



TERMINAL APP: BINDER

(Julia Ryan, Brett Rapp, Chen (Josh) Zhao, Nick Sonner)

PROBLEM

A local community are seeking an efficient way to dispose of their excess garbage by utilising the total bin space within the community more efficiently.

SOLUTION

A social network inspired terminal app allowing users to share and/or use unused bin space with other residents in their street.

The app features the ability to create an account, corresponding to the user's house number and, by default, issues them with a general waste, recycling and green waste bin. The user can find the total available bin space of their neighbours bins and send an in-app request, using (currently utilising twilio) to ask a specific neighbour if they can use their bin space.

The specific bins for each house can have rubbish added to it by the owner, and is automatically emptied on it's corresponding collection day. The ability to view upcoming collection dates, could aid with rubbish management and organisation of future events requiring extra bin space.

DEVELOPMENT PLAN

Day 1

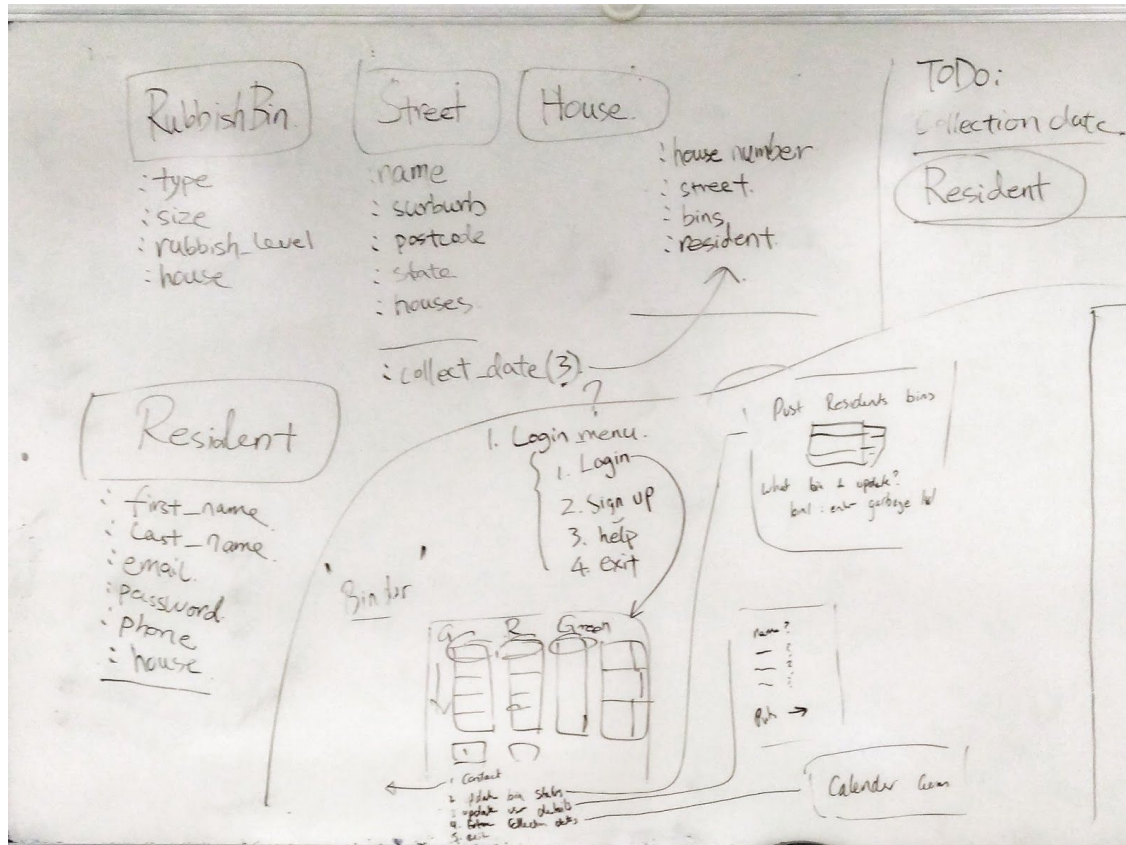
Defining the problem and solution including researching tools that could contribute to the functionality for the proposed solution.

Day 2

Planning and wireframing. Defined the methods and variables needed. Walked through the user's journey, questioning our design principles, refining the overall project.

Day 3

- Begin coding menus and methods for displaying data in tables
- Testing and implementation of app "elements"



Gems

1. `gem catpix`
2. `gem colorize`
3. `gem tty-prompt`
4. `gem twilio-ruby (and api)`
5. `gem rubyGems`

Development stages

1. Build foundation for app
2. Implement interactivity
3. Include tables to organise data
4. Implement Twilio integration
5. Implement ascii for houses (future feature)
6. Implement Twitter



UX

Welcome page

Includes:

- Binder banner using *gem catpix*
- Interactive menu with below options:
 - Sign in
 - Sign up
 - Help
 - Exit

Sign in

Requests user to enter email and password

Main menu page

Displays the next collection day for each bin, a table displaying the current available bin space of the user, and a second table with the available bin space of the neighbours bins

Menu options

1. Contact neighbour
2. Update my bin level
3. Update my contact details
4. Upcoming collection days
5. Remove me from Binder
6. Log off

1. Contact neighbour

Displays the table of neighbours bins and bin levels. The user is presented with a menu containing a list of the neighbours. If a contact is selected, their contact details (phone and email) are displayed, and an sms will be sent to the recipient asking if they have free bin space spare.

Menu options

1. neighbour
2. neighbour
3. Back to main menu



2. Update my bin level

Displays table with information about the users bins, below, displays a menu with options to update the bin types level.

Menu options

1. General
2. Recyclables
3. Green
4. Back to main menu

4. Upcoming collection days

Display calendar with all future bin collection days for next 30 days?

Menu options

1. Back to main menu

5. Remove me from Binder

Selecting this will automatically remove the house (and by extension account) then kicks the user back to the welcome menu

6. Log off

Logs the user out and returns to the welcome menu

Sign up

A series of questions, prompting the user for input to update their account information:

- Name
- Contact number
- Email
- Password
- House number

The street at present is defaulted to Victoria St, future implementation would allow users to join other pre-existing streets.



Help

Presently not implemented, if the user selects this option, it will display “sorry, this feature is not yet available” and returns the user to the welcome menu. In the future this will provide detailed information on the app.

Exit

Exits the program, returning the user to the terminal.

Classes

Class Street

Attributes

- * name
- * suburb
- * postcode
- * state
- * houses = []

Methods

welcome_message
add_house
remove_house
find_house
available_bin_space
available_recycle_space
available_general_space
available_green_space

Class House

Attributes

- * street = street object
- * house number
- * first name
- * last name
- * email
- * password



```
* phone
* bins = []
```

Methods

```
add_bin
remove_bin
general_bin_space # too similar to available_general_space
recycle_bin_space
green_bin_space
collection_day
contact_details
bin_status
upcoming_collection_days
```

Class Bin

```
* type
* size
* house
* bin_level
* collection_day
```

Methods

```
fill_bin
empty_bin
```

File Structure

```
main.rb
menu.rb
street.rb
house.rb
bin.rb
contact.rb
binder_logo.jpg
```

Product problems

- *Who updates the bin? The bin owner?*
- *What happens when users fail to update their bin?*
- *We need to add the function of persistence between sessions & clients.*



Please note:

Due to Brett's personal account information being used in the code for the twilio integration, this functionality has been disabled from the submitted code (his credentials have been replace with string values of the corresponding variable.) If anyone requires a demonstration of the code, we are more than happy to demo it again with the missing credentials.

Future Products (bin jokes)

- buber

Have a friendly buber driver pick you up at your location, and drop you off safely at the your bin destination

- Binstagram

A simple, fun & creative way to capture, edit & share photos and videos of bins with friends & family.

- Binbook

Connect with friends, family and other people you know, sharing information on bins.

- Boogle

Search the world's bin information, including webpages, images, videos and more