Practical Android - Blog

Week 3 23rd – 29th September

Our task for this week's practical was to come up with 10 ideas for an app with the subject of sustainability. The ideas we came up with through brainstorming were:

1. Car emissions calculator w/ alternate route advisor for environmental sustainability
2. College accommodation listings by students for students for human sustainability
3. UCD locator for water fountains, bins, bicycle racks around campus for environmental sustainability
4. Locating electric chargers nearby for environmental sustainability
5. Savings planner for future expenses such as holidays, wedding, college for human sustainability
6. Machine Learning litter classification app/game for environmental sustainability

Coming up with a novel app idea within this subject proved to be difficult. With the specification given for the app we had to ensure our app had the following properties:

- It must make use of a range of user interface elements, e.g. lists, buttons, spinners, dialogues, an action bar, Toasts etc
- It must include at least seven distinct screens (Activities).
- It must store data locally, e.g. in an SQLite database. Remote storage is also allowed, but some element of local storage is a requirement.
- It must make good use of at least two sensors (e.g. motion or location sensors) or web services (e.g. the Open Weather Map http://openweathermap.org/).
- It should connect with and use at least one Activity from another Android application, e.g. email, maps, contacts

Using the 10 plus 10 method discussed in class, we further developed each idea but soon realised many did not adequately address any real problems (ideas 1, 3, 4) or did not meet the specifications outlined above (5). Idea 2 seemed interesting and doable but not everyone liked it. We knew we needed to pay attention to the number of sensors we would use, so we decided as a group to spend the weekend further researching the ideas and coming up with ways to meet the project specification.

Week 4 30th September – 6th October

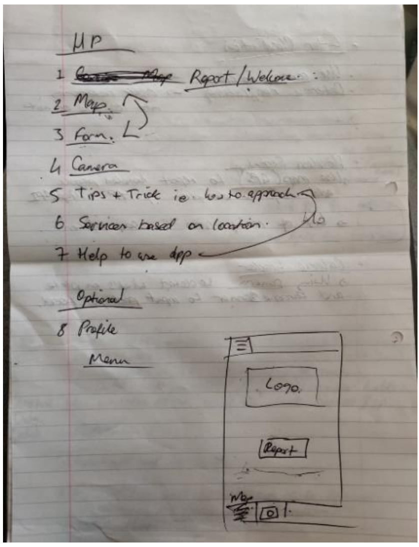
As we all narrowed down the feasibility of each idea, Kamil has an interesting idea based around the current homeless problem in Ireland. The idea would be to allow users to report any rough sleepers. We collectively still liked some of the previous ideas, particularly the litter classification app, so we created a final list of ideas:

1. Rough Sleepers
2. Litter Classification
3. Calorie Tracker/ Diet Planner

We met up as a team this week to further discuss and narrow down our selection. To help decide we drew up a list of pros and cons for each app. By referencing the specification and our list of pros and cons, we agreed that the Rough Sleeper app idea stood out most as the Litter Classification app would be difficult to implement with an AI driven classifier and the Calorie Tracker didn't provide us a means to create a novel app that would be scored highly.

Next, we came up with some of the inner workings of the Rough Sleeper app. The location sensor would be used in conjunction with Google's MAP API (web service) to select your location. The camera sensor would be used in an option to take a picture of the rough sleeper. This covered the sensor requirement of the specification so next we would need to clarify the screens we would use.

Week 5 7th – 13th October



HP

1 ~~Camera Map~~ Report / Welcome

2 Map

3 Form.

4 Camera

5 Tips + Trick ie. how to approach

6 Services based on location.

7 Help to use app

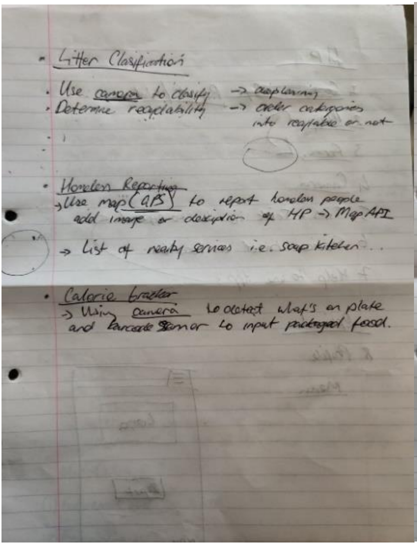
Optional

8 Profile

Menu

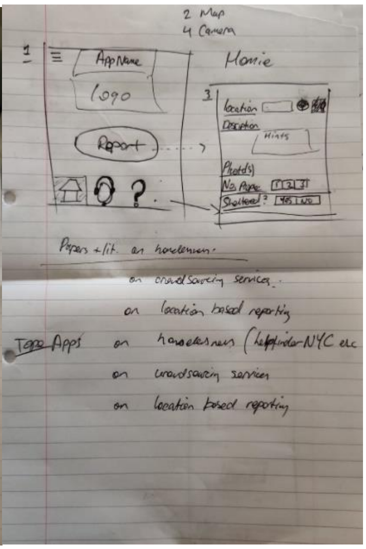
Logo.
Report
Map

Figure 1 Proposed layout of screens



- Litter Classification
  - Use camera to classify → deep learning
  - Determine recyclability → order categories into recyclable or not
- Homeless Reporting
  - → Use map (GPS) to report homeless people add image or description of HP → Map API
  - → list of nearby services i.e. soup kitchen . .
- Calorie tracker
  - → Using camera to detect what's on plate and barcode scanner to input packaged food.

Figure 2 Final 3 app ideas



2 Map

4 Camera

Papers + lit. on homelessness.

on crowdsourcing services.

on location based reporting

~~Tops~~ Apps on homelessness (helpfinder-NYC etc

on crowdsourcing services

on location based reporting

Figure 3 Interface of Rough Sleeper

We had to propose our app this week, so after some further discussion we agreed on selecting the Rough Sleeper app as our idea. We brainstormed to form the basis for the screens we would use (Figure 1) allowing us to address the requirements in the specification. The sensors or web services part was covered using Google's Map API, device location and camera. To store data locally we would use a database to store the reports made by a user. In addition, I had started researching homeless services available in Ireland so mentioned I could compile a list of the into a database that could be read from in the contacts screen.

With regards the specified requirements, we were unsure of what connection we would use to another Android application but as we would have a contact us screen, this could have the option to call or email the app developers (us) through either an email application or the phone app. The final requirements of the project (range of user interface elements & 7 or more distinct screens) would be easily met.

Week 6 14th – 20th October

This week involved creating and submitting the prototype (Figures 4, 5 & 6). The details of each are as follows:

Homepage: Display some details of the app with a report button.

Report: Allows the reporting of rough sleepers with options for images, location on map or by GPS etc.

Location: If the user selects the map button this screen uses Google's Map API to display an interactive map to set location.

Contact Services: Display a list of homeless service contact details with a drop-down list for all counties.

Tips: Displays a list of tips for approaching rough sleepers.

Side Menu: This would allow easy navigation between different screens of the app.

Statistics: Displays some homelessness statistics.

FAQ: A list of frequently asked questions to aid in using the app.

Privacy: Displays the privacy conditions that the app adheres to.

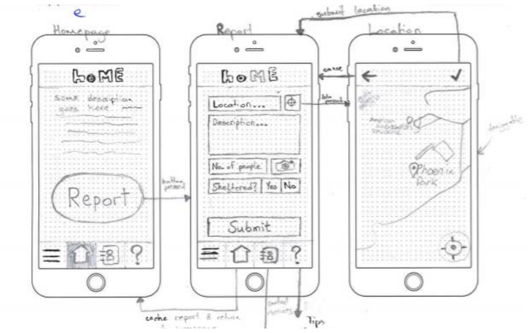


Figure 4 Homepage, Report and Location screens

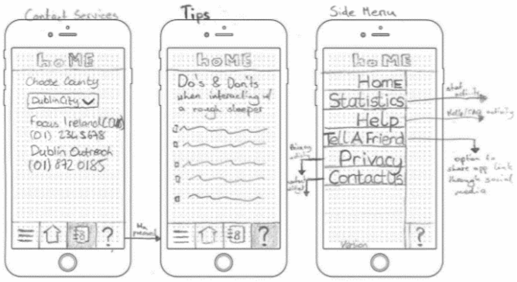


Figure 5 Contact Services, Tips and Side Menu screens

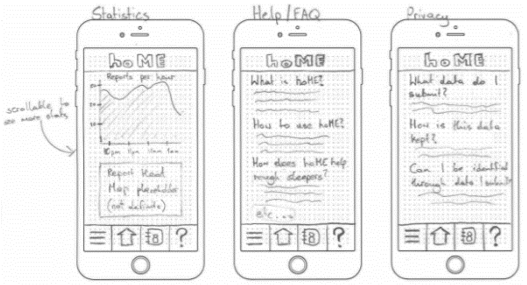
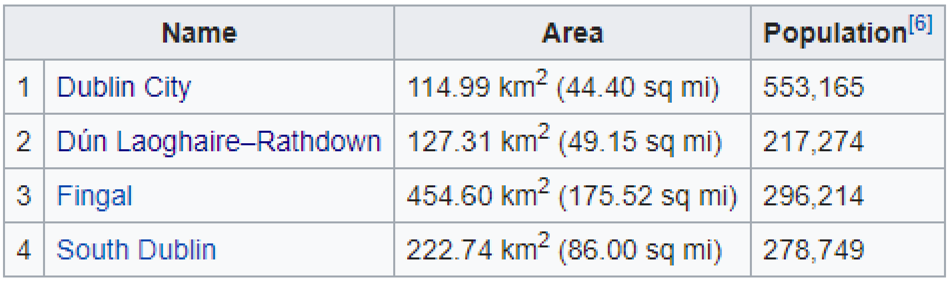

Statistics
Help / FAQ
Privacy
hoME
Reports per hour
scrollable to see more stats
Report Heat Map placeholder (not definite)
What is hoME?
How to use hoME?
How does hoME help rough sleepers?
etc...
What data do I submit?
How is this data kept?
Can I be identified through data I submit?

Figure 6 Statistics, FAQ and Privacy

Kamil started work on the structure of the project and committed some skeleton code for the home fragment. I cloned the repository locally and we designated some work between us each taking some fragments to start on. I had begun researching some of the homeless services operating in Ireland so chose to take on the contacts fragment.

The idea of this fragment is to display a list of the services available that can aid any homeless people or rough sleepers. As each county has a different list of available services, I chose to display the services by county with the plan to allow the user to select their county from a drop-down list.

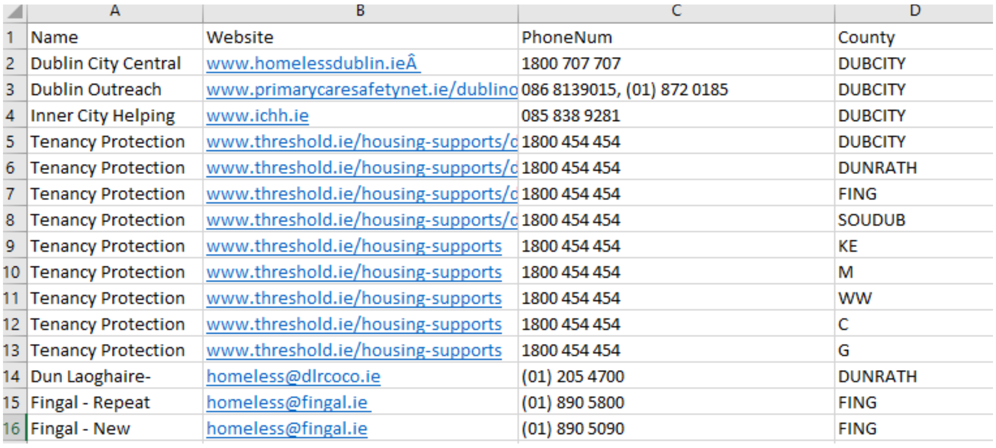
To build a list of these services I searched the internet for the main homeless services (Focus Ireland, Threshold, The Peter McVerry Trust, Simon etc.) and built a table in Excel with the service name, email address or website, phone number and county. As many of these charities have nationwide numbers this allowed me to build a decent base of services that covered most counties. For speed I entered the county using the license plate code of each county e.g. KE for Kildare but decided to split Dublin by its 4 administrative counties.



	Name	Area	Population[6]
1	Dublin City	114.99 km^2 (44.40 sq mi)	553,165
2	Dún Laoghaire–Rathdown	127.31 km^2 (49.15 sq mi)	217,274
3	Fingal	454.60 km^2 (175.52 sq mi)	296,214
4	South Dublin	222.74 km^2 (86.00 sq mi)	278,749

Figure 7 The 4 administrative counties of Dublin. [1]

I assigned each a code (Dublin City – DUBCITY etc.) and continued populating the table shown below with any services I could find.



	A	B	C	D
1	Name	Website	PhoneNum	County
2	Dublin City Central	www.homelessdublin.ieÂ	1800 707 707	DUBCITY
3	Dublin Outreach	www.primarycaresafetynet.ie/dublino	086 8139015, (01) 872 0185	DUBCITY
4	Inner City Helping	www.ichh.ie	085 838 9281	DUBCITY
5	Tenancy Protection	www.threshold.ie/housing-supports/c	1800 454 454	DUBCITY
6	Tenancy Protection	www.threshold.ie/housing-supports/c	1800 454 454	DUNRATH
7	Tenancy Protection	www.threshold.ie/housing-supports/c	1800 454 454	FING
8	Tenancy Protection	www.threshold.ie/housing-supports/c	1800 454 454	SOUDUB
9	Tenancy Protection	www.threshold.ie/housing-supports	1800 454 454	KE
10	Tenancy Protection	www.threshold.ie/housing-supports	1800 454 454	M
11	Tenancy Protection	www.threshold.ie/housing-supports	1800 454 454	WW
12	Tenancy Protection	www.threshold.ie/housing-supports	1800 454 454	C
13	Tenancy Protection	www.threshold.ie/housing-supports	1800 454 454	G
14	Dun Laoghaire-	homeless@dlrcoco.ie	(01) 205 4700	DUNRATH
15	Fingal - Repeat	homeless@fingal.ie	(01) 890 5800	FING
16	Fingal - New	homeless@fingal.ie	(01) 890 5090	FING

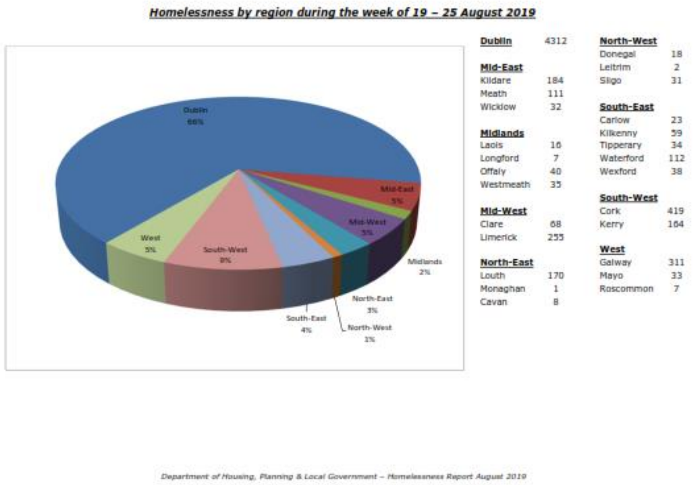
Figure 8 shows the table I created to store the services

[1] https://en.wikipedia.org/wiki/County_Dublin

Week 7 21st – 27th October

To find all the available homeless services, I extended the search to look for services county by county. For each county, I searched their respective county council website as this is the advised first point of contact for anyone suffering from homelessness. I supplemented the results with any other service I could find for that county and included any contact details from homeless charities that covered that area (e.g. add Mid-West Simon details to Clare, Tipperary & Limerick).

Alongside my continued search for services available to the homeless, I decided to search for some accurate figures pertaining to the number of homeless in Ireland. All the figures found on charity websites use the Department of Planning, Housing and Local Government as a source, so I navigated there[2] and found documents relating to the number of homeless in Ireland. The figures within these documents described the homeless by county so I downloaded the latest 3 months to use. I created a new table in Excel and manually input the values for each county from the document shown in Figure 9 into the table. Figure 10 shows the resulting table.

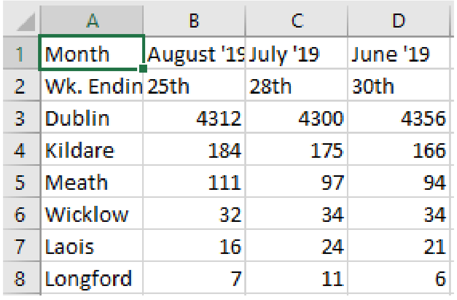


Homelessness by region during the week of 19 – 25 August 2019

Dublin	4312
Mid-East	
Kildare	184
Meath	111
Wicklow	32
Midlands	
Laois	16
Longford	7
Offaly	40
Westmeath	35
Mid-West	
Clare	68
Limerick	255
North-East	
Louth	170
Monaghan	1
Cavan	8
North-West	
Donegal	18
Leitrim	2
Sligo	31
South-East	
Carlow	23
Kilkenny	59
Tipperary	34
Waterford	112
Wexford	38
South-West	
Cork	419
Kerry	164
West	
Galway	311
Mayo	33
Roscommon	7

Department of Housing, Planning & Local Government – Homelessness Report August 2019

Figure 9 shows the format of the figures in the document from the Department of Planning, Housing and Local Government



	A	B	C	D
1	Month	August '1	July '19	June '19
2	Wk. Endin	25th	28th	30th
3	Dublin	4312	4300	4356
4	Kildare	184	175	166
5	Meath	111	97	94
6	Wicklow	32	34	34
7	Laois	16	24	21
8	Longford	7	11	6

Figure 10 shows the homeless figures table

I was still unsure where I would be using the homelessness figures but as I would be displaying the homeless services by county, I decided the best method to store these would be in a database. This would give us the ability to release an updated collection of services without having to change any code by simply updating the database. As this database would be stored locally it also covered the aspect of the project requiring local storage. I began researching database use in Android and the different ways to read in data. Although .CSV is often used, a database would give me the flexibility of using a relation database management system (RDBMS) such as SQLite.

I pulled the latest version of the project from our GitHub repository and started planning out my approach for the contacts fragment.

[2] https://www.housing.gov.ie/housing/homelessness/homeless-report-august-2019

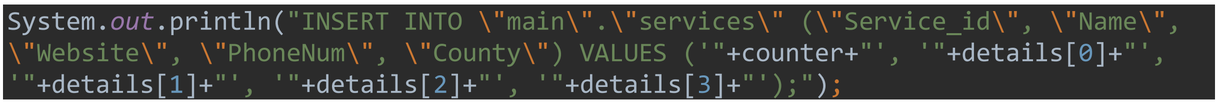
Week 8 28th October – 3rd November

I used DB Browser for SQLite to create a database with the following tables:

services – this table would contain the services from the table I had built shown in Figure 2.

county – as the county name for each service was stored as a code, I would use this table to get the full name e.g. Wicklow from WW.

To populate the tables from the Excel tables I had built, I decided to create a simple java application that would take the service details located in the .CSV and print out a list of INSERT SQL commands for the services. This involved exporting the table as a .CSV, reading that file into an ArrayList<String[]> with each String array containing the details of a single service. I then printed out a command for each service using the following statement:

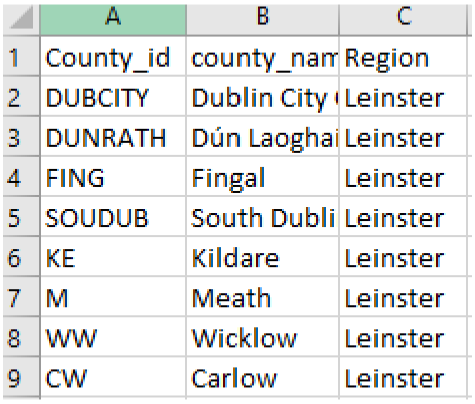


```
System.out.println("INSERT INTO \"main\".\"services\" (\"Service_id\", \"Name\",
\"Website\", \"PhoneNum\", \"County\") VALUES ('"+counter+"', '"+details[0]+"',
'"+details[1]+"', '"+details[2]+"', '"+details[3]+"');");
```

This printed statements in the format required for SQL Insertions:

INSERT INTO "main"."services" ("Service_id", "Name", "Website", "PhoneNum", "County") VALUES ('12', 'Tenancy Protection Service', 'www.threshold.ie/housing-supports', '1800 454 454', 'G');

which could be copied into DB Browser and executed to populate the tables. This process was repeated for the table containing the county details (Figure 11). The resulting database was then converted to SQLite in Command Prompt via the sqlite command "sqlite destination.db < source.sql". I created an "asset" folder in the android project and placed the newly converted database in a subfolder called "database".



	A	B	C
1	County_id	county_nam	Region
2	DUBCITY	Dublin City (	Leinster
3	DUNRATH	Dún Laoghai	Leinster
4	FING	Fingal	Leinster
5	SOUDUB	South Dubli	Leinster
6	KE	Kildare	Leinster
7	M	Meath	Leinster
8	WW	Wicklow	Leinster
9	CW	Carlow	Leinster

Figure 11 shows the table used to convert county codes to county names.

Part of the work for this week involved researching any relevant literature along with looking at any existing apps. I researched the Youth Street Connect (YTH) app, which is an app for the homeless and unstably housed youths. This is an interesting app that uses smartphone technology to provide a range of services straight to the homeless community offering service such as an interactive map, emergency hotlines, weekly health tip texts and access to sexual health information. Some aspects of this app may intersect in what we are trying to do so I plan to research the effectiveness of it.

The remainder of this week was centred around reading from the database by using a Helper class that extends the SQLiteAssetHelper class. This made forming and executing SQL commands easy on the database. I created methods to test that reading from the database was functioning as expected. I amended certain aspects of the database such as the format of the email and websites to create a more visually pleasant list of services.

Week 9 4th – 10th November

I further developed the contacts fragment this week to display the homeless services details contained in the database. To select the county in which to display services, I added a drop-down list (Spinner class in Android) and added a string array of its contents (the list of counties) to strings.xml. Selecting an item in this list first sends a query to the database to get the county code from the name e.g. selecting Carlow will return CW from this query. To do this, the county table is queried for the county code then an INNER JOIN is performed on the *services* table and the *county* table:

"services INNER JOIN county ON (services.County = county.County_id)".

The resulting table is returned containing the respective county name for each service.

I created a TextView in the contacts fragment to display the list of services for the selected county. To allow the phone number & email/website to be clickable I experimented with the autoLink feature (android:autoLink=”all”). This feature automatically creates clickable links from phone numbers, email addresses and websites but as it does not offer the ability to change the region that clicks the link, the links were too easily clicked when other regions of the screen were touched as in scrolling. I further researched different methods of making clickable links but each of these had their own problems.

I informed my teammates that I would also take the statistics fragment. I spent most of the weekend planning my approach for this fragment while exploring other options for displaying clickable links in the contacts fragment.

The app I was researching, Street Connect by YTH, is not available in the Play Store so it became difficult to find any reviews or anecdotal evidence of its efficacy. From screenshots I found online, it seemed rather rudimentary, but this alone offered some advice. Don’t over complicate it! Important to us was the user experience; this app may be used in times of stress and in a hurry so the UI would need to be extremely clear and easy to use.

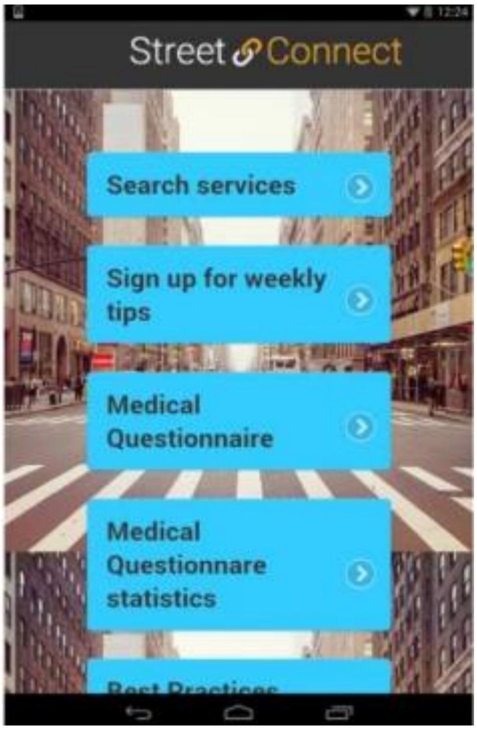


Figure 12 the homepage of YTH StreetConnect app

Week 10 11th – 17th November

Through discussion with my teammate Kamil, I decided to create a custom adapter for the contacts fragment. This would allow me to create a ListView containing rows where each row was comprised of a ViewHolder with 4 TextViews, one for each of the attributes for a service: service name, service number and service email/website and service county. By displaying the services this way I could customise each attribute and make any TextView clickable for its respective link using the Linkify class.

My plan for the statistics fragment is to create a clickable map of Ireland in which clicking a county displays the homeless figures for that county, pull from the figures table in the stored database. I found a map of Ireland that had each county coloured in shades of green. I used GIMP to manually convert this image to a vector by deleting any blue coloured parts representing the ocean.

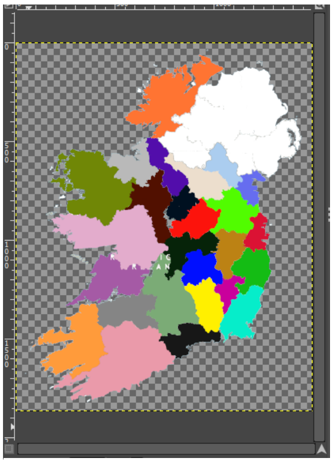


Figure 13 shows the colouring done using GIMP

With this vector, I created a new coloured map of Ireland in which each county would have a different shade. This would make it possible to track the user's touch by comparing the pixel value of the selected pixel with a stored table of all county's pixel value. This process turned out to be quite tedious in GIMP as many of the county borders had to be manually outlined then coloured. I created a table in the local database called *county_colours* that contains the respective hexadecimal value for each county.

Using the pixel location, in terms of x and y coordinates, to get the correct colour value of a selected pixel proved to be difficult. The offset for the status bar, navigation bar and any padding had to be considered. I added an option to calculate any offset in the x and y directions and amended the pixel value accordingly.

The final part of my work this week involved adjusting some of the colours used in the contact services screen to line up with the other aspects of the app.

Week 11 18th – 24th November

With numerous other deadlines looming, the plan for this week is to test our app. For me this involved checking the contact services were all displaying correctly for each county. In doing this I noticed there were a couple of counties that were lacking in any homeless services so I spent some time searching the internet for any missing agencies I could include.

I added an option to the contact services so that the location could be used to select the item in the drop-down list. As the location is requested and possibly set in the report screen, I created a global variable (MyApplication) to store the longitude and latitude values. I further added a class called AddressManager to extract the county name from the global variable. This added the difficulty of displaying the services for each of the administrative counties in Dublin, so I opted to show all of Dublin services for each.

I needed to add some clauses for when the user selected a pixel with a value not recorded in the database. I coloured any unused regions (Northern Ireland, the Atlantic, Lough Neagh etc.) white and added a check that the Cursor size be greater than 0. This would fix any NullPointerException from any oddly coloured pixels and default to the last county selected.

Week 12 25th – 29th November

With tidying up and testing the plan for this week, we each performed tests on the screens created by the other teammates. In doing this, Kamil discovered that the statistics screen I had designed was not producing the required results. The offsets I was using to find the colour of the selected pixel was incorrect for his phone. Together we worked on trying to fix this and could find fixes for one of our phones but never one that worked on both. The bigger screen in my phone, a OnePlus 7 Pro, did not scale the image used (map of Ireland with coloured counties) but did Kamil's. This problem persisted throughout the week and we are still working on a fix today, Friday 29th November.

Final Entry – My Contribution

The core of my work on this project was centred around the Contact Services and Statistics screens. The work involved in developing these screens along with any additional work is outlined below:

- Searched for and compiled a list of homeless service available in every county in Ireland.
- Searched for and compiled a list of figures for homelessness in Ireland.
- Created a database of the compiled services with columns for each attribute.
- Added the homelessness figures to the database.
- Created a database helper class to query the existing database.
- Researched YTH StreetConnect app.
- Designed and populated the Contact Services screen with the data stored in the database.
- Designed the statistics screen to display an interactive map of Ireland.
- Created a coloured map of Ireland for differentiating the selected county in Statistics.
- Added functionality to Statistics screen to get the selected county and display statistics.
- Added content to the report.