**Project Plan**

***Kwetter***

|  |
| --- |
| **Date : 28/02/2024** |
| **Version : 1.0** |
| **State : Developing** |
| **Author : Saeed Ba Wazir** |

Version history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
| **1.0** | 28/02/2024 | Saeed Ba Wazir | Adding the sections:  Project assignment, Activities and time plan, testing strategy and configuration management, Technology stack, learning outcomes and Finances and risk. | Developing |
|  |  |  |  |  |
|  |  |  |  |  |

**Distribution**

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

Contents

[1. Project assignment 4](#_Toc42673512)

[1.1 Context 4](#_Toc42673513)

[1.2 Goal of the project 4](#_Toc42673514)

[1.3 Scope and preconditions 4](#_Toc42673515)

[1.4 Strategy 4](#_Toc42673516)

[1.5 Research questions 4](#_Toc42673517)

[1.6 End products 4](#_Toc42673518)

[2. Project Organisation 6](#_Toc42673519)

[2.1 Stakeholders and team members 6](#_Toc42673520)

[2.2 Communication 6](#_Toc42673521)

[3. Activities and time plan 7](#_Toc42673522)

[3.1 Phases of the project 7](#_Toc42673523)

[3.2 Time plan and milestones 7](#_Toc42673524)

[4. Testing strategy and configuration management 8](#_Toc42673525)

[4.1 Testing strategy 8](#_Toc42673526)

[4.2 Test environment and required resources 8](#_Toc42673527)

[4.3 Configuration management 8](#_Toc42673528)

[5. Finances and Risk 9](#_Toc42673529)

[5.1 Project budget 9](#_Toc42673530)

[5.2 Risk and mitigation 9](#_Toc42673531)

# Project assignment

## Context

The project at hand involves the development of a social media application that closely resembles the functionality of Kwetter, a popular microblogging platform. This platform is designed to facilitate the exchange of short, text-based messages, with each post limited to a maximum of 140 characters. Users will be able to share their thoughts, updates, and engage with others in real-time, fostering a sense of community and instant communication.

## Goal of the project

The goal of this project is to meet the contemporary demands of online communication by developing a user-friendly social media application inspired by Kwetter. This initiative aims to create a platform that encourages efficient and real-time communication through concise messaging, fostering meaningful connections. The advantages include a competitive edge in the social media landscape, a user-centric design, and contributions to the company's growth and market positioning. Leveraging ICT capabilities, the project will deliver a scalable architecture, real-time communication features, and robust user profile management, enhancing the overall user experience and securing a prominent position in the digital space.

## Scope and preconditions

|  |  |
| --- | --- |
| **Inside scope:** | **Outside scope:** |
| 1. Managing the appliction assets | 1. Deploying the appliction in Production |
| 1. Creating the infrastructure for the application | 1. Extended Feature Development |

## Strategy

Our project will adopt an Agile approach, specifically utilizing the Scrum framework. This choice is driven by the need for adaptability in the dynamic realm of social media development. The three-week sprint cycle strikes a balance between consistent progress and the flexibility to respond swiftly to changing requirements and user feedback. This Agile strategy fosters collaboration, transparency, and a continuous improvement mindset, aligning seamlessly with the evolving nature of social media platforms.

## Research questions and methodology

Throughout the entirety of the Kwetter project, research is integral to refining and optimizing our strategies. We adopt the [DOT Framework](https://ictresearchmethods.nl/dot-framework/) to articulate our research questions and methodologies, ensuring a systematic approach. We will have main question that will be answered using the sub questions.

**Main question**

How can the enterprise ensure that its operations can grow efficiently, maintain security, and adhere to legal compliance regarding user data and communication within the application?

**Sub questions**

* How can the enterprise scale to ensure high value and optimal user experience for the databases and the services?

To address the first inquiry, we will employ a [literature study](https://www.ictresearchmethods.nl/library/literature-study) to accumulate insights into existing solutions. Subsequently, we will utilize a framework of best, good, and bad practices to discern using [Best good and bad practices](https://www.ictresearchmethods.nl/library/best-good-and-bad-practices/) methodology and select the most suitable solutions for integration into the project.

* How can the security be guaranteed of the communication within the application?

In response to the second question, our approach involves conducting a thorough [literature study](https://www.ictresearchmethods.nl/library/literature-study) to gather insights into existing solutions. Following this, we will apply a methodology based on [Best good and bad practices](https://www.ictresearchmethods.nl/library/best-good-and-bad-practices/) to carefully discern and select the most fitting solutions for integration into the project.

* How can we ensure that the handling of user data adheres to EU laws and regulations?

For the third question, we will employ the [literature study](https://www.ictresearchmethods.nl/library/literature-study) methodology to gather insights from academic and professional sources. This approach ensures a comprehensive understanding of successful and less effective approaches, guiding our decision-making based on existing knowledge and the latest advancements in the field.

## End products

The Product Breakdown Structure (PBS) for our project delineates the key end products that will be realized throughout the software development lifecycle. At the top level, the "Kwetter" component represents the overarching deliverable. Under this component, two primary end products emerge:

**Applications**

* Website: The user-facing interface where individuals interact with the social media application.
* Backend: The underlying server-side infrastructure that manages data, user accounts, and facilitates communication.

**Documentation**

* Project Plan: Outlines the project's scope, objectives, schedule, and resource allocation.
* Research Document: Summarizes the findings from research activities, guiding decision-making during development.
* Technical Design Document: Details the architectural and technical aspects of the application, serving as a blueprint for developers.
* DevOps Document: Describes the deployment and operational processes, ensuring smooth application deployment and maintenance.

The PBS acknowledges that the structure may evolve during the project, adapting to changing requirements and insights gained throughout development. Each listed end product plays a crucial role in effective handover, maintenance, and potential follow-up projects.

A diagram of a project

Description automatically generated

# Activities and time plan

## Phases of the project

The plan for Kwetter, follows a Scrum methodology with 3-week sprints. In Sprint 0-1, we focus on analysis, design inception and planning, defining scope and setting up the project. Sprints 2-4 are dedicated to development, implementing features and ensuring continuous integration. Sprint 5 is reserved for the portfolio development.

## Time plan and milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **Phasing** | **Effort** | **Start date** | **Finish date** |
| 1. Sprint 0 | * Project pitch | 12-02-2024 | 03-03-2024 |
| 1. Sprint 1 | * Project plan * Technical design document * The System scholten | 04-03-2024 | 24-03-2024 |
| 1. Sprint 2 | * Start page * Login/ logout * Search for the tweets content * Creating a tweet with 140 max letters * View profile page * Following a user * Viewing timeline of the users that is been followed | 25-03-2024 | 14-04-2024 |
| 1. Sprint 3 | * Start page * Mentions that allow users to mention another users * Trends which shows the tweets that contains a trend hashtag * Profile page * Adding hearts for a tweet | 15-04-2024 | 12-05-2024 |
| 1. Sprint 4 | * Moderation page * View all users * Remove a user * Change user role * Administration page * View all users * Remove a user * Add a user * Change user role | 13-05-2024 | 02-06-2024 |
| 1. Sprint 5 | * The Final version of the portfolio | 03-06-2024 | 23-06-2024 |

# Testing strategy and configuration management

## Testing strategy

## The testing approach encompasses automated unit tests targeting a 60% code coverage to ensure robust individual component functionality. Manual system testing will validate end-to-end scenarios, covering real-world usage. Sonarqube will automate code quality assessments, focusing on maintainability, reliability, and security. This streamlined strategy aims for comprehensive testing while emphasizing code quality throughout the development process.

## Test environment and required resources

The project will establish a CI/CD (Continuous Integration/Continuous Deployment) environment using Azure services. This automated pipeline will enable seamless transitions across Development, Testing, and Production stages. The resources required include Azure services for hosting, testing environments, and CI/CD execution, along with local hardware for development. This streamlined approach ensures efficient testing and deployment processes throughout the software development lifecycle.

## Configuration management

The project utilizes GIT for version management with a three-branch strategy: Development, QA, and Production. Features are developed in the Development branch, and new features are branched from it. The QA branch is dedicated to testing, ensuring stability before merging into Production. Change requests are incorporated into the Development branch, and problem reports are addressed systematically, with fixes tested in QA before Production. This streamlined approach ensures a structured workflow and a reliable software development process.

# Technology stack

The project adopts a modern and efficient technology stack with React for the frontend and .NET for the backend.

**Frontend - React**

Utilizing React, a JavaScript library renowned for its component-based structure, we aim to deliver a dynamic and responsive user interface. React's declarative syntax and efficient rendering make it an ideal choice for building interactive and visually appealing frontend components.

**Backend - .NET**

The backend is powered by the robust .NET framework, offering scalability, security, and a comprehensive set of tools. .NET ensures efficient server-side logic and seamless integration with various databases, providing a solid foundation for the dynamic nature of the social media application.

This carefully selected technology stack combines the strengths of React and .NET to deliver a cohesive, high-performance, and scalable solution for the social media application.

# Learning outcomes

**Learning Outcome 1 - Professional Standard**

Demonstrate a high professional standard by taking responsibility for solving ICT issues, conducting applied research using relevant methodologies, and providing well-substantiated advice in complex and uncertain contexts. Actively seek feedback for continuous improvement.

Acting professionally involves seeking continuous feedback from peers, mentors, and stakeholders. Regular self-assessment and openness to constructive criticism will be integral to maintaining a high standard.

**Learning Outcome 2 - Personal Leadership**

Exhibit personal leadership by independently formulating goals and actions that contribute to long-term development as an ICT professional. Maintain a professional attitude, plan goals, and adjust actions as needed to showcase leadership capabilities.

Planning goals and achieving them will be central to demonstrating personal leadership. Regular self-assessment, seeking feedback, and flexibility in adjusting goals as necessary will be key components of this achievement approach.

**Learning Outcome 3 - Scalable Architectures**

Master the development of enterprise software architecture with a focus on explicitly stated software quality requirements. Use Kubernetes to ensure scalability, address legal and ethical considerations, design for future adaptation, and assess the extent of quality requirements met in the software implementation.

The achievement approach involves a comprehensive understanding and implementation of Kubernetes for scalability. Continuous research on legal and ethical considerations, coupled with regular assessments of software quality, will be integral to mastering scalable architectures.

**Learning Outcome 4 - Development and Operations (DevOps)**

Implement a robust CI/CD pipeline and set up environments, tools, and processes supporting continuous software development. Deploy integrated software systems and actively monitor application parts for quality attributes, ensuring a seamless and efficient DevOps process.

Creating a CI/CD pipeline is at the core of achieving this outcome. The approach involves setting up efficient environments and tools for continuous development, deploying integrated software, and actively monitoring for quality attributes to ensure an effective DevOps process.

**Learning Outcome 5 - Cloud Native**

Adopt best practices in cloud native development by deploying application parts on a cloud platform. Integrate cloud services, including Serverless computing, cloud storage, and container management, and articulate the added value of these services for enhancing software quality.

The approach involves deploying application parts on a cloud platform and integrating various cloud services. Clearly articulating the added value of these services for software quality will require continuous learning and staying updated with cloud-native best practices.

**Learning Outcome 6 - Security by Design**

Investigate and minimize security risks for the application by researching and applying best practices throughout the entire software development process. Integrate security measures into the design choices, ensuring a secure and resilient software solution.

Researching and applying best practices for security will be the primary approach. Integrating security measures into the entire software development process will involve continuous learning, adaptation, and staying informed about evolving security threats.

**Learning Outcome 7 – Distributed Data**

Apply best practices for handling and storing various data types in software. Align design choices with non-functional requirements, especially legal and ethical considerations, to protect and distribute data without compromising other software qualities. Research and apply optimal practices for effective data management.

The approach involves researching and applying best practices for effective data management. Regularly seeking feedback on data handling processes and ensuring alignment with non-functional requirements will be essential in achieving optimal practices for distributed data.

# Finances and risk

## Project budget

The project has a budget of $100, earmarked for Azure Cloud services through a student account, covering essential expenses for cloud infrastructure. To gain budget approval, a detailed breakdown of anticipated costs for hardware, applications, libraries, and development environments will be provided.

## Risk and mitigation

The table below shows the risks, Prevention, and mitigation.

|  |  |  |
| --- | --- | --- |
| **Risk** | **Prevention activities** | **Mitigation activities** |
| 1. Azure Service Costs Exceed Budget | Regularly monitor Azure service usage and costs. Set up budget alerts and optimize resource allocation. | If costs approach or exceed the budget, assess resource usage, consider scaling down non-essential services, and explore cost-effective alternatives. |
| 1. Unavailability of Mentor | Establish clear communication protocols and schedule regular check-ins with the mentor. Identify a backup contact within the organization. | In case of the mentor's unavailability, reach out to the backup contact for guidance. Leverage online resources and communities for assistance. Document project progress comprehensively for easy transition. |
| 1. Technical Dependencies on External Libraries | Conduct a thorough analysis of chosen libraries before integration. Stay updated on library updates and potential issues. | If a critical external library faces issues or becomes obsolete, have contingency plans in place, such as identifying alternative libraries or developing in-house solutions. Regularly update dependencies to minimize compatibility issues. |