Journal of Geomatics Vol. 13, No. 1, April 2019

Development of universal geospatial data collection application and visualisation platform

Arpit Agarwal
Space Applications Centre, ISRO, Ahmedabad, India
Email: arpitagarwal@sac.isro.gov.in

(Received: Dec 20, 2018; in final form: May 31, 2019)

Abstract: Collection and efficient dissemination of ground truth data is a challenge faced by many research groups. Collecting data from multiple sources, transforming them to a common format, filtering the data based on parameters of interest is cumbersome. Presence of multimedia information such as images further adds to the complexity of collecting and storing the data efficiently. To address these issues, a centralised data collection tool is developed, which could collect data along with geo-spatial information. The collection tool is an Android application where a schema for data collection can be defined. Collections forms can be auto generated from the schema on the app. This data can be pushed to centralised data store. A visualisation portal is provided to view the collected data on Maps. The data can be downloaded in commonly used geospatial file formats like GeoJSON and kml.

Keywords: Data collection, Mobile App, Data Management, Web-GIS

1. Introduction

Efficient collection of ground truth has always remained a challenge for the researchers, particularly when collection and distribution is required across multiple functional groups. The available ground truth data is difficult to search /retrieve, filter, operate on and transform. There are some common traits in the ground truth data used for remote sensing applications viz. geo-spatial coordinates and time stamp. Other than that, every domain has some domain specific information to be collected. For example, ground truth of crop data may contain crop name, adjacent crops, disease information etc., while ground truth for wetland data may contain the name of wetland, its type (lake, pond, river etc.) and other related information.

An android application is developed which allows creation of multiple projects and could automatically collect the geospatial and time information. Each project can define its own fields using a schema. Based on the schema defined for a project, a data collection form is generated on the fly with geo-spatial and temporal information filled in. The information could be saved directly on servers. In cases where internet connectivity is not available, data can be collected on the device with option to sync it to the servers later.

For collection of multimedia data, multimedia asset management services are developed which allow upload/download of images to/from the servers. A special data type is defined in schema so as to enable upload of image for the specified field.

To make this data available to the users, a web interface is developed which allows the data to be viewed on maps using Web-GIS techniques. The interface allows search based on project or date of collection. It also allows a free text search over the collected data. The data collected through the app is available for download in most common and widely used formats such as GeoJSON and shape files.

2. State of the art

Various content management systems (Margaret, 2016) are available which allow creation, storage, search and visualization of data across multiple collaborating users. A content management system typically has two parts viz. a content creation/modification platform and a content delivery application. Drupal (Tomme, 2017), WordPress (Paulik, 2013) etc., are some of the commonly used content management systems.

Geospatial content management systems allow data to be stored along with spatial information viz. latitude and longitude. Google maps and map-server are examples, which are not only geospatial content management systems, but also provide a building block for other such systems.

Open Data Kit (ODK) (Ghosh and Dasgupta, 2015) provides tools to build forms (ODK Build), collect Data (ODK Collect) and a server (ODK Aggregate) which could save data to a database and provide generic visualization on the data. Mobile Data Conversion Kit can be used to export the data to GIS software like ESRI, QGIS or Google Earth.

3. System design

The system should be capable to store data with varying attributes along with geospatial information, provide efficient search over the collected data, support multiple users and possibly, multiple languages. The system should be able to handle multimedia data like images. Moreover, the collection should be possible in areas where internet connectivity is not available with the device.

Considering the challenges, a high level system design of the universal data collection infrastructure is shown in figure 1. The major components are explained below.

3.1 Web Service

The Web Service provides HTTP REST APIs. It is developed using the Java Spring Boot (projects.spring.io) framework. It uses an embedded tomcat server to listen to the requests, and does not need a separate tomcat server to be deployed.

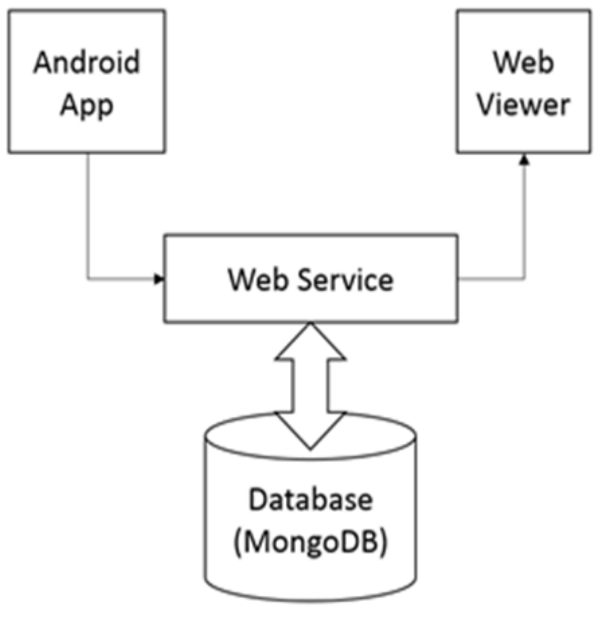

Figure 1: Architecture of universal data collection infrastructure

3.2 Database

MongoDB (Daylay, 2014) is used as database, which is a no-sql document store. As the data to be stored has variable fields, a document store seems apt for the use case. MongoDB supports UNICODE, enabling support for multiple languages. Moreover, it also provides text indices for efficient free text search. MongoDB has been proven to store millions of documents and outperform some other databases available (Bhat, 2015). The latencies generally remain less than one second.

3.3 Android app

The android app enables the user to define projects, schemas and add data. It uses the REST APIs provided by the Web Server.

3.4 Web viewer

Web Viewer enables the user to visualize the data collected using the Android App on maps using Open Layers 3. It uses the Web Server to query the requested data.

4. Multimedia asset management

Storing and transferring images is a challenge in offline stores. Images were typically stored on hard drives and the users store the path with other data elements, typically lying in a csv, excel or a DB. This way, visualisation of data along with associated image(s) becomes difficult. This also applies to data transfer. While transferring any data, images needed to be copied and provided separately.

The multimedia asset management services are developed to overcome this challenge. An image can be encoded as a string and uploaded to the server as a byte stream. The server reconstructs the image from the binary stream and stores the file on a network storage. The file path is stored in a database along with a unique identifier which is returned to the user. The user can use this identifier to download the image back using a browser whenever needed. While downloading the image, the server searches the image path in the database based on the provided identifier. Once the path is found, the image is sent in the response as byte stream, which can be rendered in the browser.

5. User interfaces

There are two main user interfaces in the universal geospatial data collection viz. Data Collection App and Data Visualization tool.

5.1 Data collection app

The data collection app is an android application. which mainly includes three pages:

(i) Project listing page

This page lists all the available projects in the system with option to edit the project or add data to the project. The projects can be saved offline for later use. In case Internet is not available, offline list can be displayed. The project listing page is shown in figure 2. Clicking on edit will take the user to schema definition page. The page also provides an option to create a new project.

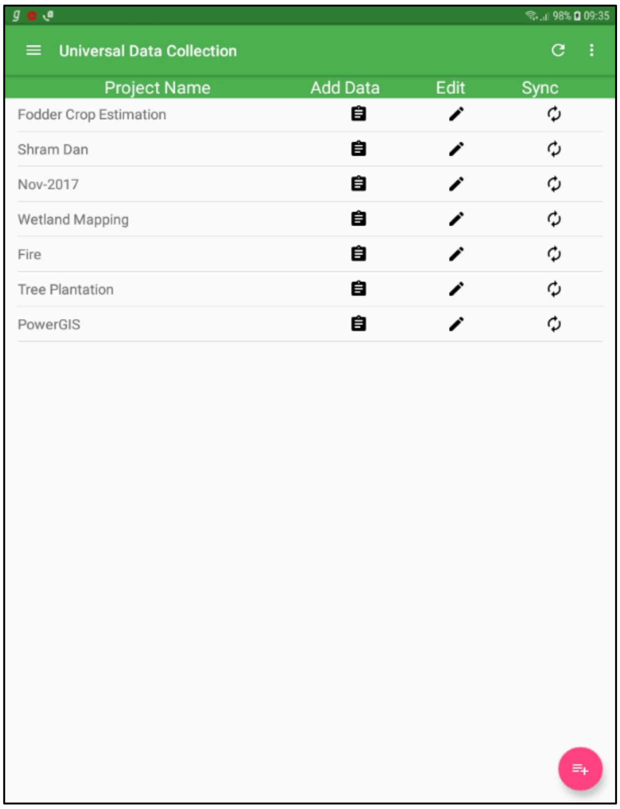

Figure 2: Project listing page

(ii) Project/Schema definition page

This page allows a user to define fields that are required to be collected as part of the project. Figure 3 shows a schema definition page for a sample project. Figure 4 shows the interface to define a field in the schema and the supported data types. A defined schema can be saved to be used later offline. This page also allows a user to edit or delete a field.

Figure 3: Schema definition page

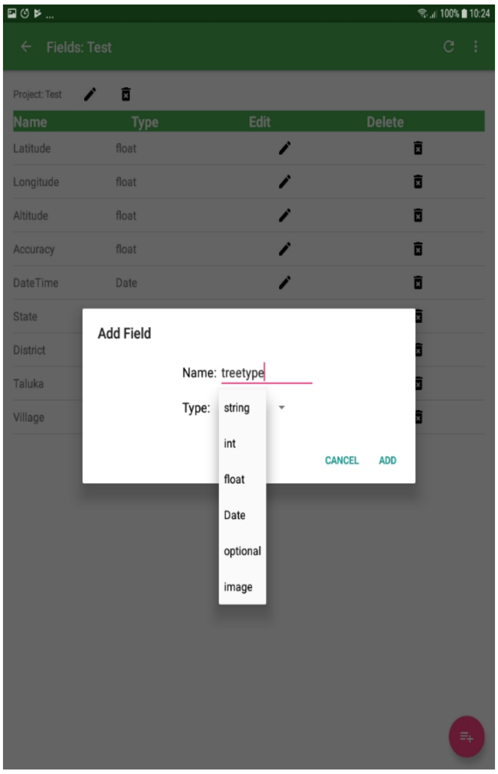

Figure 4: Field definition window

(iii) Data collection page

Based on the defined schema, this page auto-generates a form where the values corresponding to the expected fields can be filled and saved. Figure 5 shows an example of the form generated for project schema defined in figure 3. Geospatial information fields and timestamp are included as default and non-modifiable fields to every project. The geospatial fields include Latitude, Longitude, Altitude and Accuracy. The geospatial information and the current time stamp are auto populated in the respective fields. The geospatial information is read from the GPS sensor. The geospatial information can also be read from NavIC receivers making it possible to integrate the system with devices carrying NavIC receivers. In addition to the coordinates, the state, taluka and village details are also added as default and filled by the system to facilitate people to identify the place correctly. Users can modify the system suggested state, taluka and village, in case the suggestion is not correct.

The data can be directly sent to the servers if Internet is available, otherwise, it allows the data to be saved on the device with an option to sync the same later.

5.2 Data visualization tool

This is an independent web application which allows the user to visualize the project wise data on maps using Open Layers 3. It uses the same service layer as used by the android app, to retrieve the information from the database. Several Overlay layers, including administrative boundaries and satellite images, have been provided for reference. The satellite reference layers provided are RISAT, LISS 3, Bhuvan High Resolution Maps etc. Google Map is also available as a reference layer. Any layer, available as Web Map Service can be overlaid for value addition.

Figure 5: Data creation page

The data can be searched based on date of collection as well as any text string. For example, if a user searches for *mango*, all the data points which contain *mango* as a sub-string are returned. The data is plotted on the map based on geospatial coordinates. Clicking on a point shows the other information contained at that point.

Figure 6 shows the selection page for data visualization and figure 7 shows the popup which displays the information contained at a given sample point.

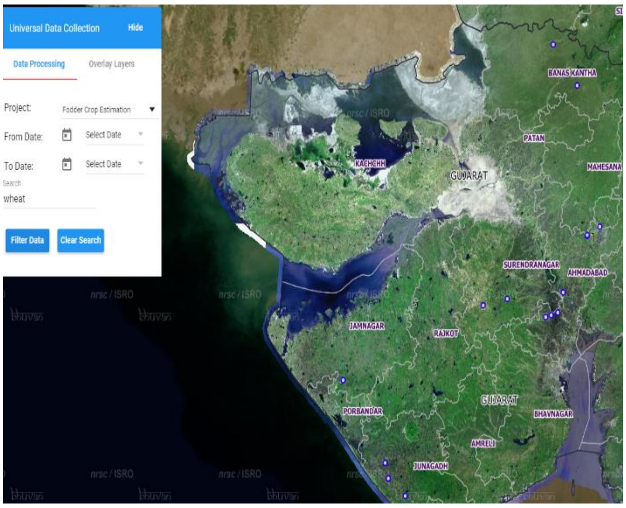

Figure 6: Content delivery application – data filtering

6. Data validation

All the data entered for a selected project is validated against the schema definition of the project. The data type of every field should match with the one defined in the schema. For example, if a user has defined a field, say quantity as integer, then filling *"abc"* for this field is considered invalid. This validation takes place both at the android app and service backend. A user should have a contributor role for the project he wants to collect and upload data for. A user can be assigned roles by the project administrator. By default, a user who creates a project, becomes its administrator, Data validation at service backend ensures that even if someone bypasses the app, and tries to call the REST end points through scripts, database is not polluted with unintended or malicious entries.

7. Deployment

The Web Server is deployed using an embedded tomcat web server. There are no CPU intensive operations and the web server can be run on commodity hardware. In case of high load, typically millions of calls per minute, RAM and networks are the first things to saturate. As MongoDB supports replication and sharing, the system can be scaled horizontally by adding more nodes.

MongoDB is set up as a service with configured paths for data and logs. Images are stored on a network storage. The images are organised based on a nested folder hierarchy based on year, month and date.

The Web Viewer is available for access at: http://vedas.sac.gov.in/vstatic_1/UDC/. The data collection android app is available at VEDAS (Visualisation of Earth Data and Archival Systems) website.
https://www.vedas.sac.gov.in/vedas/downloads/Android_Apk/Universal_Data_Collection.apk.

Figure 7: Data selection in content delivery application

8. Discussion

The app is being used by teams working in the field of agriculture and wetland mapping. It is also being used to track tree plantations by staff members at Space Applications Centre.

In future, plan is to increase the adoption by onboarding more teams and users. Work is in progress to provide authentication and authorization for data access and upload so that projects not open for public viewing can also leverage the platform.

9. Conclusion

A user friendly app has been developed for field data collection and sharing of data. An intuitive web interface is provided for data visualization. Researchers from various themes can use this platform to collaborate data among the community. The app is available in the downloads section of VEDAS website https://www.vedas.sac.gov.in/vedas.

Acknowledgements

The author sincerely thanks Shri Shashikant A Sharma, Group Head, VEDAS Research Group and Dr. Markand P Oza, Head, Content Generation and Dissemination Division at Space Applications Centre, Ahmedabad for their support, guidance and vision throughout the development process.

References

Bhat, S. (2015). High Performance Benchmarking: MongoDB and NoSQL Systems. Available: https://www.mongodb.com/blog/post/high-performance-benchmarking-mongodb-and-nosql-systems. [Accessed: 01-01-2018].

Ghosh, M. and S. Dasgupta (2015). How to use Open Data Kit (ODK): A brief tutorial, Global Change Programme-Jadavpur University Working Paper # GCP/JU/15/02.

Margaret, R. (2016). Content Management System (CMS). http://searchcontentmanagement.techtarget.com/definition/content-management-system-CMS. [Accessed: 01-01-2018].

Paulik, C. (2013). A Complete guide to WordPress content management. https://managewp.com/wordpress-content-management-complete-guide. [Accessed: 01-01-2018]

Projects.spring.io. Spring Boot. Available: https://projects.spring.io/spring-boot/. [Accessed: 01-01-2018].

Tomme, K.V. (2017). Concept: Drupal as a Content Management System. https://www.drupal.org/docs/user_guide/en/understanding-drupal.html. [Accessed: 01-01-2018].