The 57th International October Conference on Mining and Metallurgy
05 - 07 October 2026, Bor Lake, Serbia
https://ioc.irmbor.co.rs

DOI: 10.5937/IOC26093J Original Scientific Paper-Mining

# GEOLOCATION AND REAL-TIME MONITORING EXTENSIONS FOR A QR-CODE-BASED SAMPLE TRACEABILITY SYSTEM IN MINING

**Dejan Jordanović[1a], Stefan Trujić[1b], Miodrag Živković[2]**

[1]Mining and Metallurgy Institute Bor, Alberta Ajnštajna 1, 19210 Bor, Serbia
[2]Singidunum University, Danijelova 32, 11000 Belgrade, Serbia
[1a] dejan.jordanovic@irmbor.co.rs, https://orcid.org/0009-0004-7925-6398
[1b] stefan.trujic@irmbor.co.rs, https://orcid.org/0009-0003-5297-4309
[2] mzivkovic@singidunum.ac.rs, https://orcid.org/0000-0002-4351-068X

## Abstract

*A previously developed low-cost sample traceability system used QR-code labels, a Google Sheets database, and Google Apps Script automation to record status transitions through mobile shortcuts on Android and iPhone [1]. This paper extends that system with three enhancements for mining and geological field environments: GPS-based geolocation captured at each status update, a real-time dashboard visualizing sample status and location, and an offline-first workflow that queues scans for later synchronization. These extensions address the original system's key limitations, unreliable connectivity in remote areas and a lack of spatial and managerial overview, supporting practical deployment in mining operations.*

***Keywords**: sample traceability, QR codes, geolocation, offline synchronization, monitoring dashboard*

## 1. INTRODUCTION

Reliable traceability of field samples is a recurring requirement in mining and geological workflows, where specimens move through several stages before results become available. A previous study proposed a low-cost traceability system that combines QR-code labels, a Google Sheets cloud database, and Google Apps Script automation to record status transitions [1]. Field personnel update a sample's status by scanning a QR code with a predefined mobile shortcut on Android or iPhone devices, which sends a request to a Google Apps Script endpoint and updates the corresponding spreadsheet row, without requiring a dedicated server or native application.

While effective as a general-purpose solution, applying this architecture to mining and geological exploration exposes practical limitations. First, the system records only status and, for two of its four states, a timestamp, without capturing where a change occurred, limiting reconstruction of sample movement across a site. Second, supervisors must open the raw spreadsheet for an overview of ongoing sample flows, which is not suited for quick monitoring of many concurrent boreholes. Third, exploration sites and boreholes are often located in areas with unreliable or absent mobile network coverage, preventing the mobile shortcut from completing its HTTP request at the moment of scanning.

This paper proposes three complementary extensions aimed at mining and geological field conditions: (I) GPS-based geolocation capture at every status update, (II) a real-time monitoring dashboard visualizing sample status and location without accessing the spreadsheet, and (III) an offline-first workflow that queues scans locally and synchronizes once connectivity is restored. For each extension, concrete implementation steps are described for upgrading the existing spreadsheet, Apps Script backend, and mobile shortcuts.

## 2. SYSTEM ARCHITECTURE EXTENSIONS

The existing system is implemented in a single Google Sheets spreadsheet (Uzorci_DISPEC_PRIJEM), in which every row represents one sample, identified by a Uzorak ID composed of the project, borehole, and depth interval (e.g., Metro-BP1-10.0-12.0). Each row also stores the requested analysis, a single Status column (SPAKOVANO/Packed, PRIMLJENO/Received, U RADU/In Progress, ZAVRŠENO/ Completed), a free-text note, and a QR-code image generated from the Uzorak ID via the quickchart.io web service. Only two columns carry a timestamp — Datum slanja (BG) and Datum prijema (BOR) — leaving the In Progress and Completed states untimestamped. Table 1 summarizes this schema alongside the columns proposed below.

**Table 1.** *Existing columns of the Uzorci_DISPEC_PRIJEM sheet and the columns proposed by the extensions in Sections 2.1–2.3*

| Column | Content | Schema status |
|---|---|---|
| Uzorak ID | Sample identifier (Project-Borehole-From-To) | Existing |
| Projekat / Bušotina | Project name and borehole ID | Existing |
| Od / Do | Sampled depth interval (m) | Existing |
| Analiza | Requested analysis type | Existing |
| Status | SPAKOVANO / PRIMLJENO / U RADU / ZAVRŠENO | Existing |
| Datum slanja (BG) | Dispatch timestamp (Belgrade) | Existing |
| Datum prijema (BOR) | Reception timestamp (Bor) | Existing |
| Napomena | Free-text note | Existing |
| QR (slika) / QR link | QR-code image and quickchart.io link | Existing |
| Lat / Lng | Coordinates of the latest status update | Proposed – 2.1 |
| Datum: U RADU | Timestamp for the In-Progress transition | Proposed – 2.3 |
| Datum: ZAVRŠENO | Timestamp for the Completed transition | Proposed – 2.3 |

### 2.1. Geolocation capture

Geolocation capture follows the general principle of low-cost, web-service-based GPS tracking, in which a mobile client periodically reports its coordinates to a central web application rather than relying on dedicated tracking hardware [2]. Upgrading the existing sheet and its automation requires the following steps:

**Step 1:** Add two new columns, Lat and Lng, immediately after the existing QR link column, to store the coordinates captured at the moment of the most recent status update for each row (see Table 1).

**Step 2:** On Android, edit each existing HTTP Request Shortcut to include the app's built-in location variable in the request URL, appended as lat and lng query parameters alongside the existing mode and Uzorak ID parameters.

**Step 3:** On iPhone, insert a "Get Current Location" action at the start of each existing Shortcuts workflow, before the "Get Contents of URL" action, and use its output to extend the same request URL.

**Step 4:** Modify the Apps Script doGet(e) function to read e.parameter.lat and e.parameter.lng and write them into the Lat/Lng columns of the row identified by Uzorak ID — the same row-lookup already used to update Status and, where applicable, Datum slanja (BG) or Datum prijema (BOR) — on every status update, not only on SPAKOVANO (Packed) and PRIMLJENO (Received).

**Step 5:** Add a fallback rule so that a missing or zero-value GPS reading never blocks the underlying status update: Status (and, where applicable, its timestamp column) are still recorded, and only the Lat/Lng cells are left blank.

**Step 6:** As a later refinement, compare the Lat/Lng recorded at PRIMLJENO against the registered laboratory coordinates in Apps Script and highlight any sample received far outside the expected radius.

### 2.2. Real-time monitoring dashboard

The dashboard follows common real-time monitoring architectures, in which a central web layer collects status data and exposes it through a dedicated visualization front end [3]. Upgrading the existing system requires the following steps:

**Step 1:** Add a second Apps Script Web App endpoint (for example, doGet(e) triggered by a view=dashboard parameter) that reads Uzorci_DISPEC_PRIJEM and returns, for every row, Uzorak ID, Projekat, Bušotina, Status, and the last known Lat/Lng as a compact JSON array.

**Step 2:** Build a single static page, hosted through Apps Script HtmlService so that no separate server is required, that calls this endpoint and plots sample locations as markers using a free mapping library such as Leaflet.js.

**Step 3:** Add summary counters above the map for the four existing Status values (SPAKOVANO, PRIMLJENO, U RADU, ZAVRŠENO), computed client-side from the same JSON response, so supervisors see sample flow at a glance.

**Step 4:** Add automatic refresh (for example, every 30–60 seconds) so the page behaves as a live monitoring screen suitable for an office display or a manager's browser tab.

**Step 5:** Add client-side dropdown filters on the existing Projekat and Bušotina columns — for example, to isolate a single borehole such as BP1 or BP12 within the Metro project — so several concurrent projects can be monitored from one page.

**Step 6:** Deploy the dashboard as its own Apps Script Web App ("Execute as: Me", restricted access), so it can be shared as a single link without granting edit access to Uzorci_DISPEC_PRIJEM.

### 2.3. Offline-first synchronization workflow

Offline-capable field tools typically store captured data locally first and defer transmission until connectivity is available, rather than treating the network as a precondition for capture [4]. Upgrading the existing system requires the following steps:

**Step 1:** On Android, add a fallback action that triggers when the existing HTTP Request Shortcut fails: append the pending request's parameters (Uzorak ID, target status, timestamp, Lat, Lng) as one line to a local text file instead of discarding them.

**Step 2:** On iPhone, add an "If" action after "Get Contents of URL" that checks for a failed response; on failure, store the same parameters locally using an "Add to List" action.

**Step 3:** Create one additional "Sync Queue" shortcut, on both platforms, that resubmits every locally stored entry as the same HTTP GET request used for a normal scan, removing each entry only after a successful response.

**Step 4:** Extend the Apps Script doGet(e) function with a duplicate check on the Uzorak ID / target-status pair before writing a new value, so a delayed retry from the queue cannot overwrite Status, Datum slanja (BG), or Datum prijema (BOR) with a stale event; reuse the same capture-time timestamp to also populate the two new columns from Table 1, Datum: U RADU and Datum: ZAVRŠENO, closing the gap left by the original schema.

**Step 5:** Where supported (iOS Shortcuts Automations, or Android via a companion automation app), trigger the Sync Queue shortcut automatically whenever the device regains connectivity, instead of requiring a manual step.

**Step 6:** Show a short local notification ("Saved offline — will sync automatically") so operators can tell at a glance whether a scan was recorded online or queued.

## 3. RESULTS AND DISCUSSION

As the three extensions have not yet been deployed on Uzorci_DISPEC_PRIJEM, the discussion below reflects expected impact rather than measured results. Geolocation capture adds spatial context at negligible cost, with occasional missing GPS fixes as its main limitation. The monitoring dashboard removes the need to open the raw spreadsheet for a status overview, at the cost of an additional component to build and

maintain. The offline-first workflow addresses the original system's most significant practical constraint — dependency on continuous connectivity — and, as a direct consequence, also closes the existing gap between the four Status values and only two tracked timestamps, at the cost of extra logic on the mobile side and a short synchronization delay that must be clearly communicated to operators. Taken together, the three extensions preserve the original system's low-cost, serverless philosophy while targeting constraints specific to mining and geological fieldwork; a pilot deployment with quantitative evaluation of processing time, error rate, and synchronization delay is proposed as future work.

## 4. CONCLUSION

This paper presented three architectural extensions to a previously developed QR-code-based sample traceability system, together with concrete implementation steps for upgrading its existing Uzorci_DISPEC_PRIJEM spreadsheet, its Google Apps Script backend, and its existing Android and iPhone mobile shortcuts: GPS-based geolocation capture at each status update, a real-time monitoring dashboard for supervisors, and an offline-first mobile workflow for areas with unreliable connectivity. As a further benefit, the offline-first extension also closes an existing gap in the original schema, where two of the four Status values currently have no associated timestamp. Future work should focus on implementing the proposed steps in a working prototype and validating them through a pilot deployment under real exploration conditions, including a quantitative assessment of their impact on data completeness, error rate, and operator workload.

## ACKNOWLEDGEMENTS

*Ministry of Science, Technological Development and Innovation of the Republic of Serbia, Agreement on the implementation and financing of scientific research work of the National Research Institute in 2026, No. 451-03-33/2026-03/200052.*

## REFERENCES

[1] D. Jordanović, S. Trujić, M. Živković, Field Sample Traceability System Using QR Codes and the Google Sheets Platform, Proceedings of Sinteza 2026 – International Scientific Conference on Information Technology and Data Related Research, Belgrade, Serbia, 2026, pp. 315-321, doi:10.15308/Sinteza-2026-315-321.

[2] N. Chadil, A. Russameesawang, P. Keeratiwintakorn, Real-Time Tracking Management System Using GPS, GPRS and Google Earth, Proceedings of the 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Krabi, Thailand, 2008, pp. 393-396, doi:10.1109/ECTICON.2008.4600454.

[3] N. Paudel, R.C. Neupane, A General Architecture for a Real-Time Monitoring System Based on the Internet of Things, Proceedings of the 2019 3rd International Symposium on Computer Science and Intelligent Control (ISCSIC), Amsterdam, Netherlands, 2019, pp. 1-12, doi:10.1145/3386164.3387295.

[4] C. Hartung, A. Lerer, Y. Anokwa, C. Tseng, W. Brunette, G. Borriello, Open Data Kit: Tools to Build Information Services for Developing Regions, Proceedings of the 4th ACM/IEEE International Conference on Information and Communication Technologies and Development (ICTD), London, UK, 2010, pp. 1-12, doi:10.1145/2369220.2369236.