a user to perform a simple action may result in them initiating a chain of behaviours, ending in the original target behaviour. For example, if one persuades "someone to walk for 10 minutes a day, (then) that person may buy some walking shoes without any external triggering or intervention" [82]. Therefore, the art of providing a prompt is to design a behaviour chain, but only *prompt* the user to perform the first action in the chain.

The relationship between the three elements of Fogg's model is best explained in Figure 2.17:

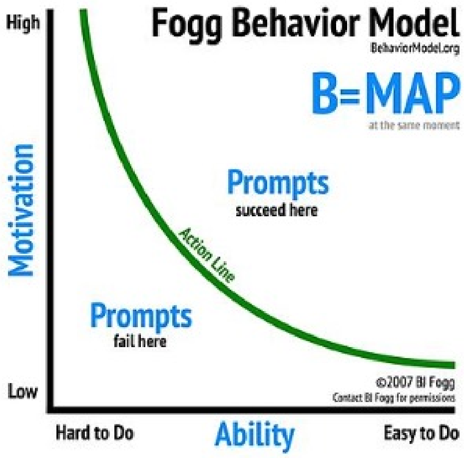


Figure 2.17: The Fogg Behaviour Model [82]

This graph shows an inverse relationship between motivation and ability. This represents the fact that the greater someone's ability, the easier an action becomes, and from there, the lower the motivation required for the person to perform it. Furthermore, the (green) action line denotes what actions are performed - below this line, and the user does not have enough motivation and/or ability to perform the task. At this point, no matter what the prompt is, the action will not get performed. The opposite is true for above the action line. Good models should aim for both high motivation and high ability in order to maximise the chance of the action being performed.

In conclusion, when persuading a user to perform an action, motivation provides the *incentive*, ability provides the *means*, and a prompt *sets the action in motion.*

2.4.2 Designing with Intent

A shortcoming of Fogg's model is the fact that it does not outline how to *transform* the principles discussed into system design. In fact, "little work has been done (at all) to *link* ideas and techniques from disparate fields, identify common themes and present them in a form which can be applied during the design process" [84]. Therefore, the Design with Intent Method "aims to address this deficiency, suggesting relevant design techniques for influencing types of behaviour" [84]. The approach of the model is best described by Figure 2.18.

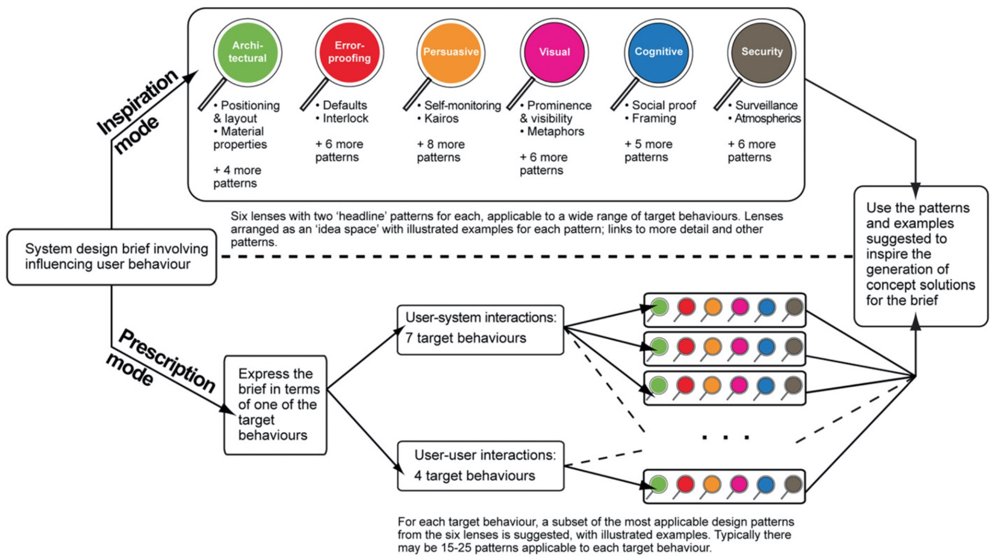


Figure 2.18: Design with Intent Method Structure [84, Fig. 1, pg. 384]

.

This shows that the model consists of two 'modes' [84]:

- **Inspiration**: "the designer takes inspiration from a set of 'headline' design patterns which are applicable to a wide range of target behaviours, grouped into six different 'lenses' representing particular disciplinary perspectives on using design to influence behaviour".
- **Prescription**: "the designer formulates the brief in terms of one of a range of target behaviours, describing interactions; for each target behaviour, a subset of the most applicable design patterns from each lens is presented".

The model defines target behaviours as "intended outcomes or particular user behaviours which we want to achieve through design" [84]. Example target behaviours can be seen in Figure 2.19. As this project will have a set of specific target behaviours focused around improving a user's recycling performance, the prescription mode is the most suitable, and will be focused on in this section.

User–system interaction: influencing interactions between a user and a system		*Examples*
S1	The user follows a process or path, doing things in a sequence chosen by the designer	Customer places order via website without missing out any steps
S2	The user follows a process or path that's optimised for those particular circumstances	User only spends as much time as really needed in the shower
S3	Decision among alternatives: a user's choice is guided	Diners choose healthier meal in office canteen
S4	Only certain users/groups of users can use something	Only users who know PIN can access bank account via ATM
S5	Only users already behaving in a certain way get to use something	If a driver's travelling below the speed limit, the next set of traffic lights turn green, otherwise they stay red
S6	No users can use something in a particular way, regardless of who they are or what they've done before	Park bench fitted with central armrest to prevent anyone lying down
S7	Users only get functionality when environmental criteria are satisfied	Office lighting cannot be switched on if ambient daylight adequate
User–user interaction: influencing interaction between users and other users, mediated by system		*Examples*
U1	Multiple users are kept separate so they don't affect each other while using a system	Traffic follows one-way system into/out of car park
U2	Users (and groups of users) do interact with, and affect each other while using a system	Staff from different departments mix socially in a building's atrium
U3	Users can't block or dominate a system to the exclusion of others	Wide pedestrian concourses prevent groups blocking passage for others
U4	Controlled rate of flow or passage of users	Visitors to popular museum exhibit routed past it slowly on moving walkway

Figure 2.19: Design with Intent Method - Target Behaviours with Examples [84, Table. 2, pg. 386]

Once the target behaviours have been identified, the design patterns for each behaviour can be created. These may come from each of the six lenses postulated by the model, as seen in the top half of Figure 2.18 and also in Figure 2.20.:

Architectural lens	The architectural lens draws on techniques used to influence user behaviour in architecture, urban planning and related disciplines such as traffic management and crime prevention through environmental design (Crowe, 2000; Katyal, 2002; see also the security lens). While the techniques have been developed in the built environment (e.g. Alexander et al., 1977), many ideas can also be applied in interaction and product design, even in software or services; they are effectively about using the *structure of systems* to influence behaviour.
Errorproofing lens	The errorproofing lens treats deviations from the target behaviour as 'errors' which design can help to avoid, either by making it easier for users to work without making errors, or by making errors impossible in the first place (Shingo, 1986; Chase and Stewart, 2002; Grout, 2007). This view on influencing behaviour is often found in health & safety-related design, medical device design and manufacturing engineering.
Persuasive lens	The persuasive lens represents the emerging field of persuasive technology (Fogg, 2003), where computers, mobile phones and other systems with interfaces are used to persuade users: changing attitudes and so changing behaviour through contextual information, advice and guidance.
Visual lens	The visual lens combines ideas from product semantics, semiotics, ecological psychology and Gestalt psychology about how users perceive patterns and meanings as they interact with the systems around them, and the use of metaphors (e.g. Saffer, 2005; Barr et al., 2002).
Cognitive lens	The cognitive lens draws on research in behavioural economics looking at how people make decisions, and how this is affected by heuristics and biases (Kahneman et al., 1982). If designers understand how users make interaction decisions, that knowledge can be used to influence interaction behaviour. Where users often make poor decisions, design can help counter this.
Security lens	The security lens represents a 'security' worldview, i.e. that undesired user behaviour is something to deter and/or prevent though 'countermeasures' (Schneier, 2003) designed into products, systems and environments, both physically and online, with examples such as digital rights management. From a designer's point of view, this can be an 'unfriendly' and, in some circumstances unethical view to take, effectively treating users as 'guilty until proven innocent'.

Figure 2.20: Design with Intent Method - "six 'lenses' on influencing user behaviour" [84, Table. 1, pg. 386]

These design patterns are used to design the user interface and experience (UI and UX) of the app. The example discussed in the report [84] is the design of a new ATM. With the example target behaviour of "we do not want users to leave their cards in ATMs after use", suggested design patterns can be seen in Table 2.5.

Lens	Example design pattern
Architectural	"Change spacing between interface elements, so that card slot adjacent to cash dispensing slot."
Error-proofing	"Don't dispense cash until card has been removed."
Persuasive	"Wizard, on-screen checklist or flow-chart indicating what actions user has taken and needs to take to complete process correctly."
Visual	"Simply make the elements which are important, e.g. the card slot,very prominent in the ATM fascia,by size, or using bright/contrasting colours."
Cognitive	"Make it imperative not to lose their card, by charging penalty fines for replacement."
Security	"Use of cameras/monitoring(overt or covert) to alert user to error: 'Excuse me, sir/madam, don't forget your card.' "

Table 2.5: Design with Intent Method - Example Design Patterns [84, Table. 3, pg, 389]

In conclusion, the 'prescription' mode of the Design with Intent method involves deciding on the *target behaviours* of the user and, from there, planning design patterns based on the 6 'lenses'. Linking this to Fogg's model, the target behaviour(s) can be thought of the initial actions to set a behaviour chain in motion, and the design patterns as the ability and/or prompts.

2.4.3 Target Behaviours

Sections 2.4.1 and 2.4.2 discussed designing to achieve target behaviours, with the latter going on to discuss their corresponding design patterns. To achieve these patterns, psychological principles can often be exploited. This section will cover some example target behaviours, as well as techniques and theorems to achieve their design patterns.

2.4.3.1 Controlling in-app behaviour

Designing the UI of an app in a particular way can influence users to perform certain actions, or visit certain in-app pages more frequently than others, which can be particularly useful in app design. An example of this is if the designer wants to influence the user to subscribe to a service provided by their app, they could encourage the user to visit the 'subscription' page of the app.

This could be achieved using the **Serial Position Effect**. This effect observes people's tendency to "recall the first and last items in a series best, and the middle items worst" [85]. Users being more likely to remember the first item is known as the primacy effect, and the last item the regency effect, as seen in Figure 2.21.

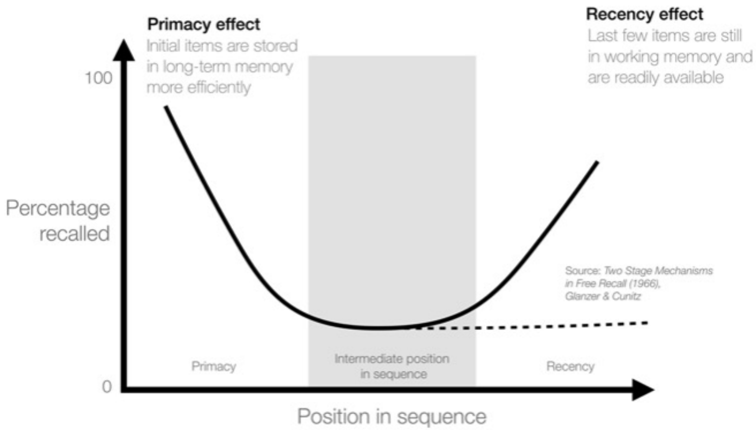


Figure 2.21: The Serial Position Effect [86]

An example of exploiting this effect in terms of technical design is when designing the positions of items in a menu bar - placing the most important items at the top (or left), and then the next most important at the bottom (or right) of the menu will increase the likelihood of a user visiting these pages.

A further example of manipulating user in-app behaviour is the **Chameleon Effect**. Also known as Mirroring, this is the observed phenomenon of how "one person unconsciously imitates the gesture, speech pattern, or attitude of another" [87]. App designers often make use of this effect to get users to feel a certain way about a certain event or occurrence in the app. This is often defined as 'Emotional Design', or "products that elicit appropriate emotions, in order to create a positive experience for the user" [88]. This is a key concept to exploit, as influencing a user's emotions can influence their opinion or perception of the app.

This effect can be implemented in many ways, but a useful example is on Duolingo, a language learning app [89]. This app has a cartoon green bird, known as 'Duo', as a mascot. When users get too many questions wrong in a round, Duo looks upset. The app designers have therefore made use of the Chameleon Effect to make users feel bad about getting questions wrong, and motivate them to get them right, and hence stay on the app for a longer amount of time.

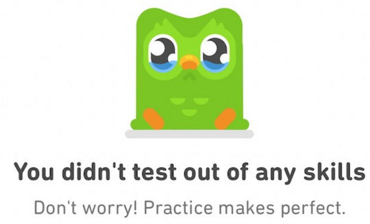


Figure 2.22: Duolingo Screenshot - Sad Duo [89]

A final example of influencing user behaviour in-app is **Pavlovian Conditioning**, also known as classical conditioning. This is when a subject is 'trained' to give a conditioned response to a conditioned stimuli, that was previously an unconditioned response to an unconditioned stimuli. This technique was famously practiced on dogs by Ivan Pavlov in 1897 [90];

Figure 2.23: "Classical Conditioning Diagram" [90]

An example of this psychology technique being applied to UI and UX is the use of the same design of a button consistently performing a certain action, in order to train the user to expect that same response every time. This expectation could then be used to get the user to do something they did not originally plan to do, such as not cancel their subscription, by switching around the outcomes that corresponded to each of the button appearances.

2.4.3.2 Motivating users to complete an activity

Motivating users to complete an activity can be difficult, especially when it is not part of their daily routine. However, one example of how to achieve motivation is the **Chameleon Effect**, as discussed in Section 2.4.3.1 - the sad Duo example discussed can be used to motivate users to do better next time.

A further technique that can be used to motivate users to complete an activity, to the best of their ability, is **Social Comparison**. This was first defined by Festinger in 1954 "as a driving force behind competitive motivation" [91]. This relates closely to the concept of **Social Motivation** and belonging, as mentioned in Section 2.4.1. Social motivation can be defined as "the human need to interact with other humans and to be accepted by them" [92]. It can therefore be seen that the presence of social interaction can influence user behaviours. An example of how this could be used is to persuade users to get involved in tasks in the app is by showing them the activity of people that they (may) know. This could then be taken one step further, by implementing competition between users in order to motivate users to out-perform their family and friends.

This concept of social competition links closely to the technique of **Gamification**. This term has been around since the beginning of the century, but was defined in 2014 as "the use of game mechanics and experience design to digitally engage and motivate people to achieve their goals" [93]. This technique has massively grown in popularity over the past 5 years, with the global education gamification market alone growing by more than a factor of 10, from $93 million in 2015, to $1.25 billion in 2020 [94]. At this point, it is important to note that *to gamify* does not mean to convert an activity fully into a game. Instead, it is to add *elements* of a game to the system, such as a points, badges or a reward system. It therefore works to provide *motivation* to users, one of the three elements discussed in Section 2.4.1. Examples of how this could be used to achieve certain target behaviours can be seen in Figure 2.24.

Game Element	Purpose	Example
Points	These are awarded based on a player's participation in certain actions or performing desirable behaviours. They could highlight an achievement or status.	Inviting a friend to join a platform could earn you 5 points.
Levels	They are used to show a player's progression through a gamified system.	Moving from a less difficult game level to a more difficult one.
Leaderboards	These are used to create a platform for players to compare their performance to each other.	Comparing positions
Achievements	They are used as a form of positive reinforcement and feedback to highlight mastery.	Badges and Trophies
Narratives	Create a story to cause you players to be engaged. These stories can also help players learn fast and put into practice what they learnt.	A customer is stranded on the top of hill, it is your mission to save that customer.
Feedback	Feedbacks are used to continuously keep the player in the know of the consequence of their actions (either positive or negative).	A simple text, Points, Progression
Adventure	Allows the player to look for and find treasured times.	Hidden treasures.

Figure 2.24: Examples of Gamification [95, Table. 2, pg. 18]

2.4.3.3 Getting users to re-visit an app

Persuading users to re-visit an app after the first visit is recognised as a key milestone in app development. Furthermore, getting users to *continue* to re-visit is known as user retention and, along with number of daily and monthly active users (DAU and MAU), is a key metric that defines an app's success. Several techniques can be employed to persuade users to re-visit an app, but one example is push notifications.

Push notifications have been extensively used since the introduction of push emails on Blackberry devices at the beginning of the century. Before this point, users had to enter their email application and *manually* check for any new emails. This sparked a change in the industry, and in June 2009, Apple released iOS 3.0, which included Apple's version of push notifications [96]. Soon after, Android adopted this too, and push notifications have been used extensively ever since. In fact, they are so common that they have now caused the **Pavlovian Conditioning** effect, as mentioned in Section 2.4.3.1, to occur. When a push notification alert occurs on a user's device, they now expect important information, such as a message, to be displayed, and so will check their phone. This effect can be exploited by apps - sending a push notification to the user can often influence them to pick up their device. Here, the conditioned stimulus is the push notification, and the conditioned response is the user picking up their device. At this point, if the notification has been designed well enough to catch their attention, they will often re-visit the app. In fact, research has found that "consumers who receive push notifications from shopping apps spend almost double the time using these apps as those who do not opt in" [97].

In terms of Fogg's Behaviour Model discussed in Section 2.4.1, a push notification is an example of a prompt, the user picking up their phone is the initial action, and the process of entering their device to re-visit the app is the behaviour chain resulting in the target behaviour.

2.4.3.4 Improving user experience

Improving the UX is inherently linked to influencing users to re-visit an app. This is because badly designed apps are often hard to use, making the user not enjoy the experience. This in turn can de-motivate them from wanting to return to the app. An example of a psychological principle that can be exploited to improve the user experience is the **Mere-Exposure Effect**. This is a "psychological phenomenon by which people tend to develop a preference for things merely because they are familiar with them" [98]. In UI and UX design, this essentially means that users often exhibit a preference for apps and websites with similar designs. This is one of the reasons why most of the successful apps today have similar layouts, colour schemes and fonts. Therefore, an app is likely to be well received by their user base if the design is similar to existing, well known, successful apps, like Instagram, Twitter and WhatsApp. Furthermore, if a user feels a preference towards an app, they are more likely to enjoy the experience of interacting with it.

A further example is the **Placebo Effect**. This is where the subject thinks that something has an effect, when in fact nothing has changed. This is commonly seen in medicine, when a subject is given a 'placebo' pill but is told it is a particular pill with a a certain effect. This causes certain signals in their brain to fire, tricking them into experiencing a decrease, or complete disappearance of symptoms. This technique is often employed to save resources - research has even shown that "the placebo was 50% as effective as the real drug to reduce pain after a migraine attack" [99]. An example of applying this effect to app design is when a user refreshes an app. In reality, the designers will have no control over how long this takes, as it is determined solely by the strength and speed of the internet connection. However, implementing a 'pull-to-refresh' feature and an animation, such as a spinning circle, whilst the page is loading, will allow the user to feel in control and improve their experience.

A final example of how to improve the user experience is by making decisions that may need to be made by the user, easier. This can be achieved by following the principle of **Hick's Law**. This law observes that "the more choices a person is presented with, the longer the person will take to reach a decision" [100]. The formula is defined as:

$$RT = a + blog_2(n) \tag{2.4}$$

Where RT is the reaction time, n the number of options, and a, b, the measurable constants that depend on the task being carried out, and the conditions under which it is being carried out [101].

Applying this law to UI and UX is straightforward; If the designer wants to reduce the time taken for a user to make a decision, they should reduce the number of options available. Hick's Law is most commonly applied in the navigation systems of websites and applications - too many options, such as the choice to visit every page on the site, is definitely going to overwhelm the site visitor or app user and negatively impacting their experience. Instead, navigation is typically split into categories, such as an online fashion retailer grouping their products by high-level clothing types, such as ‘tops’, rather than all the different styles of top available.

2.4.4 Ethics of Persuasions

There is a fine line between persuasion, and coercion or manipulation. Therefore, when designing persuasive systems, the ethics behind the design must be considered. Fortunately, with the growth of the use of persuasive technology, work has also been done to develop further design principles that help designers ensure that their work is ethically sound. A key, and perhaps obvious example of such principles, is, as introduced by Berdichevsky and Neuenschwander [102] in 1999, the fact that the designer “of a persuasive technology should never seek to persuade a person of something they themselves would not consent to be persuaded to do” [103, Table. 1, pg. 93]. Furthermore, Smids postulated that “a voluntary change brought about by a persuasive technology implies both the absence of controlling influences like manipulation and coercion, and an agent who acts intentional in changing his behaviour” [104]. Therefore, considering these two principles alone, a designer should believe in the change they are trying to bring about, and the user should be voluntarily open to bringing about that change. However, ethics is such a variable, and even subjective, concept that no single set of principles exist that is suitable for design cases. Each designer should therefore take responsibility for the design that they are implementing.

2.4.5 Summary

Behaviour Design is a widely-used and highly-useful technique to influence user behaviour, that can be used by designers to ensure that users get the most out of their app. Furthermore, multiple design frameworks and principles exist, produced by industry-leaders, that can be utilised to formulate such a design in a structured and effective manner. However, careful thought and consideration must be undertaken at every step of the design process to ensure that the designer is not taking advantage of their unique access to users to manipulate them in such a manner that the designers themselves would not appreciate. Designers should therefore aim for choice, transparency and mutual benefit in their technology, and prevent deception and self-interest from featuring in the design. As long as designers accept responsibility for the work that they are producing, persuasive technology and behaviour design is a powerful tool that could bring about substantial change.

2.5 Machine Learning

Two key applications of machine learning are text recognition and image classification, both of which can be used to identify an item to be recycled. The former could be used to recognise a recycling or product label, and the latter, the item itself. Whilst the area of Machine Learning, and hence the number of different techniques used, is vast, extensive research has found that each of these applications has certain techniques that provide the best performance.

2.5.1 Dataset Selection

The first stage to building a machine learning model is to establish what dataset it will be trained on. This could be a pre-built model, such as MNIST[105], Caltech101[106] or IMDB Reviews[107], or even a custom dataset, specific to the required application. In fact, pre-built models are sometimes necessary, in new areas or research or novel applications of machine learning and computer vision.

However, creating custom datasets can be complex. Firstly, it is important to make sure that classes are evenly distributed, and don't exhibit any particular biases. Furthermore, a dataset needs to contain enough data to adequately train a model, but not so much that the model becomes overfitted to the data. It is also often beneficial to ensure that the data is sufficiently varied and augmented (i.e. different angles, lighting etc) so that the model can better deal with unseen data, i.e. any data not previously seen during training. Fortunately, this can also often be achieved by performing augmentation on the data already belonging to the dataset, such as seen in Figure 2.25. This can also help to solve the problem of too small a dataset.

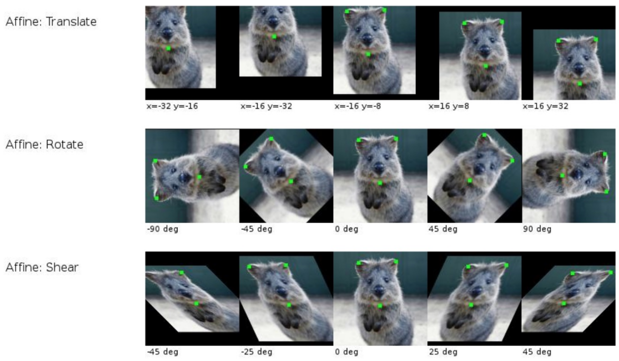


Figure 2.25: Data Augmentation[108]

Once such obstacles have been taken into consideration, datasets can be built in one of two ways; manually, or using an automated process. Whilst manually creating a dataset may be time consuming, it provides a high level of control over the data being included. However, it increases the risk of biased data. Alternatively, a dataset can be built using a method such as web scraping, or perhaps scripting a webcam to (with consent) take pictures of a user over a given time frame. Approaches such as these are not only much more time efficient, but allow utilisation of a much wider range of sources. However, they are also more computationally expensive and potentially require the time-intensive task of the designer going through the generated dataset to check for correct labelling and relevant data. Therefore, the optimal approach to creating a dataset will depend on the intended use, and potentially result in a combination of both manual and automated processes.

2.5.2 Text Recognition

The main goal of text recognition is to detect and interpret text that appears in data such as documents, images or videos. Whilst a technique known as Optical Character Recognition, or OCR, is well developed for text recognition and interpretation in documents, when applied to more complex data such as images or videos, the problem of text recognition is often still considered as open[109]. Thus, techniques have been developed to detect, locate and extract text before applying OCR. The high-level process of text recognition has hence evolved to implement four key stages[110]:

1. Text Detection
2. Text Location
3. Text Extraction
4. Character or Word Recognition

Inspired by [110], a visual representation of this process can be seen in Figure 2.26.

Figure 2.26: Text Recognition Process

Essentially, a text recognition model is be able to take an image or video as input, and output any text found as ASCII Text. This can then go on to be further interpreted by the user, or another computer program. Multiple approaches exist for each stage, each exhibiting different strengths and weaknesses, as well as particular formats of data where they can achieve optimal performance. Whilst text recognition has been developed to be used on multiple styles of data, such as images, videos etc, for simplicity, this section will focus on images.

2.5.2.1 Text Detection and Location

This is the initial stage, where "no prior information (is available) on whether or not the input contains any text"[111]. Essentially, text detection is the process of identifying if text is present in a given image or frame. This stage is often considered as key for optimising a text recognition's model efficiency, as images determined to not contain text can be discarded early-on in the process, saving resources and time that would have otherwise been spent on this image being fed through the later text recognition stages. Often combined with detection, Text Localization is then used to determine "the text location and generate bounding boxes around it"[111].

For the most part, the techniques of detection and location can be classified into 2 groups of 2 methods each; region based, which can be further split into connected component analysis and edge based, and background based, which can be split into texture based and frequency based methods[112, 113]. The optimal application of each of these various approaches will depend on the format of the data, but their overall advantages and disadvantages can be seen in Table 2.6.

	Method	**About**	**Discussion**
Region	**Connected Components**	Similar components are grouped into increasingly larger components, until they form text regions[111, 112].	Fast, but struggles with in-homogeneous or non-dominant text, or when characters are not well separated. Best performance arises when "either the text is monochrome, or the background"[114].
	Edges	Exploits "differences between the text colour and its immediate background"[110].	Fast, with high recall but false positives and/or negatives are common. Can struggle with low-contrast images or letters with "strong parallel edges" (such as i or l)[115].
Background	**Textures**	"Distinguishes text from backgrounds using the textural properties of the text"[110].	Computationally complex and expensive, unnecessary scanning of non-text regions and sensitive to background noise[112].
	Frequencies	"Text is extracted from the background in the frequency domain"[112].	Computationally complex and expensive, and the frequency representation is not necessarily better than the spatial one. Examples include Wavelets and the Fast Fourier Transform (FFT).

Table 2.6: Comparison of styles of Text Detection and Location

2.5.2.2 Text Extraction

Also known as Binarisation or Segmentation, this stage involves separating the text from its background and "converting the text image to a binary image and enhancing it"[111]. In this way, it can be thought of as a pre-processing stage for the process of Character Recognition.

The first step of this stage is essentially a binary classification problem; either a pixel is part of the text, or it is part of the background. However, this classification will depend on whether the input image is grayscale or in colour. For example, for a grayscale image, assuming a (relatively) plain background, the background will be white and the text black, or vice versa. Therefore, the text and background can be quickly and easily identified, and thus separated, by splitting an image into black and white pixels by assigning a threshold - Intensities above this threshold are white, and below, black. Its only potential hurdle is selecting the optimal threshold. This approach is often referred to as pixel counting[116] or the threshold based method[117]. Due to the increased range of potential values that pixels can take, and increase complexity as to what each value represents, colour images require a more sophisticated approach. Such an approach, suitable for both grayscale and colour images, is a clustering method that applies "K-means clustering to generate colour layers in YCbCr colorspace, and then heuristic rules are exploited to identify the text layer"[117, 118]

Once the text image has been extracted, the 'pre-processing' before the character/word recognition involves "digitization, noise removal, binarization and normalization"[116] - standard image pre-processing techniques. This stage is often argued as "crucial for recognition accuracy"[117], as it isolates text from potentially degraded or complex images, so that the dataset has a more constant standard of image quality and makes the recognition model's job easier by removing irrelevant parts of the images. Examples of the process of digitization and segmentation can be seen in Figures 2.27a and 2.27b respectively.

(a) Digitization

(b) Segmentation

Figure 2.27: Examples of Stages of Text Recognition[119, Figs 1. & 2.]

Challenges are often encountered in this stage of the process due to the "variation in fonts, size, colour, alignment, orientation, illumination and background"[110] of images. This is somewhat to be expected, as intuitively, text from images with clearer text borders, better illumination and plainer backgrounds will be easier to extract than images that are of poorer quality.

2.5.2.3 Character or Word Recognition

With the images from the dataset adequately pre-processed, the process of character and/or word recognition can begin. As mentioned previously, the industry leading approach is to implement a technique known as Optical Character Recogntition, or OCR. Alongside manual implementation of the technique, many commercial applications exist, the most popular of which is most probably Google's Tesseract[120], or their Cloud Vision[121].

Optical Character Recognition actually has a very interesting history, originating from a technology invented by Emanuel Goldberg in 1914 to help the blind, namely "a machine that read characters and converted them into standard telegraph code"[122, 123]. This technology has only continued to evolve and develop over the years, into the technique known and used today, with the "first widely commercialised" version being IBM 1418 [124]. Versions of OCR even exist today that are able to recognise other alphabets, such as Chinese and Arabic. In fact, OCR is often considered as "one of the earliest addressed computer vision tasks"[125].

As with the previous stages, the classification stage of text recognition can be implemented by various approaches, including, but not limited to, matrix matching, feature extraction, structural analysis or neural networks[126], all of which are defined as approaches to achive optical character recognition. Furthermore, OCR can be categorised into either offline or online recognition, where the former is where "the source is either an image or a scanned document"[119], and the latter, "successive points are represented as a function of time"[119]. Essentially, offline OCR is performed on pre-existing data, whereas online OCR is in real time. Later in this report, several techniques will be implemented and compared to ascertain the approach with the highest performance.

2.5.3 Image Classification

As the name might suggest, image classification is the process of a computer program classifying images based on the category of the object(s) that the image contains. This process can be split into 4 steps;

1. **Image Pre-processing**: The purpose of this step is to process input data into a form expected by the model. This involves tasks such as removing large areas of blank space or increasing the illumination of an image, in order to improve the quality of data and maximise the chances of a correct classification.

2. **Object Detection**: This step locates the object in question in the image.

3. **Feature Extraction**: Here, key features of an image can be detected and extracted using statistical or deep learning methods. Features are used to differentiate images of different classes and relate images of the same class. This works on the basis that an image is represented by a matrix of pixels that can be mathematically manipulated and understood.

4. **Classification**: This is the key stage, where the inputted image is classified into one of the possible classes by comparing the features of the image to the data that the model knows about each class.

2.5.3.1 Classification Techniques

There is a variety of techniques that can be used for the machine learning example that is image recognition. This section will detail the most commonly used and result in a comparison and selection of the method to be used in this project.

2.5.3.1.1 Support Vector Machines

Support Vector Machines, or SVM, is a supervised learning technique that is generally used for binary (two-class) problems. However, it can also be used for multi-class problems, as required by this project. Supervised learning is where input data is classified into one of a set of known classes. In contrast, unsupervised learning is where no classes are known. To fully understand how SVM works, it is important to introduce a few definitions:

- **Support Vectors**: these are the data points describing the data that are closest to the hyperplane.
- **Hyperplane**: this is a decision plane used to seperate objects belonging to different classes
- **Margin**: given the closest object on each side of the hyperplane, this is the distance between these two objects.

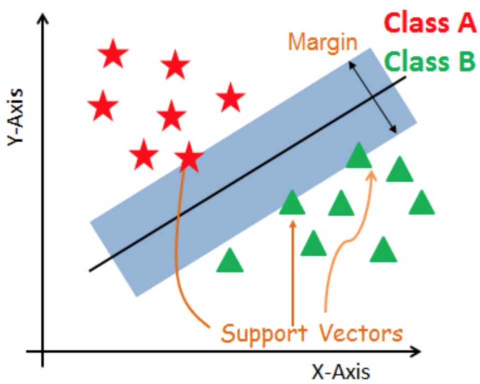


Figure 2.28: Support Vector Machines Components [127]

SVM works to choose a hyperplane such that the margin between different classes is maximised. First, multiple hyperplanes are generated, before the one producing the greatest intra-class separation is selected. Figure 2.29 illustrates the generation of multiple hyperplanes performing different splits of the data. In this example, the black hyperplane would have been selected.

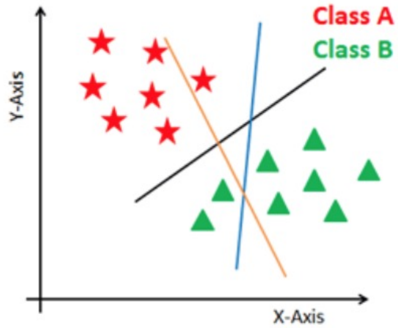
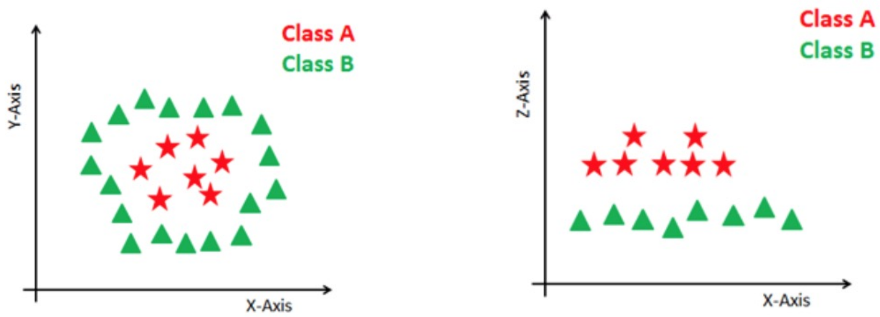


Figure 2.29: Support Vector Machines - Multiple Hyperplanes [127]

Clearly, in situations where the split between classes does not follow a linear pattern, this approach is not appropriate. At this point, SVM "uses a kernel trick to transform the input space to a higher dimensional space" [127] to produce a linear separation. This transformation can be seen in Figure 2.30, with the non-linear separation shown in Figure 2.30a and the projection to a higher dimensional space in Figure 2.30b.



(a) Before Transformation

(b) After Transformation

Figure 2.30: SVM Kernel Trick

Depending on the type of non-separability seen before the transformation, different kernels are used to perform the transformation. For example, Figure 2.30a shows a radial separation between classes. Therefore, the Radial Basis Function (RBF) Kernel would be selected.

2.5.3.1.2 Random Forests

Random forests is another example of a supervised learning technique, often selected for its simplicity and flexibility. The algorithm is based on Decision Trees - these are heirarchical tree-like structures based on a series of yes/no questions about whether the data has certain features, an example of which can be seen in Figure 2.31.

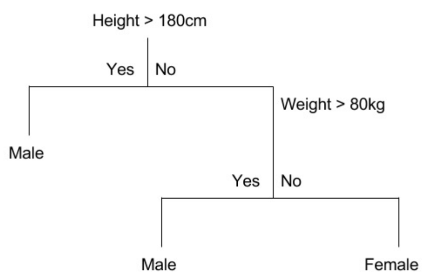


Figure 2.31: An Example Decision Tree [128]

The algorithm works in 4 steps, and can be visualised in Figure 2.32.

1. Select multiple random samples from the dataset
2. For each sample, construct a decision tree
3. Feed the test set through each decision tree and obtain a prediction for each image
4. Select the most commonly occurring prediction for each image (majority voting)

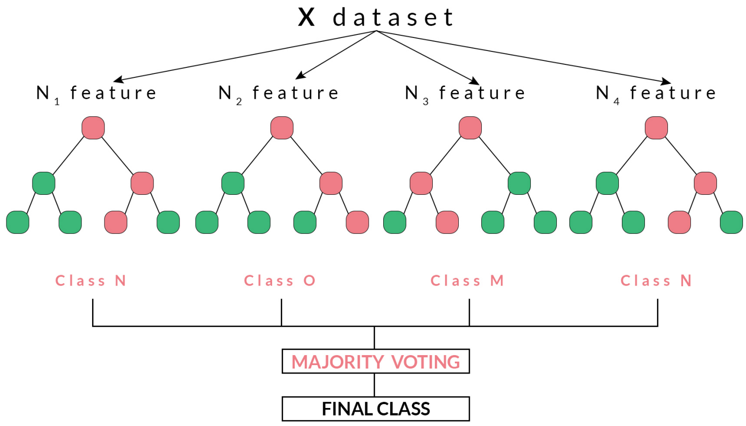


Figure 2.32: Random Forests Algorithm [129]

2.5.3.1.3 Convolutional Neural Networks

Convolutional Neural Networks, or CNN, are a special type of Neural Network. Neural Networks, modelled on the neural networks found in the brain. These are therefore networks of connected nodes, or 'neurons', that each perform a mathematical calculation on the data passed through the network.

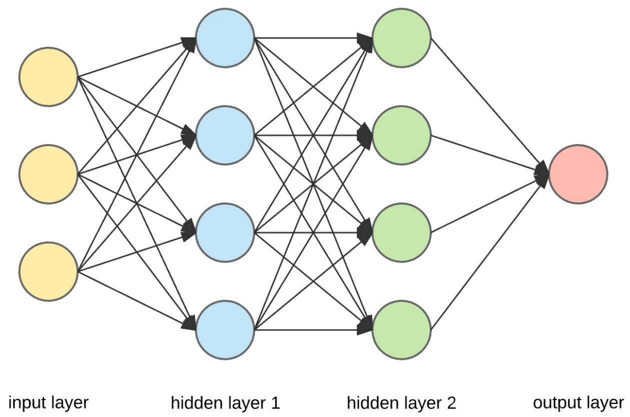


Figure 2.33: Example of a Neural Network [130]

The input to a CNN model is always a feature map. This a 3D matrix, where the first two dimensions are the length and width of the input images, and the 3rd dimension represents the number of colour image channels (3). Convolutional Neural Networks are a special architecture composed of multiple layers of these artificial Neural Networks, each of which performs three operations.

Operation 1: Convolution
Mathematically, convolution is "the summation of the element-wise product of two matrices" [131]. This operation works by 'sliding' a filter over the entire input feature map and performing convolution at each location to extract features about the image. Using different filters can extract different features, such as the filter seen in Figure 2.34, used for vertical edge detection.

1	0	-1
1	0	-1
1	0	-1

Figure 2.34: Vertical Edge Detection Convolution Filter

The output of the convolution at each location of an $N * N * 3$ array will produce an output $N * N$ array, as seen in Figure 2.35.

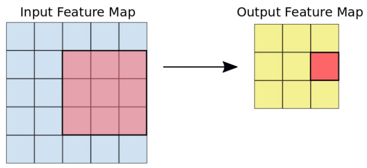


Figure 2.35: Convolution [132]

Operation 2: ReLU
After each convolution operation, a Rectified Linear Unit, or ReLU, transformation is applied to the feature:

$$F(x) = max(0, x) \tag{2.5}$$

This allows the model to account for non-linearities.

Operation 3: Pooling
The next layer calculates an *aggregate statistic* (such as the maximum, or average) to reduce the dimensions of the feature map and make the result invariant to transformations. For example, if the chosen aggregate statistic is the maximum, this operation works by again 'sliding' over the inputted feature map and taking the largest value found at that location. Therefore, the size of the pooling filter will directly impact the size of the outputted matrix.

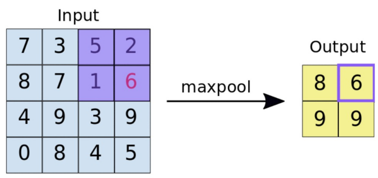


Figure 2.36: Pooling [132]

The output of the pooling operation is flattened out into a column vector. The column vectors formed from each iteration of the convolution-relu-pooling process, form a final, fully connected layer. This is where "every node in the first layer is connected to every node in the second layer" [132]. The final fully connected layer is typically a softmax activation function, which outputs a probability that the image belongs to each class. The class with the highest probability is then selected as the class that the input image belongs to. An example of a full CNN model can be seen in Figure 2.37.

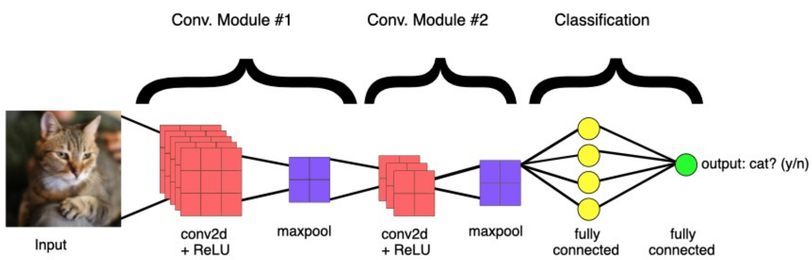


Figure 2.37: Example Convolutional Neural Network

2.5.3.2 Comparison

The 3 image classification techniques discussed above have their own advantages and disadvantages that will define their suitability for different projects. These will be introduced and discussed in this section to decide which method is most suitable for this project.

Characteristic	SVM	Random Forests	CNN
Accuracy	Average	High - due to the number of decision trees	High - due to multiple layers
Overfitting	Not susceptible - has a regularization feature	Not susceptible - predictions are averaged to remove bias	Susceptible - can be prevented by adding more data, using data augmentation and reducing architecture complexity
Classification Speed	Fast	Slow - caused by multiple decision trees	Fast
Computational Cost	High	High	High - but minimised compared to regular neural networks
Other Requirements	Feature scaling	None	Feature scaling, Large Dataset

Table 2.7: Image Classification Techniques Comparison

The most important characteristics of those above are Accuracy, Classification Speed and Computational Cost. Based on accuracy, Random Forests or CNN should be chosen. Based on classification speed, SVM or CNN should be chosen. Therefore, convolutional neural networks appears to be the best choice. However further exploratory will be done later in the project to ensure that this is in fact the best design decision.

2.5.4 Performance Evaluation

Evaluation of machine learning models is essential to understand performance, and check for common pitfalls such as overfitting. The most common metric to quantify the success of a classification problem is the **accuracy**. This is defined as the percentage of correct classifications made out of all data.

$$\text{Accuracy} = \frac{\text{Number of correct classifications}}{\text{Total number of classifications}} * 100 \tag{2.6}$$

This can be calculated for both the training and testing subsets of a data set, providing information on how well the model performs on seen and unseen data respectively.

The accuracy looks at the performance of the model over the entire data set. However, performance on specific classes can also be evaluated, using **precision** and **recall**. The precision is defined as the percentage of correct classifications out of all predictions made for a certain class.

$$\text{Precision} = \frac{\text{Number of times a class was predicted correctly}}{\text{Total number of times the class was predicted}} * 100 \tag{2.7}$$

In contrast, the recall is the percentage of class members that were classified correctly.

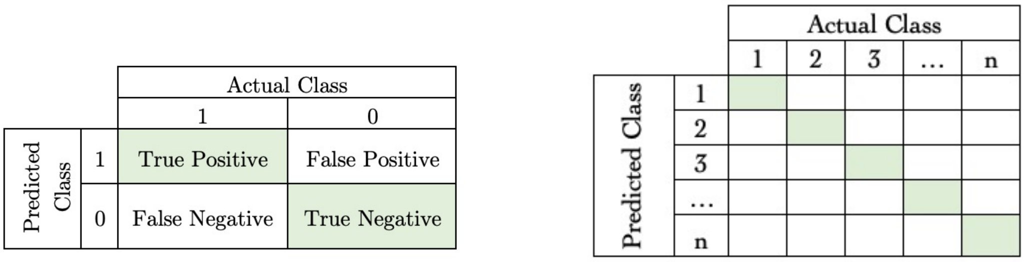
$$\text{Recall} = \frac{\text{Number of times a class was predicted correctly}}{\text{Total number of images belonging to the class}} * 100 \tag{2.8}$$

From here, the **F1-score** can be defined. This is the *harmonic mean* of the precision and recall for a given class. This metric causes lower numbers to have a greater negative effect on the result than higher numbers have a positive effect. This metric can therefore "measure the effectiveness of identification when just as much importance is given to recall as to precision" [133].

$$\text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.9}$$

The overall aim is to maximise all of these metrics. However, the precision tends to *decrease* as the recall *increases*, meaning that a trade-off between the two must often be made. However, maximising the F1-score can help to alleviate this problem.

To further understand the performance of a machine learning model, one can consider a confusion matrix, which analyses and visualises the false positives and negatives alongside the true positives and negatives.



(a) Binary-class Models

(b) Multi-class Models

Figure 2.38: Confusion Matrices for Different Types of Models

Figure 2.38a shows the structure of confusion matrix for a binary classification model, and Figure 2.38b for a multi-class classificaion model. It can clearly be seen that multi-class classification models produce a larger, more complicated matrix. For both matrices, the vertical axis represents the predicted class for each input image, and horizontal axis represents the actual class. Therefore, for N classes, the matrix intuitively has a dimension of NxN. Each time an image is fed into the model, 1 is added to the cell with the row corresponding to the class predicted for that image and the column corresponding to the image's actual class. Therefore the diagonal, as marked in green on Figure 2.38b, represents the correct predictions. The larger the numbers on the diagonal, and the smaller the numbers away from the model, the better the performance of the model.

An advantage of the confusion matrix is that it is a useful and easy-to-interpret way of assessing a model's performance, even when classes are unbalanced within the data set. This is beneficial, as accuracy can often be calculated as high, even when the model is not making useful predictions.

2.5.5 Summary

This section has introduced and given an initial insight to the area of Machine Learning. Whilst this topic is not the focus of this project, it is still a featuring component, and therefore having at least a high level understanding is important when reading about design decisions, as well as implementation and testing later on in this report.

With this in mind, this section has highlighted the fact that there are many stages to machine learning that must be carefully considered and designed in order to achieve the optimal performance when implementing a Machine Learning model. This starts with the choice of dataset, which could be a pre-built one that is perhaps already well-established in the literature, or a custom one for an application that is yet to establish a state of the art. From here, data pre-processing may need to be performed, but this will vary according to both the dataset and model choices. For example, deep learning techniques, such as neural networks, have been proven to perform best with large amounts of data. Therefore, many datasets may require pre-processing techniques, for example, data augmentation, in order to satisfy this requirement. However, when building a model, the designer should always consider multiple pre-processing techniques, architectures and hyper parameter values in order to find the optimal setup for their application. This setup can then be validated by comparing metrics such as the F1 score, and the generated confusion matrix.

2.6 Related Work

The previous sections of this chapter introduced the areas of background research relevant to this project. To further expand on this, this section introduces and provides a literary survey of related work, from which further understanding of the problem specification and potential solutions can be gained.

2.6.1 Waste Classification

When considering the problem of classification of waste for assessing whether it can be recycled, the decision of "recyclable" or "not" intuitively depends on the material of the object. Therefore, classification for recyclability status can be performed in one of two ways; (1) detect the object itself, and from there, infer the material, or (2) recognise the material directly.

2.6.1.1 Classification Based on Materials

To this effect, Liu et al [134] implemented a Bayesian Framework to "combine (the pool of features) into an effective material recognition system". Using the Flickr Materials Database [135], which consists of 1000 images equally split between ten material categories, this approach "sampled color, jet, SIFT, micro-jet and micro-SIFT features on a coarse grid", before "running an aLDA algorithm to select features". The recognition accuracy achieved was 44.6%, demonstrating that the introduction of aLDA not only provided a performance increase of 7.2%, but also the approach provided a 20.8% performance increase over the state of the art [136]. However, these low accuracy levels highlight the difficulty of detecting the material of an object based on vision alone. Often, to correctly infer the material of an object, other senses, such as touch and sound may come into play. This suggests that this approach may not always be suitable for waste classification applications, such as those where only vision is available, due to the difficulty of achieving sufficient accuracy.

2.6.1.2 Classification Based on Objects

In contrast, multiple papers have been completed in the area of waste classification based on object detection, and inferring the material from there. For example, in 2016, Yang and Thung [137] found that the Flickr Material Database used in [134] "does not accurately represent the state of recycled goods", so instead built a dataset of 2,400 images across 6 classes, titled TrashNet. This paper then compared the ability of CNN and SVM models for classification for recyclability status, achieving 63% accuracy with SVM, compared to only 22% with CNN. This highlights the fact that CNN requires more data to be able to provide high accuracy. It was also pointed out that there is a large variety of possible objects "that can be classified into one of the waste of categories", highlighting the need for a "large and continuously growing data source" in order to achieve highly accurate systems.

The same year, Sakr et al [138] also compared the implementation of Convolutional Neural Networks (CNN)[9] and Support Vector machines (SVM) to classify waste into 3 categories - plastic, paper and metal. This paper provided an improvement of the performance seen in the work by Yang and Thung, with a 94.8% classification accuracy achieved by SVM, and 83% when implementing CNN. The difference in performance between SVM and CNN was suggested to be caused by the the limitation the GPU introduced on batch size used for the CNN, further supporting the discoveries of Yang and Thung, and postulating that "more images and more GPU memory in the future will favour CNN", as it will reduce overfitting.

The next year, Awe et al [139] proposed a Faster R-CNN approach, "to get region proposals and classify objects" from the TrashNet dataset into three categories; recyclable, landfill, and paper. Whilst this classification is not specific to recycling, nor were the final accuracy results discussed, this model achieved a mean average precision (mAP) of 0.683.

In 2018, Aral et al [140] then modelled the performance of various convolutional network architectures, again on the TrashNet dataset. The architectures selected were eXception, MobileNet, DenseNets and Inception V4, achieving accuracies of 82%, 84%, 84% and 89% respectively. With further tuning of various hyperparameters, this paper went on to achieve a test accuracy of 95% using the DenseNet model, concluding that "deep learning algorithms can be used to classify recyclable waste". However, the paper also highlighted the point that real time systems often don't achieve as high accuracy, "due to the relatively small amount of data, and the white background of the images".

[9] Specifically AlexNet

Later in the same year, Bircanoğlu et al [141] developed RecycleNet, a "carefully optimized deep convolutional neural network architecture for classification of selected recyclable object classes". Also utilising the TrashNet dataset, accuracies achieved during the experimentation discussed in the paper ranged from 75% using ResNet50, to 95% using DenseNet121, with the final result of the RecycleNet model achieving 81% accuracy in 352ms for 200 Epochs. However, the paper also observed that sometimes, during the process of recycling waste, materials such as plastic bottles can get compressed or crushed. During this process, they "do not lose their material properties, but (do) lose their key properties to be identified as an intact object". This "requires the system to generalize extremely well when trained with a relatively small training set". However, the results were seen as able to prove that "meticulously training" of CNNs can provide "industrial-grade results to solve these types of problems".

Where the previous papers had only considered CNN and SVM approaches, later in 2018, Satvilkar [142] also provided an insight into other techniques; Random Forest (RF), eXtreme Gradient Boosting (XGB) and K Nearest Neighbours (KNN) for classification of the TrashNet dataset. In contrast to earlier papers, Satvilkar actually found that, post data pre-processing, CNN produced the highest accuracy, at 89%, compared to 65.67% for SVM, 70.1% for XGB, 62.61% for RF and only 52.5% for KNN. Other metrics considered where sensitivity, specificity, prevision and recall, for which CNN also consistently performed the best.

Finally, last year, Özkaya and Seyfi [143] "developed a deep learning application which detects (and classifies) types of garbage into trash, in order to provide recyclability with (a) vision system". Again using the TrashNet dataset, seemingly a common occurence in the related literature, an accuracy of 90% was reached when using a convolutional neural network known as VGG16, and a Softmax classifier. The paper observed that this accuracy could be further increased to 97.46% when swapping the Softmax classifier for an SVM. With this classifier, the paper then achieved an even higher accuracy of 97.86% with an alternative CNN architecture, GoogleNet. This paper therefore achieved the highest classification accuracy of any paper discussed so far, *without* any data augmentation.

2.6.1.3 Recycling Apps

Most relevant to this project is perhaps when such classification techniques are implemented in a smartphone app. An example of this is SpotGarbage, an Android app developed by Mittal et al [144], that utilises a CNN, GarbNet, to "automatically detect and localize garbace in unconstrained real-world images". The work built a dataset named the Garbage in Images (GINI) dataset, a "collection of several in-the-wild images containing garbage", using the Bing Image Search API[145], and then used this to train the network. This resulted in an accuracy of 87.69% ± 0.93 and a computation time of 1.50s with a Fully Convolutional network (FCN) without Local Response Normalization (LRN), and 87.70% ± 1.67 in 4.11s *with* LRN. They observed that these Deep Learning methods "considerably outperform the approach relying on image processing" by 7%. Furthermore, their FCN approach is "11 times faster than naïve sliding window based CNN", and "6 times faster (than) traditional image processing". However, this app is (i) only deployed for the Android platform and (ii) focuses on detecting rubbish in the streets, rather than classifying how to recycle an item.

The following apps are relevant apps found from exploration of app stores, and not research papers. To this effect, there is a limited amount of information that could be learnt about each one. However, after the introduction of each app, a competitive analysis was carried out. This can be seen in Table 2.8.

The first example is *Recycle Academy by Constant Click LLC* [146], which teaches users how to recycle their items by locating the recycling symbol on their packaging, and identifying this within the app. However, the symbol has to be of a specific type[10] for the app to be able to provide information about it. Furthermore, it does not provide information about where their recycling information is specific to, which can vary greatly according to location.

Along the same theme, *We Recycle by OPRL* [147] allows users to scan barcodes of items they wish to recycle. However, the app is still in 'phase 1', so only a small number of items are recognised by the system. Despite this, once the item has been identified, the app will provide information about how to recycle it. It also includes a reward scheme - the more you recycle, the more points and 'achievement badges' you can collect.

A further waste classification app *Recycle Wizard by Shuo Feng* [148], which helps users identify how to recycle their waste by offering a choice between a search function, or taking a photo, from which the item is identified using a Google API. The app then displays specific information about how to dispose of that item. However,

[10] As pictured in the fourth screenshot.

the disposal information is only specific to the City of Toronto guidelines.

The final example of an app that provides in-app classification of waste is *RUcycle* [149]. This is an Android app developed by a group of students for HackPrinceton Spring 2018 [150] that uses "image recognition to classify objects as trash or various types of recycling". It is able to categorise into recycling, waste and specialist recycling. Upon identifying an item that requires specialist treatment, such as a laptop, it includes a feature showing the user the nearest recycling centre for that item. However, due to it being a project developed just for a hackathon, it is not available on the Google Play Store.

An example of a recycling app without a classification feature is *Recycle! by Bebat Vzw* [151]. This provides information about recycling different materials, waste collections and collection points. However, its design means that, when it comes to identifying how to recycle items, the user has to be able to correctly identify the item's material, rather than the app offering a feature to do this for them. The app also only shows information for specific postcodes in Belgium.

A further example is *Recycle Smart* [152], which provides information on how to recycle items, displays nearby recycling events, and also offers a feature for users to arrange a pick up of specialist items. If the user does not know the material of their item, they can also search directly for the item in a helpful search bar. Unfortunately, the app only provides recycling information specific to Australia.

Recycle it! by Tobias Wiedow [153] is a useful app for finding nearby collection points based on the user's current location. However, it does not allow users to search for specific items. Furthermore, it only provides information for collection points in Germany.

Finally, *Surrey Recycles by Surrey County Council* [154] allows residents of Surrey to search for items that they wish to recycle. It then shows, with a helpful map, the relevant information about recycling the item. The app even provides further information about recycling centres, including opening hours and directions from the user's location. For more complicated items (i.e. where only parts can be recycled), the app will also provide instructions on this. For example, for a crisp tube: *"Crisp tubes have a thin layer of foil inside which means they cannot be recycled. Plastic lids can be put in your recycling container"*. Unfortunately, this app is not able to provide information about recycling centres in the rest of the UK. How the item specific information about how to recycle will generally still apply.

These apps[11] can be compared and summarised as seen in Table 2.8. Here, features such as where and how long the apps have been available, as well as features and ratings can be compared and assessed. The results, especially the customer reviews, can be used later on when defining the specification of this project.

[11] with the exception of RUcycle, as it is not present on an App Store.

Competitor	**Recycle Academy**	**We Recycle**	**Recycle!**	**Recycle Smart**	**Recycle Wizard**	**Recycle It!**	**Surrey Recycles**
Category	Education	Lifestyle	Reference	Education	Reference	Lifestyle	Reference
Platforms	iOS	iOS	iOS, Android	iOS, Android	iOS	iOS	iOS, Android
Time on Market	1 year	1.5 years	6 years	3 years	2 years	4 years	3 years
Main Features	Contains list of recycling symbols that you match up with your product. Under the symbol it tells you which bin this belongs to.	Scan barcodes to identify item. Collect points as you recycle. Recycling Information displayed.	Keep track of rubbish collections. Search by item. Find nearby collection points.	Find out how to recycle by item type. Find nearby recycling facilities. Report issues. Keep track of waste collections & book pickups.	Identify item by taking a photo. Can also search by keyword. Find out how to recycle item.	Shows nearby collection points based on location.	Provides postcode-specific information. Search for item by keyword.
Issues	Only provides information about one type of symbol. Does not provide location information. Users have to scroll through all symbols to find the one they're looking for as there is not a search option.	No longer available on the app store. Only a few barcodes are stored in their database so only a small number of items are recognised by their system.	Only shows information specific to Belgium. The user needs to know what the item is/made of.	Only shows information specific to Australia. The user needs to know what the item is/made of.	Only shows information specific to the city of Toronto. Poor UI.	Cannot search how to recycle an item. Only shows information for collection points in Germany.	Only shows information specific to the county of Surrey.
Customer Ratings (/5)	3.0	4.5	1.0	No ratings	No ratings	No ratings	4.4
Customer Review Insights	Good Idea. Scrolling through all the symbols is arduous. A search function would be useful.	No reviews	Buggy. Nothing shows in the calendar(only one rate and review)	No reviews	No reviews	No reviews	Helpful app. Need to know the right keyword. Should provide what day bins go out. UI Colour choices make it hard to read.

Table 2.8: Competitive Analysis of Recycling Apps

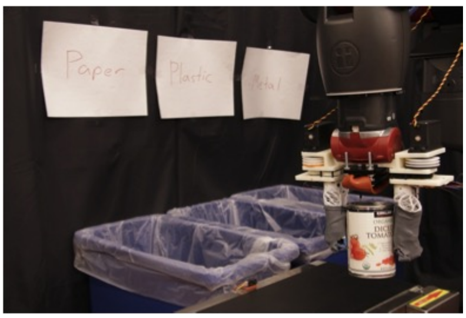
2.6.1.4 Recycling Technology

An alternative method to those already discussed is the development of technology to classify and separate recycling and waste post-collection. With the expansion of Machine Learning and introduction of Deep Learning over the past decade, as well as the increased awareness of climate change, this area has seen a large number of technologies and inventions emerge.

An example of technology making use of computer vision and machine learning in this application is “TrashBot”, by the CleanRobotics Team [155]. This is “an autonomous system that uses robotics, computer vision and artificial intelligence to detect and separate landfill from recyclables” [155]. All a user needs to do is place an item to be disposed of inside of the machine, and the machine will perform all of the identification and sorting “more accurately than human beings”. Furthermore, “cloud connectivity allows individual units to learn from the global (TrashBot) fleet”, meaning that the detection system, and therefore the accuracy, is constantly improving over time.

Figure 2.39: TrashBot - Example Sort [155]

A further example is RoCycle, developed by a collaboration of students from MIT and Yale [156]. In contrast to TrashBot, RoCycle uses “compliant robotic grippers”[157] that identify whether the item is made of paper, plastic or metal, based on the strain and pressure capacitance of the material. However, as with TrashBot, RoCycle was designed to replace the current technology used in industrial waste sorting. The system managed to achieve “85% accuracy with a stationary gripper and 63% accuracy in a simulated recycling pipeline” [156].



(a) System in Action

Algorithm 1: Sorting Algorithm

```
Calibrate hand by recording sensor values upon open
  and close;
while True do
    Items move along conveyor belt;
    if IR breakbeam is broken then
        Conveyor stops;
        Baxter moves to break beam location and
          grasps object;
        Read strain / pressure sensors and normalize
          them based on calibrated open / close;
        Use linear regression on strain sensor values to
          estimate size;
        Calculate average and difference between
          pressure sensor readings, dividing by size;
        Sort via classifier to determine material type;
        Place object into appropriate bin;
    end
end
```

(b) Sorting Algorithm

Figure 2.40: RoCycle [156]

No analysis has been performed of the impact of either system when deployed into industrial applications, yet their tested performance rates would suggest that they will provide an improvement. Furthermore, introducing more automation into the industry will remove the risk of human error.

2.6.2 Behaviour Design

As discussed in Section 2.2.6, a large part of improving the recycling rate will be through motivation and persuasion of consumers to change their attitudes and behaviour. An example of this is how "We Recycle", as discussed in Section 2.6.1.3, offers points the more the user recycles. Further examples not specific to recycling and waste are Portia, "a user adapted persuasion system in the healthy eating domain" [158], and Playful bottle, "a mobile social persuasion system to motivate healthy water intake" [159].

Portia is "the argumentation module of a dialogue system" that, after acquiring information about its users, "informs them about various eating habits' advantages and disadvantages, provides suggestions in this domain, and tries to persuade them to gradually modify problem behaviours" [158].

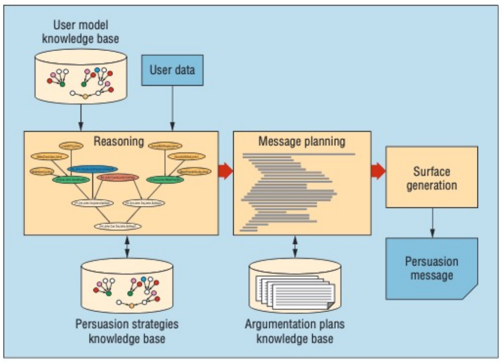


Figure 2.41: Portia System Architecture [158, pg. 44, fig. 1]

Figure 2.41 highlights the process that the system goes through to output messages to persuade the user, starting from initial assumptions made about the user. The system is capable of "suggesting lines of action" for the user to follow, and "tries to persuade the user to follow them when needed, (entering) into an argumentative sub-dialogue to justify and support its choices or revise them if needed" [158]. However, the paper does not go into details about the success of the project and therefore the effectiveness of this approach cannot be assessed.

Secondly, *Playful Bottle* is a "mobile persuasion system" that "makes use of a mobile phone attached to an everyday drinking mug and motivates office workers to drink healthy quantities of water" through implementation of two 'hydration games' [159]. The study assessed the outcome of two hypotheses:

1. "How effective are the hydration games for improving the water behaviours of users?"
2. "What aspects of water drinking behaviour were effected by the uses of the Hydration games?"

These were evaluated through testing of 16 members of university hospital staff over 7 weeks [159]. The results can be seen in Figure 2.42 and show that the implementation of both hydration games *increased* the water intake of the subjects being tested. This suggests that both gamification and a desire to meet social expectation (as the subjects could see the performance of their fellow study participants) can motivate people to change their behaviours.

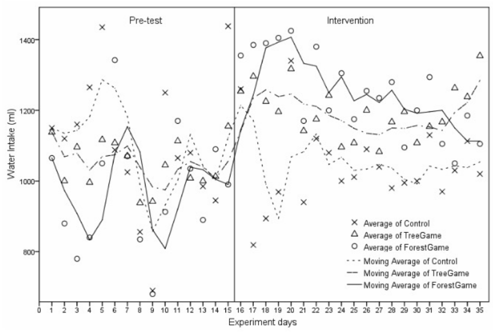


Figure 2.42: Playful Bottle Experiment Results [159]

A final example is "ClassApp: A Motivational Course-level App" [160], a "persuasive mobile application for engaging students and promoting learning using various persuasive strategies". This app utilised the psychological principals of social comparison and social motivation, as discussed in Section 2.4.3.2. However, the paper did not assess the effect that this application had on students in order to evaluate its effectiveness.

2.6.3 Summary

This section has introduced some of the most recent work related to this project, in order to fully appreciate what is currently the state of the art, and what foundation has been laid for this project to build on. In fact, it has highlighted that, whilst a wide variety of work has been done in the separate sections of waste classification, recycling apps, and persuasive apps, little work has been done that is a combination of the areas, which is a gap that this project aims to fill.

This project therefore aims to build on the waste classification work detailed in Section 2.6.1, but in a manner that incorporates behaviour design in order to *motivate* users to improve their recycling performance. Furthermore, it should utilise the concept of recycling apps that aims to *educate* users, in order to result in a final deliverable that is both a *motivational and educational* smartphone application.

Chapter 3

Project Specification

3.1 Introduction

As per the project title, the high-level aim of this project was to produce a *persuasive* smartphone app for *improving* recycling performance. With the relevant areas of background research now introduced and discussed, enough information is present to expand this title into a full product specification. This chapter therefore aims define a comprehensive project specification that will facilitate efficient project planning and implementation, as well as ultimately allow for clear measurement and evaluation of the success of this project.

3.2 Requirement Capture

When defining a project specification, it is important to ensure that the project requirements have been fully researched so as to fully capture them in the project brief. This is especially important in user-focused projects such as this one, as operational functionality is no longer the only concern - user requirements need to also be captured. To this effect, the requirements of this project can be split into two key categories; choice of platform development / development tools and user requirements, and will be discussed in the following sections.

3.2.1 Choice of Platform Development

3.2.1.1 Web vs Native Mobile Applications

In 2017, it was estimated that 78% of adults personally use a smartphone, with the average user spending 2 hours 49 minutes per day using their phone (a 24% increase since 2015) [161]. Furthermore, it has also been found that mobile users spend 2x longer on their devices than desktop users [162]. Therefore, it can be seen that the best choice for an application would be a mobile based one; either on a mobile web browser, or natively on the device itself. A simple but effective way to choose between a native mobile and web application is by comparing their advantages and disadvantages, as seen in Table 3.1.

Web App	**Native Mobile App**
Adapts to the device.	Needs to be designed with a specific device in mind.
Code is the same regardless of device.	Different devices require different programming languages.
Do not need downloading or installing.	Has to be downloaded / installed.
Do not require App Store approval.	Require App Store approval.
Do not work offline.	Work offline.
Do not have access to/difficult to access hardware features.	Easy access to hardware features such as the camera and greater processing power.
Slower opeartion.	Faster operation.
Not always guaranteed to be secure.	Offer greater security.
Need good SEO[1] to be discovered.	Easily discovered on App Stores.
Good UI and UX hard to obtain due to having to search for the website etc.	Easy to achieve good UI/UX.

Table 3.1: Web vs Mobile Applications

From here, the best choice depends on what the priorities and main features of the application in question are. In this case, as the aim of the project is to educate and motivate consumers to improve their recycling behaviour. The persuasive nature of the application means that a key priority will be convenience and ease of use for the user. Therefore, the most important concepts will be accessibility, performance and experience.

Feature	**Best Choice**	**Reason(s)**
Accessibility	Mobile	Whilst both choices are accessible directly on the device, a web app will require the user to first access their web browser app, and then the extra step of navigating to the web application. Furthermore, web applications can only be used when the device is online, whereas native mobile applications do not require an internet connection to work.
Performance	Mobile	To prevent user frustration, the application needs to run fast and efficiently. Native mobile applications often perform faster than web applications, and also have *full* access to the device's hardware component whereas web applications often only have *limited* access.
Experience	Mobile	Whilst often down to preference, it is generally accepted that a native mobile app offers a better UI and UX, due to the nature of being enclosed in an on-device app rather than just being displayed in a browser. Furthermore, native apps allow features such as push notifications and user-specific data to be stored to really personalise, and therefore improve, the experience.

Table 3.2: Key Features and Best Platform Choice

3.2.1.2 iOS vs Android Development

Table 3.2 clearly shows that a mobile application would be the best choice for this project. However, a choice now needs to be made regarding which operating system to develop for. The two major operating systems available today are Android and iOS. To make an informed decision about which to use, several factors have to be considered, including, but not limited to, the number of users, ease of development and available features.

The choices of mobile operating system based on various factors, as introduced in Section 2.3, have been summarised in Table 3.3.

Factor	**iOS**	**Android**	**Choice**
User Base	48.71% of UK Market Share	50.8% of UK Market Share	Either[2]
Development Hardware	Requires a Mac	No restrictions	Either
Development Software	Objective-C or Swift	Java based	iOS
Device Fragmentation	Low - 19 different models	High - 24,000+ different models	iOS
Software Fragmentation	Low - 50%+ of devices have latest OS	High - 21% of devices have latest OS	iOS
Development Time & Cost	Average	Long testing time due to device fragmentation	iOS

Table 3.3: Comparison of Mobile Operating Systems

Analysis of this comparison highlights the fact that it makes sense to produce a native iOS application.

[2] Whilst 50.8% > 48.71%, the size of the UK market makes this 2.09% difference indistinguishable.

3.2.2 User Requirements

Due to the user-centered nature of this project, this project specification will be influenced by feedback from the target audience, to ensure that their needs are captured. An initial task was therefore to carry out an "initial insights" survey, the structure of which can be seen in Figure 3.1.

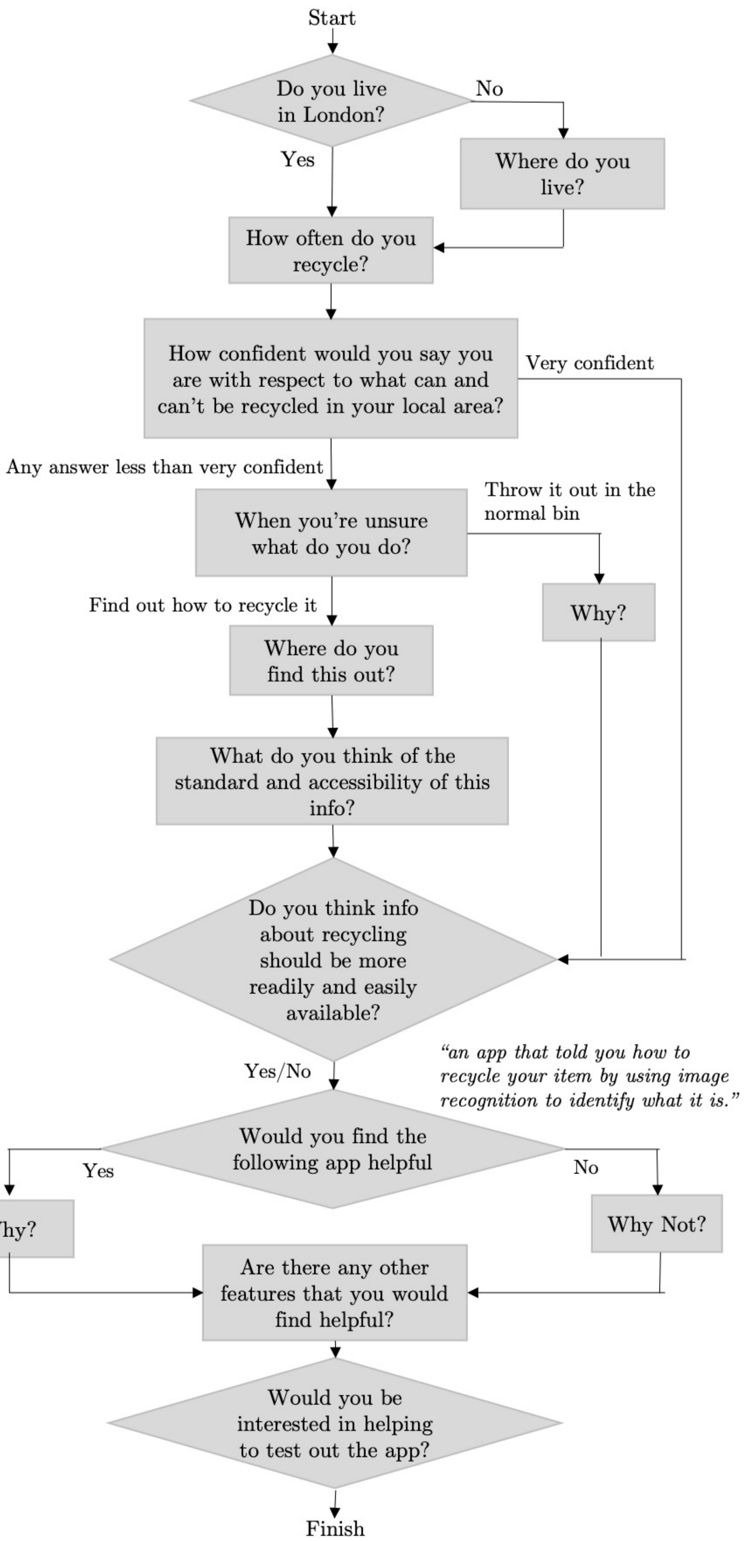


Figure 3.1: Initial Insights Survey Flowchart

The full results of this survey can be found in Appendix A. However, key results can be seen in Figure 3.2.

(a) Question 3

(b) Question 4

(c) Question 5

Figure 3.2: Key Results from the Initial Insights Survey

Figure 3.2 highlights that, whilst the majority of survey participants recycle regularly and know how to recycle 'standard items', when they are unsure, they often just throw the item out rather than find out what to do with it. This was self-diagnosed by participants as due to "laziness" and the fact that "it's easier than finding out what to do with it".

Furthermore, the 23.4% that answered "Recycle it anyway and hope it's recyclable" to Question 5 indicates a different issue, as, as already mentioned, attempting to recycle items that are not recyclable can lead to contamination and rejection of the entire bag of recycling collected. However, it was discovered that 66.7% of the 23.4% that answered in this manner were not actually aware of this fact. This suggests that, with further education about recycling, the majority of contamination could be prevented. In contrast, the remaining 33.3% who knew this risk but did it anyway will require less *education* and more *motivation* to recycle correctly.

Continuing the analysis of Question 5, of the final 20.3% that answered "Find out to recycle", 69% reported that they turn to the internet to find out further information. However, they observed that the method is "time consuming" and that the information they find is "not specific to (my) their area", "inaccurate" and "helpful for common items but often unclear/limited for unusual items". They went on to comment that they "would prefer if there was a straightforward and clear way to find out".

3.2.3 Summary

The initial findings suggested that a simple, accurate and easily accessible tool in the form of an iPhone app should be developed, to help users identify how and when to recycle. It was found that this would be well received with those who already have the motivation to recycle correctly. However, to maximise the potential impact of such a solution, a focus will be placed on *how* to motivate and educate will be required to impact those who do not currently care if they are recycling correctly.

3.3 Project Brief

3.3.1 Problem Statements

The combined research of literature and user attitudes highlighted three key areas, for which problem statements can be defined:

1. With such a variety of *(i)* products to be recycled, *(ii)* methods of recycling and *(iii)* what is accepted in different locations, it can be hard to keep track of if, how, where and when an item can be recycled.
2. It is hard and often time consuming to find out the information mentioned in problem 1, with different recycling methods and quality of information provided in different locations.
3. The general public lack the motivation to *(i)* recycle correctly, *(ii)* find out how to recycle correctly and/or to *(iii)* improve their recycling behaviour and performance. Generally, they either think that *(i)* "it's not my job", *(ii)* "I will not have much of an effect" and/or *(iii)* "I'm already doing my bit".

3.3.2 Project Goal

It has already been highlighted that the aim of this project is to produce an iPhone app as a tool for improving the motivation and education of consumers when it comes to recycling correctly. This can represented by the flow diagram seen in Figure 3.3.

Input	User presents an item that they wish to dispose of.
Process	1. App identifies item 2. From here, the material is identified or inferred 3. Recycling information for that item is obtained from the database
Output	App displays what the item is and the required steps to recycle it correctly at the user's specific location

Figure 3.3: Project System Diagram

3.3.3 Project Requirements

The problems outlined in Section 3.3.1 can each be summarised into three key deliverables; Waste Classification (Problem 1), Recycling Information (Problem 2) and User Motivation (Problem 3). From here, meeting these deliverables can be planned by outlining project requirements, as seen below.

3.3.3.1 Waste Classification

- The app will identify an item placed in front of the device camera by the user.
- This identification method will be accurate to ensure that users aren't taught how to recycle incorrectly.
- The app will provide at least one back up option for identifying the item in case the primary fails or incorrectly identifies the item. This could also be used if the user prefers the back up method.

3.3.3.2 Recycling Information

- Upon item identification, the app will display the relevant information; item name, whether it can be recycled, steps that need to be taken to recycle it, and where to recycle it.
- The item-specific recycling information will only be stored on the app in order if it takes up minimal resources. Otherwise, it should be stored off-device.
- The app will use location technology to determine the user's location and provide information specific to their location.

3.3.3.3 User Experience and Motivation

- The app will use psychological and persuasive techniques to motivate and influence the user to improve their recycling performance.
- The app will keep track of the user's recycling over time and show usage & impact statistics.
- The app will provide some sort of gamification or points-based system, or similar, to further motivate the user to improve their recycling performance.

- The app will be optimised for ease of use with a minimal, if not no navigational system

As the project develops, it is expected that these requirements will evolve to match updated goals and factor in any technical difficulties. However, should these goals be met on time with little to no technical difficulty, the project can be extended to include the following 'stretch goals':

3.3.3.4 Stretch Goals

- **Further Training of the Model**: The app will ask the user if the item was identified correctly to allow for continuous training of the identification.
- **Back-End Application for Local Authorities**: A back-end application will exist to allow other locations not currently included in the app to upload their recycling information and location. Upon admin approval, the app will then factor this information into its database to further extend the user base and area reach. A portal could also exist in this back-end application to enable the authorities of each location to log in to track usage stats and/or update information.
- **Alternative Sustainable Choices**: If the app detects repetitive scanning of the same item that has a more sustainable alternative (e.g. cardboard vs re-usable coffee cup) then it will display this information to the user and persuade them to make a more sustainable choice, in an effort to promote waste reduction. If this more sustainable choice is displayed, the app will also provide information about how much this more sustainable choice saves and what it will take to counteract the impact of production.
- **User Performance**: The app will have a sign-in feature and leaderboard to allow users to compare their performance and progress with friends. If a user has a good recycling performance over an extended period of time, the app will offer real-life rewards, such as discounts or vouchers from sustainable brands, to further incentivize users.

3.3.4 Design Features

With the project requirements now defined, they can be translated into the features to be implemented in the app. These features can be split into four categories by priority, as outlined by the 'MoSCoW' technique [163], namely must, should, could and would have features.

3.3.4.1 'Must Haves'

Intuitively, these features are the minimum amount of features that make the app usable with respect to the project goal. This means that they are essential to the app and its success. Therefore they take first priority when in the development stages.

Of the project requirements detailed in Section 3.3.3, it can be seen that the main 'must have' feature is a method of identification of the item the user wishes to dispose of. Without this, the subsequent features, such as the app displaying information about the item, will be rendered useless. Whilst this feature's requirements could initially be reduced to be just a search feature or similar, a computer-vision-based method of identification is defined as a 'must have'. This is because of the length of time and work required to build a image classification model that is accurate and reliable. Therefore, setting it as a 'must have' feature will ensure that enough time is scheduled to make sure that it is implemented on schedule. From here, it also makes sense to have the information about how to recycle the item to also be a 'must have' feature. This is important as, without this, users would not be able to use the app to learn how to recycle items. Finally, all of this information must be stored somewhere. Therefore, some form of database must exist as a (back-end) feature. However, at this stage, it is more important that the database *exists* rather than worrying about *where* this database stores the information (i.e. on-device or in the cloud).

3.3.4.2 'Should Haves'

These features are those that are important, but not necessary essential *initially.* I.e. whilst they *should* be included by the end of the project, they do not *have* to be implemented in the initial version. This means that, whilst they can be implemented in later versions of the app, they will still have an effect on the overall success of the project.

From here, the first key feature that will be classified as 'should have' is a back up method of identification, should (*i*) the computer vision method completely fail, (*ii*) the classification be incorrect, (*iii*) the user wishes to recycle a more unusual item that the model is not trained to detect, or (*iv*) the user prefer a different method. This will increase the system reliability and robustness, and also improve the UX for some users.

Furthermore, whilst location was chosen to *not* be a 'must have' feature and therefore not included in the first version of the app, it should definitely be included to make sure that the user is receiving the correct information for where they are. This is vital, as recycling techniques vary massively throughout the UK. A further extension from the 'must have' features is the database. It is now important to define here that, if it is large, it *should* be off-device, i.e. in the cloud, in order to conserve device resources. Finally, the app should have a simple yet aesthetically pleasing UI, in order to improve the user experience in-app, by making it easy to use. Furthermore, the UX should make use of psychological and persuasive techniques, to motivate the user. These will include those mentioned in Section 2.4, such as Push Notifications.

3.3.4.3 'Could Haves'

This group is for the features that do not fit in either of the 'must have' of 'should have' sections but are currently planned to be implemented. This means that, whilst they may be desirable, they are not essential - this group of features is also known as 'nice to have' features. They are generally features that "could improve the user experience or customer satisfaction, for (a) little development cost" [163]. This therefore means that their implementation depends on the remaining time and project resources once the 'must have' and 'should have' features have been implemented, and will be the first to go if these resources are limited or not available.

As already mentioned, these features will generally focus on user experience. 'Nice to have' features with respect to this therefore include a usage tracking feature that monitors the user's recycling over time and displays it in a simple UI. This could be extended to include a feature that allows users to compare with friends. Furthermore, the app could feature a gamified, points-based, or other achievement system that allows the user to compete with their past performance, or even with friends.

3.3.4.4 'Won't haves'

These features are those that will not be implemented in this project, but would be good to implement in future work should the project be carried on past the final deadline, or picked up again at a later date. This means that they are not planned into the development process. However, should the project run ahead of schedule, these features could be used to extend the project goal and improve the app further. Therefore, these features are those mentioned in the stretch goals section of the project requirements, Section 3.3.3.4.

3.4 Conclusion

This chapter has highlighted the process of capturing the project requirements, from which the project specification and goals were defined. An overview of the features mentioned in this chapter, ordered by their priority, can be seen in Table 3.4.

Type	Feature	Further Information
Must have	Waste Classification	Using computer vision
	Recycling Information	Specific to the identified item
	Database	For storing recycling information
Should have	Backup Identification Method	Search bar / barcode scanning
	Location-Specific Information	To ensure correct recycling regardless of location
	Off-Device Database	To conserve device resources
	Well-Designed UI	Easy to use and aesthetically pleasing
	Persuasive Techniques	To motivate the user
Could have	Usage Tracking	To monitor the user's performance over time
	Gamification and Rewards	As a motivation for improvement
	Compare with Friends	To add a social element
Would have	Continuous Model Training	To further improve the classification accuracy
	Back-End Application	To allow local authorities to add their recycling information
	Sustainable Alternatives	To encourage users to reduce waste

Table 3.4: App Features

The different types of features (must have, should have etc) can be used to structure the project into development stages and will be used as a comparison in the evaluation chapter, when the success of the projected will be assessed.

Chapter 4

Analysis and Design

This chapter introduces **RecycleHelper**, a self-contained iPhone app that utilises computer vision and persuasive techniques to not only *enable*, but also to *motivate* users to improve their recycling performance. The objective of this chapter is to develop the criteria specified in Section 3.2, as well as to introduce and justify the design decisions of the final app version.

4.1 Development Strategy

RecycleHelper was developed using an iterative development strategy, which is a continued process of planning, development and testing, as seen in Figure 4.1.

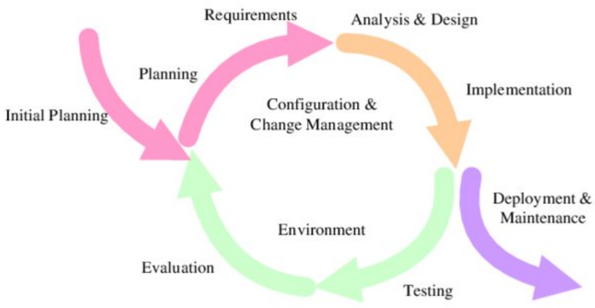


Figure 4.1: The Iterative Development Model [164, Fig. 2, pg. 2]

Figure 4.1 highlights a circular process of repeated planning, design, implementation, testing and evaluation. The process is finished when the latest app version satisfies the full project specification and requirements.

The process started by developing the minimum viable product, or MVP, that is designed based on a set of requirements smaller than the full project specification. In this project, this was the 'must have' features, as defined in Section 3.3.4. Once the MVP was implemented, an initial round of testing and evaluation was carried out, providing crucial feedback that was used to further develop the app design and implementation into the next, improved version, before testing it again. This cycle of developing and testing continued until the completion of the project. Each iteration involved a requirements capture process, where the evaluation of the previous stage was used to define a new set of features and goals for the next versions. At each stage, a decision also has to be made whether the next app version will develop the previously tested version, or if it was a 'fresh start'. This approach "can be likened to (the) mathematical method (of) successive approximation to arrive at a final solution" [165]. This process is also similar to the agile project management structure.

This approach was chosen because of the following advantages [165]:

1. Working software is developed early in the project life cycle
2. Small iterations allow for easy testing and debugging
3. Manages risk as "risky pieces are identified and handled during its iteration"
4. Iterations are "easily managed milestones"

Furthermore, this approach provides the developer with multiple 'chances to get the product right', and can gain invaluable insights from the feedback and results obtained from the multiple stages of testing, which results in a highly fine-tuned product at the end of the process. This in turn provides security that the final version of the app will have features whose usefulness has been verified by users, eliminating bias that may arise from the developer's own preferences. However, this approach can cause system architecture problems, as "not all requirements are gathered up front for the entire software life cycle" [165].

4.2 App Design

4.2.1 Architecture

RecycleHelper was developed using an architectural pattern widely used in industry, known as the Model-View-Controller, or MVC, pattern. As one might expect, this paradigm "separates an application into three main logical components: the model, the view and the controller, (each of which) are built to handle specific development aspects" [166]. The relationship between each of these components can be seen in Figure 4.2.

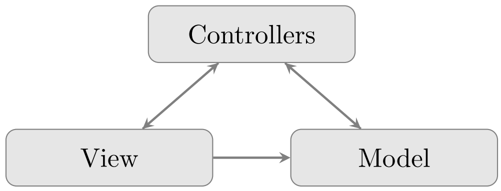


Figure 4.2: MVC Architectural Pattern

Utilising this design framework provides several key advantages. For starters, its modular nature means that it is easy for features to be added or removed without major alterations to the remaining code. This allows for faster development, especially considering the iterative development strategy used by this project. Furthermore, this structure allows for multiple developers to simultaneously work on an application, perhaps one focusing on each of the model, view and controller components, which will intuitively increase the development speed further. Also, separating the view components from the rest of the system allows for UI alterations, such as component placing and colour, to take plae without affecting the operation of the application.

4.2.1.1 Model

Model components are responsible for data transfer and retrieval. This data can be "data that is being transferred between the View and Controller components or any other business logic-related data" [166]. An example of this is when a user goes to log in to their account - their information will be retrieved from a database according to the log in details they entered, and then the relevant data will be rendered and displayed on screen.

In the context of RecycleHelper, a model component is used to retrieve and display the relevant recycling information and instructions when an item to be recycled is detected and classified. This information was obtained from `recyclenow.com`. Furthermore, model components were built to implement custom classes and structures.

4.2.1.2 View

View components form the UI of the application that the user interactions. When a user interacts with a view, data flows from View → Controller → Model, to perform the relevant actions and update the display. To this effect, each view will have a corresponding controller. However, multiple controllers may share the same model.

The app was designed using the XCode storyboard feature, which allows the UI to be designed by dragging and dropping components onto a visual representation of the app. Using this technique, each version of RecycleHelper was built using a combination of 7 unique view components:

1. Onboarding View
2. Home View
3. Settings View
4. Camera View
5. List View
6. Information View
7. Location View

Initially, each of these views were designed in the form of wireframes, a style of mock up that essentially only details the skeletal components of an application's UI. This allows a designer to have an idea of the layout, how components may interact, and what information they may display, without being too constrained to a particular design or theme. The wireframe designs for each of the views 1 to 5 can be seen from right to left respectively in Figure 4.3.

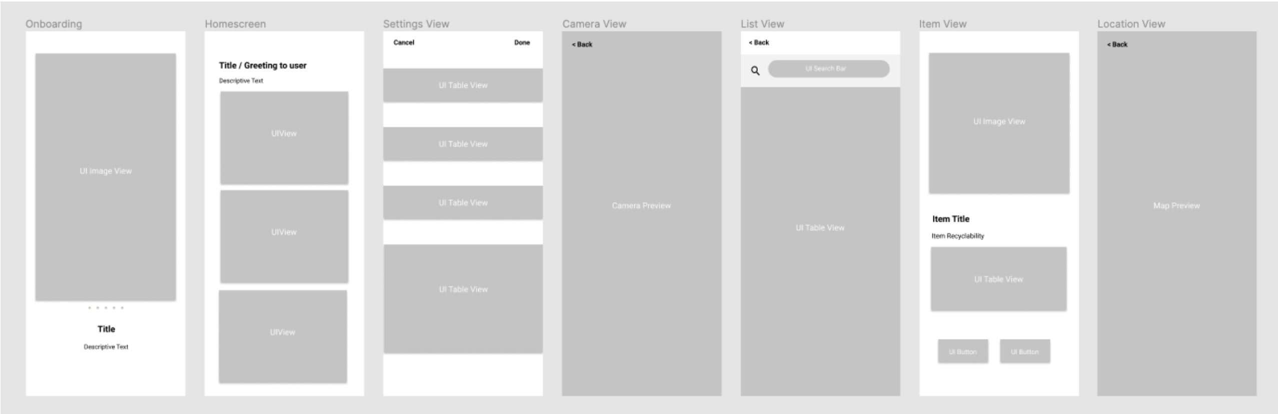


Figure 4.3: App Wireframes

The stages of UI design that form the final app will be discussed in the Implementation chapter.

4.2.1.3 Controller

Finally, the Controller components "interface between the model and view components" [166] and control any logic that goes between them. Essentially, controllers can be thought of the 'brains' of the app, that control its operation. When developing an iOS application in XCode, the operation of any view designed in the storyboard must be controlled by a ViewController.swift file. However, whilst every view must have an associated controller, not every controller must have an associated view. Instead, a heirarchy can be formed where view controllers can have sub-view controllers, equivalent to classes and sub classes in standard programming. This means that a sub-view controller will be of the type of its parent view controller, but can be called according to what operation is being performed by its parent view controller, in order to provide extra functionality.

4.2.2 Design Principles

The design principles utilised in the development of RecycleHelper can be split into two categories; UI principles to optimise the user experience, and psychological principles to implement behaviour design.

4.2.2.1 UI Design

Over the past 4 decades since the release of Apple's first product, the 'Apple Computer 1' [167], Apple has built a reputation for not only high quality, but also consistent, user interface design. This has caused Apple customers, both current and potential, to expect a certain standard of Apple products and any software that may be installed on them. To this effect, Apple expects the same from any third party developers that wish to release an app on the Apple App Store. To standardise this expectation, Apple released 'Human Interface Guidelines', that "set high expectations for quality and functionality" [168]. These guidelines provide advice and guidance around 3 themes and 6 design principles, as seen in Table 4.1.

Design Themes	**Design Principles**	
1. Clarity	1. Aesthetic Integrity	4. Feedback
2. Deference	2. Consistency	5. Metaphors
3. Depth	3. Direct Manipulation	6. User Control

Table 4.1: Apple Human Interface Guidelines

To optimise the user experience, RecycleHelper was therefore designed with these guidelines in mind. Details and specific examples for each of these themes and principles will be further discussed in the Implementation chapter.

4.2.2.2 Behaviour Design

As highlighted in Chapter 3, RecycleHelper was designed to be a *persuasive* smartphone app. This can be achieved through behaviour design to influence user behaviour, and was implemented in this project by designing four behaviour chains, as seen in Figures 4.4 to 4.7. The idea here is that the app only provides the ability or motivation to the user to perform the first step in the chain, and then they perform the rest on their own, with the final action being the goal of that particular behaviour chain. These chains are created by deciding on the goal behaviour, and working backwards from there.

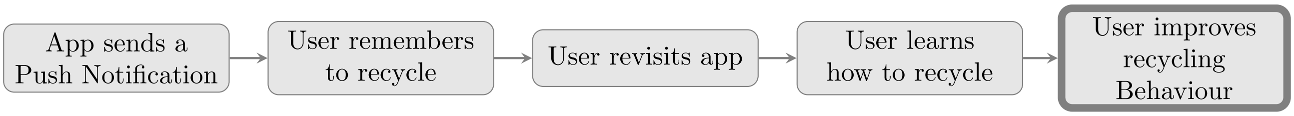


Figure 4.4: Behaviour Chain 1

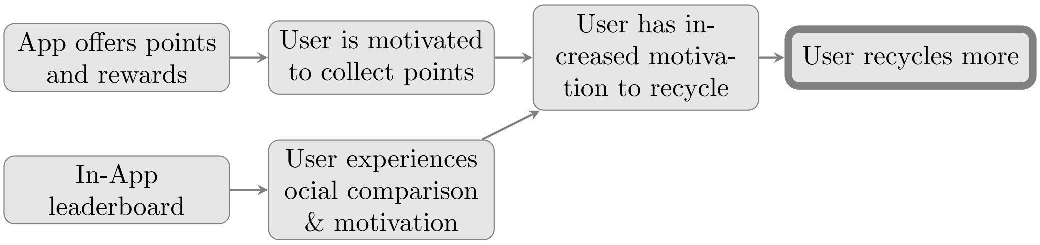


Figure 4.5: Behaviour Chain 2

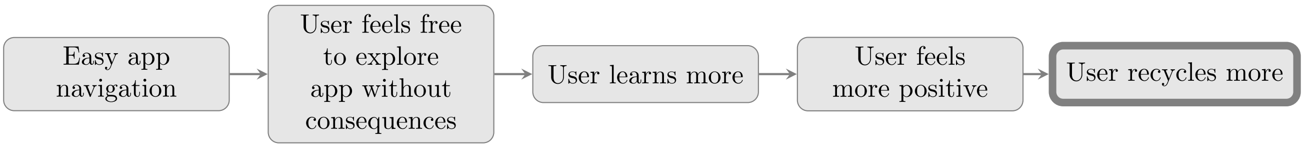


Figure 4.6: Behaviour Chain 3

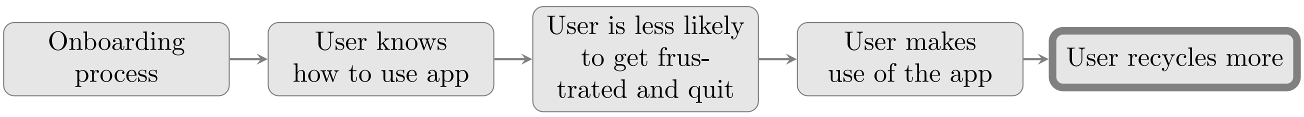


Figure 4.7: Behaviour Chain 4

These behaviour chains highlight that four key features that should be implemented in-app are push notifications, a points system and/or leaderboard or similar gamification technique, easy app navigation and an onboarding process.

Push notifications are an example of a prompt, as seen in the Fogg Behaviour Model discussed in Section 2.4.1. Furthermore, implementing a points system with leaderboard is an example of social comparison and motivation, as discussed in Section 2.4.3.2. Finally, easy app navigation, alongside with a clean and consitent UI is an example of following Apple's Human Interface guidelines.

4.2.3 Waste Classification

In order for the user to be able to identify *how* to recycle the item(s) in question, the app must first be able to identify *what* each item was. It was therefore deemed necessary, as set out in Section 3.2, to implement a machine learning model to perform the identification and classification of objects presented to the camera of the user's device. To improve the user's experience of the app, it was initially decided to have a back up classification method, where the user could use a text recognition feature to scan recycling labels found on packaging. This section outlines the process of selecting the most suitable machine learning method for each of these applications, in order to optimise performance.

4.2.3.1 Object Detection

The first step to assess the suitability of object detection techniques is to outline a series of criteria that a model should meet in order for it to be suitable for implementation.

1. Fast but accurate classification
2. Can be converted for use in an iOS application
3. Can provide accurate results when trained using the TrashNet dataset [137]

Criteria 1 was defined with the end use in mind - an iPhone application. This intuitively means that classification process must be not only accurate, but also fast, in order to allow for a positive user experience. If this criteria was not met, slow classification times and inaccurate identification would actually work *against* the entire aim of this project, i.e. instead of persuading users to improve their recycling performance, they would instead be dissuaded.

Criteria 2 was perhaps the most important of all criterions, as if the model selected does not conform to this specification, it would not be able to be used in-app. This is important to consider because coremltools, the python package built by apple to "integrate machine learning models into apps"[169], only works with a limited number of machine learning libraries, namely TensorFlow, Keras, Caffe, scikit-learn, LIBSVM, and XGBoost.

Criteria 3 was defined following research of the related work, as covered in Section 2.6. This section highlighted the high accuracy achieved by models trained using Thung and Yang's dataset. Following further resesearch of other pre-existing datasets that could be suitable for this application, as well as assessing the difficulty of manually building a second dataset, it was concluded that this dataset was the best choice for this project, due to having the most relevant categories, the largest number of images, and a range of research achieving high accuracy that suggested its suitability for a waste classification model.

4.2.3.2 Label Classification

As with the object detection model, to assess suitability of different text classification techniques, the requirements set out in Section 3.3.3.1 were used to define a list of criteria:

1. Fast but accurate classification
2. Can cope with a relatively small dataset
3. Can handle a dataset containing feature vectors of varying lengths
4. Can be converted for use in an iOS application

Criteria 1 was defined for the same reasons as criteria 1 for the object detection model. Criterions 2 and 3 were then defined when considering the two potential datasets selected for use for training the model. These were the Chars74K dataset [170] and a handmade dataset of Reycling Labels, built by scraping data from the web using the Microsoft Cognitive Services Bing Image Search API [145] and only have 7112 and 249 images respectively, meaning that they are both on the small end of typical machine learning datasets. This has an effect on the approach selected, as typically Deep Learning techniques (such as Recurrent or Convolutional Neural Networks) require vasts amount of data to provide superior performance, as highlighted by Figure 4.8. Furthermore, both datasets contain images of different sizes and ratios, meaning that their resulting feature vectors[1] are of varying lengths. Not all machine learning techniques are capable of dealing with such a characteristic, and therefore these criterions were introduced.

As with Criteria 1, Criteria 4 was defined for the same reasons as for the Object Detection model criteria. However, as the label classification can be performed by classification of the words on the label, and is therefore not necessarily specific to this application, a potential solution to this limitation is integrating a pre-built model, such as as Apple's 'Vision' Framework [172], or Google's Tesseract Enginge [120], that is specifically designed for iPhone integration. This was not possible for the object detection model as the classification problem could not be simplified in this manner. If a pre-built model was chosen, neither criteria 2 or criteria 3 would be relevant anymore, as pre-built models do not require training, therefore meaning that they do not require a dataset.

[1]Column vectors containing the pixel intensities of each image

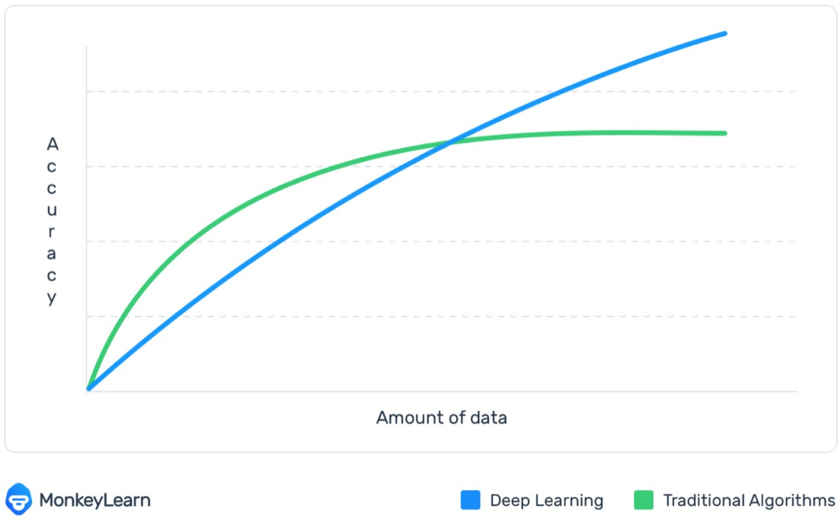


Figure 4.8: The Effect of the Amount of Training Data on the Accuracy of Machine Learning Methods [171]

Therefore, a decision had to be made between implementation of a custom model specific to this application, or integrating a pre-built model. Considering the two options actually highlighted how much the process would be simplified by choosing a pre-built model; A dataset would no longer be required, the models have already been optimised to provide fast and accurate classification, and integrating such technologies into an iPhone app is relatively trivial. For this reason, the design decision to implement a pre-built model for this method of classification was made.

The experimentation on and comparison of the various models that could be used to implement both the object and label classification will be covered in the Implementation section.

4.3 Testing

The final stage of an app development project is to carry out user and performance testing. This is essential to validate the features and the design, as well as to detect any bugs that could've potentially been missed during the development stage. Detecting bugs is especially important, as they will have a negative effect on the user experience.

4.3.1 User Testing

This section is concerned with the app's usability and overall impact on a user's recycling performance, as well as the user's perception of the app. This section of the project is unique when compared to performance testing or machine learning analysis, as it will result in both quantitative and qualitative data. The quantitative data from this section will be easy to measure and compare to results produced by previous rounds of testing. However, in comparison, the qualitative data, such as answers resulting from user surveys, will be open to interpretation. In an effort to keep this interpretation consistent, users will be asked questions with Likert scale [173] answers, rather than open-ended questions. The Likert scale asks survey participants "to rate their level of agreement to items that describe a topic" [173], as can be seen in Figure 4.9.

Strongly Disagree	Somewhat Disagree	Neither Agree Nor Disagree	Somewhat Agree	Strongly Agree
1	2	3	4	5

Figure 4.9: The Likert Scale [173]

4.3.1.1 Structure

Due to the COVID-19 pandemic, this stage of the project had to evolve from the original plan of face-to-face interviews, to a social-distancing-suitable approach. This meant adapting the testing interviews to ones that could be feasibly performed via a video conferencing software, such as Zoom, or online forms, such as Microsoft Forms or Typeform.

To this effect, second round interviews were held using Zoom [174], a video conferencing software, and final round interviews using Microsoft Forms. The high level structure of interviews at each stage of testing can be seen in Figure 4.11.

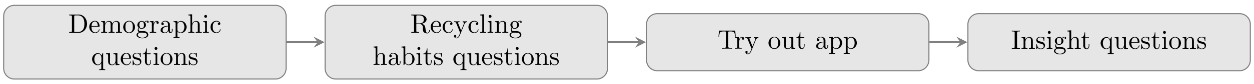


Figure 4.10: User Testing Interview Structure

With each stage of development came an iteratively improved app version. However, an iteratively improved testing strategy and series of questions was also implemented. Essentially, at each stage of testing, alongside evaluating the app, the testing, questions and results were also evaluated, in order to maximise the useful information that could be gained at the next stage of testing. For the three question stages, the questions were generated using the process seen in Figure 4.11.

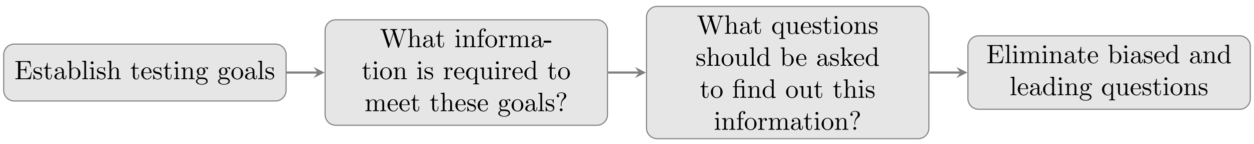


Figure 4.11: User Testing Question Generation

The questions will be designed so as to collect information from app testers on the following areas:

- Overall Impression
- Features
- Support
- Interfaces
- Outputs
- Screen Layout
- Sequence of Screens

For the final round of testing, to fully establish the effect of the app, each participant completed a recycling knowledge test both before and after using the final app for a period of time. They were also asked to assess their likeliness and motivation to recycle at each stage. This allowed the impact of the app on their knowledge and motivation to be analysed.

4.3.1.2 Metrics

As each response in Figure 4.9 corresponds to a number from 1-5, the average for each response can therefore be calculated and interpreted. Furthermore, if all questions are posed such that a 'strongly agree' answer corresponds to a positive aspect of the design, then averaging the scores over the survey can generate a 'user satisfaction score'. This in turn could be averaged over all users to find the 'average satisfaction score'. Intuitively, both of these scores should be maximised.

$$\text{User Satisfaction Score} = \frac{1}{Q}\sum_{q=1}^{Q} S_q = \text{USS} \tag{4.1}$$

$$\text{Average Satisfaction Score} = \frac{1}{N}\sum_{n=1}^{N} \text{USS}_n \tag{4.2}$$

Where N is the number of users who completed the survey, Q represents the number of questions in the survey, and S_q represents the score assigned to each question by the user's response.

An example of interpreting a score from a specific question is as follows; if 5 people are interviewed, and the sum of the responses to the statement "the screen layout of the app is helpful" is 21, then the average response can be calculated to be 4.2. This means that the majority of testers 'strongly agree' with the statement and it can therefore be concluded that the layout should not change. In contrast, if the score is 2.2, they 'mostly disagree', and hence it should be changed.

Beyond these questions, usability of the app can further be measured through the simplicity and efficiency of the design, as well as the satisfaction provided by the app. Table 4.2 outlines the criteria required to meet the various usability goals, as well as metrics to be measured to assess if the goal has been met.

Goal	**Criteria**	**Metric**
Simplicity	It should be simple to input data	Time taken to input data
	The output should be easy to understand	Rating scale
	It should be easy to learn how to use the app	Time taken to learn
		Number of mistakes made
Efficiency	In-app tasks should be quick to complete	Time taken to complete task
	The app should respond quickly	App response time
	It should be easy to learn how to use the app	Time taken to learn
Satisfaction	The app should have a positive effect on the user	User survey
	Users should not feel frustrated when using the app	User survey
	Users should be happy with the UI	Rating scale
	Users should feel familiar with the UI	Rating scale

Table 4.2: Measuring App Usability

Finally, any change in the user's recycling performance can be assessed through comparison of the user's perception of their recycling behaviour, their motivation and their actual recycling knowledge before, during and after using the app.

Both the user's perception and their motivation can be measured through participation in a survey that asks them various questions about what they think about their recycling behaviour and knowledge. This can then be quantified using the same method as the User Satisfaction Score in Equation 4.1. This result will be referred to as the User Perception Score.

The user's recycling knowledge can then be assessed in a similar way. However, instead of a survey, each user can be asked to complete a quiz about how to recycle certain objects, both usual and unusual. These scores will be referred to as The User Knowledge Scores, and can then the results from before using the app, whilst using the app[2] and after using the app can be compared. Additionally, the longer term impact of the app on a user can be tracked by periodically getting them to answer further quizzes to test their knowledge.

4.3.1.3 Evaluation

After the completion of each round of user testing, the results should be evaluated, to look for correlation between variables, as well as to assess statistical significance. This essentially evaluates how likely it is for the given occurrence (e.g. correlation) to actually occur. This means that, say a bug or problem was encountered, if it was calculated to not be statistically significant then it could be safely assumed that the majority of users would not experience it. To this effect, fixing it would not be classed as high priority. However, this evaluation must also take into account the sample size, as well as the statistical power of the observations and statistics. This process of looking into correlation and statistical significance that may arise from usability testing is also known as A/B testing.

First, null and alternate hypotheses, H_0 and H_a respectively, for the correlation should be defined. An example of correlation that should be analysed is the relationship between a user's recycling knowledge, and their recycling knowledge. Therefore:

Null Hypothesis: There is no correlation between a user's recycling knowledge, and their likeliness to recycle.
Alternate Hypothesis: There is a positive correlation between a user's recycling knowledge, and their likeliness to recycle.

[2] i.e. using the app to find the answers to the questions

The first stage to establishing which hypothesis should be accepted is by calculating the Pearson Correlation Coefficient:

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}} \tag{4.3}$$

Where x and y each represent the scores assigned by each user i to two different variables. r = 1 represents a perfect positive correlation, and r = -1, a perfect negative correlation. Therefore, r = 0 intuitively represents no correlation.

From here, the statistical significance of any results can be calculated, by calculating the p-value of a t-distribution. First, t is calculated:

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \tag{4.4}$$

Where r is the Pearson Correlation Coefficient previously calculated, and n is the sample size. This score assumes that the results follow a normal distribution, which, by the Central Limit Theorem, is a valid assumption for a sequence of independent, identically distributed random variables, with finite mean and non-zero variance, provided that n is sufficiently large.

Looking this calculated value of t up in statistics tables, for $n - 2$ degrees of freedom, will produce the p-value. If this is below the set significance level (normally $\alpha = 0.05$), the result can be deemed to be statistically significant. If the result is statistically significant, the null hypothesis is rejected, and the alternate hypothesis is accepted.

Whilst the above describes the comparison of *intra*-testing-round[3] results, a further stage of statistical analysis was also designed as a means of comparing *inter*-testing-round[4] results. For this comparison, due to the results being obtained from different groups of participants, on different app versions, the results can be thought of as from separate populations, and hence the Pearson Correlation Coefficient was no longer relevant. Instead, a calculation known as the T-Test can be performed[5]. However, the actual calculation carried out depends on whether the samples from each population are repeated/dependent, or independent.

For this project, repeated samples are those that participated in both the rounds of testing that are being compared:

$$t_{repeated} = \frac{\frac{1}{n}\sum(x-y)}{\sqrt{\frac{\sum(x-y)^2 - \frac{1}{n}(\sum(x-y))^2}{(n-1)n}}} \tag{4.5}$$

In contrast, independent samples are those that only participated in one of the rounds:

$$t_{independent} = \frac{\bar{x} - \bar{y}}{\sqrt{\left[\frac{(n_x-1)s_x^2 + (n_y-1)s_y^2}{n_x+n_y-2}\right]\left[\frac{1}{n_x} + \frac{1}{n_y}\right]}} \tag{4.6}$$

Where x and y are the scores reported from two separate rounds of testing, and n is the number of samples, $\bar{x}$ and $\bar{y}$ represent the mean scores for each round of testing, n_x and n_y are the number of samples for rounds x and y respectively and s_x^2 and s_y^2 are the sum of the squared standard deviations, divided by $n - 1$, for rounds x and y respectively. E.g.:

$$s_x = \frac{1}{n_x - 1}\sum_i (x_i - \bar{x})^2 \tag{4.7}$$

Essentially, this approach of statistical analysis contains one less step when compared to the correlation analysis, as all that is required to calculate statistical significance is to look up the t statistic for n - 2 degrees of freedom, in order to find the p-value. This is then again compared to the set significance level, to determine which hypothesis is accepted, and which rejected.

At this point it is important to note that, whilst statistical significance is important, it is not *essential* in user testing, as user testing only provides input to the design process, rather than defining it.

[3] I.e. within a testing round.

[4] I.e. between testing rounds.

[5] This value of t represents the same as the value calculated for the Pearson Correlation Coefficient, but has to be calculated different due to the different relationship between results.

4.3.1.4 Ethics and Data Consideration

Finally, GDPR consent needed to be gathered, and the ethics of the testing assessed. GDPR consent was obtained through verbal confirmation during each video interview, which was recorded (with the interviewee's permission) for documentation purposes[6]. For the final round of testing, held using multiple microsoft forms, GDPR consent was gathered during completion of the form. Whilst user testing results can be found in the GitHub repo, to preserve compliance with GDPR, these were altered so that all results were anonymous and couldn't be linked to participants. Furthermore, all copies of these results, both online and offline, will be deleted after the examiner's meeting in September 2020.

In terms of the ethics of the user testing, this was considered by consulting Imperial's Ethics Approval Overview. This states that "a research must consider the ethical implications of any work that" [176]:

- Could have a negative effect on mental or physical health of participants
- Could jeopardise participants' safety
- Could compromise participants' data
- Involves "sensitive methods or subject matter"
- "Has the potential for environmental impact"
- Has the risk of a conflict of interest by the researcher and/or the College

Taking these into consideration highlights that this testing does not require ethics approval.

4.3.1.5 Challenges

With the challenge of the COVID-19 pandemic mentioned, the main challenge faced at this stage was internet connection. If either person in the call was experiencing this problem, the interview couldn't take place. Furthermore, the setup meant that anyone with a phone not meeting the requirements (Apple, running iOS 13+) was unable to take part in the testing, unless someone that they were isolating with also had access to this hardware. Previously in this situation, they could've been lent a spare phone that had the suitable software, to allow them to complete the testing.

4.3.2 Performance Testing

This section will focus on testing and evaluating the *performance* of the developed app. This is a key area to focus on, as a badly performing app will have a negative effect on the user experience. Furthermore, it is good practice to check for things like any fatal crashes or impractical uses of the device memory before allowing users to download the app onto their device for testing.

The metrics to be used to evaluate the app performance can be seen in Table 4.3. All of these metrics mentioned can be measured using one or more of three Xcode tools:

1. **Instruments Developer Tool**: This can be used to profile an app's performance - selecting different profiling templates allows the developer to profile, analyse and understand what is going on the background when the selected application is running.

2. **Debug Tab**: This can be used to analyse the CPU Usage of an app. They key difference between the Debug tab and Instruments is that Instruments offers more information and greater functionality when analysing app performance.

3. **Organiser**: This offers data from TestFlight about Crashes, Energy Usage and Metrics such as Memory Usage and Hang Rate.

In addition to analysing the results of the metrics seen in Table 4.3, further analysis of app performance can be completed using an Apple-developed framework, `XCTest` [177]. This can be used to write unit, integration and UI tests for Xcode projects. Furthermore, this framework has `measure` blocks that can be used to "measure the performance of a block of code" [177] within these XCTests. This will be useful for measuring the performance of specific portions of the code, but not for the app as a whole.

Due to the iterative development style of this project, these metrics can be evaluated by comparing to previous

[6] "The GDPR is clear that consent requires clear affirmative action, and Recital 32 sets out additional guidance on this: "Consent should be given by a clear affirmative act... such as by a written statement, including by electronic means, or an oral statement." " [175]

versions of the app, as well as industry standards. Furthermore, the Debug tab provides a comparison of the app's performance to other apps on the device. These comparisons can then be used to set a threshold of improvement for the next development stage.

Metric	**How to Measure**	**Reason for Importance**
Number of Crashes	Xcode Metrics Organiser	An app crashing generally signifies a fatal error and also frustrates the user - monitoring any crashes can locate the cause and fix it.
Responsiveness & Launch Time	Time To First Byte (TTFB) Instruments (*App Launch, Core Animation*)	iOS has a feature known as the 'watchdog timer' that terminates any apps taking too long to launch. Therefore, monitoring and improving an app's launch time will minimise the risk of this happening, as well as improve the UX.
CPU and Thread Usage	Xcode Debug Tab Instruments (*Time Profiler, System Trace*)	This can provide understanding of what app features are the most process-heavy. This is especially important in apps involving machine learning. For example, a high CPU Usage whenever the app is running suggests that the model is constantly trying to make predictions, which is most likely unnecessary. Furthermore, app performance and responsiveness are heavily linked with CPU usage.
GPU Usage	Instruments (*Game Performance*)	GPU usage and performance is linked to frame rate, which is in turn linked to user experience. Intuitively, a low frame rate is classified as low performing as it will make an app feel "sluggish or disruptive to its users" [178]. Therefore monitoring GPU usage will help find the cause of any moments of low frame rate.
Battery Consumption	Xcode Debug Tab Instruments (*Energy Log*)	If an app uses up too much battery, the user will not be able to use their phone when needed. Therefore monitoring battery consumption can help highlight any features that might be particularly power hungry so that the developer knows if they need to change their implementation method for that particular feature.
Memory Usage	Xcode Metrics Organiser Instruments (*Allocations*)	If an app takes up too much memory, then when the user tries to use another app, the operating system may 'delete' some of this in order to free up space for the use of other apps. If this occurs, the app's responsiveness will decrease when it is re-opened. Therefore, monitoring Memory Usage and ensuring that it is not to high can reduce the chance of this happening.
Hang Rate and Duration	Xcode Metrics Organiser	Minimize the rate and length of time that an app 'hangs' for makes an app more responsive.

Table 4.3: App Performance Metrics

4.4 Conclusion

This chapter focussed on introducing RecycleHelper at a high-level, highlighting project planning and design choices made. An iterative development strategy was chosen to speed up the development process, as well as mitigate any potential risks that may arise due to the implementation of more complicated components. Furthermore, to improve modularity and allow ease of future work being completed on the project by someone else, the app was developed using an MVC architecture, and the UI designed according to Apple's Human Interface Guidelines. Behaviour chains were designed to implement behaviour design according to target behaviours, and the requirements of each of the machine learning models were defined. Finally, the user testing process was introduced, and considerations made regarding the COVID-19 situation, ethics, GDPR and other challenges.

Table 4.4 summarises how the success of the project and the app will be assessed, as has been discussed in Section 4.3.

Project Requirement	**Evaluation Method(s)**	**Metric Goal(s)**
Accurate & Reliable Waste Classification from Images	Confusion Matrix	Maxmise values on the diagonal
	Macro Precision	Maximise
	Macro Recall	Maximise
	F1-score	Maximise
Good App Performance	No. of Crashes	Minimise
	Responsiveness	Maximise
	Launch Time	Minimise
	CPU, GPU, Thread Usage	Minimise
	Battery Consumption	Minimise
	Memory Usage	Minimise
Well Designed UI and UX	User Satisfaction Score	Maximise
	Average Satisfaction Score	Maximise
	Time taken to learn to use app	Minimise
Impacting a User's Recycling Performance	User Perception Score	Maximise Increase
	User Knowledge Score	Maximise Increase

Table 4.4: Project Evaluation with Corresponding Requirements

These design decisions will be further elaborated in the following chapter, where the process of the development of RecycleHelper is explained.

Chapter 5

Implementation

As highlighted in Section 4.1, RecycleHelper was built following an iterative development strategy. This development procedure occurred over three stages of design, building, testing and analysis - this chapter will cover the implementation of this.

The code written and full experimentation and analysis performed for all stages of this project can be found in a GitHub repository at

https://github.com/rch16/RecycleHelper

5.1 Minimum Viable Product

As highlighted in Section 4.1, the iterative development of RecycleHelper began with the implementation of a Minimum Viable Product, or MVP. This was defined as the first version of the app, that has most, if not all of the 'must have' features, only a few of the 'should have' features and none of the 'could have' features. The app features that were therefore implemented in this version have been covered in Section 3.3.4.

5.1.1 Design

The simple nature of the fact that only the 'essential' features are included an MVP meant that the UI should also uncomplicated and un-embellished. To this effect, only two of the views discussed in Section 4.2.1 were implemented; the camera and information views. The first step at this point was to produce a mockup of what the UI would look like. This was designed using a software called Figma [179], and can be seen in Figure 5.1.

(a) Scanning an item (b) Item detected in image (c) Item classified

Figure 5.1: Mockup of Minimum Viable Product

This highlights how the user can use their device camera to look for an object in *real time*. A box (the bounding box) would then appear on screen to highlight that an object had been detected, before a pop up appears, displaying what material the item had been detected as, and the corresponding information for how to recycle it. Tapping the 'X' in the top right hand corner of the pop up will dismiss it and allow the user to identify a new item. It is important to note that, for the MVP, no extra work was placed into developing the UI or persuasive techniques, as these were defined as 'should have' features. However, they will be implemented in the subsequent versions.

5.1.2 Build

5.1.2.1 Waste Classification Model

Whilst the final version of the app has an optimised machine learning model in order to achieve the best possible accuracy, precision and recall, the model used in the first version of the app was trained with Apple's Create ML Developer Tool found within Xcode [180]. Create ML is a simple drag-and-drop tool that automatically trains a model and outputs an .mlmodel file, the format expected by Xcode. This design decision was made in order to save time and focus instead on the first round of testing and evaluation.

Training any image classification model requires a training and testing data set large enough to allow the model to learn enough information about each class to be able to accurately classify future images. However, creating an image data set of household waste large enough is by no means a small task and so a previously-made data set [137] was initially used, as set out in Section 4.2.3. The distribution of the dataset among its categories can be seen in Table 5.1:

Category	Number of Images
glass	501
paper	594
cardboard	403
plastic	482
metal	410
trash	137

Table 5.1: TrashNet Dataset [137]

These images were then split into a 70 : 13 : 17 ratio split to create training, validation and testing sub sets respectively. Training the model using the Create ML software mentioned previously produced the following results:

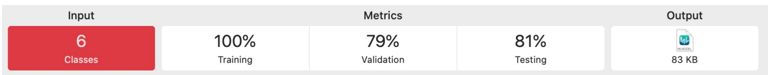


Figure 5.2: Accuracy of Classification of Training, Validation and Testing Data Sets

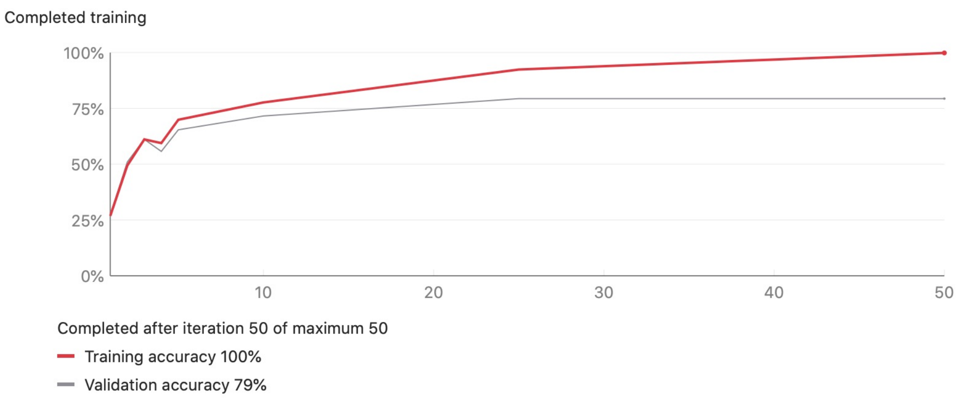


Figure 5.3: Effect of the Number of Iterations on Classification Accuracy

Figure 5.2 highlights the accuracy of classification of the model on each of the data sets. It is intuitive that running the model on the training data set produces an accuracy of 100%, as this is what the model has trained its classification decisions on, and therefore this data has been previously seen by the model. Figure 5.3 further highlights that increasing the number of iterations increases the accuracy of classification for both the training and validation data sets. The increase in training accuracy with the number of iterations shows that with each iteration, the model is learning new features about the data. The number of iterations here is equivalent to the number of epochs when building a machine learning model in python. As the accuracy has reached 100% by 50 iterations, this suggests that increasing the number of iterations will not provide any further improvement in the performance of the model. In contrast, the validation accuracy is merely a representation of how similar

Class	Precision	Recall
cardboard	100%	100%
glass	100%	100%
metal	100%	100%
paper	100%	100%
plastic	100%	100%
trash	100%	100%

Table 5.2: Accuracy of Classification of Training Data Set per Class

the training data is to the validation data.
The trend of 100% accuracy for the training set seen in Figure 5.2 is continued in Table 5.2, which shows 100% precision and recall for all classes. Again, this is expected, due to the nature of how the model was trained. However, the size of the 'trash' class when compared to the size of all the other classes suggests that the TrashNet data set is actually not evenly weighted between classes. Improving this will be included as an important part of the implementation of the final waste classification model, as an uneven distribution between classes can lead to model bias.

Class	Precision	Recall
cardboard	86%	88%
glass	79%	78%
metal	78%	80%
paper	82%	88%
plastic	78%	69%
trash	59%	56%

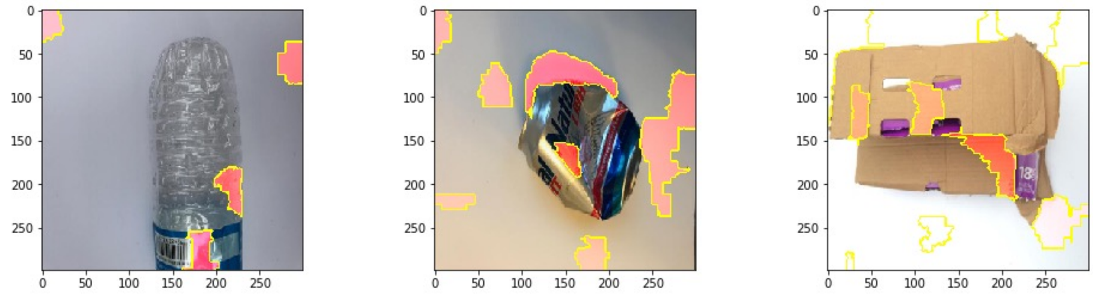
Table 5.3: Accuracy of Classification of Validation Data Set per Class

Class	Precision	Recall
cardboard	92%	90%
glass	82%	75%
metal	77%	77%
paper	85%	89%
plastic	80%	83%
trash	51%	51%

Table 5.4: Accuracy of Classification of Testing Data Set per Class

Tables 5.3 and 5.4 illustrate the classification accuracies for each class in the validation and testing data sets. Both tables clearly show a decrease in accuracy when compared to the training data set. This is partly due to the fact that the model is now performing classification on unseen images, i.e. images not used in training. Both tables highlight that the model is able to consistently predict cardboard, paper and glass the accurately, but struggles when classifying trash. This could be due to a multitude of reasons, such as over-fitting on the training data, or even having images originally mis-classified in the data set.

A further useful evaluation of model performance is by using a method known as Local Interpretable Model-Agnostic Explanation, or LIME, Analysis [181][182][183]. This is a technique that can be applied to *any* machine learning model, and can interpret *why* certain predictions were made by machine learning models, i.e. what areas of the image was the model using to make that classification. Analysis using this method is completed in Python, and therefore the coremltools python library [169] was used to import the waste classification model and convert its predictions into the desired format. An example of applying the LIME model to an example image from the dataset can be seen in Figure 5.4.



(a) Plastic (b) Metal (c) Cardboard

Figure 5.4: Test Images with complete LIME Analysis

Figure 5.4a clearly shows that around 50% of the areas of the image that caused the model to predict the item to be plastic were not in fact part of the item in question, and were instead part of the background. This suggests that in future models, some further image processing needs to occur to remove large areas of plain background before the model is trained or makes a prediction. Further examples can be seen in Figure 5.4b and 5.4c. Whilst these images show the model using a larger portion of the actual objects, they still show that some areas of the background of the images are being using in the classification by the model. This highlights areas of improvements that need to be focused on when implementing the optimised waste classification model.

5.1.2.2 App Development

The implementation of this app version had four main areas:

1. The UI and UX
2. Live capture using the device camera
3. Registering when the device was being held still (*Registration*)
4. Running the waste classification model on the image detected (*Detection*)

As mentioned in Section 4.2.1, the app was designed using the XCode storyboard feature. This is used to drag and drop UI components onto different views, and define transitions, or segues, between screens. The storyboard for the MVP App can be seen in Figure 5.5. Due to the nature of the MVP version of the app only having 'must have' features, this initial storyboard is intuitively very simple.

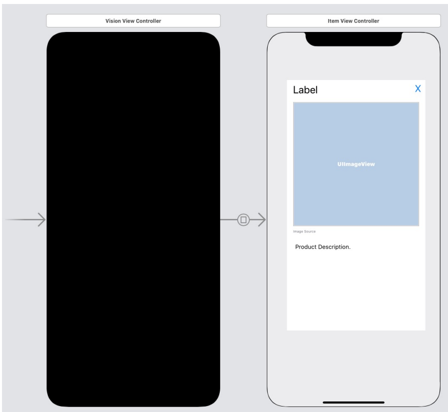


Figure 5.5: Storyboard for MVP App

The black screen of the Vision View Controller represents how the screen shows a preview of the images being captured by the device camera in real time. Upon registration and detection, a pop up was then designed to appear over this display, providing the relevant recycling information for the image detected. This pop up and information display is represented by the Item View Controller. The light grey background of this screen was modified to be translucent, so that it would have the 'pop up' effect, rather than making the user feel like they had navigated to a new screen. Applying constraints within the storyboard feature can allow the ratios and locations of all elements to remain the same, regardless of device size or orientation. The items added onto the UI using storyboard can be linked to the code using IBOutlets. This allows the code to detect when a button has been pressed, or know which information on screen needs changing.

The key implementation and design of the different features of this app version can be summarised into three categories:

Figure 5.6: MVP Features

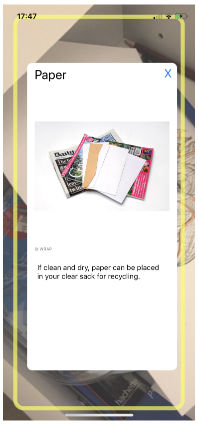
Live Capture: Live image capture can be achieved in iOS design using an AVCaptureSession. This is configured by defining the video input and output, as well as embedding the capture process in a background thread to ensure that it does not block other app operations. Each image processed by this method was stored as a pixel buffer. The nature of Apple libraries and classes means that implementing features such as this are always very standard and are therefore easily set up by following the relevant documentation [184]. Furthermore, the Apple Developer site has an extensive list of example apps and explanations to allow even the most inexperienced developer to learn how to develop an app of their choice. An example of such documentation is the "Recognizing Objects in Live Capture" sample project [185].

Registration: Once the AVCaptureSession has been configured, the app needs to detect when the device is being held still. Without this, the app would continuously attempt to identify items. Not only would this require a lot of processing power, but would also likely mean that the model is performing classification on out-of-focus images, when not actually *needed*. Image Registration is the "process of overlaying two or more images of the same scene taken at different times" [186]. This detects when a device is being held still by assessing whether two images taken in short succession are similar, within a set threshold. If the images are different, this suggests that the camera has been moved and thus the device is *not* being held still. In contrast, if the images are similar, this suggests that the camera has only moved slightly, if at all, and therefore the device is being held still. Furthermore, the device being held still suggests that the user is trying to get the app to detect an item. This method of Image Registration can be used by creating a VNTranslationImageRegistrationRequest [187]. Essentially, the manhattan distance between the current image and the previous image is calculated. If this distance is below a defined threshold, the system defines that scene stability has been achieved and therefore the device is held still. At this point, the design was changed slightly from that of the mockup; instead of a bounding box appearing when an object was detected, a box was displayed on screen to indicate that the system had detected that the device was being held still and hence detection had begun.

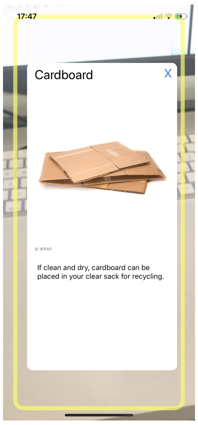
Detection & Classification: When the device is defined as being held still, image detection and classification can begin. This was achieved using Apple's Vision Framework [172]. Each time the device is registered as being held still, the current pixel buffer is fed into the request handler. This handler was set up to use the model described in Section 2.5. It returns confidence levels that correspond to how confident the model is that the image belongs to each of the classes. When a confidence is returned that is greater than 98%, the request is marked as complete. As soon as the item has been identified, a segue from the Vision View Controller to the Item View Controller occurs. This mean that the pop up seen in Figure 5.5 appears, with the corresponding label, image and description, specific to the Item ID (class) identified by the classification model. At this point, it is important to note that even at a confidence of 100%, the image classification could be incorrect - the ideal threshold varies from model to model, and further experiments must be made to find the optimal confidence threshold that limits the number of false positives returned.

With each of the various app features implemented, the app can now be built and run. Whilst Xcode offers a great simulation tool, this does not work with apps that use the device camera. Therefore the app had to be simulated on an iPhone from the beginning of the project. However, this offers extra portability in the testing and evaluation stage.

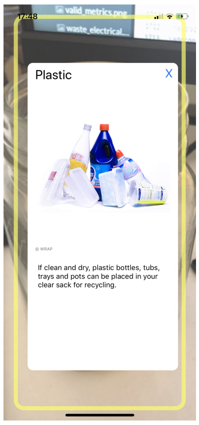
The pop-ups for each of the potential classes can be seen in Figure 5.7.



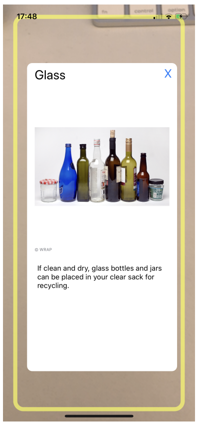
(a) Paper



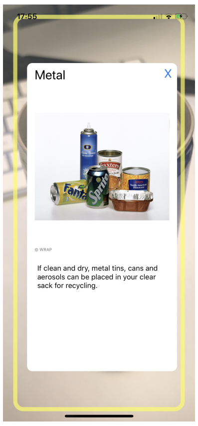
(b) Cardboard



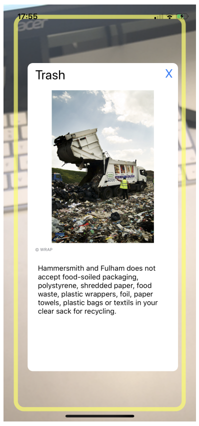
(c) Plastic



(d) Glass



(e) Metal



(f) Trash

Figure 5.7: MVP App in Action

5.1.3 Testing and Analysis

As highlighted in Section 4.1, an essential stage of app development is testing. This stage is two fold; both the app performance and the usability need to be assessed. Therefore, both performance and user tests were carried out.

A key component at this stage, essential to testing the app with real users, is an Apple Developers Licence. Without this, Apple does not grant developers access to TestFlight Beta Testing. With a licence, TestFlight can be used to invite users to download, test and provide feedback on Beta versions of the app. Furthermore, key performance metrics are often not collected and therefore not available to view before the app has been tested by users in TestFlight. Therefore, a Developers licence was obtained.

5.1.3.1 Performance Testing

Section 4.3.2 highlighted the range of metrics that can be measured to evaluate the performance of an iOS application. Excellent performance is not necessarily required or expected at this stage in the development process, due to the skeletal nature of the MVP. However, taking some measurements is beneficial for use as a point of comparison for the later app versions. To this effect, the launch time and thread usage were analysed. Figure 5.8 visualises the different threads active during operation of RecycleHelper.
Whilst this is pretty hard to interpret, it highlights the number of threads that run during app operation, to allow for simultaneous operations to occur. This is important, as it is good practice to minimise the operations that are placed in the main thread, as these can become blocking. However, updates to the UI must be performed on the main thread. Essentially, the worker threads are used to convert a single-thread application, into a multi-threaded one. These are generally used for CPU-intensive, non-UI-related tasks [188].
A further interesting thread is the camera thread. The activity on this thread can be seen in Figure 5.9. Each 'spike' in this profile highlights when a new pixel buffer is processed by the camera.

Finally, the application launch time was tested. According to Apple, developers should aim to build an application should ideally launch within 400ms, and *never* take longer than 20s [189]. Therefore, this metric was

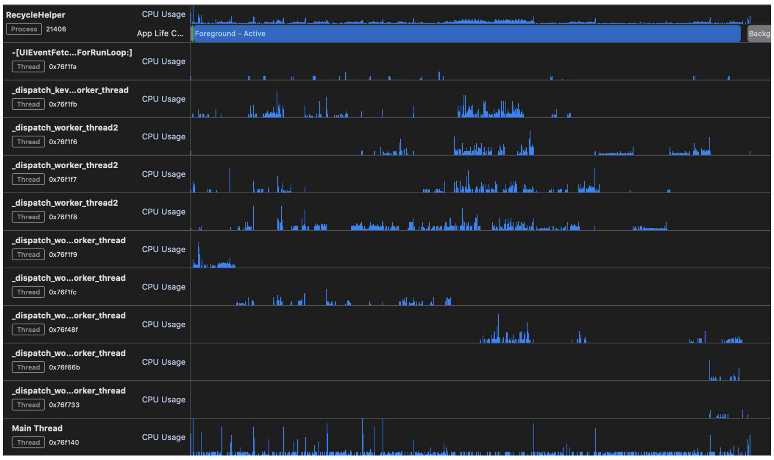


Figure 5.8: MVP Thread Activity

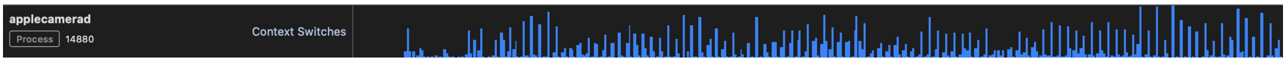


Figure 5.9: MVP Camera Thread Activity

measured to set a baseline and assess how/if the operation of the app needed to be improved. Figure 5.10 highlights the average launch time of the application over 7 different scenarios:

1. Following average use of phone
2. Having just turned phone on
3. After force-quitting the app
4. Having just used a large app
5. With multiple other apps open
6. With a few (5) other apps open
7. With no other apps open

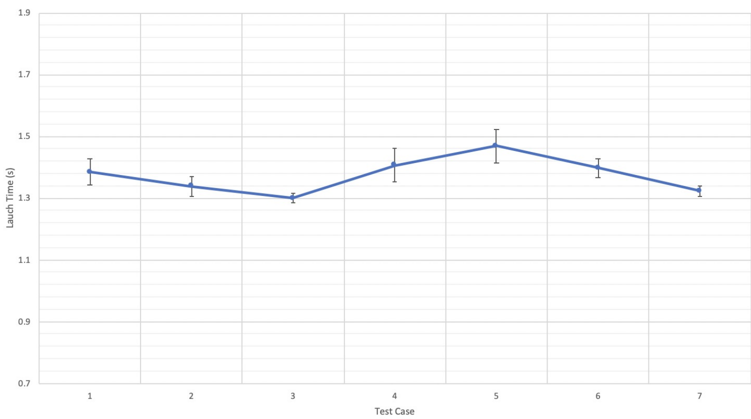


Figure 5.10: MVP Launch Times

This graph shows that, regardless of test case, app launch time is between 1.3-1.5s, with an average standard deviation 0.0699. This highlights that, although it these times may be well under the maximum specified time of 20s, and can most likely be classified as a 'quick' launch, there is still room for plenty of optimisation and improvement.

5.1.3.2 User Testing

The user testing of this app version was carried out through 7 (independent) user interviews, consisting of 16 questions, which can be found in Appendix B. The interviewee age range was 18 to 23, and the unique locations can be seen in Figure 5.11.

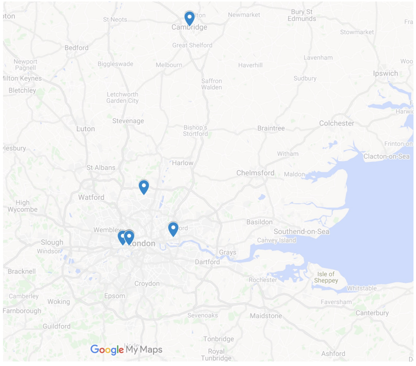


Figure 5.11: Unique Locations of MVP Testers

The first stage of the interview was to obtain data about each interviewee's recycling habits and opinions, such as how they rated their recycling knowledge, and how likely they were to recycle, on a scale of 1 to 5 (i.e. the Likert Scale). These results can be seen in Tables 5.5. Included is their overall likeliness to recycle, calculated as the average of their likeliness to recycle when at home and when out. Additionally, the User Perception Score, or UPS, was calculated, as the average of how each participant perceived their knowledge and likeliness to recycle.

Participant Number	**Recycling Knowledge**	**Likeliness to recycle... At home**	**When out**	**Overall**	**User Perception Score**
1	2	2.5	1.5	2	2
2	4	4	2	3	3.33
3	3	5	3	4	3.67
4	4	5	3.5	4.25	4.17
5	3	5	2	3.5	3.33
6	3.5	4.5	4	4.25	4
7	4	5	2.5	3.75	3.83
Average	**3.357**	**4.429**	**2.643**	**3.536**	**3.475**

Table 5.5: User Interview Recycling Habits

These results highlight two key trends. First, all participants reported themselves to be more likely to recycle at home when compared to outside of the house. Upon further questioning, this was discovered to be mainly down to a lack of recycling facilities in public spaces. The second trend was that, in general, the higher a participant felt their recycling knowledge was, the greater they assessed their likeliness to recycle to be, regardless of location. This can be further seen, as in Figure 5.12 by sorting the recycling knowledge scores in ascending order, and then plotting this against overall likeliness to recycle. This can then be used to plot a trendline of overall likeliness with increasing recycling knowledge.

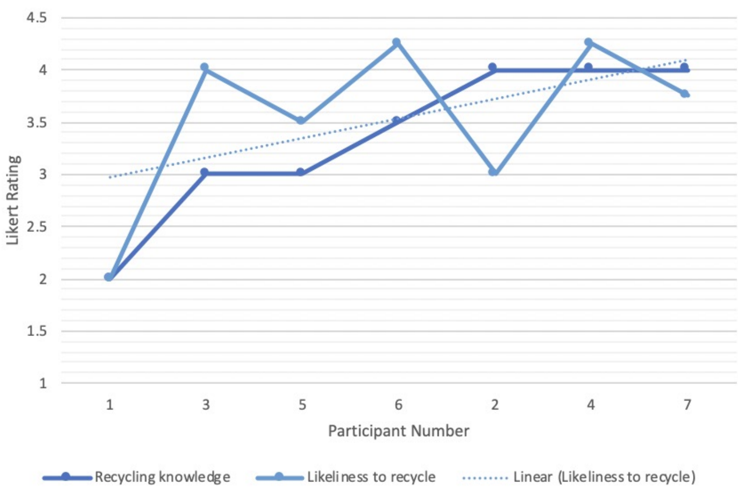


Figure 5.12: The Effect of Recycling Knowledge on Likeliness to Recycle

This trend highlights how lack of education can act as a barrier towards improved recycling behaviour. In contrast, the results of participant number 2, whose recycling knowledge is perceived as high, but their likeliness to recycle low, suggest that for them, the barrier is not education, and instead, motivation. This observation was further supported by a calculated Pearson Correlation Coefficient of $r = 0.629$, indicating a **moderate positive correlation** between recycling knowledge and likeliness to recycle. However, the t-value was calculated to b $t = 1.811$, which, at $n - 2 = 5$ degrees of freedom, results in $0.05 < p < 0.1$, highlighting that these results could **not** be confirmed to be statistically significant at $\alpha = 0.05$. This could perhaps be due to the small sample size of only 7 participants, and therefore the next round of testing was planned include a much larger group of participants. At this point, the statistical significance of any results will again be analysed.

Once the interviewees' recycling views and habits were collected, they were presented with the app, allowed to experiment with the various features, and then asked another series of questions. At this stage, specific emphasis was placed on app usability, and whether the recycling information that was presented post-classification was easy to understand. To this effect, the participants were again asked to use the Likert scale, this time to rate the Usability and 'Understandability' of the app. The 'User Satisfaction Score', or USS, was then calculated by taking the average of these two scores. The results for this can be seen in Table 5.6.

Participant	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Average**
Usability	5	3.5	5	3.5	5	4.5	4	**4.357**
Understandability	5	3.75	4	5	4	4	5	**4.393**
USS	5	3.625	4.5	4.25	4.5	4.25	4.5	**4.375**

Table 5.6: MVP Usability and Understandability

Table 5.6 highlights average usability and understandability scores of 4.357 and 4.393 respectively, resulting in an average user satisfaction score of 4.375. This allows the conclusion to be drawn that, in its present state, the app is extremely usable. This was further assessed by observing whether each participant was able to use the app without guidance, which returned a 100% success rate. However, given this high usability, special consideration must be made when developing the app further. This is because introducing further features will further complicate the app, and hence they must be carefully designed in order to maintain this usability. A target was therefore set to keep this score above 4.

A final comparison was made when considering the effect that each interviewee's recycling knowledge could have on their satisfaction with the app. The graph for this comparison can be seen in Figure 5.13. As with Figure 5.12, the recycling knowledge scores were sorted in ascending order, and a trend line applied to the user satisfaction scores. This actually highlights a decrease in user satisfaction as recycling knowledge increases. This in turn suggests that, whilst the app helps those with less recycling knowledge, due to the simplicity of recycling information currently available in-app, those with more recycling knowledge found that it didn't provide much value to them as it wasn't able to provide them with information that they didn't already know.

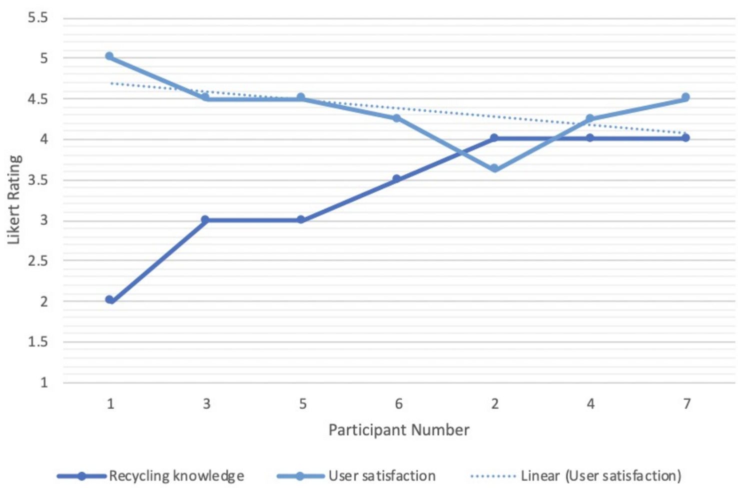


Figure 5.13: The Effect of Recycling Knowledge on User Satisfaction with the App

As before, the Pearson Correlation Coefficient was calculated, and found to be $r = -0.772$, highlighting a

strong negative correlation between recycling knowledge and user satisfaction score. This confirms the observations seen in Figure 5.13. Furthermore, a t-value of $t = -2.721$ was obtained, producing $0.025 < p < 0.01$. This highlights that the **result is statistically significant** at $\alpha = 0.05$, therefore allowing the alternate hypothesis of "there is a negative correlation between recycling knowledge and user satisfaction score" to be accepted.

Finally, each interview was concluded with a general discussion about how they interpreted certain aspects of the UI, and how the UX made them feel. This brought up 5 issues, as seen in Table 5.7, prioritised according to severity. That is, critical if an issue made it impossible for users to complete tasks, serious if an issue was frustrating for many users, and minor if the issue is annoying, but not going to drive users away.

Issue Encountered	**Priority**	**Occurrences**	**Frequency (%)**
No explanation or on-screen instructions	Minor	3	42.86
Misinterpreting the yellow border	Minor	7	100
No clear 'recyclable' or 'not'	Minor	1	14.29
Classification too quick → pop-up becomes annoying	Serious	2	28.57
iOS too low for app to work on participant's own phone	Critical	2	28.57

Table 5.7: Issues Observed from User Testing Round 1

5.1.4 Conclusions

The issues highlighted in the previous section were used to create a plan for going forward into the second round of testing. Namely, for each issue encountered, a solution was a proposed, with a plan to implement each solution in order of priority. Thus, the requirements were captured for stage 2 of implementation, as outlined in Table 5.8.

Issue Encountered	**Proposed Solution**
No explanation or on-screen instructions	Provide instructions or onboarding
Misinterpreting the yellow border	Instructions with (registration) frame, or change what frame represents
No clear 'recyclable' or 'not'	Clear title alongside instructions
Classification too quick → pop-up becomes annoying	Only begin classification when user presses start
iOS too low for app to work on participant's own phone	Extend iOS functionality to older versions
Those who knew more about recycling gained less from the app	Provide a wider range of recycling information
App launch time > 400ms	Investigate cause and try to improve
Results not statistically signifant	Increase the number of user testing participants

Table 5.8: Implementation Round 2 Requirement Capture

5.2 Design Iteration

With the first stage of development, testing and analysis complete, work could beging to iteratively improve RecycleHelper. Therefore, this section highlights how the findings of stage 1 were used to produce RecycleHelper Version 2.0.

5.2.1 Design

The first step was to combine the findings of the first stage of testing, with the project specification, to produce the requirements for this build. In terms of the project specification, at this stage in the development process, it was decided that work should begin to implement the 'should have' features. Therefore, the combined requirements became:

- A backup identification method
- Location-specific information
- Off-device database
- Well-designed UI
- Persuasive techniques
- Onboarding process
- On-screen instructions
- Clear 'Recyclable' or 'Not' title
- User-initiated classification
- Extended iOS functionality
- Wider range of recycling information
- Improved app launch time

This list of 12 features was therefore used as the design specification for this app version. A new app version requires a new UI mockup to be designed, which was again performed using Figma. This can be seen in Figure 5.14.

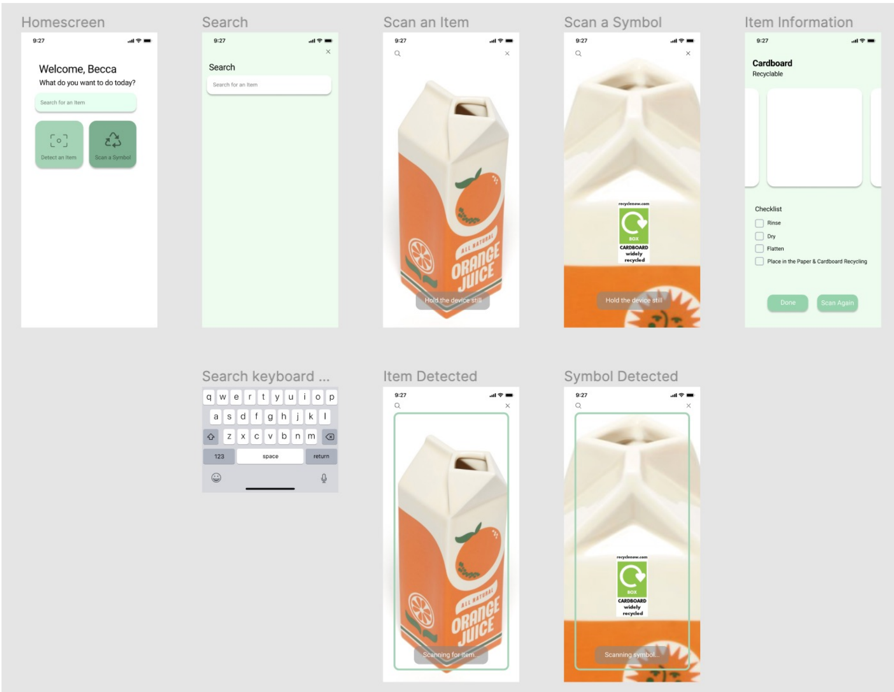


Figure 5.14: Mockup of RecycleHelper V2

This design highlights the introduction of two back up methods for waste classification; a search function, and a recycling symbol scanner. Furthermore, the UI was updated so that the user could select when to begin scanning, so as to minimise any annoyance that may be caused by quick classification. In addition, a colour scheme was chosen to be used throughout the app, to create a more seamless design. Finally, the item information view was updated to provide a checklist of what to do with an identified item, and the scanning view was updated to provide instructions about holding the device still.

5.2.2 Build

5.2.2.1 Waste Classification Model

Due to the volume of new features being added, a decision was made to focus on the usability for this stage of development, as with increasing number of features comes an increasing risk of having poor usability. To this effect, it was decided that the waste classification model would be improved in the final stage of development.

5.2.2.2 App Development

As with the first round of development, this round can be divided into key areas:

1. The UI and UX
2. Search functionality
3. Recycling symbol classification
4. The onboarding process
5. Recycling information

Due to the live capture, registration, camera preview, and object detection/classification already being implemented in stage 1, these were not included in this list. The XCode storyboard design for this version can be seen in Figure 5.15.

Figure 5.15: Storyboard for App Version 2

The Navigation Controller on the left of Figure 5.15 was a new addition to the design. This is the point of entry to the app that, although unseen by the user, controls the navigation through the app by pushing or popping views on/off the stack. This allows for easy implementation of navigation between screens, as a back button is automatically added in the navigation bar. The light grey lines between screens represent the transitions, or segues, that occur when a user interacts with the UI, or perhaps a classification is made.

The implementation followed the design laid out in the mockup seen in Figure 5.14. The only differences are the introduction of two information buttons on the home screen, that cause pop ups displaying further information, and the layout of the buttons to the different app features on the home screen. Finally, the item view controllers' backgrounds were changed from green to white, to improve the readability as well as make the design more consistent.

Furthermore, the recycling information displayed in-app was previously information specific to the London Borough of Hammersmith & Fulham. In addition, it only covered a small subset of materials, namely Paper, Cardboard, Plastic, Glass, Metal and Rubbish. Testing highlighted that this limited information decreased the likeliness of users with above average recycling knowledge actually using the app. Therefore, the design decision was made to expand and improve this information. To do so, the website `recyclenow.com` was used. This website features an exhaustive list of 240 items, and how to correctly recycle or dispose of them. Furthermore, it offers a feature where visitors can input their postcode, and the information will be updated to be specific to their location. This list of objects and relevant information was therefore extracted and condensed into 86 items and stored in a property list file (.plist) for use in the app[1].

Alongside this expansion of the information available in-app, three new features were implemented in this version, namely:

Search Function: The list of 86 items mentioned previously were displayed in the SearchViewController by dynamically populating a tableView component. A user is then able to scroll through the list to find a particular item, or alternatively use the search bar to filter the list. This instantaneously reloads the table to show only the items that meet the search criteria.

Recycling Symbol Classification: This method of classification is based on the OPRL recycling labels introduced in Section 2.2.4.1, and works on the assumption that these recycling labels include a piece of text describing the material of the object. To this effect, text recognition was used to perform the classification. Whilst the live capture and preview layer was implemented the same way as for the Object Detection described in Section 5.1.2.1, the subsequent processing of the pixel buffers was achieved via different steps.

Figure 5.16: Recycling Symbol Classification

First, a VNDetectTextRectanglesRequest is made, which uses Apple's Vision Framework to locate any text present in the pixel buffer. This returns a series of coordinates known as a 'bounding box', which locates the text within the pixel buffer coordinate system. This is then converted to the preview layer's coordinate system to display the bounding boxes on screen, as well as to crop the image to only the area containing text. The currently processing pixel buffer is then updated to be this cropped image, which prevents the model from having to process any images that do not contain text. The next step involves this new pixel buffer undergoing text recognition. After experimentation using the Vision Framework, and two third party models known as SwiftOCR[2] and Firebase MLKit, the latter, MLKit was chosen as providing the optimal performance. This was installed into the application framework using CocoaPods, " a dependency manager for Swift and Objective-C Cocoa projects" [190].

Onboarding Process This was implemented in a single view controller, using a ScrollView, allowing multiple pictures and text to be displayed in one view. The aim of this screen was to act as an introduction to the app and its functionality, and therefore it was programmed to only appear on first launch of the application. This was achieved by tracking the status of a "LaunchedBefore" UserDefault in the SceneDelegate.swift file. Upon checking the boolean value stored within this UserDefault, the app would launch to either the home screen, or the onboarding one.

[1]The website was credited as the information source whenever this information was displayed in-app.
[2]This is a swift 'shell' for the C++ based OCR technique, Tesseract, by Google.

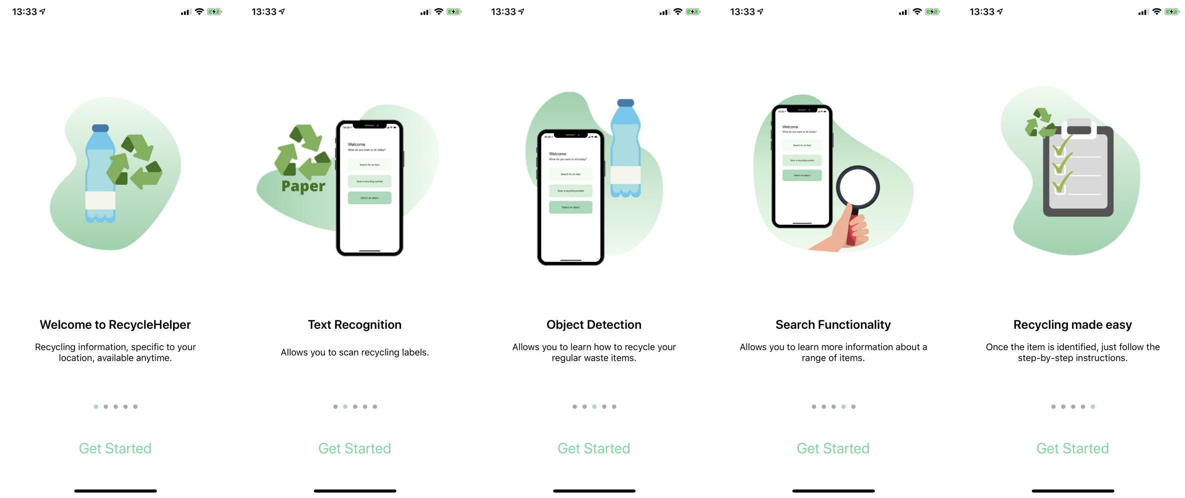


Figure 5.17: RecycleHelper Version 2 Onboarding

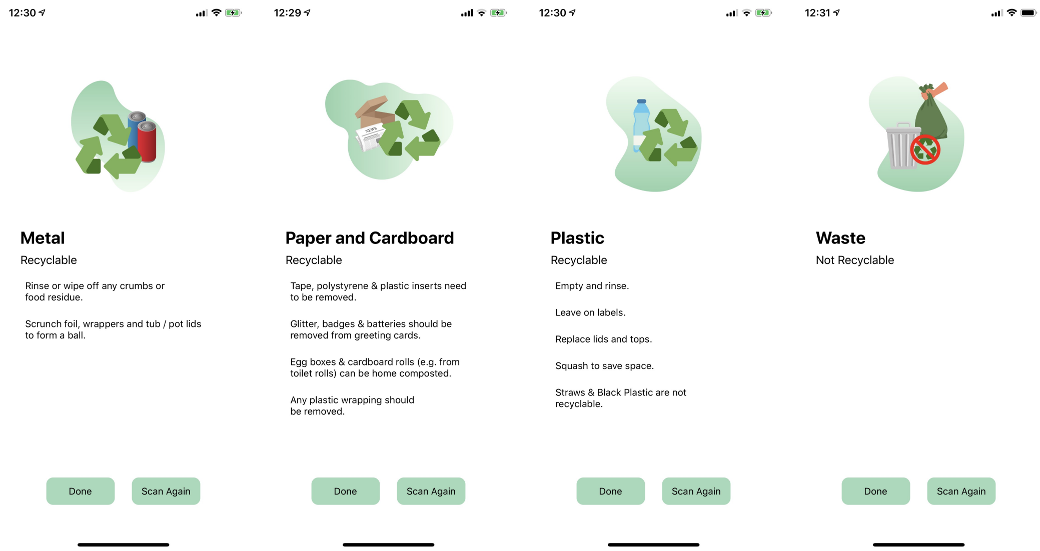


Figure 5.18: Examples of RecycleHelper Version 2 Item Views

Figures 5.17 and 5.18 highlight that the UI re-design also included designing various graphics to go alongside the text/instructions of the onboarding and item information screens. These were designed to be consistent in terms of colours, shapes and graphics used, in order to provide a seamless user experience. To further improve the UX, the colour scheme seen here was used throughout the app, including the launch screen and app icon. The set of colours involved in these scheme can be seen in Figure 5.19. The colours were chosen for their relation to recycling and recycling symbols, as well as creating a neutral and simplistic feel.

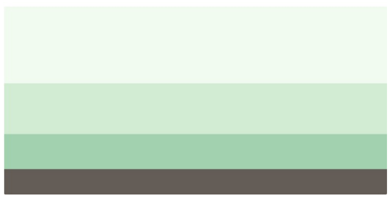


Figure 5.19: RecycleHelper Colour Scheme

5.2.3 Testing and Analysis

5.2.3.1 Performance Testing

One of the key comparisons made during this stage of testing, was the launch time of this app version, compared to that of the previous version, as this had been highlighted as an area of improvement. This comparison, for the same test cases as before, can be seen in Figure 5.20.

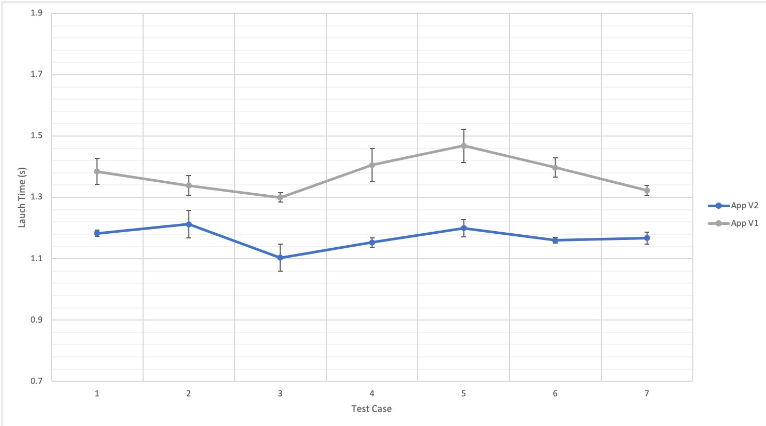


Figure 5.20: Comparison of Launch Times of App Versions 1 and 2

This highlights an improvement in the application launch time between app versions 1 and 2, with the launch time now ranging between 1.1 and 1.3s. This was partly due to the fact that the application no longer launches directly to the camera session, instead opening to a home screen. Whilst there is still work to do to get it under 400ms, this is definitely a step in the right direction. To improve the user experience and distract away from this launch time, a launch screen was designed, that shows whilst the application is loading. This was designed to be simplistic and inline with the rest of the UI design, and can be seein in Figure 5.21.

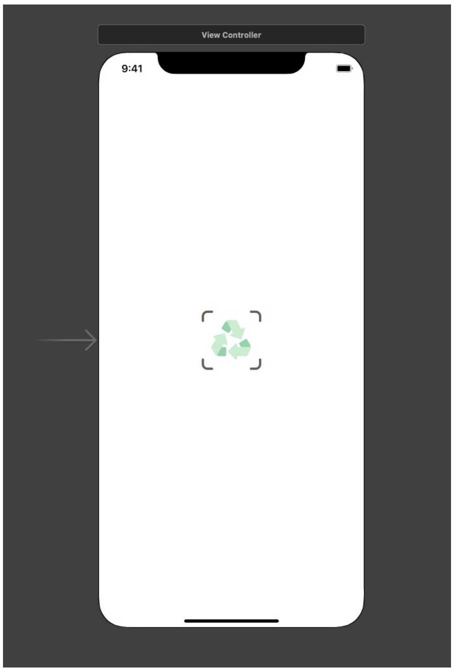
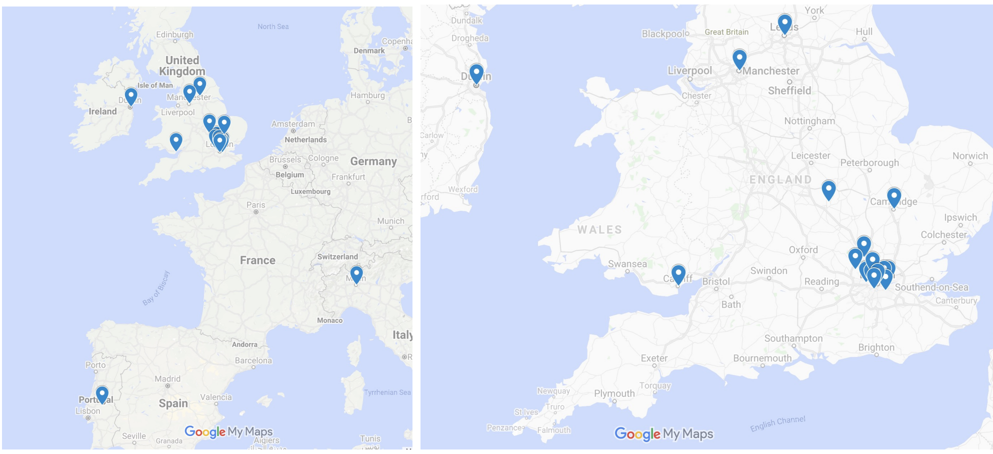


Figure 5.21: RecycleHelper Launch Screen

As the operation design in terms of thread allocation and usage etc was not modified from the previous version, performance testing in terms of CPU and thread usage monitoring was deemed unnecessary. More emphasis will be placed on performance testing in the final stage of development, as that round will produce the app version to be released to the general public and must therefore have optimal performance and operation.

5.2.3.2 User Testing

Whilst only 7 user interviews were held in the first round of user testing, with interviewees from 5 unique locations, this round of testing involved 21 user interviews, across 19 unique locations. These locations can be seen in Figure 5.22.



(a) All Locations (b) UK Locations

Figure 5.22: Unique Locations of Round 2 Testers

Furthermore, the age range of participants was increased, from 18 to 23 in round 1, to 18 to 58 in round 2. Finally, based on the answers received from various questions in the first round, the interview content was developed, now containing 30 questions, with questions that would hopefully provide further observations.

As discussed in Section 4.3.1, the approach used to build the user interview, was to first generate a list of goals that this round of testing should achieve. In no particular order, these were:

1. How do app users feel about their onboarding experience?
2. Do users feel like the app will *help* them to recycle (more)?
3. Do users feel like the app makes them *want* to recycle (more)?
4. Does the app work as expected?
5. Does the app provide accurate information?
6. Is the information provided clear and easy to understand/follow?
7. How does using the app make the user feel? / What is the user experience like?
8. Is the app easy/intuitive to use?
9. Does the layout make sense?
10. What features would the user like to see in a general recycling app?
11. What features would the user find useful in a general recycling app?
12. Which method of identification (search/labels/object) was preferred?

These goals were then split into topics, namely Usability (4-9), Feature Sets (10-12), and Sentiment and Impact (1-3,7). The questions that were designed, and hence the full interview plan, can be found in Appendix D.

The advantage of an increased number of testers over a wider range of locations is that any results gathered become more likely to be statistically significant. Furthermore, the wider age and location demographic increased the likeliness of the data being more representative of the UK population. However, due to the size of the test set, the full results can be found in the git hub repo.

A further difference between this round of testing and the previous, was the fact that the UK COVID-19 lockdown was introduced very early-on in this stage of development. This meant that all interviews had to be held remotely, as covered in Section 4.3.1. However, this provided the opportunity to collect data on whether participants' recycling behaviour varied between being in lockdown and not. This will be discussed later on in this section.

The first stage of each interview was to collect information of participants' recycling habits and views. As with the first round, a general trend of increasing likeliness to recycle with increasing recycling knowledge was observed, as seen in Figure 5.23.

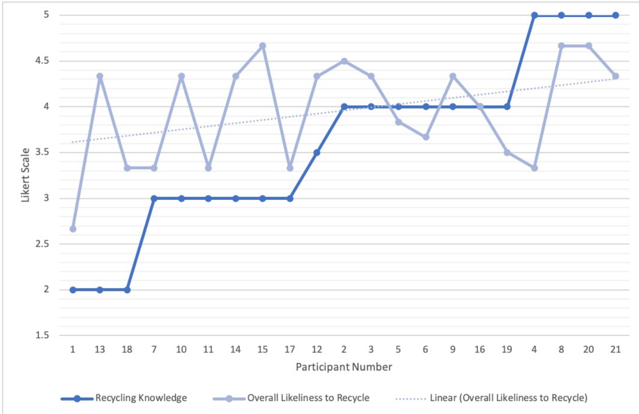


Figure 5.23: The Effect of Recycling Knowledge on Likeliness to Recycle

A repeat of the comparison made in the first round of testing, the goal here was to re-evaluate the correlation and statistical significance, now that the sample size was 3 times larger. This time, the Pearson Correlation Coefficient was calculated to be $r = 0.534$. While lower than before, this still highlights a **moderate positive correlation** between a participant's recycling knowledge and their likeliness to recycle. Furthermore, the t-value was calculated to be $t = 2.751$, producing a p-value of $0.005 < p < 0.01$ at $n - 2 = 19$ degrees of freedom. In contrast to the first round of testing, this allowed the conclusion to be drawn that the result **is statistically significant** for $\alpha = 0.05$, thus allowing the alternate hypothesis of "there is a positive correlation between recycling knowledge and likeliness to recycle" to be accepted, and the null hypothesis rejected.

A further comparison was then made between recycling knowledge and motivation to recycle, as seen in Figure 5.24. As a reminder, these variables were found to be positively correlated in the previous round of testing. Initial inspection of the results highlights that, whilst in general most participants reported a 5 out of 5 for motivation, if they reported lower than this, the value reported was more likely to be lower if they also rated their recycling knowledge as lower. This suggests a slight correlation between knowledge and motivation. Calculation of the Pearson Correlation Coefficient yielded a result of $r = -0.181$, confirming that any correlation was only weakly positive, if correlated at all. A suspicion at this stage of the testing was that user's motivation to recycle was more influenced by the promotion of awareness of climate change that is currently present in social media, rather than by their own personal recycling ability.

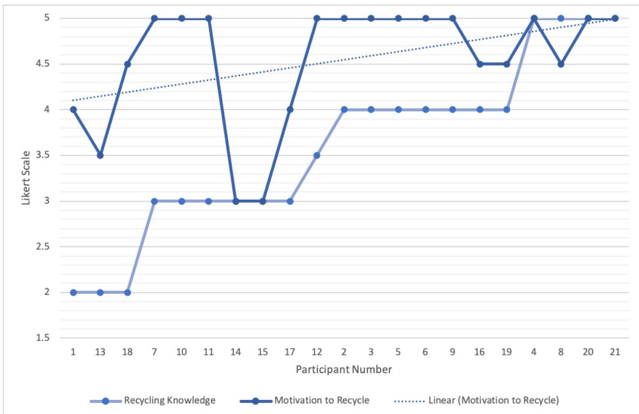


Figure 5.24: The Effect of Recycling Knowledge on Motivation to Recycle

Next, test participants were asked to assess how much they viewed themselves as 'someone who recycles'. As seen in Figure 5.25, this highlights perhaps the clearest trend of all, with *all* participants responding with a score within 1 point of how they rated their recycling knowledge. This insinuates a correlation between recycling knowledge, but perhaps in reverse when compared to the previous trends. That is, the more someone recycles, not only the more they consider themselves as a 'recycler', but also the better their recycling knowledge. However the Pearson Correlation Coefficient between these two variables was calculated to be $r = -0.084$, which, due to its proximity to 0, instead suggests no correlation. Furthermore, the t-value was calculated as $t = -0.3692$, corresponding to $p > 0.25$, highlighting that the data was not statistically significant.

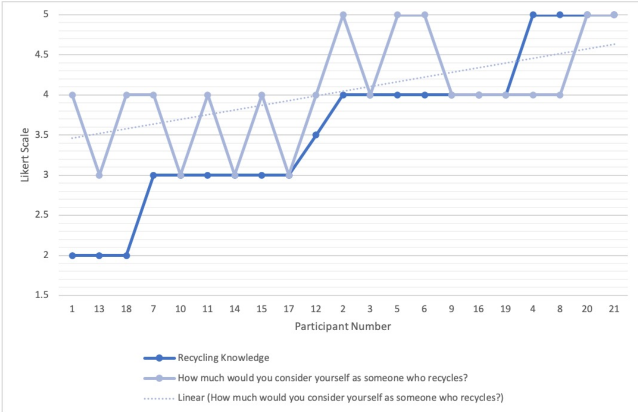


Figure 5.25: The Effect of Recycling Knowledge on How Much Someone Views Themselves as a 'Recycler'

Finally, participants were asked what factors prevented them from recycling more. The aim of this question was to assess whether any of these factors could be controlled or prevented by the app, to have a further impact on the recycling rates. The responses to this question can be seen in Figure 5.26. This highlights that the most common factor was the lack of availability of recycling bins in the general area. Unfortunately, this is not something that RecycleHelper can control. However, the second and third most common responses were related to recycling information, a factor that RecycleHelper *is* being designed to combat.

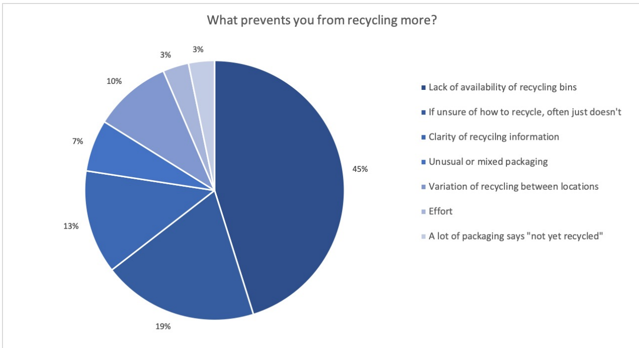


Figure 5.26: Factors preventing users from recycling more

With these insights gathered, each interview then progressed to the app testing stage. As seen in Appendix D, questions were posed to each interviewee both before and after they were given access the app. Before, an exploratory conversation was held, where participants were asked to describe features that they would find helpful in a recycling app, as well as features that would be 'nice to have'. The idea here was to obtain ideas and insights from the potential users, before they became 'biased', due to having seen the app. Therefore, the aim of this discussion was to help with the requirement capture for the final stage of the iterative development process, as user insights and opinions should carry a high priority when designing such a user-focused product. The various features suggested at this stage can be seen in Figure 5.27.

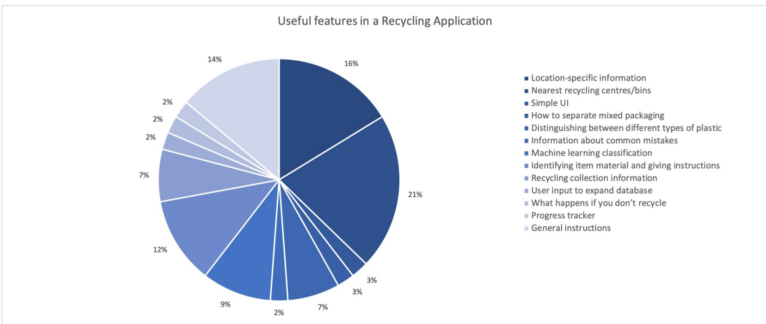


Figure 5.27: Features users would find useful in a Recycling App

At this stage, test participants were provided access to RecycleHelper, via Apple's "TestFlight" Beta Testing program. They were encouraged to try out all of the application's features, and how they interacted with the interface was observed. Then, they were asked a series of 5 questions related to usability, that were used to calculate the new User Satisfaction Score. Increasing the number of questions increases the range of information covered, and therefore intuitively improves how much the USS represents the usability of the app. The questions were:

1. How clear / easy to follow was the recycling information?
2. How easy / intuitive was the app to use?
3. How helpful is the app?
4. How likely are you to use this app?
5. How helpful was the onboarding experience?

The average results and standard deviation for these questions can be seen in Table 5.9.

Question No.	1	2	3	4	5
Average	4.690	4.690	4.286	4.119	3.619
Standard Deviation	0.460	4.60	0.582	0.865	1.254

Table 5.9: Average and Standard Deviation of Results to Usability Questions

To calculate the user satisfaction score for this round of testing, the average of the results of all 5 questions mentioned above per was taken. This resulted in a score of **4.281** with a standard deviation of 0.409. However, for accurate comparison between this version of the app and the previous, only the user satisfaction scores of the 5 participants that participated in both rounds of testing were compared. These can be seen in Table 5.10.

Participant No.	1	2	3	4	5	**Average**	**Std. Dev.**
Round 1	5	3.615	4.5	4.25	4.25	**4.325**	**0.497**
Round 2	5	3.9	4.6	4.4	4.3	**4.44**	**0.404**

Table 5.10: Comparison of User Satisfaction Scores of the Participants who Completed Both Rounds of Testing

This highlights an increase in the User Satisfaction Score for **all** participants, apart from participant one, who had already reported the maximum score of 5 in the first round. This results in an average score of **4.44** for this second app version, an increase of 0.115 from the previous stage. Furthermore, the standard deviation was calculated to be **0.404**, smaller than the previous stage, suggesting that the results are more in agreement. This observation was then validated using the T-test, and the following hypotheses:

Null Hypothesis, H_0 = No increase in usability was seen from app version 1 to app version 2
Alternate Hypothesis, H_{a_1} = An increase in usability was seen from app version 1 to app version 2

During this analysis, it was calculated that the data collected resulted in $t = 2.438$ and $0.025 < p < 0.05$, therefore concluding that the result is **statistically significant** for $\alpha < 0.05$. In contrast, the independent samples (i.e. those that only completed one of the testing rounds) found a *decrease* in usability, with averages of 4.5 and 4.23 for the first and second rounds respectively. This suggests a second alternative hypothesis of:

Alternate Hypothesis, H_{a_2} = A decrease in usability was seen from app version 1 to app version 2

However, this result was found to be not be statistically significant at $\alpha = 0.05$ due to a calculation of t = 0.900 and $0.1 < p < 0.025$. Hence, the original alternate hypothesis of an increased usability from app version 1 to app version 2 in the repeated samples was **accepted**, and the null hypothesis rejected.

The final comparison that was made was between particpants' lockdown and non-lockdown recycling habits. In general, it was found that the interviewees found very little difference between their habits in lockdown compared to when not in lockdown. However, observations were made that, due to reduced recycling collections and facility availability (such as Tips being closed), recycling had become more difficult. For those who did report a difference between the lockdown and non-lockdown recycling practices, lockdown was found to cause an increase for some, as they were home more and therefore had more opportunities to recycle. However, for others, it was reported that when not in lockdown, they lived with people with better recycling practices than whoever they were quarantined with, and therefore they felt that their recycling had decreased during the lockdown period. Due to this variation in responses, no real trends could be established between lockdown and non-lockdown behaviour, but it was an interesting comparison nonetheless.

To conclude the user testing section, a second list of issues encountered was generated, with each issue assigned a priority. This can be seen in Table 5.11. These issues, alongside the participants' insights, and the initial requirement capture, will be used to generate the set of requirements for the final stage of development.

Issue Encountered	Priority	Occurrences	Frequency (%)
Low camera preview quality on iPhone 11	Minor	1	4.76
No error message if searched item wasn't found	Minor	3	14.29
UI constraints slightly off on smaller phones	Minor	4	19.05
If phone was in dark mode, some text couldn't be seen	Serious	5	23.81
Object Detection too quick when user tapped begin	Serious	5	23.81
Classification sometimes inaccurate	Serious	7	33.33
Recycling symbol classification didn't work or wasn't applicable	Critical	15	71.43
App crashed	Critical	3	14.29

Table 5.11: Issues Observed from User Testing Round 2

5.2.4 Conclusions

As before, solutions were proposed for each of the issues encountered in this stage of testing, for use in the requirement capture process of the next development stage. These can be seen in Table 5.12.

Issue Encountered	Proposed Solution
Low camera preview quality on iPhone 11	Implement a feature where the camera quality adapts according to what model phone the application is running on.
No error message if searched item wasn't found	Display message saying search item not found and suggest closest matches
UI constraints slightly off on smaller phones	Update constraints
If phone was in dark mode, some text couldn't be seen	Check adaptive text colours are implemented for all text
Object Detection too quick when user tapped begin	Implement delay between opening up the camera view and classification begins
Classification sometimes inaccurate	Develop machine learning model
App crashed	Recreate crashes to find causes

Table 5.12: Implementation Round 3 Requirement Capture

5.3 Final Version

As before, the end of the previous round of development called for the beginning of the next. This time, the final round. To this effect, this section highlights how the findings of the previous stages were used to produce the final version of RecycleHelper.

5.3.1 Design

This stage involved capturing the final set of requirements. These were identified from the features already included in previous versions, the results of the testing and evaluation, and the various features outlined in the initial project specification. The requirements for this stage and hence for the overall project can therefore be seen in Table 5.13.

Feature	**Notes**	**New?**
Waste Classification	This feature was defined as a must-have in Section 3.3.1.	
Wide Range of Recycling Information	This feature was defined as a must-have in Section 3.3.1. Limited recycling information was included in Version 1, which was extended further in Version 2. This should be extended even further in Version 3.	
Off-Device Database	This feature was defined as a must-have in Section 3.3.1, and implemented on-device for both Versions 1 and 2. To optimise performance and allow for the implementation of location specific information, this should be off-device in Version 3.	✓
Backup Identification Method	This feature was defined as a should-have in Section 3.3.1, and implemented in the form of search and symbol scanning features in Version 2. This should be further developed in Version 3.	
Location-Specific Information	This feature was defined as a should-have in Section 3.3.1, and should be implemented in Version 3.	✓
Well-Designed UI	This feature was defined as a should-have in Section 3.3.1, and has been developed further throughout each stage of implementation.	
Persuasive Techniques	This feature was defined as a should-have in Section 3.3.1, and should be implemented in Version 3.	✓
Usage Tracking	This feature was defined as a could-have in Section 3.3.1, and also identified as a feature that users would like to have during user testing. This should therefore be implemented in Version 3.	✓
Onboarding Process	This was identified as necessary in the first round of user testing to ensure that users understand the purpose of the app, as well as how to use it. This was implemented in Version 2, but should be refined in Version 3.	
Issue Fixes	**Notes**	**New?**
Clear Recyclable Label	Highlighted in round 1 testing, this was implemented in Version 2.	
Low Camera Quality	Users with an iPhone 11 reported low camera quality on the camera preview screen. Highlighted in round 2 testing, this should be fixed in Version 3	✓
No Search Results Error Message	Users searching for an item that wasn't found weren't informed that it didn't exist, and instead the full list was shown. Highlighted in round 2 testing, this should be fixed in Version 3.	✓
UI Constraints	Highlighted in round 2 testing, this should be fixed in Version 3	✓
Dark Mode Behaviour	Some text did not adapt to if a User's phone was in dark mode, making it hidden for some users. Highlighted in round 2 testing, this should be fixed in Version 3	✓
Object Detection Speed	This is where the object detection feature works too quickly. Highlighted in both round 1 and 2 testing, this should be fixed in Version. 3	✓
Classification Accuracy	Poor classification accuracy was highlighted in both round 1 and 2 testing. This should be fixed in Version 3	✓
iOS Version Functionality	Highlighted in round 1 testing, this was not fixed in Version 2 as lower iOS versions do not offer the required functionality. A workaround should be provided in Version 3.	✓

Table 5.13: Requirement Capture for the Final Version of RecycleHelper

From here, the final app mock ups could be designed, as seen in Figure 5.28.

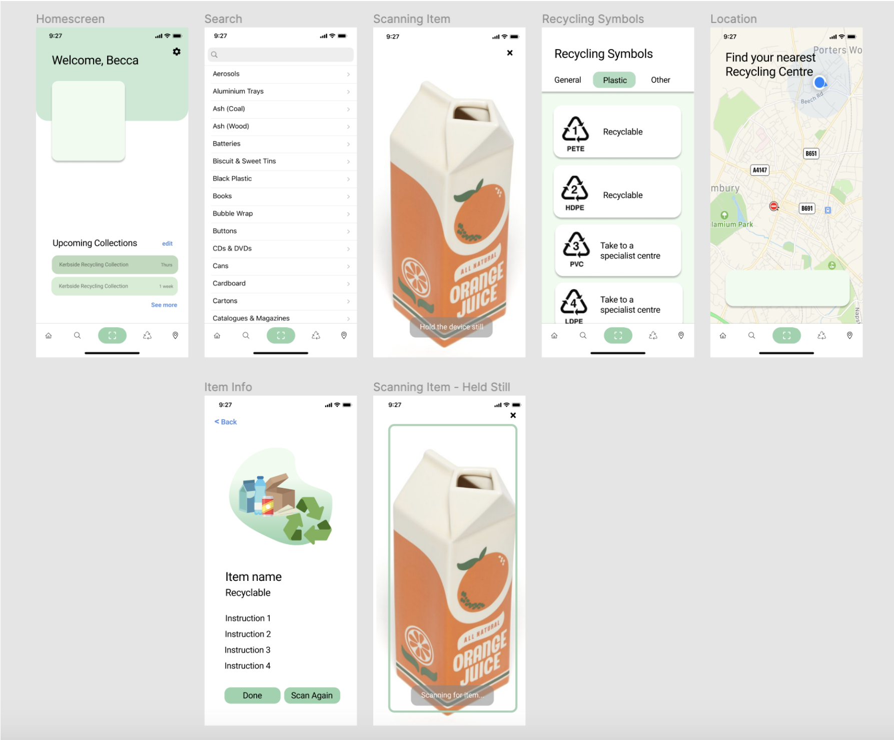
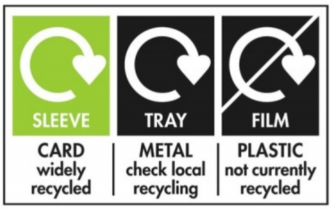


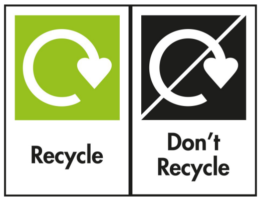
Figure 5.28: Mockup of RecycleHelper V3

Due to the significant increase in the number of features implemented in this app version, the design was modified from a single view app, to one which is navigated using a tab bar. This design choice was made to ensure that navigation between features was still easy and intuitive, ensuring that users wouldn't get lost in the navigation hierarchy - a key requirement that can 'make or break' an app's usability. The tab bar was chosen specifically over other potential methods of navigation in order to exploit the Mere-Exposure Effect. As mentioned in Section 2.4.3.4, this is a "psychological phenomenon by which people tend to develop a preference for things merely because they are familiar with them" [98]. Use of a tab bar is an example of this effect, as some of the most popular apps on the appstore, such as ZOOM Cloud Meetings (#2), WhatsApp (#3), TikTok (#4) and Instagram (#5), all use a tab bar to allow users to navigate through their app. This means that iPhone users are likely to already be very familiar with this technique, thus minimising the learning curve that users must experience when first using RecycleHelper. Furthermore, both TikTok's and Instagram's 1st, 2nd and 3rd Icons on their tab bar are Home, Search and Camera respectively. Therefore, this sequence of views was replicated in RecycleHelper to further increase a user's familiarity with the app. Finally, the tab bar was designed to have 5 icons to simplify a user's decision progress, which gets longer and more complicated the more choices there are, due to Hick's Law. Therefore, exploiting the Mere-Exposure effect and Hick's law through implementation of a tab bar can improve an App's success and usability, as if a user feels a preference towards an app, and if it is simple to use, they are more likely to enjoy the experience of interacting with it.

In addition, Figure 5.28 highlights how the recycling symbols feature no longer employs text recognition, and instead displays three different lists of symbols using a segmented view controller. One of the reasons that this design decision was made was because originally, the text recognition was used to read the materials listed on OPRL recycling labels. However, over the past few months, these labels have transitioned from those seen in Figure 5.29a, to those seen in Figure 5.29b, highlighting that the material can no longer be inferred by performing text recognition on the symbols. In addition, there is such a large range of labels that can appear on packaging, many of which appear side-by-side, that programming a machine learning model to be able to accurately, and potentially simultaneously, classify all of them was deemed as beyond the scope of this project. Therefore, the recycling symbols information view was changed to be similar to the search function; implementing a table view where users could scroll through the options, and then select one to learn more. Furthermore, this method of identification was designed to be an additional back up method to the computer vision waste classification method, and therefore it should have high reliability - something that the text recognition model wouldn't necessarily be able to provide.



(a) Old



(b) New

Figure 5.29: On Pack Recycling Labels

A final new addition to the app that can be seen in Figure 5.28 is tracking bin collections. The idea here was to allow users to program in their local waste and recycling kerbside collections, so that the app could implement reminders, using push notifications, to put their bins out. These reminders can be configured to be a one-off, or repeating weekly or fortnightly. This is an example of a prompt to achieve a certain behaviour design, as highlighted in Section 4.2.2.2. Furthermore, push notifications can encourage a user to re-visit an app, increasing its user retention rate.

5.3.2 Build

5.3.2.1 Waste Classification Model

The first stage of improving the waste classification model was to assess the performance of the initial one built in the first stage of implementation. Whilst this model achieved training, validation and testing accuracies of 100%, 79% and 81% respectively, once in-app, its performance was much poorer, achieving around 50% accuracy. It was therefore suspected that the original model had been overfitted to the paper and cardboard classes, as, when an object was incorrectly classified, it was instead classified as paper or cardboard 90% of the time. This is most likely due to the heavily imbalanced nature of the classes of the dataset, with paper taking up 23.5% of the images, and trash only 5%.

The model improvement process therefore aimed to address the class imbalance. This can be achieved in a number of ways, for example by:

1. **Under sampling**: Removing images from each class until all classes are the same size as the smallest class
2. **Over sampling**: Adding images to each class until all classes are the same size as the largest class
3. **Synthetic sampling**: Implementing data augmentation to, or synthetically creating similar versions of the images of the smaller classes
4. **Weighting**: Applying class weighting when training the model, essentially telling the model to apply more emphasis to the under-represented classes

Of these, the method chosen was class weighting, as this can be achieved easily by assigning an array of class weighting integers to the model when fitting it to the training data. The class weightings were calculated by considering the total number of images in the dataset, their current class distributions, and what the classes would look like if they were evenly distributed. That is, the dataset has 2,527 images, and therefore, if it were evenly distributed, each class would contain around 422 images. This value was then divided by the number of images in each class, to calculate the ratio of how under- or over-represented a class was. These calculations can be seen in Table 5.14, which were validated by sklearn.utils.class_weight.compute_class_weight function, which produced the same results.

Class	**N**	**Weighting**
Card	403	1.046
Glass	501	0.841
Metal	410	1.026
Paper	594	0.708
Plastic	482	0.872
Trash	137	3.093

Table 5.14: Calculating Class Weightings

It was now time to build the new, optimised machine learning model. As mentioned in Section 2.6, previously completed literature and research has highlighted the suitability of a Convolutional Neural Network for this application. To this effect, this style of model was therefore implemented. The architecture of choice was chosen to be relatively simple, as highlighted in Figure 5.30.

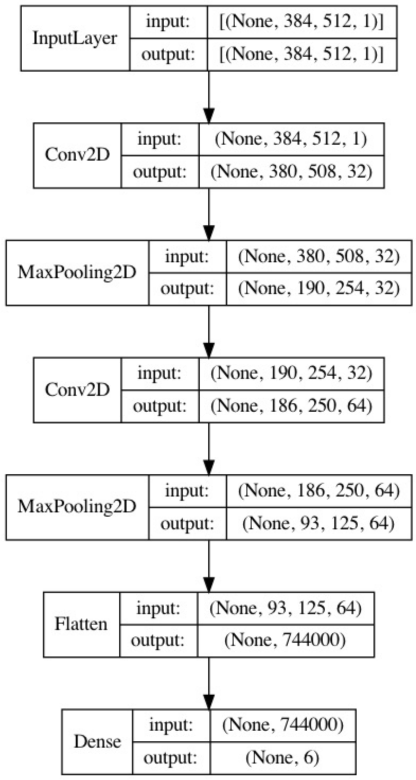


Figure 5.30: Model Architecture

This choice was made as the optimal number of layers and hence the depth of the architecture depends on the dataset. For example, natural language processing, or other more temporal datasets, of which TrashNet is not, often require deeper architectures. To this effect, the initial architecture was chosen to only have two convolutional layers.

Development then called for experimentation, altering various hyperparameters until an optimal accuracy was achieved. The starting point of each parameter can be seen in Table 5.15.

Loss	**Optimiser**	**Batch Size**	**Epochs**	**Dropout Prob**
Mean Squared Error (MSE)	Stochastic Gradient Descent (SGD)	50	5	None

Table 5.15: Initial Hyperparameters

These were then varied one-by-one, to investigate their effect on the model's classification accuracy. First to be varied was the batch size, which corresponds to how many subsets a dataset is split into when training. This is necessary as, especially with larger datasets, it is impractical to feed all images into the model at once. However, too small a batch size and the gradient of the loss function will have too much variation. An optimal size must therefore be found, where the gradient is updated an optimal amount of times during training such that the loss is minimised, and the accuracy maximised. The effect of varying the batch size on the model's loss and accuracy can be seen in Table 5.16[3].

Batch Size	50	75	100	125	150
Accuracy (%)	19.84	29.13	23.15	20.62	20.62
Loss	0.118	0.116	0.116	0.116	0.116

Table 5.16: The Effect of Varying Batch Size on Accuracy and Loss

Table 5.16 highlights a trend of increasing accuracy with increasing batch size, until a batch size of 75, after

[3]OOM Error stands for 'Out of Memory' Error - the GPU ran out of memory, which occurred due to the size and number of images. This is fixed by decreasing the batch size and therefore 150 was the maximum tested.

which the accuracy started decreasing again. This suggests that, above 75, the batches begin to be too large and thus the gradient of the loss function is not updated enough. Therefore, the batch size was chosen to be 75.

Despite the dataset containing 2,500+ images, the effect of varying the batch size was only observed up to 150 for two reasons. Firstly, a peak in accuracy had been observed at a batch size of 75. Secondly, at a batch size of 200, an out of memory error was experienced, due to the GPU running out of memory. This occurred due to the size of the images (512 x 384 pixels), and was fixed by decreasing the batch size, thus imposing an upper limit of 200. This was perhaps a restriction of running the code on Google Colab.

Next to be varied was the number of epochs, which represents the number of times that the data is passed through the network during training. Intuitively, increasing this increases the model's opportunity to learn, which should in turn increase the accuracy. However, this also causes an increase in computation time, as well as increasing the risk of the model being overfitted to the data. Therefore, as with the batch size, an optimal number of epochs must be found through experimentation. The results of this can be seen in Table 5.17.

Number of Epochs	5	10	15	25	50	75	100	200
Computation Time (s)	60	130	180	300	600	900	1200	2400
Accuracy (%)	29.13	23.47	31.65	32.60	34.33	37.48	41.10	45.67
Loss	0.116	0.115	0.115	0.114	0.112	0.106	0.103	0.104

Table 5.17: The Effect of Varying the Number of Epochs on Computation Time, Accuracy and Loss

As expected, this table highlights a trend of increasing computation time and accuracy and decreasing loss with increasing number of epochs. This means that a trade off must be made between accuracy and computation time. To this effect, the number of epochs was chosen to be 100.

Next, the optimiser used in the model was varied, experimenting with Stochastic Gradient Descent, Adam, Adagrad, Adamax and RMSprop algorithms. However, no improvement in accuracy or loss was achieved, and hence the optimiser for this model was kept as the Stochastic Gradient descent algorithm. This algorithm "estimates the error gradient for the current state of the model using examples from the training dataset, then updates the weights of the model using the back-propagation of errors algorithm" [128]. The amount that these weights are updated is known as the learning rate, and how much a previous weight update impacts the next update is known as the momentum. These are both configurable hyperparameters, and therefore the next stage was to tune these. The results from this experimentation can be seen in Tables 5.18 and 5.19. Initially, for the learning rate experiment, the momentum was kept at the default value of 0.0.

Learning Rate	0.001	0.01	0.05	0.1	0.2
Accuracy (%)	28.35	41.02	26.14	19.84	19.06
Loss	0.1152	0.1030	0.1474	0.1916	0.1906

Table 5.18: The Effect of Varying the Optimiser Learning Rate on Accuracy and Loss

Learning Rate	0.0	0.2	0.4	0.6
Accuracy (%)	41.02	30.71	29.29	28.35
Loss	0.1030	0.1151	0.1150	0.1143

Table 5.19: The Effect of Varying the Optimiser Momentum on Accuracy and Loss

Table 5.18 highlights a trend of increasing loss and decreasing accuracy with increasing learning rate, above a rate of 0.01. As with all previous hyperparameters, a middle ground must be found between small and large values - too small a learning rate means small weight updates and can potentially cause the model to hang. In contrast, too large a learning rate can cause the model to converge too quickly [128]. With the results of Table 5.18 in mind, the learning rate was selected to the original default value of 0.01. Table 5.19 highlights that similar trend of decreasing accuracy and increasing loss was also seen with increasing momentum, and therefore this was also kept at its default value of 0.0.

At this point, the hyper parameter variation had only produced a maximum accuracy of 41.02%. Therefore, with inspiration from Vohra [191], who also worked with the TrashNet dataset, the network architecture was redesigned, introducing a further convolutional layer, as well as multiple Dropout layers. These are regularisation layers that 'drop' randomly selected neurons with a given probability, reducing the sensitivity of the network to specific neuron weights. This allows for overfitting on data to be minimised. To this effect, the new architecture proposed by Vohra [191] can be seen in Figure 5.31.

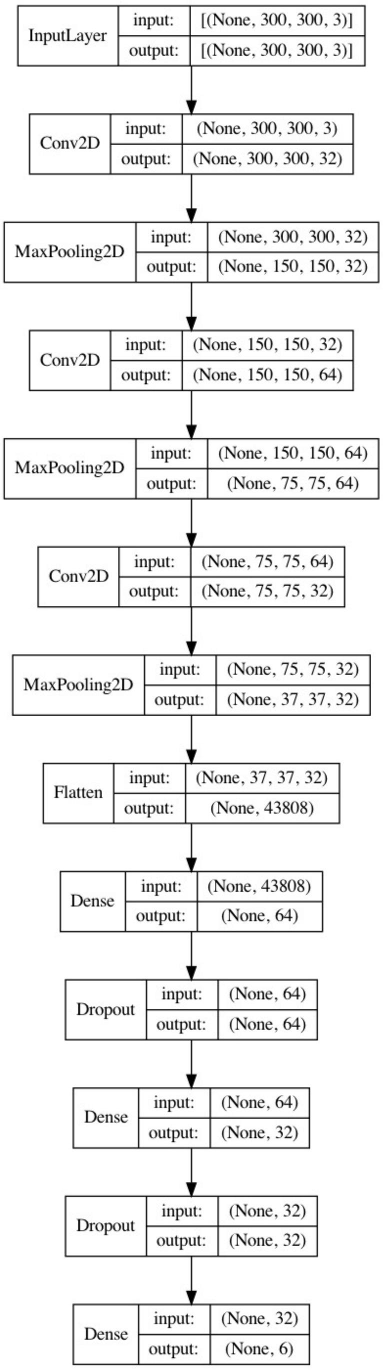


Figure 5.31: Updated Model Architecture

Alongside this architecture change, was the introduction of some pre-processing, again following in the path of Vohra [191], using keras' ImageDataGenerator function.

Loss	Optimiser	Learning Rate	Momentum	Batch Size	Epochs	Dropout Prob
MSE	SGD	0.001	0.2	75	200	0.2

Table 5.20: Optimised Hyperparameters

This architecture, in combination with the best performing hyperparameters of the previous rounds of experimentation, which can be seen in Table 5.20, were used to run one final experiment. Furthermore, the code was written in such a way that the maximum validation accuracy experienced over all epochs was regularly updated, and the model exported according to its state at the corresponding epoch. This ensured that the maximum accuracy was achieved. To this effect, on the final run, this model resulted in a training accuracy of 72.77%, and a validation accuracy of 70.91%, a vast improvement from the 19.84% achieved in the first experiment of this process. The variation of the training and validation accuracies with epochs of this experiment can be seen in Figure 5.32.

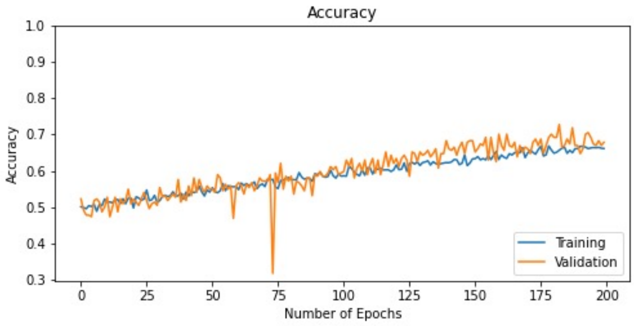
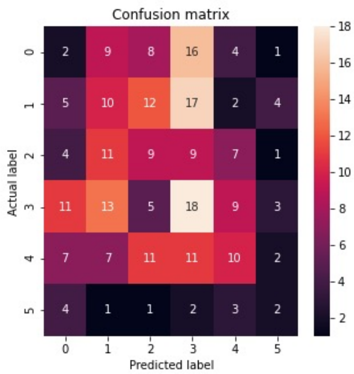


Figure 5.32: The Variation of Training and Validation Accuracy with Increasing Epochs

This highlights that, as expected, accuracy can be seen to increase with epochs. This is because the more epochs that had been completed, the further the model had gone through its training process. Furthermore, the further through the training process the model is, the more it has learnt, and therefore the greater its ability to accurately perform classifications.

A further analysis of this model can be achieved through inspection of the confusion matrix, shown in Figure 5.33a.



(a) Confusion Matrix

	precision	recall	f1-score	support
cardboard	0.18	0.15	0.16	40
glass	0.16	0.16	0.16	50
metal	0.17	0.20	0.18	41
paper	0.23	0.29	0.26	59
plastic	0.31	0.23	0.27	48
trash	0.08	0.08	0.08	13
accuracy			0.20	251
macro avg	0.19	0.18	0.18	251
weighted avg	0.21	0.20	0.20	251

(b) Classification Report

Figure 5.33: Further Analysis

In Figure 5.33a, classes 0 through 5 represent cardboard, glass, metal, paper, plastic and trash respectively. The matrix shown highlights that there is still some overfitting with respect to the paper category - with cardboard and glass often getting predicted as such. This is confirmed by the class-specific analysis of the model's performance that is detailed by the classification report (generated using sklearn.metric.classification_report), which can be seen in Figure 5.33b. At this point, it is important to note that the cardboard mis-classification is not of high importance, as classifications of paper and cardboard both result, by design, in RecycleHelper displaying information for paper *and* cardboard together. This design decision was made as they are generally recycled together, and involve the same instructions. Therefore, future work should concentrate on increasing the accuracy of glass mis-classification. However, the performance of this model was deemed to have provided an improvement on the basic model implemented at the beginning of the project.

The python notebook for this experimentation can be found in the github repo at the path:

/Machine Learning/Optimised Model/wasteclassification.ipynb

5.3.2.2 App Development

The first step of implementing the design involved dividing the features into the 5 views controlled by the tab bar controller. This categorisation can be seen in Table 5.21.

Tab	Features	New?	Tab	Features	New?
Home	Personalised Welcome Message	✓	Scan	Waste Classification	
	Randomised Recycling Fact	✓		Usage Instructions	✓
	Progress/Usage Tracker	✓		User-Controlled Scan Start	✓
	Kerbside Collections Tracker	✓	Symbols	List of Recycling Symbols	✓
	Settings	✓		Recycling Tracker	✓
Search	List of Items			"Find Your Nearest..." Feature	✓
	Search Functionality		Locate	Nearest Recycling Centres	✓
	Favourites	✓		Nearest Supermarkets	✓
	Recycling Tracker	✓		Nearest Charity Shops	✓
	"Find Your Nearest..." Feature	✓		Directions to Nearest Facility	✓

Table 5.21: Version 3 Features by Category

This table highlights the volume of new features that were added to this version of the app, from listening to feedback from the two previous rounds of testing. The storyboard for the implementation of these in XCode can be seen in Figure 5.34. This figure shows a few features that were implemented that were not initially included in the mockup, namely:

- **Personalised Welcome Message**: This was implemented to make the app more welcoming, and subconsciously improve a user's perception of the app.
- **Randomised Recycling Facts**: RecycleHelper has been designed to help improve recycling knowledge and motivation. In one respect, this means literally *how* to recycle actions. However, on the other hand it can be used to educate more about the positive effect of recycling and why it is necessary. To this effect, a database of (initially) 40 recycling facts was compiled, and a random one is displayed each time the user's homescreen is loaded.
- **Start/Stop Button for Scanning**: Previously, it was observed that the waste classification feature often performed classification too quickly - i.e. before the user was ready. This often resulted in incorrect classification, as well as user frustration, negatively impacting the uesr experience. To mitigate this, a trigger button was implemented to ensure that scanning of an object only occurred after a user's command.
- **Activity Indicator during Scanning**: This is a spinning indicator that is displayed when the app's machine learning model has been activated. The idea here is to take advantage of the placebo effect (as explained in Section 2.4.3.4) so that users have an indication that the model is working, and are perhaps more patient while it works, incase the classification isn't as quick. This in turn can allow the user to feel more in control, and improve their experience, when in reality the speed of classification is out of their control.
- **App Settings**: This screen was implemented to allow users to easily modify various app variables, such as their personalised message, whether it was animated, and their recycling goal. Furthermore, this was designed to mimic the UI of the settings app designed by apple, as another implementation of the Mere Exposure Effect.
- **Progress Tracker**: This keeps track of how many items a user has recycled by asking them to press a 'I recycled it' button each time they recycle something. The default goal is to recycle 50 items, but, as mentioned in the previous bullet point, the user can customise this in settings. Each time the goal is met, a congratulatory message is displayed and the count reset. This aims to motivate the user to recycle more, and in future versions will include some sort of badge/achievement system as well as leaderboards, to exploit gamification and social motivation techniques.
- **Search Favourites**: This feature was added so that items that users regularly recycle can be accessed quickly and easily, rather than having to scroll through potentially 100s of items. The user can easily add or remove items from this, and toggle whether the screen shows just their favourites, which are identified by a star (another Mere Exposure Effect application, as this symbol is commonly used throughout applications to indicate favourites), or all items. The search bar function introduced in version 2 works on both views.

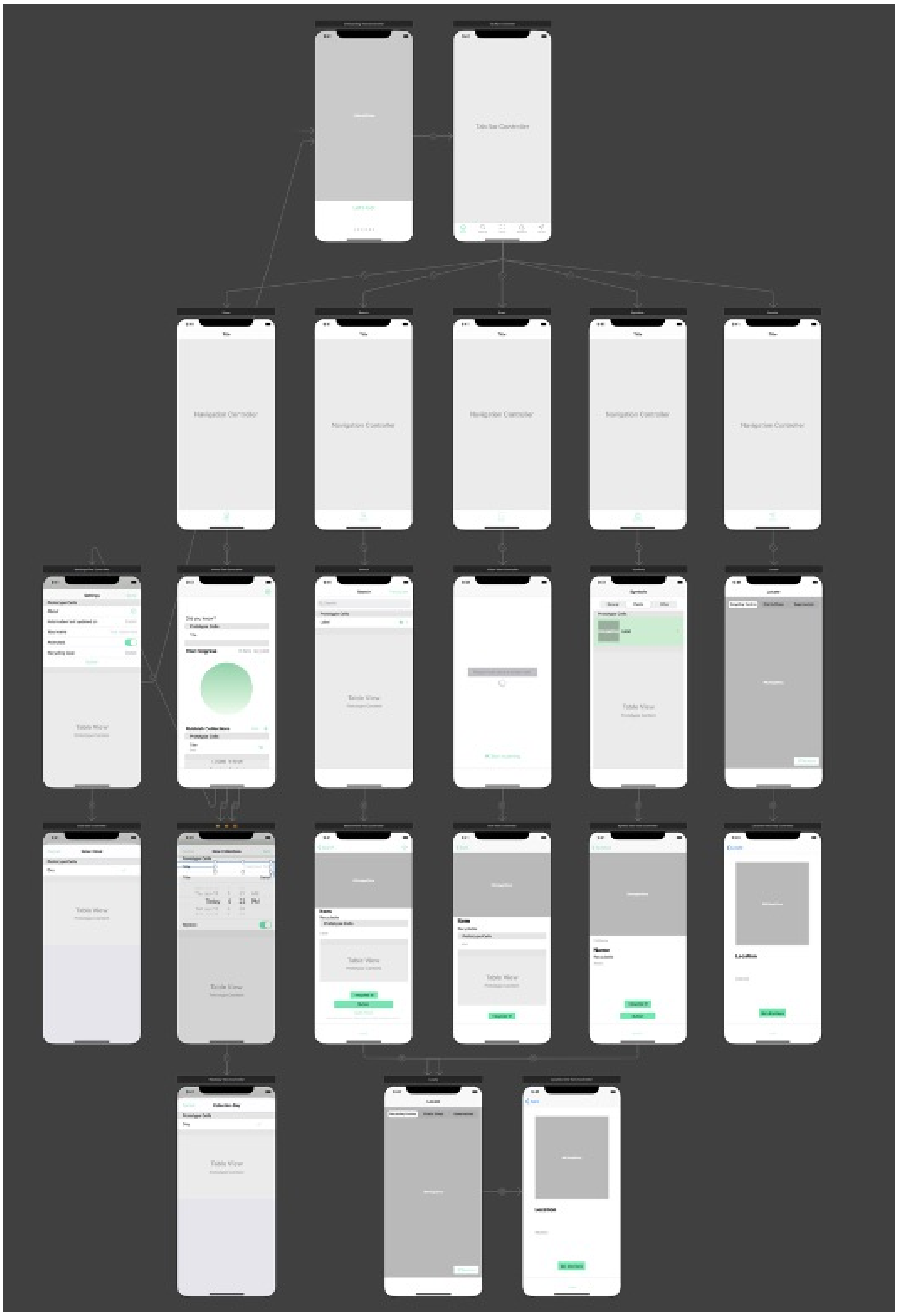


Figure 5.34: Storyboard for App Version 3

To implement the location-specific information mentioned in Table 5.13, the first step was to move the recycling information off-device. This was vital, so that the app did not have to contain a specific variant of recycling information for each location that the app was designed to work for. If this had been implemented on-device, the memory resources required by the app would increase linearly with the number of locations added, which is extremely inefficient in terms of conserving resources. Furthermore, off-device information allows for information to be updated as it changes, and new locations to be added, without having to release a new app version each time. The initial plan was for the app to access RecycleNow's database via an API. However, potentially due to COVID-19, the team has been unresponsive to requests about providing access to this. Therefore, a proof of concept method was implemented in its place. Namely, a database was built using Google's Firebase Realtime Database, and specific recycling information manually added for the location of each of the users that participated in the final round of usability testing. As suggested by the name, this information updates in real time, without the user being required to restart the app. It also offers offline handling for when the user's phone is unable to connect to the internet.

Each user's location was obtained using the Core Location Framework, which "uses instances of the CLLocationManager class to configure, start, and stop the Core Location services" [192]. A location request returns the user's latitude and longitude coordinates, which are then converted into their city and the first half of their postcode. These two values are combined as a string, which is used as the key under which the relevant location-specific information is store in the database. When implementing this, a choice between three location-access methods was made:

1. Visits location service
2. Significant-change location service
3. Standard location service

The visits location service "delivers location updates when the user has spent time in one location and then moves on" [193], minimising battery consumption but sacrificing accuracy. In contrast, the standard location service "delivers the most accurate and immediate location information" [193]. This is intuitively very accurate, but also extremely battery consuming. Clearly these two options offer a trade off between power consumption and location accuracy. However, in between these two methods is the significant-change location service, "a power-friendly alternative for apps that need to track the user's location but do not need frequent updates or the precision offered by GPS" [193]. This method offers a midpoint between the impact of the power consumption of the standard location service, and the accuracy of the visits location service, and was therefore selected for use in RecycleHelper.

The final location-specific feature was showing the user's nearest Recycling Centre, Supermarket or Charity Shop. These categories were shown on a map by using a MKLocalSearch for nearby places fitting these descriptions. The map view was then configured to zoom in/out to show the nearest search results. These results were those that would be displayed if the search was made in Apple Maps, and were returned with just the facility name and address, as not all locations provide other information, such as phone numbers, websites or opening hours.

The location functionality of the app is only enabled if the user has granted RecycleHelper access to their device's location. Without this permission, the app displays an alert stating that permission has not been granted, with an option to redirect the user to the location in their phone's settings where they can grant this permission. However, if the user doesn't follow this path and chooses to keep the permission as denied, then the app displays non-location-specific information that is saved under the "Default" key in the database. The same process of handling privacy access (i.e. displaying an alert message and potentially re-directing to settings) is implemented for both camera and push notification access for this app.

Finally, to ensure that the user's information was retained even if the app crashed, was restarted or the device itself restarted, data was stored using user defaults, Apple's method of providing persisting data. Examples of data stored in this manner are whether the app had been visited before (as the onboarding was designed to only be shown on first visit), the user's personalisation, waste collections and reminders, and search favourites.

The screenshots for this final app version can be seen in the ReadMe of the git hub repository: https://github.com/rch16/RecycleHelper. A summary of the key methods used to implement this app version can be seen in Table 5.22.

Functionality	**Methods**
Live Capture	AVCaptureSession
Image Registration	VNTranslationImageRegistrationRequest
Image Classification	Vision Framework
Information Database	Google Firebase Realtime Database
Location Specific Information	Core Location Framework
Nearest Facility	MKLocalSearch
Pins on Map	MKAnnotationView
Push Notifications	User Notifications Framework
Data Persistence	User Defaults
Persuasive Design	Mere Exposure Effect, Hicks Law, Placebo Effect, Behaviour Design, Personalisation

Table 5.22: App Version 3 Implementation Methods

5.3.3 Testing and Analysis

5.3.3.1 Performance Testing

As with the previous two rounds of testing, the aim of this section is to highlight the app's performance in terms of the metrics outlined in Section 4.3.2. Evaluating an app in such a way is important to ensure that its performance is optimal, i.e. it operates efficiently, doesn't take up unnecessary resources and doesn't drain the user's phone battery. An optimally performing app is important, as badly performing apps may be unresponsive, crash frequently, and potentially render a user's device unusable if it drains the battery - all of which would negatively impact the user experience.

As before, this section starts with a comparison of the launch times of each version for 7 test cases, which can be seen in Figure 5.35.

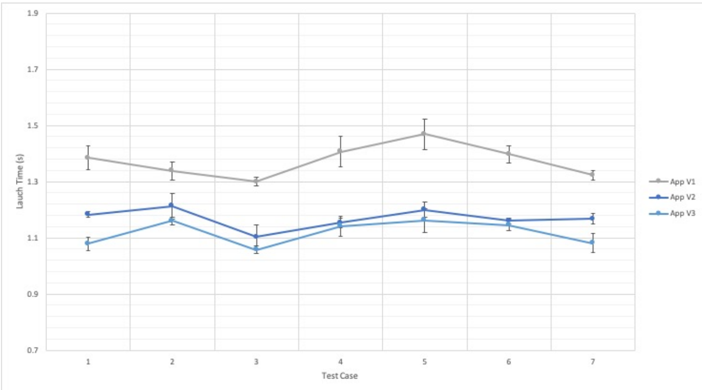


Figure 5.35: Comparison of Launch Times of all App Versions

This figure highlights that, in the same manner as how development of Version 2 improved app launch time compared to Version 1, further launch time improvement was achieved during development of Version 3. However, the change was not as large, potentially due to the introduction of features such as location monitoring, and communication with an off-device database. Whilst the launch time could be further improved, the effect on the user is mitigated through implementation of a launch screen that is displayed to the user whilst the application loads. To this effect, many users will most likely not notice the one second that is spent launching the app, as multiple other well known and successful apps, such as Whatsapp and Instagram, also display a similar launch screen, for similar, if not longer lengths of time.

Following from the metrics listed in Table 4.3, next, the number of app crashes was monitored. This section of analysis was completed alongside the usability testing detailed in the following section, as catching potential crash causes is often achievable only by testing the app on multiple phone models and software versions. This experimentation revealed, in contrast to the previous app versions, that there were a few features that were initially causing the app to crash. These are highlighted in Table 5.23.

Build Number	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Number of Crashes	0	3	2	1	6	7	0
Resolved?	n/a	✓	✓	✓	✓	✓	n/a

Table 5.23: App Version 3 Crashes

Table 5.23 details the number of crashes experienced by each build of RecycleHelper Version 3. The reason there are 7 builds is because each build was created to provide a fix to the crashes experienced in the previous build, the cause of which was debugged by looking through crash logs and recreating the error using the XCode simulator. Causes of these crashes were found to be due to CoreLocation access permission, and handling of user's search favourites. The final row, as well as the lack of crashes that occurred in the final build highlights that all crash causes were identified, located and resolved, to produce a crash-free application.

Next to be analysed was the CPU and thread usage, as this version of RecycleHelper had seen the introduction of several new, potentially CPU-resource-expensive, features, such as location services. The CPU usage of the app whilst open and whilst running in the background, anlaysed using XCode's debug tab, can be seen in Figures 5.36 and 5.37 respectively.

Figure 5.36: CPU Usage of RecycleHelper Whilst In Use

Figure 5.37: CPU Usage of RecycleHelper Whilst Running in the Background

Both figures highlight that the CPU usage of RecycleHelper is within an acceptable operating range, and lower than that of other processes and apps that were running on the device at the same time. This low usage highlights that RecycleHelper is unlikely to cause the device to become slow or unresponsive. The same observations could be made for the memory usage of the app, as highlighted by Figure 5.38.

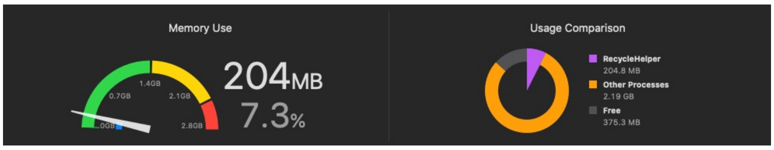


Figure 5.38: Memory Usage of RecycleHelper

This highlights that around 200MB pf memory is used by RecycleHelper, used for storing local variables, such as the user's name, their bin collections and their favourite search items. Intuitively, as users add or remove collections or favourites, this usage will change, but not by much. This is assessed to be an acceptable level by comparison to the available storage of current iPhone models, which generally range from 64GB upwards, of which 200MG is less than 0.5%.

Finally, the energy usage of RecycleHelper was analysed, again using XCode's Debug tool. This can be seen in Figures 5.39.

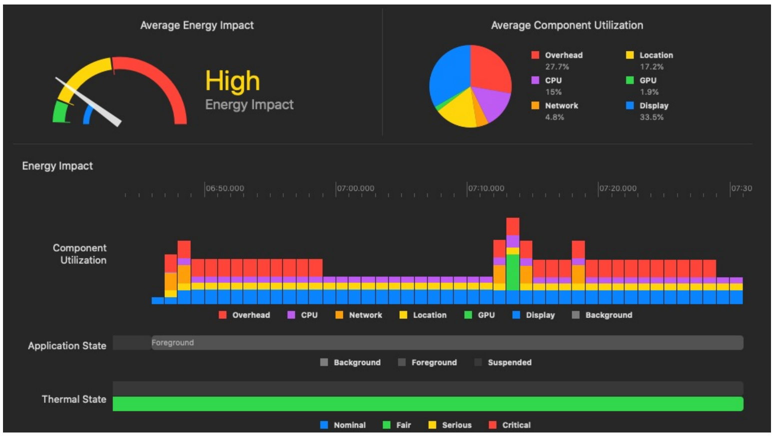


Figure 5.39: Energy Usage of RecycleHelper Whilst In Use

This clearly shows that RecycleHelper is classed as a high-energy-usage application. This was expected, due to it offering features that use location and the camera, both of which generally require more energy. However, as discussed in Section 5.3.2.2, these were implemented using energy-efficient techniques, such as a significant-change location service instead of the standard one.

5.3.3.2 User Testing

The 21 participants that completed the second round of user testing were asked to also complete this final round of testing, in order to allow for an assessment of any potential change in usability that may have occurred from adding in a variety of new features. Of these 21, 16 responded, and the unique locations of these respondents can be seen in Figure 5.40. The age range of participants was the same as that for round 2; 18 to 58.

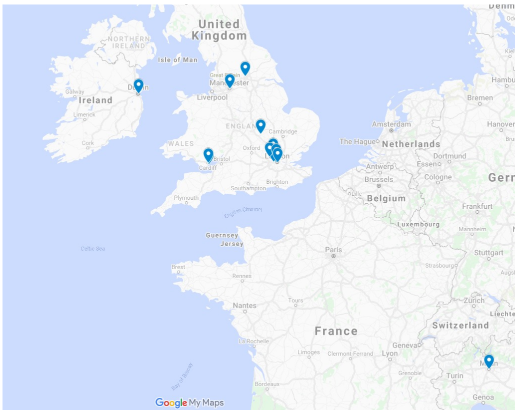


Figure 5.40: Unique Locations of Round 3 Testers

The first step to designing this round of testing was to generate a list of information to discover/goals to meet:

1. What effect does the app have on a user's recycling knowledge?
2. What effect does the app have on a user's recycling motivation?
3. How do users feel about their onboarding experience?
4. How easy and intuitive is the app to use?
5. Does anything not work as expected?
6. How easy is it to understand the presented information?
7. Does the app provide helpful, valuable and interesting information?
8. Did the app meet user's expectations?
9. What feature do users prefer?
10. How helpful is the app?
11. How likely are users to use this app?
12. Would users change anything about the app?
13. How would users rate the app if they had found it on the app store and downloaded it?

A concern from the previous round of testing was that users might feel awkward about giving a bad score to the app in a face-to-face (video) interview. To this effect, to remove any potential bias, this round of testing was implemented using three forms:

1. Assessing user's recycling knowledge before using the app (not anonymous)
2. Testing out the app and assessing usability (anonymous, to remove bias)
3. Assessing user's recycling knowledge after using the app for a period of time (not anonymous)

The purpose of forms 1 and 3 were to asses the effect that the app may have on a user's recycling knowledge, i.e. goal 1. Form 2 was then designed to answer the remaining questions. Implementing this round of testing also allowed for users to try out the app in their own time without any influence, as well as for tests with multiple users to be carried out simultaneously, maximising the amount of work that could be completed in the limited remaining time frame. The full questionnaires can be found in the Git Hub Repo: **/Testing/Round 3.** Due to the fact that all participants of this round of testing had also taken part in the previous round, the questions regarding recycling knowledge, motivation and similar were not repeated.

For the second form, i.e. the usability part of the testing, each participant tried out all features of each view, and each was asked to provide ratings for ease of use and understability using the Likert scale. This created an average score for ease of use and understandability of information for each view for each user. These scores could then be averaged over the entire testing cohort, to find scores for each view, as seen in Table 5.24. The fact that each view was rated according to every feature, and the overall usability rated according to the average of all views ensures that all features of the app are considered. This in turn increases the accuracy of the usability score.

View	Ease of Use	Understandability	Usability
Onboarding	-	-	4.22
Home screen	4.49	-	4.49
Search view	4.64	4.59	4.53
Scan view	4.65	4.28	4.46
Symbols view	4.40	4.66	4.58
Locate view	4.52	-	4.52
Average	**4.56**	**4.53**	**4.529**

Table 5.24: RecycleHelper Version 3 Usability Scores

Table 5.24 highlights that the onboarding screen was not rated for ease of use or understandability, as this information was not relevant. Instead, users were asked whether they enjoyed the onboarding, how helpful they found it, and how they found it as a introduction to the app. These scores were then averaged to find the overall usability score. On a similar vein, the home and location views were not rated according to understandability, as they did not present information to understand, such as how to recycle an item. Therefore, their ease of use scores were used as the overall usability score. For all other views, the ease of use and understandability scores were averaged to generate the overall usability score.

The scores in Table 5.24 were used to quantifiy each tester's perception of specific features. In contrast, to assess their overall perception of the app, the following questions were asked:

- What was your favourite feature?
- How helpful did you find the app?
- How likely are you to use the app?
- How clear and helpful was the information presented?
- How easy it was to navigate the app?
- How much sense did the layout make to you?
- What would you rate the app, if it was a (potentially) random one that you had found on the App Store?

Figure 5.41 shows the responses of the first question - the users' favourite features. This highlights that the symbols feature was the most popular, with the locate and scan features coming a close second and third respectively. When users were asked why they voted for the symbols feature, they responded that it had the widest options of items, and provided the most detailed information. This highlights that, in future versions, the scan feature should be further developed to recognise a wider range of items, and the information displayed should also be expanded.

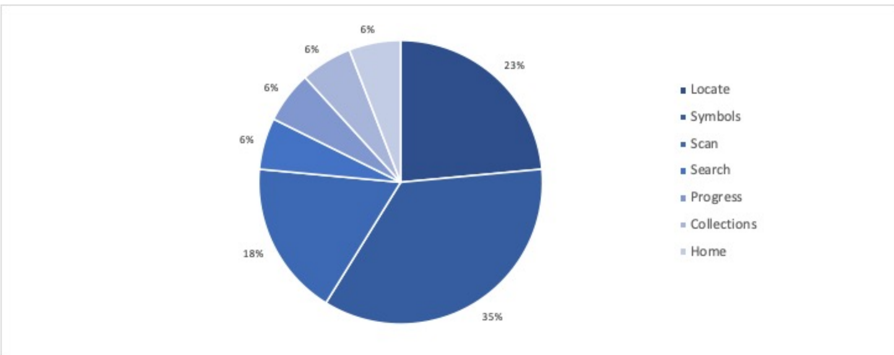


Figure 5.41: Favourite Features

The remaining questions detailed in the above list were used to form each user's satisfaction score, the average of which was taken to calculate the testing cohort's overall satisfaction with the app. This was calculated to be **4.583**. Due to the anonymous nature of the second form, no comparisons could be made with a user's answer from the previous round of testing, or with their recycling knowledge score. However, the fact that the calculated usability score was higher than that calculated for the second round of testing means that it can safely be assumed that no bias was present in the users when answering questions regarding the app. Furthermore, it highlights the fact that the introduction of multiple new features into the app extended its functionality, and hence, its usability.

The final point of consideration for this section is the impact of the app. In Section 3.2, it was determined that this app should increase a user's recycling knowledge, as well as their motivation. To this effect, these were both measured during this final round of testing.

First, a users motivation to recycle. This was assessed by asking each test participant how likely they were to recycle before they had access to the app, and then asking the same question again, after they had access for a period of time. This is most relevant for users who rated their motivation as below the maximum value of 5, as they have room for improvement, whereas intuitively those that initially responded that they were 'extremely likely' to recycle (5/5) were not going to show an increase. To this effect, the average increase in motivation was analysed for two groups of participants; all test participants (Group A), and those that rated themselves as < 5 for motivation (Group B). This can be seen in Table 5.25.

Likeliness to Recycle	**Group A**	**Group B**
Before using the App	4.188	3.818
With access to the App	4.688	4.545
Change	+0.5 (10%)	+0.727 (15%)

Table 5.25: The Impact of RecycleHelper on Test Participant's Likeliness to Recycle

Not shown in Table 5.25 is the fact that RecycleHelper increased the motivation to recycle of *all* members of Group B. However, this table does highlight that the magnitude of increase in motivation for this group was 15%, a notable change. When considering all test participants, this change decreases to 10% as those already the most likely to recycle intuitively did not change, affecting the average. However, a 10% average increase still provides conclusive evidence that RecycleHelper was successful in its attempt to improve user motivation.

Second to be assessed in terms of RecycleHelper's impact was each user's recycling knowledge. As mentioned previously, this was assessed by test participants completing two recycling knowledge tests - one before using the app, and one after using it for a while. The first comparison made was how users perceived their recycling knowledge with how the initial recycling knowledge test assessed their knowledge to be. This can be seen in Figure 5.42, which highlights that all but 1 of the testers (92.31%) perceived their recycling knowledge to be higher than it actually was.

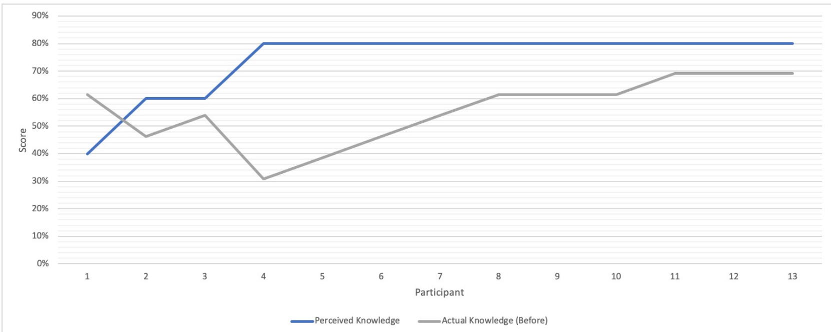


Figure 5.42: Test Participants Perceived vs Actual Recycling Knowledge

The next comparison was a user's level of recycling knowledge before using the app, with their level of knowledge after. The average score in the first recycling knowledge test was 55.62%, or 7.23 out of 13, compared to an average of 86.39%, or 11.23 out of 13, in the second. This highlights that the average impact of RecycleHelper in terms of a user's recycling knowledge was 30.77%, a significant change. In fact, all test participants were able to improve their recycling knowledge test score after having access to the app, with a minimum change of 1, to a maximum of 8. Furthermore, the standard deviation in test scores was 1.513 before, and 1.191 after, also highlighting that RecycleHelper improved the consistency of scores. A graph highlighting each user's test performance can be seen in Figure 5.43.

Taking the scores of a 15% increase in motivation and a 31% increase in knowledge into account, it can therefore be concluded that, as planned, RecycleHelper succcessfully had a positive impact on each user's motivation and ability to Recycle.

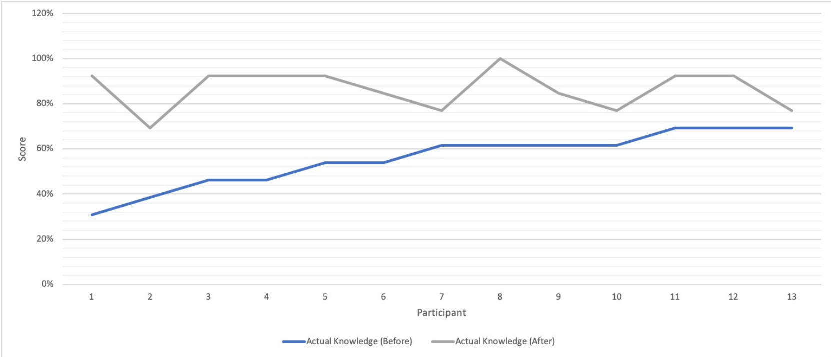


Figure 5.43: The Impact of RecycleHelper on Test Participant's Recycling Knowledge

5.4 High Level Overview

The aim of this section is to provide a high level overview of the final design of RecycleHelper that was the result of the three rounds of iterative development documented in this chapter. To this effect, the design hierarchy and the framework of the application can be seen in Figures 5.44 and 5.45 respectively.

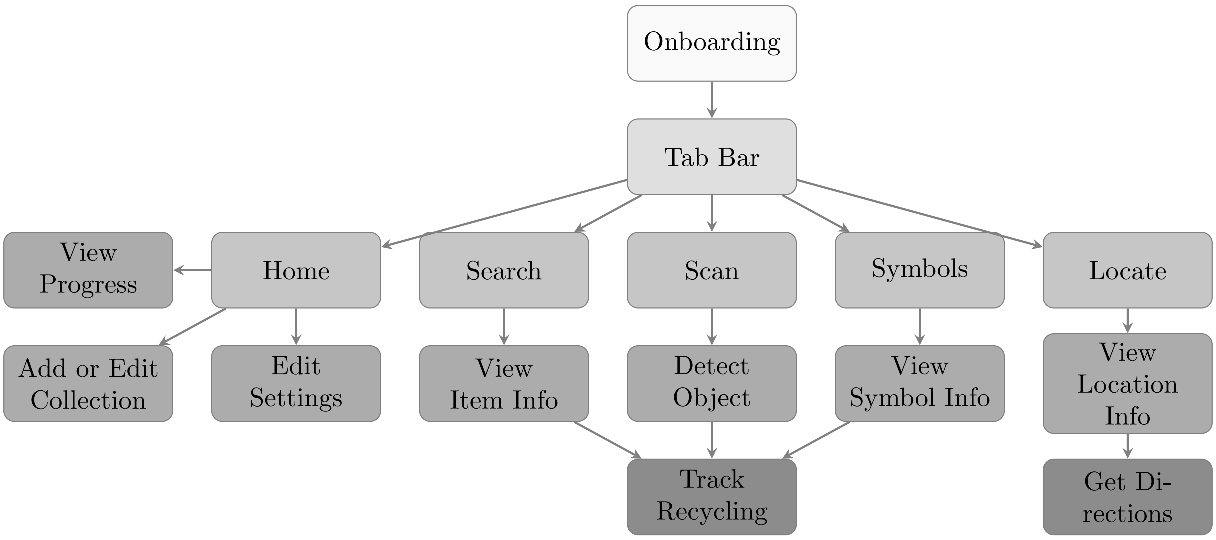


Figure 5.44: RecycleHelper Design Hierarchy

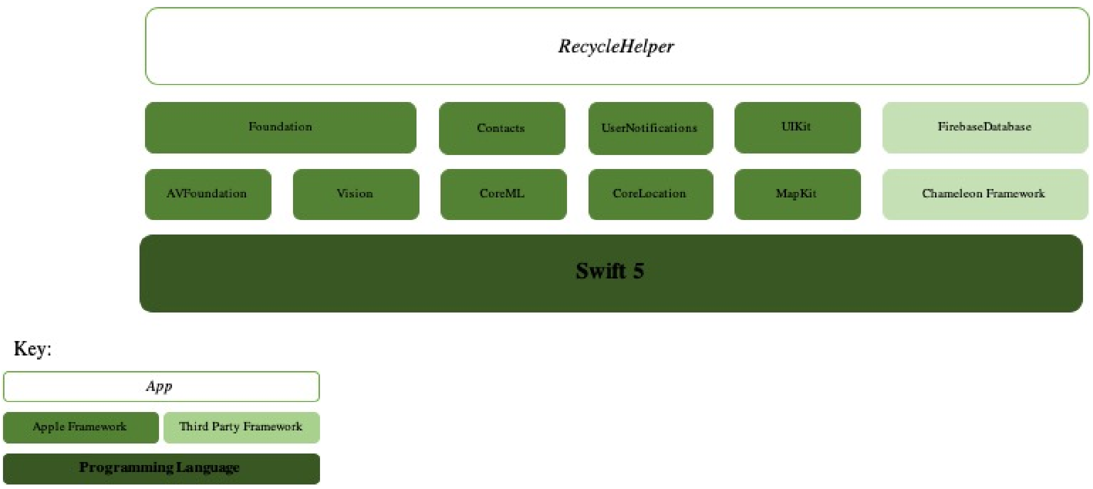


Figure 5.45: RecycleHelper Framework

Chapter 6

Evaluation

The aim of this chapter is to provide a critical evaluation of the work completed throughout the project duration, as well as to evaluate the project's success. To this effect, this chapter can be split into three sections; The first section will assess whether the project title, "a persuasive smartphone app for improving recycling performance", was met, as well as compare the outcome to the original project specification defined in Section 3. Secondly, the final proudct will be compared and contrasted against solutions currently available to consumers. Finally, the rounds of usability testing will be analysed and evaluated, with key findings highlighted.

6.1 A Persuasive Smartphone App

The high-level aim of RecycleHelper was to produce a persuasive application that could work to improve a user's recycling behaviour. That is, their motivation to recycle, and their ability to do so correctly. This was then extended through the project specification in Section 3, where three problem statements were defined, namely:

1. Lack of standardisation of recycling
2. Lack of availability of information
3. Lack of motivation to recycle

Project deliverables were then devised to combat these problems. This part of the evaluation assesses whether the final version of RecycleHelper satisfied the project title and achieved the project deliverables, and assess any limitations of the app.

6.1.1 Satisfaction of the Project Requirements

Whether RecycleHelper satisfied the original project requirements can be assessed by first revisiting Table 3.4, where the desired features were proposed and classified according to the 'MoSCoW' technique.

6.1.1.1 Waste Classification

This feature was proposed in order to provide a quick and simple method of classification for users with little recycling knowledge and/or ability. To this effect, using this feature was designed to be simple and require minimal effort; a user must simply place an object in front of the device camera and provide the 'start scanning' command through interaction with a UI button. In terms of usability, this feature met the requirements, rating an average usability score of 4.416 from participants in the final round of testing, and comments such as "it saves time looking up the item properties", "it was very easy and quick to use and gave me all the info I needed" and "so easy - just point and it'll tell you what it is" throughout the testing process.

The classification model was implemented through iterative design and development of a machine learning model, with the result being a Convolutional Neural Network (CNN) model capable of achieving high levels of accuracy. However, a difficulty experienced in this implementation was accurate specific classification of item materials, as this is what defines whether an object is recyclable or not. A specific example of this is types of plastic - often it is nearly impossible for someone who is physically holding an object to identify what type of plastic it is made out of. This therefore makes correct classification by a machine learning model unlikely. This model was therefore implemented as a proof-of-concept, predicting an object's general material, but not the object specifically. However, this can lead to users being presented with potentially misleading information. For example, a plastic bottle and the plastic film from some food packaging would both be *correctly* predicted as members of the plastic class by the model, which in turn has been defined as a recyclable category. However, plastic bottles are recyclable, and plastic film is not, a fact which is presented to the user when they search for an item specifically using the search feature.

The current model and its implementation should therefore be interpreted more as an education tool for users to learn about what materials items are, rather than using the result as a definitive answer regarding whether an item is recyclable or not. This opens up an area of future work that aims to change how the feature is implemented, in order to improve identification of items that are achieved using this technique. An example of such an implementation can be seen in Figure 6.1.

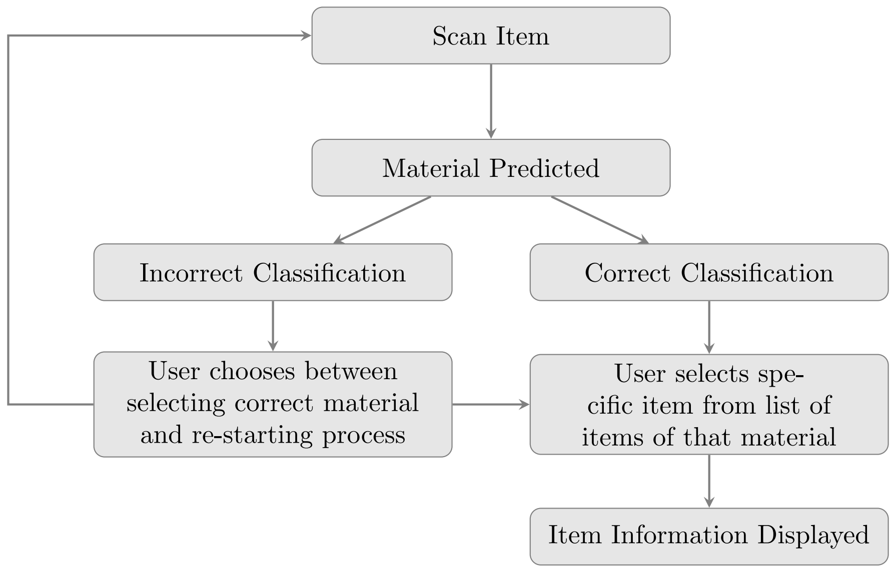


Figure 6.1: Workflow for Proposed Improvement of Classification Model Implementation

A further limitation of the model was the choice to use the TrashNet dataset. This choice was made as building a dataset by hand was not feasible given the project time frame and resources, and TrashNet was the most suitable dataset available. However, the majority of the images in the dataset were of objects extremely close up. This meant that, whilst the model could perform accurate classification if a user held the object close to the camera, usability testing observed that this was not the default approach that users took when scanning an item. Instead, they tended to position the item and phone such that the entire item fit within the preview frame displayed on-screen. This behaviour resulted in an increase in incorrect classifications. However, this issue could be minimised in future work, by pre-processing the image such that it was cropped to show only the material, before being analysed to the machine learning model

Finally, such a classification method cannot be used in low light or when it is dark. Whilst users will arguably not be recycling in the pitch black and/or the middle of the night, this fact still provides a limitation to the occasions when users can make use of this feature. For this reason, future work should look into providing a flash feature for when the user wishes the app to perform classification, but the lighting is unsuitable.

Despite these observations, it can still be concluded that this requirement was satisfied, as RecycleHelper does provide a waste classification feature that uses computer vision.

6.1.1.2 Backup Identification Method

Given the limitations discussed above, it became increasingly clear that the project requirement to have a back up identification method should evolve from a should-have feature, into a must-have, in order to preserve the app usability. However, originally the back-up method of identification was expected to potentially be a barcode scanning feature. This was not implemented for a number of reasons:

- Barcodes are often specific to the shop of origin, and therefore the same product from two different sources could have different barcodes, making it difficult to compile a resource capable of handling all potential barcodes.

- Barcode information is not always available for access.

- Not all products have barcodes, for example a piece of paper, or a plastic cup. As the back-up identification method was designed to provide functionality when the object detection software could not, implementing this feature would still mean that a range of items could not be identified.
- As with the waste classification feature, barcodes would not be able to be scanned in low light.
- As they are printed with ink, barcodes can potentially fade with use, or alternatively the label could get creased or ripped. Furthermore, this is likely to occur when an item is cleaned in preparation for recycling. If this happens, it would be difficult for the software to read the barcode, rendering the feature useless.
- Most importantly, whilst barcodes provide information about item prices, and potentially sources or ingredients, they do not typically contain information about the packaging or its material, making it difficult to infer whether the item can be recycled.

Therefore, the back-up item identification method was implemented in the form of the combination of an item search function and a feature to understand what various recycling symbols represent. Essentially, the symbol feature was chosen as a replacement for the barcode identification method, to allow the information provided by producers to still be utilised.

Alongside allowing for increased accuracy of item classification, and improved recycling of more unusual items, the search feature provides suggestions of more sustainable alternatives for a variety of items, and the symbols feature also provides education about packaging symbols that are often mis-interpreted as related to recycling. With this taken into consideration, it can therefore be concluded that this project requirement was satifisied.

6.1.1.3 Recycling Information

Both the waste classification and back-up identification methods provide detailed information about the item, namely:

- Item name
- Whether it can be recycled
- How it can be recycled

As detailed in Table 5.24, when users were asked to rate the understandability of the information presented by each of the identification methods (search, scan or symbols), all features performed well. However, closer inspection of the scores highlighted that the search and symbols features scored higher, with scores of 4.59 and 4.61 respectively, compared to a scanned item information understandability rating of 4.23. This result was expected, as both the search and symbols features enable a more specific classification to be made, and the more specific the item, the more detailed the recycling instructions can be. An example of this, when comparing the search and scan features, is an envelope. The scan feature would classify this as paper and cardboard, and then present a list of instructions regarding general recycling of paper objects. In contrast, the search feature would allow the user to learn specifically about how to recycle the envelope, and discover information such as the fact that the plastic insert window does not need to be removed, and instead the envelope can be placed, as is, in the recycling. A further example would be scanning a plastic item vs finding and looking up its plastic resin code. As mentioned previously, the scan feature would classify this as plastic, deem it as recyclable and tell the user simple, generic plastic-related steps, such as to make sure that is clean. However, different types of plastic must be treated differently, and looking up the resin code in the symbols feature provides more information about this. For example, type 1, or PETE plastics, are generally easy to recycle, but type 3, or PVC plastics are generally difficult to recycle and therefore not accepted in recycling collections. Furthermore, the scan feature is not currently able to deal with objects made from multiple materials. In contrast, the search feature offers information on such products, such as a pringle crisp tube, and instructs the user how to deal with each material.

It can therefore be seen that, in general, the search and symbols features provide more specific and detailed instructions to the user, under the assumption that if a user wanted a general ‘recyclable’ or ‘not’ conclusion, then they would just scan the object, but if they wanted to learn more, or wanted to be provided with step-by-step instructions, then they would use a combination of the search and symbols features.

With this in mind, despite how it may be lacking in some areas, it can be concluded that the scan feature provides information specific to the material that has been detected, as detailed as necessary in the project requirements. However, it can also be concluded that, due to the specificity of information provided by both the search and symbols features, RecycleHelper is actually able to provide *in-depth* information about how to recycle specific items. Therefore, this project requirement was determined to be satisfied to a high level.

6.1.1.4 Location-Specific Information

As mentioned previously, it was originally envisioned that RecycleHelper would be able to interface with RecycleNow's database, to ensure that the recycling information is updated frequently and accurately by the local councils and boroughs themselves. However, due to a lack of response from the RecycleNow team, it was not possible to implement this.

In its place, a proof-of-concept feature was implemented, one that interfaced to a different database, where location-specific recycling information was stored under keys describing each location's city and postcode. However, a limitation of this method was that maintaining this database is inefficient and time consuming, as it involves manually researching and entering each item's recyclability status and instructions, each time functionality for a specific location is to be added. Furthermore, this information does not update automatically if a location's recycling instructions changes, and therefore regular checks must currently be performed to ensure that all information is up to date.

However, despite these limitations, the feature was found to still be able to provide location-specific information to all usability test participants, just with a small amount of preparation beforehand. The method implemented also removes the step present in RecycleNow's locator that requires the user to input their location, as RecycleHelper extracts the information automatically. Furthermore, the feature was implemented in such a way that, if the RecycleNow database access was granted, this could be integrated through simple modifications to only a few lines of code. It was therefore concluded that the location-specific information requirement was met.

6.1.1.5 Off-Device Database

Due to the potentially resource-expensive nature of the machine learning model and the device location monitoring, it was determined that all other resource usage should be minimised. Therefore, as explained in Section 5.3, and mentioned above, recycling information for the search and symbols features, as well as the recycling facts displayed on the home-screen, were all stored off-device in a Firebase Realtime Database. This allows for the information to be updated or extended without the need to release a new app version, therefore drastically cutting down development times. Furthermore, it means that the iOS watch dog is less likely to terminate RecycleHelper for using too many resources, as no unncessary information is stored on device. In addition, the realtime nature of the database means that users will see the most up-to-date information available without needing to update or restart their app, providing that they are connected to the internet. Without internet access, their phone will just display the most recent information that it was able to obtain. The date that the information was last updated is also displayed in the app's settings. All these factors not only mean that the requirement was satisifed, but that the app usability is optimised.

6.1.1.6 Well-Designed UI

To validate the user interface design of RecycleHelper, testing was performed on all available iPhone models that could run iOS 13 and hence run the application. This meant that the UI was tested on all possible screen sizes, during both the Uuability testing, on user's phones, and the performance testing, using the XCode simulator.

Implementation of resizing UI constraints, rather than fixed-width or fixed-height, and scroll views where necessary meant that all views that can be reached within the app have:

- A layout that adapts to the user's device
- No text truncation or image cropping
- Clear titles
- Easily readable font sizes

Furthermore, the table cells of all table views, and the annotations of all map views, were reused from custom classes and designs, to ensure a continuity of design throughout the application, as well as minimise resources that would have otherwise been used when creating a new table cell for each item, or new map annotation for each location. This continuity of design was extended through use of the same font, colour scheme and graphics consistently throughout the design. To this effect, RecycleHelper clearly satisfies the aesthetically pleasing aspect of a well-designed UI.

However, a well-designed UI is not only aesthetically pleasing, but also easy to use. This aspect was analysed during the three rounds of usability testing, where test participants were asked to rate features, views and the overall app in terms of ease of use, information clarity and satisfaction. As summarised in Section 5.3, these were used to generate usability and user satisfaction scores. Whilst the testing itself will be further evaluated in Section 6.3, the high values of each of these scores highlights that RecycleHelper performs extremely well when considering this requirement.

6.1.1.7 Persuasive Techniques

As introduced in Section 2.4, behaviour design is a well established concept, and thus many techniques exist that can be applied to UI design and persuading users to improve their behaviour, some of which are relevant to RecycleHelper, and others that are not. Therefore, each method, technique or principle that was included in the design, was carefully considered to ensure that it was relevant, and would add value to the project.

Due to COVID-19 limitations, the effects of these could not be tested as extensively as was originally hoped; in-person focus groups could not be held, and neither could each test participant be provided with two devices with different versions of the app - one persuasive and one not. However, despite this, the impact of the persuasive techniques could still be assessed, by analysing the effect each app version had on a user's likeliness to recycle. Specifically, versions 1 and 2 did not utilise persuasive design, whereas version 3 did. Therefore, the effect of the persuasive principles could be assessed by comparing the change in likeliness to recycle brought about by the version(s) with and without the implementation of persuasive principles. Furthermore, the change in likeliness to recycle brought about by the final app version was assessed both immediately after using the app, and after having used it for varying periods of time. Special attention was paid to those that originally reported the most unlikely to recycle, specifically if they rated their likeliness at a 4 or below. This was observed as they were deemed the most likely to be affected by lack of motivation, rather than just a lack of knowledge.

These tests highlighted positive results, as all users reported an increased likeliness to recycle after having been provided with access to the app, with an average increase of 15%, suggesting that various app features were able to persuade them to recycle more. Therefore, this project requirement could be categorised as successfully met.

6.1.1.8 Usage Tracking

This feature was included in the specification as a method of allowing users to track their progress and understand their environmental impact. Included in this feature are also the 'could-have' gamification, rewards and social comparison features. Initially, the plan was to use this feature to quantify a user's recycling performance so that they could understand the positive benefits. E.g. The energy they had saved, or the CO_2 they had prevented from reaching the atmosphere. The end goal of this was users understanding the benefits of recycling and reduced waste and therefore influencing them to gradually change their shopping and usage habits to be more sustainable. Overtime, their overall impact would be tracked and quantified to provide badges and rewards, and allow users to view their position on a leaderboard with their friends that was ranked by the greatest positive impact.

However, it quickly became clear that it would be hard to accurately quantify the impact of recycling each item without asking the user such an extensive series of questions that the app usability and efficiency would be negatively impacted. This is because the item's material, size and weight and specific destination would need to be known in order to fully calculate the energy required to recycle it vs the energy saved, and similar metrics. To this effect, it was also hard to implement rewards and social comparison, without a metric to determine a user's ranking, or if they deserved a reward.

Therefore, to ensure that at least a proof-of-concept version of this feature was included in the app, the design was simplified down to allowing the user to set a target of how many items they wanted to recycle, and providing a encouraging comment once they met the target, before resetting their progress back to zero. With further time and development, this could be extended into a sophisticated data analysis feature, similar to the CO_2 impact trackers that are available on the app store, to allow users to really understand their impact. Furthermore, rewards such as in-app badges and achievements, real-life vouchers and discounts, and other incentives could be implemented to further increase the impact that RecycleHelper would have on a user's motivation to recycle. This will be discussed further in Section 7.3.

6.1.2 Summary

A summary of the comparison of the project requirements to the outcome of RecycleHelper can be seen in Table 6.1.

Requirement	**Outcome**
Waste Classification	
Item Classification	A CNN model allows users to classify items using their device camera.
Accurate classification	The model achieved an accuracy of 70%.
Back-up identification method	The app also offers a search function and a list of recycling symbols.
Recycling Information	
Relevant item information	Displays item name, recyclability and steps to follow.
Minimised device resources	Information is stored in an off-device database.
Location-specific information	Device location is used to retrieve location-specific instructions.
UI, UX and Motivation	
Well-Designed UI	RecycleHelper is aesthetically pleasing and easy to use.
Usability optimisation	A usability score of 4.529.
Persuasive techniques	Implementation of techniques such as the Mere-Exposure Effect and Hick's Law. Found to increase motivation by 15% and ability by 31%.
Usage tracking	Item recycled counter and goals.
Gamification and Rewards	To be included in future work.

Table 6.1: A Comparison of the Original Project Requirements with the Project Outcome

6.2 Comparison with Existing Solutions

6.2.1 Competitive Analysis

As highlighted in Section 2.6, various work has been done in the area of machine learning for waste classification, application development for recycling education, and application development for behaviour design. However, very little work has been done on a combination of these three areas. Despite this, a group that can be used to compare RecycleHelper to, in order to assess whether it builds on what is already available on the market, is the existing recycling information apps. Whilst a comparison of these apps has already been made, as seen in Table 2.8, this section will complete a competitive analysis, where each app will be directly compared, as well as compared to the final version of RecycleHelper. This can be seen in Table 6.2, where S represents the standard, + favourable performance, and - unfavourable.

	Recycle Academy	**Recycle!**	**Recycle Smart**	**Recycle Wizard**	**Recycle It!**	**Surrey Recycles**	**Recycle Right**	**RecycleHelper**
Availability of Information	-	-	-	-	-	S	+	+
Relevance of Data	-	-	-	-	-	S	+	+
Usefulness of Data	-	-	-	-	-	S	+	+
Ease of Use	-	-	+	+	S	+	+	+
User Interface	+	+	+	-	-	S	+	+
Other Functionality	-	-	+	S	-	+	+	+
App Rating	S	-	-	-	-	+	+	+
Persuasive Techniques	-	-	-	-	-	-	S	-

Table 6.2: Competitive Analysis of Recycling Information Apps

6.2.2 In-Depth Comparison

Table 6.2 highlights that two apps were able to consistently out-perform competition; RecycleHelper and Recycle Right. To this effect, this section discusses a more in-depth comparison of the two with respect to the criteria seen in Table 6.2 that was completed, in order to ascertain which one truly provides the best tool.

6.2.2.1 Availability of Information

This criteria refers to the range of locations for which each app provides specific recycling information. Currently, RecycleHelper has the ability to provide location-specific information for 12 locations. In contrast, Recycle Right only provides location-specific information for 6 locations. A direct comparison of these locations can be seen on the map shown in Figure 6.2, where RecycleHelper locations can be seen in pink, and Recycle Right in orange.

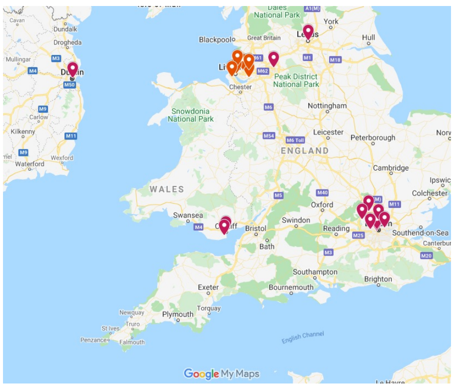


Figure 6.2: Comparison of the Availability of Location-Specific Recycling Information

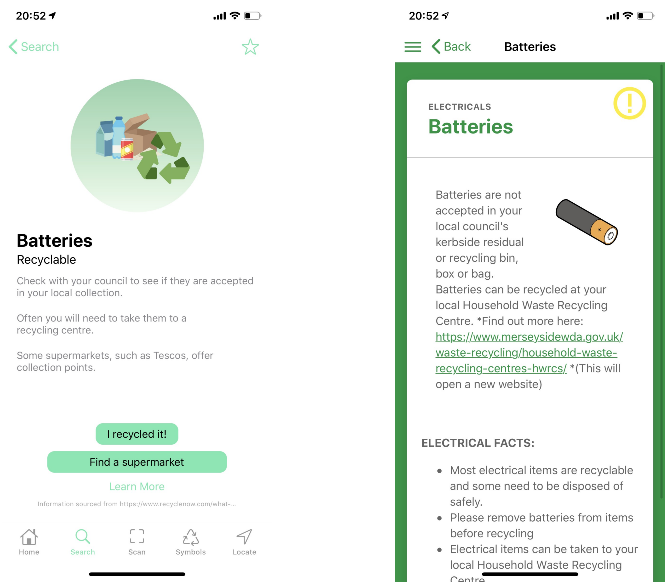
This highlights that RecycleHelper provides location-specific information for not only a larger amount of locations, but over a wider area of the UK and Ireland. Therefore, for this criteria, it can be concluded that RecycleHelper out-performs Recycle Right.

6.2.2.2 Relevance of Data

This criteria refers to whether the information that is provided in each app is relevant to consumers in the UK. As both RecycleHelper and Recycle Right are aimed at a UK audience, they provide equally relevant data, and therefore no clear winner can be chosen.

6.2.2.3 Usefulness of Data

This criteria was designed to assess the usefulness of information provided for the same item by different application. To compare RecycleHelper and Recycle Right with respect to this, batteries were chosen randomly, and the information presented compared. To this effect, screenshots from both applications can be seen in Figure 6.3.



(a) RecycleHelper (b) Recycle Right

Figure 6.3: Comparison of the Usefulness of Information

Inspection of both screenshots highlights that both apps tell the user that the batteries need to be taken to an alternative location, and provide the user with a link to do so. The difference here is that Recycle Right sends users away from the app to provide this information, whereas RecycleHelper provide this information in-app. A further comparison highlights that Recycle Right has a more extensive list of items, and also provides some facts about the item. However this information doesn't provide the user with further instructions and therefore does not add value with respect to usefulness of information. Furthermore, RecycleHelper offers a similar style of facts on the home page of the app. What's more, RecycleHelper offers information about symbols found on items, which Recycle Right does not provide. To this effect, as each app provides the minimum required information, but also provides something which the other does not, no clear winner can be determined.

6.2.2.4 Ease of Use and User Interface

This criteria is pretty self explanatory, assessing how easy it is to use the app. In terms of initially knowing what to do, Recycle Right opens to a welcome screen that says "Find out what should and shouldn't be recycled and why", immediately informing users what the app does and how to do it. Whilst RecycleHelper does not have this information on the home screen, the first time the user downloads and opens the app, they are presented with a series of 6 screens that introduces the app and explains its functionality. Furthermore, it is possible to review this information in settings. Additionally, both apps have a simple, clear and well-annotated layout and user interface that is aesthetically pleasing and allows users to work out what features do. To this effect, this criteria does also not have a clear winner.

6.2.2.5 Other Functionality

This evaluates what functionality each app offers above and beyond providing recycling information. Therefore, a conclusive list of features offered by both RecycleHelper and Recycle Right can be seen in Table 6.3.

Features	**RecycleHelper**	**Recycle Right**
Personalisation	✓	
Recycling Facts	✓	✓
Progress Tracker	✓	
Bin Collections Reminder	✓	
Search for an Item	✓	✓
Scan an Item	✓	
List of Packaging Symbols	✓	
Locate the nearest Recycling Centre	✓	✓
Locate the nearest Supermarket	✓	
Locate the nearest Charity Shop	✓	
Apply for a Van Permit		✓
Total	**10**	**4**

Table 6.3: Comparison of the Functionality Offered

Inspection of Table 6.3 highlights that RecycleHelper is a clear winner over Recycle Right in terms of overall, as well as the quantity of extra functionality, providing 10 features, 7 of which Recycle Right does not offer. This win is further solidified by the fact that the “Apply for a Van Permit” feature, the only unique feature offered by Recycle Right, currently does not work, due to the permit scheme currently being suspended.

6.2.2.6 App Rating

Whilst RecycleHelper is not currently available on the app store, as mentioned in Section 5.3, test participants were asked to rate the app as if it had been an app that they had found and downloaded from the app store. This resulted in an average rating of 4.35 out of 5, from a total of 16 participants. In contrast, Recycle Right only has a singular rating on the app store, where it was given only 1 star. Whilst this review was from 3 years ago, it was the only review and rating made visible for the app. Therefore, RecycleHelper clearly wins out in terms of popularity with users.

6.2.2.7 Persuasive Techniques

This criteria was introduced as a means of assessing whether other persuasive applications for improving recycling behaviour exist on the market. As detailed in this report, RecycleHelper has been designed to utilise specific persuasive techniques to influence users to improve their recycling behaviour. In contrast, Recycle Right has no persuasive techniques that can be identified and therefore does not perform well in this category.

6.2.3 Summary

Table 6.4 summarises the results of the detailed comparison of RecycleHelper and Recycle Right, compared as they were the best performers of in original competitive analysis detailed in Table 6.2.

Criteria	**Best Performance**
Availability of Information	*RecycleHelper*
Relevance of Data	*Both*
Usefulness of Data	*Both*
Ease of Use	*Both*
Other Functionality	*RecycleHelper*
App Rating	*RecycleHelper*
Persuasive Techniques	*RecycleHelper*
Overall	**RecycleHelper**

Table 6.4: Outcome of the Comparison between RecycleHelper and Recycle Right

Inspection of each criteria highlights that RecycleHelper was able to consistently out-perform Recycle Right, and therefore all present competition, to provide a useful and easy-to-use resource for recycling information.

6.3 Usability Testing

As documented in Section 5, this project followed an iterative development strategy, comprising of three rounds of designing, building, testing and analysis. An especially important component of each round was the usability testing, how it evolved as the project progressed, and how its results affected the next stage. This section therefore aims to evaluate this process.

6.3.1 Test Participants

Of the 68 initial insights survey participants, a total of 15 were recruited for the usability testing, and a further 8 were recruited who did not complete the initial survey. Of these 23, 3 of took part in all three rounds of testing, 10 in two rounds, and the remaining 10 in just one round. This distribution was purely by chance, corresponding to factors such as the COVID-19 virus, university or work deadlines, and personal reasons.

Both the survey and usability testing participants were recruited via word of mouth, personal requests, or recruitment posts on social media, to ensure a diverse range of backgrounds. 78.26% of the usability testing participants were students from Imperial College London, Leeds University, the University of Cambridge and the University of Oxford. The testing cohort had over 10 different nationalities, and an age range of 40 years. This demographic range highlights the wideness of the testing recruitment process, and meant that all test participants didn't just fit into one category.

6.3.2 Interview Design

As highlighted in Section 4.3.1, each round of user testing interviews was planned by first establishing the goals for that round. These could then be grouped according to feature or style of question, in order to structure the interview, before being converted into questions. For each question, it was important to ensure that they were not biased to influence the user into answering a certain way. For example, "did the onboarding experience make you feel welcomed to the app?" would be a leading question, whereas "how did the onboarding experience make you feel?" would not.

To further eliminate any potential bias that may be present in each test participant's answers, the final round of usability testing was completed anonymously using an online form. This round of testing was performed in this manner in case users felt pressured to provide a higher rating, when being interviewed by the developer directly, than they normally would. When the results of this round produced similar trends as the previous two stages, it was concluded that no bias of this type had been present throughout the process.

Before each stage of usability testing was designed, the results of the previous stage were critically analysed in terms of factors like whether they provided the desired information, and if user responses could be misinterpreted. An example of this can be seen by comparing the questions from the first and second rounds of testing. For example, the first two questions of the first round of testing, namely...

1. Could you tell me a bit more about your current recycling habits.
2. Is there a particular time in the day/day in the week that you do your recycling?

...were removed before the next round of testing. This was because their results didn't offer much insight and the required response was often interpreted differently by different participants. Furthermore, all the questions of the first round that gathered insight about participants' recycling knowledge and practice were converted from qualitative-style to quantitative-style questions. This change was performed in order to generate more comparable metrics, such as a participant's perception of their recycling knowledge, as well as their motivation to recycle. Previously, each participant had answered slightly differently, and therefore each response could be interpreted differently according to how one might understand the meaning of a certain word. In contrast, a quantifiable result using the Likert scale made for much easier comparison and reduce the chance of misinterpetation. Furthermore, the questions asked *during* testing were limited in the second round, so as to observe how a user would realistically interact with an app without any external influence. This generated an opportunity to gather insight on what features users were naturally try out first, as well as which features were easier to use, and which required more of a learning curve. Finally, the number of questions about the app usability and information clarity was increased, in order to increase the statistical significance of any results.

However, a limitation of the testing at all stages of the process was the potential interpretation of questions. This concern arose as definitions of "clear", "understandable", "difficult", or other descriptive words, that users were asked to rate features on, were not provided. This meant that questions were potentially open to misinterpretation when compared to the desired response. In hindsight, this was an oversight, especially as around 25% of participants' first language was not English. Therefore, any further rounds of testing that may be implemented in future should include this.

6.3.3 Participant Feedback

In the final round of testing, participants were given the opportunity to provide their thoughts and feedback on the app. These were overwhelmingly positive, including comments such as "it's extremely useful and intuitive", "it meets my needs plus more", "I think that there's nothing like it out there" and "I loved how easy it was to use and how effortless everything felt on my side, it did exactly what I wanted or needed it to". These responses were really encouraging to read, and further validated the work that had been completed to date.

However, alongside these comments was a repeated trend referring to the low accuracy of the waste classification feature. This was to be expected, given the discussion in Section 6.1.1.1, but definitely highlights the need for this to be a focus in any future work.

6.4 Summary

This chapter has provided a detailed analysis and discussion of RecycleHelper when compared to the initial project requirements, and highlighted how it builds on recycling information apps that are currently available on the market. In fact, following competitive analysis against 7 other recycling information apps currently available on the App Store, and subsequent in-depth comparison with its closest competition, RecycleHelper was found to consistently perform best. This conclusion was based on comparison of availability and quality of recycling information, usability, other functionality offered and persuasive techniques deployed.

Key areas to develop going forward have been highlighted, such as the accuracy of the waste classification, scalability of the location-specific database and improvement of the usage tracking feature. These areas, as well as the plethera of potential ideas and features that did not "make the cut" to be included in this work, provide plenty of opportunity to further improve the app going forward. Despite this, RecycleHelper can still be classified as a key tool to improving recycling performance of consumers within the UK.

Chapter 7

Conclusion and Future Work

Ever since the Industrial Revolution, activities such as burning fossil fuels, and vast amounts of waste ending up landfill, have caused an increasing volume of greenhouse gases to be emitted into the atmosphere. In turn, these have contributed to the heating up of the Earth above what would be considered normal levels, and caused this change in climate to develop into one of the most urgent challenges that the World faces today.

In an effort to combat this effect, emphasis has been placed on recycling, composting and reducing waste, to help to minimise the volume of greenhouse gases reaching the atmosphere. This is especially important in the UK, as we are currently one of the largest waste exporters in the World. Despite this, research has found that, in the UK, there is a lack of both education on and motivation towards recycling correctly. RecycleHelper was therefore designed to fill in a gap in the market; providing a recycling information app that is not only easy, but enjoyable to use.

This report has detailed the design and development process of bringing RecycleHelper to life. Initially, background research was carried out to further understand the areas concerned with the project and to identify the needs of consumers with respect to improving their recycling performance. Previous work in the areas of waste classification, recycling information apps and persuasive apps were also analysed and compared in this section. This research was then used to define the scope and goals of the project, which were then further developed into project requirements, and later, features to implement.

RecycleHelper was developed in XCode, using Swift and a MVC design architecture, in three stages of development, to produce three successive app versions, each an improved version of it's predecessor. Furthermore, frameworks such as Vision, CoreLocation, FirebaseDatabase and UserNotifications were used to provide the app with extra functionality. At each stage, the app version went through thorough performance and usability testing, to highlight and potential issues or bugs, after which further requirements and goals were established for the following stage of development. The app includes a convolutional neural network model, built in python, used to perform predictions based on an object's material, and subsequently provide instructions on how to recycle it. Alongside this feature was two back up methods of identification - a search feature and a list of symbols commonly found on packaging. However, it is important to note that the focus of this project was not to produce the best machine learning model possible. Instead, focus was placed on principaled design, psychological principles and thorough testing and development.

The aim of this chapter is to discuss the success of the project and its respective achievements, highlighting key findings, challenges and limitations discovered throughout the process. Future work is also discussed to further extend the app's functionality and impact.

7.1 Achievements

This report has highlighted how RecycleHelper has achieved the main goals of improving user's recycling knowledge and motiviation, whilst still maintaining a usable design. Evaluation of the testing results of Section 5.3 found that on average, use of RecycleHelper for less than a week improved user's motivation by an average of 15%, and their recycling knowledge by an average of 31%. These figures suggest that, given such a drastic improvement over a short period of time, prolonged usage of the app could see an increase in motivation and knowledge to near-perfect levels. Furthermore, test participants rated the app an average of 4.529 out of 5 for usability.

At this point, emphasis should therefore be placed on the positive impact that this app could potentially have on user's lifestyles, as an educational resource promoting sustainability. Given the current state of the Earth's climate, everyone needs to begin working towards living more sustainable lifestyles, and, whilst the end goal should be to reduce, if not eliminate, waste overall, RecycleHelper could be a key tool in the process of increasing worldwide recycling rates and therefore transitioning to a more circular economy.

The positive impact of RecycleHelper was definitely influenced by its clean UI and usable design. Navigation through the app is simple, as all features can be accessed in two taps or less, and switching between features can be achieved through only a single tap of the relevant section of the tab bar. Furthermore, a consistent colour scheme of soft and pale greens is used throughout the app, for green's common association with not only recycling symbols but also positivity. Additionally, the muted colours were chosen in attempt to produce a calming effect on the user. Apple's Human Interface Design Guidelines were followed to provide users with an app that includes many features, without feeling cluttered. Finally, a short onboarding process minimises the learning curve required to use the app. This design was implemented in an attempt to not deter users from using it, in a way that an app with a less thought out design would, and instead making the user experience more enjoyable.

A further interesting feature is the persuasive nature of the app. As introduced in Section 2.4 and detailed in Section 5.3, many persuasive techniques and principles were used to improve the UX and persuade users to perform a series of target behaviours. Examples of such techniques that were implemented include the Mere-Exposure Effect, Hick's Law and the Placebo Effect. The implementation of these can be attributed to the improvement in every user's recycling knowledge and motivation that was observed in the final round of usability testing. This utilisation of persuasive techniques is what separates RecycleHelper from its competition, whether this be recycling apps like Recycle Right, or online recycling information resources like RecycleNow. This is because these apps and resources only provide recycling information and, as highlighted in Section 2.2, consumers recycling rate is influenced by not only their knowledge, but also their *motivation* to recycle. Therefore, employing persuasive techniques and principles alongside the provision of recycling information in the way that RecycleHelper does can potentially provide a larger improvement in recycling rates.

Finally, competitive analysis of RecycleHelper when compared to other recycling information apps on the market highlighted it as a leader in the field and insinuated that it filled a gap in the market. Its closest competition in this analysis, Recycle Right, was assessed to offer comparable performance in terms of relevance and usefulness of data, as well as ease of use. However, RecycleHelper was proven to be superior due to its wider range of locations that information was supplied for, as well as extra features supplied, and persuasive techniques implemented.

7.2 Limitations and Challenges

Whilst this project focus was not machine learning, a key limitation of this project is the accuracy of the waste classification model, as discussed in Section 6.1.1.1. Although the training accuracy was high, translating this to in-app use resulted in a decrease in performance, most likely due to the style of images contained in the dataset. Namely, they were often very close up, and always with a plain background. However, in real life, users will most likely wish to be able to scan an item in any situation and without having to hold the phone camera extremely close to the object. Furthermore, the dataset exhibits extreme class imbalances, which, if left untreated, result in overfitting on the larger classes, and underfitting on the smaller. If RecycleHelper is released on the app store and subsequently gains in popularity, it would be deemed necessary for the machine learning model to be optimised in order to maximise the user retention rate. This is because the computer vision aspect will be one of the more attracting features of the app, due to its uniqueness, and therefore a low level of performance could deter many from using the app again. However, for the self-contained nature of the project, the model served its purpose and its performance did not have any sizeable negative effect on the test participants' perception of the app.

Another limitation of RecycleHelper is the current lack of scalability that the location-specific recycling information database exhibits. This was an unavoidable challenge, due to the wide variety of recycling instructions throughout the nation, and the lack of a resource that compiles into all into one place but is still easily accessible to scrape data from. Currently, if a location needs to be added to the database, the developer must manually find the correct recycling information on the council's website, and input the recyclability status and instructions for all items listed to be included in the app. Whilst this is achievable when completing testing with < 20 users, integrating location-specific information for all of the UK, which is a long-term goal, would not be feasible following this technique.

Finally, as with most projects submitted this year, the current COVID-19 pandemic posed a challenge to testing plans, as focus groups and randomised testing with members of the public had been planned. The lockdown implemented back in March required careful re-planning of the final two rounds of testing, resulting in a round of 20-30 minute video interviews, and then a round of questionnaires in the structure of three online forms. This required extra steps for the user, such as setting up sharing their phone screen so that their behaviour when trying out the app during the second round of testing could be observed. However, that being said, other than in the testing respect, this project has been rather fortunate in terms of a minimised impact due to the coronavirus pandemic.

7.3 Future Work

Whilst the work completed within the project timeframe is definitely noteable, this project is open-ended. Should the project be taken further, or picked up by another student, there will always be plenty of features that can be implemented to further extend RecycleHelper's functionality. These areas are described in this section.

7.3.1 Machine Learning Model Improvements

As highlighted in the section above, there is still much to be desired in terms of the accuracy of waste classification machine learning model. Therefore, a key area of the future work should be focussed on improving this design, to maximise its performance once deployed in the app. Alongside improvement of the model should also be improvement of the dataset, in terms of variety of images, class balance and size of the dataset. Due to the number of papers that have utilised this dataset, developing it further would allow the state of the art to improve above the current level, and benefit a wider audience than just those involved in this project.

7.3.2 Extended Recycling Information

A key step to extend the recycling information available in-app is to improve the scalability of the dataset in terms of adding both locations and new items. If access to the RecycleNow dataset is never granted, a similar service could be established where, as set out in Section 3.3.4.4, a back end application for local authorities is built, to allow councils of other locations not currently included in the app to upload their recycling information and location. Upon admin approval, the app will then factor this information into its database to further extend the user base and area reach. A portal could also exist in this back-end application to enable the authorities of each location to log in to track usage stats and/or update information. This would solve the problem of the data update process being time-consuming for the developer.

7.3.3 Environmental Impact

The current recycling tracker could be further developed into a feature that calculates the impact that a user has on the environment from improving their recycling behaviour. For example, how many trees they saved, or the amount by which they reduced their energy and/or CO_2 impact. To implement this, in app calculations could be performed, or a partnership with an already existing environmental impact tracking service could be established.

This display of a users impact could in turn be used to influence their shopping habits, to those of a more sustainable lifestyle. Initially, users could be influenced to select products made of the materials that have longer lifetimes and are easier to recycle, such as glass and metal. Eventually, they could be educated on the benefits of a zero-waste lifestyle, where shoppers take reusable packaging with them to the store (such as produce bags, glass jars and tupperware) to minimise the amount of disposable packaging they receive. Potentially, this change in behaviour could also be tracked in the application, to make their calculated environmental impact more accurate.

7.3.4 Gamification and Rewards

To improve the user experience and provide more motivation to recycle, further features should be added to the recycling tracker. Examples of this could include badges/achievements awarded at milestones, such as 10 items recycled, 50 items, 100 etc, similar to how, for example, Duolingo awards badges to users the more they use the app, tracked by variables such as the amount of XP earned, or the amount of new words learned by a user. This method of incentivisation could be easily applied to RecycleHelper and extended further, by establishing deals with well-known supermarket brands, so that the best performing users could earn or trade in their points for discounts and vouchers for the brand of their choice.

Finally, a log-in feature could be added to the app, so that each user has a social profile. They could then use this profile to connect with friends, view their ranking on a leaderboard, and participate in competitions. However, this feature should not be required by users, i.e. users should be able to use the other app features without logging in, so as to not provide a barrier against recycling education for those that do not wish to create a social account.

7.3.5 App Store Publishing

Finally, an important goal to be achieved in this project is to get RecycleHelper published on the App Store. This is essential, so that all of the hard work completed to date can be used to benefit the general public. Although the steps to achieve this are relatively simple, they are time consuming, due to the complex approval process that Apple requires apps to go through, in order to maintain the standard of applications that are available on the App Store. Therefore, to make the most of the opportunity, and minimise the risk of the app being rejected, the standard of RecycleHelper should be fine-tuned, ironing out any last minute issues or bugs that were highlighted in the final round of testing. Furthermore, the branding of the app, such as the name and app icon, should be reviewed at this stage in order to maximise its impact. Once this is complete, and RecycleHelper has been approved such that it is published on the app store, regular updates should be provided to continuously improve functionality and attract new users.

7.4 Summary

The main objective of this project was to produce a tool, in the form of a smartphone app, that could be used to improve user's recycling performance, primarily through the use of persuasive techniques. To this extent, a vast proportion of the project was spent focusing on design and usability of the app. This produced an aesthetically pleasing, well packaged app, that fills a gap in the current market. Despite the limitations of the waste classification model and the challenges posed by COVID-19, and whilst there is still work that can be done to further improve and extend the app's functionality, the quantifiable achievements of the implementation of RecycleHelper can be used to highlight the success of this project.

References

[1] "Global Greenhouse Gas Emissions Data." [Online]. Available: https://www.epa.gov/ghgemissions/global-greenhouse-gas-emissions-data

[2] "Climate Change." [Online]. Available: https://www.un.org/en/sections/issues-depth/climate-change/

[3] NASA, "The Effects of Climate Change." [Online]. Available: https://climate.nasa.gov/effects/

[4] "UK Statistics on Waste," Tech. Rep. [Online]. Available: http://www.statisticsauthority.gov.uk/assessment/code-of-practice/index.html.

[5] K. Buchholz, "The Biggest Exporters of Plastic Waste in the World," 2019. [Online]. Available: https://www.statista.com/chart/18229/biggest-exporters-of-plastic-waste-and-scrap

[6] U. Plastics Pact, "Defining what's recyclable and best in class polymer choices for packaging," Tech. Rep.

[7] Department for Environment Food and Rural Affairs, "Digest of Waste and Resource Statistics, 2018 Edition," Tech. Rep., 2018. [Online]. Available: www.gov.uk/government/collections/waste-and-recycling-statistics

[8] WRAP, "Recycling Tracking Survey 2019," Tech. Rep., 2019. [Online]. Available: www.wrap.org.uk

[9] "Collections and Reprocessing — WRAP UK." [Online]. Available: https://www.wrap.org.uk/collections-and-reprocessing

[10] "Emissions in the United Kingdom (UK)," Statista, Tech. Rep., 2019.

[11] "MRFs Comparison of efficiency and quality Materials Recovery Facilities," Tech. Rep., 2006. [Online]. Available: www.wrap.org.uk

[12] "WasteRecycling.org.uk — The Evolution of Recycling." [Online]. Available: http://www.wasterecycling.org.uk/the_environment/evolution_of_recycling.php

[13] "What happens to my recycling?" [Online]. Available: http://www.wiseuptowaste.org.uk/recycle/what-happens-to-my-recycling/

[14] "What is the Scheme? - OPRL." [Online]. Available: https://www.oprl.org.uk/get-involved/what-is-the-scheme/

[15] WRAP, "OPRL Brochure," Tech. Rep. [Online]. Available: www.oprl.org.uk

[16] Plastics Europe Deutschland e.V. and Messe Düsseldorf, "Plastics - the Facts 2019: An analysis of European plastics production, demand and waste data," Tech. Rep., 2019.

[17] "Recycling your plastics - Chelmsford City Council." [Online]. Available: https://www.chelmsford.gov.uk/bins-and-recycling/what-goes-in-each-bin/clear-sack-for-plastics/recycling-your-plastics/

[18] Which?, "How To Recycle In The UK." [Online]. Available: https://www.which.co.uk/reviews/recycling/article/how-to-recycle-in-the-uk

[19] London Recycles, "Recycling symbols explained." [Online]. Available: https://londonrecycles.co.uk/recyclingsymbols

[20] "Dizzying array of recycling symbols 'is confusing consumers' — Environment — The Guardian." [Online]. Available: https://www.theguardian.com/environment/2018/jul/30/dizzying-array-of-recycling-symbols-is-confusing-consumers

[21] "Going for increased recycling. A social cost-benefit analysis - CE Delft." [Online]. Available: https://www.cedelft.eu/publicatie/going_for_increased_recycling_a_social_cost-benefit_analysis/1422

[22] W. P. Chang, C. C. Chan, and J. D. Wang, "60Co contamination in recycled steel resulting in elevated civilian radiation doses: Causes and challenges," *Health Physics*, vol. 73, no. 3, pp. 465–472, 1997.

[23] National Recycling Coalition and University of Colorado Recycling Services, "Responses to 10 Common Anti-Recycling Arguments," Tech. Rep. [Online]. Available: http://www.nrc-recycle.org/

[24] "Recycling Advantages & Disadvantages: The Ups & Downs of Recycling." [Online]. Available: https://www.renewableresourcescoalition.org/recycling-advantages-disadvantages/

[25] "10 Ways Recycling Hurts the Environment - Listverse." [Online]. Available: https://listverse.com/2013/01/27/10-ways-recycling-hurts-the-environment/

[26] A. Micks, "The Costs of Recycling," 2012. [Online]. Available: http://large.stanford.edu/courses/2012/ph240/micks2/

[27] "Plastic, recovered plastics market — WRAP UK." [Online]. Available: https://www.wrap.org.uk/node/10792

[28] "TALA." [Online]. Available: https://www.wearetala.com/

[29] "Netherlands: share of customers paying attention to sustainability," 2019. [Online]. Available: https://www.statista.com/statistics/656337/share-of-customers-paying-attention-to-sustainability-of-products-in-the-netherlands/

[30] "Department for Environment, Food and Rural Affairs Government Buying Standards for showers, taps, toilets and urinals," Tech. Rep., 2014. [Online]. Available: http://www.europeanwaterlabel.eu/

[31] Thermostatic Mixing Value Manufacturers Assosciation, "Recommended Code of Practice for Safe Water Temperatures," Tech. Rep., 2000. [Online]. Available: www.tmva.org.uk

[32] "First Law of Thermodynamics Summary." [Online]. Available: https://webhome.phy.duke.edu/~rgb/Class/phy51/phy51/node59.html

[33] "Joules to kWh conversion calculator." [Online]. Available: https://www.rapidtables.com/convert/energy/Joule_to_kWh.html

[34] "recycle-more,Top Facts." [Online]. Available: https://www.recycle-more.co.uk/household-zone/top-facts

[35] Viridor and Edelman Intelligence, "Recycling Index," Tech. Rep., 2019.

[36] Recycle Now, "Local Recycling - Recycling Locator." [Online]. Available: https://www.recyclenow.com/local-recycling

[37] Gov.UK, "Recycling collections." [Online]. Available: https://www.gov.uk/recycling-collections

[38] WRAP UK, "About Us." [Online]. Available: https://wrap.org.uk/about-us/about

[39] L. Smith and P. Bolton, "Household recycling in the UK," Tech. Rep., 2018. [Online]. Available: www.parliament.uk/commons-library|intranet.parliament.uk/commons-library|papers@parliament.uk|@commonslibrary

[40] Recycle Now, "About us." [Online]. Available: https://www.recyclenow.com/about-us

[41] "Recycling in the United Kingdom - Wikipedia." [Online]. Available: https://en.wikipedia.org/wiki/Recycling_in_the_United_Kingdom

[42] Wrap, "A framework for greater consistency in household recycling in England," Tech. Rep.

[43] A. Holmes, J. Fulford, and C. Pitts-Tucker, "Investigating the Impact of Recycling Incentive Schemes Full Report," Tech. Rep., 2014. [Online]. Available: www.serco.com

[44] Prepared by SLR Consulting for SUEZ, "At this rate... exploring England's recycling challenges," Tech. Rep., 2015.

[45] Science & Environment, “Where does recycling and rubbish from the UK go?” 9 2019. [Online]. Available: https://www.bbc.co.uk/news/science-environment-49827945

[46] European Environment Agency, “Waste Recycling in Europe,” 2019. [Online]. Available: https://www.eea.europa.eu/data-and-maps/indicators/waste-recycling-1/assessment-1

[47] Resource London and LEDNET, “Guide to Improving Waste Management in the Domestic Rented Sector-Appendix,” Tech. Rep., 2017.

[48] I. M. L. Robertson, “The Measurement of Population Turnover: A New Approach by Coordinate References in a Scottish City,” *Town Planning Review*, vol. 53, no. 1, p. 79, 1982. [Online]. Available: https://doi.org/10.3828/tpr.53.1.f3054gp45864706l

[49] G. Morton and L. Cross, “Communications guidance - Improving Recycling through Effective Communications,” WRA, Tech. Rep., 2013. [Online]. Available: www.wrap.org.uk

[50] F. Islam, “No-deal Brexit 'will see more waste going to landfill',” 8 2019. [Online]. Available: https://www.bbc.co.uk/news/business-49440290

[51] L. Haskell, “What Impact Might Brexit have on the Waste Industry?” 2019. [Online]. Available: https://www.tj-waste.co.uk/blog/what-impact-might-brexit-have-on-the-waste-industry/

[52] “CIWM Report Policy implications of Brexit for the UK resource & waste management sector,” Tech. Rep., 2017. [Online]. Available: www.ciwmawards.com

[53] “Why the world's recycling system stopped working — Financial Times.” [Online]. Available: https://www.ft.com/content/360e2524-d71a-11e8-a854-33d6f82e62f8

[54] A. Brooks, S. Wang, and J. Jambeck, “The Chinese import ban and its impact on global plastic waste trade,” *Science Advances*, vol. 4, 1 2018.

[55] Huatai Insurance Agency & Consultant Service Ltd, “China Solid Waste Import Regulations,” 2018. [Online]. Available: https://www.londonpandi.com/documents/solid-waste-regs-china/

[56] A. Evans, “Coronavirus: Starbucks bans reusable cups to help tackle spread,” 3 2020. [Online]. Available: https://www.bbc.co.uk/news/uk-51767092

[57] M. A. Zambrano-Monserrate, M. A. Ruano, and L. Sanchez-Alcalde, “Indirect effects of COVID-19 on the environment,” *Science of The Total Environment*, vol. 728, p. 138813, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0048969720323305

[58] “Rejected recyclable waste up 84% in England since 2011, data shows,” 8 2016. [Online]. Available: https://www.bbc.co.uk/news/uk-37159581

[59] S. Laville, “Recycled plastic could supply three-quarters of UK demand, report finds,” 2018. [Online]. Available: https://www.theguardian.com/environment/2018/jun/14/recycled-plastic-could-supply-three-quarters-of-uk-demand-report-finds

[60] I. Tiseo, “Coal use for the manufacture of plastic products in the United Kingdom (UK) from 2005 to 2016,” 2018. [Online]. Available: https://www.statista.com/statistics/501039/coal-use-for-plastic-products-united-kingdom/

[61] J. Ambrose, “War on plastic waste faces setback as cost of recycled material soars,” 10 2019. [Online]. Available: https://www.theguardian.com/environment/2019/oct/13/war-on-plastic-waste-faces-setback-as-cost-of-recycled-material-soars

[62] “UK becomes first major economy to pass net zero emissions law,” 2019. [Online]. Available: https://www.gov.uk/government/news/uk-becomes-first-major-economy-to-pass-net-zero-emissions-law

[63] “Gove takes action to ban plastic straws, stirrers, and cotton buds,” 2019. [Online]. Available: https://www.gov.uk/government/news/gove-takes-action-to-ban-plastic-straws-stirrers-and-cotton-buds

[64] R. Smithers, “UK to consult on plastic packaging tax, chancellor says,” 10 2018. [Online]. Available: https://www.theguardian.com/environment/2018/oct/29/uk-to-consult-on-plastic-packaging-tax-chancellor-says

[65] J. Clement, “Global Digital Population as of April 2020,” 2020. [Online]. Available: https://www.statista.com/statistics/617136/digital-population-worldwide/

[66] “Internet Usage Statistics.” [Online]. Available: https://internetworldstats.com/stats.htm

[67] StatCounter Global Stats, “Mobile Operating System Market Share - United Kingdom.” [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/united-kingdom

[68] ——, “Mobile Operating System Market Share - Europe.” [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/europe

[69] “Android vs iOS: Which Platform to Build Your App for First?” [Online]. Available: https://medium.com/@the_manifest/android-vs-ios-which-platform-to-build-your-app-for-first-22ea8996abe1

[70] Wikipedia, “iPhone.” [Online]. Available: https://en.wikipedia.org/wiki/IPhone

[71] Quartz, “There are now more than 24,000 different Android devices.” [Online]. Available: https://qz.com/472767/there-are-now-more-than-24000-different-android-devices/

[72] “iOS vs Android Development: Pros and Cons of Each Platform.” [Online]. Available: https://rubygarage.org/blog/ios-vs-android-development

[73] Business Insider, “Android fragmentation is worsening.” [Online]. Available: https://www.businessinsider.com/android-fragmentation-is-worsening-2017-11?r=US&IR=T

[74] D. Luu, “How out of date are Android devices?” [Online]. Available: https://danluu.com/android-updates/

[75] “App Store - Support - Apple Developer.” [Online]. Available: https://developer.apple.com/support/app-store/

[76] J. Clement, “Google Play: number of available apps 2009-2020,” 2020. [Online]. Available: https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/

[77] ——, “Number of apps available in leading app stores as of 1st quarter 2020,” 2020. [Online]. Available: https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/

[78] Cambridge Dictionary, “Captology.” [Online]. Available: https://dictionary.cambridge.org/dictionary/english/captology

[79] ContextLogic Inc, “Wish - Shopping Made Fun.” [Online]. Available: https://apps.apple.com/gb/app/wish-shopping-made-fun/id530621395https://www.wish.com/m/mobile-ios-app

[80] J. T. Cacioppo, R. E. Petty, C. F. Kao, and R. Rodriguez, “Central and peripheral routes to persuasion: An individual difference perspective.” *Journal of Personality and Social Psychology*, vol. 51, no. 5, pp. 1032–1043, 1986.

[81] R. B. Cialdini, “The Psychology of Persuasion,” Tech. Rep., 2006.

[82] D. B. J. Fogg, “Fogg Behaviour Model.” [Online]. Available: https://www.behaviormodel.org

[83] Grow The Engineering, “Fogg’s Behaviour Model,” 2017. [Online]. Available: https://www.growthengineering.co.uk/bj-foggs-behavior-model/

[84] D. Lockton, D. Harrison, and N. A. Stanton, “The Design with Intent Method: A design tool for influencing user behaviour,” *Applied Ergonomics*, vol. 41, no. 3, pp. 382–392, 2010.

[85] Wikipedia, “Serial-position effect.” [Online]. Available: https://en.wikipedia.org/wiki/Serial-position_effect

[86] “Serial Positioning Effect design pattern.” [Online]. Available: http://ui-patterns.com/patterns/Serial-positioning-effect

[87] Wikipedia, “Mirroring (psychology).” [Online]. Available: https://en.wikipedia.org/wiki/Mirroring_(psychology)

[88] Interaction Design Foundation, “What is Emotional Design?” [Online]. Available: https://www.interaction-design.org/literature/topics/emotional-design

[89] “Duolingo - Language Lessons on the App Store.” [Online]. Available: https://apps.apple.com/gb/app/duolingo/id570060128

[90] Wikipedia, "Classical conditioning." [Online]. Available: https://en.wikipedia.org/wiki/Classical_conditioning

[91] P. Chen and S. M. Garcia, "Yin and Yang Theory of Competition: Social Comparison and Evaluation Apprehension Reciprocally Drive Competitive Motivation," University of Michigan, Tech. Rep., 2010. [Online]. Available: https://deepblue.lib.umich.edu/bitstream/handle/2027.42/77608/patchen.pdf?sequence=1&isAllowed=y

[92] "Social Motivation Definition." [Online]. Available: https://www.alleydog.com/glossary/cite-my-term.php?term=Social+Motivation

[93] B. Burke, "Gartner Redefines Gamification," 2014. [Online]. Available: https://blogs.gartner.com/brian_burke/2014/04/04/gartner-redefines-gamification/

[94] Statista Research Department, "Global education gamification market value," 2017. [Online]. Available: https://www.statista.com/statistics/680255/education-gamification-market-value-worldwide/

[95] H.-H. Selassie, "Persuasive Technology and Gamification at the Workplace: Engaging Employees in Effective Documentation of Analysis and Evaluations," University of Saskatchewan, Saskatoon, Tech. Rep., 2017. [Online]. Available: https://harvest.usask.ca/bitstream/handle/10388/8128/SELASSIE-THESIS-2017.pdf?sequence=1&isAllowed=y

[96] Namraata B, "The History of Push Notifications," 2019. [Online]. Available: https://vwo.com/blog/history-of-push-notifications/

[97] N. Dvorak, "Push Notifications Change The Way Consumers Engage With Apps," 2015. [Online]. Available: https://go.forrester.com/blogs/15-01-13-push_notifications_change_the_way_consumers_engage_with_apps/

[98] Wikipedia, "Mere-exposure effect." [Online]. Available: https://en.wikipedia.org/wiki/Mere-exposure_effect

[99] "The power of the placebo effect," 2017. [Online]. Available: https://www.health.harvard.edu/mental-health/the-power-of-the-placebo-effect

[100] Interaction Design Foundation, "What is Hick's Law?" [Online]. Available: https://www.interaction-design.org/literature/topics/hick-s-law

[101] ——, "Hick's Law: Making the choice easier for users." [Online]. Available: https://www.interaction-design.org/literature/article/hick-s-law-making-the-choice-easier-for-users

[102] D. Berdichevsky and E. Neuenschwander, "Toward an Ethics of Persuasive Technology," Communications of the ACM, Tech. Rep., 1999. [Online]. Available: https://cacm.acm.org/magazines/1999/5/7904-toward-an-ethics-of-persuasive-technology/abstract

[103] S. Berkovsky and J. Freyne, Eds., *Persuasive Technology*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, vol. 7822. [Online]. Available: http://link.springer.com/10.1007/978-3-642-37157-8

[104] G. Goos, J. Hartmanis, J. Van, L. E. Board, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, A. Kobsa, F. Mattern, E. Zurich, J. C. Mitchell, M. Naor, O. Nierstrasz, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, and G. Weikum, "LNCS 7284 - Persuasive Technology: Design for Health and Safety," in *7th International Conference, PERSUASIVE*, M. Bang and E. Ragnemalm, Eds. Linköping, Sweden: Springer, 2012. [Online]. Available: https://link.springer.com/content/pdf/10.1007%2F978-3-642-31037-9.pdf

[105] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database,." [Online]. Available: http://yann.lecun.com/exdb/mnist/

[106] L. Fei-Fei, R. Fergus, and P. Perona, "Learning Generative Visual Models from Few Training Examples: an Incremental Bayesian Approach Tested on 101 Object Categories," in *Workshop on Generative-Model Based Vision*, 2004. [Online]. Available: http://www.vision.caltech.edu/Image_Datasets/Caltech101/

[107] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning Word Vectors for Sentiment Analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[108] P. Popien, "How to improve your image classifier with Google's AutoAugment," 2018. [Online]. Available: https://towardsdatascience.com/how-to-improve-your-image-classifier-with-googles-autoaugment-77643f0be0c9

[109] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *The 10th Asian Conference on Computer Vision*, Queenstown, New Zealand, 2010. [Online]. Available: http://waltz.felk.cvut.cz/~matas/papers/neumann-text-accv10.pdf

[110] S. Tehsin, A. Masood, and S. Kausar, "Survey of Region-Based Text Extraction Techniques for Efficient Indexing of Image/Video Retrieval," *International Journal of Image, Graphics and Signal Processing*, vol. 6, no. 12, pp. 53–64, 11 2014. [Online]. Available: http://www.mecs-press.org/ijigsp/ijigsp-v6-n12/v6n12-8.html

[111] K. Jung, K. I. Kim, and A. K. Kain, "Text information extraction in images and video: a survey," *The Journal of the Pattern Recognition Society*, vol. 37, pp. 977–997, 2004.

[112] R. Lienhart, "Video OCR: a survey and practitioner's guide," 4 2003, pp. 155–183.

[113] M. Zhao, S. Li, and J. Kwok, "Text detection in images using sparse representation with discriminative dictionaries," *Image and Vision Computing*, vol. 28, pp. 1590–1599, 2010. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.443.1003&rep=rep1&type=pdf

[114] R. Lienhart and A. Wernicke, "Localizing and Segmenting Text in Images and Videos," *IEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 4, pp. 256–268, 2002. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=999203

[115] D. Chen, J. M. Odobez, and H. Bourlard, "Text detection and recognition in images and video frames," *The Journal of the Pattern Recognition Society*, vol. 37, pp. 595–608, 2004. [Online]. Available: https://www.idiap.ch/~odobez/publications/ChenOdobezBourlard-PatternRecognition2004.pdf

[116] G. Mehul, P. Ankita, D. Namrata, G. Rahul, and S. Sheth, "Text-Based Image Segmentation Methodology," in *2nd International Conference on Innovations in Automation and Mechatronics Engineering*. Elsevier Ltd, 2014, pp. 465–472.

[117] P. Hu, W. Wang, and K. Lu, "A Novel Binarization Approach for Text in Images," in *IEEE International Conference on Image Processing*, 2015. [Online]. Available: https://www.researchgate.net/publication/308873110_A_novel_binarization_approach_for_text_in_images

[118] L. B. Fu, W. Q. Wang, and Y. W. Zhan, "A robust text segmentation approach in complex background based on multiple constraints," in *Lecture notes in Computer Science*. Berlin Heidelberg: Springer-Verlag, 2005, pp. 594–605.

[119] P. K. Charles, V. Harish, M. Swathi, and D. CH, "A Review on the Various Techniques used for Optical Character Recognition," *International Journal of Engineering Research and Applications*, vol. 2, no. 1, pp. 659–662, 2012. [Online]. Available: https://pdfs.semanticscholar.org/53c0/1391a54aeef12576e46b4e92b7850e07b653.pdf

[120] Google Open Source, "Tesseract OCR - An optical character recognition engine." [Online]. Available: https://opensource.google/projects/tesseract

[121] Google Cloud, "Cloud Vision API - Detect Text in Images." [Online]. Available: https://cloud.google.com/vision/docs/ocr

[122] Wikipedia, "Optical Character Recognition." [Online]. Available: https://en.wikipedia.org/wiki/Optical_character_recognition#cite_note-Scantz82-3

[123] S. V. Dhavale, *Advanced Image-Based Spam Detection and Filtering Techniques*, Manish Gupta, Ed. IGI Global, 2017. [Online]. Available: https://books.google.co.uk/books?id=InFxDgAAQBAJ&pg=PA91&lpg=PA91&dq=1914+Emanuel+Goldberg&redir_esc=y&hl=en#v=onepage&q=1914%20Emanuel%20Goldberg&f=false

[124] D. Berchmans and S. S. Kumar, "Optical Character Recognition: An Overview and an Insight," in *International Conference on Control, Instrumentation, Communication and Computational Technologies*, 2014, pp. 1361–1365. [Online]. Available: https://ieeexplore.ieee.org/document/6993174

[125] G. Shperber, "A gentle introduction to OCR," 2018. [Online]. Available: https://towardsdatascience.com/a-gentle-introduction-to-ocr-ee1469a201aa

[126] R. Salunk, D. Badhe, V. Done, Y. Raykar, and B. Thakare, "The State of the Art in Text Recognition Techniques," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 2, 2015. [Online]. Available: https://ijarcce.com/wp-content/uploads/2015/03/IJARCCE7J.pdf

[127] A. Navlani, "Support Vector Machines with Scikit-learn," 2019. [Online]. Available: https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python

[128] J. Brownlee, "Classification and Regression Trees for Machine Learning," 2019. [Online]. Available: https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/

[129] S. Tahsildar, "Random Forest Algorithm In Trading Using Python," 2019. [Online]. Available: https://blog.quantinsti.com/random-forest-algorithm-in-python/

[130] K. Sorokina, "Image Classification with Convolutional Neural Networks," 2017. [Online]. Available: https://medium.com/@ksusorokina/image-classification-with-convolutional-neural-networks-496815db12a8

[131] S. Bhatt, "Understanding the Basics of CNN with Image Classification," 2019. [Online]. Available: https://becominghuman.ai/understanding-the-basics-of-cnn-with-image-classification-7f3a9ddea8f9

[132] "Introducing Convolutional Neural Networks." [Online]. Available: https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks

[133] V. Foss, "Multiclass Classification Model Evaluation," 2018. [Online]. Available: https://parasite.id/blog/2018-12-13-model-evaluation/

[134] C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz, "Exploring Features in a Bayesian Framework for Material Recognition," Tech. Rep. [Online]. Available: http://people.csail.mit.edu/celiu/CVPR2010/maltRecogCVPR10.pdf

[135] L. Sharan, C. Liu, R. Rosenholtz, and E. H. Adelson, "Accuracy and speed of material categorisation in real-world images - Flickr Material Database (FMD)," *Journal of Vision*, vol. 14, no. 9, 2014. [Online]. Available: https://people.csail.mit.edu/lavanya/fmd.html

[136] M. Varma and Z. Zisserman, "A statistical approach to material classification using image patch exmplars," 2009.

[137] G. Thung and M. Yang, "Classification of Trash for Recyclability Status," Stanford, Tech. Rep., 2017. [Online]. Available: https://github.com/garythung/trashnethttp://cs229.stanford.edu/proj2016/poster/ThungYang-ClassificationOfTrashForRecyclabilityStatus-poster.pdf

[138] G. E. Sakr, M. Mokbel, A. Darwich, M. N. Khneisser, and A. Hadi, "Comparing deep learning and support vector machines for autonomous waste sorting," in *2016 IEEE International Multidisciplinary Conference on Engineering Technology, IMCET 2016*. Institute of Electrical and Electronics Engineers Inc., 12 2016, pp. 207–212.

[139] O. Awe, R. Mengistu, and V. Sreedhar, "Final Report: Smart Trash Net: Waste Localization and Classification," Tech. Rep., 2017.

[140] R. A. Aral, S. R. Keskin, M. Kaya, and M. Haciömeroğlu, "Classification of TrashNet Dataset Based on Deep Learning Models," in *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*. Institute of Electrical and Electronics Engineers Inc., 1 2019, pp. 2058–2062.

[141] C. Bircanoglu, M. Atay, F. Beser, O. Genc, and M. A. Kizrak, "RecycleNet: Intelligent Waste Sorting Using Deep Neural Networks," in *2018 IEEE (SMC) International Conference on Innovations in Intelligent Systems and Applications, INISTA 2018*. Institute of Electrical and Electronics Engineers Inc., 9 2018.

[142] M. Satvilkar, "Image Based Trash Classification using Machine Learning Algorithms for Recyclability Status," 2018.

[143] U. Özkaya and L. Seyfi, "Fine-Tuning Models Comparisons on Garbage Classification for Recyclability," Tech. Rep.

[144] G. Mittal, K. B. Yagnik, M. Garg, and N. C. Krishnan, "SpotGarbage: Smartphone app to detect garbage using deep learning," in *UbiComp 2016 - Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. Association for Computing Machinery, Inc, 9 2016, pp. 940–945.

[145] Microsoft Azure, "Bing Image Search API." [Online]. Available: https://azure.microsoft.com/en-gb/services/cognitive-services/bing-image-search-api/https://dev.cognitive.microsoft.com/docs/services/8336afba49a84475ba401758c0dbf749/operations/56b4433fcf5ff8098cef380c

[146] "Recycle Academy on the App Store." [Online]. Available: https://apps.apple.com/us/app/recycle-academy/id1413748789

[147] "We Recycle by OPRL." [Online]. Available: https://appadvice.com/app/we-recycle/1437817757

[148] "Recycle Wizard on the App Store." [Online]. Available: https://apps.apple.com/ca/app/recycle-wizard/id1300177023

[149] J. Siegel, A. Rawson, S. Laghate, and R. Chawla, "RUcycle," 2018. [Online]. Available: https://devpost.com/software/rucycle

[150] "HackPrinceton Spring 2018," 2018. [Online]. Available: https://hackprinceton-spr18.devpost.com

[151] "Recycle! on the App Store." [Online]. Available: https://apps.apple.com/be/app/recycle/id730904895

[152] "RecycleSmart on the App Store." [Online]. Available: https://apps.apple.com/au/app/recyclesmart/id903864242

[153] "Recycle it!! on the App Store." [Online]. Available: https://apps.apple.com/gb/app/recycle-it/id975116191

[154] "Surrey Recycles on the App Store." [Online]. Available: https://apps.apple.com/gb/app/surrey-recycles/id1142250204

[155] CleanRobotics, "TrashBot," 2019. [Online]. Available: https://cleanrobotics.com/trashbot/

[156] L. Chin, J. Lipton, M. C. Yuen, R. Kramer-Bottiglio, and D. Rus, "Automated Recycling Separation Enabled by Soft Robotic Material Classification," Tech. Rep.

[157] L. Chin, M. C. Yuen, J. Lipton, L. H. Trueba, R. Kramer-Bottiglio, and D. Rus, "A Simple Electric Soft Robotic Gripper with High-Deformation Haptic Feedback," Tech. Rep.

[158] I. Mazzotta, F. De Rosis, and V. Carofiglio, "IEEE INTELLIGENT SYSTEMS Portia: A User-Adapted Persuasion System in the Healthy-Eating Domain," Tech. Rep., 2007. [Online]. Available: www.computer.org/intelligent

[159] M.-C. Chiu, S.-P. Chang, Y.-C. Chang, H.-H. Chu, C. C.-H. Chen, F.-H. Hsiao, and J.-C. Ko, *Playful Bottle: a Mobile Social Persuasion System to Motivate Healthy Water Intake*, 2009.

[160] F. Orji, R. Deters, J. Greer, and J. Vassileva, "ClassApp: A Motivational Course-level App," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2018, pp. 49–53.

[161] "Ofcom Communications Market Report 2018," Tech. Rep., 2018.

[162] D. Chaffey, "Mobile Marketing Statistics Compilation," 2019. [Online]. Available: https://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/

[163] "MoSCoW Method," 2017. [Online]. Available: https://en.wikipedia.org/wiki/MoSCoW_method#cite_note-1

[164] J. Lin, C. Miao, Z. Shen, and W. Sun, "Goal Oriented Agile Unified Process (GOAUP): An Educational Case Study," in *International Conference on Software Engineering and Computer Science (ICSECS)*, Yichang, 2013. [Online]. Available: https://www.researchgate.net/publication/259885800_Goal_Oriented_Agile_Unified_Process_GOAUP_An_Educational_Case_Study

[165] A. Ghahrai, "Software Development Methodologies," 2016. [Online]. Available: https://devqa.io/software-development-methodologies/

[166] "MVC Framework - Introduction," 2020. [Online]. Available: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

[167] "Apple I," 2020. [Online]. Available: https://en.wikipedia.org/wiki/Apple_I

[168] Apple Developer Documentation, "Human Interface Guidelines." [Online]. Available: https://developer.apple.com/design/human-interface-guidelines/

[169] Apple Inc, "coremltools," 2017. [Online]. Available: https://apple.github.io/coremltools/

[170] T. E. de Campos, B. R. Babu, and M. Varma, "The Chars74K Dataset: Character Recognition in Natural Images," in *Proceedings of the International Conference on Computer Vision Theory and Applications*, Lisbon, Portugal, 2009. [Online]. Available: http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/

[171] MonkeyLearn, "Text Classification." [Online]. Available: https://monkeylearn.com/text-classification/

[172] Apple Inc, "Vision — Framework." [Online]. Available: https://developer.apple.com/documentation/vision

[173] J. Sauro, "5 Things to Know About Likert Scales," 2015. [Online]. Available: https://measuringu.com/likert-scales/

[174] "Zoom," 2020. [Online]. Available: https://zoom.us

[175] Information Commissioner's Office, "What is valid consent?" 2020. [Online]. Available: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/consent/what-is-valid-consent/

[176] Imperial College London, "Ethics approval overview." [Online]. Available: https://www.imperial.ac.uk/research-ethics-committee/ethics-approval-overview/

[177] Apple Inc, "XCTest." [Online]. Available: https://developer.apple.com/documentation/xctest

[178] ——, "Using Metal System Trace in Instruments to Profile Your App." [Online]. Available: https://developer.apple.com/documentation/metal/using_metal_system_trace_in_instruments_to_profile_your_app

[179] Figma Inc., "Figma." [Online]. Available: https://www.figma.com

[180] Apple Developer, "Create ML." [Online]. Available: https://developer.apple.com/documentation/createml

[181] M. Tulio Correia Ribeiro, "LIME Analysis." [Online]. Available: https://github.com/marcotcr/lime

[182] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why should i trust you?" Explaining the predictions of any classifier," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-August-2016. Association for Computing Machinery, 8 2016, pp. 1135–1144.

[183] "Decrypting your Machine Learning model using LIME - Towards Data Science." [Online]. Available: https://towardsdatascience.com/decrypting-your-machine-learning-model-using-lime-5adc035109b5

[184] Apple Inc, "AVCaptureSession." [Online]. Available: https://developer.apple.com/documentation/avfoundation/avcapturesession

[185] "Recognizing Objects in Live Capture — Apple Developer Documentation." [Online]. Available: https://developer.apple.com/documentation/vision/recognizing_objects_in_live_capture

[186] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885603001379

[187] Apple Inc, "VNTranslationImageRegistrationRequest." [Online]. Available: https://developer.apple.com/documentation/vision/vntranslationalimageregistrationrequest

[188] R. Lauer, "Offloading Tasks to Worker Threads with NativeScript," 2019. [Online]. Available: https://www.nativescript.org/blog/offloading-tasks-to-worker-threads-with-nativescript

[189] N. Kledzik and L. Gerbarg, "Optimizing App Startup Time," 2016. [Online]. Available: https://developer.apple.com/videos/play/wwdc2016/406/

[190] D. Koutsogiorgas, D. Tomlinson, E. Amorde, O. Therox, P. Beusterien, S. Giddins, and et al, "CocoaPods." [Online]. Available: https://cocoapods.org

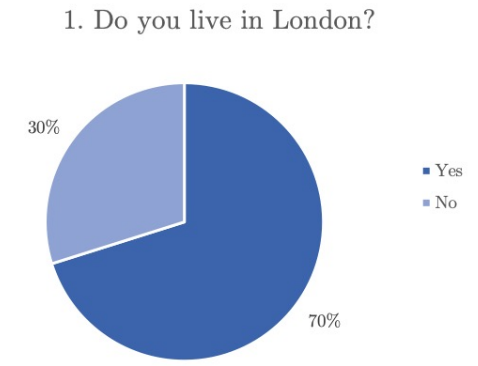
[191] V. Vohra, "Deep Learning based Waste Segregation Project to classify waste images into different classes," 2019. [Online]. Available: https://github.com/vasantvohra/TrashNet

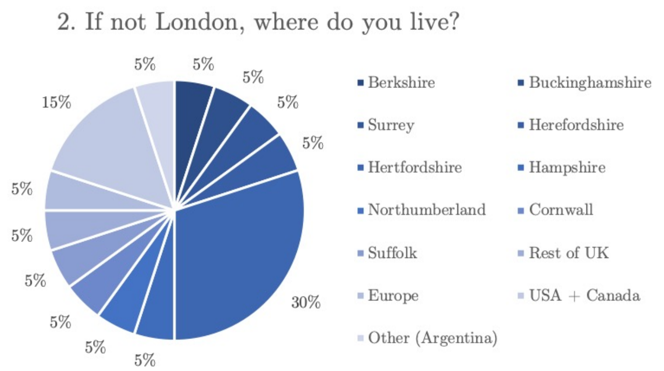
[192] Apple, "Core Location Framework." [Online]. Available: https://developer.apple.com/documentation/corelocation/

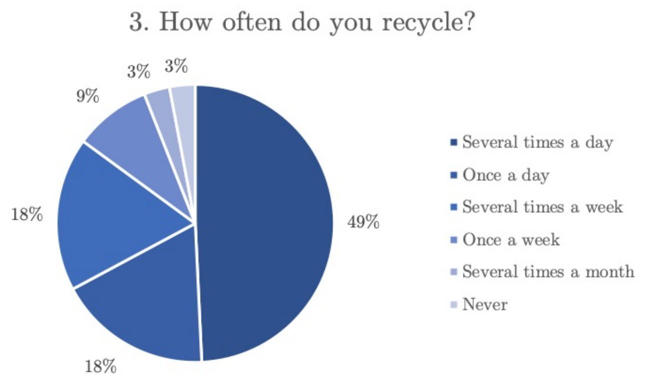
[193] ——, "Getting the User's Location." [Online]. Available: https://developer.apple.com/documentation/corelocation/getting_the_user_s_location

Appendix A

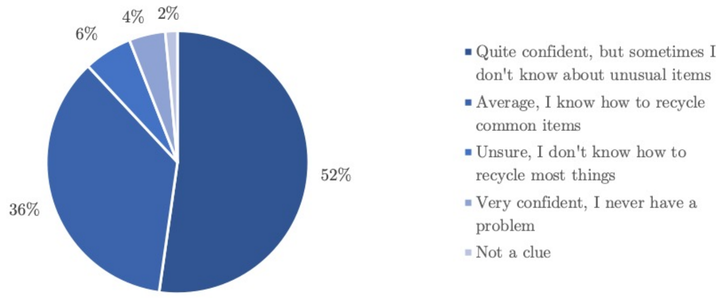
Initial Insights Survey Results



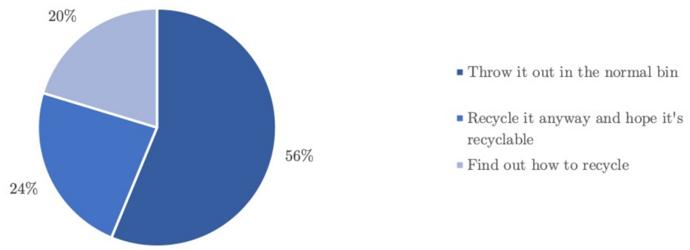




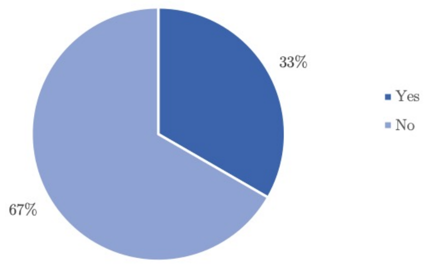
4. How confident are you about recycling?



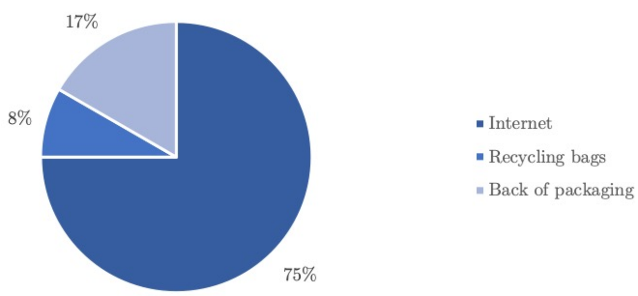
5. When you're unsure about how to recycle an item, what do you do?



6. Are you aware that if you place a non-recyclable item in a recycling collection, it creates a risk that the whole collection will not be recycled?

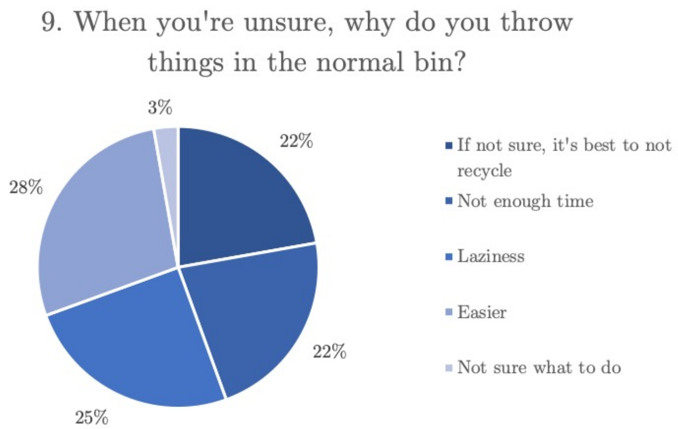


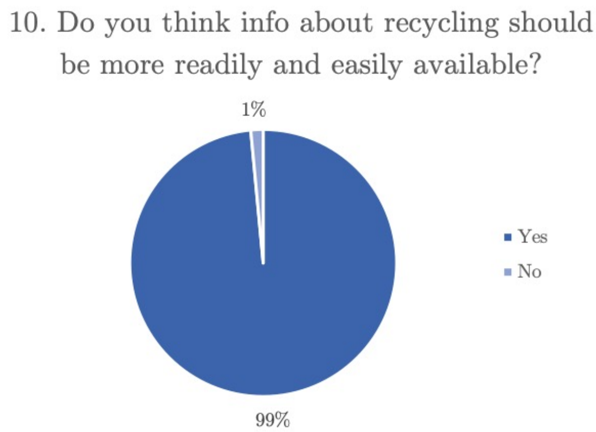
7. When you're unsure, where do you find out how to recycle something?

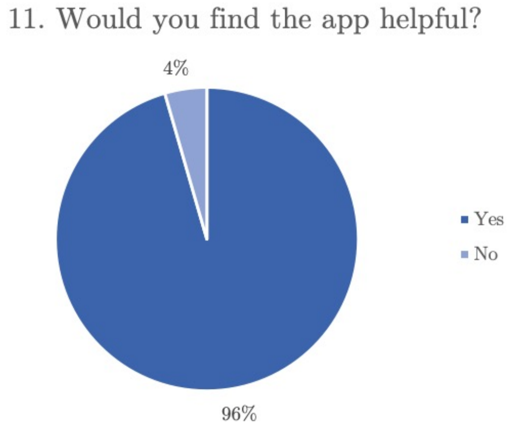


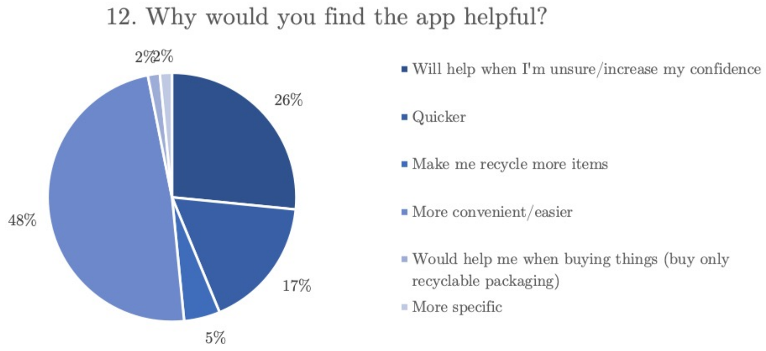
8. What do you think about the standard of the accessibility and information provided about recycling?

- "It's okay, but would prefer if there was a more straightforward and clear way to find out"
- "Average, there is information but it's not specific to my area (or at least not readily apparent without additional research)"
- "OK"
- "Time consuming and do not always manage to find the full information out"
- "Average"
- "There is limited information on my borough's website"
- "Generally OK"
- "Good - local council website (where I end up from Google) is clear"
- "Usually accurate"
- "The information is helpful for common items but often unclear/limited for unusual items"
- "Pretty good"
- "Accessible, inaccurate"
- "Usually it's okay but sometimes I do not understand the symbols so put it in the normal bin"









13. Why would you not find the app helpful?

- "It would have to be very sophisticated as recycling is not as simple as recognising item but constitutes of each individual item. For instance, plastic degrades in quality when it is recycled and after a couple cycles will end in landfill - so identifying a plastic bottle of water seems irrelevant in grand scheme. Only real solution is making and consuming less plastic. Single-use coffee cups have a thin, plastic coating inside the cup which make it difficult to separate in recycling plants and lid is a soft plastic so tend to fragment into small, unusable pieces. It's debatable as some experts recommend they should be thrown straight into the trash bin."
- "I know what the item is, just being able to search for it on the app would be easier."
- "It will not work. You would have to fully scan the inside and outside of a product. For example, any packaging with fat on it or any other food waste cannot be recycled. It would be a terrible user experience as all of the packaging would have to be scanned, including the inside (no idea how you're planning on getting a camera in a small piece of packaging) and would take way to long. I also am sceptical of the program that you would be able to create being able to distinguish between different grades and types of plastic, normally not noticeable even to the human eye and only by feel and density. Even if all of this was possible to over come you have not gotten around the fact humans are lazy, and would never spend the extra 3 seconds you are proposing to scan an item. They would just throw it in the normal rubbish."

14. What other features would you like or would you find helpful in a recycling app?
Technical:

- Compatibility with all current smartphones
- Barcode scanning
- Search for the item when you know what it is
- Check for cleanliness

User Info:

- Usage stats and tracking
- See how your friends are doing

Education:

- Training Quizzes

- If not recyclable, other sustainable ways of getting rid of an item (e.g. Ecobricks)
- Information about recycling labels
- Recycling 'fun facts'
- Information about the benefit of recycling
- Tips to increase sustainability
- Recycling instructions

Motivation:

- Points and rewards

Location:

- Information about the nearest recycling bin/facility
- Map of recycling facilities
- List of recyclable items in postcodes
- How well the local area does with respect to recycling
- Bin days
- Where to get recycling bags

Appendix B

User Testing Round 1 Questions

Section 1: Insights and general information

1. Could you tell me a bit more about your current recycling habits.
2. Is there a particular time in the day/day in the week that you do your recycling?
3. When you're out and about, do you usually recycle things?
4. What are your main motivations and reasons for your recycling behaviour?
5. How would you rate your recycling ability / knowledge? (1 - 5)
6. How likely are you to recycle something when you're at home? (1 - 5)
7. How likely are you to recycle something when you're out? (1 - 5)

Section 2: Using the app

1. Could you show me how you would use this app to identify how to recycle something?
2. What did you expect would happen?
3. Was that the expected outcome?
4. Could you tell me how you interpreted the yellow borders?
5. Could you tell me what actions you would take based on the displayed information?
6. Could you show me how you would go about scanning a second item?

Section 3: Rating the app

1. Rate the usability of the app (1 - 5)
2. How easy to understand was the information? (1 - 5)
3. How did you feel when using the app? (1 - 5)
4. How often do you think you would use such an app? (1 - 5)

Appendix C

User Testing Round 2 Questions

Introduction: Starter questions

A. What iOS Version is your phone running?
B. Where do you live?

Section 1: Insights and general information

1. How much do you consider yourself as someone who recycles? (1 - 5)
2. Out of all waste you produce, how much do you recycle in an average (lockdown) week? (1 - 5)
3. Out of all waste you produce, how much do you recycle in an average (non-lockdown) week? (1 - 5)
4. How confident are you in knowing how to recycle things correctly? (1 - 5)
5. How motivated are you to recycle correctly? (1 - 5)
6. How likely would you say you are to recycle when at home (lockdown)? (1 - 5)
7. How likely would you say you are to recycle when at home (non-lockdown)? (1 - 5)
8. How likely would you say you are to recycle when out and about? (1 - 5)
9. What prevents you from recycling more?

Section 2: Before using the app

If you could have an app that helped you to recycle better / more...

10. What features would you find useful to help you recycle more?
11. Why / What problems would this address?
12. What features would you like?

Section 3: After using the app

13. Was the app what you expected?
14. Why / Why Not?
15. Which classification method did you prefer?
16. Why?
17. Did the layout work for you / make sense?
18. How did using the app make you feel / how was the user experience?
19. How did you feel about your onboarding experience?
20. How clear / easy to follow was the information? (1 - 5)
21. How easy / intuitive was the app to use? (1 - 5)
22. How helpful is the app? (1 - 5)
23. How likely are you to use this app? (1 - 5)
24. How helpful was the onboarding experience? (1 - 5)
25. How likely are you to recycle when at home (lockdown) with this app? (1 - 5)
26. How likely are you to recycle when at home (non-lockdown) with this app? (1 - 5)
27. How likely are you to recycle when out and about with this app? (1 - 5)
28. Any suggestions, comments, improvements or ideas?

Appendix D

User Testing Round 3 Questions

This can be found in the github repository, at:

https://github.com/rch16/RecycleHelper/tree/master/Testing/Usability/Round%203