Project analysis Ciri2

Table of Contents

Description:

Ciri2 is a platform designed for PC gaming enthusiasts. At its core, Ciri2 serves as a hub where users can assess their system's capability to run specific games and benchmark their performance against others in the community. Here's an overview of its key functionalities:

Account Creation: Users can easily create personalized accounts, unlocking access to a host of features tailored to their gaming needs.

PC Rig Configuration: Within the application, users have the flexibility to construct and customize their PC rigs using intuitive part pickers for various components. This empowers them to tailor their systems to meet the demands of their favorite games.

Game Overview: Through Ciri2, users gain access to a comprehensive overview of games, including key details such as system requirements, gameplay mechanics, and community-generated performance data. This empowers gamers to make informed decisions about which titles to explore based on their hardware capabilities.

FPS Submission: Gamers can submit their frames per second (FPS) benchmarks for specific games and graphics presets, allowing them to share and compare their performance metrics with fellow users. This feature fosters a sense of community and healthy competition among players.

Steam Integration: Administrators have the capability to seamlessly import games from Steam, expanding the library of titles available for benchmarking. Users can then contribute their FPS data to enrich the database and provide valuable insights to the community.

Challenges:

Steam Integration and Third-Party Dependencies: Integrating with external platforms like Steam introduces additional complexities and dependencies. Ciri2 must ensure seamless communication and synchronization with Steam's APIs, handle potential API rate limits and downtime gracefully, and mitigate risks associated with changes to Steam's platform or policies that could impact the functionality or accessibility of integrated features.

Data Integrity and Accuracy: Maintaining the integrity and accuracy of benchmarking data is crucial for the credibility and reliability of Ciri2. Ensuring that user-submitted FPS benchmarks are authentic and representative of actual gaming experiences requires implementing validation mechanisms, data normalization processes, and possibly even crowd-sourced validation techniques to detect and mitigate inaccuracies or anomalies.

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 all 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

US11: As a software architect, I want to determine the most suitable architectural styles for my system's requirements so that I can design an effective and scalable solution.

US12: As a developer, I need to understand how various services or components within the system communicate with each other, so that I can ensure seamless interaction and integration between different parts of the system.

US13: As a developer, I need to ensure that the application is secure, so that sensitive data and user information are protected from unauthorized access or malicious attacks.

US14: As a developer, I need to deploy a scalable application to accommodate varying levels of user demand, ensuring that the system can efficiently handle increased traffic without compromising performance or reliability.

US15: As a developer, I want to ensure the reliable deployment of the application without encountering breaking issues, so that users can experience uninterrupted access and functionality.

(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%

Learning outcome time-plan

Sprint	Learning outcomes
1	Setup a good environment for my project: • Professional standards (orienting/beginning) • Personal leadership (orienting/beginning) • Devops (Orienting)
2	Create initial walking skeleton: • Scalable architecture (Beginning) • Devops (Beginning) • Cloud Native (Orienting/Beginning) • Security By Design (Beginning) • Distributred Data (Beginning)
3	Refine walking skeleton + work on research and look at deployment • Professional standards (Beginning/Proficient) • Scalable architecture (Proficient) • Devops (Proficient) • Cloud Native (Proficient) • Security by design (Beginning/Proficient) • Distributed Data (Beginning/Profficient)
4	Work on setting up more services to show distributred data: • Professional standards (Proficient) • Cloud Native (Proficient) • Security by design (Proficient) • Distributed Data (Profficient)
5	Refine project + Fill in missing learning outcomes: • Professional standards (Proficient/Advanced) • Scalable architecture (Proficient/Advanced) • Devops (Proficient/Advanced) • Cloud Native (Proficient/Advanced) • Security by design (Proficient) • Distributed Data (Profficient)