D4.3 Prototype demonstrator applications at South Lakeland
# History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modification reason</th>
<th>Modified by</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>2018-01-12</td>
<td>Initial draft</td>
<td>ULANC-SCC</td>
</tr>
<tr>
<td>0.2</td>
<td>2018-01-26</td>
<td>Final draft for internal review</td>
<td>ULANC-SCC</td>
</tr>
<tr>
<td>0.3</td>
<td>2018-01-29</td>
<td>Revisions from FTB</td>
<td>ULANC-SCC</td>
</tr>
<tr>
<td>0.4</td>
<td>2018-01-30</td>
<td>Revision</td>
<td>ULANC-SCC</td>
</tr>
<tr>
<td>0.5</td>
<td>2018-01-31</td>
<td>Quality check</td>
<td>ULANC</td>
</tr>
<tr>
<td>1.0</td>
<td>2018-01-31</td>
<td>Final reviewed deliverable</td>
<td>ULANC</td>
</tr>
</tbody>
</table>
## Contributors to this Document

<table>
<thead>
<tr>
<th>Document Chapter</th>
<th>List of Sections</th>
<th>Contributors Information</th>
</tr>
</thead>
</table>
| **1. Introduction** | 1.1, 1.2, 1.4 | Frank Berker  
Frank Reins | FTB  
FTB |
| **1. Introduction** | 1.2, 1.3, 1.4, 1.5, 1.6 | Christopher Bull | ULANC-SCC |
| **2. The Mobile-Age Ecosystem: Stakeholders and Users** | All | Michail Papamichail  
Manolis Falelakis | AUTH  
AUTH |
| **3. The Demonstrator Application** | All | Christopher Bull  
Nigel Davies  
Mateusz Mikusz | ULANC-SCC  
ULANC-SCC  
ULANC-SCC |
| **4. Data Consideration** | All | Christopher Bull | ULANC-SCC |
| **5. Implementation and Deployment** | All | Christopher Bull  
Mateusz Mikusz  
Peter Shaw | ULANC-SCC  
ULANC-SCC  
ULANC-SCC |
| **6. Conclusions** | All | Christopher Bull  
Nigel Davies | ULANC-SCC  
ULANC-SCC |
# Table of Contents

**History** ................................................................................................................................... 2  
**Contributors to this Document** .............................................................................................. 3  
**Table of Contents** .................................................................................................................. 4  
**List of Figures** ......................................................................................................................... 8  
**List of Tables** ........................................................................................................................ 10  
**List of Abbreviations** ............................................................................................................ 11  
**Executive summary** ............................................................................................................. 13  
**1 Introduction** ....................................................................................................................... 14  
  1.1 Objectives of WP4 .............................................................................................................14  
  1.2 Scope and Relationship with other Deliverables .............................................................15  
  1.3 Document Structure .......................................................................................................17  
**2 The Mobile-Age Ecosystem: Stakeholders and Users** ...................................................... 18  
**3 The Demonstrator** ............................................................................................................. 21  
  3.1 Introduction .....................................................................................................................21  
  3.2 Scenarios ..........................................................................................................................21  
  3.2.1 Older Adult Searching for Local Activities ..................................................................21  
  3.2.2 Intermediary ..............................................................................................................22  
  3.2.3 Contributing Data Through Uploading Photos .........................................................22  
  3.2.4 Analytics ...................................................................................................................22  
  3.3 Key Features ....................................................................................................................23  
  3.3.1 Core Features ...........................................................................................................25  
  3.3.2 Application Features ...............................................................................................26  
  3.4 Architecture ....................................................................................................................27  
  3.4.1 Overview ..................................................................................................................27  
  3.4.2 Application and Data Container Manager .................................................................28  
  3.4.3 Analytics Framework ...............................................................................................28  
  3.5 Navigation and Flows .......................................................................................................29  
  3.5.1 Navigation Diagram ..................................................................................................29  
  3.5.2 Example Workflows - Launching Mobile-Age ..........................................................31
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.1</td>
<td>Social Events</td>
<td>91</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Routing Service</td>
<td>91</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Weather</td>
<td>93</td>
</tr>
<tr>
<td>4.1.4</td>
<td>Nearby facilities and Amenities</td>
<td>93</td>
</tr>
<tr>
<td>4.2</td>
<td>Data Processing</td>
<td>94</td>
</tr>
<tr>
<td>4.3</td>
<td>Data Creation and Sharing</td>
<td>96</td>
</tr>
<tr>
<td>5</td>
<td>Implementation and Deployment</td>
<td>98</td>
</tr>
<tr>
<td>5.1</td>
<td>Implementation Details</td>
<td>98</td>
</tr>
<tr>
<td>5.2</td>
<td>Quick Guide</td>
<td>99</td>
</tr>
<tr>
<td>5.3</td>
<td>Download Links and Demonstration</td>
<td>100</td>
</tr>
<tr>
<td>6</td>
<td>Conclusions</td>
<td>101</td>
</tr>
<tr>
<td>7</td>
<td>References</td>
<td>102</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1.1: Mobile-Age work package structure and inter-relationships ........................................16
Figure 2.1: Overview of Mobile-Age users and stakeholders .........................................................18
Figure 2.2: Mobile-Age users and stakeholders in co-creation of open data based public services ..................................................................................................................20
Figure 3.1: Android’s accessibility options in the device’s settings menu ......................................24
Figure 3.2: An app screen using the default accessibility settings ..................................................24
Figure 3.3: An app screen using adjusted accessibility settings (larger font and display size) .................................................................................................................................25
Figure 3.4: High-level architecture diagram consisting of the analytics framework (green), external services and data sources (blue), and the application and personal data store management (yellow) ..................................................................................................................27
Figure 3.5: Navigation diagram ....................................................................................................30
Figure 3.6: Navigation workflow - Launching the Mobile-Age application ....................................31
Figure 3.7: Navigation workflow - Adding a location to constrain location-based searches .........32
Figure 3.8: Navigation workflow - Configuring the preferred walking distance to one mile ......33
Figure 3.9: Navigation workflow - Configuring the maximum driving time to one hour ..........34
Figure 3.10: Navigation workflow - Specifying the preferred maximum cost for social events to £50 per event ........................................................................................................35
Figure 3.11: Navigation workflow - Specifying the preferred transportation method to ‘Car’ .................................................................................................................................36
Figure 3.12: Navigation workflow - Searching for an event occurring ‘This month’ ..............37
Figure 3.13: Navigation workflow - Searching for an event within walking distance ...............38
Figure 3.14: Navigation workflow - Adding a featured event to a personal calendar ...............39
Figure 3.15: Navigation workflow - Viewing an event on the map ............................................39
Figure 3.16: Navigation workflow - Viewing detailed information about an event ..................40
Figure 3.17: Navigation workflow - Viewing contact information for an event ..........................41
Figure 3.18: Navigation workflow - Viewing all services that are offered and retrieving details about a specific service .................................................................42
Figure 3.19: Navigation workflow - Viewing new volunteering opportunities .........................42
Figure 3.20: Navigation workflow - Searching for a specific volunteering opportunity around fundraising .........................................................................................................43
Figure 3.21: Navigation workflow - Viewing a set of good and bad example poster photos .................................................................................................................................44
Figure 3.22: Navigation workflow - Taking and uploading a picture as a contribution ..........45
Figure 3.23: Navigation workflow - Disabling analytics tracking for the Events app ..............46
Figure 3.24: Navigation workflow - Viewing information about the Mobile-Age application .................................................................................................................................46
Figure 3.25: The Launcher screen ................................................................................................48
Figure 3.26: Events app - Start screen ........................................................................................50
Figure 3.27: Events app - Start screen (carousels 2-3) ................................................................51
Figure 3.28: Events app - Start screen (carousels 4-5) ................................................................51
Figure 3.29: Events app - Start screen (end of carousel) ..........................................................52
Figure 3.30: Events app - Event description .............................................................................56
Figure 3.31: Events app - Event actions ..................................................................................56
Figure 3.32: Events app - Event details (1 of 4) .......................................................................57
Figure 3.33: Events app - Event details (2 of 4) .......................................................................57
Figure 3.34: Events app - Event details (3 of 4) .......................................................................58
Figure 3.35: Events app - Event details (4 of 4) .......................................................................58
Figure 3.36: Events app - Event location .................................................................................60
Figure 3.37: Events app - Event location and journey ...............................................................61
Figure 3.38: Events app - Map interactions (event details) .......................................................61
Figure 3.39: Events app - Amenities and facilities map layers ...............................................62
Figure 3.40: Events app - Predefined searches (‘This month’) ..................................................65
Figure 3.41: Events app - Predefined searches (‘walking distance of home’) .........................65
Figure 3.42: Events app - Customise search keywords ..............................................................66
Figure 3.43: Events app - Customise search for all categories ...................................................66
Figure 3.44: Events app - Customise search for individual categories ....................................67
Figure 3.45: Events app - Customise search by location ...........................................................67
Figure 3.46: Events app - Customise search’s travel preference (unspecified) .........................68
Figure 3.47: Events app - Customise search’s travel preference options .....................................68
Figure 3.48: Events app – Customise search travel preference (car reachability) ......................69
Figure 3.49: Events app - Customise search travel preference (reachability options) ...............69
Figure 3.50: Events app - Customise search by date ...............................................................70
Figure 3.51: Events app - Customise search by date range .......................................................70
Figure 3.52: Events app - Customise search’s date selection popup .........................................71
Figure 3.53: Events app – Search results for nearby events (in Kendal, South Lakeland) .........72
Figure 3.54: Events app - On-device keyword filtering of search results ....................................72
Figure 3.55: Services app - Landing screen ............................................................................74
Figure 3.56: Services app - Example service (scrolled viewport; 1 of 3) ....................................75
Figure 3.57: Services app - Example service (scrolled viewport; 2 of 3) .................................75
Figure 3.58: Services app - Example service (scrolled viewport; 3 of 3) .................................76
Figure 3.59: Services app - Filtering listed services .................................................................76
Figure 3.60: Volunteering app - Landing screen .....................................................................77
Figure 3.61: Volunteering app - Volunteering opportunity details .............................................78
Figure 3.62: Volunteering app - Volunteering opportunity details (continued) .......................78
Figure 3.63: Volunteering app - Filtering by keyword ..............................................................79
Figure 3.64: Contribute Poster app ........................................................................................80
Figure 3.65: Contribute Poster app - After photo uploaded .....................................................81
Figure 3.66: Contribute Poster app - A user’s example of a good photo ....................................82
Figure 3.67: Contribute Poster app - A user’s example of a bad photo .....................................83
Figure 3.68: Profile app .........................................................................................................84
Figure 3.69: Profile app - Expanded locations section ...............................................................84
Figure 3.70: Profile app - Expanded locations section (continued) ..........................................85
Figure 3.71: Profile app - Expanded ‘Travel and Mobility’ section (1 of 3) ...............................85
Figure 3.72: Profile app - Expanded ‘Travel and Mobility’ section (2 of 3) ...............................86
Figure 3.73: Profile app - Expanded ‘Travel and Mobility’ section (3 of 3) ...............................86
Figure 3.74: Profile app - Expanded ‘Financial’ section .........................................................................................................................87
Figure 3.75: Settings app .............................................................................................................................................................................88
Figure 3.76: Settings app – Analytics management screen ....................................................................................................................89
Figure 3.77: About app ..................................................................................................................................................................................90
Figure 4.1: OpenRouteService example – 30-minute driving distance .................................................................................................92
Figure 4.2: OpenRouteService example – 30-minute walking distance ................................................................................................93

List of Tables

Table 1.1: overview of the use cases covered by the field sites .............................................................................................................14
Table 3.1: Data interaction - Example data service interaction ........................................................................................................47
Table 3.2: Data interaction - Collecting analytics ..................................................................................................................................48
Table 3.3: Data interaction (Events app) - Get featured events IDs ..................................................................................................52
Table 3.4: Data interaction (Events app) - Get event details for specific event IDs ..............................................................................53
Table 3.5: Data interaction (Events app) - Query for events ................................................................................................................53
Table 3.6: Data interaction (Event view) - Get an event’s weather forecast ........................................................................................59
Table 3.7: Data interaction (Event map) - Get data from Overpass service, showing amenities on a map ..................................................................................................................................................62
Table 3.8: Data interaction (Event map) - Get data from OpenRouteService, showing a route on a map .........................................................................................................................................................................63
Table 3.9: Data interaction (Search results) – Request an isochrone ..................................................................................................73
Table 3.10: Data interaction (Contribute Poster app) - Upload a photo ...............................................................................................81
Table 3.11: Data interaction (Profile app) - Get auto-complete list of location suggestions ........................................................................87
Table 3.12: Data interaction (Settings app) - Request analytics details ..................................................................................................89
Table 4.1: Data interaction (Back-end) - Geocoding during back-end processing of data ............................................................................91
Table 4.2: Database schema for event data ........................................................................................................................................94
Table 4.3: Database schema for uploaded images ..................................................................................................................................97
### List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>AUTH</td>
<td>Aristotle University of Thessaloniki</td>
</tr>
<tr>
<td>BEM</td>
<td>Block Element Modifier Methodology</td>
</tr>
<tr>
<td>CC-By-SA</td>
<td>Creative Commons by Attribution-ShareAlike</td>
</tr>
<tr>
<td>CMS</td>
<td>Content Management System</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>DOM</td>
<td>Document Object Model</td>
</tr>
<tr>
<td>FTB</td>
<td>Forschungsinstitut Technologie und Behinderung</td>
</tr>
<tr>
<td>GIS</td>
<td>Geographic Information System</td>
</tr>
<tr>
<td>HTML</td>
<td>Hyper Text Markup Language</td>
</tr>
<tr>
<td>ifib</td>
<td>Institut für Informationsmanagement Bremen</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>NGO</td>
<td>Non-Governmental Organization</td>
</tr>
<tr>
<td>OGS</td>
<td>Open Geospatial Consortium</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>OSCPSEP</td>
<td>Open Senior Citizen Public Service Engagement Platform</td>
</tr>
<tr>
<td>OSM</td>
<td>OpenStreetMap</td>
</tr>
<tr>
<td>PaaS</td>
<td>Platform as a Service</td>
</tr>
<tr>
<td>POI</td>
<td>Point of Interest</td>
</tr>
<tr>
<td>RCM</td>
<td>Region of Central Macedonia</td>
</tr>
<tr>
<td>SaaS</td>
<td>Software as a Service</td>
</tr>
<tr>
<td>SASS</td>
<td>Syntactically Awesome Stylesheets</td>
</tr>
<tr>
<td>SCC</td>
<td>School of Computing and Communications (Lancaster University)</td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
<tr>
<td>SLDC</td>
<td>South Lakeland District Council</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>ULANC</td>
<td>University of Lancaster</td>
</tr>
<tr>
<td>UPM</td>
<td>Universidad Politécnica de Madrid</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Identifier</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>WCAG</td>
<td>Web Content Accessibility Guideline</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>WFS</td>
<td>Web Feature Service</td>
</tr>
<tr>
<td>WMS</td>
<td>Web Map Service</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>ZGZ</td>
<td>City of Zaragoza</td>
</tr>
</tbody>
</table>
Executive summary

This deliverable reports on the prototype demonstrator application developed for the South Lakeland fieldsite. The demonstrator application is designed to encourage independent living by tackling the problem of social isolation and loneliness and, in particular, by allowing older adults to benefit from on-line information regarding social events in their area. The demonstrator includes a set of internal apps and core features that are used across all these apps. The core features include:

- **Launcher:** A homepage screen that enables end-users to navigate to applications within the Mobile-Age eco-system.

- **Profile Management:** The user profile manager allows end-users to supply general preferences that are shared across all applications within the Mobile-Age ecosystem.

- **Analytics:** A framework for capturing and reporting user interactions with applications and services. The analytics framework supports the capture of interaction events across Mobile-Age applications, Web sites and services.

These core features underpin a series of internal apps that were iteratively developed through co-creation engagements. These include:

- **Events:** The Events app is created to provide greater visibility to local social events, and to reduce barriers to attending the events (for example providing queries that filter events by a person’s accessibility requirements).

- **Services:** The Services app provides information about local council services.

- **Volunteering:** The Volunteering app provides information on volunteering opportunities in the local area of the user.

- **Contribute Poster:** The Contribute Poster app allows end-users to take photos of posters or newsletters and upload them to our servers, creating a new data source for future processing.

The South Lakeland apps use a variety of open data, including OpenStreetMap data and events data from the local authority’s website. The demonstrator also generates data, through the collection of analytics and the uploaded photos from the Contribute Poster app.

The demonstrator is implemented as a hybrid mobile web application using Apache Cordova, and is supported by a NodeJS/Express/MongoDB back-end. The demonstrator targets the Android and iOS platforms and is available from: [https://scc-mobileage.lancs.ac.uk/app/](https://scc-mobileage.lancs.ac.uk/app/).

The description of the demonstrator application is accompanied by supporting scenarios that illustrate the use of the application’s key features.
1 Introduction

1.1 Objectives of WP4

Deliverable D4.3 is part of Work Package 4, which is responsible for the development of front-end components and demonstrator applications. More specifically, the objectives of Work Package 4 are as follows:

- Specify technical requirements and specification of demonstrator applications for each of the four trial sites (Table 1.1).
- Development of assistive generic front-end components - not specific to any particular trial site.
- Design studies and mock-ups for co-creation purposes.
- Develop prototype demonstrator applications at each of the four trial sites.

The main objectives of this deliverable are:

- Present the key features, design, implementation, and deployment of the demonstrator application at the South Lakeland field site.
- Illustrate navigation and workflows within the app.
- Describe the data considerations within the app, including used and created data sources, data processing and data sharing.

Table 1.1: overview of the use cases covered by the field sites

<table>
<thead>
<tr>
<th>Use Case ID</th>
<th>Socio-spatial Aspects of Social Inclusion</th>
<th>Extending Independent Living through Reducing Loneliness and Social Isolation</th>
<th>A Safe and Accessible City for Elderly People</th>
<th>Personal Health Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile Services</td>
<td>Map-based social networking and mobile open information services</td>
<td>Assessing the needs of older adults to extend independent living</td>
<td>Map-based data curation and collaborative map creation</td>
<td>Health-related open data information services for older adults</td>
</tr>
<tr>
<td>Problem Domain</td>
<td>Connecting people, open data &amp; place through social networking for older adults</td>
<td>Assessing and tracking the service provision for older adults to support their independent living</td>
<td>Empowering older adults to create collaboratively maps with accessible routes, alert city</td>
<td>Consuming open data feeds for older adults</td>
</tr>
</tbody>
</table>
1.2 Scope and Relationship with other Deliverables

As illustrated in Figure 1.1, Work Package 4 consists of five tasks. This deliverable is the result of work performed within Task 4.3 “Design studies and mock-ups for co-creation purposes”, and Task 4.4 “Development of mobile front-ends for demonstrator applications”.

The work presented here also relates to Task 4.2 “Development of generic front-end components”, whose front-end components (reported in D4.1) are utilised within all of the demonstrator applications across all of the trial sites, where appropriate.

The functionality, as well as the look and feel of the delivered applications, were elaborated and iterated during the co-creation activities performed in Work Package 3, which are reported in D3.3 “Senior citizen engagement report South Lakeland”.

This deliverable is a counterpart to D4.2 “Prototype demonstrator application at Bremen”; the trial site in Bremen has been creating demonstrator applications in parallel to South Lakeland. Other prototype application demonstrators at the remaining two trials sites (Zaragoza and the Region of Central Macedonia) will be reported in upcoming deliverables (D4.4 and D4.5, respectively).
This deliverable builds directly upon the technical requirements and specifications reported in D4.1. The use of analytics within this demonstrator prototype relates to deliverable D2.4 “Behaviour Analytics & Workflow Software Components”.

In February 2018 the Task 3.6 "Mobile Service Evaluation" will start to evaluate the developed demonstrator applications. It is planned to incorporate the evaluation results in another version of the demonstrators. That means that the described stage of the demonstrators in this document will not reflect the final versions at the end of the project. During the evaluation of the application by the older adults, it is to be expected that some bugs or problems may be reported. The improvement of the app will be continued, of course, so the content and the look of the app may change slightly in the future.
1.3 Document Structure

The remainder of the document is structured as follows: we first describe the stakeholders and users of the Mobile-Age ecosystem (section 2), clearly defining terms such as ‘User’. We then describe the demonstrator application designed and developed for South Lakeland (section 3). This includes introducing illustrative scenarios to frame the app’s usage.

In section 4, we describe the specific data considerations of the demonstrator application including data sources, created data, data processing, and data sharing.

Finally, we describe the implementation and deployment details of the demonstrator application (section 5), and conclude the deliverable with key findings (section 6).
2 The Mobile-Age Ecosystem: Stakeholders and Users

This section was first developed as a response to reviewer feedback for deliverables D2.1-D2.4, and is reproduced here to aid the reader.

Figure 2.1 provides an overview of the roles of Mobile-Age users. Please note that some roles may overlap.

![Figure 2.1: Overview of Mobile-Age users and stakeholders.](image)

**Local/regional governments**: May be managing the co-creation activities, define features of the applications, and serve as experts for a specific service domain. In many cases, local governments are also the data owners.

**Software developers**: May be independent developers or companies, or working for IT-departments in public authorities or civil society organizations such as the Open Knowledge Foundation. They develop applications using the platform and participate in the co-creation activities, adjusting the applications to accommodate for the participants’ requests and demonstrate results in an iterative process.

**Older adults**: Key stakeholders of the Mobile-Age project and the main users of mobile applications being developed. They may participate in core project group or engage in broader co-creation activities.

**Service providers** such as government, social welfare organizations, religious congregations or NGOs may be part of the core project group or be engaged for specific input. Some service providers may also provide (open) data.
Intermediaries include professionals and non-professionals that may support the co-creation activities by providing input for specific tasks in the co-creation process. They may become users of the applications developed.

Facilitators are experienced individuals in the work with older adults and/or groups. They support the co-creation activities through e.g. running workshops, focus groups, interviews.

Other organizations & individuals comprise for example, senior citizen organizations, senior citizens’ clubs (e.g. computer clubs) but also media and journalists that report on the co-creation activities, and thereby support engagement as well as dissemination.

Based on how these entities interact with the Mobile-Age ecosystem, they can be categorized in two main groups, as illustrated in Figure 2.2:

- **MADE Users**: These are users that make direct use of the Mobile-Age Deployment Environment and consist of the following:
  - Software Developers
  - Service Providers

- **End Users**: These are users of the mobile applications developed using the Mobile-Age platform.
  - Older adults
  - Local / regional governments
  - Intermediaries (professional and non-professional)
  - Facilitators
  - Other Organizations and Individuals
Figure 2.2: Mobile-Age users and stakeholders in co-creation of open data based public services.
3 The Demonstrator

3.1 Introduction

The South Lakeland Mobile-Age demonstrator application is designed to encourage independent living by tackling the problem of social isolation. The demonstrator was developed using a process of co-creation as described in deliverable D3.3.

We approached our co-creation group with an initial brief to explore the area of independent living. The group quickly focused in on the issue of loneliness and social isolation and our application development followed this path. In particular, the group identified challenges in finding information on local events and determining their accessibility. It is this challenge that our demonstrator seeks to address in the form of an “events app” that provides event information and travel support designed specifically to meet the needs of older adults. The events app is implemented within a broader framework for app development and deployment that supports Mobile-Age concepts including analytics and user profiles to address trust issues. A small number of additional apps were also created to supplement the “Events app” and to demonstrate the extensibility of the core framework.

3.2 Scenarios

To illustrate the use of the demonstrator application, we have compiled a number of usage scenarios. These help situate the application within the problem-space and describe the benefits to the demonstrator stakeholders. Note that throughout the remainder of this document, the term ‘user’ will be used as shorthand for the ‘older adult’ user, as defined in section 2. This is because older adults are the primary users of the demonstrator applications.

3.2.1 Older Adult Searching for Local Activities

This scenario describes a typical use case for an older adult looking for local social events.

*Margaret, a 78-year-old widow, is interested in finding something to do this week. She has limited mobility - she is comfortable walking, but only for short distances. She opens the Mobile-Age app and starts a search for events within 20 minutes walking distance. Her search results present her with many events that she was not aware of. She checks the address of the events to ensure she is happy walking to them. Once she has finished searching for events, she chooses to open the Volunteering app to look for any local opportunities that she can apply for -- she has previously found volunteering enjoyable and a good way to socialise with people outside her immediate social circles.*

*Later that week, Margaret is talking to a friend. They discuss her friend’s future knee surgery, and how she has concerns about being able to do things around the house. Margaret opens the Mobile-Age app, navigates to the Services app, and browses for services (such as the ‘assisted bin collection’ service she had heard about) to help her friend in the home -*
she ends up reading about the ‘Independent Living’ service and passes the contact information to her friend.

In this scenario Margaret has the opportunity to find information about social events, with the search criteria taking into consideration her mobility limitations. The initial benefit is the ability to find information on events and activities that she is not already aware of. Another benefit is the ability to limit results by considering whether an event can be reached by walking specifically, which ensures a route is available (end-to-end) that is suitable for walking and takes a maximum distance or time specified by Margaret. Also, in this scenario, Margaret explores the Services and Volunteering apps for local information.

A typical navigation workflow of using the Events app is depicted in section 3.5.4, for the Services app in section 3.5.5, and for the Volunteering app in section 3.5.6.

### 3.2.2 Intermediary

This scenario describes a situation where a user shares their profile information with an intermediary.

Patrick, 83, lives rurally and spends a lot of time at home. He can use the Mobile-Age application to search for local services but finds it hard to configure the search to reflect his specific travel and financial constraints. He is able to share his profile with an intermediary at his local library who can help him store a set of preferences that are used in future searches.

The key here is that users are able to share their profiles with those they trust to help them access services.

A typical navigation workflow of using the Profile app is depicted in section 3.5.3.

### 3.2.3 Contributing Data Through Uploading Photos

User-contributed data can help where data is scarce.

Dorothy, 71, is attending a routine GP appointment. Whilst sat in the waiting room, she passes the time by reading the posters and leaflets on the wall. One poster lists local events. Remembering the Mobile-Age app, she pulls out her phone, opens the ‘Contribute Poster’ app, and takes pictures of the poster to upload it.

Posters often have information about local or national services and events. By contributing photos of these posters, users are contributing data that can be used in a wide range of applications.

### 3.2.4 Analytics

Knowledge about citizen interest in events and whether they attend is potentially useful information to local authorities:
George, 72 and Barbara, 74, are looking for activities they can participate in together. They use the Events app to search for events this weekend. After looking at the initial search results, they decide to perform another search, but this time limiting it to events within 30 minutes drive of their home. George browses the new results and selects an event they are interested in and that they subsequently attend.

Council official Joe is interested in why certain council-run events have better participation than others. Some of the organisers know anecdotally that some dates, times, and places are better for attendance than others - Joe would like to explore and better understand why this may be. Joe looks at analytics reports and sees patterns in the way people use the social Events app: for the events that he is studying those users querying car transport options are more likely to actually attend the event than those using public transport - who often search and select an alternative event once the travel options are displayed. Thus, the system provides Joe with evidence that many older adults would like to attend specific events but are deterred by public transport options.

During this process, analytics data is collected and sent to the Mobile-Age analytics service. Analytics data is collected on aggregate app usage, including user interaction with specific events (based on pageviews), a user’s potential intent to attend an event (tracking ‘add to calendar’ or ‘Find transport options’ button events), and tracking conversion rates from intent to attendance at events (using either automatic location-based geofence tracking at the event location, or asking the user for confirmation of attendance through the user interface).

Local authorities can use the analytics data to determine characteristics such as event intent-attendance conversion rates. During the deployment and evaluation phases we plan to test a broad range of analytics capabilities. Other opportunities for analytics are described in deliverable D2.4.

### 3.3 Key Features

The demonstrator application is a cross-device app running on Android and iOS. Key features are discussed in the following subsections, grouped into core features and application-specific (within the Mobile-Age demonstrator app) features.

The application has been designed specifically to integrate with standard device accessibility options. In the Android operating system these are found under “Accessibility” within the “Settings” app. For example, font and user interface elements can be increased or decreased in size system-wide (Figure 3.1). These options have been adjusted to show larger text and elements in the demonstrator application, which can be seen throughout all of the screenshots that are presented in this deliverable.
Figure 3.1: Android's accessibility options in the device's settings menu.

Figure 3.2: An app screen using the default accessibility settings.
3.3.1 Core Features

Core features are those that are independent of any specific application and provide the basic framework for interaction with the demonstrator application.

The first core feature is the app launcher. The launcher is the first screen that appears when the demonstrator application is opened, providing an interface for the user to access the application’s internal apps from a single screen. The available apps are loaded as a list of buttons; pressing one loads and navigates to that app. The launcher also provides a framework for sharing functionality and data (such as user profile and analytics) across apps. These internal apps are modular, utilising the core features described here; the application-specific features are described in the following section.

The second core feature is the user profile manager, which is also accessible across the demonstrator’s apps. The profile manager provides apps with access to a data container for the current user, which houses user data and preferences across the apps. Furthermore, the user’s profile can be shared with people external to the Mobile-Age application, for example intermediaries, enabling mechanisms for user support (see section 3.6.11).

The third core feature is the availability of analytics across the launcher and all of the apps. The analytics service provides an innovative end-user analytics module which enables local authorities, application developers and other organisations the ability to improve the quality of services they offer for older adults. To support this, the analytics service collects web, device, and cross-app analytics of users’ app use and services. This analytics service is
reported in greater detail in deliverable D2.4 ‘Behaviour Analytics & Workflow Software Components’.

### 3.3.2 Application Features

Complementing the core features of this demonstrator application are the internal apps, and their app-specific features. The available apps in the prototype demonstrator application are: (i) Events app, (ii) Services app, (iii) Volunteering app, and (iv) Contribute Poster app.

These apps open up available data and services, providing co-created features that are specific to older adults. The data available in these apps is provided by a local authority (South Lakeland District Council) and an age charity (AgeUK South Lakeland¹). Other open data is sourced to provide features that consider, for example, locations of local amenities, car parks, and bus stops (see section 4.1.4 for more details).

**Events app** – This provides access to information for social events in a user’s local area. A primary intention of the South Lakeland demonstrator application is to reduce social isolation and loneliness. This app aims to reduce barriers to social-connectedness by providing information about social events and their accessibility. One of the mechanisms to achieve this is presenting serendipitous social events on the homepage of the Events app based on proximal features that are relevant to the user (e.g. showing groups of events for today, this weekend, nearby, and near home). A search mechanism is also provided, allowing a user to search for events based on (e.g.) their location and reachability preferences, further supporting the ability to find events based on the user’s accessibility. Once an event is selected, it can be added to the user’s calendar, which allows the user to organise their schedule. Whilst viewing their chosen event, a user can request to view transportation options to the event, and also show routes to the event on a map in conjunction with other open data (including the location of nearby toilets and benches). This app is designed to function with limited network functionality by providing offline caching of nearby events, and where possible providing features that function when offline.

**Services app and Volunteering app** – These two apps provide information on local services and volunteering opportunities, respectively. This data is also sourced from the local authority and age charity. These are presented to the user with information about the service or opportunity, and also provide contact information if the user is interested. The list of available services/opportunities can be filtered using a keyword search, displayed in the app’s footer.

**Contribute Poster** – This app enables end-users to become data contributors for the application. It asks end-users to take pictures of posters or newsletters, which are then uploaded to our servers. These images are uploaded along with geographic location coordinates. This data source has future potential for building up local knowledge, or even processing the image data to obtain additional event data, for example.

¹ [https://www.ageuk.org.uk/southlakeland/](https://www.ageuk.org.uk/southlakeland/)
3.4 Architecture

The demonstrator application architecture was designed to accommodate the requirements from the co-creation process and supports the types of scenarios described in Section 3.2.

3.4.1 Overview

Our architecture design for the demonstrator apps is shown in Figure 3.4. This architecture consists of three main elements designed specifically to encourage use by older adults: an analytics framework that enables widespread data collection yet addresses privacy concerns raised by older adults, a data storage and exchange facility that aims to lower the barrier of use while providing a security model that reflects older adults’ ways of working, and a unified application framework that provides “a walled garden” that helps reassure older adults of the trustworthiness of applications. For example, applications within the demonstrator such as Events, Profile, Services and Volunteering, are isolated applications that live within the
application framework and individually communicate with other components across the framework and external services.

### 3.4.2 Application and Data Container Manager

A recurring theme in our discussions with older adults, and reflected in the architectural design of the demonstrator application, is the notion of trust. Older adults are often wary of technology and extremely sensitive to issues of data sharing and privacy. Moreover, they fear the consequences of technology failing or being compromised (e.g. identity theft) and hence this limits their willingness to install new applications on their mobile devices.

The demonstrator architecture uses data containers, similar to [1] in which users and applications can safely store data (particularly user profiles) and manage access. This provides a means for applications to exchange common information in order to reduce the need for older adults to repeatedly enter the same data. Profile sharing between apps is supported in some current mobile operating systems. However, within our platform, profile containers are persistent and can be shared with other users - including remote users in order to facilitate technology assistance.

In addition to concerns about data, older adults often distrust new applications and are reluctant to install these on their mobile devices. Our solution is to employ a model in which individual applications form part of a larger suite that are invoked by the user through a common “launcher”. This type of approach has been widely used in multiple domains such as student portal and smart city applications [2]. The benefit of such an approach is that users are only required to install a single application, but new services can be provided dynamically and the user experience tightly controlled.

While neither the use of an application framework nor the use of data containers is new, we believe this is the first time these approaches have been suggested for use specifically to meet the needs of older adults.

### 3.4.3 Analytics Framework

The analytics framework in the demonstrator application is a crucial component to allow us and other application developers to understand how older adults interact across the Mobile-Age ecosystem. The opportunities that evolve from the analytics framework and its front- and back-end architectures have been described in more detail in D2.4 (“Behaviour Analytics & Workflow Software Components”).

We developed an analytics architecture that collects and processes data related to a user on their personal mobile device. This analytics architecture supports data captured by the device’s operating system, applications running under the “App Manager” umbrella and analytics data about the user that was originally captured outside of the mobile device, enabling the correlation of user-related datasets without the need for uploading any personal and sensitive datasets into other services. In addition, our analytics system utilises the concept of Privacy Mediators [3] allowing users to control the level of sharing of their personal data and providing a high level of confidence and control about the ownership and
use of personal data. The analytics framework consists of four main components: the Import Module, the Analytics Manager, Privacy Mediators, and the Export Module.

**Import Module**: This module allows the analytics framework to combine data from various sources. For example, it accesses analytics-related data from external services (e.g. statistics on council services), device-related logs (e.g. usage statistics), events from applications that are running within the demonstrator platform (e.g. interaction patterns), and user profiles stored within the demonstrator (e.g. user preferences). To be able to import data from these external sources, our framework features a set of import modules each of which are specific to the data source and which implement the relevant application programming interface to connect to the external service (e.g. accessing and parsing phone records).

**Analytics Manager**: The analytics manager orchestrates import modules, the combination of user-owned datasets (both from internal and external sources) to gain new analytical insights, and the privacy mediator pipeline.

**Privacy Mediators**: Due to the high sensitivity of the collected datasets, the privacy mediator component is designed to address issues in trust and data privacy. Prior to the export and use of any dataset, it is routed through a pipeline of user-specified privacy mediators that restrict the access to datasets both in terms of access control and level of information sharing. Example mediators could provide features to remove user-identifiable information prior to export, and define levels of redaction.

**Export Module**: To allow third party services or other users to gain a better understanding on interaction events of a person, the export component enables the specification of “injection modules” that post analytical insights in compliance to user configuration and privacy mediators to both external services as well as applications that are part of our application framework.

Within this prototype demonstrator, we currently utilise a subset of components from the analytics framework. We support the collection of user interactions and device locations through the Import Module, that are exported in real-time to the Pheme analytics service for reporting purposes.

### 3.5 Navigation and Flows

#### 3.5.1 Navigation Diagram

Figure 3.5 illustrates the demonstrator application’s screens and where each can navigate to.
In the following sections we present illustrative workflows for common tasks. These workflows are all generated using screen captures of the application when configured to run on a tablet.

*Figure 3.5: Navigation diagram.*
3.5.2 Example Workflows - Launching Mobile-Age

Launching the Mobile-Age application:

Figure 3.6: Navigation workflow - Launching the Mobile-Age application.
### 3.5.3 Example Workflows - Profile Settings

**Adding a location to constrain location-based searches:**

*Figure 3.7: Navigation workflow - Adding a location to constrain location-based searches.*
Configuring the preferred walking distance to one mile:

Figure 3.8: Navigation workflow - Configuring the preferred walking distance to one mile.
Configuring the maximum driving time to one hour:

Figure 3.9: Navigation workflow - Configuring the maximum driving time to one hour.
Specifying the preferred maximum cost for social events to £50 per event:

Figure 3.10: Navigation workflow - Specifying the preferred maximum cost for social events to £50 per event.
Specifying the preferred transportation method to ‘Car’:

Figure 3.11: Navigation workflow - Specifying the preferred transportation method to ‘Car’.
3.5.4 Example Workflows - The Events App

Searching for a specific event taking place ‘This month’:

Figure 3.12: Navigation workflow - Searching for an event occurring ‘This month’
Searching for an event within walking distance:

Figure 3.13: Navigation workflow - Searching for an event within walking distance.
Adding a featured event to a personal calendar:

Viewing an event on the map:

**Figure 3.14: Navigation workflow - Adding a featured event to a personal calendar.**

**Figure 3.15: Navigation workflow - Viewing an event on the map.**
Viewing detailed information about an event:

Figure 3.16: Navigation workflow - Viewing detailed information about an event.
**Viewing contact information for a specific event:**

*Figure 3.17: Navigation workflow - Viewing contact information for an event.*
3.5.5 Example Workflows - The Services App

Viewing all services that are offered and retrieving details about a specific service:

![Navigation workflow - Viewing all services that are offered and retrieving details about a specific service.](image1)

3.5.6 Example Workflows - The Volunteering App

Viewing new volunteering opportunities:

![Navigation workflow - Viewing new volunteering opportunities.](image2)
Searching for a specific volunteering opportunity around fundraising:

Figure 3.20: Navigation workflow - Searching for a specific volunteering opportunity around fundraising.
3.5.7 Example Workflows - The Contribute Posters App

Viewing a set of good and bad example poster photos:

Figure 3.21: Navigation workflow - Viewing a set of good and bad example poster photos.
Taking and uploading a picture as a contribution:

![Diagram of the navigation workflow for taking and uploading a picture as a contribution.]

Figure 3.22: Navigation workflow - Taking and uploading a picture as a contribution.
3.5.8 Example Workflows - App Settings

Disabling analytics tracking for the Events app:

Figure 3.23: Navigation workflow - Disabling analytics tracking for the Events app.

3.5.9 Example Workflows – Mobile-Age About Screen

Viewing information about the Mobile-Age application:

Figure 3.24: Navigation workflow - Viewing information about the Mobile-Age application.

3.6 Description of Screen

This section describes each of the individual screens (pages) within the demonstrator. Each screen’s description is accompanied with screenshots, a list of data service interactions used by that screen, and details of the generic front-end components that are utilised (developed in Task 4.2, reported in D4.1).

Below in Table 3.1 is an example of how the data service interactions are documented:
Table 3.1: Data interaction - Example data service interaction.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Succinct description of what this request is for.</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Element</td>
<td>The user interface element that this data request relates to.</td>
</tr>
<tr>
<td>Component</td>
<td>The type of user interface component it refers to; in particular, state which of, if any, the generic front-end components developed in Task 4.2 is being utilised (e.g. MobileAge.Component.List).</td>
</tr>
<tr>
<td>Description</td>
<td>A brief description of what happens during this data request, including user interactions.</td>
</tr>
<tr>
<td>Endpoint</td>
<td>/url/of/the/endpoint</td>
</tr>
<tr>
<td>Parameters</td>
<td>List the parameters that are used in the data interaction, if any.</td>
</tr>
<tr>
<td>Example Request</td>
<td>GET <a href="http://url/of/the/endpoint">http://url/of/the/endpoint</a></td>
</tr>
</tbody>
</table>

All of the screens within the demonstrator app report analytics to our analytics service. A description and example of an analytics data service interaction is described in Table 3.2 (section 3.6.1.3). To avoid duplication in this report, Table 3.2 is referenced (rather than repeated) in subsequent screen descriptions. All of the available analytics service requests have been documented and reported in deliverable D2.4.

3.6.1 Launcher

3.6.1.1 Description

The launcher is the first screen a user is presented with when opening the application. This screen is the hub of the application where a user can navigate to any of the internal apps.

This screen lists a series of buttons which navigate the user to a new screen for each respective app.
3.6.1.2 Screenshots

![Launcher screen](image)

*Figure 3.25: The Launcher screen.*

3.6.1.3 Data Service Interactions

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Succinct description of what this request is for.</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Element</td>
<td>-</td>
</tr>
<tr>
<td>Component</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>The analytics library collects app usage data in the background. This includes pageviews, custom events (e.g. button clicks), and location (with reduced precision).</td>
</tr>
<tr>
<td>Endpoint</td>
<td>/api/</td>
</tr>
<tr>
<td>Parameters</td>
<td>There are numerous combinations of parameters for pushing analytics data to the server, both on this screen and throughout the demonstrator app. These have previously been reported in deliverable D2.4, but up-to-date documentation can be found online here: <a href="https://analytics.scc-mobileage.lancs.ac.uk/doc/rest/#api-Analytics_Event-PushAnalytics">https://analytics.scc-mobileage.lancs.ac.uk/doc/rest/#api-Analytics_Event-PushAnalytics</a></td>
</tr>
</tbody>
</table>
Note that analytics are collected on all screens within the demonstrator application. To avoid duplication in this report, Table 3.2 will be referenced (rather than repeated) in subsequent screen descriptions. All of the available analytics service requests have been documented and reported in deliverable D2.4; up-to-date documentation is available online for the server’s RESTful endpoints\(^2\) and the JavaScript client library\(^3\) allowing developers to utilise the service.

### 3.6.2 Events – Homepage

#### 3.6.2.1 Description

The Events app is all about finding local social events. This screen is the entry point for the Events app, and lists various local events organised into useful groups. The event data is requested from our application back-end server.

Each group uses a carousel to display a subset of the available events (10 events chosen at random) for each given grouping (displaying the event name and category image); for example, the ‘Near home’ category shows any 10 random results that are near the user’s home (and are not subject to any additional sorting or filtering, such as distance). Swiping left on a carousel will show more events for that respective group, up to a maximum of 10 events. Once a user has swiped all the way to the end of the carousel (Figure 3.29), the last item in each carousel (stating “Find more >”) allows the user to navigate to a list of remaining available events.

Currently, the default event groups are:

- **Featured** – Events that are highlighted as featured by the data source (e.g. South Lakeland District Council website).
- **Today** – Any event that is on today.

\(^2\) [https://analytics.scc-mobileage.lancs.ac.uk/doc/rest/](https://analytics.scc-mobileage.lancs.ac.uk/doc/rest/)

\(^3\) [https://analytics.scc-mobileage.lancs.ac.uk/doc/js/](https://analytics.scc-mobileage.lancs.ac.uk/doc/js/)
Near home – Any event that is within a predefined radius from a preset home location. Group will not appear if a home location is not set (can be set within the Profile).

Nearby – Requests the device’s current location and performs a query on the server for any events within a predefined radius of the user’s current location.

This weekend – Any event that occurs on the coming weekend.

The ‘Featured’ events request first queries the server for featured event IDs (Table 3.3). If these events are already cached on-device, they are then read from local storage. If they are not cached, the event data is requested from the server by requesting data for specific IDs (Table 3.4).

The remaining groups of events on the homepage use a standard event query (Table 3.5), with parameters specific to the event grouping (e.g. distance parameters).

### 3.6.2.2 Screenshots

![Events app - Start screen.](image-url)

*Figure 3.26: Events app - Start screen.*
Figure 3.27: Events app - Start screen (carousels 2-3).

Figure 3.28: Events app - Start screen (carousels 4-5).
3.6.2.3 Data Service Interactions

All of the data service interactions on the events app start screen occur in the background, and do not require any user interaction except opening the app.

Table 3.3: Data interaction (Events app) - Get featured events IDs.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Get a list of featured event IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Element</td>
<td>The ‘Featured’ carousel at the top of the screen.</td>
</tr>
<tr>
<td>Component</td>
<td>Carousel (3rd party)</td>
</tr>
<tr>
<td>Description</td>
<td>The Events app requests featured events. It occurs in the background when the Events app is opened by the user. Request IDs of featured events from the event app’s server. If the event details are already cached on the device, the events are displayed on the carousel. If not cached, then a subsequent request for individual event details is made (Table 3.4).</td>
</tr>
<tr>
<td>Endpoint</td>
<td>/api/events/sources/groupings/</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
</tbody>
</table>
Table 3.4: Data interaction (Events app) - Get event details for specific event IDs.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Get details of specific events</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Element</td>
<td>The carousels on the start screen.</td>
</tr>
<tr>
<td>Component</td>
<td>Carousel (3rd party)</td>
</tr>
<tr>
<td>Description</td>
<td>This request is used if details of specific events are not cached on-device. Returns an array of event JSON data.</td>
</tr>
<tr>
<td>Endpoint</td>
<td>/api/events/</td>
</tr>
<tr>
<td>Parameters</td>
<td>sourceeventids:</td>
</tr>
<tr>
<td></td>
<td>A comma separated list of event IDs.</td>
</tr>
<tr>
<td>Example Request</td>
<td>GET <a href="https://events.scc-mobileage.lancs.ac.uk/api/events/?sourceeventids=28481,28451,28128,28490,28521">https://events.scc-mobileage.lancs.ac.uk/api/events/?sourceeventids=28481,28451,28128,28490,28521</a></td>
</tr>
</tbody>
</table>

Table 3.5: Data interaction (Events app) - Query for events.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Query for events</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Element</td>
<td>The carousels on the start screen.</td>
</tr>
<tr>
<td>Component</td>
<td>Carousel (3rd party)</td>
</tr>
<tr>
<td>Description</td>
<td>This request queries for events and filters the results. Used for queries to populate the carousels ‘Today’, ‘Nearby’, ‘Near home’, and ‘This weekend’; also used later in the search results screen (section 3.6.6).</td>
</tr>
<tr>
<td>Endpoint</td>
<td>/api/events/</td>
</tr>
<tr>
<td>Parameters</td>
<td>type:</td>
</tr>
<tr>
<td></td>
<td>[String.geo] -- Informs server to validate for other geographical parameters. A geospatial query on the demonstrator’s back-end is supported by a 2dsphere Index for fast geospatial queries; queries that use this index return ordered results according to the distance from the search point.</td>
</tr>
<tr>
<td></td>
<td>maxdist:</td>
</tr>
<tr>
<td></td>
<td>[Number] -- The maximum distance for the geospatial query.</td>
</tr>
</tbody>
</table>

4 https://docs.mongodb.com/manual/core/2dsphere/
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>distunit:</td>
<td>[String=m</td>
</tr>
<tr>
<td>lat:</td>
<td>[Number] -- The latitude of the geospatial centre point.</td>
</tr>
<tr>
<td>lng:</td>
<td>[Number] -- The longitude of the geospatial centre point.</td>
</tr>
<tr>
<td>isochrone:</td>
<td>[String] -- Restrict results by reachability (isochrones or equidistants). This parameter should be a stringified JSON object of parameters to pass to the OpenRouteService API (for example: locations, profile, range_type, and range). See section 4.1.2 for more details.</td>
</tr>
<tr>
<td>date:</td>
<td>[String] -- Query for events occurring on a specified date. Date strings formatted as YYYY-MM-DD (e.g. 2018-01-26).</td>
</tr>
<tr>
<td>start_date:</td>
<td>[String] -- The start of a date range to query for events; when this parameter is specified, the back-end ignores the contents of the ‘date’ parameter. Date strings are formatted as YYYY-MM-DD (e.g. 2018-01-26). Specified date must occur chronologically before the ‘end_date’.</td>
</tr>
<tr>
<td>end_date:</td>
<td>[String] -- The end of a date range to query for events; use instead of ‘date’ parameter. Date strings are formatted as YYYY-MM-DD (e.g. 2018-01-27). Specified date must occur chronologically after the ‘start_date’.</td>
</tr>
<tr>
<td>category:</td>
<td>[String] -- A pipe separated list (‘</td>
</tr>
<tr>
<td>txtsearch:</td>
<td>[String] -- Perform a text search for events. This query type is backed up by a Text Index which uses an event’s name and description field. Queries that use this index will sort the results by a computed relevance score.</td>
</tr>
<tr>
<td>sample:</td>
<td>[Number] Return the specified number of random results from an executed query. Useful for populating the start screen’s carousels with different events that meet the carousels criteria.</td>
</tr>
<tr>
<td>limit:</td>
<td>[Number] -- Sets a limit on the number of returned results. Takes the first N ordered results from a query. Order is determined by which database index is used (distance for geospatial queries, and relevance for text queries).</td>
</tr>
</tbody>
</table>

5 https://docs.mongodb.com/manual/core/index-text/
3.6.3 Events - Event View

3.6.3.1 Description

The ‘Event View’ screen displays all of the details about a given event. The user arrives at this screen after pressing on an event icon on the Events homepage (section 3.6.2), or after pressing on an event in the search results screen (section 3.6.6).

This screen displays key event information to the user at the top of the screen (Figure 3.30), followed by buttons listing key actions that a user may desire to perform (Figure 3.31). These actions include:

- **Add to calendar** – Adding the event to a calendar adds it to the device’s built-in calendar, enabling the user to utilise built-in calendar app features, but also to better integrate into a user’s existing services.
- **Show on map** – Showing the event on a map navigates the user to the Events app’s map view (section 3.6.4) and centres the map around the event’s location.
- **Find transport options** – Pressing this button opens the map view and finds a travel route between the device’s current location and the event location.

Further event details can be explored by scrolling the screen down and opening one of the collapsible elements:

- **When & Where** – This collapsible element contains more details on the location, date, time, admission price, and the weather forecast for the event.
- **Contact** – This collapsible element contains details about the event’s contact person/entity, including their name, location (which can also be opened on a map), telephone, email, website, and social media (where available).
- **Accessibility** – When accessibility information is available it is displayed within this collapsible. Currently this is limited to sunset times, which could affect a user’s decision to attend an event to due to daylight visibility or not wanting to drive at night. The current data sources do not have accessibility information about event venues, however other accessibility data sources can be integrated in the future.

Pressing on buttons for websites, email addresses, or telephone numbers fires the appropriate respective intent, e.g. when pressing the button with a phone number on, the app opens the phone’s built-in telephone app and prepares the phone number for ringing.
3.6.3.2 Screenshots

**Mandala- Embroidery Workshop**

*Date:* Saturday 29 September 2018  
*Time:* 10am - 3pm

Mandala is an abstract design with one identifiable centre point, from which emanates an array of shapes and forms. Aimed at both beginners and experienced embroiderers. This workshop provides you with the six basic stitches used within the Quaker Tapestry: Stem, Split, Chain, Quaker, Bayeux Point and Peking Knot. In a relaxed and friendly environment our tutors will guide you through the stitches and help you work towards an embroidered picture to finish at home.

**Actions**

*Figure 3.30: Events app - Event description.*

workshop provides you with the six basic stitches used within the Quaker Tapestry: Stem, Split, Chain, Quaker, Bayeux Point and Peking Knot. In a relaxed and friendly environment our tutors will guide you through the stitches and help you work towards an embroidered picture to finish at home.

**Actions**

*Figure 3.31: Events app - Event actions.*
Figure 3.32: Events app - Event details (1 of 4).

Figure 3.33: Events app - Event details (2 of 4).
Figure 3.34: Events app - Event details (3 of 4).

Figure 3.35: Events app - Event details (4 of 4).
3.6.3.3 Data Service Interactions

Table 3.6: Data interaction (Event view) - Get an event’s weather forecast.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Get details of weather at the event</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Element</td>
<td>-</td>
</tr>
<tr>
<td>Component</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>If the event is within the forecast period (5-days), then the weather is requested for the event at the given location, date, and time. A weather icon is displayed under ‘When &amp; Where’.</td>
</tr>
<tr>
<td>Endpoint</td>
<td>/api/events/:EventID/weather/forecast</td>
</tr>
<tr>
<td>Parameters</td>
<td>-</td>
</tr>
<tr>
<td>Example Request</td>
<td>GET <a href="http://events.scc-mobileage.lancs.ac.uk/api/events/5a1ee15a85ae3a5a2149c47f/weather/forecast">http://events.scc-mobileage.lancs.ac.uk/api/events/5a1ee15a85ae3a5a2149c47f/weather/forecast</a></td>
</tr>
</tbody>
</table>

3.6.4 Events - Event Map View

3.6.4.1 Description

The event map view’s primary purpose is to show the location of an event using Mobile-Age’s generic map mobile front-end component. This map loads map tiles which have been custom-rendered to provide better contrast between features presented on the map tile images (see deliverable D4.1 for more information). This screen is navigated to from the event view screen (section 3.6.3).

At the top of the screen is the map component (Figure 3.36), which centres on the event’s location. The map can be interacted with and holds the event address information within the event map pin (Figure 3.38). The map also shows other amenities and facilities, including toilets and benches - these can be important as part of journey planning. Scrolling down the screen you can see the event address again, and then a list of map options (Figure 3.39). This set of flip switch elements enable and disable their corresponding features on the map; e.g. turning off ‘Toilets’ hides the toilets on the map component.

If the map view screen is loaded based on a query that uses the user’s location, for example finding events near home, then the map will automatically calculate a route between the event and the user’s home location (Figure 3.37). This also works for event queries using the user’s current location (GPS).
Lastly, there is a convenience option to open the event location in a separate external map application installed on the user’s device. This is only intended to provide the user with choice of map application, in case they have an existing application preference.

3.6.4.2 Screenshots

Figure 3.36: Events app - Event location.
**Figure 3.37: Events app - Event location and journey.**

**Figure 3.38: Events app - Map interactions (event details).**
3.6.4.3 Data Service Interactions

Table 3.7: Data interaction (Event map) - Get data from Overpass service, showing amenities on a map.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Get coordinates of different amenity types like e.g. toilets or benches</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Element</td>
<td>The map</td>
</tr>
<tr>
<td>Component</td>
<td>MobileAge.Component.Map including: MobileAge.Layer.Marker</td>
</tr>
<tr>
<td>Description</td>
<td>Show markers for OpenStreetMap entities, using the Overpass api. The service is fetched by the MobileAge.Layer.Marker component itself. When a user presses one of the Flip Switches under the map, it will request data on the corresponding amenities to be added to the map (e.g. toilets or benches).</td>
</tr>
<tr>
<td>Endpoint</td>
<td>/overpass/interpreter</td>
</tr>
<tr>
<td>Parameters</td>
<td>The overpass api interpreter has a special syntax (Overpass query language [4]) facilitating complex queries. In the example below the parameters are:</td>
</tr>
</tbody>
</table>
(node|way|relation) ['amenity' = 'toilets']:
The datatype to search for with a given tuple of key / value pairs. Here 'amenity' = 'toilets'.

(North, West, South, East):
A box defining the area, in which the search is performed.

Example Request
GET http://geo.mobile-age.eu/overpass/interpreter?
data=[out:json][timeout:25];(node[amenity%20=%20%27toilets%27%27];);out%20body;%3E;out%20skel%20qt;

Table 3.8: Data interaction (Event map) - Get data from OpenRouteService, showing a route on a map.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Get route between two locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Element</td>
<td>The map</td>
</tr>
<tr>
<td>Component</td>
<td>MobileAge.Component.Map</td>
</tr>
<tr>
<td></td>
<td>including:</td>
</tr>
<tr>
<td></td>
<td>MobileAge.Layer.Path</td>
</tr>
<tr>
<td>Description</td>
<td>Show a route on the map between two locations. If this event's map was accessed by a search using the user’s current location or home coordinates, then it assumes those as a starting point for a route - the end is of course the event location.</td>
</tr>
<tr>
<td></td>
<td>The route is calculated using OpenRouteService. This service is fetched by the MobileAge.Layer.Path component itself.</td>
</tr>
<tr>
<td>Endpoint</td>
<td>/ors/routes</td>
</tr>
<tr>
<td>Parameters</td>
<td>This data interaction uses OpenRouteService’s routing API [5].</td>
</tr>
<tr>
<td></td>
<td>profile:</td>
</tr>
<tr>
<td></td>
<td>The profile to calculate the available route (e.g. foot-walking or driving-car)</td>
</tr>
<tr>
<td></td>
<td>coordinates:</td>
</tr>
<tr>
<td></td>
<td>A list of pipe (‘</td>
</tr>
<tr>
<td></td>
<td>geometry_format:</td>
</tr>
<tr>
<td></td>
<td>The format of the result data. The map component requires GeoJSON.</td>
</tr>
</tbody>
</table>
3.6.5 Events – Search

3.6.5.1 Description

The Events app search screen is accessible by pressing on the “Search” button at the top of any screen within the Events app. This search screen presents the user with a list of predefined searches (Figure 3.40 and Figure 3.41). Pressing on any of these listed searches selects it and expands to show additional information; for example, ‘Within walking distance of home’ is reasonably self-explanatory, but it clarifies that it means 30 minutes walking time of the user’s home location.

Scrolling to the bottom of the screen, there is a “Customise Search” button (Figure 3.41). This expands to show all of the available options that can be changed when querying the Events app back-end. The options available for a custom query are:

- Keyword text search (Figure 3.42).
- Event category selection (Figure 3.43 and Figure 3.44).
- Location, whether to search around the user’s home or current location (Figure 3.45).
- Travel preferences (Figure 3.46 - Figure 3.49).
- Date (Figure 3.50 - Figure 3.52).

The keyword text search allows a user to query for keywords within events name and description fields. The category selection allows further filtering by specific event categories.

All queries to the back-end include a geospatial query component, to ensure relevance of events for a user. The location field specifies the centre point for geospatial query (i.e. home or device’s current location).

The travel preferences allow a user to specify the method of travel to use during the events query (including car, walking, or cycling). If no method of travel is selected (i.e. ‘unspecified’), then a standard radial geospatial query is performed - the distance of the radius can be specified by the user. Alternatively, if the user specifies a method of travel (e.g. car) then an additional query to the OpenRouteService is made; this will create an isochrone or equidistant (a geographical polygon, based on reachability from a central point) which is then used Events app back-end to perform a geospatial query (instead of a simple radius) for events within that polygon (see section 4.1.2 for further details on this service). Along with the method of travel, the user can specify the maximum distance or maximum travel time.

Dates can be specified as a single date (Figure 3.50), or a range of dates (Figure 3.51). Pressing on the input field to change the date value presents the user with a date selection user interface overlay (Figure 3.52).

Upon pressing the search button, the query parameters are passed to the search results screen, which performs the query and then displays the results.
3.6.5.2 Screenshots

**Figure 3.40: Events app - Predefined searches ('This month').**

**Figure 3.41: Events app - Predefined searches ('walking distance of home').**
Figure 3.42: Events app - Customise search keywords.

Figure 3.43: Events app - Customise search for all categories.
Figure 3.44: Events app - Customise search for individual categories.

Figure 3.45: Events app - Customise search by location.
**Figure 3.46:** Events app - Customise search's travel preference (unspecified).

**Figure 3.47:** Events app - Customise search's travel preference options.
Figure 3.48: Events app – Customise search travel preference (car reachability).

Figure 3.49: Events app - Customise search travel preference (reachability options).
Figure 3.50: Events app - Customise search by date.

Figure 3.51: Events app - Customise search by date range.
3.6.5.3 Data Service Interactions

No data services, except analytics, are requested.

3.6.6 Events - Search Results

3.6.6.1 Description

The Search Results screen is displayed after the “Search” button is pressed on the Search screen. The query is executed on the Events app back-end, and the results are returned and listed on this screen (Figure 3.53).

Each search result presents the event’s name, date and time, rough location, and distance. When a distance value is available, the results are also sorted by distance from the queries centre point. Pressing on an event in the search results list will open that event in the Event View screen (section 3.6.3).

There is also a filter text field at the bottom of the screen, which will filter the results list with the text entered into the filter field (Figure 3.54). The filter text also filters events by text in the event’s description, despite that text not being shown.
3.6.6.2 Screenshots

Figure 3.53: Events app – Search results for nearby events (in Kendal, South Lakeland).

Figure 3.54: Events app – On-device keyword filtering of search results.
### 3.6.6.3 Data Service Interactions

*Table 3.9: Data interaction (Search results) – Request an isochrone.*

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Get reachability from a Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Element</td>
<td>-</td>
</tr>
<tr>
<td>Component</td>
<td>-</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>If a user specifies transport preferences in the search screen, then an additional query to OpenRouteService’s isochrone (reachability) service is made. This generates a geographical polygon which plots someone’s reachability (calculating how far the user could travel in a set time or distance in any direction). This polygon is then used to query a subset of events in the database – which is more accurate than a simple radius search.</td>
</tr>
<tr>
<td><strong>Endpoint</strong></td>
<td>/ors/isochrones</td>
</tr>
</tbody>
</table>
| **Parameters** | This data interaction uses OpenRouteService’s isochrones API [5].  
*profile:*  
The profile to calculate the reachability (e.g. foot-walking or driving-car)  
*locations:*  
A list of pipe (‘|’) separated coordinate pairs (lon,lat), specifying the coordinate to visit in order.  
*range_type:*  
Calculate reachability based on ‘time’ or ‘distance’.  
*range:*  
The maximum range; seconds for time and meters for distance. |
| **Example Request** | GET http://geo.mobile-age.eu/ors/isochrones?profile=foot-walking&locations=8.8845,53.0555&range_type=time&range=60 |

This screen uses the same endpoint to query for events as reported earlier for the screen “Events - Homepage” (section 3.6.2.3), see Table 3.5.

### 3.6.7 Services

#### 3.6.7.1 Description

The Services app presents the user with services that are specific to older adults. The services are listed in a set of collapsible elements (Figure 3.55). When selected, they expand to show
further details about the respective service (Figure 3.56 - Figure 3.58), including contact or location details.

There is also a filter text field at the bottom of the screen, which will filter the services list with the text entered into the filter field (Figure 3.59). It filters by the content of the collapsible element, not just by the collapsible title text.

3.6.7.2 Screenshots

![Services app - Landing screen.](image)

*Figure 3.55: Services app - Landing screen.*
Age UK South Lakeland

Staying independent is all about receiving advice and practical support so that you or someone you care for can keep living well at home.

Who is this for?

Anybody over 50 or their carer(s).

What is available?

A complete, confidential service tailored to your needs. You’ll be able to talk about anything that’s bothering you. An experienced officer will then make sure you have the right information and advice. Remember that we’ll often be able to identify issues you may not have considered.

Usually we’ll be able to deal with everything from within Age UK South Lakeland. Or we might agree that it’s best for us to get in touch with partnership organizations and make sure they call you.

There are lots of possibilities for advice and support:

- Improving your income
- Moving into nursing and residential care
- Coping with an illness or disability
- Making your home safe and warm
- Meeting other people

Filter by keyword...
Where and when?

This is to suit you. It may be that a phone call is all that’s needed. But usually a home visit is the best way for us to work with you. This would be arranged at a time during the day that you find convenient.

How do I find out more?

Contact our helpline and ask about staying independent:

030 300 30003
helpline@ageuksouthlakeland.org.uk

Filter by keyword...

Figure 3.58: Services app - Example service (scrolled viewport; 3 of 3).

Figure 3.59: Services app - Filtering listed services.
3.6.7.3 Data Service Interactions

No data services, except analytics, are currently requested. This is because the list of available services is hard-coded within the prototype; we are negotiating access to data from AgeUK South Lakeland, which we will access through our back-end.

3.6.8 Volunteering

3.6.8.1 Description

The Volunteering app presents the user with volunteering opportunities that are specific to older adults. The opportunities are listed in a set of collapsible elements (Figure 3.60). When selected, they expand to show further details about the respective opportunity (Figure 3.61 - Figure 3.62), including contact or location details.

There is also a filter text field at the bottom of the screen, which will filter the volunteering opportunities list with the text entered into the filter field (Figure 3.63). It filters by the content of the collapsible element, not just by the collapsible title text.

3.6.8.2 Screenshots

Figure 3.60: Volunteering app - Landing screen.
Exercise Class Volunteer Leader

Role Title: Exercise Class Volunteer Leader

Role Description: The role involves teaching a series of seated exercises to local older people, helping them keep fit and active. Full training and support provided. Out of pocket expenses paid.

Application Process: Please phone or email for more details and make an appointment for an informal chat about the role.

Expires: 31/01/2019

Figure 3.61: Volunteering app - Volunteering opportunity details.

Organisation: Age UK South Lakeland
Website: [Website]

Contact Details

Contact Name: Charlotte Robinson
Contact Telephone: 01539 728118
Contact Email: volunteering@ageuksouthlakeland.org.uk

Figure 3.62: Volunteering app - Volunteering opportunity details (continued).
3.6.8.3 Data Service Interactions

No data services, except analytics, are currently requested. This is because the list of available volunteering opportunities is hard-coded within the prototype; we are negotiating access to data from AgeUK South Lakeland, which we will access through our back-end.

3.6.9 Contribute Poster

3.6.9.1 Description

The contribute poster application allows a user to take a picture of a newsletter or poster, which gets uploaded to our server for future processing. The photo is uploaded along with a set of location coordinates.

The purpose of this is to collect a record of local posters and newsletters. These can be processed in the future to extract local event information. This supports the notion of enabling the users to contribute data.

Pressing the first button “Take picture” (Figure 3.64) opens the device’s camera app, allowing the user to take a picture. Once they have taken the picture, they are taken back to this app’s screen (Figure 3.65) where they are presented with a message informing them the photo has been successfully uploaded, a message thanking them for their contribution, and a button
that says, “Send another picture”. This performs the same action as before, and opens the device’s camera app. A typical workflow is presented in section 3.5.7.

Pressing the button “Example poster pictures” at the bottom of the screen navigates the user to a new screen with examples and descriptions of what makes a good or bad photo to upload.

### 3.6.9.2 Screenshots

![Screenshot of the Contribute Poster app](image)

**What is this?**

We would like to improve our knowledge of local events and services, and we need your help!

Sending us pictures of posters and local newsletters related to local events and services will help us provide a better service for all.

Simply press the button below to take a picture of a poster, and it will be sent to us!

![Take picture button](image)

**What makes a good picture?**

What does a good poster image look like? View some good and bad example pictures of posters here:

![Example poster pictures](image)

*Figure 3.64: Contribute Poster app.*
3.6.9.3 Data Service Interactions

*Table 3.10: Data interaction (Contribute Poster app) - Upload a photo.*

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Push photo of poster/newsletter to server</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Element</td>
<td>Button ‘Take picture’</td>
</tr>
<tr>
<td>Component</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>Uploads a photo of a newsletter or poster to the server. Sends location coordinates along with the uploaded image. Uses content-type: multipart/form-data.</td>
</tr>
<tr>
<td>Endpoint</td>
<td>/api/upload/</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>file:</td>
</tr>
<tr>
<td>The image file.</td>
</tr>
<tr>
<td>lat:</td>
</tr>
<tr>
<td>The latitude of the photo upload location.</td>
</tr>
<tr>
<td>lon:</td>
</tr>
<tr>
<td>The longitude of the photo upload location.</td>
</tr>
</tbody>
</table>
3.6.10 Contribute Poster - Example photos

3.6.10.1 Description

This screen provides the user with an example of a good photo of a poster (Figure 3.66), with a succinct and clear description of why it is a good photo.

Scrolling down the screen (Figure 3.67) there is another example photo of a poster. This one shows and describes what a bad and unusable photo looks like.

3.6.10.2 Screenshots

Example of a good poster picture:

This poster fills the picture, has clear text, and has good lighting.

Figure 3.66: Contribute Poster app - A user’s example of a good photo.
3.6.10.3 Data Service Interactions

No data services, except analytics, are requested.

3.6.11 Profile

3.6.11.1 Description

The Profile screen contains personal preferences and information that is used in this demonstrator application.

This screen separates preferences into logical groupings (Figure 3.68): Locations, Travel and Mobility, and Financial. These can be used across apps and services, for example the travel preferences can be taking into consideration when searching and filtering events in the events app.

Pressing on “Locations” expands the locations preferences (Figure 3.69). The user can save their home location by pressing the button “Update with current location”, which gets their current location and saves it as their Home location. Once saved it will display the home location above that button.

Another feature of the profile is the ability to share it with other people should the user decide to do so. Pressing the “Share profile” button currently compiles the user profile data into an email.
3.6.11.2 Screenshots

Figure 3.68: Profile app.

Here you can adjust your personal profile preferences.

Need to share this information? Press this button:

- Locations
- Travel and Mobility
- Financial preferences

Figure 3.69: Profile app - Expanded locations section.

These are locations that you have saved to make location-based searches quicker.
**Figure 3.70:** Profile app - Expanded locations section (continued).

**Figure 3.71:** Profile app - Expanded ‘Travel and Mobility’ section (1 of 3).
Figure 3.72: Profile app - Expanded ‘Travel and Mobility’ section (2 of 3).

Figure 3.73: Profile app - Expanded ‘Travel and Mobility’ section (3 of 3).
3.6.11.3 Data Service Interactions

Table 3.11: Data interaction (Profile app) - Get auto-complete list of location suggestions.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Retrieve an auto-completed list of location suggestions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UI Element</strong></td>
<td>The input field under ‘Other saved locations’ labelled ‘City/Town/Place’.</td>
</tr>
<tr>
<td><strong>Component</strong></td>
<td>MobileAge.Component.AutoComplete</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>This input field will suggest location names as the user types. Making text entry easier.</td>
</tr>
<tr>
<td><strong>Endpoint</strong></td>
<td>/frontend/php/data.placesUK.php</td>
</tr>
<tr>
<td><strong>Parameters</strong></td>
<td>places: Query for place names that contain the text passed in this parameter. The text typed by the user into the input field will be sent as this parameter.</td>
</tr>
<tr>
<td><strong>Example Request</strong></td>
<td>GET <a href="https://mobileage.ftb-esv.de/frontend/php/data.placesUK.php">https://mobileage.ftb-esv.de/frontend/php/data.placesUK.php</a></td>
</tr>
</tbody>
</table>
3.6.12 Settings

3.6.12.1 Description

The Settings slide-in screen is accessible from the launcher screen. The settings panel slides in from the left and exposes overall application settings for the user to change.

The analytics ‘Preferences’ button on this settings panel allows a user to manage the analytics collection occurring on their current device. Pressing this button navigates the user to the Analytics Management screen.

3.6.12.2 Screenshots

![Settings app](image)

*Figure 3.75: Settings app.*

3.6.13 Settings - Analytics Management

3.6.13.1 Description

This screen allows a user to enable or disable analytics on their device. The screen lists all Mobile-Age analytics tracking IDs that have been activated on the user’s device. Typically, one tracking ID will relate to one application, but this is at the discretion of the application.
The screen requests user-friendly information about each tracking ID (name and
description) from the analytics service. The user can then choose to enable or disable
collection of analytics for each tracking ID separately.

### Screenshots

![Analytics management screen.]

**Figure 3.76: Settings app – Analytics management screen.**

### Data Service Interactions

**Table 3.12: Data interaction (Settings app) - Request analytics details.**

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Request metadata about an analytics tracking IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI Element</td>
<td>-</td>
</tr>
<tr>
<td>Component</td>
<td>-</td>
</tr>
<tr>
<td>Description</td>
<td>Get information about a set of tracking IDs, for example its name and a description of what it is used for.</td>
</tr>
<tr>
<td>Endpoint</td>
<td>/api/tid-info/</td>
</tr>
<tr>
<td>Parameters</td>
<td><code>tids[]:</code></td>
</tr>
<tr>
<td></td>
<td>A Tracking ID to request information about. Each instance of this parameter in the URL is placed into an array on the back-end.</td>
</tr>
</tbody>
</table>
3.6.14 About

3.6.14.1 Description

This is the application’s About screen. It contains a brief description of the application, and a list of necessary licenses for libraries that are used within the application.

3.6.14.2 Screenshots

![About app](image)

Mobile Age is an application targeted at older adults, aiming to encourage independent living by reducing loneliness and social isolation.

This application provides a range of services, which include: helping you find local social events, and finding volunteering opportunities and services. This is all done whilst taking you and your preferences into consideration.

Sunrise/sunset calculations:
- `suncalc.js`: [BSD 2-clause "Simplified" License](https://www.bsd.org/licenses/bsd-simplified)

Figure 3.77: About app.

3.6.14.3 Data Service Interactions

No data services, except analytics, are requested.
4 Data Considerations

4.1 Data Sources

4.1.1 Social Events

The initial prototype demonstrator uses web-scraped data from the South Lakeland District Council website\(^6\). Web scraping events’ data has its limitations and has therefore only been considered a short-term solution during a prototype phase of the demonstrator application; changes to the source website could break the automated event collation process. Hence, we are working with Age UK South Lakeland and in the process of migrating to their source of event data on the ‘Compass in Cumbria eHub’ website\(^7\).

See section 4.3 Data Processing for more details on the specific challenges of handling web-scraped data from the SLDC website.

4.1.2 Routing Service

To support location-based searches and location guidance to events selected by the user, we use the OpenRouteService (ORS)\(^8\). It offers routing and routing-related services built on top of geographic data from OpenStreetMap. The services offered by OPS that are used in this demonstrator application include: \textit{geocoding}, \textit{routing}, and \textit{reachability} (isochrones and equidistants).

\textit{Geocoding}

The Events app supports geospatial queries and can sort queries by distance to the search centre point. As such, the server requires location coordinates for all of the events stored in the database. If a data source does not provide coordinates, because the data is scraped from a website for example, then the event data requires post-processing to get the coordinates.

OpenRouteService offers a geocoding endpoint in its API. Geocoding is a method of converting textual addresses into geographical coordinates. If geographical coordinates are not available when the event data is sourced, the Events app back-end will geocode the event’s address using this service.

\textit{Table 4.1: Data interaction (Back-end) - Geocoding during back-end processing of data.}

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Geocode a location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Convert a location address or name into geographic coordinates, using the OpenRouteService API [5].</td>
</tr>
</tbody>
</table>

\(^6\) \url{http://applications.southlakeland.gov.uk/eventsearch/}
\(^7\) \url{https://www.compassinhub.com/}
\(^8\) \url{https://openrouteservice.org/}
As geocoding only occurs in the back-end, it is not reported earlier in the description of screens (section 3.6). Table 4.1 describes the endpoint used by the Events app back-end.

**Travel route**

OpenRouteService’s primary purpose is for calculating routes between two or more points. The service provides route calculations for multiple profiles, including car, walking, and cycling. An endpoint description is provided earlier in Table 3.8.

This open service can be used to find routes that cater for users’ mobility limitations.

**Reachability (Isochrones and Equidistants)**

OpenRouteService provides a reachability API which can calculate isochrones (reachability based on travel time) and equidistants (reachability based on travel distance). Isochrones and equidistants are polygons that plot a distance away from a central point based on a constant time or distance measure. Examples of these polygons are visualised by OpenRouteService in Figure 4.1 and Figure 4.2, and an endpoint description is given in Table 3.9.

These polygons can be used in database geospatial queries to return results (in this case social events) that reside within those polygons – e.g. events within walking distance.
One study of the UK older population found that, in older adults over 65 years of age, the “mean walking speed was 0.9 m/s in men and 0.8 m/s in women” [6]. A routing service that considers walking reachability should therefore consider different walking speeds – something that OpenRouteService provides.

Figure 3.2 demonstrates two walking distance isochrones with different walking speeds. The outer red isochrone shows distances that can be reached with 30 minutes of walking using the services default setting of 6 km/h (approximately 3.7 mile/h). The inner green isochrone shows 30 minutes again, but with a reduced maximum walking speed of 3 km/h (approximately 1.9 mile/h).

### 4.1.3 Weather

Information on the weather at an event is important to the South Lakeland co-creators, and forms an important part of their decision-making process for whether they will attend.

Open Weather Map⁹ is used to get weather information either for now, or over a 5-day forecast (for reporting weather at the date and time of an event). An endpoint description for a wrapper to Open Weather Map is given in Table 3.6, during the screen description for Event View.

### 4.1.4 Nearby facilities and Amenities

The Overpass API [4] is a read-only optimised API that provides a query language to access OpenStreetMap (OSM) data, such as amenities and facilities (and other OSM data). An instance of this API, and a clone of the OSM data, is deployed on a server controlled by FTB¹⁰. This service is used, for example, to get information on nearby toilets and benches; this

---

⁹ [https://openweathermap.org/](https://openweathermap.org/)
information may be important to some older adults, as it can affect their decision to take particular routes or attend certain events.

An endpoint description for Overpass is given during the screen description for Event Map View, in Table 3.7.

### 4.2 Data Processing

The web-scraper retrieves its data from the South Lakeland District council website, and periodically checks for updates to the data. The scraped textual data is then parsed into corresponding data types (e.g. Date objects) and post-processed, structuring event data into a more readily queryable format. An example of post-processed data is converting textual address locations (from the source website) into location coordinates, using OpenRouteService’s geocoding capability.

The processed event data is then stored in the MongoDB database on the demonstrator’s server. The schema is shown in Table 4.2.

**Table 4.2: Database schema for event data.**

<table>
<thead>
<tr>
<th>Key</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the event</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>The description of the event</td>
</tr>
<tr>
<td>category</td>
<td>String</td>
<td>The category assigned to the event</td>
</tr>
<tr>
<td>details.rawDate</td>
<td>String</td>
<td>The raw scraped text representing the event’s date(s). May represent a single start date, a range, or multiple ranges. Requires processing.</td>
</tr>
<tr>
<td>details.date</td>
<td>[Object]</td>
<td>An array of objects representing date ranges (with ‘start’ and ‘end’ keys).</td>
</tr>
<tr>
<td>details.date.start</td>
<td>Date</td>
<td>The start of a date range.</td>
</tr>
<tr>
<td>details.date.end</td>
<td>Date</td>
<td>The end of a date range (setting equal to start date indicates event occurs on a single date).</td>
</tr>
<tr>
<td>details.rawTime</td>
<td>String</td>
<td>The raw scraped text representing the event’s time. May represent a single start time, a range, or multiple ranges. Requires processing. May contain additional data that would be dropped during parsing (e.g. textual info such as “doors close 10 minutes after start”).</td>
</tr>
<tr>
<td>details.time</td>
<td>[Date]</td>
<td>An array of Date objects, to represent a single or range of times. The date component of the Date object is</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>details.loc</td>
<td>Object</td>
<td>The location of the event, in GeoJSON format. Contains additional metadata. GeoJSON objects can be used by database indexes for performant geo-queries.</td>
</tr>
<tr>
<td>details.loc.str</td>
<td>String</td>
<td>The unprocessed textual address. May contain additional information that could be lost during geocoding (e.g. “behind the car park”).</td>
</tr>
<tr>
<td>details.loc.strShort</td>
<td>String</td>
<td>A calculated location name retrieved during geocoding (e.g. town or neighbourhood name).</td>
</tr>
<tr>
<td>details.loc.geocoded</td>
<td>Boolean</td>
<td>Whether this event has been successfully geocoded.</td>
</tr>
<tr>
<td>details.loc.geocodeConfidence</td>
<td>Number</td>
<td>Remember the geocoding service’s confidence value in the returned result.</td>
</tr>
<tr>
<td>details.loc.type</td>
<td>String</td>
<td>The GeoJson type of the ‘details.loc’ field - defaults to ‘Point’.</td>
</tr>
<tr>
<td>details.loc.coordinates</td>
<td>[Number]</td>
<td>An array of location coordinate: [&lt;longitude&gt;, &lt;latitude&gt;]</td>
</tr>
<tr>
<td>details.admissionprice</td>
<td>String</td>
<td>The price of admission to an event. NB: Scraped data often contains complex instructions that are not readily parsable, hence String type.</td>
</tr>
<tr>
<td>details.registrationrequired</td>
<td>Boolean</td>
<td>Whether registration is required for the event.</td>
</tr>
<tr>
<td>contact.name</td>
<td>String</td>
<td>The name of the person or entity to contact regarding the event.</td>
</tr>
<tr>
<td>contact.loc.str</td>
<td>String</td>
<td>The unprocessed textual address.</td>
</tr>
<tr>
<td>contact.loc.coordinates</td>
<td>[Number]</td>
<td>Geocoded coordinates of a contact’s location. Is not a GeoJSON object as database is limited to only using one field for a coordinate index.</td>
</tr>
<tr>
<td>contact.loc.geocoded</td>
<td>Boolean</td>
<td>Whether the contact’s address was successfully geocoded.</td>
</tr>
<tr>
<td>contact.loc.geocodeConfidence</td>
<td>Number</td>
<td>Remember the geocoding service’s confidence value in the returned result.</td>
</tr>
<tr>
<td>contact.telephone</td>
<td>String</td>
<td>The contact’s telephone number</td>
</tr>
<tr>
<td>contact.email</td>
<td>String</td>
<td>The contact’s email address</td>
</tr>
<tr>
<td>contact.website</td>
<td>String</td>
<td>The contact’s website</td>
</tr>
</tbody>
</table>
### 4.3 Data Creation and Sharing

In response to the previously stated limited open data for events (deliverable D4.1, section 8.1.1), we have created an app within the prototype demonstrator to enable older adult users to contribute a poster or newsletter to the project (see section 3.6.9, ‘Contribute Poster’). It is a simple application that allows users to upload a picture (with accompanying location coordinates). Currently the images are not processed and just reside on the server, but we intend to process the image data using Optical Character Recognition to automatically pull event data from the images.

| contact.socialmedia.facebook | String | The contact’s Facebook account name |
| contact.socialmedia.twitter  | String | The contact’s Twitter account name  |
| source.eventid              | String | The ID for this event given by the data source. |
| source.url                  | String | The web address where this event was sourced from. |
| meta.postProcessingRequired | Boolean| Whether this event’s entry requires post-processing (e.g. raw textual locations require geocoding). |
| meta.expiry_date            | Date   | The event expires and is deleted from the database 30 days after the end of the event. |

A limitation of processing this source of web-scraped data is that the geocoding service is unable to resolve some addresses. In particular, an event address sometimes includes non-address information, such as “behind the car park”. It is this additional text that the geocoding service is unable to resolve.

Another known limitation of the scraped web-data is how the date and time are presented to the user on the SLDC website. There have been numerous instances of events that occur on specific recurring dates being presented to the user (and therefore the web scraper) with fractured information across the date field and event description. For example, one event’s description would contain the text “We meet on the second and fourth Tuesday of each month”, and the date field would only state “From Tuesday 2 January 2018 to Monday 31 December 2018”. It is common on the SLDC website to only show the date range of the event, and not the individual occurrences of an event. To resolve this whilst still web-scrapping the data would require complex processing of the event description text to reliably determine additional date or time data (as it is written by a person, and not presented in a formulaic or easily parsable fashion). The effort and complexity is beyond the scope of the prototype. This will be resolved once we migrate to AgeUK’s events data source on the Compass in Cumbria website (see section 4.1.1 ‘Social Events’ for details on the newer events data source).
The uploaded images are stored on the demonstrator’s server filesystem, and the file’s metadata (including the path to the file) is stored within the MongoDB database. The schema is shown in Table 4.3.

<table>
<thead>
<tr>
<th>Key</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The file’s name.</td>
</tr>
<tr>
<td>path</td>
<td>String</td>
<td>The file’s path on the local filesystem.</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>The file’s Mime-Type.</td>
</tr>
<tr>
<td>size</td>
<td>Number</td>
<td>The size of the file in Bytes.</td>
</tr>
<tr>
<td>last_modified</td>
<td>Date</td>
<td>The last modified date of the file.</td>
</tr>
<tr>
<td>md5_hash</td>
<td>String</td>
<td>The MD5 hash of the file. Can be used to ensure duplicate files are not uploaded.</td>
</tr>
<tr>
<td>loc</td>
<td>Object</td>
<td>The location the image was uploaded from, using the standardised GeoJSON format.</td>
</tr>
<tr>
<td>loc.type</td>
<td>String</td>
<td>The GeoJSON type - defaults to ‘Point’</td>
</tr>
<tr>
<td>loc.coordinates</td>
<td>[Number]</td>
<td>An array of location coordinate: [longitude, latitude]</td>
</tr>
<tr>
<td>uploaded_on</td>
<td>Date</td>
<td>The date the file was uploaded on.</td>
</tr>
</tbody>
</table>

Data can also be shared in the Profile screen, as described in section 3.6.11. There a user can opt to email their user profile with another person.

Analytics data is another source of data created from the demonstrator application. An example usage of this analytics data is given in the scenario section (3.2.4), and further descriptions of the analytics data and usage scenarios are given in deliverable D2.4.
5 Implementation and Deployment

5.1 Implementation Details

The demonstrator application mobile front-end is implemented as a hybrid mobile web application using Apache Cordova\(^{11}\). The user interface is built using web technologies (HTML, CSS, JavaScript) and access to native capabilities are provided through Cordova Plugins. These plugins provide a JavaScript interface to features such as device sensors, GPS, notifications, and other operating system specific features. We use a hybrid application approach to maximise the number of devices that we support (both Android and iOS), whilst maintaining a single codebase. The use of Cordova also helps facilitate rapid prototyping as part of the co-creation process. The application was developed for mobile devices, but Cordova also provides a pathway to more general access via a web browser.

The user interface of the demonstrator application uses JQuery Mobile (version 1.5.0-alpha.1). JQuery Mobile is a touch-optimised HTML5-based user interface system for interfaces that are responsive across touch device and desktop screens.

The demonstrator back-end server is built using NodeJS\(^{12}\), utilising Express\(^{13}\) to create the RESTful endpoints. Query responses are given using JSON. The server persists its data in MongoDB\(^{14}\), a NoSQL document-based database.

\(^{11}\) [https://cordova.apache.org/](https://cordova.apache.org/)

\(^{12}\) [https://nodejs.org/](https://nodejs.org/)

\(^{13}\) [https://expressjs.com/](https://expressjs.com/)

\(^{14}\) [https://www.mongodb.com/](https://www.mongodb.com/)
Welcome to the Mobile-Age Demonstrator Applications for older adults at South Lakeland. With the application you will be able to find nearby events, services and volunteering opportunities in your local area. You will be also able to share events in your local area with the community using the Mobile-Age application. Using Mobile-Age, we guarantee full transparency regarding the collection and use of your data.

To install the application, please visit: https://scc-mobileage.lancs.ac.uk/app
5.3 Download Links and Demonstration

The release of the full version of the demonstrator application is planned for spring 2018. A prototype release is currently available to accompany this deliverable via our servers:

https://scc-mobileage.lancs.ac.uk/app/

Installing the app as a download from our servers requires additional steps: It can only be installed, if the user enables the "Allow installation of apps from unknown sources" options in the security dialog of the android settings menu. The user should disable the option immediately after installing the mobile-age app.

We have not made the prototype version available in the official Google or Apple app stores at the time of writing - this will change to coincide with the application’s release in spring 2018.
6 Conclusions

In this deliverable we have described the design and implementation of the South Lakeland Demonstrator. The application was co-created with older adults to address the issue of social isolation and provides users with an easy way to find information on local events, services and volunteering opportunities.

Key to the South Lakeland Demonstrator is the split between the core application framework that supports features such as profile sharing and analytics and the apps themselves that provide users with the functionality they require. This split enables us to easily expand the demonstrator in the face of new requirements.

The application has been designed to operate on a wide range of mobile devices and will be released through the relevant application stores in Spring 2018.
7 References


