

DublinCity.ie Heuristic Review

Marcus Meldon, Salihah Alshamrani, Calum Haughton





Table of Contents

	Introduction	3
	Executive Summary	4
	Positives	5
	Help	6
	User Controls & Inputs	7
	Language	8
	Error Handling & Prevention	9
	Complex Workflows.....	11
	Conclusions	13
	Prototype	14



Introduction

This document contains results of a heuristic review conducted on the 'Report an Abandoned Vehicle' workflow on DublinCity.ie. The workflow was run through numerous times, attempting all different variations of the task, breaking the workflow as much as possible. With each run through, it was noted how different elements compared to Nielsen's [Usability Heuristics](#). These heuristics are intended to be a descriptive guide; they help to describe issues that crop up in the workflow and allow us to find suitable solutions. Throughout the process of reviewing, the set of heuristics were used as a lens with which to view issues. They were not used as a checklist to complete; each issue may refer to multiple heuristics within it, without explicitly listing them out.

The review shows five areas of concern that if improved, would provide huge value to the users of DublinCity.ie. Each area is a summary of multiple issues found across the workflow that fall into similar categories. Those categories are **Help, User Control & Input, Language, Error Handling & Prevention** and **Complex Workflows**.

Each issue is given a severity: Minor, Moderate, Major or Critical. It will be displayed in the right hand corner of each page, as above.



Minor

Fix it whenever



Moderate

Fix it soon



Major

Fix it urgently



Critical

Fix it now

Within each issue, multiple instances are shown. This report will highlight some, but not all of them. This is in order to provide a concise, readable report. It is by no means a definitive list of all the things that could be improved on in the workflow.

Issue Rating Panel



Frequency 
Impact 

The severity given to each issue is driven by two factors: frequency and impact. Both these factors are rated on a scale of 1-3 with each point having a specific meaning.

Frequency	
1	Happens less than half the time
2	Happens more than half the time
3	Happens everytime, without fail

Impact	
1	Aspects of the interactions are misleading or irritating, costing the user time and errors.
2	The task can only be completed with considerable difficulty and/or takes considerably more time than it should.
3	The user cannot complete the task because of the issue.



Executive Summary

The heuristic review highlighted some positive aspects of the workflow, and 5 key areas that need attention.

Positives

There are a number of good things that are shown throughout the workflow. There is evidence of some nice use of language, speaking in the user's terms in a very task focused way. There is also some good use of navigation, although this is not quite executed in the right way.

Help

There is little or no user help throughout the workflow. Individually, this is a minor problem; however it is compounded by the complexity of the workflow. When users become confused or misled, there is nowhere for them to go. They are forced to work it out for themselves. The only time help is provided, it is not in the right location. Users have to remember the help they read previously, later on in the workflow.

User Controls & Input

The site has user controls that do not fulfil a purpose, such as dropdowns with one item, or a set of tabs with only one option. There is also a problem with consistency; the site is inconsistent with itself, and also inconsistent with wider web practices. Links should look like links, and when they are clicked they should warn the user if a new tab or window will open.

Language

While there is some good language, there are also some issues. Frequently, users are shown system language. It is often un-descriptive too, offering users little to aid them in moving through the workflow. Again, there are inconsistencies with wider web practices, where clicking a link can bring you to a page with a different name.

Error Handling & Prevention

Errors cause issues throughout the workflow. At best, they are confusing, not showing users how to resolve them, and at worst, they can stop the user from completing the task. Errors shouldn't be displayed in alert boxes, as it removes the context of the page. This makes errors look like popups that users will ignore.

Complex Workflows

Smaller interactions often take much longer to complete than needed. Each step could be simplified by focusing on the user: only include things that add value for them. Multiple tools that do the same thing should be removed. At times, the workflow becomes so convoluted that users are likely to give up, meaning there is a significant user drop off at certain points. There are additional, unneeded steps that can be completed by the system, reducing the cognitive load on the user.

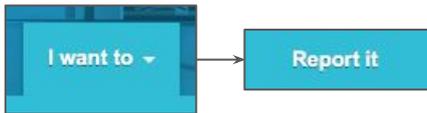
+ Positives

During the review, a number of positive aspects were noted. Some of these aspects are great ideas that could be pushed to the wider application, while others have the right core idea but need some refinement.

Simple, task focused language

From the main menu, users can select 'I want to...' and are then offered the choice of four options: 'Apply For It', 'Find It', 'Pay For It' and 'Report It'. The use of language here is excellent: it is simple and task focused. It is easily discoverable by the user, meaning they don't have to waste time searching the screen for the option that they want. As soon as they land on the screen, it is obvious and immediate where to go to find the task. It is also very clear what each section is about, and what users will be able to do once they have clicked on that option.

This simple language could be expanded to the site with relative ease, and with huge value to the user.



An example of how simple language makes tasks easily discoverable. This practice should be widened to the whole site.

Informing the user before something happens.

Small clues to the user to help pre-empt interactions are very helpful. The main menu incorporates hints like this very well. The menu uses arrow icons to indicate which menu options will open a dropdown menu. While it is a bad practice to have different behaviours in the same menu, it is a well implemented and useful feature to tell users up front where the differences lie.

Telling the user where in the process they are.

It is important throughout the process to inform users where they are, and how long is remaining in the workflow. The navigation used in the Create Service Request section has the right idea, but it is not executed as well as it could be. The core idea is good, but the navigation should be present for the entire workflow, from inception to completion. Only having two steps is also not giving the user enough information, so additional steps could be included.



This navigation is a great core idea, but needs a few extra steps to provide real value to the user.

Throughout the Dublin City Council website there is little or no help documentation for the user, leading to confusion. When help instructions are shown, they are often in the wrong place. This requires the user to remember the information to use on a different page.

No Help or documentation for the user.

While navigating through the site and performing what should be basic tasks, the user is often left perplexed or lost and the site offers no clear help documentation to users to compensate for this. Below is a clear example of this in the self-service portal section of the website while creating a service request.



Request Type **Abandoned Vehicle Inspection**

*Summary

This Summary text box does not explain what the user is meant to write into it. The text box is also marked as a required field and the user can not progress without filling it in. This summary textbox could easily be fixed by adding a small information button beside it, which the user could click and it would explain what kind of input is expected.



Details

Your Name

Your Contact Telephone Number

Even within simple forms, some explanation and help would allow the user to fill the form in with a lot less stress. The same solution can be applied again, with a small info button placed at the side of the text fields that the user can expand to get more information on the task they are doing.



The user is required to remember instructions.

The site has a set of instructions, as shown below, placed on a page for the user to read and then remember. The actual task that these instructions are written for is a few pages later, which means the user can not reference these instructions while doing the task. The user must remember them, or try to navigate backwards to go and read them again.

There are two steps to complete:

- **Step One:** Identify Problem: Type a summary of your request. Then type the name of the nearest street into the box. As you type, a list of matching streets will appear. Select the correct one from the list. Click Select Map.
 - When the map appears, follow the instructions to the left of the map to mark the location of the incident.
 - Make sure that you click the Save button  then click Next
- **Step Two:** Problem Details – Enter details of the request and click Submit*

*Please note that failure to provide your name and contact details may result in your service request not being dealt with.

This is very unhelpful for the user. They have to attempt to recall these quite intricate instructions while doing a task on a new page. The information should focus on the user's current task; information about future tasks should not be presented until it is relevant.



This simple inclusion of an info button that will give the user more help or instructions if needed is a clever way to help resolve these issues.

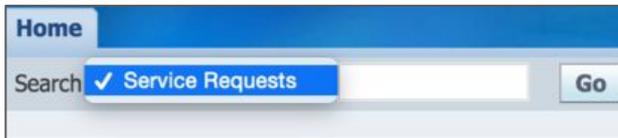
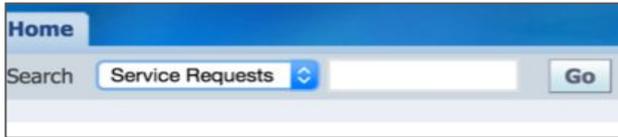


User Controls & Inputs

The website has a number of different controls and inputs that do not follow standard web practices. When users interact with them, they commit the user without giving reasonable warning, bringing them off the page they were on and without a clear way of returning.

The site does not give the user effective control and freedom.

Some of the controls in the website have no obvious purpose. There is a drop down menu that only has one item to select, making the drop down control redundant. The drop down works in conjunction with a search box for finding previously submitted requests. It appears to the user that this would be used for searching for a request type, but that is not the case.



If the user searches for 'Abandoned Vehicle Inspection', the system sends them to a blank page saying no search results found. This removes them entirely from the page they were on, with no clear way to get back.

Moderate Issue



Frequency ● ● ●
Impact ● ○ ○

The site does not have good consistency and standards.

The websites various controls, buttons and links do not follow good web standards and also are inconsistent in the website itself. It is easy for the user to get lost because of this.

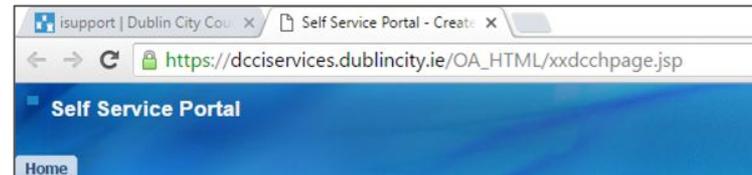
[Proceed to Create a Service Request](#)

How it Works

On the following page, you will see a list of Service Requests types. Simply click on the one that

There are two steps to complete:

The above hyperlink is an important step for the user to report something to Dublin City Council. This should be a button with a more obvious call to action, as it is the most important item on the page for the user. When displayed as a header, it is very unclear that this is where the user should click. As shown in the image below, when the user clicks on this link to move further on in the process of reporting, it sends them to a new tab and what looks like a new website. There is no obvious connection to the last website. This is inconsistent with what users expect, and does not follow good web standards. The page should be part of the original website; but failing that, icons are used to point out when things will open in a new tab.

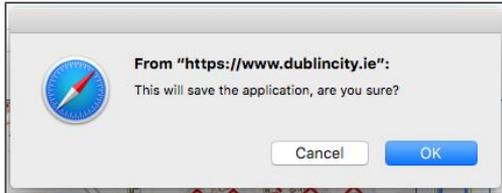


There are a number of issues with the language used across the site. It leads to confusing, misleading and frustrating workflows, that consistently slow the user down and point them in the wrong direction.

System language is regularly shown to the user.

The site uses complex and unfamiliar terms that do not follow real world patterns and conventions. This turns simple interactions into confusing challenges for the user to puzzle through.

Instances of system language are found throughout. Users are shown labels like 'Proceed to Service Request', or 'Abandoned Vehicle Inspection', which they are not familiar with. While these instances are minor, the volume of system terms becomes a major issue. It creates constant doubt in the mind of the user: 'Am I doing this the right way? I don't know'. There are also instances where system language can cause a major stumbling block.



The above is shown when a user saves the map location of a reported issue. It causes major confusion for users: 'Am I using an application - what is that? Is this saving to come back to later? Where is it going to save to? Am I sure - what happens if I don't?' This message may be clearer with a simple statement: The location of your report has been saved. The user could hit 'OK' and proceed to the next section.



Language and format used is inconsistent.

This issue is most evident in the initial steps. From the main menu, the user clicks 'I want to', which opens 4 options. Some of these options are written in title case, and others in sentence case. It is a small but noticeable problem that adds to a jarring experience for the user. The user is then brought to a page that presents another problem. The user clicks on 'Report it' but is brought to a page titled 'Create a Service Request'. This inconsistency further fosters the jarring user experience; a simple page transition becomes something that users need to think about. When a user clicks on a link, the next page must display a title that matches. It allows users to stay focused on the task, and reduces memory load.



Instructions are not descriptive enough.

In many places, while the language used is simple, it is just not descriptive enough. This is most evident in labels for form fields. It does not tell the user enough information, so they are left to guess what the intention is.



The button on the map page, shown above, is labelled 'Select Map'. This is confusing, and does not make sense to the user. Labels should describe what action the user will take: 'Search the map'.



Error Handling & Prevention

Error handling is a critical problem throughout the workflow. At best it slows and frustrates users, and at worst is so severe that users will be unable to complete the workflow. It is imperative that this problem is fixed as soon as possible, both to enable users a complete and satisfying experience, and allow the system to fully complete user requests.

Uncontextualized errors

Errors are frequently displayed as alert boxes with limited information, offering no context to the user. The nature of these error means it is difficult to track through the page to find where the error is, adding significantly to a user's cognitive load. To compound this problem, the page does not highlight anywhere to show the user where to look. The onus is on the user throughout to figure out what they have done wrong, and how they should fix it.

The sample error shown to the right highlights how difficult a user's workflow can be. The title on the alert box shows an unfamiliar URL to the user. This URL, coupled with how alert boxes are normally used, makes the error message look like a spam popup. The initial reaction to this is to close the box without reading it. To add to this problem, the user can block further popups from the page. This will mean that there is no further messaging at all, and the user is left at a complete dead end with no explanation of what to do next.

The solution to this is to use contextual, inline messaging to show users where there is a problem. Highlight the input box with a red outline, and provide a descriptive message telling the user how to fix the problem.



Little user aid in error recognition, diagnostics or recovery

Mentioned in the previous section on Language, this issue is present throughout error messages shown to the user. They frequently use system language, and do not inform the user of where the error is or how they should fix it. In the example below, the message is "Select map is mandatory for Service Request." The language is unclear, and does not follow error messaging standards. The message should explain to the user what the error is, and what they should do to repair it. The actual issue being described is that the user has not selected a location on the map, so cannot proceed. Clearer text could read: "You have not selected a location for the problem yet. Please do so using the map."





Error Handling & Prevention

The biggest problem with errors throughout the workflow happens within the Problem Details page, and must be urgently fixed.

Inefficient page structure

Page format contributes to error prevention in a big way here. 'Name' and 'Contact Details' are highlighted as required fields at the top of the page, outside of the context of the form itself. This will lead to users skipping this section and not reading it, missing the important information. The fields are also not marked in anyway to indicate that they are required. The standard practice is to mark fields with an asterisk, or if all fields are required state that clearly at the top of the form. This is an easy, achievable fix.

Create Service Request: Problem Details

Your name and contact details are required to deal with your service request

Request Type **Abandoned Vehicle Inspection**
Summary **Car is abandoned on my road, needs to be moved.**

Details

Your Name

Critical Issue



Frequency
Impact



No errors displayed

Within the body of this form, users must fill out details related to themselves and the problem. However, the user can submit the form blank, with nothing filled in. They can also submit with errors in the form. This is still processed, and the user is shown the confirmation screen. It appears as if the request has been successfully created, and there have been no issues at all. This leads to issues for both the system and the user. Users are unaware that their request will not be carried out; it may as well have never been created. For the system, there is a lot of waste. Every request that is created is going somewhere, either within the system to be filed and never looked at, or to a person's desk to see what the problem with the form was. It is very inefficient, and both sides end up losing.

Is Vehicle on Public Property
<Select One>

Vehicle Registration Number, if known

Make of Vehicle

Colour of Vehicle

The form should follow standard best practices for displaying errors: tell the user what is needed, inform them of what should be input, and if there is a mistake highlight it and ensure that it is fixed before the user can proceed.

Users can submit this form with errors, or even completely blank



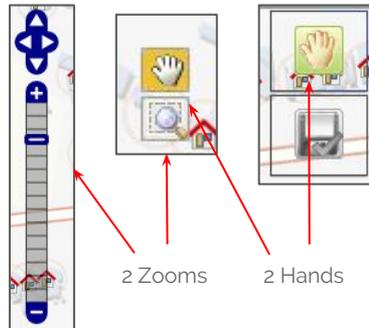
Complex Workflows

The workflow suffers a lot from convoluted and complexity. Users struggle to move through stages, with lots of additional, unnecessary steps slowing users down greatly. The highlight of these issues is the Map selection process, which will present a hard stop for most users. This will be the point of drop off, and many users will simply not finish the workflow because of it.

Simplify each step in the process.

On the 'Identify Problem' page, the map functionality presents critical problems at each stage of interaction. It must be immediately clear to the user what they should be doing. The language of the instructions mean it is not. There should be one clear instruction: 'Click on the map where the problem is.'

The screen is also not intuitive, Buttons swap when clicked, making it difficult to find where things are. Users can only zoom and pan at certain times making navigation extremely difficult. There are 2 hand icons, and 2 zoom widgets on the map, making completion of this step incredibly difficult.



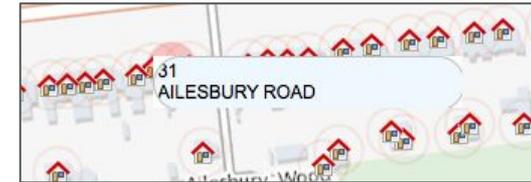
Critical Issue



Frequency 
Impact 

Icons are not labelled, meaning it is not obvious what their purpose is. Once a user has selected the location, the save icon becomes available; but the difference in color is so minimal it presents another challenge for the user to work out what they are supposed to do. Users will not know how to move through the workflow, and will end up dropping off.

Each house on the map is shown, and can be highlighted individually. This does not offer any benefit to the user, and only adds visual noise to the page. A simple, minimalist aesthetic leads to easier user comprehension.



No user value is added with this functionality, and it should be removed.

Overall, the entire map should be simplified to one step. The map should open at the user's location; the user can then drag a pin to the location of the vehicle. There is no reason the user needs to have this process of multiple clicks and numerous steps just to locate a point on a map.



Complex Workflows

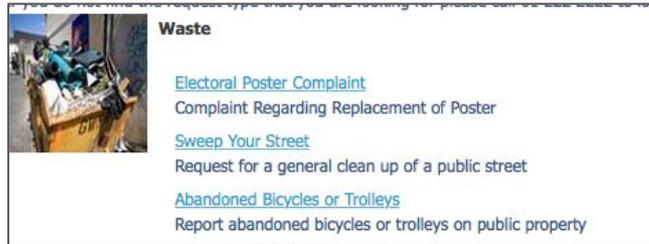
There are many places throughout the entire process that the workflow could be streamlined and simplified. A minimalist design should be the preference where possible; user needs should be the focus, rather than system needs.

Repeated steps

At times, the user is forced to repeat information. They are required to give data that they have already provided earlier in the workflow. During the final form, users are asked to input the 'exact location of the vehicle'. On the previous step, they input the location directly onto a map. It is unnecessary to repeat the input of this information. It tarnishes the experience of the user, and leads to a lack of faith in the system. It makes it seem as if the system is providing a really poor service to the user.

Many screens are superfluous and unneeded.

A number of screens are not necessary for the user; they give work to the user that the system could be doing. This is poor practice, as it overloads the user with things they do not need.

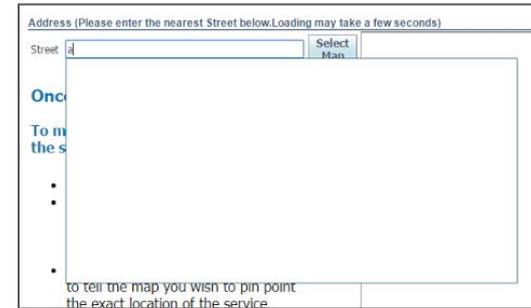


The above page could be simplified into a Search bar, that could suggest the different Requests as the user types.



Lack of system status shown to the user

Informing the user what the system is doing is very important. If this is not done, it will seem as if the system is broken and not doing its job properly. A prime example of this is within the map interaction. At first load, the map stays blank. This looks as if it has bugged, and will not display anything at all. The map should load as an overview of Dublin. This will give the user the added benefit of being able to both search for a street, and to find a street using the map controls.



This happens again when the user types into the Street search box above. The box looks broken, but in reality it is just loading really slowly. A very simple fix, the box should display 'Loading...' when the system is doing so.



Conclusion

There are a number of important takeaways from the heuristic review. There are some positive elements in the workflow. These elements should be examined, and the constructive things can be applied on a wider scale to the entire workflow. Some of these elements need to be reconsidered, but their core idea is strong, and should be developed further.

A range of issues were uncovered during the review process, from moderate issues that cause irritation and frustration for users, to critical issues that prevent users from completing the task. Issues should be fixed based on the rating that they have been given.

Moderate issues, Help and User Input & Control, should be fixed soon. The help, which alone is an issue, becomes more important based on the fact that a complex and convoluted workflow means users are more likely to need help. A rework of how the help functions is needed, moving help to appropriate locations under help icons.

Major issues, like Language, should be fixed urgently. System language that is used throughout is confusing, causing user misunderstanding and making complicated tasks out of simple interactions. Language should follow the pattern laid out in the main menu: simple, task focused language that leads to easy user comprehension and discovery.

Finally, critical issues must be fixed now. Error handling is damaging user experience so badly that at times it becomes impossible to complete the workflow. Alert box errors should be moved inside the context of forms, allowing users to see where errors are and correct them with ease. All forms must display errors; no user should be able to complete a form without entering required fields.

A reevaluation of the workflow as a whole will allow many steps to be condensed, and some removed. Each step should be clear and concise, with an obvious path to completion. The user should never have to do work for the system.



Prototype: Design

We employed the 'report a vehicle' use case to develop a user persona and scenarios to evaluate the Dublin City Council website. We took the findings and conclusions from the heuristic evaluation to guide the build of the prototype. We also used the findings to update our small user model, to help keep a focus within our design. The updated user model contains three new contexts that extend the use case as seen on the right hand side of the page.

Design and Solution

The prototype demonstrates a very task focused application that's core goal is to be simple, intuitive and user friendly. For the home and landing page of the app we took influence from the 'I want to' menu of the Dublin City Council website, which we noted as a positive feature but badly implemented. In our prototype the user can very quickly determine which option is meant for them. For this use case, the user will choose 'Report it'. We looked at the core idea of reporting something and split that task up into 3 steps; **What**, **Where** and **Who**. We used these steps to split the report process into bite sized chunks, so users would not feel lost or overwhelmed at any stage. A progress bar shows the user exactly where they are in the process and how far they have to go. This was something we noted as seriously lacking throughout the council website.

Step 1: What

This section addresses what the user wants to report, and a description of that thing. The prototype gives the user an open text field where they can type in what he or she wants to report. The text field will respond with options corresponding to what the user has typed. This makes it easy for the user to quickly state what it is they are reporting, which in this case is a car. The user can then choose how they would like to describe the vehicle they are reporting, either with its registration plate, a photo of the vehicle or a short description. These options give the user freedom and also address the different contexts the app might be used in, by solving the unique problems each context induces.



User Persona

Name: John Doe

Scenario

John Doe has noticed a car he thinks is abandoned near to where he lives. The car is outside number 16 Bride Street in Dublin city. John want to to notify Dublin City Council about the car. He has the registration number of the car and he has downloaded the new Dublin City Council app on his iphone 6.

Context 1: John doesn't have the registration but has a photo of the car on his phone.

Context 2: John doesn't have have the registration or a photo on his device but he is out and near the car so takes a photo of the car within the app.

Context 3 :John doesn't have a registration or photo and he is at home. But he remembers what the car looks like and wants to add a description of it.

Step 2: Where

This section allows the user show exactly where the car is, in very simple steps. The map will use locationing to determine where the user is and move the map to that location. The user can then move the map so that the pin points to where the car is. Alternatively, users can search the location of the car in the search box. This section solves the overly complicated map that is on the council website, by giving the users more freedom, simpler interface and a much more logical process to showing where the vehicle is located.

Step 3: Who

This part we stripped back to a have a very simple form that only has two text fields; Name and Contact Number. We made both compulsory and the Submit Form button will not highlight until both have been filled in. This section is important for Dublin City Council, but is of least importance to the user. We kept it as simple and quick to complete as possible.

Summary And Confirmation.

The summary section allows the user review what they have entered throughout the report and go back and make changes if they need to. This was not offered on the council website and was much needed. When the report is submitted the user gets a confirmation screen displaying they have successfully submitted a report. This was also lacking from the council website.



Prototype: Walkthrough

Springboard Screen

Click on Dublin City Icon on bottom right of screen

Home Screen

Click on 'Report it' button on bottom left of screen.

Step 1: What

Click on text box, then click on the keypad to type. Choose first option from list. Click one of 3 options; Photo, Write Description or Type Registration Number.

- **Reg Plate Option:** click text box, then click on keypad to type, then click done on bottom right of screen. Then click next step
- **Photo Option:** click 'Take a photo' from menu, click camera button, click next step. Or click 'Choose from library', click photo bottom left then click done top right. Then click next step.
- **Write Option:** click keypad to type then click ok, then click next step.

Step 2: Where

Click either search box or the pin to use the map.

- **Search Option:** click search box, click keypad to type then choose 16 new bride street from list. Then click next step.
- **Pin Option:** click pin and then click next step.

Step 3: Who

Click on text box, then click on the keypad to type. Then click 'Done' or Arrow Right. Click keypad to type, then click 'Done'. Click Next Step.

Summary Screen

Check Info on screen, click submit.

Confirmation Screen

Click button to return to home screen

Prototype Limitations

Back Button

The back button that appears on the top left of the prototype has been left unclickable on most screens. The prototype software is limited in what it can do, so in the real application the back button would retain the user's state, and allow the user to go back and check things at each stage of the workflow. This would not delete what they have inputted when they return.

User Account Button

This has been left unclickable also and would be implemented in the actual application. The user account would allow the user to view previous reports and it would auto fill parts of a new report for them speeding up the process.

Additional Screens

There are some additional screens in the prototype that do not appear in the walkthrough, again due to limitations in the software. At the end of the workflow, there are a set of summary screens that show if the user had entered a photo or description of the vehicle. It is not possible to tell which one the user has entered, so the prototype will always show the Reg Plate option on the summary screen. Other options are included for demonstrative purposes.

Interactions

Within the map section, the user would interact differently with the pin. The user would drag the map to bring the pin to the correct location, in the same way Hailo works.