Service Finder

User Stories & Visuals

Version: Draft

Date: 14th October 2019

Author: VidaVia

Contents

Background 3
Approach 3
Vision Statement 3
Requirements list 4
Proposed Solution 5
Roles 6
User Stories 6
Data Visuals 9
Main components 9
Search method 10
Search results 11
Error & Feedback 12
Expected API Calls 13
Initialisation 13
Type-ahead / Helper 13
Email / Sharing / Feedback 13
Service Search 14
Notes on Session Support 14

Background

The LGA are running a programme of pilots to consider how to collect/maintain local service information then classify it so that it can be found through an understanding of needs and circumstances and finally to assure that everything is completed and up to date before publishing it as open data.

It is expected that applications will consume the open data and support citizens, family, friends, carers, intermediaries, third sector workers and public sector frontline workers connecting people to appropriate services. At this early stage it has been recognised that there is a need for an example application to show the sort of thing possible but also to give the pilots a tangible means of exploring the open data and testing the classification model.

The example app should seek, as a minimum, to display information about services appropriate to needs and circumstance terms and a post code which will be generated from the user interface. There is a limited budget available. This document aims to suggest a set of potential user stories with visuals so that the pilot projects can feedback and prioritise to enable VidaVia to maximise the development budget and deliver the most useful Service Finder app they can.

Approach

VidaVia have reviewed the requirements and drafted a vision statement to help with defining the scope. They have provided an outline of their suggested solution but then tried to empathise with a frontline worker to identify a set of user stories. These stories have then been visualised to help understand the stories. VidaVia have attempted to prioritise the stories but will consult with the LGA pilots to gain feedback and their priorities for the stories.

Vision Statement

The following has been developed, using an agile development format, as a potential vision statement.

- **A** suggested list of appropriate services
- **For** particular needs and circumstances within the proximity of a post code (irrespective of public sector boundaries)
- **That** allows them to select the ones of interest and that they are eligible for
- **So that:**
    - they can know (up to date) details of what local (radius of a post code rather than any boundary) support is available to improve their personal quality of life
- **Unlike** the current service directories which may not have reliable data and will be constrained to a sector or geographical boundary

Requirements list

These are the requirements gathered from the client

Feature requested	Initial thoughts
1, Provide a means of ascertaining a user's location, needs and circumstances	• VidaVia would expect to request the post code to ascertain location. This is expected to be the simplest for the end user and then it will be converted to lat-long for use with the open data and any maps. • VidaVia recognises that there are many need and circumstance terms. Asking the end user to select the most appropriate one is an interface issue. Knowing that frontline workers prefer to have a conversation with their clients/patients we would aspire to provide a series of questions (could implement the three conversation method) which can be configured by the admin and align a set of terms as potential answers. The terms will be a set of button/panes that can be touched/clicked to indicate a selection (see balsamiq). However, for version 1, we would expect to provide a set of four defined questions. The first to understand the group of circumstances and then the detail level and the second to understand the grouping of need and then the detail level. We also understand that there may be a mapping between circumstance and need that could pre-populate but for now we won't concern ourselves with this. • Vidavia will store the needs and circumstances selected in a cookie so as not to lose them from a session but will not, at this stage, provide a user account. A simple user account will be useful for saving location, need & circumstance profile, personas (see later), any liked services and reporting of any errors/feedback. • Vidavia recognises this is about using the LGA terms rather than simply searching for a text match and so would offer a search of terms capability to help an end-user find the appropriate terms if they didn't want to use the touch pane/button options.
2, Request from the API a prioritised list of services with as much or little of the above information as the user provides	• VidaVia will work with whatever APIs are provided but it is expected initially that the APIs will be relatively simple and offer one need and one circumstance to extract appropriate service-types to then filter with the long-lat. VidaVia would want to relate to audience-type and eligibility where the data is included. • If Porism develop an API of need and circumstances with Boolean then VidaVia will make use of it as it would if there were mappings between circumstance and need. • VidaVia believe that people prefer to be specific on timings and days and so will gather available times to prioritise services that match. This may be through an API or ordering the data that is retrieved. VidaVia recognises that this data will not always be available for services so would prioritise those that are at the top of any suggested services.
3, Present information in an easily consumable format that enables the user to discover & use their chosen service(s)	• Based on VidaVia's experience of working with Sefton then VidaVia is likely to mock-up a solution which has a summary listing of a service allowing the user to 'like' services and view them on a 'my services' page. Also to have the option to view full details by clicking on a summary listing

	and then an option to enter an email address to receive a PDF of the services on the 'like' list. • The user will also have the option to click the service url to open a browser window to view the services' website.
4, Offer extra features to make the application appealing	• VidaVia have already mentioned focusing on conversations and using questions to help the user interface determine which terms are appropriate. We would want to extend this functionality to allow the 'admin' of a system to define the questions e.g. 3 conversations or asset-based questions such as 'What do you enjoy'. The configuration will make the LGA needs and circumstance terms relevant to however a particular context works. • Vidavia understands that this app will be used by frontline professionals who will be dealing with various circumstances, needs and conditions. To allow workers to work faster and be proactive with clients whilst avoiding any GDPR implications then VidaVia would like to off an implementation of personas. This will allow the creation of a persona e.g. Elderly lady or Out of hospital or Frail person or Young family or Single parent or Unemployed person etc with certain needs and circumstances already defined. These can be stored without any GDPR issues as they are circumstantial but when the specific post code is applied then will provide a really fast and easy way to suggest local services to the person presenting to the frontline worker. This could even be used by family, friends and carers to describe the people in their family or who they are looking after. • VidaVia would like to include a simple map of where a service is located physically and a map view of all liked services but an extra feature offered (within the prioritisation exercise) would be the ability to get directions (walk, drive, public transport) from the post code entered.
5, Work on desktop, tablet & mobile devices with the latest version of Chrome, Firefox, Safari, Edge and IE browsers.	• Chrome 50+ From 2016; Firefox 45+ From 2016; Internet Explorer 11 From 2009; Edge 14+ From 2016; Safari 9+ From 2015. In practice, however, it is likely that the system will also work on versions prior to the above.
6, Be delivered as open source code hosted on GitHub	• VidaVia are used to working with Github and would deliver key delivery points to the Github repository chosen by the client (and would include a README to provide helpful information for developers wishing to use the system). Our internal source code control system is based on Subversion and so development would make use of that in-between delivery points.

Proposed Solution

VidaVia will develop a responsive browser-based application which will consume the LGA service-type model and services open data using given API calls to suggest services to meet selected needs and circumstances, age and proximity to a post code. If the budget permits then further filtering features will be added.

Roles

This app will focus on the frontline workers who need to find services in order to help their particular clients. This will include the typical roles of link workers or community connectors and referrers. The app could be used by service users but they are not target audience.

User Stories

The suggested epic stories have been prioritised using MoSCow[1] - Must have, Should have, Could have, Won't have this time. This prioritisation will be refined through pilot's feeding back so that VidaVia can then make best use of the budget available.

Some basic Balzamiq[2] sketched visuals have been developed to help in understanding the user stories but are not designs. A user interface design will be included in this work but pragmatic decisions will be made based on functionality required as to how much resource spent on the design.

Ref	Role	When	I Need	So that	VV Notes	MoSCoW
1.1	Frontline worker	I am talking with the client	to be able to quickly and easily find services appropriate to my client	I don't need to keep entering the same profile for similar people	Don't want to use a login and so will be device specific	M
1.2	Frontline worker	I am talking with the client	to be able to easily complete a user profile of my client	appropriate services can be suggested	Will identify main criteria that could be determined as a profile e.g. gender, age, location, needs, circumstances	M
1.3	Frontline worker	I am talking with the client	To be able to indicate the time of day that my client is available	so that the suggested services will be in line with user preference	Should we keep this simple	M
1.4	Frontline worker	I am talking with the client	to be able to search for a particular service	I can go straight to something they want without needing to add any other details	Will need to know if all text or specific fields	M
1.5	Frontline worker	I am talking with the client	to be able to search by type of service when I know what will be useful	I can see appropriate services quickly	Will need to use function levels to get to service-types and allow search as you type (if API is fast enough)	M
1.6	Frontline worker	I am talking with the client	to be able to search according to needs	I can see services related to their need without searching for something	Should you be able to choose several needs Will this prompt circumstances	M

[1] https://www.agilebusiness.org/content/moscow-prioritisation-0
[2] https://balsamiq.com/

				specific		
1.7	Frontline worker	I am talking with the client	to be able to search according to circumstances	I can see services related to their circumstances without searching for something specific	Should you be able to choose several circumstances Will this prompt needs	M
1.8	Frontline worker	I am talking with the client	to be able to ask a question with a defined set of terms as potential answers	so that I can use set assessment questions and be consistent with clients	Will hard code in some questions as part of a config file but there won't be an interface to configure questions and terms	M
1.9	Frontline worker	I am talking with the client	to filter out all services with eligibility criteria	I know that they are eligible		M
1.10	Frontline worker	I am talking with the client	to be able to send a list of the liked services to someone	I can pass the details to them, their intermediary and to myself	Perhaps just enter an email address and receive a PDF list with specific summary info	M
1.11	Frontline worker	I am talking with the client	to search across a particular area e.g. 1 mile of post code, neighbourhood, district, county,	I can see everything going on there	Hoping Porism can sort out the API for this 😉	M
1.12	Frontline worker	I am either back with a client or working for the client	to access contact details only	so that I can get in touch with someone as quickly as possible	Perhaps toggle between service details and contact details	M
1.13	Frontline worker	I am talking with the client	Add in criteria that will determine eligibility then use it to include those with eligibility criteria	I know that they are eligible	May have to limit it to whatever the profile determines	S
1.14	Frontline worker	I am talking with the client	to mark the services that my client likes from the ones suggested	I can focus on just those they like	Via list results and detailed page	S
1.15	Frontline worker	I am either back with a client or working for the client	to access previous searches	I can be reminded without having to remember what criteria I added last time	Show like a history feature in a browser. Ability to delete a search but keep useful ones?	S
1.16	Frontline worker	I am considering the search results	to refine my search by adding in other terms	I can reduce the number of suggested services	Make the search terms visible and allow easy clear and refine	S

1.17	Frontline worker	I am about to do a new search	to quickly clear my search and start new one	so that I don't run a search that includes previous info	Make the search terms visible and allow easy clear	S
1.18	Frontline worker	I am keeping myself up to date with information	to know new services in my area	so that I can keep myself abreast of services in my area	Allow notification to be accepted so that they disappear. Leave those not accepted as reminders	C
1.19	Frontline worker	I am keeping myself up to date with information	to know amended services in my area	so that I can keep myself abreast of services in my area	Allow notification to be accepted so that they disappear. Leave those not accepted as reminders	C
1.20	Frontline worker	I am checking how appropriate the particular service is for the client	to know the means used to deliver the service i.e. online, phone, venue, home visit	so that I can determine whether they can access the service ok	Check with Porism whether this is a controlled list	S
1.21	Frontline worker	when service has a physical venue to attend	to see a location map	so that I can determine that the client can get there ok	Could even just link out to open a map based on the lat/long etc. – depends on the info returned from API.	C
1.22	Frontline worker	when service has a physical venue to attend	to know the directions/distance from the given post code as walking, driving, public transport	so that I can see specifically if the client can get there ok	Does open street map allow this? Should we use Google or direct to a travel API?	C
1.23	Frontline worker	I am looking at all my liked services	to see them on a map display which has a scale appropriate to my proximity setting	So that I can easily see what I am looking to access	Can only show physical venues	C
1.24	Frontline worker	When I notice or hear about an error	To be able to report an error about a service	it is amended quickly	A simple link to the publisher is through their email from the open data so we could capture the error and email it.	C
1.25	Frontline worker	When my client is telling me about their experiences	to provide feedback to/about the service	the service provider has chance to improve		C

Data Visuals

The following are initial Balzamiq sketches to represent the user stories suggested. These are simply to help understand the sort of features that VidaVia is considering. There is not a lot of graphical data but attempts will be made to use any provider logos or even aligning service-types to images.

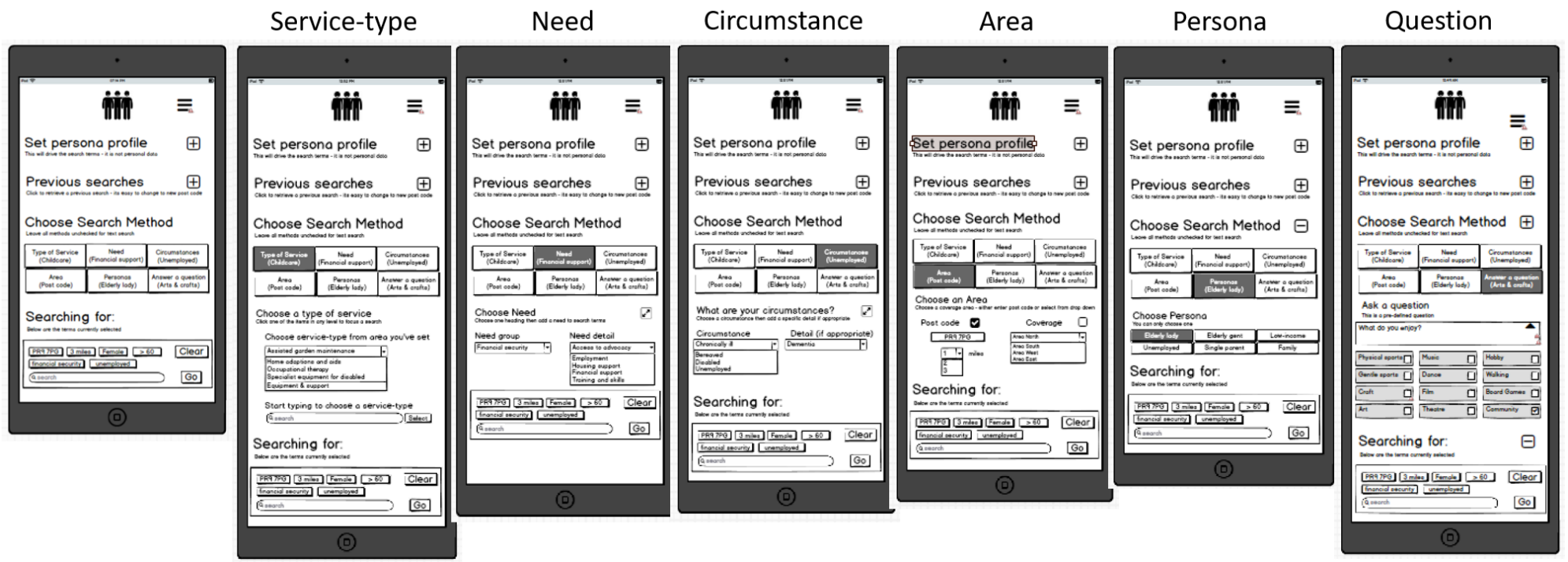
The visuals are also provided in PowerPoint to provide a larger view.

Main components

Sketch: below shows the home page and main features

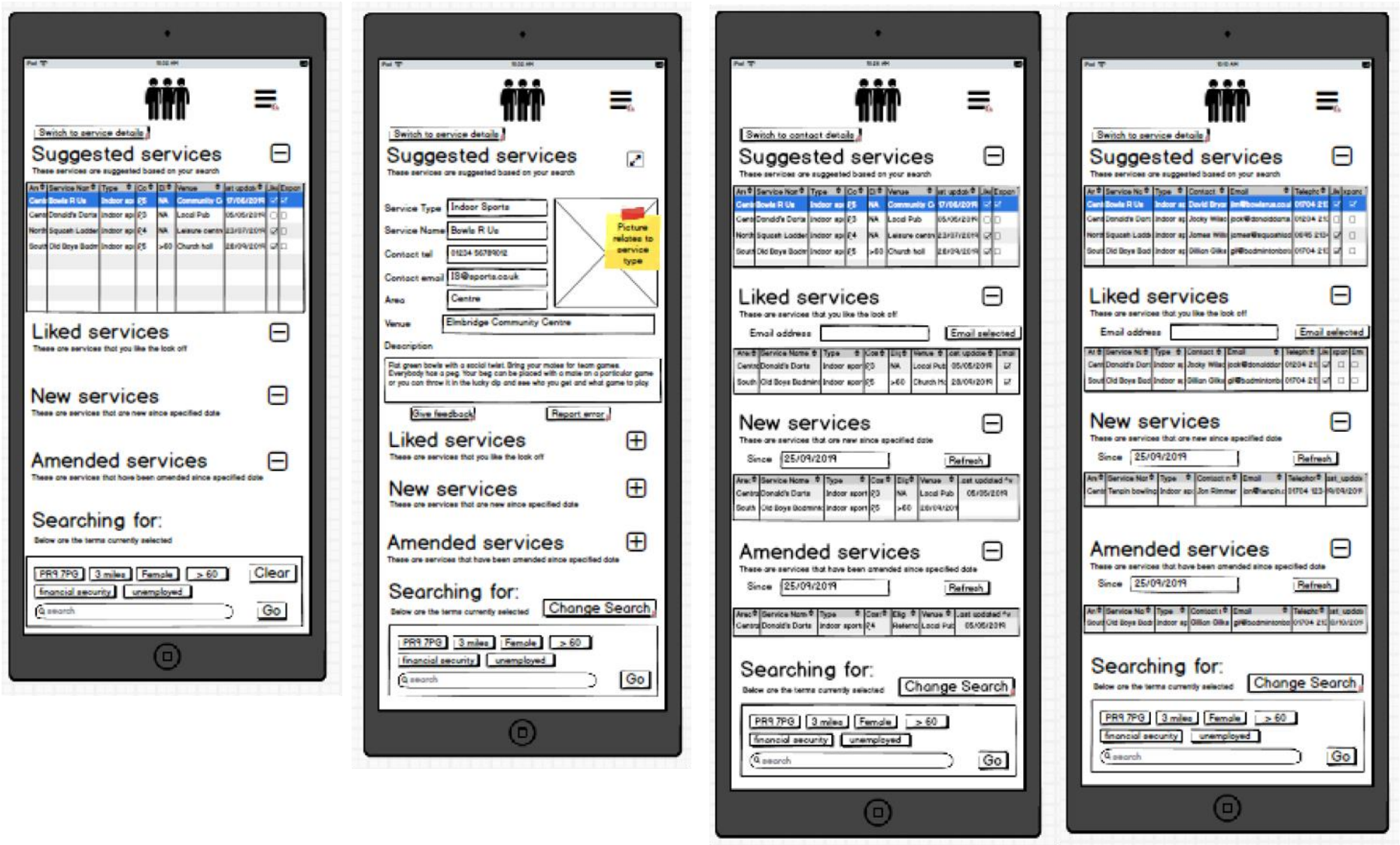
Search method

Sketch: below shows the methods provided to find services



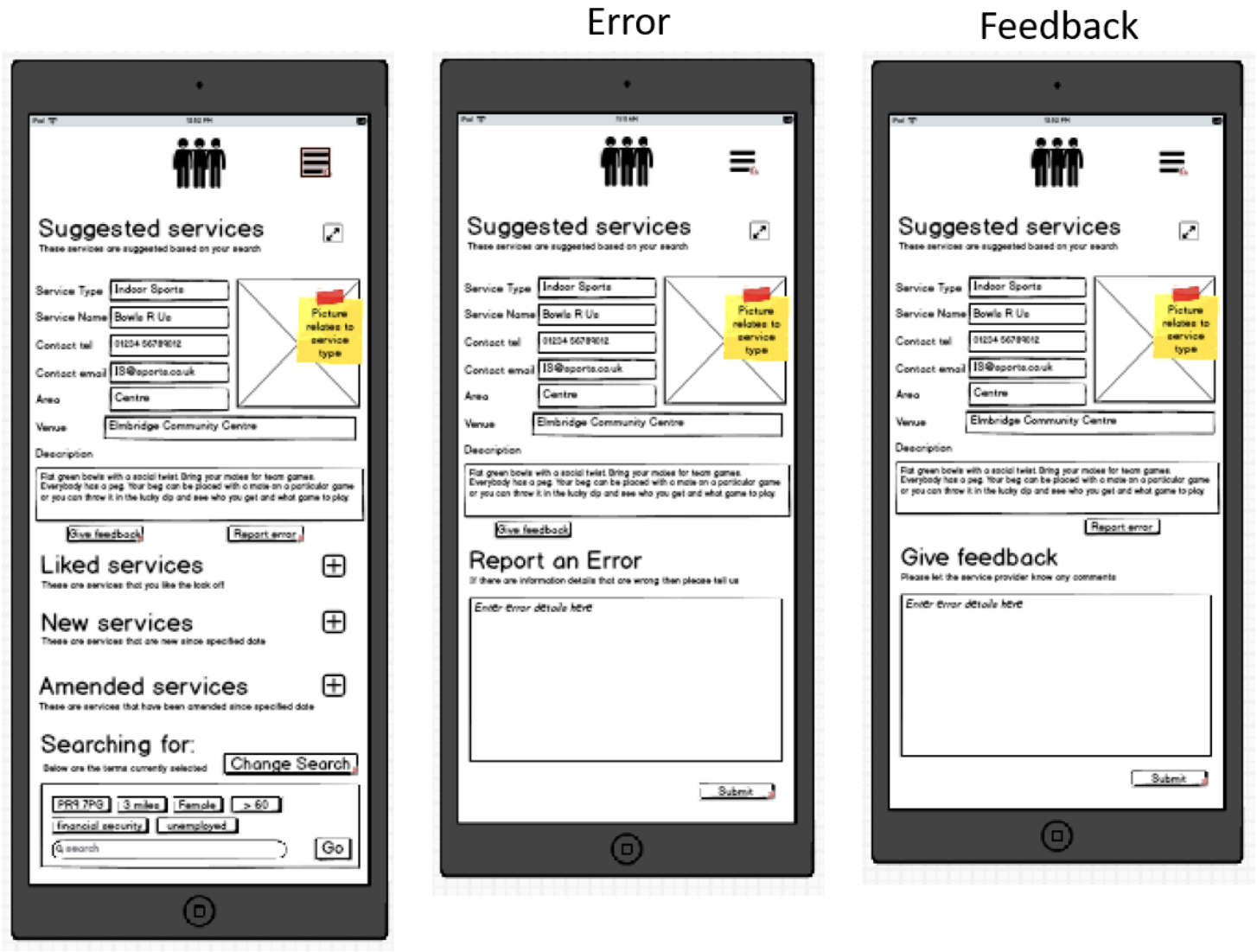
Search results

Sketch: below shows search results as lists for services or contacts and details of an individual service



Error & Feedback

Sketch: below shows reporting an error and providing feedback



Expected API Calls

The following are the API calls which we believe will be required in order to support the requirements marked as “Must have”, “Should have” and “Could have”.

Initialisation

We perceive that it may be useful to do a little early optimisation on the Service Finder such that we pre-populate some dropdowns/selects with the values that are available. For this we would require API calls to obtain the list of valid values for:

- Service Type
- Towns/Cities within postcode/proximity

Type-ahead / Helper

To allow for a type-ahead feature when searching with free text, we would require a lightweight rapid-response API that would help suggest terms based on the currently typed text. We would require this with the following:

- Service name : This would search just the name of the service and return a list of applicable service names.
- Service type: This would search the available set of service types and return a list of service types, each with name and ID.
- Service general : This would search the fields of the service which may contain information useful when searching (name, description, location) and return a list of service names.

In each of the above cases, the method parameter would be free text (perhaps limited to a certain maximum number of characters).

Email / Sharing / Feedback

Since the Service Finder will be front-end tool (with minimal back-end facilities) we would require the API to provide some features to help with the email/sharing/feedback aspects of the Service Finder.

- Report Error (optional service ID, optional search terms, free text description)
- Give Feedback (free text description)
- Send list of services (list of service IDs, email address)
- Send service (service ID, email address) – this may be satisfied with the previous method above (providing a list of just one service)

Service Search

To provide the bulk of the Service Finder functionality, the API will need to be able to support the searching for services with the following API methods.

- Search by postcode and proximity
- Search by postcode, proximity and service type
- Search by postcode, proximity, service type and keywords
- Search by postcode, proximity, service type and keywords
- Search by postcode, proximity, service type, keywords and eligibility

It may be better to provide a single API method that allows for each of the above to be optional (with some minimum parameters required to reduce the potential load on the server) – and that would also be acceptable for the Service Finder to use.

We may also require to provide the Service Finder user with results that have been “changed since” or “new since” – however, provided the API includes the “created” and “updated” timestamps on each result, this filtering can be done at the Service Finder.

Notes on Session Support

Note that the Service Finder can handle the aspects of “Liked” services and the details of handling a session. However, it may be worth considering the role that the API may be able to play in enabling this. For example, it may be possible to allow the concept of an anonymous “session” that is created by the API caller and referenced by a GUID created by the API. Each aspect of the session would then be subject to a session duration determined by the API – so that the API would be able to “garbage collect” outdated sessions to recover resources. This would allow the API to enable useful features such as lists, saved searches and liked searches – over a medium/short timeframe and without the need for account registration.

In this way, the API may be able to provide support such as “create session”, “close session”, “add service to list”, “send list to email” etc. which would then help to keep front-end clients more lightweight.

I am unclear of the GDPR and data privacy aspects of the above but since the session would be anonymous from start to finish – and since all sessions would ultimately be deleted in time – then there may not be much of an issue.

It may be possible to say that the API’s session facilities are used with the agreement that the API can make use of anonymised statistical information – which may help to provide useful information back to Service Commissioners etc.