Q: Can we prioritize these requirements? Are there requirements that are more important than others?

A: Only hard require is: visualize AWS. A better visualization than the table view AWS offers. Something that’s intuitive and not cluttered.

This project should have links to the specific components on Amazon. To the tables. 2 depths per component. First depth is the visual part of the object.

Goal is to create great experience, build something cool, and get experience outside of the classroom.

Q: (Question about Jira thing we signed up for)

A: Still working on getting that set up.

Why use Syncfusion? They have community license. Unique for professional license. Use their tools for free. May have to claim one license for each developer. But they’re free.

Q: With desktop application, what is the tech stack?

A: I’m a .net developer. Have been the entire time. .net core is the latest from Microsoft. Compiles to Linux, compiles to Windows. Core runs in Docker.

Syncfusion creates library of components that can go in. File format reading. Lots of people know this for web stuff. WPF stuff.

Flowcharts, organizational charts. (WPF diagrams) nodes, complex shapes. Name of server, status: etc.

Researchers touch AWS stuff. YouTubers touch it. AWS hosting is servers in the cloud that host our .net apps and web apps. We want databases in cloud.

AWS has dumb ways of naming things. It’s not “Virtual Server”. EC2 is a virtual server. Hosting Docker is ECS. EKS is a big one. File Storage is S3. Databases are RDS. AWS hosts a crap ton.

Virtualizing AWS environment. Not whole deal, but 90% of what people use. SQS is queueing system, not super necessary. It’s big in Microservices. IAM is security user securities, permissions. We are *visualizing all this.*

Big issue here is “How are these connected?”

App will visualize all components, all running services, in the red areas, show visually in a way where user can quickly see entire environment, not jumping console to console. See all these in a visual environment, with connections between them

Desktop application. No mobile? Correct. No mobile.

Within a year, whatever is easiest. Desktop is easiest. Web based application is more complex: Web portion, backend portion, and hosting of that app.

Visualization is a diagramming tool.