Priscilla Rodriguez

Journal 8-2: Software Design Document

August 25, 2024

The Gaming Room, a versatile application game developer with a history of success, launched a puzzle competition game on the Android platform. To extend the game's availability across various platforms and devices, the client needed to create an online version of the existing Android application.

I composed a comprehensive recommendation section highlighting key areas for the client's consideration, such as the host operating system (OS), memory and storage management solutions on the servers, the API framework, security, and the benefits of utilizing serverless architecture to deploy the product. I made a compelling argument for why a cloud platform was the optimal choice for the client and their goals.

Analyzing the domain model as a UML class diagram was instrumental in understanding the relationships between entities representing the application components. This facilitated the creation of succinct classes with straightforward methods that leveraged the indirect style of other parts to form a seamless model.

The section discussing popular operating systems (e.g., Mac, Windows, and Linux) and flexible platforms could be improved by delving deeper into the differences between the platforms on the client side. The primary client-side concern is the web browser, as it is the main engine that runs the program for the user. An exploration of the differences between operating systems for web applications provided limited insights. Preferences for certain operating systems for specific applications (e.g., Windows for Edge, Mac for Safari, etc.) could be further examined to determine if unsupported apps impose particular or performance constraints.

The client's primary considerations when developing electronic applications are speed and intuitiveness (e.g., ease of use). Users may continue to use the application if they can tolerate its appearance, but they will abandon it if its functionality is unclear or slow. Meeting user demands is crucial, as they are a significant source of revenue for the client.

When programming, start with broad concepts and gradually refine them to achieve precision. Break down high-level goals into iterative processes to arrive at the final design. For instance, a client's business objectives can be divided into functional requirements that highlight the essential features needed to achieve those goals. Functional requirements can further be broken down into technical requirements that specify the exact methods to ensure user satisfaction. These technical requirements can then be translated into design documents, architecture, and code that fulfill those requirements and deliver the final solution.