Ciri2 Requirements

Table of contents

[Context 1](#_Toc160450471)

[Requirements 1](#_Toc160450472)

[User Account Management: 1](#_Toc160450473)

[PC Recreation and Sharing: 1](#_Toc160450474)

[Performance Submission and Tracking: 1](#_Toc160450475)

[Admin Functions: 1](#_Toc160450476)

[Component Details: 2](#_Toc160450477)

[Stretch Goal: Predicted FPS: 2](#_Toc160450478)

[User stories 2](#_Toc160450479)

# Context

Ciri2 is a web application where pc-gamers can asses and compare their pc’s. In this application a pc-gamer can recreate their pc with different part pickers and manage it. They will also be able to share their pc . A PC gamer can then also submit the fps they get for games, other users can then see how a game performs.

# Requirements

## User Account Management:

* FR1: Users should be able to register for an account with a unique username and password.
* FR2: Users should be able to log in and log out of their accounts securely.
* FR3: Users should be able to reset their passwords if they forget them.

## PC Recreation and Sharing:

* FR4: Users should have the ability to recreate their PC configurations using a part picker tool.
* FR5: Users should be able to save and manage multiple PC configurations associated with their account.
* FR6: Users should have the option to share their PC configurations publicly with others.

## Performance Submission and Tracking:

* FR7: Users should be able to submit the FPS (Frames Per Second) they achieve for a particular game using one of their PC configurations.
* FR8: Users should be able to view their submitted performance data for different games and PC configurations.
* FR9: Users should be able to view aggregated performance data submitted by other users for different games and PC configurations.

## Admin Functions:

* FR10: Admins should have the ability to import games from Steam into the platform's database.
* FR11: Admins should be able to add, edit, and delete PC components from the system's database.

## Component Details:

* FR12: Users should be able to click on individual components within their PC configurations to view detailed information about those components.

## Stretch Goal: Predicted FPS:

* FR13: Users should have the option to see predicted FPS for their PC configurations based on the performance data submitted by other users.

# User stories

US1: As a PC-gamer I want to create an account, so that I can add and view account specific data

US2: As a PC-gamer I want to recreate my PC with part pickers, so that I can share my rig with others

US3: As a PC-gamer I want to view my own PC’s, so that I can get an oversight of my own pc’s

US4: As a PC-gamer I want to share my PC with others, so they can see which parts my pc has

US5: As a PC-gamer I want to submit the FPS I get for a game for one of my rigs, so that I can contribute to the insight of the performance of a game

US6: As a PC-gamer I want to get a clear overview of my submitted performances, so I can get a good overview of how my pc performs.

US7: As a PC-gamer I want a clear oversight of the submitted performances, so that I can see if my PC is able to run a game properly

US8: As an admin I want to import games from steam, so that PC-gamers can submit their fps data for this game

US9: As admin I want to be able to add pc-components, so that PC-gamers can use these components in their PC

US10: As PC-gamer I want to click on the component of a pc, so I can view the details of that specific component

(stretch) US11: As PC-gamer I want to see predicted FPS for my pc based on the performance of other PC’s, so that I can get a quick overview of my estimated performance

# Non-functional requirements

* **Performance:**
  + The platform should respond to user actions within a maximum acceptable time frame of 300ms 99% of the time.
  + The system should be able to handle peak loads during periods of up to 500.000 concurrent users, ensuring consistent performance and responsiveness.
* **Security:**
  + Secure user data storage: The data should be securely stored in a secure environment like auth0.
  + Access controls and permissions: All endpoints must be protected with a permissions system.
  + The platform should adhere to industry-standard security practices and compliance requirements, such as GDPR
* **Scalability:**
  + Scaling: The system should scale so it can handle a minimum of 100.000 concurrent users where the 99th percentile has a response time of 300ms or less.
  + Caching mechanisms and distributed data storage: Implement caching mechanisms and distributed data storage that achieve a minimum cache hit rate of 50%.
* **Maintainability and Extensibility:**
  + **Documentation:** Document at least 90% of the code or API endpoints.
  + **Continuous integration and automated testing:** Implement continuous integration and automated testing that achieves a minimum of 75% code coverage.
* **Compatibility:**
  + The platform should be compatible with the latest versions of major web browsers, such as Chrome, Firefox, Safari, and Edge.
* **Deployment and updates**
  + The platform should have an up time of 99.5%