Simply the Gathering Challenge

Full Stack app that includes SQL Database, Express, NextJS, Solidity Smart Contract, and Swift.

You can use any resources on the internet, moodle, or past projects you have on your computer. You can not get any help from the teacher or your neighbor. This year we will allow a little help on the Smart Contracts since we didn't cover those quite as detailed as the rest of the material. You can take breaks and chat with your neighbors but do not share code with each other. This is an individual project.

The Client Pitch

We want a safe, efficient way to get together and play card games. We want to be able to verify the host on the blockchain without worrying about going to some random unknown person's house. We want to see all the current meetings in an app and get together and have fun.

The Package (Dev Ops)

This will be in one GitHub repository. You will need to organize it so that it makes sense. You will do a **git init** on the outer folder and remove all the internal git folders on any packages that might have one by default. We don't want any pesky submodules. When you create your Swift app, make sure to uncheck the git repository. When you are done with the project, you will add your instructors, jeremy.skrdlant@fhnw.ed, and morgan.pritchard@fhnw.edu, to your repository so that it can be graded. Do this first so that we can verify that everything is good to move forward. Make sure you push multiple times per day. Use a branching process that allows you to do a pull request when you have a part of the project working.

Time Management (Project Management)

You will have a rubric that has all the items you will be graded on. Copy the list of items on the left into a spreadsheet. Create a column called time estimate. This will be in T-Shirt Sizes, so easy stuff will be small, and items you think will take a long time will be an X-Large.

Add another column for estimated start date. You can go in any order you want. You just need to plan out which day you will be completing which activity.

You will have a third column that will be the actual time spent. Each day, fill out the time you spent on each task using a formula like (= 5 + 3), where each number is a different day. This will be in hours.

Sequence Diagram (Project Management)

This project has a lot of moving parts. The NextJS Web app will allow admins to add other admins on the blockchain. This will include their name and a hash of the address. The web app will also allow admins the ability to create new events that will be pushed to the database. The iOS App will be for local individuals to be able to search for game locations to play cards with friends. The backend will be able to verify a location for a person by communicating with the blockchain. Your sequence diagram can be a screenshot, a PDF, or an image file. Make sure your sequence diagram goes through the stories of a user finding a location to play and admins adding events.

This diagram should have the database, Express server, Next.js, blockchain, and iOS app as parts of it.

Design (Project Management)

Before any code, always comes the design aspect. For this project, you will need to create a Figma (High Fidelity) prototype in Figma. You should write down the Hex colors and font used in the Figma document for easy reference. Once you are done, export the Figma as a PDF. You should have both your Next.js and iOS App wireframes in one file.

Object Model (Backend)

Create a UML document that will act as a guide as you build your apps. What data will you store in the database? What structures will you need in your smart contract? This can be a draw.io, a keynote, or a picture of UML drawn out on paper.

Database (Backend)

Create an SQL database on Docker. Your database will contain who is running our meetup, and what location the meetup is happening at. You should also make sure that you store the dates and times of the meetups as well! When you are done with the project, download a backup of the SQL (You can get an instructor's help with this).

Express Server (Backend)

Your express server should be the glue between your SQL database and both your NextJS app and your iOS app. The NextJS will be adding data, while the iOS app will be reading and filtering the data. Make sure you use a post route when you are submitting data from NextJS to Express.

Smart Contract (Crypto)

An admin will be able to use their wallet to add other admins information. There will also be a wallet in the backend that lets one route work as verification. It would call a verify on the smart

contract based on the user and verify the address hashes correctly. Only current admins can add other admins.

Extra Credit – This is a quick contract. Write a small paper listing all the vulnerabilities that could happen in this contract in case someone wants to hack it. This can also include vulnerabilities throughout the entire app.

NextJS Web App (Front-end/Crypto)

This part of the application will act as an admin console. We will be adding events through the backend to the database. The admins location hash will be able to be fetched from the smart contract and then verified with the address in the database. This console will also allow admins to add other admins.

iOS App (Front-end)

Your iOS app will work with your Express server to show data to customers. They can get a list of all the meetings where users can play games. Bonus if you can filter on certain locations close to you. (Super bonus if you use GPS and CoreLocation services)

Presentation Video (Project Management)

If you press Command + Shift + 5 on your Mac, you can record the screen. Let the instructor know if you need a microphone. You will have your entire project running. Then go through the application, showing the Next.js app and the iOS app working. Also, show the data going into the database. This should be about a 5-minute video.

Reflection Paper (Student)

Write a one-page reflection paper on this project. Which items took longer than expected, and which items were easier than you thought? Were there any parts of the application that you wanted to make better? What were some of the key things you learned while you were going through this project? Which areas do you want to study more?

Submission (Dev Ops)

Make sure everything is in your git repository. Share it with your instructors and make sure they verify they got it. Sit back and relax, and enjoy the summer. (DO NOT BRAIN DUMP OVER THE SUMMER). Find little things every once in a while to practice your skills.