**NOTE: There is an invisible “[other]” after every question. If the solution is not one of the suggested examples, please provide the alternative clearly.**

**<FILTERING> Which (*mechanically-speaking*) is most similar to how users will toggle filters: radio buttons or tick boxes?**

**Example 1:** [radio buttons]

**Dogs Allowed:** Yes ( ) No ( ) No Preference (O)

i.e.

Only one allowed at a time. Default of No Preference. Three possible states.

**Example 2:** [tick boxes]

**Dogs Allowed:** Yes [ ] No [ ]

i.e.

|  |  |  |
| --- | --- | --- |
| **“Yes” state** | **“No” state** | **Result** |
| False | False | Display all |
| True | False | Display dog-friendly only |
| False | True | Display no-dogs only |
| True | True | Display all |

**<CRAWL GENERATION (GENERAL)> After browsing the pubs, and selecting to generate a crawl, are the filters locked-in?**

**Example 1:** [yes]

1) User is at the full list of pubs

2) They do or do not apply some filters to browse the selection

3) They press a button to go to a new screen geared for crawl generation

4) They are prompted whether they wish to retain their current filters, or reset to the full list

5) The pool of available pubs is restricted to whichever choice they selected (for the duration of the crawl generation)

**Example 2:** [no]

1) User is at the full list of pubs

2) They do or do not apply some filters to browse the selection

3) They press a button to go to a new screen geared for crawl generation

4) They are prompted whether they wish to retain their current filters, or reset to the full list

5) The initial pool of pubs displayed is based off of their choice

6) The selection of pubs can still be tweaked (filters can continue to be added/removed), and pubs retain their “selected” or “not selected” status regardless of whether they are in the current “search“ results

**<CRAWL GENERATION (USER-CREATED)> What happens when the user enters a time for a pub that might be closed?**

**Example 1:** [not allowed]

Program denies the user from making the choice, potentially with a warning/error message.

**Example 2:** [warned post-selection]

A message alerts the user that the pub may be closed when selected, but does not prevent the selection.

**Example 3:** [warned pre-selection]

Closing times are displayed next to the pubs in bold text, allowing the user to make an informed decision.

**Example 4:** [no warning]

The user is free to enter whatever times they see fit unrestricted and with no notice.

**<CRAWL GENERATION (USER-CREATED)> When is the user prompted to put times against each pub?**

**Example 1:** [as they select them to use in the crawl]

When they select a pub, a pop-up (or similar) forces them to specify a time.

**Example 2:** [after they select all pubs]

The user simply selects which pubs they want to visit first. After confirming their selection, they go to a new screen to enter the times against each one.

**<CRAWL GENERATION (USER-CREATED)> Is the user forced to enter the pub times chronologically?**

**Example 1:** [yes]

If the user attempts to enter a time for a pub that occurs earlier than a previous pub’s slot, they are denied and an error message informs them.

**Example 2:** [no – and the list is generated live]

The system rearranges the pubs into chronological order as times are applied.

**Example 3:** [no – and the list is generated after all times provided]

The system rearranges the pubs into chronological order after all times have been given.

**<BROWSING PUB LISTS> Are empty pub lists displayed?**

**Example 1:** [yes – but they are not selectable]

A pub list containing no pubs is still listed, but they may not be selectable, or produce an error message when selected.

**Example 2:** [yes – and they are selectable]

A pub list containing no pubs is still listed, and can be selected to bring the user to a page containing no pubs.

**Example 3:** [no]

If a pub list containing no pubs is provided with the software, it is not displayed in the list of available pub lists.

**<BROWSING PUB LISTS> What happens if there are no pub lists in the software?**

Example 1: [software quits]

When launching the software, it will identify the lack of pub lists, inform the user, and quit.

Example 2: [software loads – but a message informs the user]

When launching the software, it will display a message to the user if no pub lists have been supplied, but it won’t prevent the user from keeping the app open.

**<GENERAL STRUCTURE> How will information (like pub lists and pubs) be accessed from the mobile software?**

Example 1: [information is supplied manually with the release of the software]

The developer decides what information to provide with the app, and adds/removes pub lists and pubs in advance of the release to the users.

Example 2: [information is accessed from a remote server]

The developer decides what information will be available to those using the app, and adds/removes pub lists and pubs to an accessible server. When the user wishes to access the information, the selection of information is downloaded from the internet.