**Project Plan**

***Knowledge Import Tool***

*Customer: Ali Odaci, IQADOT*

|  |
| --- |
| **Date : 08/11/2023** |
| **Version : 0.2** |
| **State : Work in progress** |
| **Author : Group 3** |

#### Version history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
| 0.1 | 09/10/2023 | CodeCraft | Initial version | Complete |
| 0.2 | 08/11/2023 | CodeCraft | Research question, project scope, start of development | Complete |
| 0.3 | 01/12/2023 | CodeCraft | End product scope updated, research | Work-in-Progress |

**Distribution**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Receivers** |
| 0.1 | 11/10 | Ali Odaci, IQADOT |
| 0.2 | 08/11 | Ali Odaci, IQADOT |
| 0.3 | 01/12 | Ali Odaci, IQADOT |

Contents

[1. Project assignment 3](#_3dy6vkm)

[1.1 Context 3](#_1t3h5sf)

[1.2 Goal of the project 3](#_4d34og8)

[1.3 Scope and preconditions 3](#_2s8eyo1)

[1.4 Strategy 3](#_17dp8vu)

[1.5 Research questions and methodology 3](#_3rdcrjn)

[1.6 End products 4](#_26in1rg)

[2. Project organisation 6](#_lnxbz9)

[2.1 Stakeholders and team members 6](#_35nkun2)

[2.2 Communication 6](#_1ksv4uv)

[3. Activities and time plan 7](#_44sinio)

[3.1 Phases of the project 7](#_2jxsxqh)

[3.2 Time plan and milestones 7](#_z337ya)

[4. Testing strategy and configuration management 8](#_1y810tw)

[4.1 Testing strategy 8](#_2xcytpi)

[4.2 Test environment and required resources 8](#_3whwml4)

[4.3 Configuration management 8](#_2bn6wsx)

[5. Finances and risk 9](#_qsh70q)

[5.1 Project budget 9](#_3as4poj)

[5.2 Risk and mitigation 9](#_49x2ik5)

# Project assignment

## Context

IQADOT is a R&D Investment company, which helps to find, share and store knowledge in a sustainable way. It is building a community of people to share the knowledge that lies with them.

## Goal of the project

The goal of the project is to create an application that is able to source information from authentic and trusted sources all over the internet and store it in the internal database. This will provide a more diverse set of knowledge, points of view, all in the name of finding the best possible answer to someone’s query.

## Scope and preconditions

|  |  |
| --- | --- |
| **Inside scope:** | **Outside scope:** |
| 1. Handling API requests | 1. Adding filtering for articles |
| 1. Transforming data for it to be suitable for the internal API | 1. Adding search functionality for imported data |

## Strategy

The chosen approach for this project is Agile, specifically Scrum. The rationale behind this choice is to adapt and respond effectively to the evolving nature of the project. Agile methodologies prioritize flexibility, regular feedback, and iterative development, all of which align with the dynamic goal of creating a knowledge aggregation platform for IQADOT. This approach allows for continuous improvement, user engagement, and the ability to incorporate changing requirements and knowledge sources as the project progresses.

## Research questions and methodology

1. What are the most effective methods for importing and converting diverse knowledge types (articles, Q&A data) into basic HTML markup using the Knowledge Import Tool (KIT)?
2. How does the deployment of KIT as a standalone web-based tool influence its accessibility and usability for different user groups, especially with regard to deployment challenges and strategies?
3. What are the challenges and opportunities associated with incorporating real-time history tracking and management features for imported items within KIT, and how does this impact user engagement and system performance?
4. How do students perceive the effectiveness of KIT in facilitating their progression to a Junior Developer level, and what improvements can be suggested based on their feedback and experiences during the project?
5. How can KIT be optimized to efficiently handle a large volume of data imports while maintaining responsiveness and performance, considering scalability and optimization techniques for data processing?
6. How can we centralize API data imports and process efficiently data that is in such diverse formats?

## End products

**1. Website Application**

* The main web application that users interact with.

**2. API Integration**

* Integration with external data sources to gather information.
* Rijksmuseum importing and formatting of data.
* Custom API imports with saving a history of received information and requests made.

**3. IQADOT API**

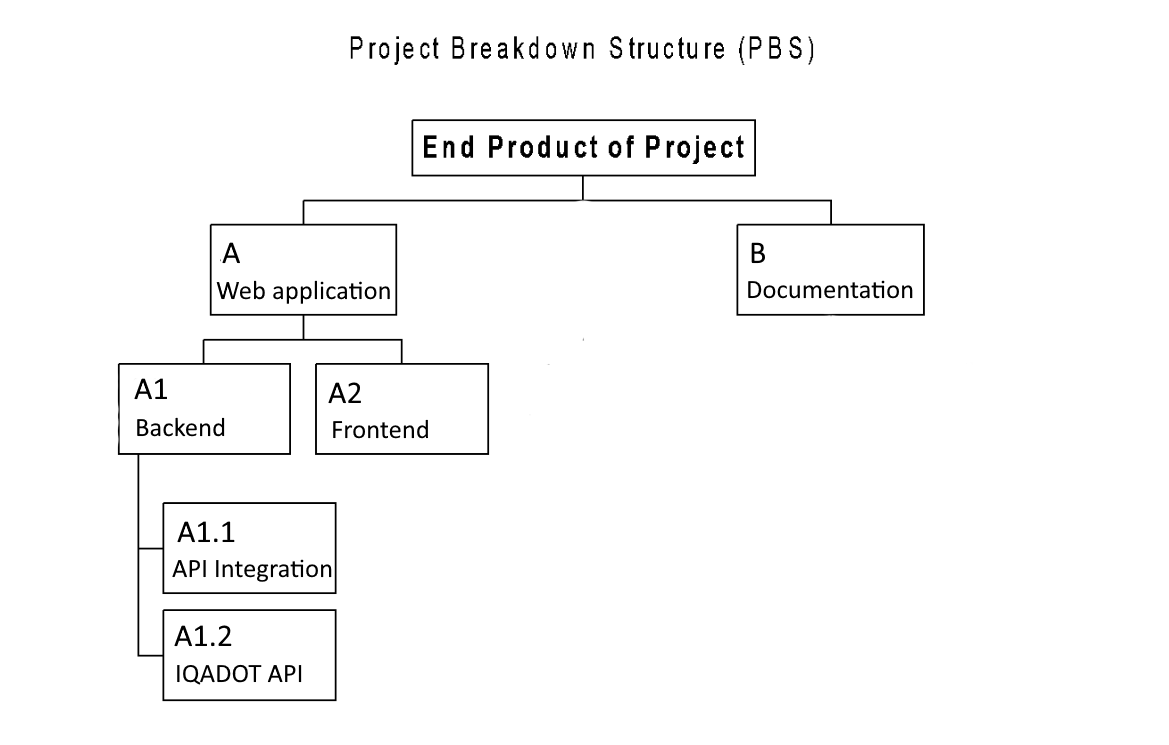
* Integration with IQADOT's API for data submission and retrieval.
* Submission of manually imported articles (example: pdf’s)

**4. Documentation**

* Documentation related to our plan and workflow – Project Plan, Jira board with user stories.
* Comprehensive test plan

**5. Testing Reports**

* Reports on testing and quality assurance.



# Project organisation

## Stakeholders and team members

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Abbreviation** | **Role and functions** | **Availability** |
| *Ali Odaci* | *Ali* | *Product Owner* | *Contact is made via email.* |
| *Atanasov Dimitar* | *D.A.* | *SCRUM Master, developer* | *Monday-Friday* |
| *Balan Mihai* | *B.M.* | *Project Manager, developer* | *Monday-Friday* |
| *Bakalov Ivan* | *I.B.* | *Backend Developer* | *Monday-Friday* |
| *Angelov Pavel* | *P.A.* | *Frontend Developer* | *Monday-Friday* |
| *Sorbala Oliver* | *S.0.* | *Developer, Tester* | *Monday-Friday* |
| *Stefan-Nikola Stanev* | *S.S.* | *Backend Developer* | *Monday-Friday* |

## 

## Communication

Communication between team members will happen in person, or online via texting and calling apps like Discord.

Communication with the client will happen via emails, where we can also schedule meetings during the times all/most members of the team, and the client respectively, are available.

# Activities and time plan

## Phases of the project

1. Initialization (Week 1-4):

- Problem Analysis: Define the scope and objectives of the KIT project in alignment with Fontys IQADOT PiE-partnership goals.

- Documentation Setup: Establish project documentation and version control systems.

- Requirements Gathering: Identify and document specific requirements, such as the types of sources to be supported and the user interface design.

- System Design: Create the architectural and UI/UX design of the web-based tool.

- Security Planning: Outline security measures, including basic UID/PW login.

2. Implementation (Week 5-16):

- Development (Sprints 1-4): Implement the web-based tool following the designed architecture.

- Testing and Quality Assurance (Sprints 5): Conduct testing, including unit, integration, security, and user acceptance testing.

- Documentation and Handover : Finalize installation and user guides, along with design documentation. Provide IQADOT with the deployable web-based tool and associated documentation.

- Evaluation, Reflection, and Wrap

## Time plan and milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **Phasing** | **Effort** | **Start date** | **Finish date** |
| 1. Setting up an environment | 3 weeks | 24.09.2023 | 13.10.2023 |
| 1. Further Development | 6 weeks | 23.10.2023 | 20.12.2023 |
|  |  |  |  |

# Testing strategy and configuration management

## Testing strategy

* *Testing will be done automatically, via unit tests, during the continuous delivery process.*
* *Testing will be also done by developers, which will try the functionalities of the app*

## Test environment and required resources

*The environment will be following the DTAP model, making use of our own CI/CD tools for automatic integration and testing.*

*1.Development (D)*

* *Development machines*
* *Version control (e.g., Git)*
* *IDE or code editor*
* *Local development servers/containers*
* *CI/CD tools (e.g., Jenkins)*
* *Test data generation tools (if needed)*

*2. Testing (T)*

* *Separate testing servers/containers*
* *Test database*
* *Test data*
* *Test automation tools*
* *CI/CD pipeline*
* *Test management tools*
* *Monitoring and debugging tools*

*3. Acceptance (A)*

* *Environment mirroring production*
* *Realistic test data*
* *UAT testers*
* *Test automation tools*
* *Collaboration and feedback mechanisms*

*4. Production (P)*

* *Production servers and infrastructure*
* *Load balancing*
* *Production database with backups*
* *Monitoring and logging*
* *Security measures*
* *Scalability and auto-scaling*
* *Disaster recovery and backups*

## Configuration management

*We will be using GIT for version control, using the GitHub Flow branching model.*

*We will be using Git tags for marking stable versions, on main and master branches.*

*We will be making use of pull and merge requests for code review and collaboration, as well as maintaining a clean documentation.*

# Risks

## Risk and mitigation

|  |  |  |
| --- | --- | --- |
| **Risk** | **Prevention activities** | **Mitigation activities** |
| 1. Insufficient Data Source Availability | Conduct thorough research on data source reliability and availability. | Identify and validate alternative data sources in case of primary source unavailability. |
| 1. Security Breach | Implement stringent security measures and best practices in software development. | Develop an incident response plan to contain breaches and take corrective actions. |
| 1. Integration Challenges | Conduct thorough integration planning and feasibility assessments. | Allocate a dedicated integration team and sufficient time for troubleshooting and refinements. |
| 4 Unforeseen Technical Issues | Conduct regular code reviews and employ sound development practices. | Maintain a flexible project timeline to accommodate unexpected technical challenges and provide robust support to resolve issues promptly. |