

User Requirements Specification - URS

Real-Time Event Dashboard

BAS World



Date	:	24th September 2024
Version	:	0.1
State	:	Draft
Author(s)	:	Jordi Blaas, Lisa Diepstraten, Anh Huynh, Alperen Korkmaz, Renato da Silva Pereira, Aleksandar Stoynov, Rositsa Mazhlenkova

Version history

Version	Date	Author(s)	Changes	State
0.1	24/9/2024	Jordi Blaas, Lisa Diepstraten, Anh Huynh, Alperen Korkmaz, Renato da Silva Pereira, Aleksandar Stoynov, Rositsa Mazhlenkova	An initial first draft of the URS	Draft
1.0	23/9/2024	All authors	Incorporated feedback from PO on user goals, error handling, and performance requirements.	On-going

Distribution

Version	Date	Receivers
0.1	30/9/2024	Raja Gorentla (Instructor) and Rob Eijgelshoven (PO)
1.0	25/10/2024	Raja Gorentla (Instructor) and Rob Eijgelshoven (PO)

Contents

1. Agreement with client.....	4
2. Functional requirements.....	4
3. Use cases.....	5
4. Prototype.....	10

1. Agreement with client

I am committed to creating the BAS World Real-Time Event Monitoring System, a platform designed to provide real-time monitoring of user interactions on BAS World's website. The goal is to offer detailed insights into user behaviour through the processing and visualisation of event data (e.g., clicks, form submissions) in real-time.

Key features include a real-time event feed, filtering capabilities, detailed event data visualisation, report generation, and alert mechanisms for specific triggers. The system will ensure secure and scalable data transmission, storage, and processing.

The deliverables include a user-friendly web application, a back-end API, and a scalable data storage solution. The project plan, guided by this User Requirements Specification (URS), outlines functional and nonfunctional requirements to meet client needs.

2. Functional requirements

- **FR – 01:** Dashboard users should be able to view a real-time feed of website events to observe user behaviours and monitor activity trends.
- **FR – 02:** Dashboard users should be able to access detailed information about each event, including timestamps, action types, and metadata, to analyse specific user actions.
- **FR – 03:** Dashboard users should be able to filter events by type (e.g., clicks, form submissions) and date range to focus on the most relevant interactions.
- **FR – 04:** Dashboard users should be able to generate summary reports for data analysis to understand overall interaction trends.
- **FR – 05:** Dashboard users should be able to receive notifications when predefined conditions, like high traffic or error spikes, are met.
- **FR – 06:** Data must be transmitted securely over HTTPS and stored with encryption to maintain data integrity.
- **FR – 07:** The system should handle a high volume of events per day without significant lag to ensure reliable real-time monitoring.
- **FR – 08:** The dashboard should be intuitive and responsive across devices.

3. Use cases

UC - 01: View Real-Time Event Feed

Actor: Dashboard User (BAS World Analyst)

Goal: Observe real-time event updates to monitor live user behaviour.

Main Success Scenario:

1. The actor accesses the dashboard.
2. The system verifies if there is sufficient recent data to be shown
3. The system displays a real-time feed of events with timestamps and metadata.
4. The actor views the feed without needing to refresh the page.

Extensions:

3a: No events are available.

1. *The system displays a "No Events Found" message.*
2. *Return to MSS step 1*

3b: Event data is corrupt or unreadable.

1. *The system notifies the actor, skips corrupted entries, and logs the issue for review.*
2. *Return to MSS step 1*

UC - 02: View Event Details

Actor: Dashboard User

Goal: Access comprehensive details about specific user actions.

Main Success Scenario:

1. The actor clicks on an event in the real-time feed.
2. The system presents detailed information about the event, including time, action, and user details.

Extensions:

2a: Event details are incomplete or cannot be decrypted.

1. *The system displays a "Details Unavailable" message, notifies the actor, and logs the error for further investigation.*
2. *Return to MSS step 1*

UC - 03: Filter Events

Actor: Dashboard User

Goal: Narrow down visible events by specific criteria for focused analysis.

Main Success Scenario:

1. The actor selects filters for event type and date range.
2. The system verifies there is sufficient data for the selected event type.
3. The system updates the event feed to display only matching events.

Extensions:

2a: No matching events found.

1. *The system displays a "No Matching Events" message.*
2. *Return to MSS step 1*

2b: Filtered data fails to load due to corruption.

1. *The system notifies the actor of data issues and logs the error.*
2. *Return to MSS step 1*

UC - 04: Generate Event Reports

Actor: Dashboard User

Goal: Create reports summarizing interaction trends for data-driven insights.

Main Success Scenario:

1. The actor requests an event summary report.

2. The system generates a report summarising key metrics (e.g., most common actions, peak activity times).

Extensions:

2a: The report fails to generate.

1. *The system notifies the actor, logs the issue, and suggests a retry option.*
2. *Return to MSS step 1*

UC - 05: Receive Real-Time Notifications

Actor: Dashboard User

Goal: Stay informed of critical activity changes or error conditions.

Main Success Scenario:

1. The system detects that a predefined condition (e.g., high traffic) has been met.
2. The actor receives a notification on the dashboard.

Extensions:

2a: The notification fails to be sent.

1. *The system retries sending the notification and logs the error.*
2. *Return to MSS step 1*

UC - 06: Secure Event Data Transmission and Storage

Actor: System

Goal: Ensure data security during transmission and storage.

Main Success Scenario:

1. The system encrypts data before storing it.
2. The system transmits event data over a secure HTTPS connection.

Extensions:

2a: Data transmission fails or encounters errors.

1. *The system retries transmission, logs the failure and alerts the support team.*
2. *Return to MSS step 1*

UC - 07: Monitor System Scalability

Actor: System

Goal: Handle real-time updates and prevent lag under high traffic.

Main Success Scenario:

1. The system processes and displays events in real time under heavy traffic.
2. The analyst experiences no noticeable delay in real-time updates on the dashboard.

Extensions:

2a: The system experiences lag or delay due to heavy traffic.

1. *The system logs the issue, prioritizes critical event updates, and notifies the support team.*
2. *Return to MSS step 1*

UC - 08: Navigate and Interact with Dashboard

Actor: Dashboard User

Goal: Ensure responsive interaction across devices.

Main Success Scenario:

1. The actor accesses the dashboard from a device (e.g., desktop, tablet, phone).
2. The system ensures the data is recent and sufficient and transmits it to the dashboard to be shown.
3. The dashboard adapts to the device, and the user can interact with all available features (e.g., real-time feed, filters, reports).

Extensions:

2a: The dashboard fails to load properly on the device.

1. *The system reloads the interface and logs persistent issues.*
2. *Return to MSS step 1*

UC - 09: System Performance Benchmarks

Actor: System

Goal: Ensure the dashboard meets performance expectations.

Main Success Scenario:

1. The system can process up to 10,000 events per hour.
2. Events appear on the dashboard within 2 seconds of generation.
3. The system supports up to 100 concurrent users without performance issues.

Extensions:

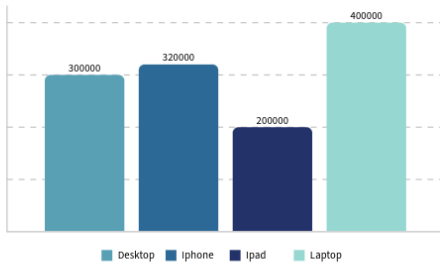
2a: Event processing exceeds response time.

1. *The system logs any performance delays and adjusts processing queues.*
2. *Return to MSS step 1*

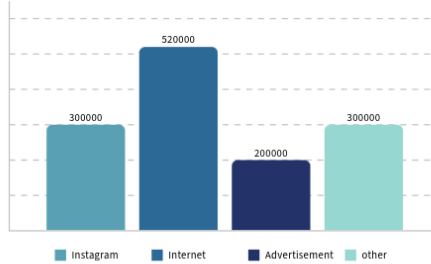
4. Prototype



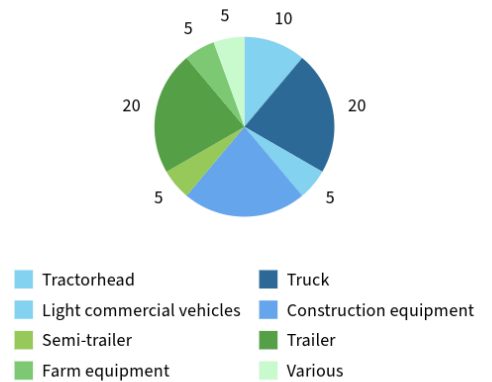
Number of sessions per device type



Number of sessions per marketing source

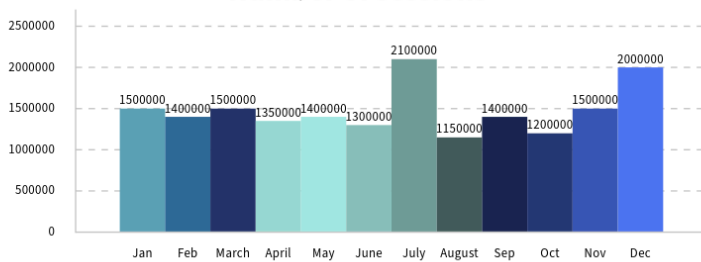


Visited product categories



January V

Number of sessions



Location	viewed product
Breda	Benalu ben
Eindhoven	Caterpillar
Eindhoven	Caterpillar
Eindhoven	Caterpillar
Eindhoven	Caterpillar