Prueba Técnica Frontend

AXPE Consulting

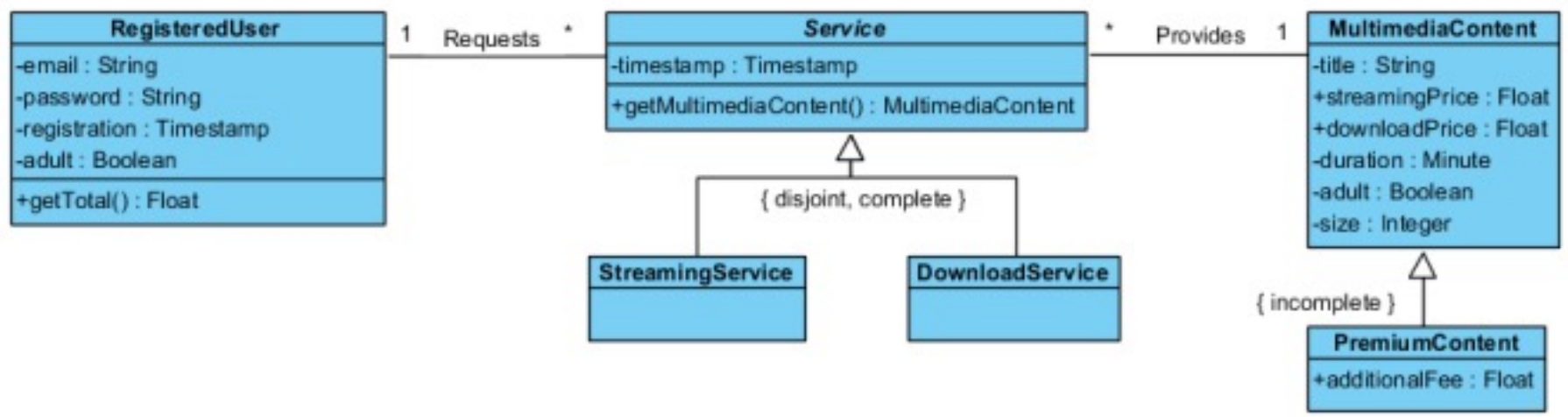
Introducción

A continuación te proponemos dos ejercicios para evaluar tu nivel técnico.

El primer ejercicio es una parte mas “teórica” y en el segundo, pasamos a una parte mas práctica con un pequeño proyecto.

Ejercicio 1

Disponemos ya de un análisis previo que podemos utilizar como punto de partida, pero que habrá que corregir y mejorar. Se trata de un software para gestionar el streaming de películas en línea.



Como podemos ver en el diagrama de partida, los usuarios tienen una operación que devuelve el importe total pagado por todos los servicios que han solicitado. El precio de un servicio se calcula de la siguiente manera:

- Para todos los usuarios que quieren ver un contenido multimedia por streaming, se aplica el precio de streaming de este contenido multimedia.

- Para todos los usuarios que quieren descargar un contenido multimedia de la plataforma, aplica el precio de descarga de este contenido multimedia.

- Si el contenido es premium, en cualquiera de los casos anteriores se añade el cargo adicional especificado en el atributo *additionalFee*.

Supongamos que la implementación de este método es como el siguiente pseudocódigo:

```
class RegisteredUser{

    constructor(services = []){
        this.services = services;
    }

    getTotal (){
        let total = 0;

        this.services.forEach(service, index => {
            let multimediaContent = service.getMultimediaContent ();

            if (typeof service == StreamingService) {
                total += multimediaContent.streamingPrice;
            }else if (typeof service == DownloadService) {
                total += multimediaContent.downloadPrice;
            }

            if (typeof multimediaContent == PremiumContent) {
                total += multimediaContent.additionalFee;
            }

        })

        return total;
    }
}
```

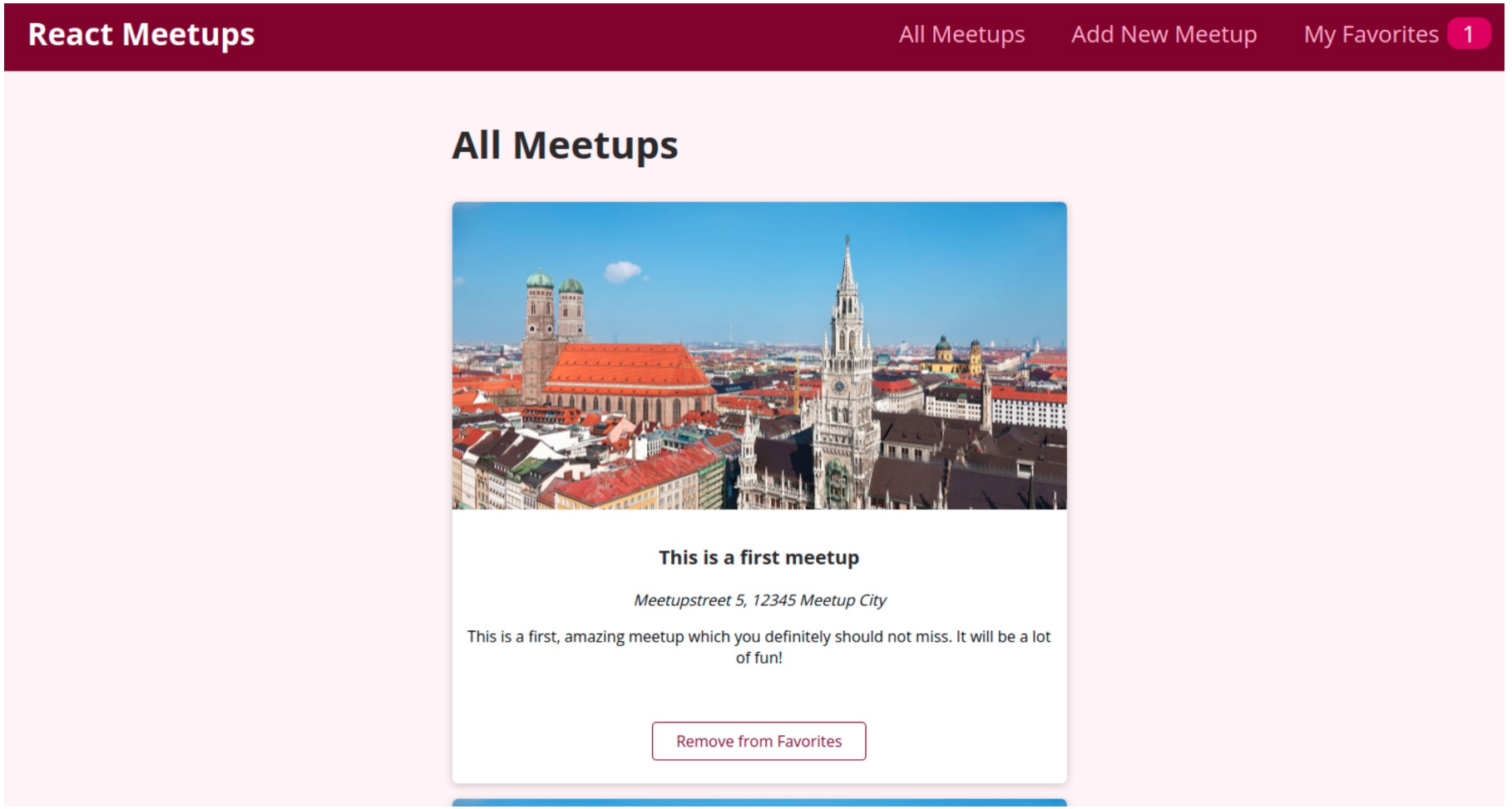
Revisamos el pseudocódigo de la operación *getTotal* de la clase *RegisteredUser* y nos preocupa que el su diseño sea un poco frágil ya que no vemos claro si contempla los posibles escenarios futuros y su impacto:

1. Que problemas detectas en la operación y razona la respuesta

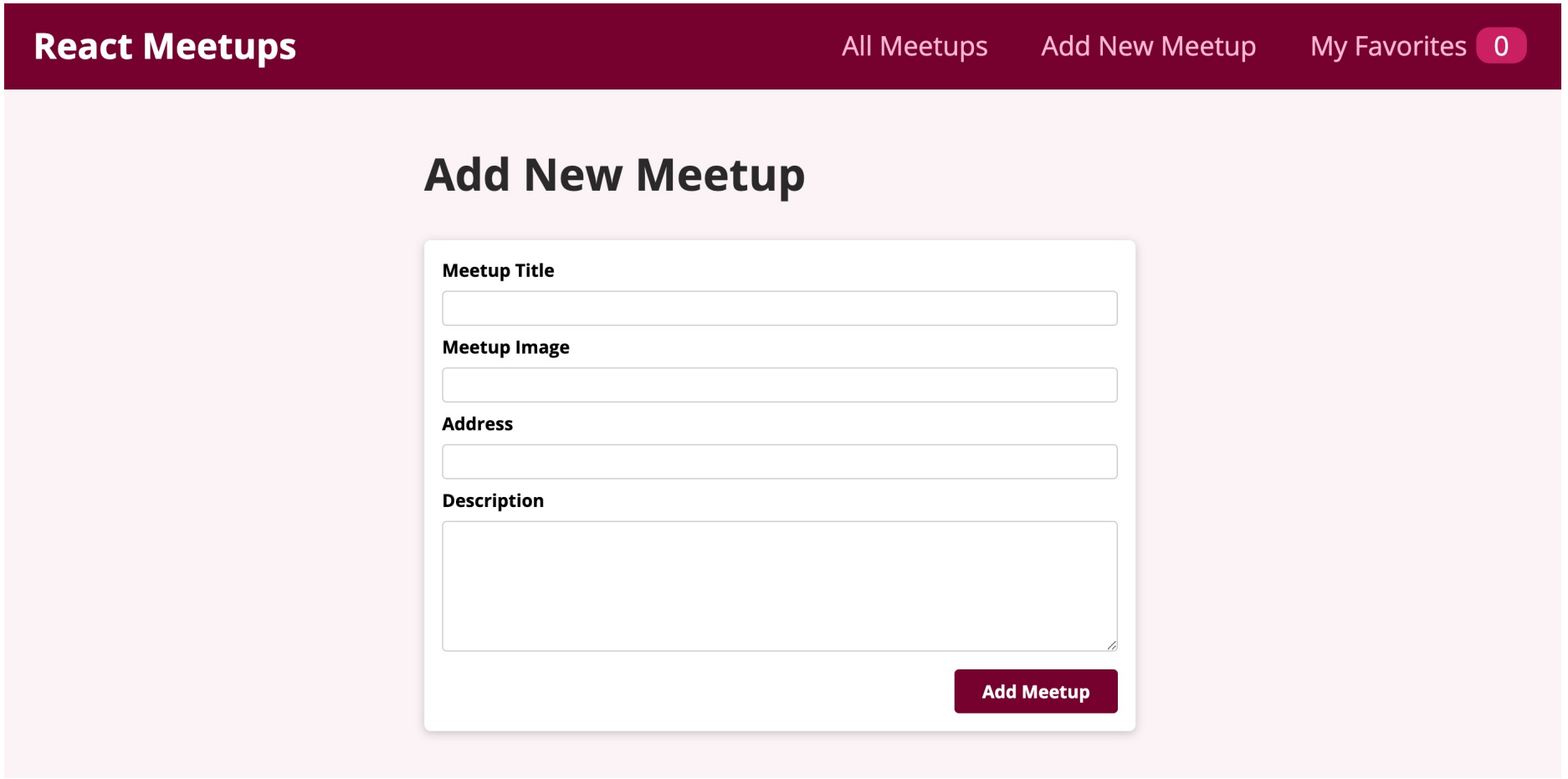
2. Propón una solución alternativa (también en pseudocódigo del mismo estilo) que corrija los problemas de la operación *getTotal* de *RegisteredUser* que has detectado en la pregunta anterior. Realiza todos los cambios que consideres necesarios en cualquiera de las clases del modelo del enunciado.

Ejercicio 2

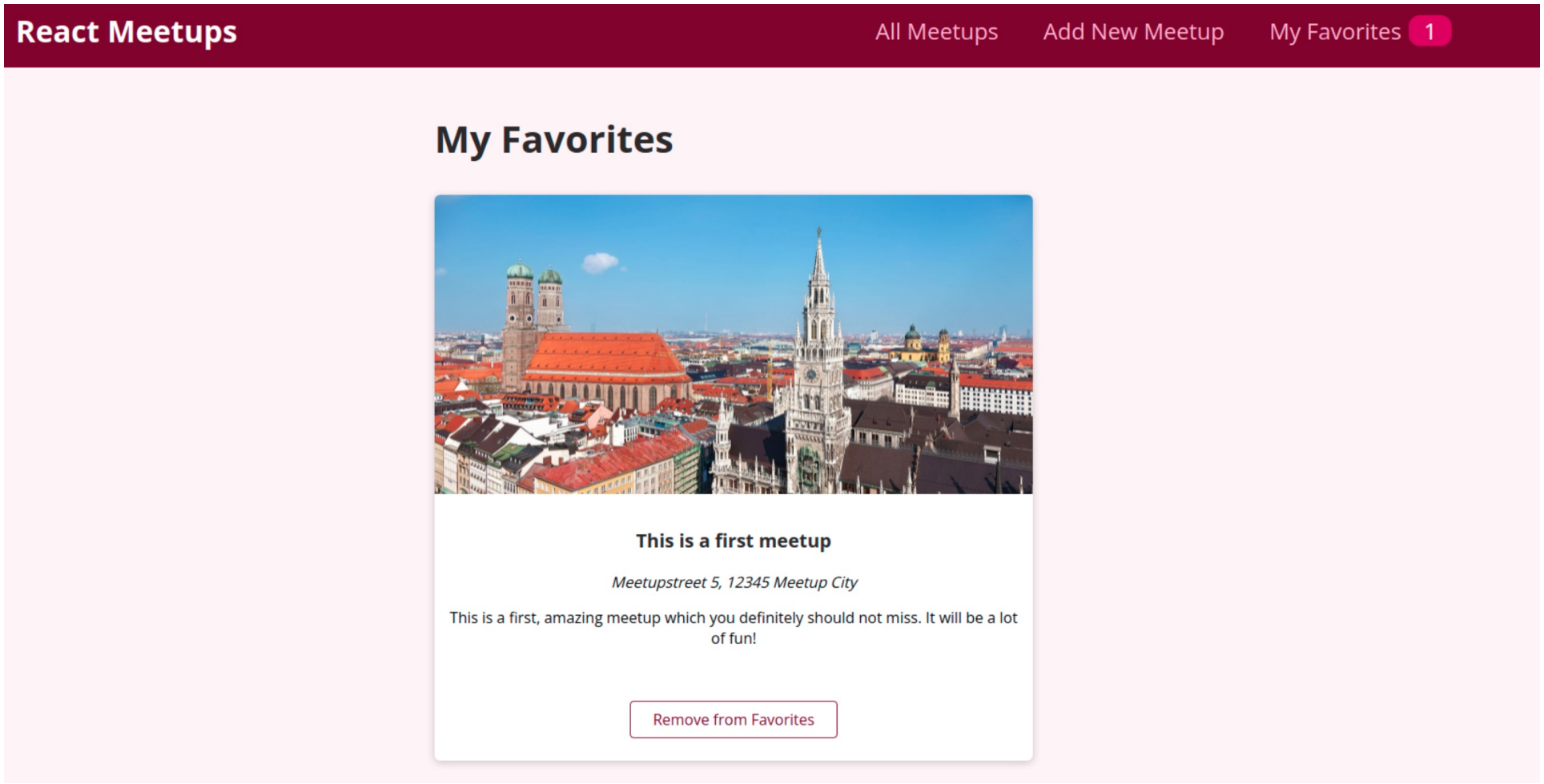
El fichero .zip que te adjuntamos, contiene un proyecto basado en una aplicación de ‘meetup’. El proyecto se encuentra en un estado bastante inicial dónde intentamos simular una posible situación, dónde un programador empezó hacer el desarrollo pero que lo ha dejado a medias, ya que por necesidades de negocio ha cambiado de proyecto y necesitamos que se termine a partir de los requisitos y condiciones que proponemos.



Una pantalla inicial dónde se ven todos los meetups que tenemos. Tanto los iniciales como los nuevos que se añadan.



Una segunda pantalla con un formulario para añadir nuevos meetups.



Una tercera pantalla dónde se ven todos los meetups que hemos seleccionado como favoritos.

Requisitos y condiciones:

Como el proyecto está en un estado inicial verás que hay que modificar llamadas y manejar estados. No te cortes a la hora de hacer los cambios que consideres oportunos para el buen funcionamiento de la aplicación. Puedes utilizar librerías externas sin problema.

- Para el header se requiere una animación para tener un acceso rápido a las distintas páginas cuando hacemos scroll. Se quiere que cuando hacemos scroll down, este tiene que desaparecer y cuando hacemos scroll up tiene que volver a aparecer en la posición de la página dónde te encuentres.
- Desde el header se puede navegar a las distintas páginas, pero por temas de SEO se requiere que esta navegación se vea reflejada en la url. (Ejemplo: la página de favoritos podría ser /favorites)
- El botón de añadir a favoritos no está funcionando. Implementa la lógica para añadir y quitar de favoritos.
- Implementación de algún test. Puede ser e2e, unitario o funcional. En el proyecto encontrarás algunos test que el programador que inició el proyecto, empezó a hacer.

El proyecto tiene que estar disponible en un repositorio git accesible, para que podamos ver el control de versiones. Además, el *README* deberá contener una explicación breve de la solución implementada y de como podemos ejecutar el proyecto.