Ahlaam Abdirahman

8-2 Journal: Portfolio Submission

The Gaming Room wanted to expand Draw It or Lose It to more platforms while making sure it runs smoothly across different devices. The main goal was to design a web based game that would work on both desktop and mobile without major compatibility issues. This meant considering things like hosting, security, scalability, and cost effective solutions.A key part of the design was choosing the right platform for hosting. Since the game needed to run efficiently online, we had to evaluate different operating systems and development tools to find the best fit. The solution also needed to support future updates and new features without requiring a complete rebuild.

One of the areas I think I did well in was breaking down the hosting options and explaining why Linux was the best choice. I compared different platforms, looking at factors like licensing costs, security, and cloud support. This made it easier to justify why Linux would be the most practical solution for long term scalability.I also made sure the document clearly outlined the development tools that would be used. Since the game needed to work on multiple platforms, choosing cross platform languages like JavaScript and Python was important. Also having a detailed design document before jumping into coding made a huge difference. It forced me to think through everything, from how the client and server would communicate to what development tools were needed. Instead of guessing or making changes later, I had a solid plan that made everything more organized and efficient.

A big part of this was understanding the client server model. The game needed to function across different devices, so separating the front end client what users see from the back end server where the game logic happens was necessary. Using a REST API allowed the two parts to

communicate efficiently while keeping them independent. This means if one side needed updates, it wouldn't break the other. If I could go back, I would spend more time detailing the user experience and front end design. Right now, the document focuses a lot on the backend setup, but adding wireframes or UI mockups would have made it easier to visualize how the game would look and function. Since Draw It or Lose It needs to be engaging and easy to use, having a section that outlines navigation, controls, and responsiveness across devices would have been helpful.I also think adding a section about scaling for future updates would have improved the design. If The Gaming Room wanted to expand the game to consoles like Xbox or PlayStation, there would need to be adjustments to the controls and UI. While the backend wouldn't change much, the front end would need modifications to support controllers instead of keyboards or touchscreens. Planning for this earlier would make future expansions easier.

A big part of software design is making sure the final product actually works for the users. Since Draw It or Lose It needed to function smoothly on desktop and mobile, I focused on making sure the design supported cross-platform compatibility. The client-server model made it easier to achieve this since it allowed the game logic to stay consistent while adjusting the front-end for different devices.User experience was also an important factor. A game that runs smoothly but is difficult to navigate wouldn't be enjoyable to play. That's why considering things like responsive design and intuitive controls was essential. Making sure the game loads quickly and doesn't lag was also a priority, which is why the document includes considerations for optimizing performance.The approach I took focused on making the game scalable, flexible, and efficient. By using a client server architecture with a REST API, the game can be easily updated without requiring major changes to the entire system. This also makes it easier to expand to new

Ahlaam Abdirahman

8-2 Journal: Portfolio Submission
platforms in the future.

If I were to design a similar software application in the future, I would use an Agile development approach to improve the design step by step based on user feedback. This would allow for adjustments throughout the development process instead of waiting until the end. I would also consider using cloud based solutions to handle increased traffic if the game grows in popularity. Working on this project helped me understand the importance of planning before coding. Having a solid design document made it easier to structure the project and avoid unnecessary mistakes. It also reinforced the idea that good software design isn't just about writing code it's about making sure everything works together smoothly and meets user needs.