

Project Plan



BAS
WORLD

trusted trade platform

Key Watch

BAS World

Date	09/09/2024
Version	0.1
State	Draft
Author	Nameless

Version history

Version	Date	Author(s)	Changes	State
0.1	09/09/2024	Nameless	First draft	Complete

Distribution

Version	Date	Receivers

Contents

1. Project assignment.....	4
1.1 Context	4
1.2 Goal of the project	4
1.3 Scope and preconditions	4
1.4 Strategy	4
1.5 Research questions and methodology	5
1.6 End products.....	5
2. Project organisation.....	6
2.1 Stakeholders and team members.....	6
2.2 Communication	6
3. Activities and time plan.....	7
3.1 Phases of the project.....	7
3.2 Time plan and milestones.....	7
4. Testing strategy and configuration management	8
4.1 Testing strategy	8
4.2 Test environment and required resources.....	8
4.3 Configuration management	8
5. Risk and mitigation	9

1. Project assignment

1.1 Context

In Semester 3 at Fontys ICT, students are expected to deliver a real-world application for an actual company. 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.

1.2 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.

1.3 Scope and preconditions

Inside scope:	Outside scope:
1 Create API or service that receives events	1 Deployment with AWS
2 A back-end system that stores and organizes data	2 Authorization and authentication
3 Front-End dashboard to displays event data in real time	3 Reactive display regarding different devices (primarily desktop)
4 Event Data Simulation	4 Modification of data

1.4 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).

1.5 Research questions and methodology

<<

Describe the research questions that are most relevant to your project. For each research question, describe the approach and/or methodology. Use the Dot Framework to specify strategies and methods - see <http://www.ictresearchmethods.nl> for details.

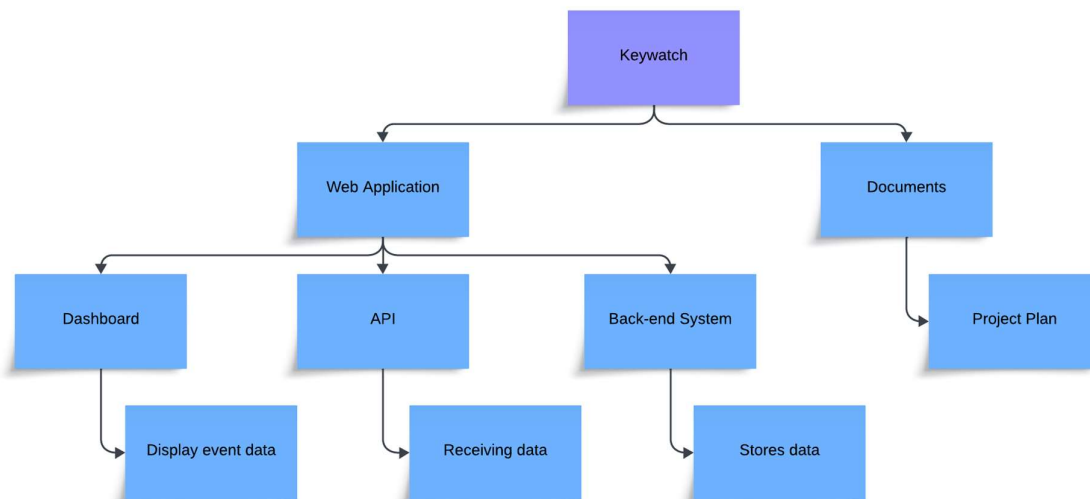
Note that research is not only part of the initial phases (like analysis) of the project, but runs throughout the whole project. E.g., in the realization phases, you will probably do research in the Workshop and Lab context.

Realize that during the project your research questions may change, and that new ones will come up. That normal for any project, and is not a problem as long as you involve the right stakeholders, and keep your deliverables updated.

>>

(To be Added)

1.6 End products



2. Project organisation

2.1 Stakeholders and team members

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

2.2 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.

3. Activities and time plan

3.1 Phases of the project

Webapp
Docker

3.2 Time plan and milestones

- Sprint A:
- Activities:
 - Form project and define roles
 - Initial meeting with the client to understand requirements.
 - Create a project name and logo.
 - Create the project plan
 - Create Jira/Trello to organize work
 - Start on UML
- Milestone:
 - Sprint delivery
- Sprint E:
- Activities:
 - Make presentation
 - Test everything
 - Record a video
- Milestone:
 - Sprint delivery

Phasing	Start date	Finish date
1 Sprint A	01-09-2024	11-10-2024
2 Sprint B	11-10-2024	08-11-2024
3 Sprint C	08-11-2024	29-11-2024
4 Sprint D	29-11-2024	20-12-2024
5 Sprint E	20-12-2024	17-01-2025

4. Testing strategy and configuration management

4.1 Testing strategy

<<Which testing strategy do you envision? E.g., on which levels will testing take place? Consider that you could choose unit, component, integration, system, or acceptance testing.

Justify your strategy, and also set goals where relevant. E.g., percentage code coverage for the relevant unit tests. For each of the planned tests, indicate what will be automated and what not.

Also think of quality testing setups like, e.g., Sonarqube.
>>

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

4.2 Test environment and required resources

<< Describe the test environment. E.g., do you envision a DTAP (Development, Testing, Acceptance, Production) environment. Can you make use of a CI/CD environment or will you develop your own?

It often helps to use a picture to visualize the test environment.

If you already know, describe which resources are required for realization and testing. Think of hardware, cloud environments and specific tooling required for development and testing.
>>

4.3 Configuration management

<< Describe the project approach with respect to version management (e.g. your GIT repository). This might include things like tooling, branching strategy, promotion-, release- and baseline strategy.

Also, when relevant, think of a mechanism to deal with change requests and problem reports.>>

Key watch will utilize a Git-based version 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:**
 - Each feature release will be tagged in Git, with detailed release notes outlining new features, fixes, and known issues.

5. 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)
2 Sickness	Even distribution of work between members	Online work and redistribution of tasks between other members
3 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