Rich Web Application

Lab 3

# Question 1

*Explain what is meant by the stream abstraction. What is the relationship between streams and the observer pattern?  What are streams useful for modelling and when might you use them in Rich Web development?*

A stream is a sequence of objects, but the items aren’t retrieved until they’re required. This means that unlike a normal List, it doesn’t have to be finitely long. You can put anything in a stream; characters, numbers or even user events and you can manipulate it in lots of different ways. A stream is built on top of an observer pattern, it acts as a mediator between the user and the interface.

You might see streams being used when dealing with HTTP requests

# Question 2

*Assume that you are building an interface to an API in your Rich Web App. Describe in detail how you could use the RxJS library to handle asynchronous network responses to API requests. In your opinion, what are the benefits to using a streams library for networking over, say, promises? And what do you think are the downsides?*

We create the request stream, by turning the desired url into an observable object. We create another stream, the response stream on the same url, and use flatmap to combine the meta stream and this new stream. The response from the url can be recorded in the response stream

These streams, also called observables can listen for 0 to many events, where as a promise can only work for 1 event at a time, so you need multiple for multiple events. You can also cancel a stream, but cannot cancel a promise. A downside of streams is that, unlike promises they cannot be chained.