ExCiteS | Extreme Citizen Science

Design Brief – Sapelli Viewer 1.0

Contents

1. Introduction

1.1. The Extreme Citizen Science group

Extreme Citizen Science, or ExCiteS, is a research group at University College London which develops and contributes to the guiding theories, tools and methodologies that will enable any community to start a Citizen Science project to address issues that concern them. Using an interdisciplinary research approach, ExCiteS aims to provide any user, regardless of their background or literacy level, with a set of tools that can be used to collect, analyse and act on information according to agreed-upon scientific methods.

Several of the ExCiteS projects focus on participatory sensing, monitoring and modelling activities, with communities deciding what measurements are taken and how they are analysed so that they can participate in and lead subsequent decision-making and actions. ExCiteS vision is to change the current state of the art by developing technologies to enable lay people to understand and manage their environment with established scientific methods and models. In the next paragraph we discuss the ECSAnVis project in more detail.

1.2 The Extreme Citizen Science: Analysis and Visualisation project: Case studies and tools

Extreme Citizen Science: Analysis and Visualisation project (ECSAnVis) is a five-years, €2.5M project funded by the European Research Council, which aims at the development of geographical analysis and visualisation tools that can be used, successfully, by non-literate people and any other community in a culturally appropriate ways, that further fit their needs and social practices. ECSAnVis case studies currently include:

- The Ju/'hoansi in the Nyae Nyae Conservancy (Namibia), who collect data on a wide range of fauna and flora as part of their ongoing community-based natural resource management work;
- Local fishermen in Pantanal wetlands (Brazil) where people collect data of their local fishing and other resource management practices to provide evidence that these are indeed sustainable;
- Local farmers in Ghana, who collect data to capture and share their local knowledge of weather prediction;
- Maasai pastoralists communities in Narok county (Kenya), which collect data of the medicinal and other properties of local indigenous flora. Small-scale farming communities in Elgeyo-Marakwet county (Kenya), which collect data of land uses and farming issues they face;
- Baka and Bantu-speaking communities in Dja Biosphere Reserve (Cameroon), which collect data to tackle: wildlife crime, illegal logging, species monitoring, resource mapping (including fishing sites and medicinal trees), abuse by forest guards, and record births and deaths.

For a full description of these and our past case studies please refer to the ECSAnVis project Story Maps.

The main tool that it is used in ECSAnVis to support communities in the data collection iprocess s Sapelli (Collector), an Android-based application and open-source project, which has a hierarchical icon-based interface, which can be used for data collection purposes by people regardless of their literacy and/or technical skills. For an in-depth description of Sapelli, please refer to Sapelli.org for examples, tutorials and the projects XML file. We also encourage you to download the app from the Google Play store, or from our GitHub repository and deploy it for their own purposes.

Other tools we are currently developing include GeoKey (https://geokey.org.uk), which serves as the default back end of Sapelli deployment and support visualisation through Community Maps (https://communitymaps.org.uk) and other platforms; and Sapelli Designer (https://sapelli-designer.netlify.app), which is a Progressive Web App (PWA) that aims to provide a user-friendly interface (i.e. XML programming skills are not required) to design and generate Sapelli Projects.

2. Sapelli Viewer Specification

We seek to recruit a developer to develop and support the ExCiteS team in testing and launching Sapelli Viewer (as an Android mobile app). Sapelli Viewer is a mobile data visualisation tool that allows users to view data they collect using Sapelli Collector.

Note that the specifications described in this document for the first version of Sapelli Viewer apply to the first iteration (i.e. viewing data only) of a final prototype which will eventually combine Sapelli Collector, Sapelli Designer, and Sapelli Viewer into a single tool to allow users with low levels of digital or print literacy to design data-collection projects and collect, visualise, and analyse data according to local needs, desires, and abilities.

Sapelli Viewer will support our two user personas which we describe in the next section to: create a project or load an existing one; visualise point data on a map that it has been collected using Sapelli Collector for the same project; display text, photos, videos and/or audios associated with data points; load data from GeoKey and from the device's database where Sapelli Collector is installed (when data are stored locally in the device); filter data; and display the user's GPS location on the map. We discuss all these requirements in more detail in Section 2.2.

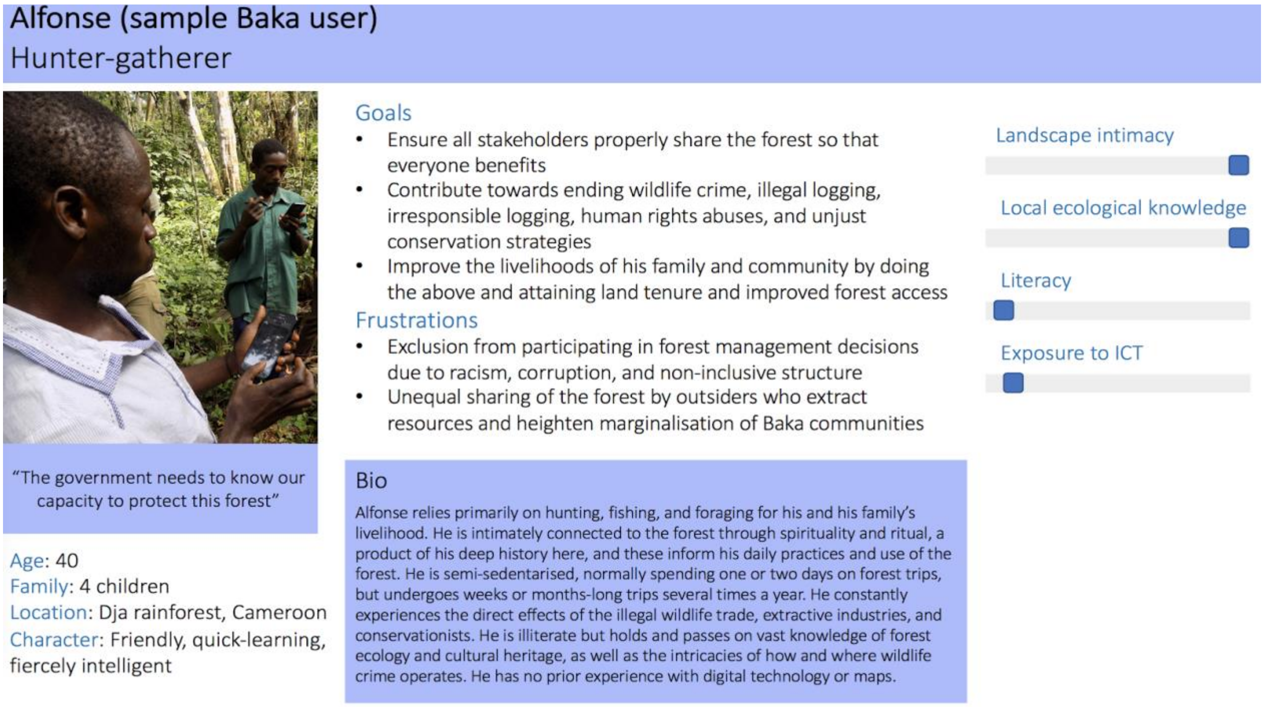
A copy of the Adobe XD prototype, which is extensively discussed throughout this document is available to view here. Note that the icons used in the prototype are only examples and the development team to be recruited will need to develop all Sapelli Viewer icons which are used in the application.

2.1 Sapelli Viewer Personas

There are two primary personas for Sapelli Viewer:

- a user with low levels of print or digital literacy who collects, visualises, and analyses data either alone or in a team (Persona A), and
- an administrator with medium-to-high levels of literacy who works with low-literate users to collect, visualise, and analyse data (Persona B). It is important that we have both an administrator option and a standard user option so that users with low levels of literacy do not change key functions and settings by mistake.

PERSONA A: Sapelli User {Cameroon Project}



Alfonse (sample Baka user)
Hunter-gatherer

Goals

- Ensure all stakeholders properly share the forest so that everyone benefits
- Contribute towards ending wildlife crime, illegal logging, irresponsible logging, human rights abuses, and unjust conservation strategies
- Improve the livelihoods of his family and community by doing the above and attaining land tenure and improved forest access

Frustrations

- Exclusion from participating in forest management decisions due to racism, corruption, and non-inclusive structure
- Unequal sharing of the forest by outsiders who extract resources and heighten marginalisation of Baka communities

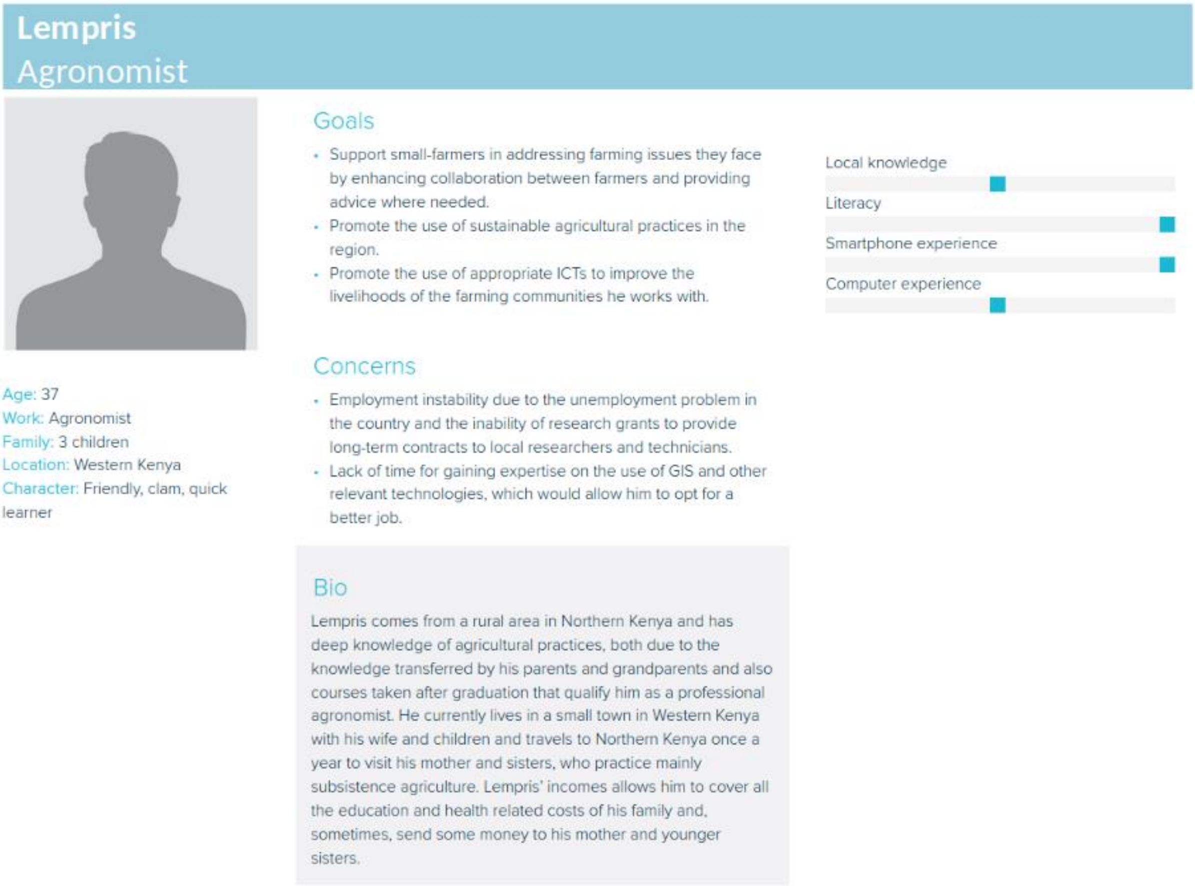
"The government needs to know our capacity to protect this forest"

Age: 40
Family: 4 children
Location: Dja rainforest, Cameroon
Character: Friendly, quick-learning, fiercely intelligent

Bio

Alfonse relies primarily on hunting, fishing, and foraging for his and his family's livelihood. He is intimately connected to the forest through spirituality and ritual, a product of his deep history here, and these inform his daily practices and use of the forest. He is semi-sedentarised, normally spending one or two days on forest trips, but undergoes weeks or months-long trips several times a year. He constantly experiences the direct effects of the illegal wildlife trade, extractive industries, and conservationists. He is illiterate but holds and passes on vast knowledge of forest ecology and cultural heritage, as well as the intricacies of how and where wildlife crime operates. He has no prior experience with digital technology or maps.

PERSONA B: Sapelli Administrator {Kenya Project}



Lempris
Agronomist

Goals

- Support small-farmers in addressing farming issues they face by enhancing collaboration between farmers and providing advice where needed.
- Promote the use of sustainable agricultural practices in the region.
- Promote the use of appropriate ICTs to improve the livelihoods of the farming communities he works with.

Local knowledge
Literacy
Smartphone experience
Computer experience

Concerns

- Employment instability due to the unemployment problem in the country and the inability of research grants to provide long-term contracts to local researchers and technicians.
- Lack of time for gaining expertise on the use of GIS and other relevant technologies, which would allow him to opt for a better job.

Age: 37
Work: Agronomist
Family: 3 children
Location: Western Kenya
Character: Friendly, clam, quick learner

Bio

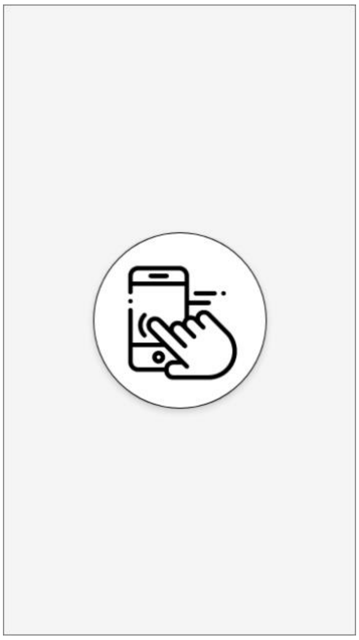
Lempris comes from a rural area in Northern Kenya and has deep knowledge of agricultural practices, both due to the knowledge transferred by his parents and grandparents and also courses taken after graduation that qualify him as a professional agronomist. He currently lives in a small town in Western Kenya with his wife and children and travels to Northern Kenya once a year to visit his mother and sisters, who practice mainly subsistence agriculture. Lempris' incomes allows him to cover all the education and health related costs of his family and, sometimes, send some money to his mother and younger sisters.

3. Sapelli Viewer functionality

In this section we describe the key functionalities for this version of the Sapelli Viewer implementation (Sections 3.2.1-2.2.13). We also discuss some optional functionalities to discuss with the appointed development team (Section 4).

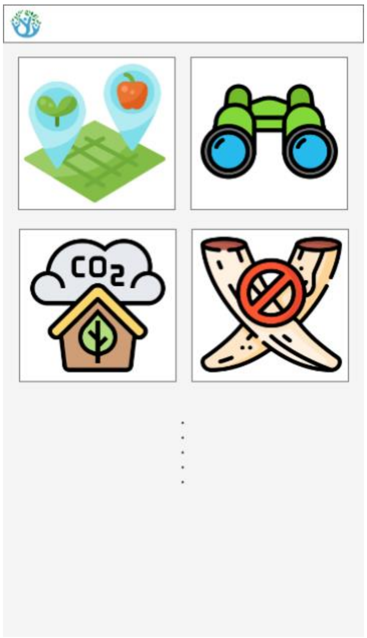
3.1 Setting up an admin account

Sapelli Viewer should open with a screen with one button that can be either clicked (for non-admin users) or holded for few seconds (for admin user).

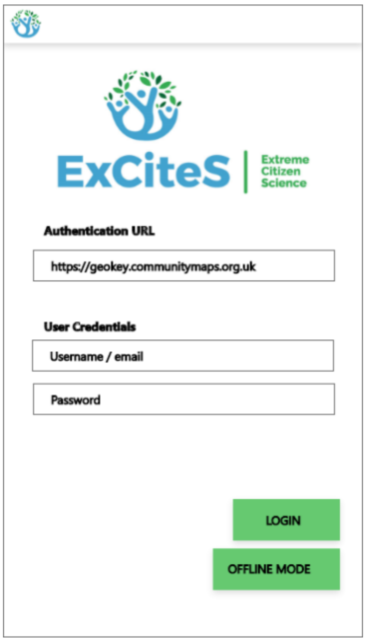


The icon, when clicked, will take the user to their projects. If only one project exists, they will be taken directly to the map interface.

Note: after the device has been set up, the screen that follows the splash screen will not show and the user will be taken directly to the screen to select the project or (if only one project exists) the visualisation interface.



An administrator will perform a 'counter-intuitive' input gesture (e.g. hold button 10 seconds) to enter the GeoKey login workspace. The dots underneath the projects indicate that if there are more projects than there is screen space, the user will be able to scroll down.



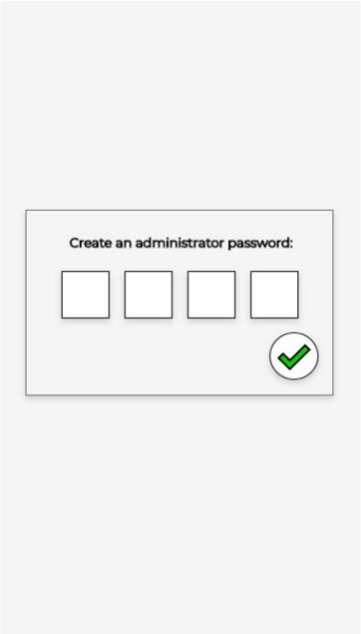
Authentication URL

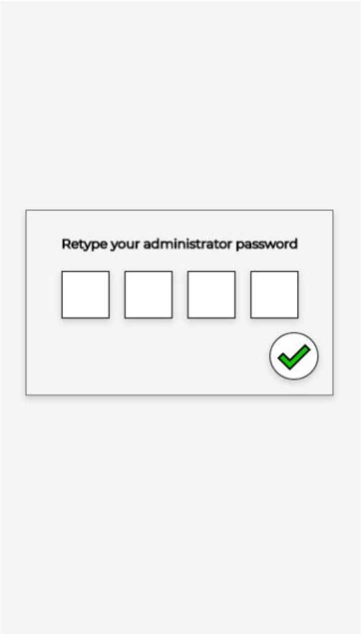
https://geokey.communitymaps.org.uk

The administrator will then be prompted to enter their GeoKey credentials. A GeoKey account must be used to set up Sapelli Viewer, and a Sapelli Project must be uploaded to GeoKey before setting up Sapelli Viewer. This must be set up prior to use.

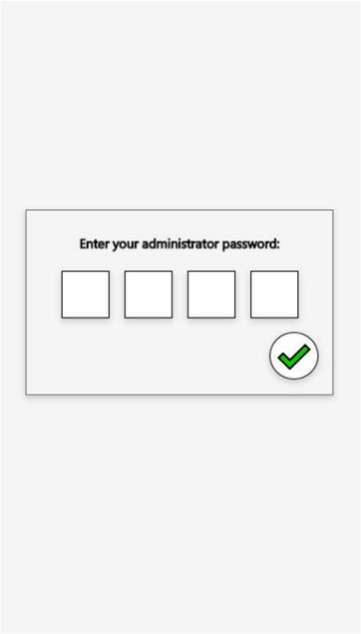
GeoKey credentials are set up online beforehand, but if possible, we would like to offer the option here to do this either when online in the app, or when the user is offline and in the app (sending the request to create an account to the server when the user comes back online) either in this or future versions of Sapelli Viewer.

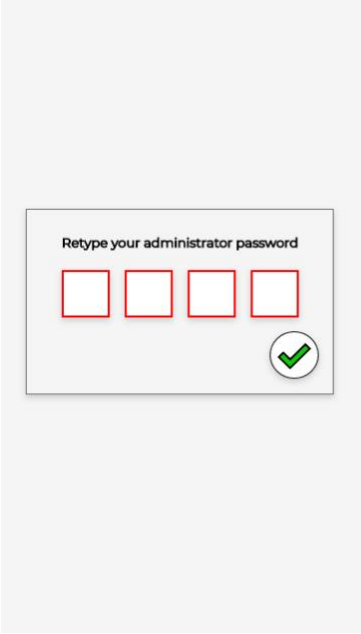
If the administrator needs to work offline, they can click the 'Offline mode' button.


Create an administrator password:


Retype your administrator password

After logging in GeoKey, or after clicking the ‘Offline mode’ button (to load manually a new Sapelli Project and basemaps), the administrator will be asked by the app to create an administrator password, which should then be used to access the administrator workspace while both online and offline. This restricts administrator rights from other users using the same device, and also can be used to assign passwords to different users that want to access different projects.


Enter your administrator password:


Retype your administrator password

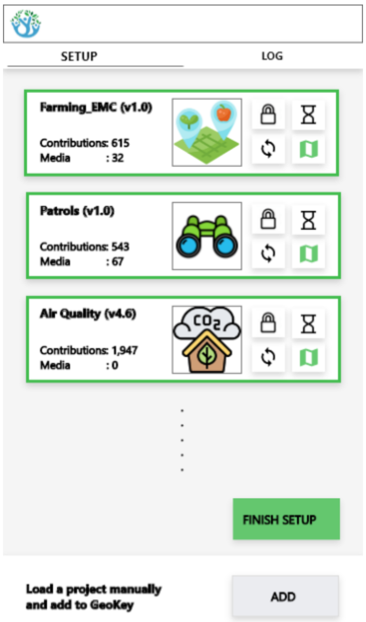
When the administrator returns to the administrator mode, they will be prompted to enter the administrator password they set up when they were online. The administrator password is stored locally so that the administrator can make changes when offline after the initial setup.

3.2 Loading Sapelli Collector Projects

Loading Sapelli Collector Projects via GeoKey

Sapelli Viewer must pull data not only from GeoKey, but also from the local storage of the device - where Sapelli Collector records are stored - to ensure that users can visualise data collected real-time, regardless of connectivity. In cases where the 'Administrator' can set up a project while online, the 'set up interface' (see below) must include a button to load data from GeoKey. In cases where users are using Sapelli Viewer while online, the map must update as the project records update (the records are transmitted to GeoKey by other users of the same Sapelli Collector project).

After logging in, the projects the admin user is a member of on GeoKey will automatically appear in a scrollable list. The administrator will then be able to load or change the base maps and get a snapshot of data (contributions and media) already on the server for that project that will be loaded to the map.



Farming_EMC (v1.0)
Contributions: 615
Media : 32

Patrols (v1.0)
Contributions: 543
Media : 67

Air Quality (v4.6)
Contributions: 1,947
Media : 0

Load a project manually and add to GeoKey

Clicking the 🔒 symbol will take the user to setting up group specific passwords for the project. (see 3.3 for further information).

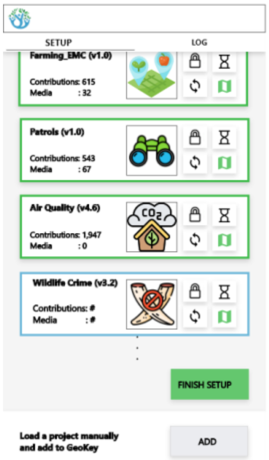
Clicking the 🗺 symbol will take the user to the default map storage folder for specific high-resolution maps. (see 3.6 for further information).

Clicking the ⌛ symbol will take the user to a new screen for assigning icons for data collection timeframes (see 3.4 for further information). *Note: Ultimately, we would like to extend this functionality to all the icons of the map interface (e.g. basemaps, recenter, …) should be included, using this button or adding another button.*

Clicking the ⟳ symbol will load the contributions and media from the GeoKey server into the device. This will allow users to view data that have been contributed by other users on their project. When the button is clicked, the user must be informed, by animating the circular arrows, that data is being imported from GeoKey. *Note: Our development team has already developed this functionality.*

Deleting Sapelli Collector projects in online mode

The admin user will have the option to remove Sapelli Collector projects that have been added manually by swiping left at the left of the project, a black delete icon appears, it becomes red when there is a click on it, and the project disappears prior to click on finish setup.


SETUP LOG

Farming_EMC (v1.0)
Contributions: 615
Media : 32

Patrols (v1.0)
Contributions: 543
Media : 67

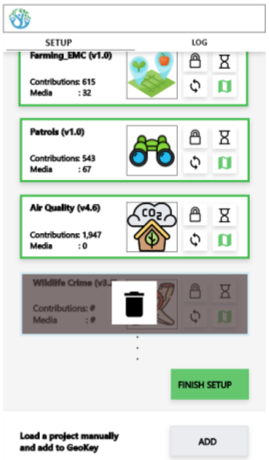
Air Quality (v4.6)
Contributions: 1,947
Media : 0

Wildlife Crime (v3.2)
Contributions: #
Media : #

FINISH SETUP

Load a project manually and add to GeoKey

ADD


SETUP LOG

Farming_EMC (v1.0)
Contributions: 615
Media : 32

Patrols (v1.0)
Contributions: 543
Media : 67

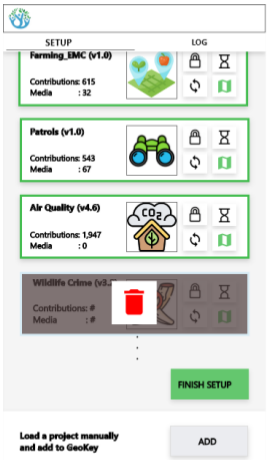
Air Quality (v4.6)
Contributions: 1,947
Media : 0

Wildlife Crime (v3.
Contributions: #
Media : #

FINISH SETUP

Load a project manually and add to GeoKey

ADD


SETUP LOG

Farming_EMC (v1.0)
Contributions: 615
Media : 32

Patrols (v1.0)
Contributions: 543
Media : 67

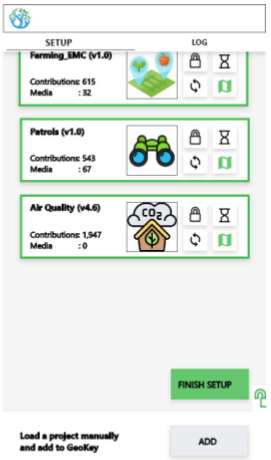
Air Quality (v4.6)
Contributions: 1,947
Media : 0

Wildlife Crime (v3.
Contributions: #
Media : #

FINISH SETUP

Load a project manually and add to GeoKey

ADD


SETUP LOG

Farming_EMC (v1.0)
Contributions: 615
Media : 32

Patrols (v1.0)
Contributions: 543
Media : 67

Air Quality (v4.6)
Contributions: 1,947
Media : 0

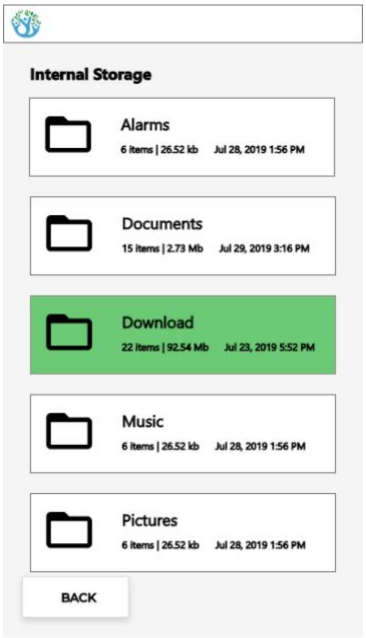
FINISH SETUP

Load a project manually and add to GeoKey

ADD

Loading Sapelli Collector Projects manually

Sapelli Viewer must take into account the scenario in which the admin user creates a Sapelli Collector and loads this project into Sapelli Viewer while offline, without communicating to GeoKey. The screenshots below show that .sap or .zip file of a Sapelli project must be accessible from Sapelli Viewer.


Internal Storage

Alarms
6 items | 26.52 kb Jul 28, 2019 1:56 PM

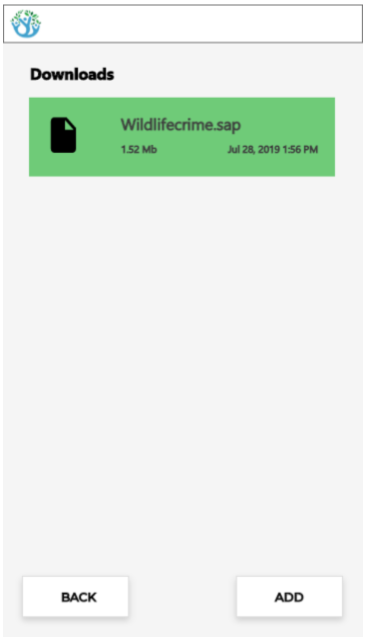
Documents
15 items | 2.73 Mb Jul 29, 2019 3:16 PM

Download
22 items | 92.54 Mb Jul 23, 2019 5:52 PM

Music
6 items | 26.52 kb Jul 28, 2019 1:56 PM

Pictures
6 items | 26.52 kb Jul 28, 2019 1:56 PM

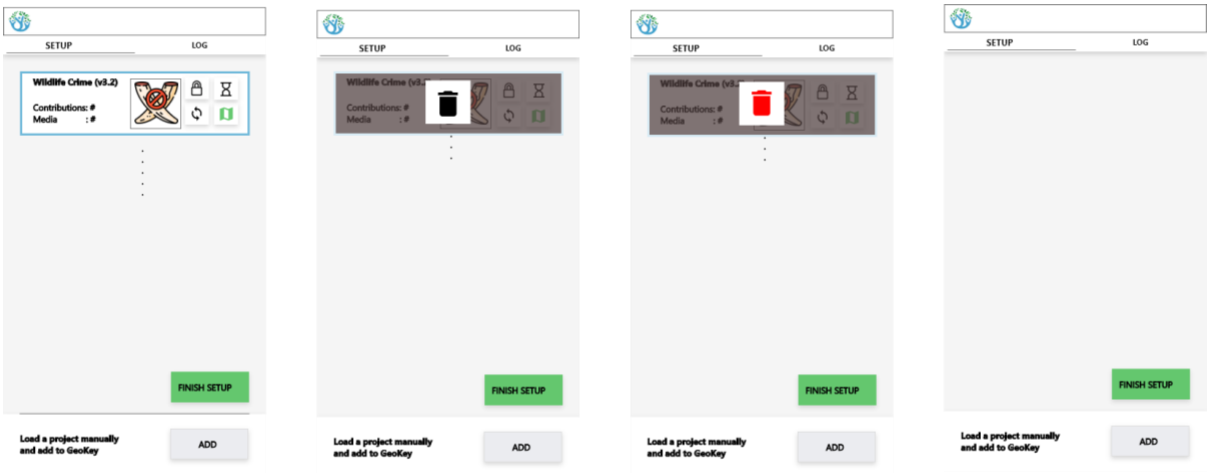
BACK


Downloads

Wildlifecrime.sap
1.52 Mb Jul 28, 2019 1:56 PM

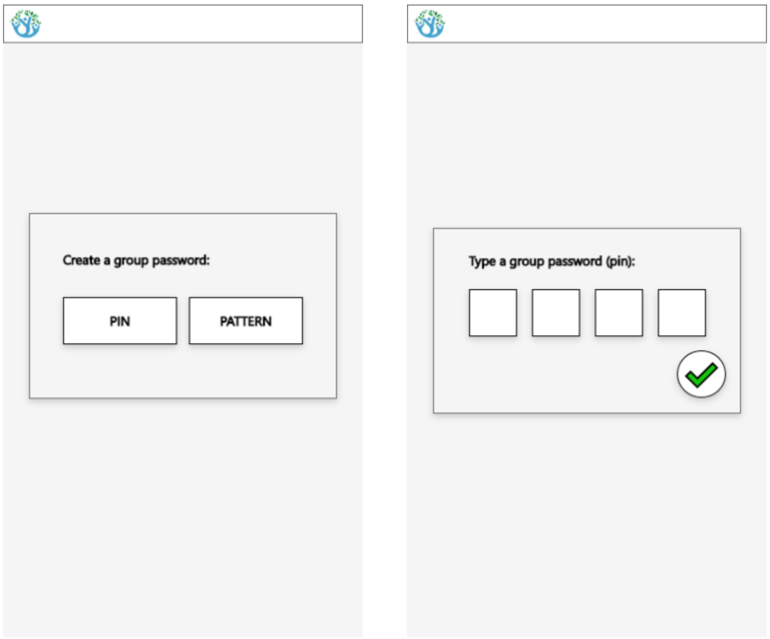
BACK ADD

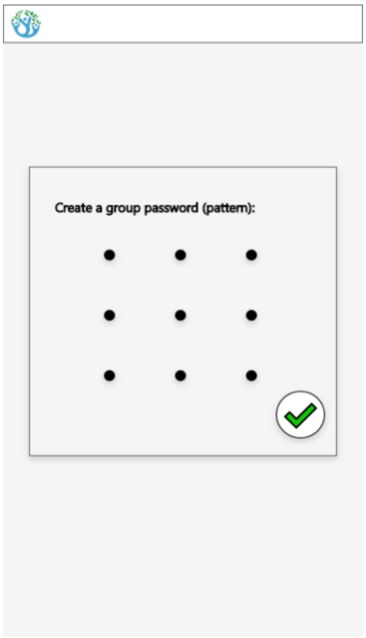
Deleting Sapelli Collector projects that have been added manually: The admin user will have the option to remove Sapelli Collector projects that have been added manually by swiping left at the left of the project, a black delete icon appears, it becomes red when there is a click on it, and the project disappears prior to click on finish setup.

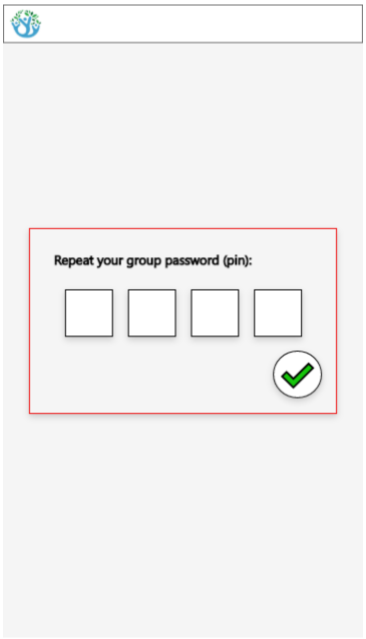


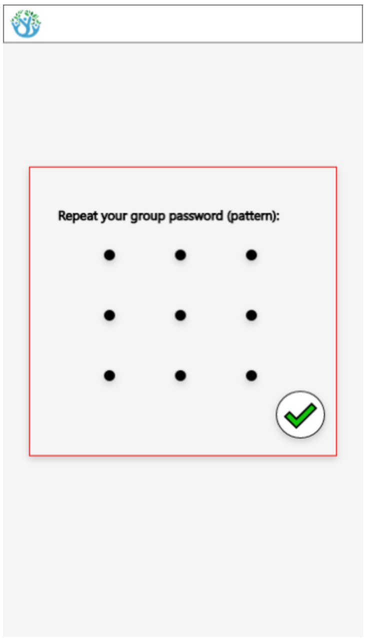
3.3 Setting group specific passwords

The screenshots below show how the admin user, together with end users, will restrict access to specific Sapelli projects by creating group passwords, either using a four-digit PIN or a pattern.




Create a group password (pattern):


Repeat your group password (pin):

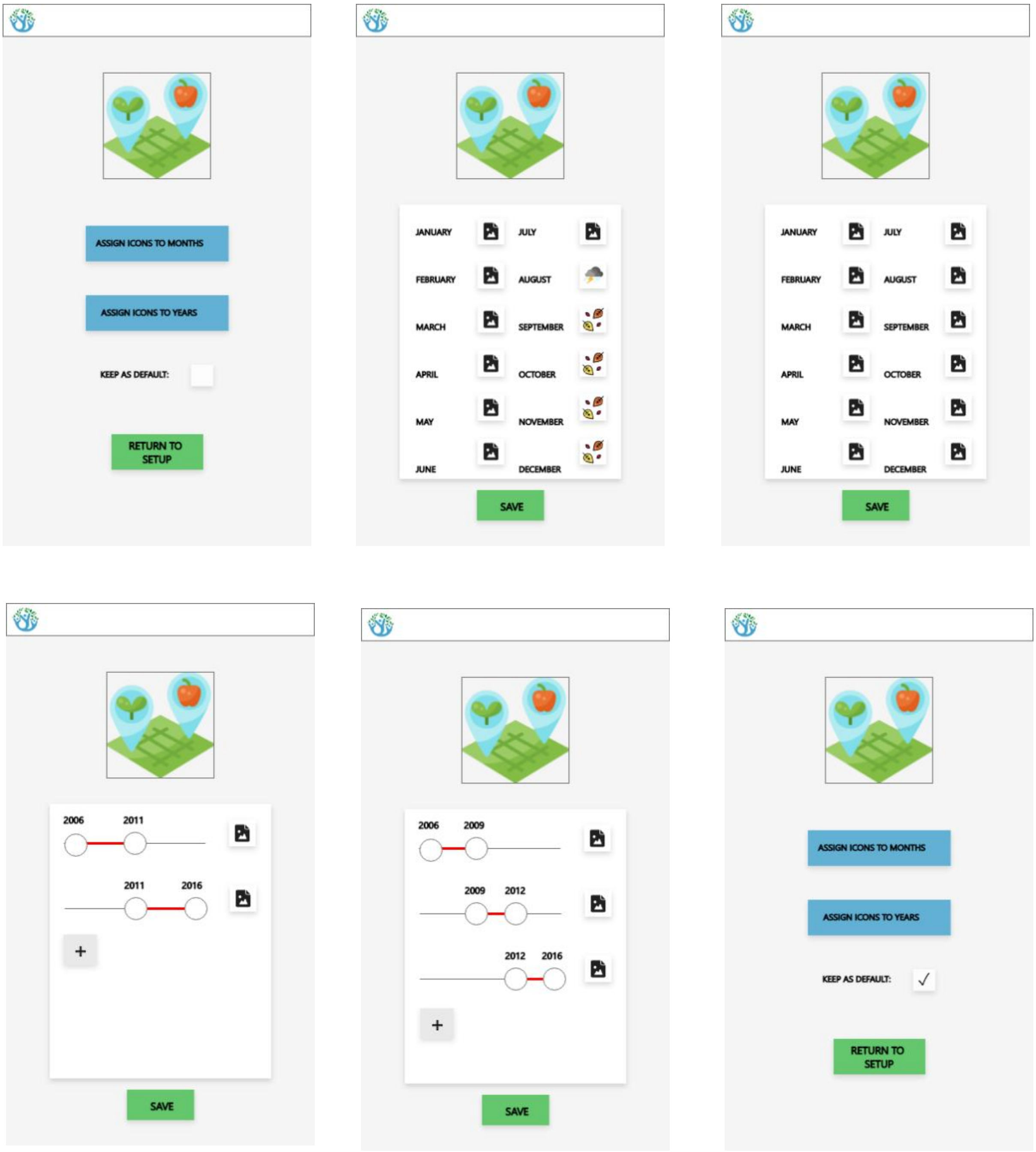

Repeat your group password (pattern):

Considerations:

- Setting up privacy settings for different user(s): If Sapelli Viewer (later versions) supports functionality to allocate specific privacy rights to specific users within a single project, then the administrator should be able to set up passwords for individual users as well as for individual projects (as detailed below). This will take place in the administrator mode.
- There are two ways to load Sapelli Collector projects: either directly from GeoKey using the credentials one sets the app up with, or manually by uploading a Sapelli Collector project to Sapelli Viewer.
- As described in the personas section many of our anticipated users are those without reliable network connections and living in rural areas far from administrative centres. They need the prototype to work offline, so they can continue their work when there is not network connectivity, but they also need to be able to connect to central servers so they can view their own data and data collected by other team members.
- *Privacy rights:* Data privacy is very important and must be considered. Mechanisms to ensure privacy that are already implemented in communities include anonymisation of participants, encrypted community devices, and specific behavioural rules decided by each community. Different access for different projects could be regulated at either the 'setting up user accounts' stage (a sign in for each user and then only able to see their contributions), or at the specific project stage (on choosing a project to view, sign in is required).

3.4 Assigning unique icons to timeframes

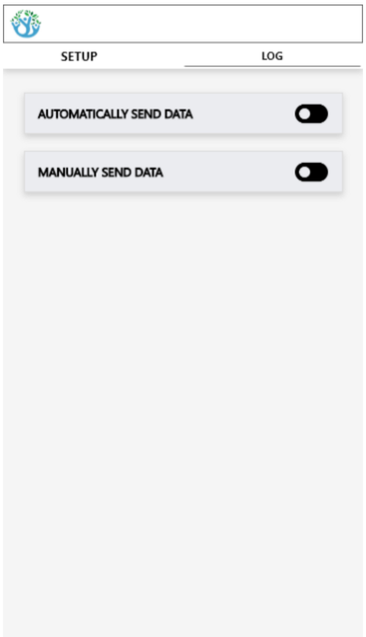
The screenshots below show how the admin user will customise the icons that represent timeframes. When the 'Assign buttons' are clicked, the file manager of the device must open to allow the admin user to select images previously downloaded and upload them in Sapelli Viewer. Sapelli Viewer should recognise as many image formats as possible.



3.5 Creating and gathering log files

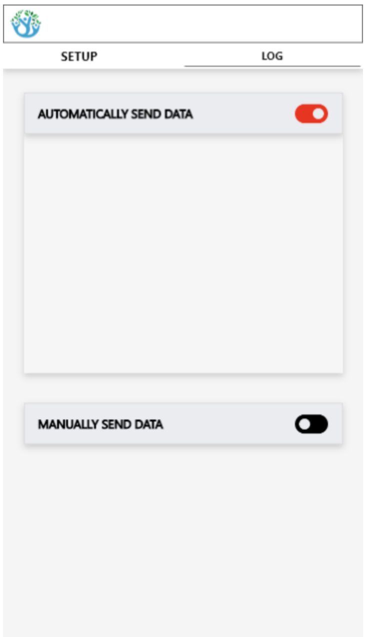
Ordinarily, ExCiteS team members collect logging statistics during usability experiments. Given the restrictions on travel and in-person research posed by the Covid-19 pandemic, we would like to place a priority on including a ‘creating log files’ function in the first iteration of the prototype so that we can track how users interact with the prototype and so that we can use this ifnromation for usability-related research purposes.

The only requirement for this functionality at this stage is that the log file is easy to export, and that the export functionality is hidden from users; i.e. only researchers that are aware of the functionality should be able to export the log file. The events to be recorded will be discussed during the development process, once the first prototype is ready. Below a screenshot of how we envision this functionality to appear in the app.



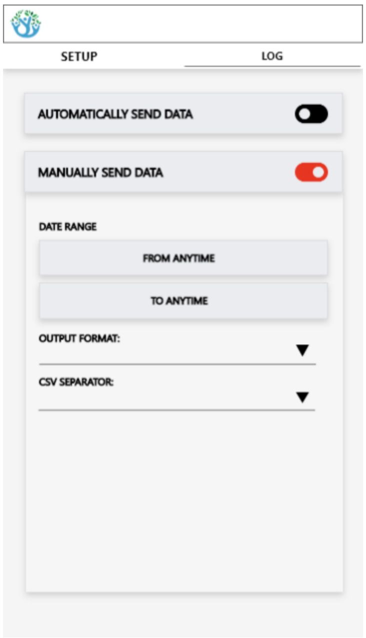
AUTOMATICALLY SEND DATA

MANUALLY SEND DATA



AUTOMATICALLY SEND DATA

MANUALLY SEND DATA



AUTOMATICALLY SEND DATA

MANUALLY SEND DATA

DATE RANGE

FROM ANYTIME

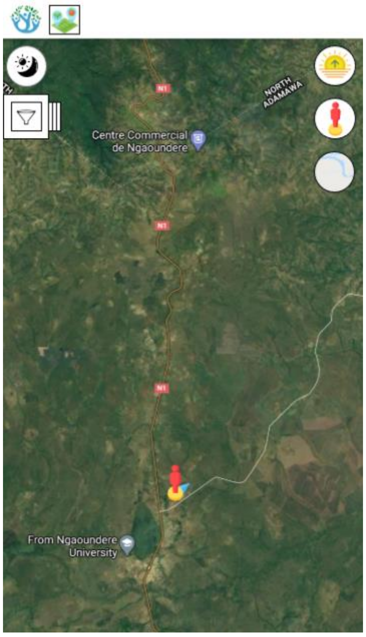
TO ANYTIME

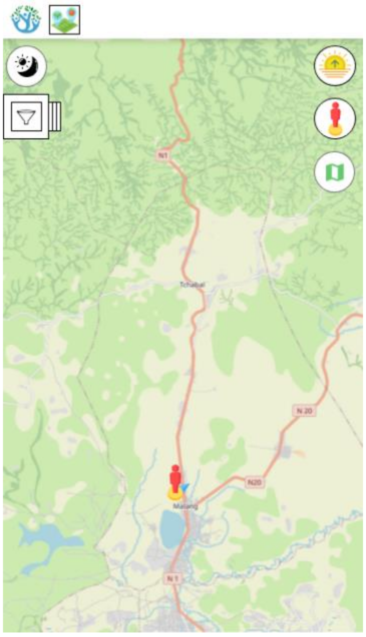
OUTPUT FORMAT: ▼

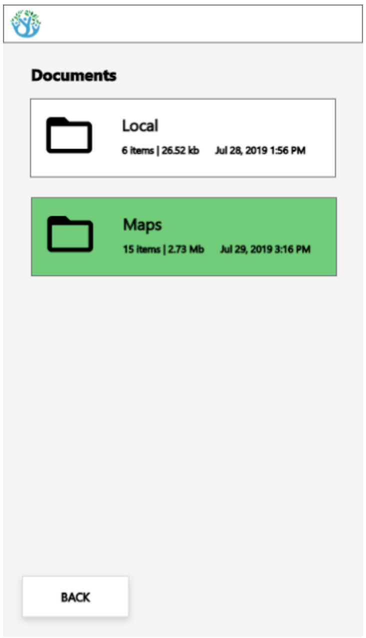
CSV SEPARATOR: ▼

3.6 Working with Basemaps

A toggle button ○ for displaying multiple basemaps must be included in the map interface. We have included three types of basemaps; a hybrid basemap with satellite imagery and map features; a physical terrain map (e.g. OpenStreetMap); a custom basemap loaded from local storage or SD card (this may be used to load hand-drawn maps by the communities themselves).





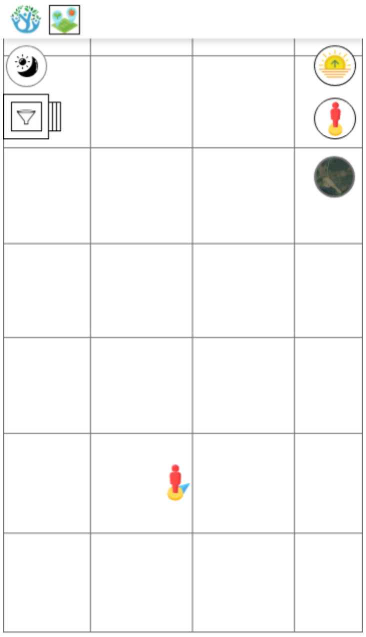


Documents

Local
6 items | 26.52 kb Jul 28, 2019 1:56 PM

Maps
15 items | 2.73 Mb Jul 29, 2019 3:16 PM

BACK



Downloading map tiles: The map interface must include a 'hidden button' to cache map tiles(~40 sqkm max). One option is to include a button that is only visible after clicking (e.g. 5 times) another visible button of the map interface. Once this 'download tiles' button is displayed and clicked, all tiles in the canvas extent will be downloaded, and the 'download tiles' button will automatically disappear. The option of loading satellite imagery 'manually' while offline should also be implemented in further versions of the prototype.

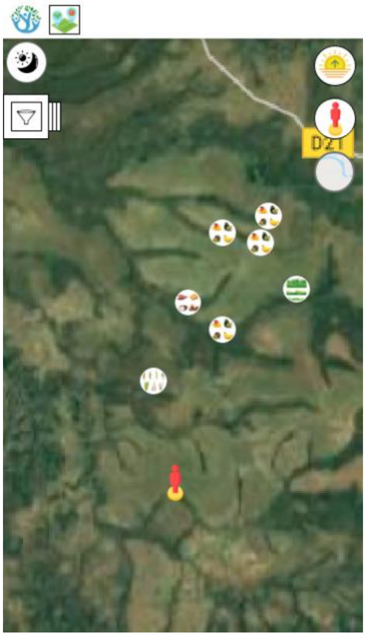
Loading basemaps from local storage: Loading basemaps from local storage is a requirement for two main reasons. First, downloading or caching tiles (e.g. Google Hybrid Map or OSM) is only possible while online. Second, in some contexts, non-publicly available maps such as georeferenced hand-drawn maps will be required. We acknowledge that this functionality will require further discussion with the development team as details on what is the most appropriate technical approach - both in terms of ease of use and performance - to achieve this are not provided in this section.

3.7 Visualising point data

Viewing data: Users should be able to view the data they collect with Sapelli Collector as individual points on the map (or clustered when too cluttered at different scales). This can be achieved by connecting Sapelli Viewer with GeoKey.

The pictorial icons that are used to visualise individual points on the map should be the same as the icons that are loaded in Sapelli Collector to support the data collection process (i.e. leaf node items).

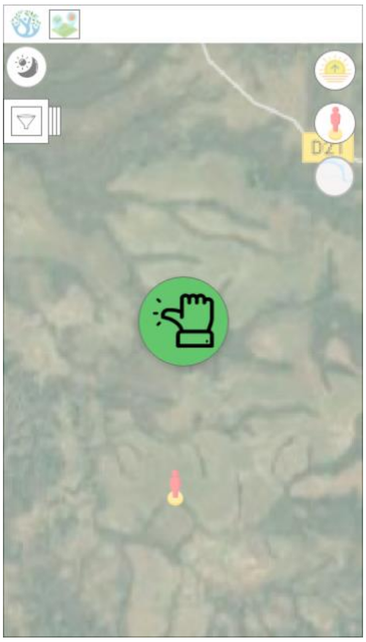
By default the user should be able to view all the data that are being collected within the project (if there are no privacy and security issues that restrict this functionality) or all the data that the specific user has collected.



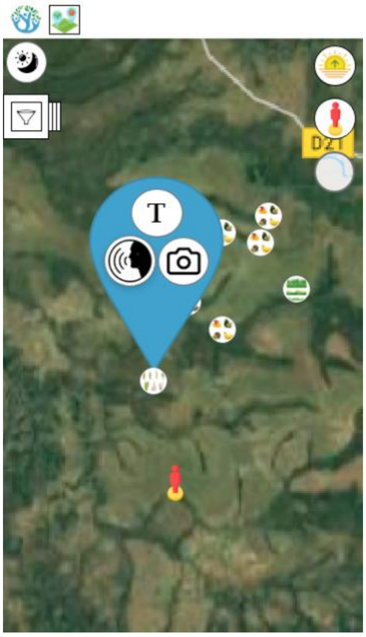
Clicking ☾ symbol will open the filter by date functionality, either overlaid in the map or as a new screen, as shown in section 2.2.6.3.

Notes:

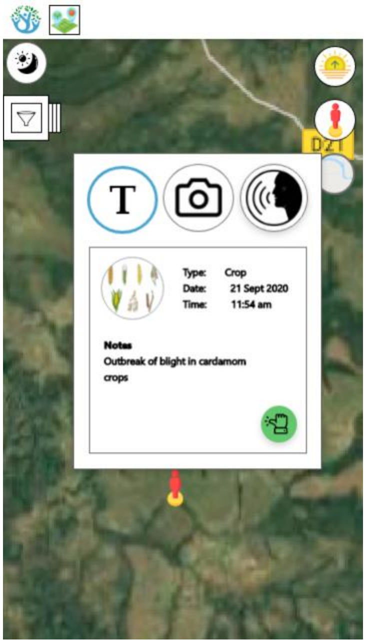
- Since the screen interface of the mobile device provides limited space, we are not keen on using thematic mapping, which would subsequently require a legend in order to read the map and the information it shows.



No data: If no data have been collected yet, hence there is nothing to show on the map, a message/symbol/icon should be displayed to indicate there is no data.



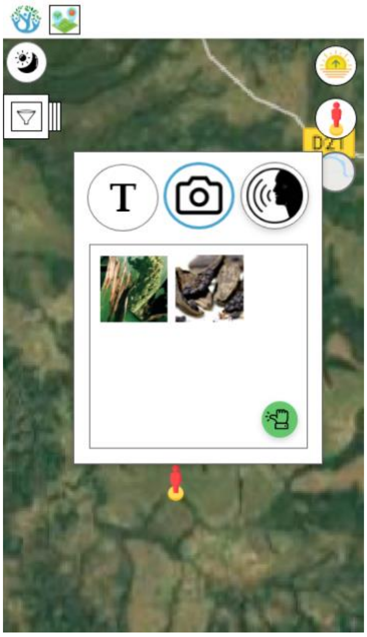
Interacting with data: When the user clicks on the icon of a data point (e.g. 🌾), a popup shows details of the data item together with a symbol describing if an audio/image file is attached to this point.



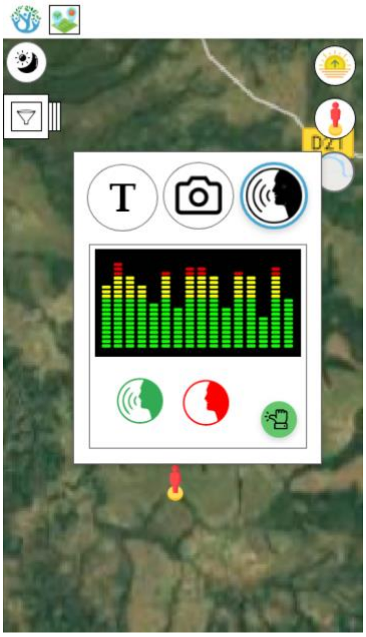
Type: Crop
Date: 21 Sept 2020
Time: 11:54 am

Notes
Outbreak of blight in cardamom crops

Viewing contribution details and text added: For each point shown on the map, a popup window will display (when the data item icon is clicked) whether there is any attribute data or semantic information associated with it. A symbol should be used to represent the availability of text or contribution data (the latter will always be present and include 'type' (the leaf node value), and 'date' and 'time' recorded. A small back button appears in the bottom right that takes the user back to the main map interface.



Viewing images for selected data points: For each point shown on the map a popup window will display (when the data item icon is clicked) whether there is any attribute data or semantic information associated with it. A symbol should be used to represent the availability of an image file. When the user clicks on the image symbol the image should be displayed. A small back button appears in the bottom right that takes the user back to the main map interface.



Listening to audio for selected data points: For each point shown on the map a popup window will display (when the data item icon is clicked) whether there is any attribute data or semantic information associated with it. The icon used to represent the availability of an audio file must act and look as a button that, when clicked, the audio file is played and also a menu to pause and exit the recording must be displayed. A small back button appears in the bottom right that takes the user back to the main map interface.

The menu interface should be displayed on top of the map and contain play/pause and exit buttons and a slide bar, which is important to show the user that audio is playing (in case the device's sound is muted) and the duration of the recording. While the menu is displayed on top of the map, opacity should be applied to the map to make clear that the map interactions are disabled, and the only interaction enabled in the audio menu.

Considerations for viewing image files:

- Size of image: When pressing on the photo it goes to full screen
- Pan & zoom an image: If full screen, zooming is possible by using the zooming mechanisms suggested in the zoom section below. Panning is then possible by moving with fingers
- Rotating an image: By rotating the device, the photo rotates which may be useful for landscape photos
- In the interest of storage space, image files will need to be in low quality (but obviously good enough to be recognisable), which may be problematic because photo-quality on the devices is only increasing with new models (Samsung XCover 4, a fairly old model, has 13MP camera, producing photos 3-5mb) and Sapelli currently does not have a means of reducing quality.

Other Considerations:

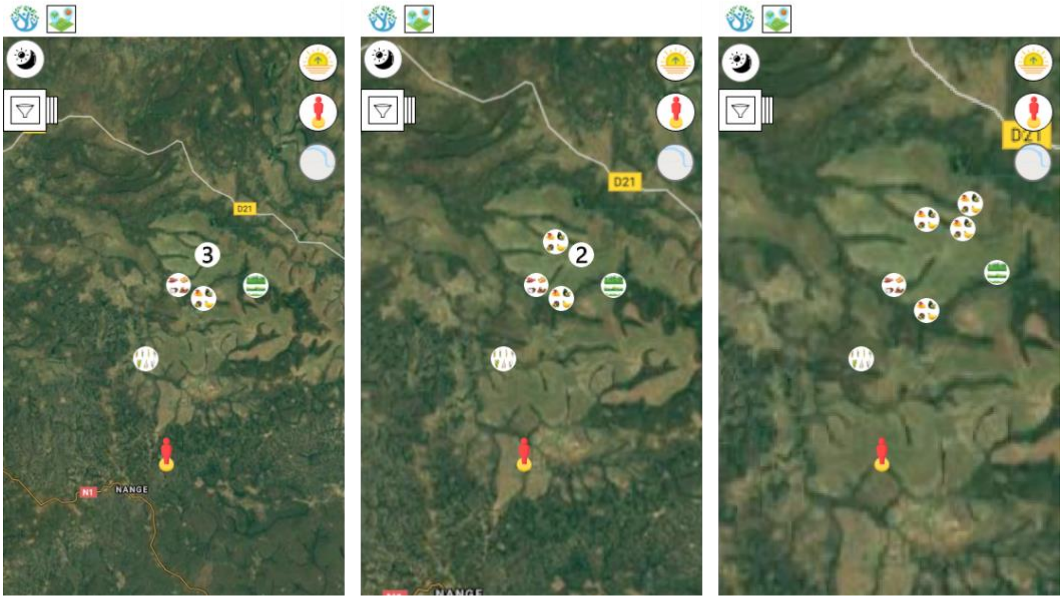
- Our collaborators have requested having the option to collect lines or polygons (either manually or using GPS tracing or waypoints), but we anticipate that this will only be possible in later versions of Sapelli Viewer. This version will only support the displaying of point data.
- *Viewing data offline:* We would like to explore the possibility of providing the user with the option to view data that have been collecting while in Sapelli Collector offline mode (i.e. viewing points stored individually on the phone before an Internet connection is established to store data on GeoKey).
- If Sapelli Viewer later supports functionality to allocate specific privacy rights to specific users within a single project then visualisation options should be in place to support viewing data collected by e.g. the user herself, the group of users with same privacy rights, the rest of the project (this will be further discussed as an option with the app developers).

3.8 Clustering and visualising multiple points

This feature will be tested in the field. As a specification for this version of Sapelli Viewer we anticipate the delivery of the two following options.

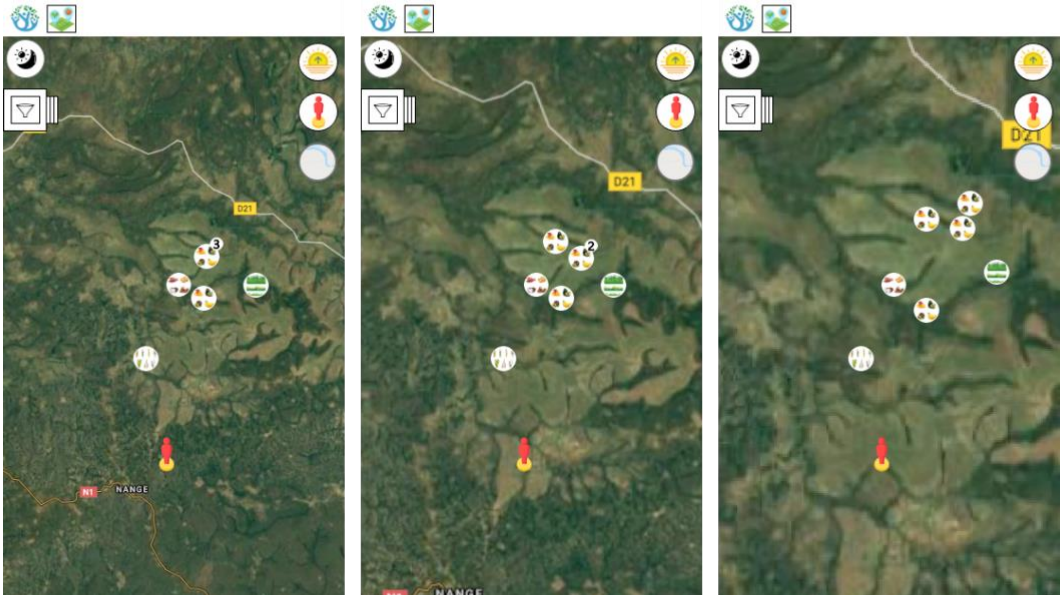
Option 1 – Numbers (version 1)

Aggregating multiple points with numbers and individual points with the icon used in Sapelli Collector.



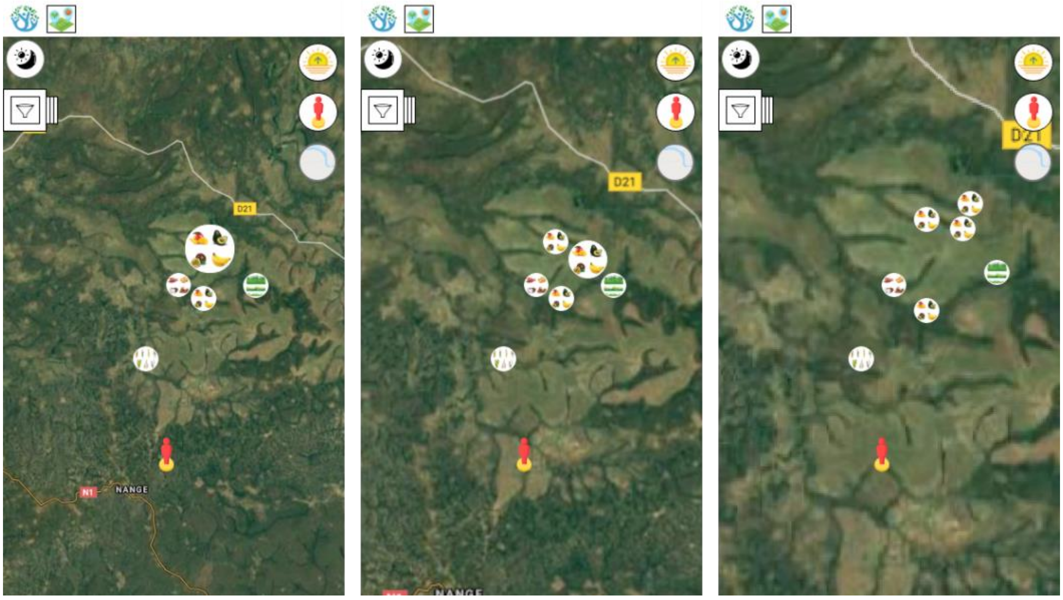
Option 2 – Numbers (version 2)

Aggregating multiple points with numbers; but here, hovering above the icon used in Sapelli Collector. If data points include multiple values, then a number will appear can appear.



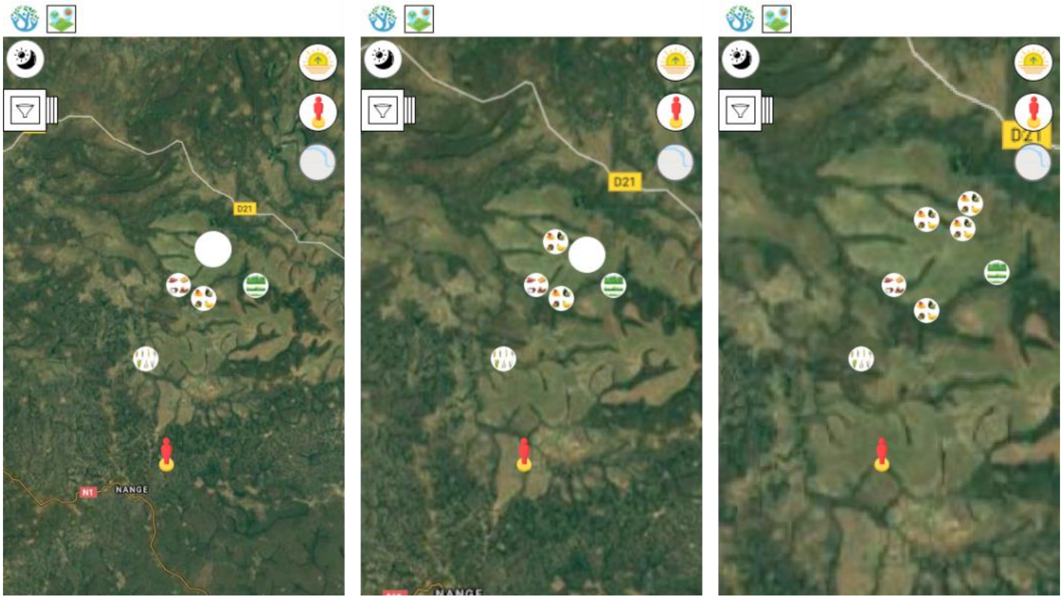
Option 3 – Scaled icons

Aggregating multiple points and displaying larger/smaller icons to indicate more or fewer data points.



Option 4 – Custom 'numbers'

Allowing the user to upload their own number category icons (e.g. hand signs used to indicate 1, 2, or many).



For all 4 options: When the user taps the number or icon, the data points will be displayed either:

1. In a spiral, with the oldest entry at the centre and the newest at the end of the spiral, or
2. In further clusters organised by type with a number in the top right corner*

3.9 Filtering Contributions

The ‘Choice’ element in Sapelli Collector allows for the creation of decision trees that range from very simple decision trees with few elements, to large decision trees with multiple root elements and many parent and leaf nodes - note the scroll function in Sapelli Collector. This, together with the fact that parent nodes (not only leaf nodes) can also play a role in the filtering by attribute functionality, poses various challenges for designing the filtering interface. We’ve agreed to tackle this design challenge by using an incremental approach. Below are included the key requirements of the first version of the Sapelli Viewer prototype:

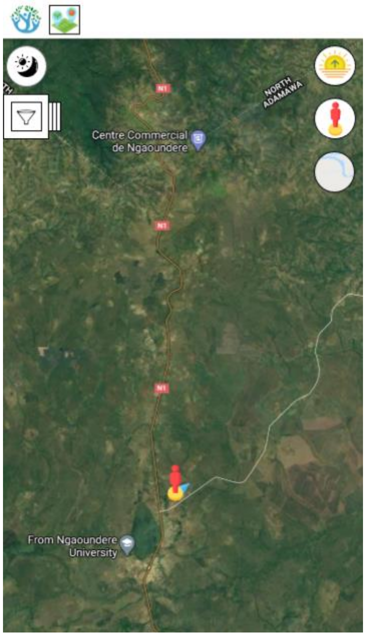
- The prototype should allow the user to filter contributions by selecting one or multiple leaf node icons of the Sapelli project, not parent nodes. These icons should be displayed in the left side of the map and must allow the user to scroll down in cases where the number of icons (leaf nodes) is, e.g., eight or more. This set of icons should not be displayed by default. Instead, the map interface must include a button (e.g. top-left funnel icon) that when clicked the icons of the leaf nodes of the project will be displayed.
- The map view must update as the user selects/deselects icons in the filtering menu, i.e., the zoom level and map boundaries must be updated in a way that all points in the map with the attributes selected must be visible when filters are applied.
- The button to ‘unfold’ the leaf node icons must be always visible, also when icons are ‘unfolded’, so the user can fold/unfold while keeping the filters on, in which case the button must highlight that filters are on (e.g. change background colour of button).
- A button to clear all filters must be added in the ‘unfolded menu’.

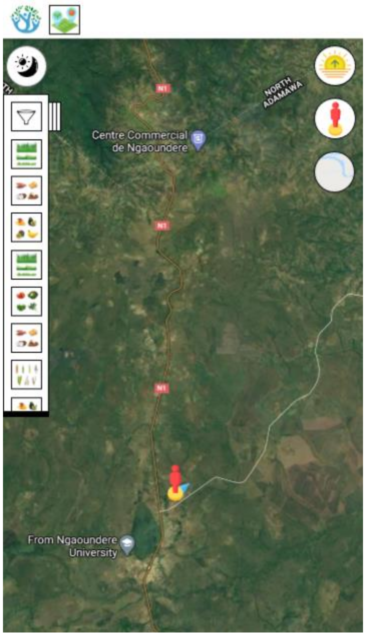
Once the basic filtering by attribute functionality described above has been developed, the next step will be to include parent nodes in the filtering options. We have included some screenshots as an option 2, but we acknowledge that there are a wide range of design issues that need further discussion and we look forward to suggestions.

3.10 Filtering by category

For the filtering options we anticipate the development of the following options which will be further tested in the field.

Option 1 – Filtering by attribute (leaf nodes only)

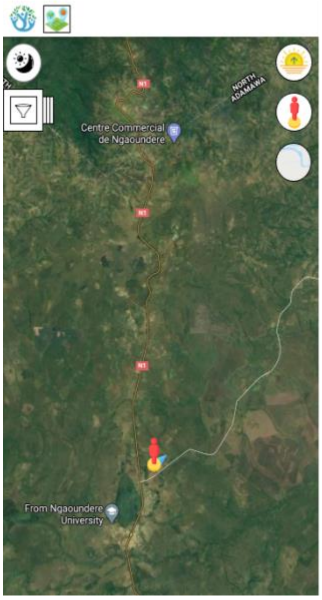


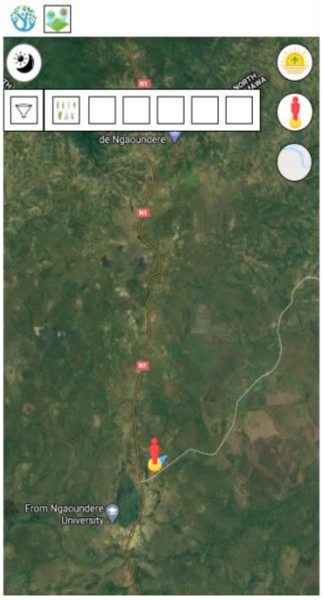


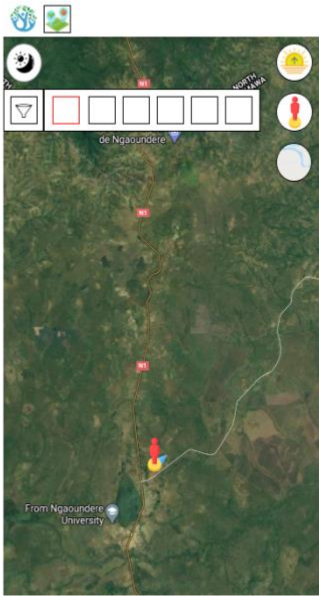
Leaf node data items appear on the left on a vertical menu.

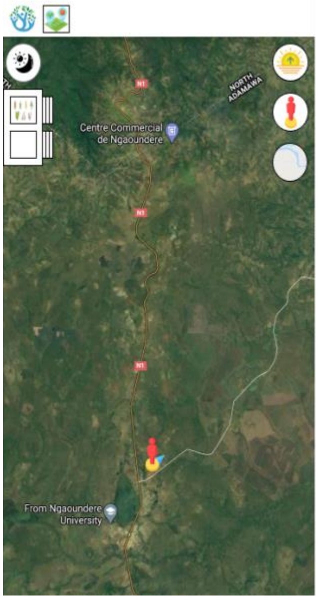
Option 2 – Filtering with a hierarchical navigation structure (all choices)

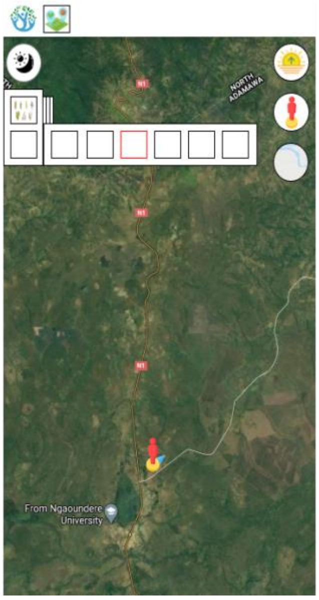
Leaf node data items appear on the top in a horizontal menu.

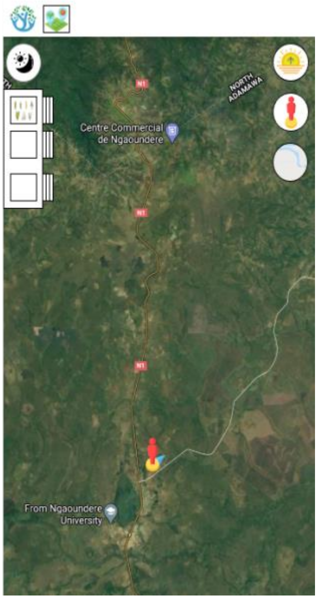


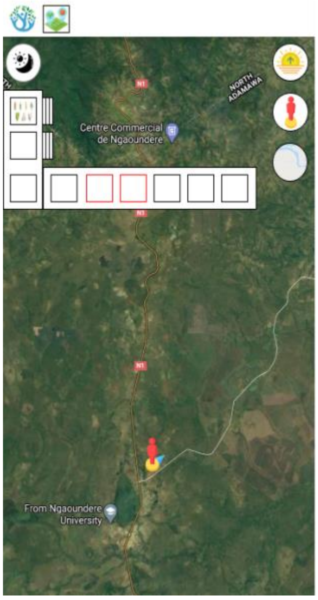


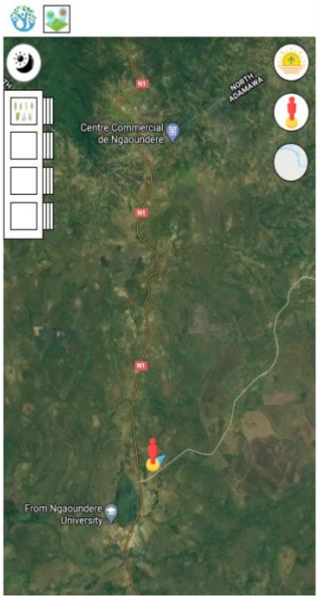


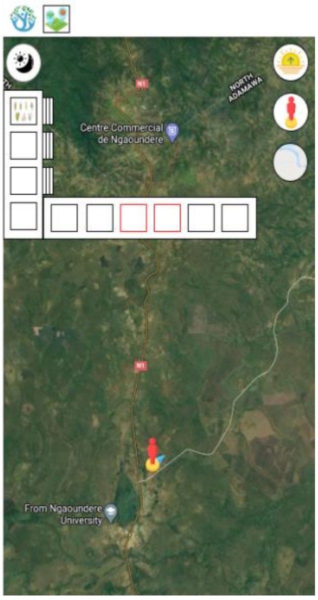

N1
NORTH ADAMAWA
Centre Commercial de Ngaoundere
N1
N1
From Ngaoundere University


N1
NORTH ADAMAWA
N1
N1
From Ngaoundere University


N1
NORTH ADAMAWA
Centre Commercial de Ngaoundere
N1
N1
From Ngaoundere University

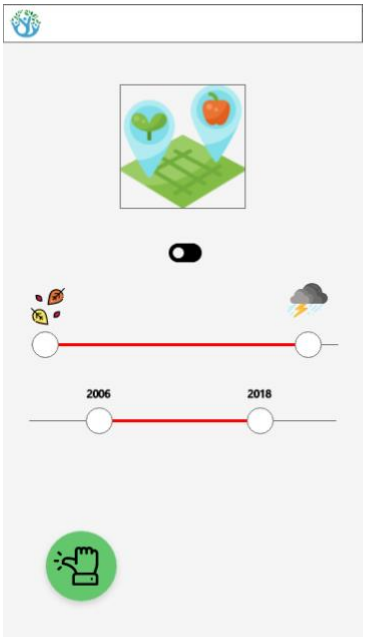

N1
NORTH ADAMAWA
Centre Commercial de Ngaoundere
N1
From Ngaoundere University

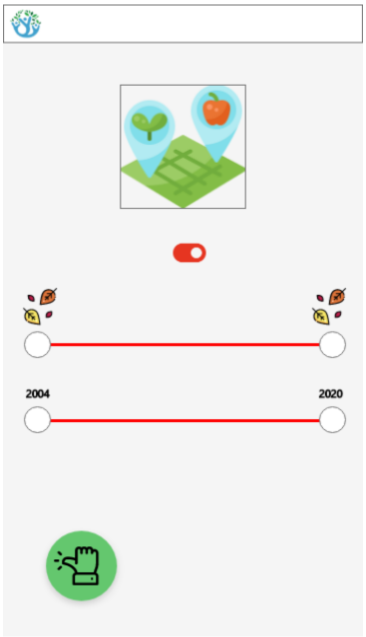

N1
NORTH ADAMAWA
Centre Commercial de Ngaoundere
N1
N1
From Ngaoundere University


N1
NORTH ADAMAWA
Centre Commercial de Ngaoundere
N1
From Ngaoundere University

3.11 Filtering by time

Filtering by time functionality must also be implemented in the first version of the prototype. We've discussed how to design this functionality and agreed that the main challenge is that days/weeks/months/years are not necessarily intuitive or meaningful units to measure time. Thus, the filtering by date functionality must be designed in a way that can be easily adapted to different contexts.

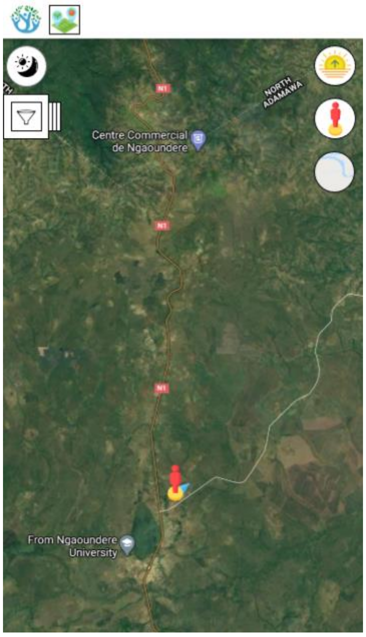




3.12 Visualising one's GPS position on the map

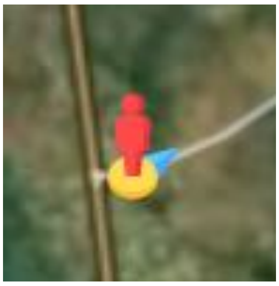
The literature shows that visualising one's current location in a map while interacting with a mapping interface is key for orientation and understanding the satellite imagery or thematic maps used as basemaps, and therefore to analyse the collected data.

Activating GPS: The user should be prompted to activate the GPS when the map opens, without any previous interaction in the map. Another option could be to prompt the user to activate it only when the recenter button is clicked.



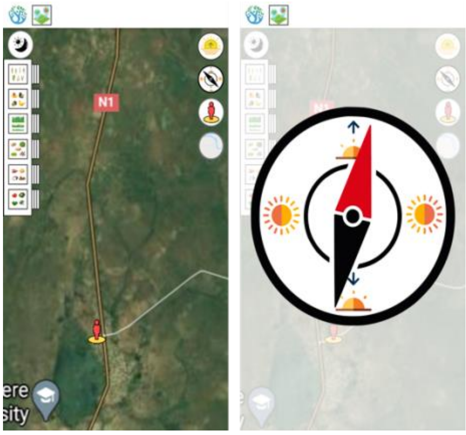
Show user's location on the map: Once GPS is activated the user should be able to view his/her location as the interface below shows. The prototype must refresh (or 'watch') the device's location both for updating the position on the map if the user moves and also to update the visualisation until maximum precision is found. While the device is searching the location and improving the accuracy, this should be represented in the map as a buffer and not as an inaccurate point in the map. For instance, while accuracy is 200m, then a buffer of 200m should be displayed, then a 100m buffer… until a good enough precision is obtained and then a point in the map is shown.

Showing the direction the user is facing: We have included two possible options for showing the direction the user is facing while interacting with the map interface.



Option 1 – An arrow on the person icon

An arrow indicates the direction the user is facing.



Option 2 – A compass (with sun icons in place of NSEW)

A compass, accessible through the top right dashboard, indicates which direction the user is facing. Standard NSEW symbols are replaced here with sunrise, sunset, and strength of sun icons and a red arrow indicates the direction. New designs will be proposed after field testing.

Recentering the map: The map interface must include a button (📍) to recenter the map to the current GPS position. Appropriate zoom level when the recenter button is clicked needs to be discussed.

Icons: In order to allow field researchers to test multiple icons for each functionality (not only GPS), a short tutorial for field researchers on how to replace such icons (i.e. icons folder's path) and generate a new .apk file should be provided.

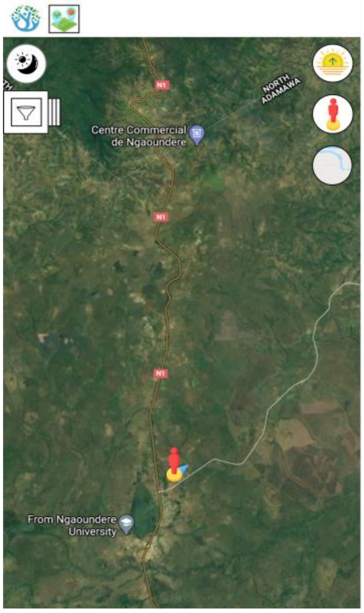
3.13 Zooming, panning, and orientation

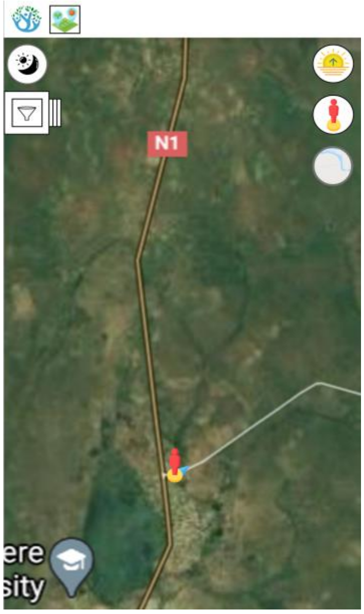
Zooming has been brought up by community members as a useful feature in order to interact with the map interface and inspect photos that participants have taken in more detail. Flexibility in orientation can be provided to facilitate different ways of viewing the map. Whilst initial experiments suggested that the pinching with thumb and finger method worked well for zooming, we would like to test three different methods:

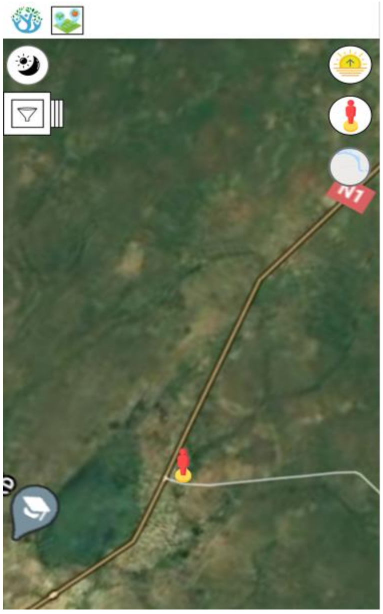
1. Pinch with thumb and finger to zoom in and out;
2. Double tap to zoom in and out by a defined amount (2x);
3. Icons for zooming in and out - these should be an icon of a big tree for zooming in and an icon of a small tree for zooming out.

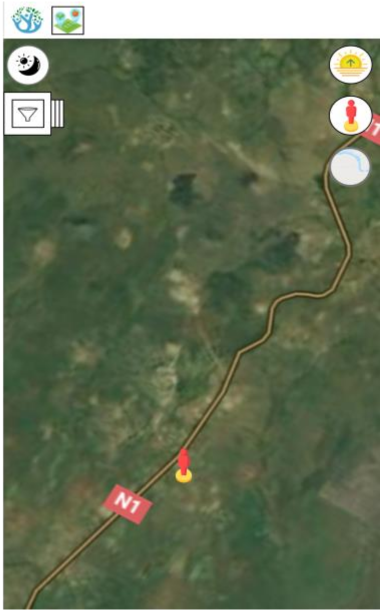
With respect to orientation and navigation:

1. For map rotation, we would like to prototype two options: First, the ability to rotate the map interface, in which case there will be a requirement for a button (compass icon) to re-orientate the map to North or East. Second, the map does not rotate and only a static rose (standard or customised) is included in the map (e.g. top-right).
2. For device's orientation, we'd also like to explore the appropriateness of representing in the map where the device is pointing at. Though Sapelli Viewer is not a navigation app, representing the device's orientation might help in the image and map interpretation process.









4. Optional Functionality for Version 1.0

This optional functionality proposed here will be discussed with the developers.

Sapelli Collector currently allows users to contribute data, but not to either view or edit the data they collect. Many of our collaborators work with local and regional partners, who view the data collected and take necessary actions - from investigating incidents of poaching, to analysing the health of given ecosystems.

Contributors have therefore expressed the need to be able to not only view and interact with the data they gather as has been described across the document, but also to be able to edit the data they have contributed or to query where they think there are errors in contributions made by other users. Though this document focuses only on the visualisation functionality, **we would like to explore the edit, reporting and adding contributions functionality.**