

# Project Plan

## 1. Goal Description

With the threat of climate change, recycling is more important than ever. However, this can be extremely time consuming and error-prone. What if there was a *smart bin* that sorted your trash into different categories - meaning all the user would have to do is bin it.

### 1.1 Overview

In 2017 only 45.7% of UK waste was recycled. According to a 2011 poll in the USA, ([Ipsos Public Affairs, 13/07/2011](#)) the two biggest reasons for not recycling are:

- It's not accessible or convenient,
- It takes up too much time.

Our all-in-one bin, *Recyclotron*, aims to solve these issues the consumer faces when recycling.

Firstly, Recyclotron scans and decides whether an object is recyclable or not, reducing user interaction to one bin. This removes any confusion for consumers about the various recycling categories - making the experience more convenient and increasing the amount of waste properly recycled.

### 1.2 Previous Projects

The "smart bin" is not a new concept. In 2017 a Polish company introduced Bin.E (<http://bine.world/>), which performs a similar function to what we are proposing, except marketing it for corporations and offices. The telegraph reported ([Katie Morley 2017](#)), that one bin cost £430, with an optional £107 monthly subscription fee for regular waste collection and disposal.

Our final product will differ from them in modularity: allowing users to attach more bins and define new recycling categories as their needs change. Our system will be sleeker - current products suggest a recycling category but require confirmation from the user. We are also aiming for markets outside of offices, such as households, as our bin will be more compact and more affordable. Overall we expect our prototype to cost a total of £110 to make, with a selling price of £200 - half the price of Bin.E.

### 1.3 Functionalities

We want our bin to detect, identify and sort rubbish into the correct category so that consumer does not have to. In order to accomplish this we aim for Recyclotron to perform the following functions:

1. Detect if object is there
2. Decide what object is, assigning category accordingly (e.g. Can → Metal)
3. Move slider over bins, and then dropping rubbish into correct bin
4. Modularity of bins, so that more bins can be added to the system
5. Compressing/squashing rubbish once sorted
6. A feedback system (monitor/lights) allowing user to see what category it was sorted into and which bins are full

## 1.4 Constraints, Assumptions and Limits

In order to realistically achieve our goals and functionalities, we have made assumptions to limit the prototype. They are as follows:

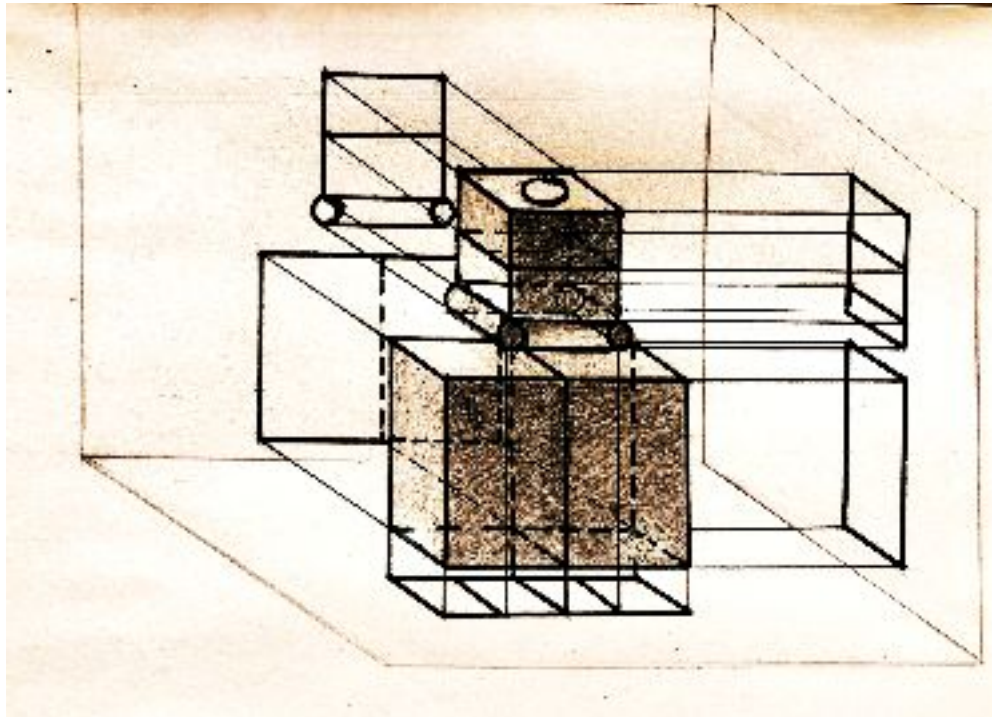
- Only dry solid recycling is sorted, with no liquids present.
- Only consider specific objects such as cans or plastic bottles (Not all types of rubbish will be identified).
- Only one object is placed into the bin and disposed of at any time.
- Objects are limited in size to be no smaller than your hand or any larger than a 750ml plastic bottle.

## 1.5 System Diagram and Mechanisms

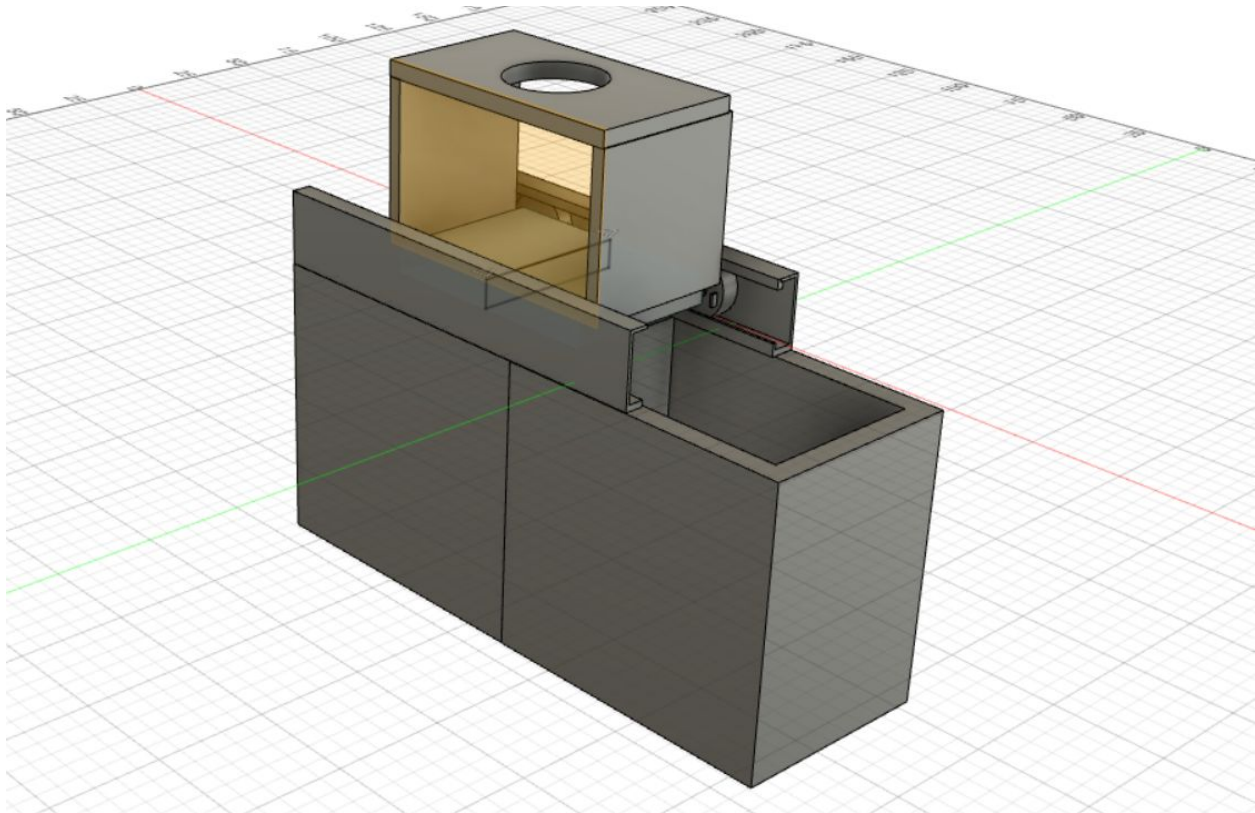
Our initial design includes two bins with a railing (like train tracks) along two parallel edges, with wheels running along those rails. There is a smaller detecting chamber on top of the two bins where a user places an object inside the detecting chamber, which contains a sensors (video camera) inside identifying whether the object is recyclable or not. The detecting chamber then moves over the correct bin (recycling) and the trap door opens, causing the can to fall into the correct bin.

After speaking to Gary and Traiko, we think our idea is completely feasible. All the mechanisms we need are available. For later iterations we want to be able to extend this system, with more bins and “reprogrammable” bins which allows the user to assign bins A and B to *recyclable* objects and C to *other* objects.

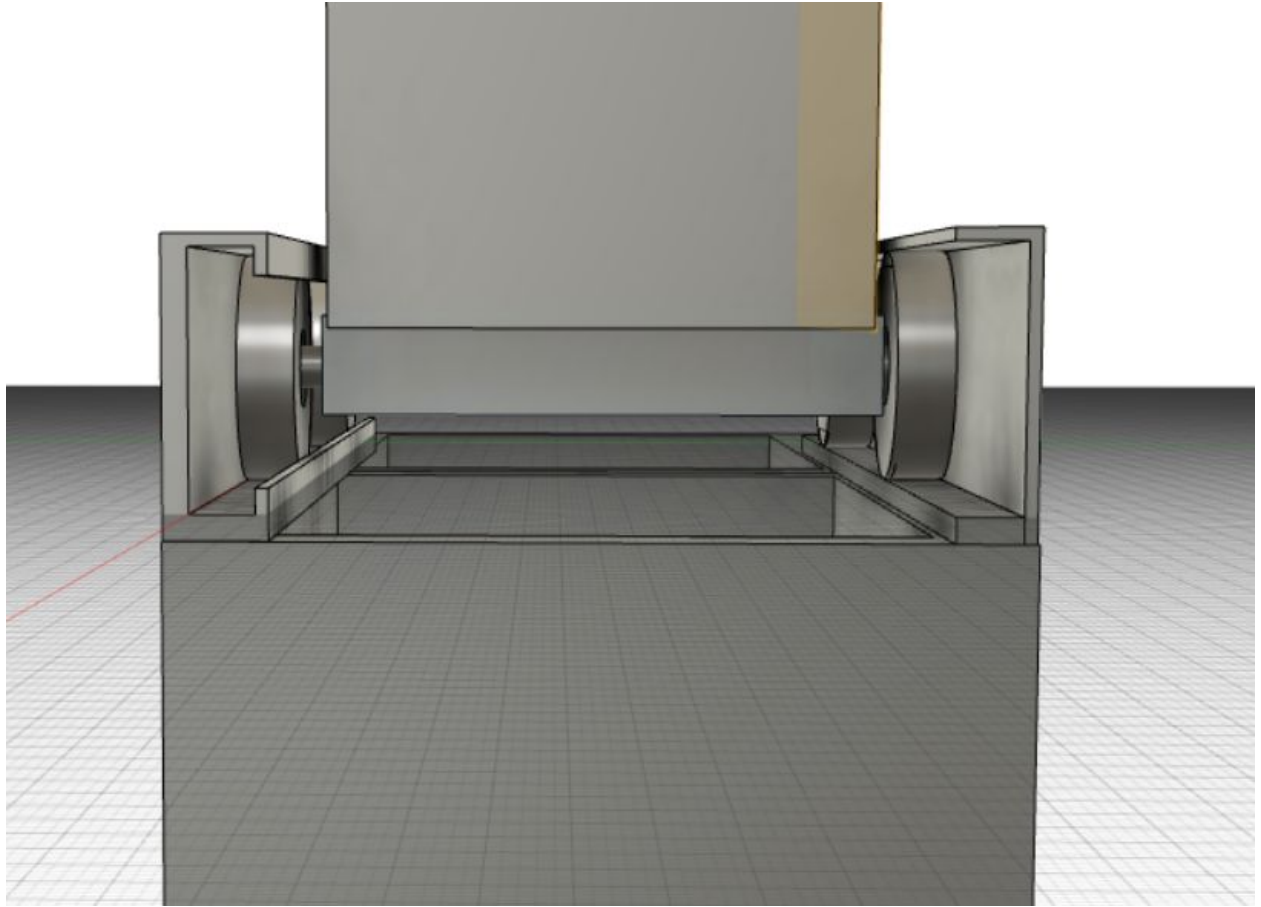
The basic design with two bins can be seen in the following sketches.



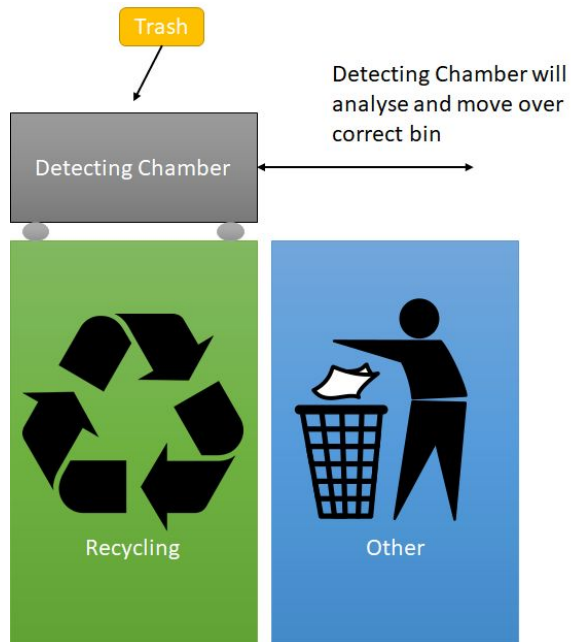
The early sketch above, shows two bins with a moving detection chamber with a trapdoor mechanism.



The CAD model above, goes more into detail about what we envision the bins to look like. In the later stages we want to include modularity, so a user can add and remove bins to accommodate more categories of trash.



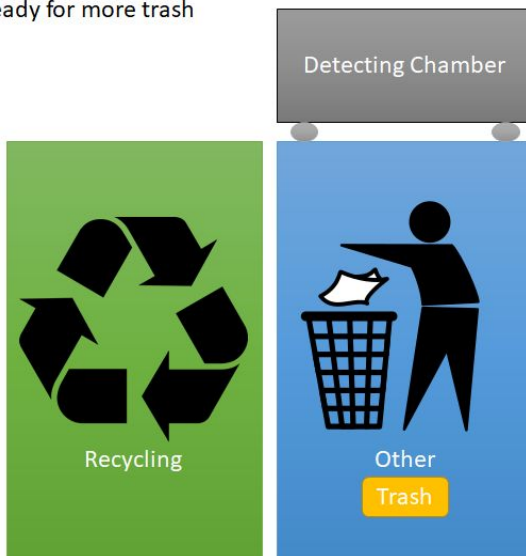
The CAD model shows the details of the track.



Trapdoor within Detecting Chamber will open, and trash will fall into correct bin



Trapdoor closes and Detecting Chamber is now ready for more trash



## 2. Task planning

### 2.1 Milestones and Task Description

We have structured our milestones to coincide with the deadlines of our demonstrations, that way we keep progressing and re-evaluate if the goals are met or not.

In the project management workshop, they recommend we avoid allocating specific durations of time for tasks. Instead, we use a metric of difficulty for each task. There are three levels, easy, medium and hard tasks. Easy tasks are menial and can be completed in a day. Medium tasks take 1-2 days of work, and hard tasks take 3-5 days of work. A day is not necessarily 20 hours.

Demo 1 - Aiming to be done by the 1st of February, so report can be submitted at 4pm Monday 3rd

*Demo 1.1 - Be able to recognise a metal can, and output a signal indicating metal recycling*

<i>Sub-Task</i>	<i>Difficulty</i>	<i>Evidence</i>	<i>Prerequisite Tasks</i>
Obtain images of cans and other generic waste items	Easy	Images are uploaded into the Github	None
Identify best metric to compare models and their feasibility in real life	Medium	Show one metric has a higher successful detection rate	None
Accurately identify if a can is present in a given image	Hard	The ML successfully identifies a metal can 70% of the time	Obtain images & Identify a metric
Create a table which maps common objects to a recycling category	Easy	The table file is uploaded to the GitHub	None
Take a photo with a webcam and output a true or false if recognised the EV3	Medium	The ML system takes an object and output a signal of its type	All of the above
Testing and Evaluation	Medium	The ML system correctly detects and outputs a metal can 70% of the time	All of the above



*Demo 1.2 - Be able to move object from one bin to another*

<i>Sub-Task</i>	<i>Difficulty</i>	<i>Evidence</i>	<i>Prerequisite Tasks</i>
Attach wheels onto two rails	Medium	The wheels are able to roll along the rails without falling off	None
Attach chamber platform in between two rails	Medium	The wheels are firmly attached onto the chamber. Show platform can move freely along the rails	Wheels are attached onto rails
Programming movement of wheels over each bin	Hard	The program is able to control the chamber to move in either directions on the rail	Attach chamber platform
Receive arbitrary signal from laptop then move for certain duration to bin	Hard	The program is able to control the chamber to accurately move itself to a given position	Programming movement of wheels
Testing and Evaluation	Medium	When chamber receives signal for metal can, it moves over recyclable	All of the above

Demo 2 - Aiming to be done by the 21st of February, so report can be submitted at 4pm  
Monday 24th

*Demo 2.1 - We can drop objects into a bin using a trapdoor*

Sub-Task	Difficulty	Evidence	Prerequisite Tasks
Attach slider to motors	Medium	Be able to extend and retract the sliders under some time limit	None
Drop the trash into one of the bins	Medium	Precisely align with the centre of either bin and release the trash	Slider to release rubbish
Testing and Evaluation	Medium	Rubbish falls directly into bin after moving	All of the above

*Demo 2.2 - The hardware system and the ML auto-detection system are fully integrated*

<i>Sub-Task</i>	<i>Difficulty</i>	<i>Evidence</i>	<i>Prerequisite Tasks</i>
Fit the camera	Medium	Camera remains operational before and after moving	None
Fit Raspberry Pi	Medium	The Raspberry Pi remains operational before and after moving	None
Photo trigger starts object detection	Hard	The photo detects the object and outputs a signal	Fit camera
Detect object using cameras	Easy	System still identifies a metal can 70% of the time	Photo Trigger
Output signal to move motors	Hard	Successfully identify and correctly dump 70% of all trash to their target bins	Detect object using camera
Testing and Evaluation	Medium	Metal can is detected and deposited in correct bin 70% of time	All of the above

Demo 3 - Aiming to be done by the 21st of February, so report can be submitted at 4pm  
Monday 24th

### *Demo 3.1 - We can classify more trash objects*

<i>Sub-Task</i>	<i>Difficulty</i>	<i>Evidence</i>	<i>Prerequisite Tasks</i>
Increase training data to include plastic bottles	Hard	Increase of the detection rate of the ML system to 80%, show bottles can now be detected	Demo 1.1 - Be able to recognise a metal can
Testing and Evaluation	Medium	As above	All of the above

### *Demo 3.2 - The user can receive feedback using monitor and lights*

<i>Sub-Task</i>	<i>Difficulty</i>	<i>Evidence</i>	<i>Prerequisite Tasks</i>
Detect if a bin is full using IR sensor	Medium	An output signal from IR sensor when the bin is full	None
Turn on the light of the bin when is full	Easy	The light should be on for any bin that is full	Detect if a bin is full
Leave a message on the monitor	Medium	A message that tells precisely which bin(s) is full	Detect if a bin is full
Exception handling, e.g. if bin is full	Medium	Produce error messages for full bins	Leave a message on the monitor
Testing and Evaluation	Medium	Show a bin is full with message on monitor only when bin is full	All of the above

Demo 4 - Aiming to be done by the 27th of March, so report and user guide can be submitted on 30th March.

*Demo 4.1 - Rubbish is compressed to reduce space*

<i>Sub-Task</i>	<i>Difficulty</i>	<i>Evidence</i>	<i>Prerequisite Tasks</i>
Create extra chamber with crushing walls	Hard	Show you can crush rubbish in chamber	None
Attach crusher to platform below detecting chamber	Medium	Show crusher attached between bin and detector	Create crushing chamber
Add code to crush before dropping into bin	Medium	Show rubbish can be crushed before being dropped	Create crushing chamber
Testing and Evaluation	Medium	Rubbish is crushed after recognised, and then dropping	All of the above

*Demo 4.2 - User can train the bin on new objects it has not seen before*

<i>Sub-Task</i>	<i>Difficulty</i>	<i>Evidence</i>	<i>Prerequisite Tasks</i>
Identify an unseen object	Medium	Pick out the glass bottle with categorising probabilities lower than a threshold	Demo 3.2 - we can detect more categories of trash
Classify the new object with appropriate category	Hard	Glass bottle is correctly labelled and stored into the database	Identify the unseen object
Testing and Evaluation	Medium	3 new objects can be added and categorised by a user	All of the above

## **2.2 Gantt Chart**

### **2.3 Resource Distribution**

Not all individuals will be able to attend workshops, thus we have decided to allocate around three workshops per person which totals around 3 hours per person. We have weekly meetings with our mentor and without which total around 2 hours from each person plus an extra hour for smaller meetings for each subgroup.

Outside of weekly meetings we have decided to spend around 2 hours of extra meetings to outline each report, not everyone will have to work on the report however everyone is expected to proofread and contribute. However, the bulk of report/writing will be done by David, Sakib and Fraser. The user guide will be allocated more time.

We will allocate 2 hours per demo, an hour before the demo and an hour after. This will allow us to prepare for the demo (excludes testing time) and iron out any bugs before the demo as well as an hour after to digest feedback afterwards.

We have decided to split the team, one to work on hardware and another to work on software and machine learning. This was done as these two parts can be developed in parallel before integration.

This means we split our total time of 2000 hours (10 people x 200 hours) into 5 distinct sections for each demo. 60% of time is allocated to create and complete tasks for the next milestones, 30% to integrate and the remaining 10% for evaluation, meetings and reports. Sprints take up 90% of our time.

We will be doing four sprints (see section 3.4), one per demo.

## **2.4 Resources Available**

### **2.4.1 Software**

The Recyclctron will detect the recycling category via a camera, which will take an image of the most recent waste placed in the bin, and convert it into a feature vector. This vector is then compared to a series of feature vectors of various garbage items, and new waste is then recycled in the same category as the sample it most closely matches.

To accomplish this, we use a variety of Python packages - Lycon for basic image operations, Torch to obtain the feature vectors, and Numpy for common mathematical operations like matrix multiplication.

#### ***2.4.2 Python Skills***

Everyone of our team member has experience using Python. The Machine Learning system and the Raspberry PI that controls the sorting mechanism will use Python, enabling us to better focus on tasks unrelated to the coding language itself.

#### ***2.4.3 Hardware Skills***

Although this is our weakest area, we have Martin and Sakib who have experience in both hardware and electronics. In order to combat our lack of skills we are teaming up the most skilled members with the least skilled members. By working together, the novice will learn and improve by following the example and guidance..

#### ***2.4.4 Machine Learning Skills***

The majority of our team is experienced in machine learning, with many taking IAML. Zhixing has made a similar system to a garbage classifier using sound and picture on a Raspberry PI, and is experienced in YOLO algorithms.

#### ***2.4.5 Vision Skills***

Anshul and Fraser have also done IVR, so have some experience with robotic and artificial vision technology.

#### ***2.4.6 Experts***

Experts are available for us to visit and ask questions for them to answer. To name a few, we are planning to contact Traiko on hardware issues, Evripidis for vision suggestions and Miruna for Machine Learning queries.

#### ***2.4.7 Hardware Equipment***

Here is a prepared list of hardware needed for the second demo.

<i>Equipment</i>	<i>Cost</i>	<i>Quantity</i>
Higher end Raspberry PI	£50	1
Webcams	£20	2
Wheel motors for EV3	£20	4
3D Printed Containers	£20	2

## 2.4.8 Financial Budget

As outlined above, we are aiming to build the prototype for only £110. This allows us to make a much more affordable product than other smart bins out there.

In order to ensure we do not exceed our budget of £200, we have a spreadsheet to track purchases. ([https://docs.google.com/spreadsheets/d/1VbvoNFD5\\_1hJZVhgyM6V8y6CXzQkSwYuV336kpg4A/edit#gid=0](https://docs.google.com/spreadsheets/d/1VbvoNFD5_1hJZVhgyM6V8y6CXzQkSwYuV336kpg4A/edit#gid=0))

Whenever a member needs to buy something, they must first confirm with either David or Sakib before purchase. We aim to only buy materials once we need them. This should prevent unnecessary costs to the budget, although it may introduce delays.

## 2.5 Risk Assessment and Contingencies

<i>Risk</i>	<i>Impact</i>	<i>Likelihood</i>	<i>Prevention</i>	<i>Contingency Plan</i>
The cameras may not detect an object being present in the environment	<b>Severe</b> - How can we sort something when we can't see it at all?	<b>Medium</b> - It's quite possible we've overestimated the accuracy of the camera	Once an object enters the chamber, a photo will be taken, and the image will be compared to previous images to see if an object is present.	Use of other sensors, weight, IR, Ultrasonic.
One camera may not recognise an object's category	<b>Severe</b> - Critical failure as the waste object may be recycled in a default category or the wrong category	<b>High</b> - Poor lighting may disrupt the sensors, or a lack of training data could disrupt classification	Obtaining lots of training data on common trash items and employing 2 cameras to capture different angles.	Unidentified waste can be placed into the generic 'trash' category to avoid mis-recycling.
The trash object is not recognisable at all	<b>Low</b> - We might not be able to recycle that trash even if it is recyclable	<b>Low</b> - this all depends on the user and the trash object presented, which we can control	The machine will try to learn the category of this trash by asking the user input from the buttons	Additional sensors should be able to identify the material of the object.



			on each bin. If no input received, the system will put it into 'others'.	
The rail-running system may be too weak to move the rubbish.	<b>Severe</b> - This is essential, and if it fails then we have no moving parts.	<b>Medium</b> - hardware is our weakest area as a team, so it's possible we've underestimated this.	By having a larger team and prioritising this first, we hope to test this as soon as possible, and make the required modifications.	We'll redesign the system to use a conveyor belt instead of a moving trapdoor.
Rail wheels cannot be moved at the same time.	<b>Severe</b> - If wheels only rotate in sequential order, and not all at once the chamber will not move	<b>High</b> - we are using EV3 wheels which we cannot get running in parallel	Researching online (SDP wiki etc) and asking experts how we can move the EV3 wheels in parallel	Looping each wheel to spin for 0.1 milliseconds sequentially still moves the chamber
Container doesn't know where it is after moving.	<b>Severe</b> - rubbish cannot be sorted correctly if we don't know where to drop it	<b>High</b> - we have no sensors	We are "hardcoding" the movement of the chamber, so it will always move the same length each time, and will update where it is in relation to the bins after moving.	Adding IR sensors at the boundaries of bins will allow us to track where the chamber is
Hardware team does not have enough experience to construct rail system	<b>Severe</b> - The rail system is needed to dispose of the rubbish	<b>Medium</b> - Only Martin and Sakib have experience in hardware, so it's quite possible they cannot complete the task on their own.	Pairing most experienced person with most inexperienced.	We'll redesign the system to use a conveyor belt instead of a moving sliding door.
Software team relies too heavily on Zhixing's ideas and experience	<b>Medium</b> - Zhixing has extensive machine learning knowledge and is very enthusiastic	<b>Medium</b> - Zhixing could fall ill or switch to a better university	We will train each of software team members, as well as ensuring Zhixing communicates and teaches the rest on his ideas	We could up with a different approach using neural networks instead of YOLO, which Zhixing is experienced in.

## **Section 3: Group Organisation**

### **3.1 Code-sharing and Communication**

As a team we are going to use Slack primarily for announcements, meetings, questions and suggestions. We also will use Facebook Messenger as an alternative for urgent and less formal messages.

We plan to use Github for software related activities. As it allows any member of the team to upload, view and edit code. Version control is extremely important in a project like this, so the ability to roll back to a previous version of working code is fundamental, and no doubt will be helpful for demos.

Github was also chosen over other platforms as it allows users to work on multiple branches of code at the same time. In addition, there is also a system for reporting bugs within the system, giving steps to reproduce, etc. This means we can all interact with each other's code fluidly, without disrupting each other's progress.

### **3.2 Team Roles**

We have decided to organise into two teams to start off with; a hardware team to design and make the mechanism for trash disposal, and a software team to design and implement item recognition. This will allow everybody to be working on something, avoiding time being wasted by waiting for tasks to be completed.

After we have functional prototypes for each system, then we will integrate both systems and delegate more people to that task.

#### **3.1.1 Team Leaders**

Both David and Sakib will function as group leaders, with Zhixing directing the machine learning team, Sakib directing the hardware team and David functioning as secretary and manager.

David has experience teaching python and organising small teams. He will prioritise his time evaluating tasks, ensuring teams are aware of direction and tracking progress. He will also be involved in writing reports and is the designated group contact.

Sakib will function as the project manager. He has experience encouraging teams and tutoring maths. He will be guiding overall design, providing direction, leading and motivating.

#### **3.1.2 Main Teams**

Hardware - Martin, Anshul, Flora, Rebecca, Sakib

Software - Shivamm, Zhao, Fraser, Zhixing

#### **3.1.4 Other Roles**

Reports - Sakib, Davy, Fraser, Zhao

Evaluation - Sakib and Davy

Integration - Zhao, Zhixing, Flora, Rebecca, Anshul

The Integration team tests and integrates the hardware and software systems at each stage of the development, connecting the systems together. Integration will be lead by Zhao.

### **3.2 Meetings**

As a group we have planned to meet every Tuesday with our mentor, and here we will inform the mentor of our plans for the week ahead, what we have accomplished and listen to any feedback he has.

After each meeting David will post an announcement summarising what we discussed and what needs done. This ensures not only are absentees kept in the loop, but everyone is reminded of the jobs they are responsible for.

Outside of that, the two teams will also always meet once a week separately to delegate tasks and work. Afterwards the team will either inform David or Sakib who is responsible for a task, or update it on Trello themselves.

Other extra meetings will be scheduled at the discretion of the sub-teams with the frequency being proportional to the current workload required to reach the next milestone.

### **3.4 Agile Method**

We have decided to use Agile practices, to promote communication and adaptability when integrating two independent systems. Our milestones are clear in what we are seeking to achieve, and an agile schedule allocates enough time to review, evaluate and update our work.

We will be conducting weekly/bi-weekly sprints where all members of the group will try to get through as many tasks on Trello as possible for a given milestone. Then the team will all switch over to integration phase. This is done so that we can make extremely fast progress but also leaves us plenty of time to make sure that all parts of the system work correctly with each other.

### **3.5 Task Allocation and Progress Tracking**

Tasks will be allocated within teams according to skills, and while multiple people may work on the same task, only one individual will be responsible for each task. This way one person can always be held responsible and it is their duty to rally their sub-team to complete a given task.

As each task only has one “representative”, progress can be tracked by regular communication between David/Sakib and the representative.

Trello will be our primary way of accomplishing this, as it can be accessed from anywhere by anyone in the group.

Once a task is assigned as an individual's responsibility, they will attached to that task on Trello. When the task is done they will move it into the "To be Evaluated" list. Only once David or Sakib have verified through testing that it is complete will it be moved into the "Done Tasks" list.

This is done so that there is a hierarchy of evaluation so that no task can ever be deemed complete when in reality it only partly done. This avoids extra work later down the road and reduces overall tech debt.