

D4.3 Mobile Data Collection System

Citation for published version (APA):

Ternier, S., Suarez, A., Van der Vegt, W., Specht, M., Mikroyannidis, A., & Parodi, E. (2014). *D4.3 Mobile Data Collection System*.

Document status and date:

Published: 30/09/2014

Document Version:

Peer reviewed version

Document license:

CC BY-NC-SA

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

<https://www.ou.nl/taverne-agreement>

Take down policy

If you believe that this document breaches copyright please contact us at:

pure-support@ou.nl

providing details and we will investigate your claim.

Downloaded from <https://research.ou.nl/> on date: 02 Jul. 2022

Open Universiteit
www.ou.nl



weSPOT

IST (FP7/2007-2013) under grant agreement N° 318499

Deliverable number	<i>D4.3</i>
Deliverable Title	<i>Mobile Data Collection System</i>
Dissemination level	<i>PU</i>
Delivery date	<i>30 Sept 2014</i>
Status	<i>Final</i>
Author(s)	<i>Stefaan Ternier, Angel Suarez, Wim van der veegt, Marcus Specht</i>
Contributors	<i>Alexander Mikroyannidis, Elisabetta Parodi</i>

The weSPOT Consortium

Number	Partner	Country
1	OPEN UNIVERSITEIT NEDERLAND	Netherlands
2	THE OPEN UNIVERSITY	United Kingdom
3	TECHNISCHE UNIVERSITAET GRAZ	Austria
4	KU LEUVEN	Belgium
5	SOFIISKI UNVERSITET SVETI KLIMENT OHRIDSKI	Bulgaria
6	FRIEDRICH-ALEXANDER-UNIVERSITAT ERLANGEN NURNBERG	Germany
7	FOUNDATION FOR RESEARCH AND TECHNOLOGY HELLAS	Greece
8	EXACT LEARNING SOLUTIONS SPA	Italy
9	IPAK INSTITUT ZA SIMBOLNO ANALIZO IN RAZVOJ INFORMACIJSKIH TEHNOLOGIJ VELENJE ZAVOD	Slovenia

Document Control

Coordinating Editor: *Stefaan Ternier*

E-mail: *Stefaan.ternier@ou.nl*

Amendment History

Version	Date	Author/Editor	Reviewers	Description/Comments
1	25/07/2014	Stefaan Ternier		First Draft
2	01/08/2014	Angel Suarez		Contribution
3	20/09/2014	Stefaan Ternier		Final version before internal review
4	24/09/2014		Elisabetta Parodi	
5	27/09/2014		Alexander Mikroyannidis	
6	28/09/2015	Stefaan Ternier		Final version

Executive Abstract

The aim of D4.3 is to present the Mobile Data Collection System (MDCS). This system is used by various weSPOT components to manage the data that is collected in the third phase of the IBL model.

Table of Contents

weSPOT IST (FP7/2007-2013) under grant agreement N° 318499	1
Executive Abstract.....	5
Table of Contents	6
Mobile data collection system	7
Introduction.....	7
Mobile Data Collection Model	8
Architecture	9
API	11
Data collection task model.....	11
Method: Create data collection task.....	13
Method: Retrieve data collection task.....	13
Method: Create response.....	13
Method: Retrieve Response	15
Smartphone client (PIM)	17
Login.....	17
List Inquiries.....	18
Data collection tasks.....	18
Google Glass client (GPIM)	19
Inquiry Workflow Engine integration.....	20
IWE: display results	21
Data export	21
Conclusion	22
References.....	23

Mobile data collection system

Introduction

Data Collection is the third phase in the weSPOT IBL model. After operationalization of the research, the weSPOT user can start collecting data samples. Traditional data collection methods include paper-and-pencil capturing of data, suffering various weaknesses and risks to data integrity. This deliverable presents the mobile data collection system, a system that supports the user to record data and that feeds the data analysis phase of the IBL model.

The mobile data collection system makes it possible for various users to collect data for a shared inquiry and to easily give all user access to the joint data set. It builds on inexpensive off-the-shelf hardware to capture this data. Many students own a smartphone and can in a bring-your-own-device (BYOD) fashion use their mobile phone to collect data samples.

The source code for all code described in this deliverable is available at

<https://code.google.com/p/arlearn/>

This project contains several repositories including

- GPIM <https://code.google.com/p/arlearn/source/checkout?repo=gpim>
- PIM <https://code.google.com/p/arlearn/source/checkout?repo=pim>
- The cloud component: <https://code.google.com/p/arlearn/source/checkout?repo=gae>

The PIM and the Mobile Data Collection system have been developed as an extension of the ARLearn software architecture. ARLearn is a system that enables authors to create and organize mobile serious games. With the authoring tool (<http://streetlearn.appspot.com>), they can for instance define a script for a fieldtrip. With the smartphone app (<https://play.google.com/store/apps/details?id=org.celstec.arlearn2.android>), students can participate in the fieldtrip and read the various messages on their smartphone. Both the ARLearn authoring tool and smartphone application rely on the ARLearn cloud to manage games and results. The ARLearn authoring tool is not used within weSPOT, as authoring data collection tasks has been integrated in the IWE. The ARLearn smartphone app is not used either and has been replaced by the PIM and the GPIM.

The Mobile Data Collection System (MDCS) manages both data collection tasks and results. This component has been developed as an extension of the ARLearn cloud. Rather small extensions have been implemented here that are specific for weSPOT. E.g. a method was developed to enable the Inquiry Workflow Engine register new users with ARLearn. So that, when a new user enters the IWE Environment, the account is copied to ARLearn. Also, all ARLearn web services were extended with functionality to make it possible for the IWE to execute methods on behalf of a user.

The Personal Inquiry Manager (PIM) is the tool that enables students to execute data collection tasks. Although the PIM can be viewed upon as a special version of the ARLearn smartphone app, it is a complete newly created software package for the weSPOT context. The PIM relies on the IWE to show the user the list of inquiries and relies on the mobile data collection system to render data collection tasks and to publish results of these tasks. The PIM is available for both iOS and Android at the following locations:

- Android: <https://play.google.com/store/apps/details?id=net.wespot.pim>
- iOS: <https://itunes.apple.com/nl/app/personal-inquiry-manager/id889937609>

Mobile Data Collection Model

The Mobile data collection system supports PIM users to perform data collection tasks. Within weSPOT five types of data collection tasks were identified:

1. Picture Task. Enable the user to respond to a task by taking one or more picture.
2. Video Task. Enable the PIM user to respond to a task by capturing one or more videos.
3. Audio Task. Enable the PIM user to send in an audio recording
4. Text Task. Enable the user to respond to a task by writing down the data as a character string.
5. Numeric value Task. Here a user responds by entering a measurement. This could e.g. the temperature at a certain place.

Furthermore, all collected data samples are extended with generic sensor data, contextualizing the data sample:

1. A GPS coordinate is added when GPS is available. This enables the sample to be located on map for a user of the Mobile Inquiry Coordination Interface.
2. The PIM calculates the time difference between the mobile device and the server. This timestamp of each data sample taken is increased with this time delta. Every data sample thus has a more or less consistent timestamp.

Architecture

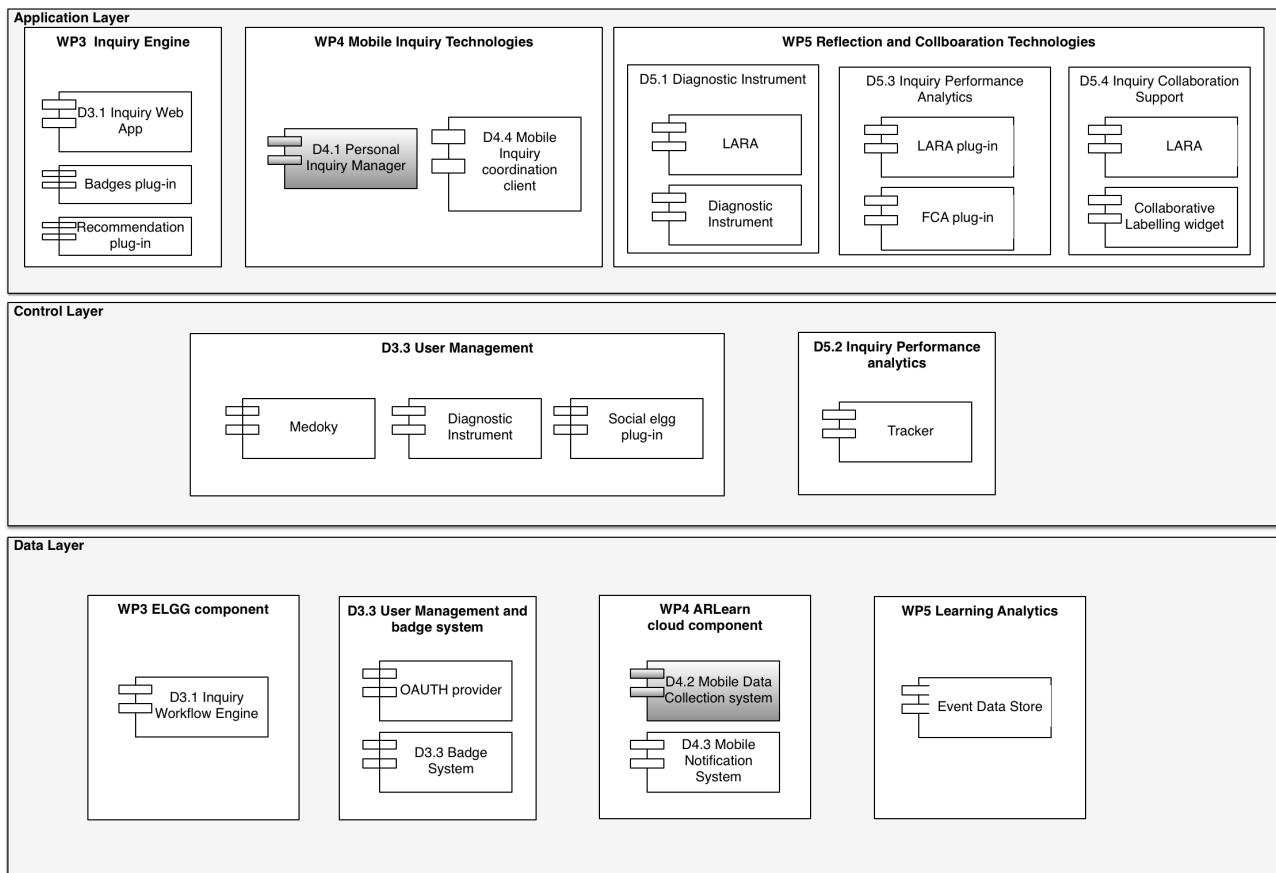


Figure 1: Overall weSPOT architecture

Within the weSPOT architecture (Fig. 1), this deliverable has a focus on the Mobile Data Collection system, a cloud based data layer component. Furthermore, it illustrates how the Personal Inquiry Manger (D4.1) builds on this functionality. Fig. 2 presents how components that reside in the application layer communicate with the WP4 cloud components. As a user creates data collection tasks from within the inquiry workflow engine (1) corresponding ARLearn concepts are created in the mobile data collection system. These artefacts are communicated via the mobile notification system to the various applications that build on this functionality (e.g. the PIM, GPIM, ...). After receiving a notification, these applications programmatically list the data collection tasks (5: listGeneralItems).

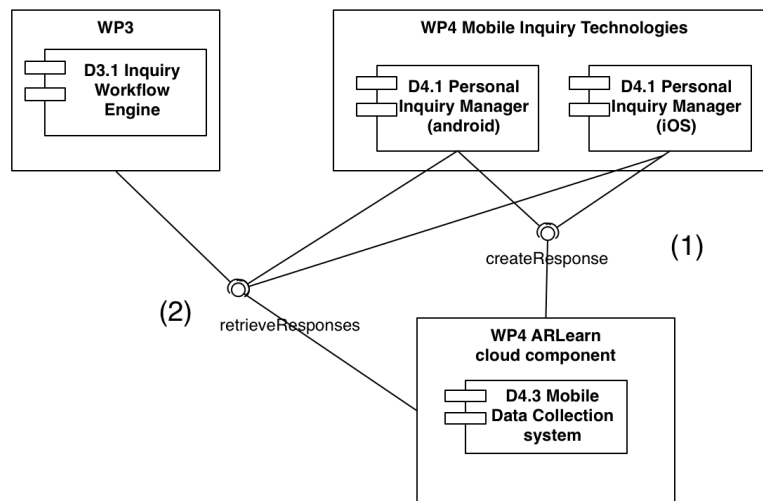


Figure 2: Create/List data collection tasks

Once the PIM has downloaded the list of data collection tasks, a user can start collecting data samples. As soon as network is available, the PIM starts submitting the data to Mobile Data Collection System (Figure 3.1). Next a user that consults the data collection phase on the Inquiry Workflow Engine (Figure 3.2) will see the results appearing. The Mobile Data Collection system builds on Google App Engine to manage both data and metadata. The response metadata store is captured in high-replication datastore. Data files including audio, video and pictures are stored in the GAE blobstore. Note that the data collected via the PIM is viewable both via the Desktop based inquiry workflow engine and is viewable by other PIM users that participate in the same inquiry.

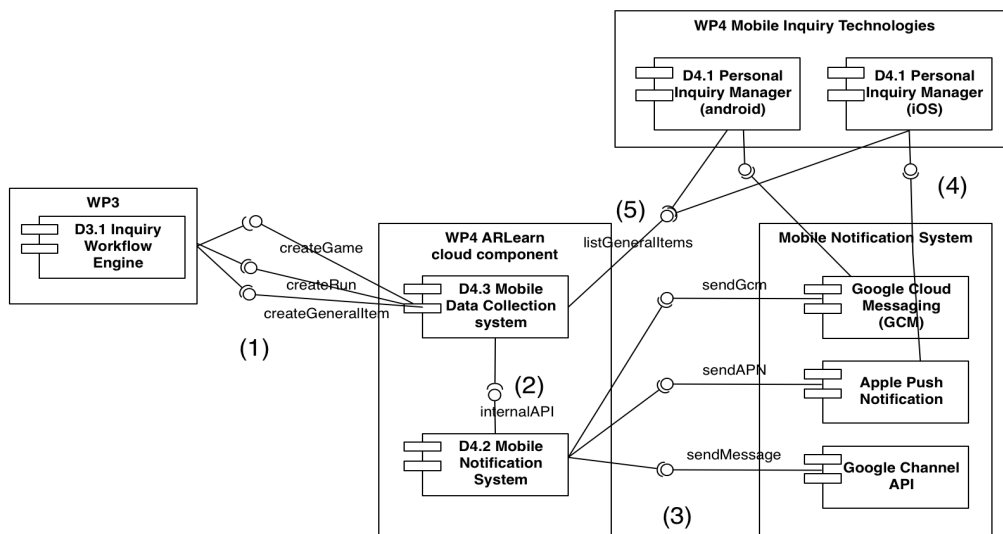


Figure 3. Creating and retrieving responses

The PIM was developed as a native application for the two main mobile platforms: iOS and Android. Furthermore a more experimental Google Glass version of the PIM - GPIM - was developed. Although the weSPOT architecture supports other clients (such as an HTML client), a native app was built as this gives the system more flexibility with respect to dealing with sensor data. Unlike HTML apps, native apps have direct access to the various sensors (e.g. GPS). They offer better support for offline working, as they can capture data and synchronize it later when a user is again online. A student that e.g. steps out of the school network will thus be able to synchronize all collected data when again entering the network.

API

The mobile data collection system builds on the ARLearn API to offer a REST based API. This API allows for management of data collection tasks and enables various applications to get access to the results. This chapter first presents the data model of a data collection task. Next, the various methods that enable the weSPOT clients (PIM, IWE) to communicate with the mobile data collection system are presented.

Data collection task model

A data collection task is encoded as an ARLearn message. This message illustrated in figure x, defines various metadata that enable the PIM to display the data collection task

- **gameId.** An identifier of the ARLearn game to which this task belongs. Every inquiry maps to one ARLearn game.
- **id:** a unique identifier of this message
- **lastModificationDate:** A unix epoch (represents the number of seconds that have elapsed since 01/01/1980) that captures the date and time at which this message was last modified.
- **name** represents the title of this message
- **richText** captures an instruction and is encoded in HTML.
- **openQuestion.** Including the openQuestion object will complement this ARLearn message with data collection possibilities. Inside the openQuestion tag, one can specify what data is to be collected (e.g. "withValue" stands for capture numeric values)

An example of an ARLearn, JSON encoded data collection task is displayed below.

```
{
  "type": "org.celstec.arlearn2.beans.generalItem.NarratorItem",
  "gameId": 524648726921216,
  "lastModificationDate": 1410429369894,
  "id": 579562450714624,
  "scope": "user",
  "name": "Temperature measurement",
  "fileReferences": [],
  "richText": "Measure <b>temperature</b> at various locations",
  "openQuestion": {
    "type": "org.celstec.arlearn2.beans.generalItem.OpenQuestion",
    "withAudio": false,
    "withText": false,
    "withPicture": false,
    "withVideo": false,
    "withValue": true
  }
}
```

Method: Create data collection task

The MDCS API enables both create, updating and querying these messages (or data collection tasks in the weSPOT context) in a REST style. The method displayed below illustrates how this task is created with a POST message. A “General Item” represents a data collection task and is published via JSON encoded http request to the cloud. On successful creation of the task, a notification is sent to all devices that are registered to the corresponding inquiry (D4.2 Context aware notification system)

```

Method: POST
Path: /rest/generalItems
Body
{
    "type": "org.celstec.arlearn2.beans.generalItem.NarratorItem",
    ...
}

HTTP Response body
{
    "type": "org.celstec.arlearn2.beans.generalItem.NarratorItem",
    "id": 1957392293
    ...
}

```

Method: Retrieve data collection task

After the creation of a data collection task, client applications such as the iOS/Android PIM can retrieve the data collection tasks from the server.

```

Method: GET
Path: /rest/generalItems/gameId/3845903

HTTP Response body
{
    "type": "org.celstec.arlearn2.beans.generalItem.GeneralItemList",
    "serverTime": 1410784918038,
    "generalItems": [
        { .. },
        { .. }, ...
    ]
}

```

Method: Create response

The methods detailed in the previous section enable management of data collection tasks and make it possible for users to pull tasks from the server. During the course of the inquiry, students capture

responses for these tasks. This translates in two web services to support this process: creation of a response and querying of existing responses.

The creation method enables the client to publish a JSON encoded response to the server. After successful publication, the server returns the same JSON together with a server generated response identifier (responseId).

```
Method: POST
Path: /rest/response
Body:
{
  "type": "org.celstec.arlearn2.beans.run.Response",
  "timestamp": 1390469153177,
  "runId": 21606001,
  "generalItemId": 20536007,
  "userEmail": "2:105939139551108473521",
  "responseValue":
  "{\"imageUrl\":\"http://streetlearn.appspot.com/uploadService/21606001/2:105939139551108473521/image-1067714692.jpg\",\"width\":306,\"height\":408}"
}
HTTP Response body
{
  "type": "org.celstec.arlearn2.beans.run.Response",
  "timestamp": 1390469153177,
  "runId": 21606001,
  "deleted": false,
  "responseId": 21616002,
  "generalItemId": 20536007,
  "userEmail": "2:105939139551108473521",
  "responseValue":
  "{\"imageUrl\":\"http://streetlearn.appspot.com/uploadService/21606001/2:105939139551108473521/image-1067714692.jpg\",\"width\":306,\"height\":408}"
}
```

Method: Retrieve Response

Next, various systems can pull results from the server. Another user of the PIM, which is subscribed to the same inquiry, can pull the results that were submitted by its peers. In addition the IWE contains a widget that visualizes the results. This pulling of resources is done via the following web service

```

Method: GET
Path: /rest/response/runId/21606001?from=0

HTTP Response body
{
  "type": "org.celstec.arlearn2.beans.run.ResponseList",
  "deleted": false,
  "responses": [
    {
      "type": "org.celstec.arlearn2.beans.run.Response",
      "timestamp": 1390469153177,
      "runId": 21606001,
      "deleted": false,
      "responseId": 21616002,
      "generalItemId": 20536007,
      "userEmail": "2:105939139551108473521",
      "responseValue":
      "{ \"imageUrl\": \"http://\\//streetlearn.appspot.com\\//uploadService\\//21606001\\//2:105939139551108473521\\//image-106771469212.jpg\", \"width\": 306, \"height\": 408 }"
    }, {..}, ...
    "serverTime": 1410785652681,
    "resumptionToken": "E-
ABAOsB8gEUbGFzdE1vZGlmaWNhdGlvbkRhdGX6AQcIlKmiwr8o7AGCAiVqDXN-
c3RyZWV0bGVhcm5yFAsSC1Jlc3BvbNlSkRPGILhiAIAFA"

```

This method returns a list of responses for an inquiry. Via the path, the system can specify a unix epoch time in the from field. Specifying an epoch time here, will yield responses created after this time.

For scalability reasons (e.g. if there are hundreds of results) the system will only return 20 responses. At the end of the response a resumptionToken enables downloading the next batch of results. This is done by issuing an HTTP get on the following URL:

```

/rest/response/runId/21606001?from=0&resumptionToken=E-
ABAOsB8gEUbGFzdE1vZGlmaWNhdGlvbkRhdGX6AQcIlKmiwr8o7AGCAiVqDXN-
c3RyZWV0bGVhcm5yFAsSC1Jlc3BvbNlSkRPGILhiAIAFA

```

Note that the query "from" query parameters must be that same as the one specified in the original request ("0" in this example)

Apart from resumptionToken, the message also contains a serverTime parameter. This parameter can be used to download (at some later point in time) the next batch of new responses:

```

/rest/response/runId/21606001?from=1410785652681

```


This enables the system to selectively download results. It does not need to query for all responses again, but can synchronize only responses that were created or modified since the last synchronization point.

Smartphone client (PIM)

The Personal Inquiry Manager was described in Deliverable 4.1 and builds on the mobile data collection system. Since the publication of D4.1, the PIM has gone through a redesign phase. Although, the base functionality has not been altered, the user experience has been improved. Rather than presenting all features again, this chapter focuses on the new interface and highlights where the mobile data collection system comes into play.

Login

The PIM now complements the login procedure with the weSPOT account system (see deliverable 3.3). Figure 3 demonstrates this login both on Android (right) and iOS (left).



Figure 3. Android login (left) and iOS login (right)

List Inquiries

After successful login, the PIM loads inquiries from the IWE and displays them as displayed in Figure 4.

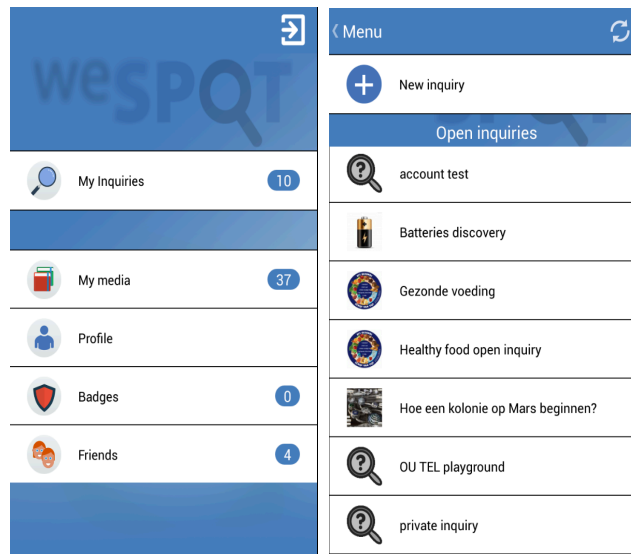


Figure 4. Start screen + my Inquiries

Data collection tasks

For each inquiry the PIM lists only those phases in the IBL model that make sense on a mobile device. As a mobile device has a limited screen size, activities such as mind mapping, analysing data with a statistical package or the creation of a presentation do not make sense here. Figure 5 illustrates how the user can from the main screen of an inquiry navigate to the data collection tasks. Next for a single data collection task, the PIM shows all collected data samples (including the samples taken by peer students).

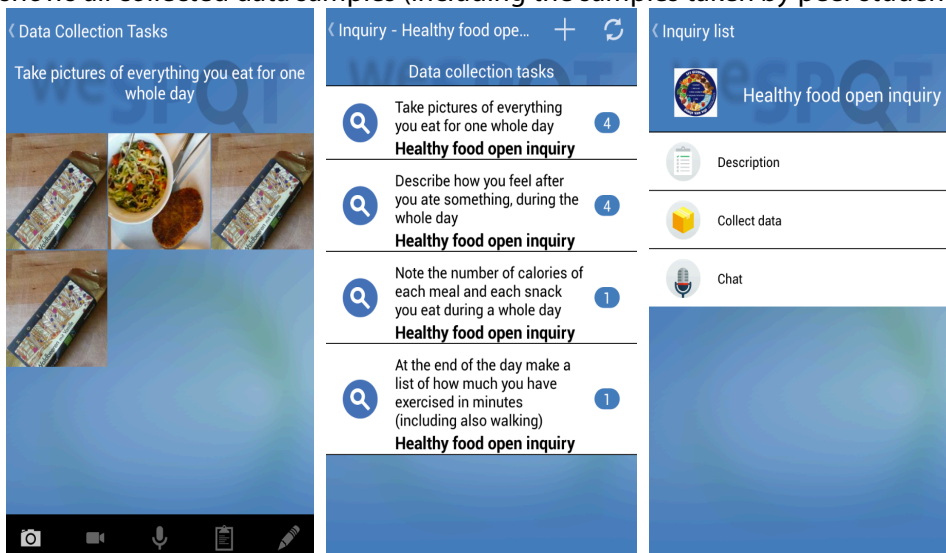


Figure 5. Inquiry screen, data collection tasks, single task + results

Google Glass client (GPIM)

The Google Glass PIM (Suarez, 2014) is a weSPOT innovation project that leverages the IBL model on wearable technology. It supports level 4 open inquiries and starts from a student initiated wonder moment.

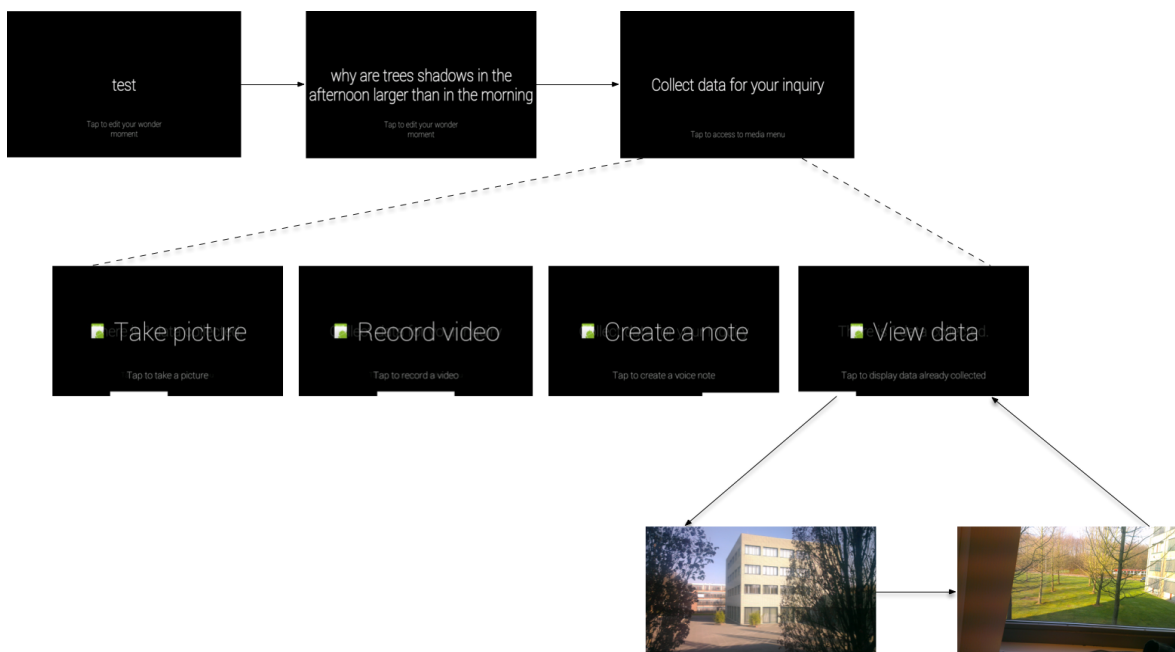


Figure 6. GPIM workflow

The user starts the GPIM to support an open inquiry process. To activate the GPIM the student does not interact physically with the device. With a simple voice command “Ok Glass, new inquiry”, a new inquiry is started on Glass.

During this initial phase, the user makes a statement about what he or she wonders about. For instance, a student notices that the shadows of the trees are larger in the afternoon than in the morning. This simple idea can trigger an entire spontaneous inquiry about the earth's rotation and how it revolves around the sun. In a next phase, the user can define and refine a hypothesis.

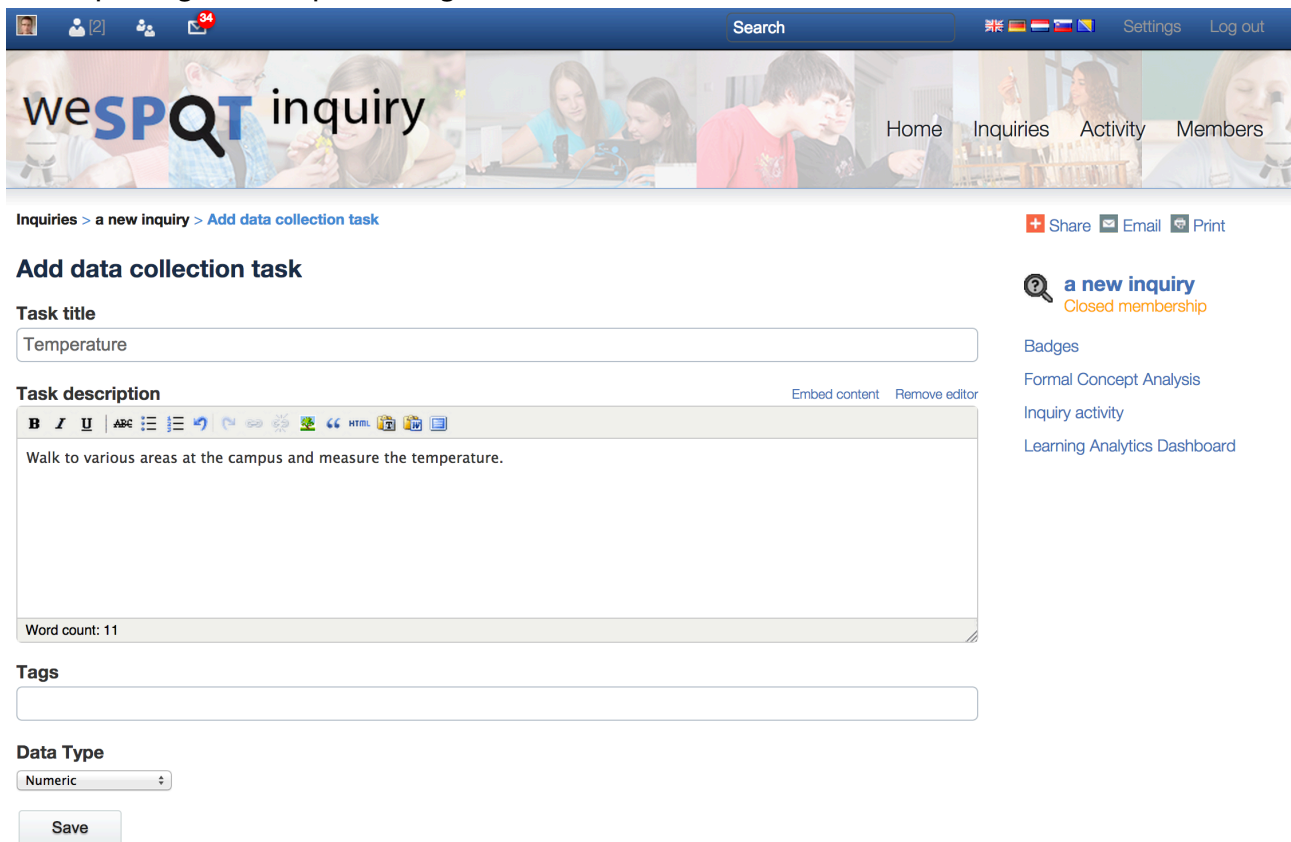
Finally, the GPIM supports data collection. Here the user can tap the “collect data card for your inquiry” and next select the data collection type e.g. “take picture”.

At the time of writing this report, the collected data is not synced with the weSPOT data layer. This remains to be done.

The GPIM builds on technology that adapts to a user's environment. It enables executing data collection tasks from offering the user a first view perspective. In contrast to the PIM, a user has both hands free and can freely interact with the environment.

Inquiry Workflow Engine integration

The Inquiry Workflow Engine provides functionality to create data collection tasks, listing results and exporting data for processing.



The screenshot shows the weSPOT inquiry interface. At the top, there is a navigation bar with a search box, language flags, and links for Settings and Log out. Below the navigation bar is a banner with the weSPOT inquiry logo and navigation links for Home, Inquiries, Activity, and Members. The main content area is titled 'Inquiries > a new inquiry > Add data collection task'. On the right side, there are options to Share, Email, and Print. Below these options is a section for 'a new inquiry' with a 'Closed membership' status and a list of badges: Badges, Formal Concept Analysis, Inquiry activity, and Learning Analytics Dashboard. The main form has three sections: 'Task title' with a text input field containing 'Temperature'; 'Task description' with a rich text editor containing the text 'Walk to various areas at the campus and measure the temperature.' and a word count of 11; and 'Tags' with an empty text input field. At the bottom, there is a 'Data Type' dropdown menu set to 'Numeric' and a 'Save' button.

Figure 7. Add data collection task

Figure 7, illustrates how a teacher can - within the IWE - define a new data collection task. As the result, the IWE will invoke the “create data collection task” method on the MDCS. Next, the MDCS will send notifications to all instances of the PIM that are subscribed to this inquiry.

IWE: display results

As data is collected by the various students, the IWE data collection widget will update and reflect the various data that were collected. Figure 8 illustrates how the IWE shows pictures that were taking with the PIM. These pictures are listed via the “retrieve response method”. In addition, users can retrieve these results with the PIM or with the mobile inquiry coordination interface.

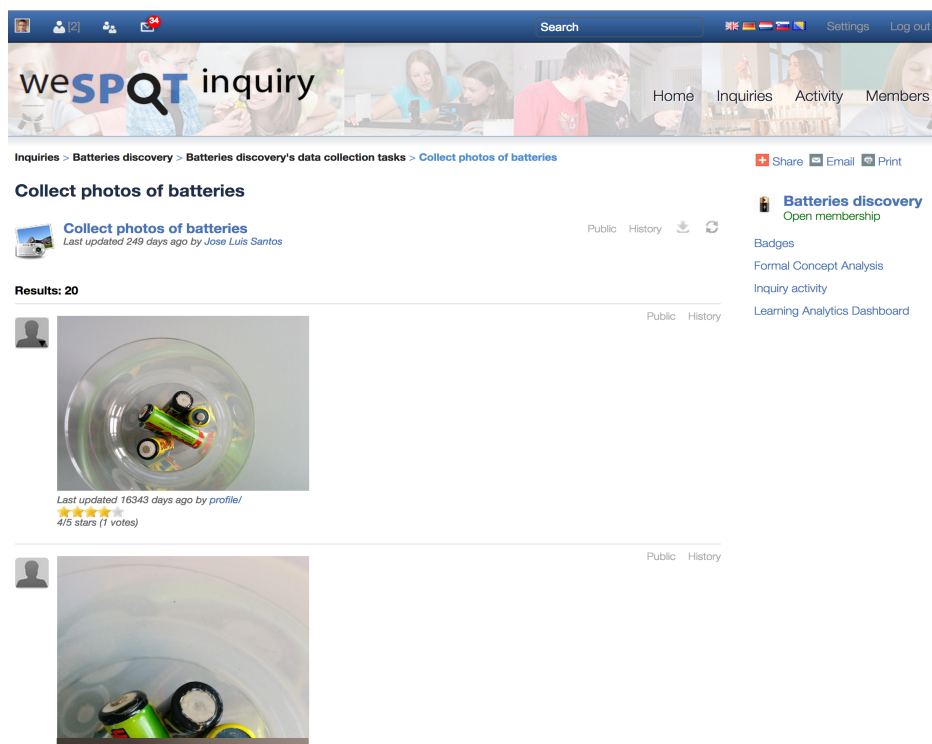


Figure 8. The IWE listing pictures from the MDCS.

Data export

Some data require further analysis. For instance, for numeric data, the IWE interface only lists the individual values. Simple functions like calculating the mean or average are not possible with this interface. In order to support this, the IWE data collection widget has been extended with support to export data as a CSV file.

A comma-separated values (CSV) file stores tabular data in plain-text form and is a relatively simple file format that is widely supported by scientific applications. Enabling the MDCS to export its data into this format results in an straightforward manner to import the data into e.g. MS Excel, various database programs or statistical software packages such as SPSS.

The corresponding User Story (US #565) for this functionality was:

As a student, I must be able to download a spreadsheet with an overview of all data collected within an inquiry including time, GPS coordinates, user identifier and value (number, text, URL of the data).

This export functionality has been realized in several phases. First the functionality to export was implemented on the MDCS infrastructure. Task #575 defined the implementation of a service to start building the CSV file. Note that as this data can potentially be very large, it is not possible to extract it from the service with a single query.

The build process is invoked by calling
 response/csv/runId/<runId>/build

As a result this service returns a json message with a result identifier and a status. E.g.

```
{
  "status": 1,
  "id": "d09bd640-b20e-4d15-9c56-1e140e2a330a",
  "info": "BUILDING (1) FINISHED (2)"
}
```

A status (1) means the cloud infrastructure is still building the CSV file, status (2) means that the file is ready to be downloaded.

Status updates can repeatedly be queried by calling
 response/csv/<result identifier>/status

When the build process is finished, the service returns a with status code 2 (FINISHED):

```
{
  "status": 2,
  "id": "d09bd640-b20e-4d15-9c56-1e140e2a330a",
  "info": "BUILDING (1) FINISHED (2)"
}
```

As soon as this service returns status (2), the CSV can be downloaded via the following URL:
 http://streetlearn.appspot.com/rest/response/csv/<result identifier>

In a second phase this service was integrated into the Elgg Data Collection Widget (Task #634). This task includes adding a button to data collection widget (Fig. 9). When a user clicks the build button, this widget starts building the CSV file. When the user clicks on the download button, it returns the CSV file.

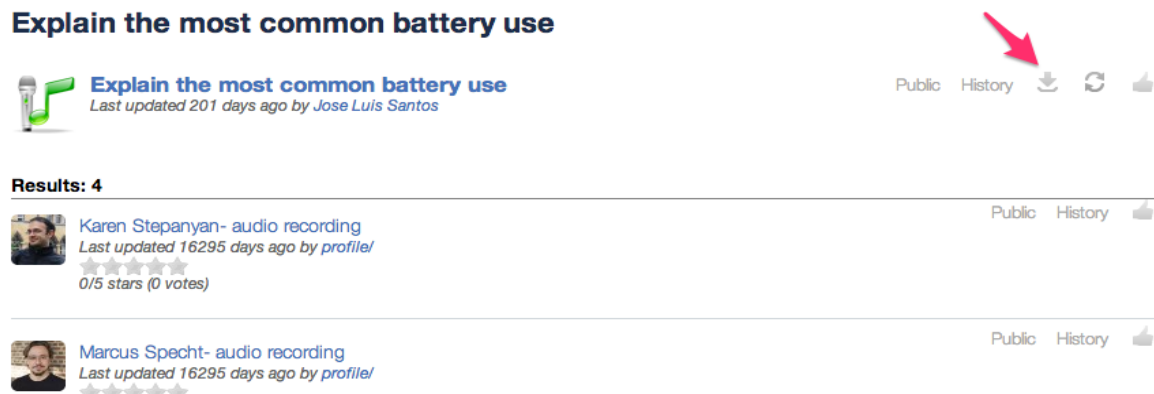


Figure 9: Mobile Data collection widget extended with button to export CSV file.

Conclusion

This deliverable presents the mobile data collection system and its integration into various weSPOT tools:

- Inquiry Workflow Engine (IWE) users can now create data collection tasks on the MDCS and view the results of these tasks

- Via the Personal Inquiry Manager, users can create new data collection tasks or access the data collection tasks were defined in the IWE. Various weSPOT pilots will in the next coming months use this app to perform data collection tasks.
- The Personal Inquiry Manager for Google Glass is an innovative approach for data collection, building on wearable devices. Although this tool will be further developed (e.g. connect to the live MDSC services) and evaluated, the limited availability of Google Glass will not make this tool usable on a large scale.

References

Suarez, A., Ternier, S., Kalz, M., & Specht, M. (2014). GPIM: Google Glassware for Inquiry-Based Learning. In C. Rensing et al (Eds.), Proceedings of the 9th European conference on Technology Enhanced Learning - EC-TEL 2014: Open Learning and Teaching in Educational Communities, LNCS 8719 (pp. 530-533). Graz, Austria.