IBF 2023 Mini Project

Name	**Theophania Lee Ning**
Email	**phalee40@gmail.com**

Date: Monday June 17 - July 18 2024

Project planning: https://miro.com/app/board/uXjVK5LZgic=/

In this final project, you are to design and develop an application of your choice using what you have learnt from this course.

The project duration starts from June 17 to July 18 2024; July 18 2024 is the presentation day.

Your application should consist of a minimum of 2 logical parts:
- Frontend - Angular
- Backend - Spring Boot

Your application must use technologies listed under Mandatory and Optional below.

The mini project will be graded according to the 4 categories:
- Application (50 marks) - How well thought out is your application, how you have implemented the application features and creativity. Think of this criteria as trying to sell your application to a VC (venture capitalist) to get money
- Aesthetics (20 marks) - Look and feel of the application
- Mandatory requirements (10 marks) - fulfilling the mandatory requirements
- Optional requirements (20 marks) - fulfilling the optional technical requirements

Mandatory

You must use **all** of the following requirements in your application.

Mark a cross X on the box beside each item to indicated that you have fulfilled the mandatory criteria

Angular	
Forms (either reactive or template driven)	x
Use GET, POST, PUT, DELETE (3 or more) to communicate between frontend and backend	x
Single Page Application (client side routing) with a minimum of 4 views	x
Abstract common functionalities into Services	x
Use Component Store for client side application state management	
Include an application manifest	x
Spring Boot	
Use of POST to handle either x-www-form-urlencoded and/or JSON and/or multipart payload	x
Making HTTP request to external RESTful API	x
Parameterized routes	x
Query string	x
Must support more than 1 user	x
Database	
Must use MySQL	x
Modeling data relationship: 1 to 1, 1 to many	x
Demonstrate data integrity and consistency when updating multiple tables	x

Angular	
Forms (either reactive or template driven)	x
Use GET, POST, PUT, DELETE (3 or more) to communicate between frontend and backend	x
Single Page Application (client side routing) with a minimum of 4 views	x
Abstract common functionalities into Services	x
Use Component Store for client side application state management	
Include an application manifest	x
Must use a different second database eg: Mongo or Redis	x
Deployment	
Containerized your Angular and Spring Boot application	x
The application that you have developed must be publicly accessible. You can deploy your application either as 2 separate deployments, frontend and backend or as a single deployment where the frontend is served from the backend. Applications can be deployed to Railway or any equivalent cloud PaaS platform like Heroku, AppEngine, etc If Angular is deployed separately, they can be deployed to static web hosting sites or JAM platforms like Vercel, Cloudflare or serving it from your hosted web server. Note: you cannot use Github pages for hosting your application	x
All databases must be deployed to the ‘cloud’. They can be deployed as VMs in public cloud or using managed database services	x

Optional

Your application should accumulate a minimum of **25 points** from the lists of optional features. This 25 points is not the marks of your project but rather a gauge of how many optional requirements you have implemented in your project. So your project must have a minimum of 25 points worth of optional requirements.

Mark a X on the selection options.

Backend Optional Requirements	
Use web sockets (8pts) in Angular and Spring Boot	x
Integrate with any API that requires OAuth2 authentication (4pts)	
Integrate with Google calendar or Drive (6pts)	
Bots eg Telegram, Slack, (6pts)	
Firebase web notification with frontend web notification (10pts). If you are using Firebase as a database, then you can ignore the mandatory second database requirement.	
Include simple AI into your application. Must be model based (not a series of if/then/else) or use 3rd party AI service (5pts)	
Use another NoSQL database not taught in the course: eg. graph, vector, timeseries, etc (6pts). If you are implementing this option, then you can ignore the mandatory second database requirement.	
Sending email (3pts)	x
Use Spring Boot security with JWT to authenticate and authorize Angular request (4pts)	x
Integrate with Ethereum's smart contract. You must also write the smart contract with Solidity (12pts)	
Ingesting and processing messages from a 'real' queue eg. Kafka, RabbitMQ, etc not Redis. Message brokers must be running in the Cloud (6pts).	

Backend Optional Requirements	
Integrating with payment gateway eg. Stripe, Paypal, etc (6pts)	
Add application metrics into your Spring Boot application. The metrics should be viewable from Prometheus. You should implement a minimum of 3 different metrics (6pts)	
Expose your Spring Boot application APIs as gRPC interface. You must include the protobuf files (8pts).	
Integrate your project with another member's project in the cohort eg. your application is a ecommerce store front; the other application is warehouse management. Integration must include the following - Both applications must communicate with each other - Integrate with the core functionality of the application Your application can integrate with more than one project in the cohort. +1 for each application that you integrate with up to a maximum of 7 points. (5 - 7pts). Official name(s) person's application you are integrating with 1. 2. 3. 4.	

Frontend Optional Requirements	
Use any Javascript/Typescript game framework eg. Phaser3, LittleJS, Kaboom, etc. (6pts)	
Bundle Angular application as iOS or Android application with hybrid app tools like Capacitor, Cordova, NativeScript (8pts)	
Use map eg Google Map, must be dynamic (4pts)	
Use a UI component framework - eg. ng-bootstrap, Material, PrimeNG (3pts)	x

Use state management libraries like Akita, NgRx, NGSX, etc (6pts). If select this requirement, then you can ignore the mandatory Component Store requirement	x
Adding a service worker to precache application assets (4pts)	

Deployment Optional Requirements	
Containerize your application and deploy into a Kubernetes cluster. Your application should be exposed with an Ingress resource (8pts).	
Apply a domain name and configure your application to use the domain name (2pts)	x
Use Github actions for continuous build and continuous deployment to automatically build and deploy your application (6pts)	

If you have ideas or requirements that are not on the above list, please discuss with the instructors before using it in your application. The instructor will assign a point to your requirements. You cannot assign points to your own requirements.

Description of your requirements	Pts

Presentation

You will be presenting your application on the days from **Thursday July 18 2024**. A presentation schedule will be sent closer to the date. Presentation is to be conducted in-person and not over Zoom. You will have 10 mins to show off your application. You only have 1 opportunity to present. There will be no makeup if you miss the presentation.

You will be presenting the application on a provided computer; your project must be deployed before your presentation. You cannot present your project running on your notebook.

Submission

You must submit your assessment by pushing it to your repository at either GitHub, GitLab or BitBucket.

Only commits before **0900 July 18 2024** will be accepted. Any commits after **0900 July 18 2024** will not be accepted. No other form of submission will be accepted (eg. ZIP file).

Fill up the form and put a cross on all the mandatory requirements and optional requirements that you have used.

Generate a PDF copy of this mini project document and rename it to your official name (as in your NRIC) and upload it to the `#miniproject` channel in the couse Slack channel.

This document must be posted to the `#miniproject` channel no later than Mon 0900 July 8 2024 for your project to be graded. No marks will be awarded for your project if the requirements is not posted to `#miniproject` channel by **Mon 0900 July 8 2024**.

You cannot change your project requirement (this document) after **Mon 0900 July 8 2024**.

Remember to generate a PDF of this document and rename the PDF file to your official name (as in your NRIC). Otherwise your submission will not be accepted.

Academic Integrity

The assessment must be your own work. You cannot ask a third party to write any part of this assessment or use AI tools such as ChatGPT to generate output and submit it as part of your assessment. This will result in an automatic disqualification from the assessment.

The NUS ISS takes a strict view of cheating in any form, deceptive fabrication, plagiarism and violation of intellectual property and copyright laws. Any student who is found to have engaged in such misconduct will be subject to disciplinary action by NUS ISS.