Karin Virtual Classroom - Risk Management Plan

Risks

1. Security
    a. Cognito AWS service for validating Frontend Users
    b. Frontend ECS Service must connect to Frontend API Securely / Privately
        i. Only allow Login request to unauthenticated users
        ii. Allow authenticated users to call Frontend API
    c. Backend API, Private s3 buckets, Codebuild runtime, dynamo tables, must be privately accessible
    d. The only public entrypoint should be the load balanced ecs service hosting the single page application
    e. DDOS attacks
2. Business
    a. Infrastructure
        i. Aws infrastructure has incurred costs based on usage, as the platform grows, the infrastructure footprint (and cost) of the service will grow as well.
    b. Adoption
        i. If the service is adopted quickly we will encounter infrastructure related risks
        ii. If the service is adopted too slowly, we risk spending more to maintain the service than the service produces.
    c. Security
        i. The application might be vulnerable in an event of remote code execution, codebuild cannot execute for an arbitrary amount of time, needs a maximum exec time
        ii. Even though the application is running smoothly without any bugs, there might be a glitch which could have been unseen, that might enable a student to run a functionality which was originally not allowed for the students.
        iii. DDOS attacks:
    d. AWS failure (low probability)
        i. In a case where AWS has a shutdown, or the support for AWS stops; in a case where AWS is being discontinued by Amazon, then this application would not be capable of running.
        ii. The probability of this happening is very low, since there is a rare possibility that AWS might be discontinued.
    e. Money
        i. AWS services have inherent cost for usage
3. Project Planning
    a. Deadlines
        i. Features may not meet the deadlines. Have to prioritize features. May not deliver all planned features (Discussion Board, Chat server, etc)

    b. Changing requirements
        i. The client(Dr. Kadiyala) may want to add additional features, or change existing features
        ii. The client may not be satisfied with the performance of the application (UI/UX, latency,etc)
    c. Deployment
        i. The client will be responsible to deploy and maintain the product after the team hands the product to the client. The client may have problems maintaining and deploying the product.
4. Programming
    a. Weak testing
        i. Test project components as their changes are committed to ensure continuous integration.
        ii. If a component does not receive rigorous enough testing it should be added to the list of priorities in later iterations.
    b. Bugs left in code after release
        i. Risk can be mitigated by ensuring continuous integration against rigorous test cases.
        ii. Known bugs also should be made a priority for later patches
    c. Releasing with incomplete features
        i. Releasing before the project is feature-complete may be unavoidable due to business constraints.
5. Architecture
    a. Component Selection
        i. If components are not selected in a way that minimizes service impact due to scaling, then the development team cannot account for errors in the system that may be caused by a change in the number of users.
        ii. This problem has been worked around in the Karin service since we have decided to use AWS s3, dynamodb, ecs, lambda, and api gateway which are all scalable services
    b. Interface Design
        i. The component interface must be designed such that functions are event driven and each sub-system is independent of its surrounding systems