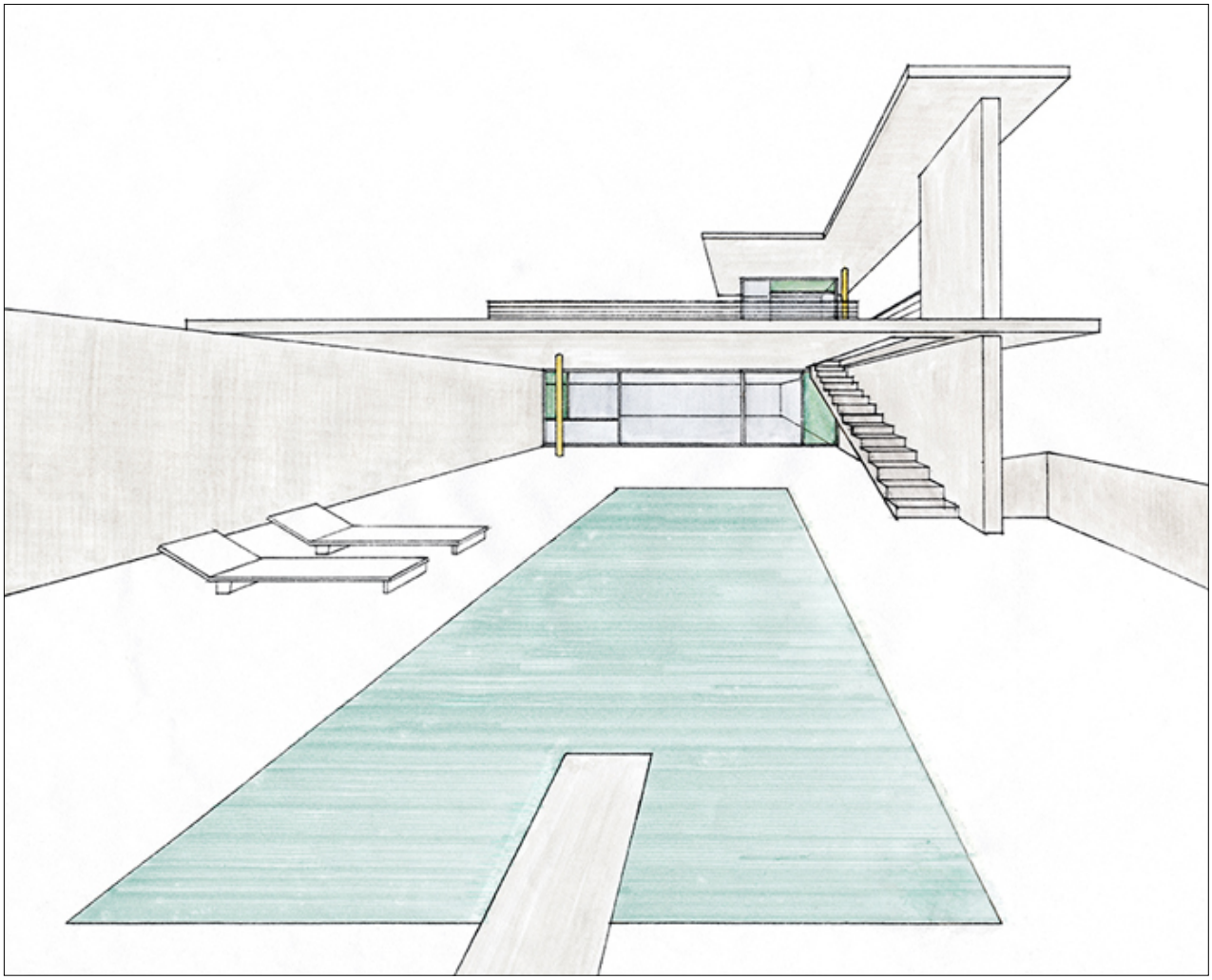
TRIANGLE CHALLENGE



Trigon-App

Prepared for: TradeShift

Prepared by: Nawanshu Choudhary

19 March 2019

TASK SUMMARY

Objective

Write a program that will determine the type of a triangle. It should take the lengths of the triangle's three sides as input, and return whether the triangle is equilateral, isosceles or scalene.

Font-end Technologies

I have decided to program the solution on Core Front-end language i.e. JavaScript 2015/ ES6 along with TradeShift UI.

Why ES6?

ES6 is a modern javascript language for dynamic web apps, build with the object oriented design in mind. Provided application is very small so have to take care for the trade off between Over engineering and Under engineering. Here, in task, we need to build a solution where there is no tricky DOM manipulation, rather, there are read and update operations as a major task.

Additionally, we are displaying design skills with Object Oriented Design, so ES well written for that, so I have decided to chosen ES6 over others.

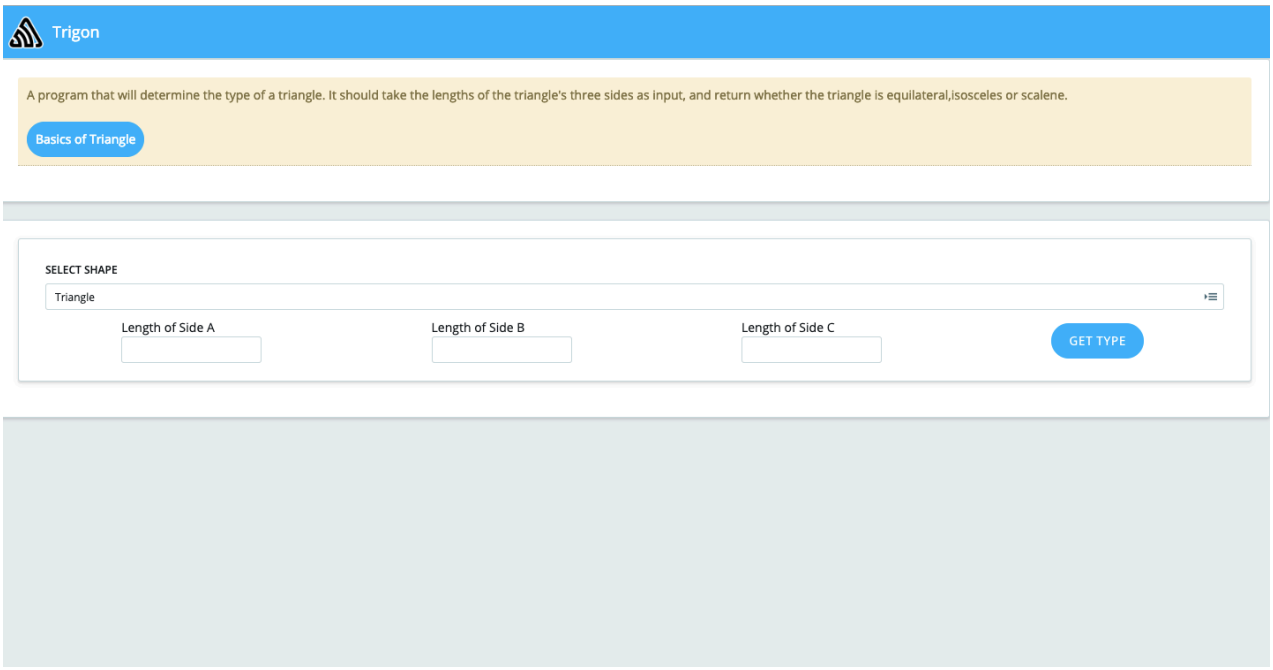
Supporting tools and technologies used

- ES6 (ES2015)
- ui-tradeshift 12.2.0
- JavaScript
- CSS
- HTML5
- Integrated development environment: Visual Studio Code 1.31.1
- Build tool: Webpack 4.29.6
- Transpiler: Babel 6/7
- Testing Tools: Mocha 6.0.2, Chai 4.2.0
- Package Manager: NPM (Node Package Manager) 5.6.0
- Node 10.0.0

Application Flow Design

Application have 4 form field in total, where dropdown provided you for the extensible development approach in mind. You can choose any shape which you like. For now, only triangle functionality provided to keep focus on the assignment.

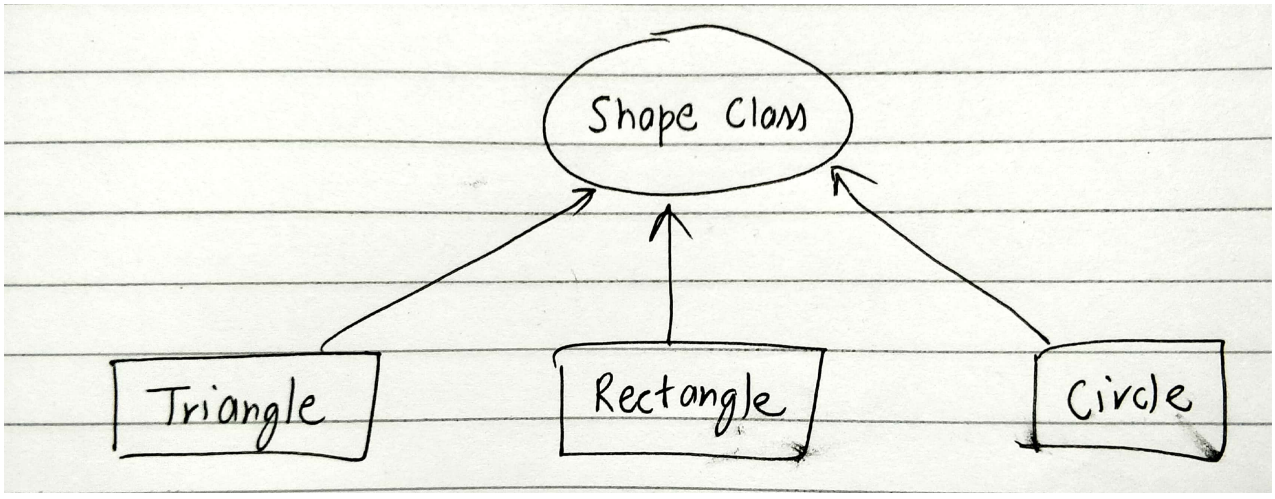
After Choosing Triangle (Default Selection), you will be provided 3 input field to enter side length, where you are only allowed to enter numeric values. Once you are done with the input, go ahead and click on the GetType button, You will get the correct output, considering input validity, triangle validate with edge cases.



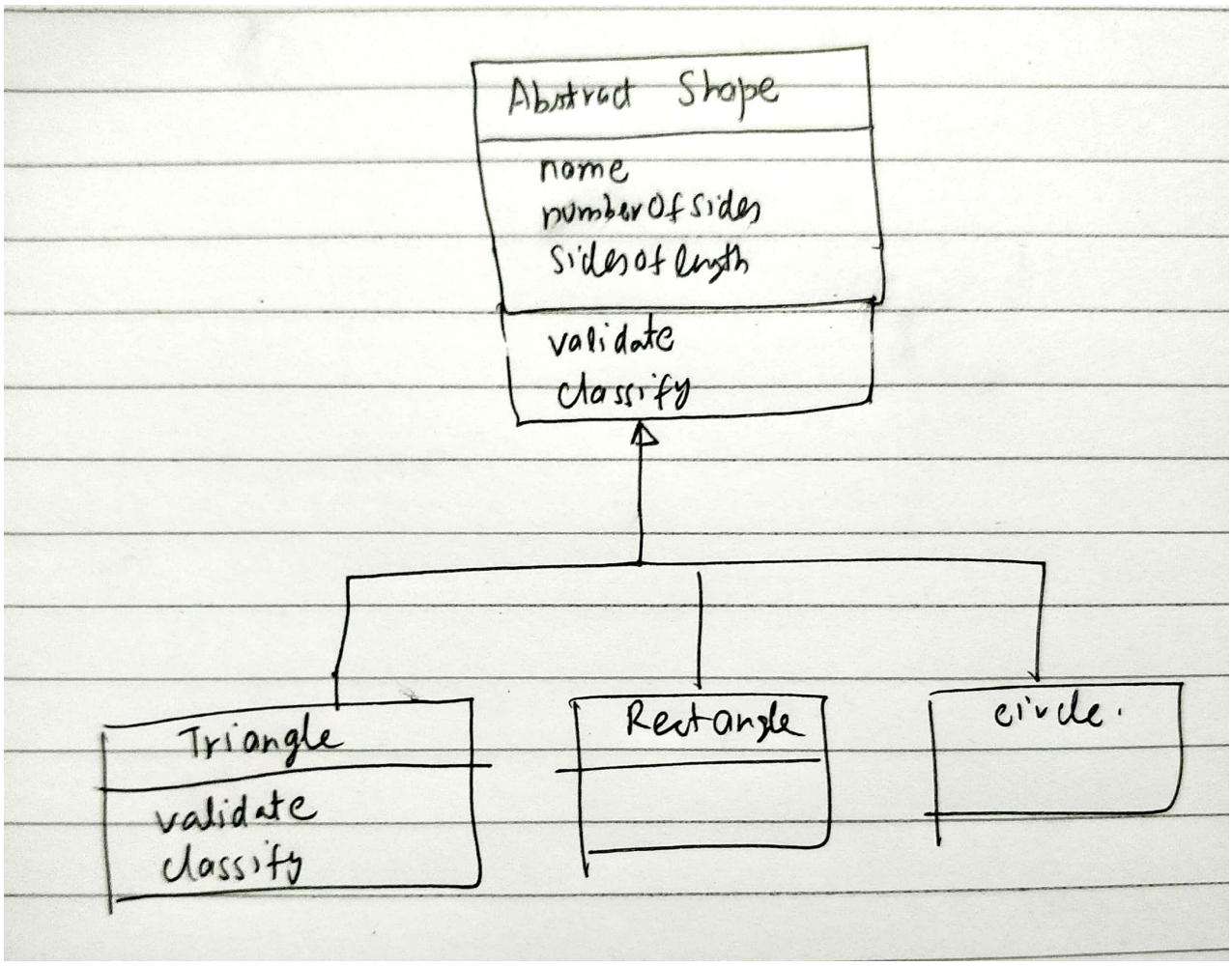
Application Architecture Design

Bases on the problem statement, I have planned to used Object Oriented Analysis and Design method.

Why OOAD:
Object Oriented Programming (OOP) refers to using self-contained pieces of code to develop applications. We call these self-contained pieces of code **objects**, better known as *Classes* in most OOP programming languages and *Functions* in JavaScript. We use objects as building blocks for our applications. Building applications with objects allows us to adopt some valuable techniques, namely, **Inheritance** (objects can inherit features from other objects), **Polymorphism** (objects can share the same interface—how they are accessed and used—while their underlying implementation of the interface may differ), and **Encapsulation** (each object is responsible for specific tasks).



Along with OOAD tried to showcase few SOLID principles especially Single Responsibility Principle, which says about individual responsibility of class and Open closed principle which says entities should be open for extension but closes for modification.



For example: If you want to introduce one more shape to this design then you just have to extend a functionality in the existing architecture and no need to modify the existing code.

Tried to show less Rippling in the application to showcase better OOAD design.

This Application is just an idea, how to implement this small problem to some big level. One thing I must say you have an awesome documentation of your tradeshift UI, haven't focus much on the design side but enjoyed a lot while using your own tradeshift UI.