**Project Plan**



***Key Watch***

*BAS World*

|  |
| --- |
| **Date** **16/09/2024** |
| **Version** **0.2** |
| **State** **Draft** |
| **Author** **Nameless** |

#### Version history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
| 0.1 | 09/09/2024 | Nameless | First draft | Complete |
| 0.2 | 16/09/2024 | Nameless | Second draft | Complete |
|  |  |  |  |  |

**Distribution**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Receivers** |
|  |  |  |
|  |  |  |

Contents

[1. Project assignment 4](#_Toc176872041)

[1.1 Context 4](#_Toc176872042)

[1.2 Goal of the project 4](#_Toc176872043)

[1.3 Scope and preconditions 4](#_Toc176872044)

[1.4 Strategy 4](#_Toc176872045)

[1.5 Research questions and methodology 4](#_Toc176872046)

[1.6 End products 5](#_Toc176872047)

[2. Project organisation 6](#_Toc176872048)

[2.1 Stakeholders and team members 6](#_Toc176872049)

[2.2 Communication 6](#_Toc176872050)

[3. Activities and time plan 7](#_Toc176872051)

[3.1 Phases of the project 7](#_Toc176872052)

[3.2 Time plan and milestones 7](#_Toc176872053)

[4. Testing strategy and configuration management 8](#_Toc176872054)

[4.1 Testing strategy 8](#_Toc176872055)

[4.2 Test environment and required resources 8](#_Toc176872056)

[4.3 Configuration management 8](#_Toc176872057)

[5. Risk and mitigation 9](#_Toc176872058)

# Project assignment

## Context

For this project, the client is “BAS World,” a leading company that specializes in buying and selling a wide range of commercial vehicles. Their inventory spans various categories, including trucks, machinery, containers, and much more.

With such a thriving business, the company attracts hundreds to thousands of daily visitors to its website, where users interact in various ways, from browsing listings to engaging with sales processes. However, BAS World currently lacks a centralized system to effectively track and analyze these user interactions.

The solution is to develop a real-time event dashboard that captures, processes and visualizes user behavior. By simulating event generation, processing these events through an API, and displaying the data on an intuitive front-end dashboard, BAS World will gain essential insights into customer actions.

This visibility will enable the company to:

* Identify high-traffic areas of the website
* Analyze device and location data of users
* Track peak interaction times
* Pinpoint conversion points and friction areas

## Problem Statement

BAS World has added events to their website that track user actions. These events capture data such as timestamps, actions taken, device types, geographic information etc. While the system is generating these events, the data is not yet being transmitted to a central location for employees to view.

## Goal of the project

This project aims to develop a real-time event dashboard to monitor, process, and display events that capture important actions performed by visitors. This dashboard will offer insights into user interactions on the website, enabling BAS World to track and analyze event data in real time. There will be a simulated or stub system to generate and post events to a queue.

## Scope and preconditions

|  |  |
| --- | --- |
| **Inside scope:** | **Outside scope:** |
| 1. Create API or service that receives events | 1. Deployment with AWS |
| 1. A back-end system that stores and organizes data | 1. Authorization and authentication |
| 1. Front-End dashboard to displays event data in real time | 1. Reactive display regarding different devices (primarily desktop) |
| 1. Event Data Simulation | 1. Modification of data |

## Strategy

This project will be following the Agile approach using the Scrum method. This means the project will work in short-term iterations known as sprints and have regular reviews and planning meetings. The focus of the Agile methodology is on continuous improvement and delivering functional increments of the product over time. This approach will break the project into manageable pieces and prioritize critical features. At the end of each sprint, there will be a demonstration of the project's progress to the Product Owner (PO).

## Research questions and methodology

* JSON Schemas & storing JSON data in databases?

1. **What are the best practices for creating and using JSON schemas, and how can JSON data be efficiently stored in MySQL databases?**
   * **Context:**
     + Backend Development Phase (Database & Data Structuring)
   * **Why It’s Relevant:**
     + JSON is commonly used for event data due to its flexibility. Efficiently defining schemas, storing and querying JSON data in MySQL is essential for managing event data. MySQL supports JSON data types, but it's important to optimize storage and retrieval.
   * **Approach:**
     + Research the role of JSON schemas for defining structure and constraints.
     + Focus on MySQL’s JSON data type and its capabilities, such as JSON functions (e.g., JSON\_EXTRACT, JSON\_CONTAINS).
     + Study how JSON schema validation can be handled outside of MySQL (e.g., within the application layer)., especially concerning querying speed, scalability, and indexing JSON fields.
     + Test MySQL JSON storage performance using first fake event data but eventually moving to realistic event data. Experiment with JSON functions and indexing strategies and evaluate their impact on both data storage efficiency and query speed.

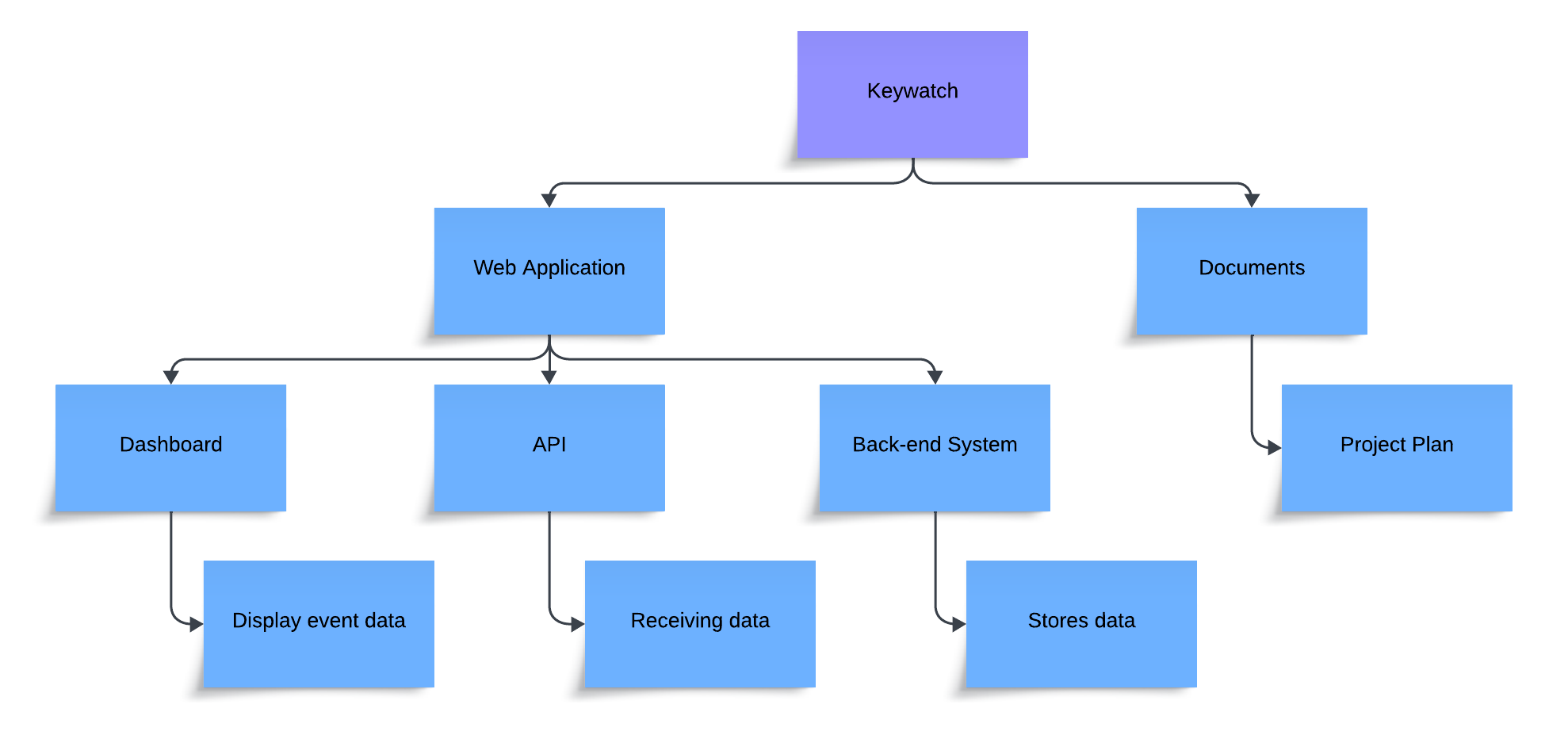
* Useful data and metrics to display in a dashboard?

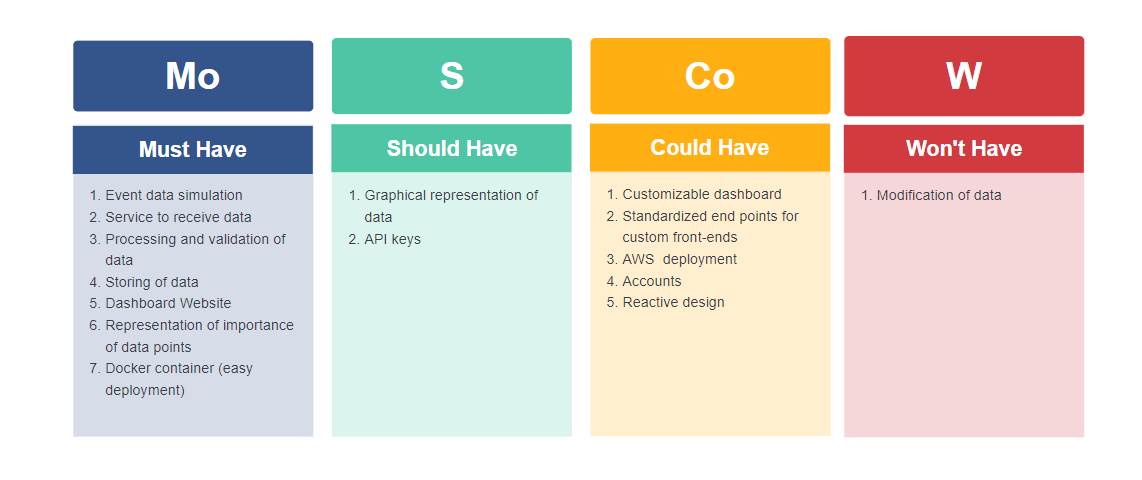
1. **What are the key metrics and data points that should be displayed on a real-time event dashboard for BAS World?**
   * **Context:**
     + Front-End Development Phase (Dashboard Design)
   * **Why It’s Relevant:**
     + A meaningful dashboard helps BAS World analyze user behavior on their website and make informed decisions. Identifying the most valuable metrics ensures the dashboard provides actionable insights into user interactions.
   * **Approach:**
     + . Investigate common metrics used in web analytics (e.g., page views, session duration, bounce rate, conversion events) as well as business-specific KPIs like clicks on key elements (e.g., product listings, contact forms).
     + Collaborate with the Product Owner to define BAS World's key goals, such as tracking user engagement, navigation patterns, and sales-related actions.
     + Use event-tracking tools like Google Analytics or open-source tools to simulate and visualize common metrics. Focus on gathering data such as:
       1. User interaction heatmaps (clicks on important buttons, images, etc.)
       2. Session duration and frequency of return visits
       3. Click-through rates for product listings
       4. Page load times and user paths through the site

* Popular dashboard designs, user features they include, etc.

1. **What are popular dashboard design trends, and what user features should a well-designed dashboard include?**
   * **Context:**
     + Front-End Development Phase (Dashboard Design)
   * **Why It’s Relevant:**
     + The dashboard must not only display data but also be user-friendly and visually engaging. By incorporating proven design trends, the dashboard will be both functional and intuitive for BAS World’s team.
   * **Approach:**
     + Study modern dashboard design trends focusing on clean, responsive designs, real-time data visualization, and interactivity.
     + Evaluate design elements like responsive layouts, color schemes for data visualization (heatmaps, graphs, and pie charts), and user interface elements that enhance the user experience (e.g., real-time updates, data filterin)
     + Use wireframing and prototyping tools (e.g., Figma) to design and test different dashboard layouts.

## End products





# Project organisation

## Stakeholders and team members

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Abbreviation** | **Role and functions** | **Availability** |
| *Rob Eijgelshoven*  *rob.eijgelshoven@basworld.com* | *R.E* | *Role: Product Owner*  *Function: Engineering Manager* | *Mail: Every Business Day*  *Other: Every End of Sprint* |
| *Raja Gorentla*  *r.gorentla@fontys.nl* | *R.G.* | *Role: Tutor*  *Function: Group project teacher* | *Mondays from 9-12*  *Tuesdays from 13-16* |
| *Tymofii Andreiev* | *T.A.* | *Role: Developer*  *Function: Student* | *Every Business Day from 9-16* |
| *Nuno Dias* | *N.D.* | *Role: Developer*  *Function: Student* | *Every Business Day from 9-16* |
| *Szymon Gancarz* | *S.G.* | *Role: Developer*  *Function: Student* | *Every Business Day from 9-16* |
| *Duy Nguyen* | *D.N* | *Role: Developer*  *Function: Student* | *Every Business Day from 9-16* |
| *Dick van de Meulenhof* | *D.M.* | *Role: Developer*  *Function: Student* | *Every Business Day from 9-16* |
| *Addi Beenen* | *A.B.* | *Role: Developer*  *Function: Student* | *Every Business Day from 9-16* |

## Communication

**R. Eijgelshoven**

After each sprint, a feedback and question meeting will be arranged, to be held at the university.

**R. Gorentla**

After each week, a meeting with the tutor will be scheduled, taking place at the university.

**Group Communication**

Communication will be conducted online through Discord and in-person at the university, depending on the day.

# Activities and time plan

## Phases of the project

**Phase 1 –Setup & Planning:**

The first phase of any project is Setup and Planning. It involves creating the first draft of a Project Plan as well as setting up the basic infrastructure to start working as a group, notably a version control system like Git and a task scheduler like Jira. Finally, when working in AGILE methodology creating a backlog of tasks and assigning some to the first sprint.

**Phase 2 –Documentation:**

In this phase, the rest of the documentation will be completed. This includes a UML, coding guidelines, code comments, user stories and an architecture document. As well as revising old documents to keep them up-to-date and modify them according to any feedback from the client.

**Phase 3 –Research:**

In this phase, research is done to get more information about the topics and technologies necessary to complete the project. Research questions are made, and these get researched and documented so that the project can be finished successfully.

**Phase 4 –Backend:**

The next phase of the project is going to be backend – arguably, the most important part of the project. The plan for this phase is to start with event handlers to get the events. Then, database for saving all the data present in events and the system to transform all the data from database to logic are going to be created. Finally, everything is going to be covered with unit testing for preventing any issues happening in the future

**Phase 5 –Frontend:**

After the plan and the backend were finished, it is time to start showing up the events on the screen. For this, the plan is pretty simple: the first (optional) step is to create a prototype so there is a visual representation of what should be on the website. Then, the structure of the website will be created with HTML and CSS. Finally the JavaScript for talking with the Backend will be added.

**Phase 6 –Testing:**

This phase involves testing the project for bugs, usability and user friendliness. Because the plan is to use CI/CD there will be continuous testing for bugs and updates that break the application. As for usability and user friendliness this will be tested manually after significant updates to these aspects are made. It is important to test these early so that fixes can be pushed before any real damage is made.

**Phase 7 –Deployment:**

This final development phase is all about deploying the WebApp to the real world and readying it for real users. A docker file for easy development is essential as are compatibility and robustness. This will also involve a sprint to find and fix as many bugs as possible.

**Phase 8 –Delivery:**

The Delivery phase is meant to round everything up and handing the final product over to the client. This should be done with a presentation. All files will be laid out, bundled and handed over in a intuitive way.

## Time plan and milestones

**Sprint A:**

* **Activities:**

1. Form project and define roles
2. Initial meeting with the client to understand requirements.
3. Create a project name and logo.
4. Create the project plan
5. Create Jira/Trello to organize work
6. Start on UML
7. Start to work on Backend

**Milestone:** Sprint delivery

**Sprint B:**

* **Activities:**

1. Formulate research questions
2. Work on architecture document
3. Proceed on UML
4. Continue to work on Backend
5. Start to work on Frontend

**Milestone:** Sprint delivery

**Sprint C:**

* **Activities:**

1. Continue to work on Backend and Frontend
2. Start on CI/CD
3. Create cultural awareness report

**Milestone:** Sprint delivery

**Sprint D:**

* **Activities:**

1. Continue to work on Backend and Frontend
2. Create more test with CI/CD

**Milestone:** Sprint delivery

**Sprint E:**

* **Activities:**

1. Make presentation
2. Put a webapp into docker
3. Test everything
4. Record a video

**Milestone:** Sprint delivery

|  |  |  |
| --- | --- | --- |
| **Phasing** | **Start date** | **Finish date** |
| 1. Sprint A | 01-09-2024 | 11-10-2024 |
| 1. Sprint B | 11-10-2024 | 08-11-2024 |
| 1. Sprint C | 08-11-2024 | 29-11-2024 |
| 1. Sprint D | 29-11-2024 | 20-12-2024 |
| 1. Sprint E | 20-12-2024 | 17-01-2025 |

# Testing strategy and configuration management

## Testing strategy

The testing strategy for **Key watch** will focus on ensuring that the system is both functional and reliable. Testing will be performed at multiple levels to ensure the quality and correctness of the system. The strategy includes:

1. **Unit Testing**:

* **Objective**: Ensure that individual components, such as functions and methods, work correctly in isolation.

1. **Integration Testing:**

* **Objective**: Ensure that multiple components work well together. Tests bigger chunks of code and multiple components running together.

1. **Acceptance Testing**:

* **Objective**: Ensure that the application meets the requirements and expectations of stakeholders. The acceptance criteria will be based on functional specifications.

Unit and Integration tests will cover all the logic and domain classes in the back end as well as some parts of the persistence and controller layer in some bigger integration tests. These tests will be written in JUnit with some Mocking done by Mockito, they will all also be automated in a CI Pipeline every time a change is committed to Gitlab.

Acceptance test will be conducted in select occasions by the team and possibly the stakeholders themselves.

## Test environment and required resources

The testing environment will be a docker container running on the computers of the development team, to save on cloud computing costs. The docker image is a simple Gradle 8.10.1, JDK 17 running on Linux officially provided by Gradle. The CI/CD pipeline will mostly consist of Continuous Integration until the end of the Project when there is a functioning app that is worth being deployed Continuously. When the need for CD arises it will consist of deploying a docker image automatically to the cloud.

## Configuration management

**Key watch** will utilize the Git versioning control system. The approach to configuration management includes:

* **Branching Strategy**:
  + A **main branch** will be used for stable releases.
  + A **development branch** for ongoing work and feature additions.
  + Feature-specific branches for new development, which will be merged into the development branch after passing automated tests.
* **Version Control**:
  + A diagram of a process

    Description automatically generatedEach feature release will be tagged in Git, with detailed release notes outlining new features, fixes, and known issues.

# Risk and mitigation

|  |  |  |
| --- | --- | --- |
| **Risk** | **Prevention activities** | **Mitigation activities** |
| 1. Loss of contact with Mentor | Frequent updates so time between loss of contact is minimal.  Discuss short but also long-term plans with Mentor | Updates over email and with an alternative guide (teacher) |
| 1. Sickness | Even distribution of work between members | Online work and redistribution of tasks between other members |
| 1. Data Loss | Version Management with Git and cloud backup with Gitlab on Fontys Servers | Checking for latest version on members computers and re-merging |
| 4 Expectations different from result | Frequent updates and sprint-end meetings to align/re-align goals and expectations with client | Reorganizing priorities and planning |
| 5 Scope Creep | Frequent internal meetings (retrospective on sprints) to assess the feasibility of the project, timelines, and goals.  MOSCOW & Phasing that prioritizes critical parts of project first | Reducing the scope on less critical parts of the project |
| 6 Time Loss | Phasing that prioritizes critical parts of project first  Thorough planning with some breathing room for bug fixing, testing and other problems that can be relied on if needed | Reorganizing priorities and planning |