**Project Plan**

**Gamehub**

|  |
| --- |
| **Date : 30th of May 2024** |
| **Version : 1.2** |
| **State : In Progress** |
| **Author : Cătălin Popoiu** |

#### Version history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
| 1.0 | 21.02.2024 | Cătălin Popoiu | Filled in most sections, such as Project Assignment and Risk | In Progress |
| 1.1 | 26.02.2024 | Cătălin Popoiu | Filled in missing parts, implemented most feedback received | In Progress |
| 1.2 | 30.05.2024 | Cătălin Popoiu | Revised the whole project plan, changed planning to a accurate version | In Progress |

Contents

[1. Project assignment 3](#_Toc159412981)

[1.1 Context 3](#_Toc159412982)

[1.2 Goal of the project 3](#_Toc159412983)

[1.3 Scope and preconditions 3](#_Toc159412984)

[1.4 Strategy 3](#_Toc159412985)

[1.5 End products 4](#_Toc159412987)

[2. Activities and time plan 5](#_Toc159412988)

[2.1 Phases of the project 5](#_Toc159412989)

[2.2 Time plan and milestones 5](#_Toc159412990)

[3. Testing strategy and configuration management 7](#_Toc159412991)

[4. Finances and risk 8](#_Toc159412992)

[4.1 Risk and mitigation 8](#_Toc159412993)

# Project assignment

## Context

My project will be developed as a comprehensive platform intended to serve the global gaming community, connecting gamers, game developers, and enthusiasts in a dynamic social networking environment.

## Goal of the project

The goal of the project is to create a unified platform where gamers can explore, share, and discuss their gaming experiences, discover new games, and connect with the community. It aims to enhance the gaming experience by providing tools for communication, event organization, and game discovery.

## Scope and preconditions

|  |  |
| --- | --- |
| **Inside scope:** | **Outside scope:** |
| 1. Secure user authentication and customizable profiles | 1. Integration with gaming consoles’ proprietary networks |
| 1. A comprehensive game database for discovery and exploration | 1. Direct game distribution or sales |
| 1. Community forums for discussions and sharing | 1. In-depth analytics for game developers in the initial launch phase |
| 1. Event and tournament organization features | 1. Advanced AI-driven personalized game recommandations in the initial launch phase |
| 1. Developer portal for showcasing games | 1. Mobile application development |
| 1. Real-time messaging and notifications |  |
| 1. Review and ratings system for games |  |

**Preconditions:** The project will leverage React for the frontend, Java Script Boot for the backend, MySQL for the database, and other technologies such as Hibernate, JPA, and WebSocket for real-time communication.

## Strategy

My project will adopt an Agile and Scrum methodology, structured around six sprints to ensure iterative development and continuous feedback integration. This strategy supports flexible project management and allows for adjustments based on user feedback and testing results.

## End products

### 1.Front-end Deliverables

* **User Authentication and Profile Management**: Implementation of secure login systems and customizable user profiles within the React-based front-end application.
* **Community Forums**: A dynamic platform for discussions and sharing amongst users, featuring real-time updates and interactive elements.
* **Event Management System**: User interface for organizing and participating in gaming events and tournaments, with tools for event creation, registration, and management.
* **Real-time Messaging System**: Implementation of real-time user interactions and notifications using WebSockets to facilitate immediate communication and updates.
* **Reviews and Ratings**: User interface allowing users to review and rate games, contributing to the community's understanding and evaluation of gaming content.
* **Friendship System**: Interface for users to send, accept, and manage friend requests, view friends' profiles, and interact with them within the platform.
* **Game Purchase System**: Interface allowing users to purchase games, view their purchase history, and check ownership status.
* **Game Recommendations**: Personalized recommendations based on user preferences and purchase history, helping users discover new games.
* **Administrator Statistics**: Interface for administrators to view and analyse game sales statistics, user engagement metrics, and other relevant data to make informed decisions.

### 2.Back-end Deliverables

* **Game Database**: A comprehensive and searchable database of games, managed by a Java Spring Boot RESTful API, allowing for efficient data retrieval and management.
* **Secure Authentication System**: Back-end logic for handling secure JWT-based authentication and authorization processes, ensuring data protection and privacy.
* **Event and Tournament Organization Features**: Server-side logic to support the creation, management, and participation in gaming events and tournaments.
* **Messaging and Notifications System**: Backend implementation for supporting real-time messaging and notifications, ensuring timely communication across the platform.
* **Friendship System**: Backend services to handle friend requests, friendship management, and interactions between friends.
* **Purchase Management System**: Backend services to manage game purchases, including transaction history, and ownership verification.
* **Recommendation Engine**: Backend logic to analyse user data and provide personalized game recommendations.
* **Administrator Statistics Service**: Backend services to collect, process, and present data for administrators to monitor platform usage and game sales statistics.

### 3.Documentation Deliverables

* **Comprehensive Project Documentation**: Including the project plan, technical specifications, and user documentation, detailing the architecture, design decisions, and functionalities of GameHub.
* **Test Reports and Quality Assurance Documentation**: Documentation of testing strategies, methodologies, results, and quality assurance measures taken throughout the development process, ensuring reliability and performance of the platform.

# Activities and time plan

## Phases of the project

#### ****Sprint 1: Initial Setup and MVP Foundation****

* **Objective:** Establish the project groundwork, including Agile workflow, and begin development on critical MVP features.
* **Activities:** Assemble the Agile team, establish the product backlog, set up the development and staging environments, outline the technological stack and architectural design, design and implement secure login/logout functionality, basic profile management, and a rudimentary interface for the game database.
* **Deliverables:** Project backlog, team composition, development environment setup, basic authentication system, user profile management, initial game database structure.

#### ****Sprint 2: Frontend development and backend integration****

* **Objective:** Begin the development of the frontend application using React and integrate it with the backend services. Complete most CRUD operations for the backend.
* **Activities:** Start developing React components for the frontend application, connect the frontend with backend services, continue developing backend CRUD operations.
* **Deliverables:** Initial frontend application setup, integration of frontend with backend services, completion of most backend CRUD operations.

#### ****Sprint 3: Enhancing frontend and backend features****

* **Objective:** Further develop the frontend application, enhancing user interfaces, and expanding backend functionalities to support more complex features.
* **Activities:** Enhance user interface components in the frontend, expand backend functionalities to support advanced features like community forums and game discovery.
* **Deliverables:** Enhanced user interface components, backend support for advanced features.

#### ****Sprint 4: Community Forums and Engagement Features****

* **Objective:** Focus on developing community forums, engagement features, and starting game discovery mechanisms.
* **Activities:** Develop community forum functionalities, finish implementing engagement features like comments and likes, initiate game discovery features.
* **Deliverables:** Community forums, engagement features (comments, likes), initial game discovery mechanisms.

#### ****Sprint 5: Real-Time Features and Polishing****

* **Objective:** Add real-time features such as chats and notifications, fix remaining bugs, and bring significant polishing to the project.
* **Activities:** Implement real-time chat functionalities, develop notification system for user interactions, fix issues identified in previous sprints, polish the user interface and user experience.
* **Deliverables:** Real-time chat system, notification system, solved bugs and issues and a polished UI/UX.

#### ****Sprint 6: Finalization, Testing and Deployment****

* **Objective:** Finalize development, conduct comprehensive testing, and prepare for deployment.
* **Activities:** Finalize all pending features, conduct extensive testing (unit, integration, user acceptance), optimize performance, prepare deployment strategies.
* **Deliverables:** Completed application, comprehensive test reports, deployment plan.

## Time plan and milestones

The project will be executed over six sprints, utilizing the Agile and Scrum framework to ensure flexibility and responsiveness to feedback. Each sprint will last three weeks, with specific objectives and deliverables.

**Sprint justification:** The choice of three-week sprints is designed to balance the need for rapid progress with enough time for meaningful feature development, testing, and review. This timeframe allows for continuous integration of feedback and iterative improvement without compromising on quality or scope.

**Sprint activities:**

* **Sprint planning:** At the beginning of each sprint, I will identify tasks, choose the current focus of the project and set sprint goals.
* **Sprint demo:** At the end of each sprint, a sprint demo will take place to showcase the completed features and improvements to the stakeholders. This allows the reception of feedback and validation of work.
* **Sprint review:** After the sprint demo, a sprint review meeting will be held. During this session, stakeholders and I will discuss the sprint’s achievements, review the progress increment and gather feedback for future iterations.
* **Sprint retrospective:** After the sprint review, I will hold a sprint retrospective to reflect on the sprint’s performance. This includes identifying areas for improvement, adjusting processes for the next sprint and discussing what went well.

# Testing strategy and configuration management

#### Unit Testing

* **Objective:** Ensure individual components function correctly in isolation.
* **Activities:** Develop and execute unit tests for both backend and frontend components to validate logic, functions, and classes.
* **Tools & Technologies:** Utilize JUnit for backend testing and React Testing Library for frontend components.

#### Integration Testing

* **Objective:** Verify that different modules or services work together as expected.
* **Activities:** Conduct integration tests to ensure that the application's modules interact correctly with each other and with external systems (e.g., databases, third-party APIs).
* **Tools & Technologies:** Use Spring Boot Test for backend integration testing and Cypress or Selenium for end-to-end frontend testing.

#### System Testing

* **Objective:** Confirm the complete and integrated software product meets specified requirements.
* **Activities:** Perform system testing to validate the application's overall performance, security, and functionality.
* **Tools & Technologies:** Employ tools like JMeter for performance testing and OWASP ZAP for security testing.

#### User Acceptance Testing (UAT)

* **Objective:** Ensure the software meets user needs and requirements.
* **Activities:** Conduct UAT with target users to validate the user experience, feature set, and usability of the application.
* **Tools & Technologies:** Feedback tools and manual testing based on user feedback scenarios.

# Risk

## 4.1.Risk and mitigation

|  |  |  |
| --- | --- | --- |
| **Risk** | **Prevention activities** | **Mitigation activities** |
| 1. Technical challenges with new technologies | Assess project complexity and adjust it to personal skills | Allocate time for research and potential training |
| 1. Overestimating development capacity | Assess time management and spend time accordingly on development | Prioritize core features and allow for scope adjustment |
| 1. Bugs and issues during deployment | Testing throughout the whole development phases | Keep an older stable version to rollback to in case it is needed |
| 1. Data security breaches | Implement strong encryption methods | Implement a measure that alerts users in case of security breaches and has them change their password |