Text, letter

Description automatically generated

Users like Douglas are looking to learn more about homelessness, while having limited experience with technology. He is initially met with a large graphic and a prompt to give his understanding of homelessness. This engagement satisfies Douglas desire to learn more. Answering this or clicking on the surrounding box leads to the big picture page where he can read more interesting facts.

Douglas, having also recently experienced homelessness himself, wants to learn more about homelessness in his own community and is given the option to put in his postcode or local government to see some shallow but related statistics. As Douglas does not have a lot of experience with technology, the feature needs to be as simple as possible. The textbox and arrow following a text prompt or question come together to create a match between System and the Real World, a Neilson usability heuristic. As arrows refer to a forward concept that most people commonly understand, including Lisa and Dr. Lily, all of who can intuit that they should proceed with one of these actions. One of *StreetSmart Australia*’s strength was their subtle call-to-action at all points of the website, which is replicated in this feature.

Text, whiteboard

Description automatically generated

Once the users choose their specifications for their search, they are met with a box of information that has their requirements.

Douglas, having browsed into his own Local Government Area (LGA) from the front page, and after having chosen his filters, is able to see his LGA and corresponding state in red, distinct from the others. Having it change according to the selected option provides Visibility of System Status, where the user receives feedback

Visibility of System Status

Each feature:

* Description of what it allows the user to accomplish / how UI should be used
* Explanation of quality (via Neilson's)
* Satisfy at least one of your personas

Lisa (Figure 1, Trubea J 2019) is a university student. She is studying social sciences and are requiring data on homelessness for a project. She is competent in navigating websites and has some experience in researching for projects. Her goal is to extract quality data from the website to use in their project. This function needs to be accessible to her. Lisa is more likely to comprehend the data if it is in a visual format. When she needs to use the data though, she may need it in a format that allows her to work with it.

Dr. Lily (Figure 2, Chavez H 2019) has dedicated her life to understanding and educating social issues and has spent the last decade running a nonprofit dedicated to women who have been adversely affected by domestic violence and specifically those who no longer lives in insecure housing. She has been involved with topics such as mental health and homelessness as well. Dr. Lily is always open to new ways of doing things and needs to be aware of the discussions that people are having about these topics outside of the work that she does. Her goal is to look online for these discussions and find out what is going on, and given her expertise, to help where she can. She is not proficient at newer technology and software, and frequently takes some time to learn to navigate.

Douglas (Figure 3, Fout D 2018) is a 26-year-old who suffered from unemployment after high school. Without the support of his parents, he found himself not having a place to sleep for a brief period, and then sleeping in overcrowded dwellings, before reaching out to friends and asking to sleep on their couch. When back on his feet, he is reminded of the kindness and understanding shown to him during the hardest time in his life, and now needs to repay this

kindness and pay it forward. His goal is to learn more about homelessness and wants to provide emotional support to those who need it. Douglas has not developed technical skills and finds navigating websites frustrating.

**Neilson's Usability Heuristics**

* 1. Visibility of System Status - "what is going on?"
     1. The system should always keep the user informed of what is going on through appropriate feedback in reasonable time (e.g. Lamp)
     2. Always make sure your system is communicating clearly and thoroughly with the user
  2. Match between System and the Real World - "what are you talking about?"
     1. Speak the user's language, with words and concepts that are familiar to them (e.g. Children's book)
     2. Always make sure your user can understand what your system is communicating to them
  3. User Control and Freedom - "oops"
     1. Users make mistakes, and need an emergency exit to get back to where they want to be as fast and as simply as possible (e.g. Back button)
     2. Always make sure that your system has a simple and efficient emergency exit to "fix" user error
  4. Consistency and Standards - "I know that!"
     1. Always make sure that your system has continuity across your platform (e.g. Bathroom men and women)
  5. Error Prevention - "Glad I didn't do that"
     1. The best designs don't only have great error recovery, but prevents users from making those errors (e.g. Car dashboard)
     2. Always make sure that your system has efficient error prevention
  6. Recognition or Recall - "wait, what was that already?"
     1. Systems should minimise the user's memory load by making objects, actions and options more visible (e.g. Google smart search)
     2. Always make sure that your system prioritises recognition over recall whenever appropriate
  7. Flexibility and Efficiency of Use - "it doesn't matter if you're an expert or newbie"
     1. System should incorporate accelerators, which are unseen to the novice user, but that allow the expert user to navigate faster with frequency actions (e.g. Photoshop key shortcuts)
     2. Always make sure that your system incorporates accelerators to facilitate expert users without affecting novice users
  8. Aesthetic and Minimalist Design - 'wow, and woah!"
     1. Dialogues should not contain irrelevant information and that any extra unit of information competes with the already present relevant units of information, and therefore diminishes visibility
     2. A design needs to b minimal and as efficiency as possible, and aesthetically beautiful (e.g. Apple, oh its beautiful and woah it works so well)
     3. Always make sure your system is aesthetically pleasant and efficiently composed
  9. Help Users Recognise, Diagnose and Recovery From Errors - "my bad"
     1. Error messages should be expressed in plain languages and precisely indicate the problem
     2. Always make sure your system indicates errors and messages in plain language
  10. Help & Documentation - "now I get it!"
      1. Even though the system can be used without documentation, it might be necessary to provide it (e.g. Ikea)
      2. Always make sure your system provides documentation to help the user