**Project Plan**

***<<FitQuest>>***

*<<FitLife Studios>>*

|  |
| --- |
| **Date : 16.01.2025** |
| **Version : 1.0** |
| **State : Final** |
| **Author : Alexandru Mazilu** |

#### Version history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
| 0.1 | 04.09.2024 | Alexandru Mazilu | Initial Draft | Draft |
| 0.2 | 11.10.2024 | Alexandru Mazilu | Initial Draft + modifications | Draft |
| 0.3 | 08.11.2024 | Alexandru Mazilu | Basic endpoints for user and authentication & authorization and sonarqube | Draft |
| 0.4 | 29.11.2024 | Alexandru Mazilu | Endpoints for user and workout + front-end | Draft |
| 0.5 | 06.12.2024 | Alexandru Mazilu | Endpoints for user,exercises,workout, web-sockets and front-end implementation and unit tests finished | Pre-Release |
| 1.0 | 16.01.2025 | Alexandru Mazilu | Endpoints for user,exercises,workout,trainer-user,booking,statistics, filtering and front-end implementation + integration tests, e2e tests, deployment in pipeline | Release |

**Distribution**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Receivers** |
| 0.1 | 04.09.2024 | Jordan Mitchell (CEO) |
| 0.2 | 11.10.2024 | Jordan Mitchell (CEO) |
| 0.3 | 08.11.2024 | Jordan Mitchell (CEO) |
| 0.4 | 29.11.2024 | Jordan Mitchell (CEO) |
| 0.5 | 06.12.2024 | Jordan Mitchell (CEO) |
| 0.1 | 04.09.2024 | Jordan Mitchell (CEO) |

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 **Error! Bookmark not defined.**](#_Toc42673517)

[1.6 End products 5](#_Toc42673518)

[2. Project Organisation 6](#_Toc42673519)

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

[2.2 Communication **Error! Bookmark not defined.**](#_Toc42673521)

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

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

[3.2 Time plan and milestones 9](#_Toc42673524)

[4. Testing strategy and configuration management 10](#_Toc42673525)

[4.1 Testing strategy 10](#_Toc42673526)

[4.2 Test environment and required resources **Error! Bookmark not defined.**](#_Toc42673527)

[4.3 Configuration management 10](#_Toc42673528)

[5. Finances and Risk 11](#_Toc42673529)

[5.1 Project budget **Error! Bookmark not defined.**](#_Toc42673530)

[5.2 Risk and mitigation 11](#_Toc42673531)

# Project assignment

## Context

FitQuest is a fitness and health tracking application designed to revolutionize the way users engage with their fitness routines. The project aims to cater to a broad audience, including clients and trainers, by offering a platform that integrates progress tracking, paid trainers and personalized workout plans. The application will be developed with a focus on scalability, user experience, and data integrity.

## Goal of the project

FitQuest aims to deliver a seamless and engaging fitness experience by equipping users with tools to track their progress and receive personalized workout plans. It enhances trainer-client interaction, allowing trainers to efficiently manage client progress and offer support. Additionally, the project prioritizes data integrity by providing platform administrators with the necessary tools to manage the system securely and effectively, ensuring a reliable and optimized fitness platform for all users.

## Scope and preconditions

|  |  |
| --- | --- |
| **Inside scope:** | **Outside scope:** |
| 1. Fully functional web application | 1. Support for third-party fitness apps beyond the initial wearable integrations. |
| 1. Implementation of user dashboards for clients, trainers, and admins. | 1. Multi-language support. |
| 1. Real-time functionality using WebSockets. | 1. Web hosting |
| 1. Authentication and authorization systems for different user roles. |  |
| 1. Completely functional database |  |

**Preconditions:**

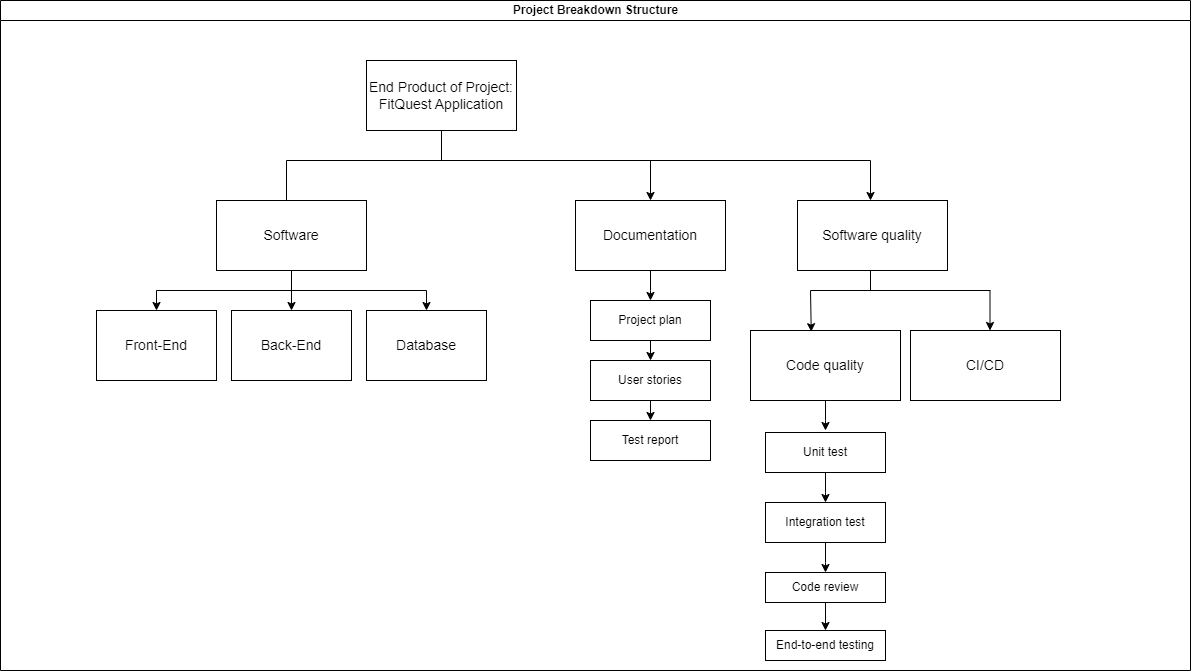
* The tech has been pre-selected, including React.js for frontend development, Spring Boot for backend services and MySQL for database.
* The project must comply with the company’s existing data protection policies.

## Strategy

The project will follow an **Agile Scrum** approach, with development divided into six sprints. This strategy is chosen to accommodate iterative feedback and continuous improvement throughout the project lifecycle. Each sprint will focus on delivering specific features and functionalities, ensuring that the project remains flexible and adaptive to changes.

## End products

* **Authentication and Authorization System:** A secure system managing user roles.
* **User Dashboards:** Interactive dashboards for clients, trainers, and admins.
* **Real-Time Module:** WebSocket-based feature.
* **Deployment Package:** Application ready for cloud deployment.
* **Documentation:** Project plan, test report, user stories.



# Project organisation

## Stakeholders and team members

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Abbreviation** | **Role and functions** | **Availability** |
| *Jordan Mitchell* | *JM* | *CEO* | *meeting / 2 weeks* |
| *Alexandru Mazilu* | *AM* | *Developer* | *Full-time* |

# Activities and time plan

## Phases of the project

**Sprint 1: Project Setup and Initial Planning**

Tasks:

* Research & Planning: Finalize project scope, objectives, and requirements.
* Implementation: Set up Git repository, project documentation, and initial wireframe prototype.
* Testing: Review and refine wireframe and user stories with stakeholders.
* Refinement: Make adjustments based on feedback, finalize initial plans.

Expected Deliverables:

* Completed project documentation.
* Git repository set up.
* Initial wireframe prototype and UML diagrams.
* Refined user stories.

**Sprint 2: Initial Core Feature Development**

Tasks:

* Research: Identify best practices for authentication and UI frameworks.
* Implementation: Develop the user authentication system (login, registration) and begin UI for the client dashboard.
* Testing: Perform initial tests on authentication flows and UI responsiveness.
* Refinement: Debug issues and refine UI/UX based on feedback.

Expected Deliverables:

* Functional user authentication system.
* Initial client dashboard UI.
* Completed and refined tests for implemented features.

**Sprint 3: Advanced Core Feature Development**

Tasks:

* Research: Gather insights on role-based authorization and dashboard UI improvements.
* Implementation: Develop role-based authorization and build trainer and admin dashboards.
* Testing: Test role-specific access controls and dashboard functionalities.
* Refinement: Address bugs and refine UI elements based on testing outcomes.

Expected Deliverables:

* Completed role-based authorization.
* Functional trainer and admin dashboards.
* Refined access controls and UI improvements.

**Sprint 4: Backend API and Database Development**

Tasks:

* Research: Optimize database design and backend architecture.
* Implementation: Design and set up the database schema and develop core backend APIs (CRUD operations).
* Testing: Conduct unit tests on backend services and database operations.
* Refinement: Debug backend functionalities and refine database interactions.

Expected Deliverables:

* Completed database schema design and setup.
* Core backend APIs developed and tested.
* Refined backend architecture.

**Sprint 5: Real-Time Features and WebSockets Implementation**

Tasks:

* Research: Explore WebSocket integration techniques and real-time system design.
* Implementation: Set up WebSockets and develop a real-time notifications system.
* Testing: Test real-time data flow and notifications across all dashboards.
* Refinement: Debug and optimize real-time features for performance and reliability.

Expected Deliverables:

* Real-time WebSocket setup.
* Functional real-time notifications system.
* Refined real-time interactions with dashboards.

**Sprint 6: Final Integration, Testing, and Deployment**

Tasks:

* Research: Final review of best practices for deployment and performance optimization.
* Implementation: Integrate all features, finalize UI, and complete backend integrations.
* Testing: Conduct comprehensive testing including unit tests, integration tests, and user acceptance testing.
* Refinement & Deployment: Final bug fixes, code optimization, and deploy the application.

Expected Deliverables:

* Fully integrated and tested application.
* Finalized deployment to production.
* Completed final documentation and presentation preparation.

**Total Hours:** 300 Hours

## Time plan and milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **Phasing** | **Effort (hours)** | **Start Date** | **Finish Date** |
| 1. Sprint 1 - Establish the project's foundation and initial planning. | 40 | 02.09.2024 | 20.09.2024 |
| 1. Sprint 2 - Develop initial core features and basic UI components. | 50 | 21.09.2024 | 11.10.2025 |
| 1. Sprint 3 - Continue core feature development with focus on UI components and backend integration. | 50 | 12.10.2024 | 08.11.2024 |
| 1. Sprint 4 - Implement real-time features and integrate with existing dashboards. | 50 | 09.11.2024 | 29.11.2024 |
| 1. Sprint 5 - Conduct comprehensive testing, bug fixing, and refinement of all features. | 50 | 30.11.2024 | 20.12.2024 |
| 1. Sprint 6 - Finalize all features, perform deployment, and prepare final documentation and presentation. | 60 | 21.12.2024 | 17.01.2025 |

# Testing strategy and configuration management

## 

## Testing strategy

* **Unit Testing:** Automated tests for each component using JUnit for backend and Jest for frontend.
* **Integration Testing:** Testing the interaction between different components and services.
* **System Testing:** End-to-end testing of the entire application.
* **Acceptance Testing:** Final testing phase, involving stakeholders to validate the end product.
* **Code Coverage Goal:** Aim for 80% coverage across the application.

## Configuration management

* **Version Control:** GitLab will be used for version control, with a branching strategy that includes development and master branches.
* **Release Management:** Regular releases will be tagged and stored in GitLab, with automated deployments to the staging environment.

# Finances and risk

## Risk and mitigation

|  |  |  |
| --- | --- | --- |
| **Risk** | **Prevention activities** | **Mitigation activities** |
| 1. Illness | - | Get enough sleep and eat properly |
| 1. Burnout and productivity decline | Set realistic goals to avoid overworking and ensure steady progress. | If signs of burnout begin to appear, take a short break to recharge. |
| 1. Hardware malfunction | Push to git everyday | - |