

## Basic Concept

### 1. Coding

#### A. Code smell: Too many functionalities in class or method

Explanation: Too many functionalities is not a bug and will not show as an error in any debugger or console. However, too many functionalities mean that one class maybe have too many responsibilities given to it. It is true that class follow open/closed principle, which means open for extension but closed for modification. However, if one class have too many responsibilities, it is susceptible to change. For example, a class that responsible for getting a data from database and process the data until become a readable format, it has too many responsibilities from retrieving data, parsing, formatting and others. If the programmer needs to change one function, it can produce a side effect because of human fault or a technical fault. Furthermore, modify a complicated class (a class that have too many functionalities) need more effort. This code smell I identified is a code smell that deviates from SOLID design principle.

#### B. Dependency injection means methods that bestow responsibilities to create an object that someone (class) depends on other class. It makes a dependent class that depend on particular object independent from creating the object because the task to create the object done by the others. One benefit of using dependency injection is that the object can be modified at runtime.

### 2. Rest API

#### A. Do's and do not example

##### i. POST

do :

do not:

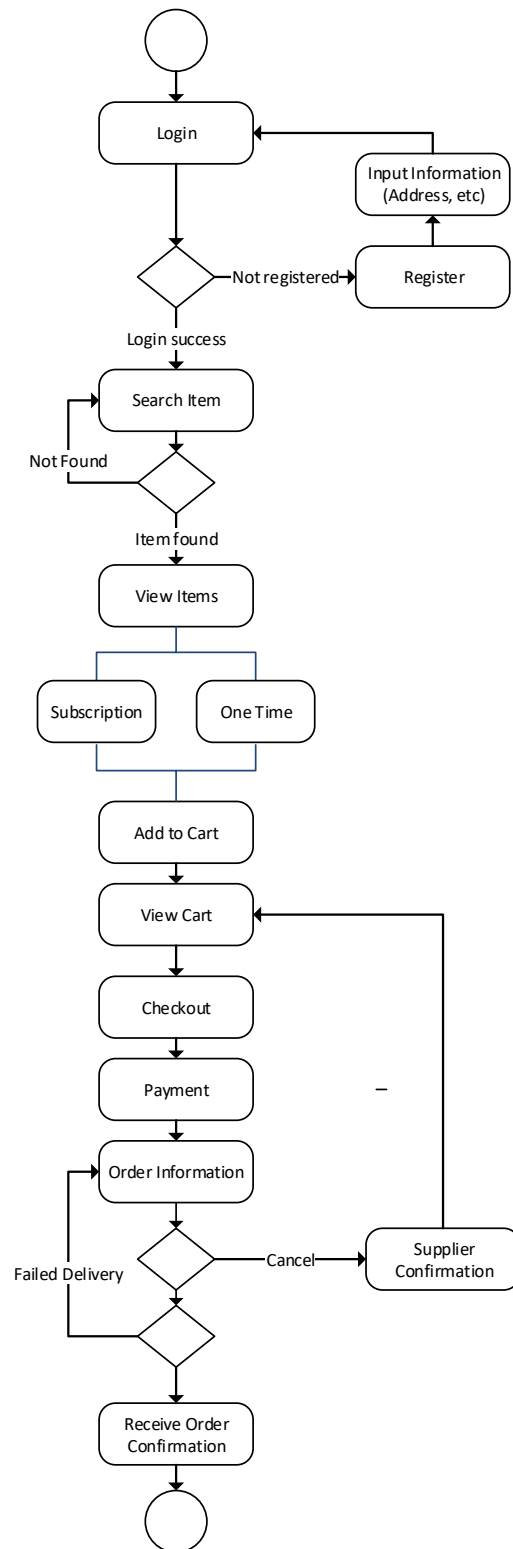
##### ii. GET

do: take into account the data length

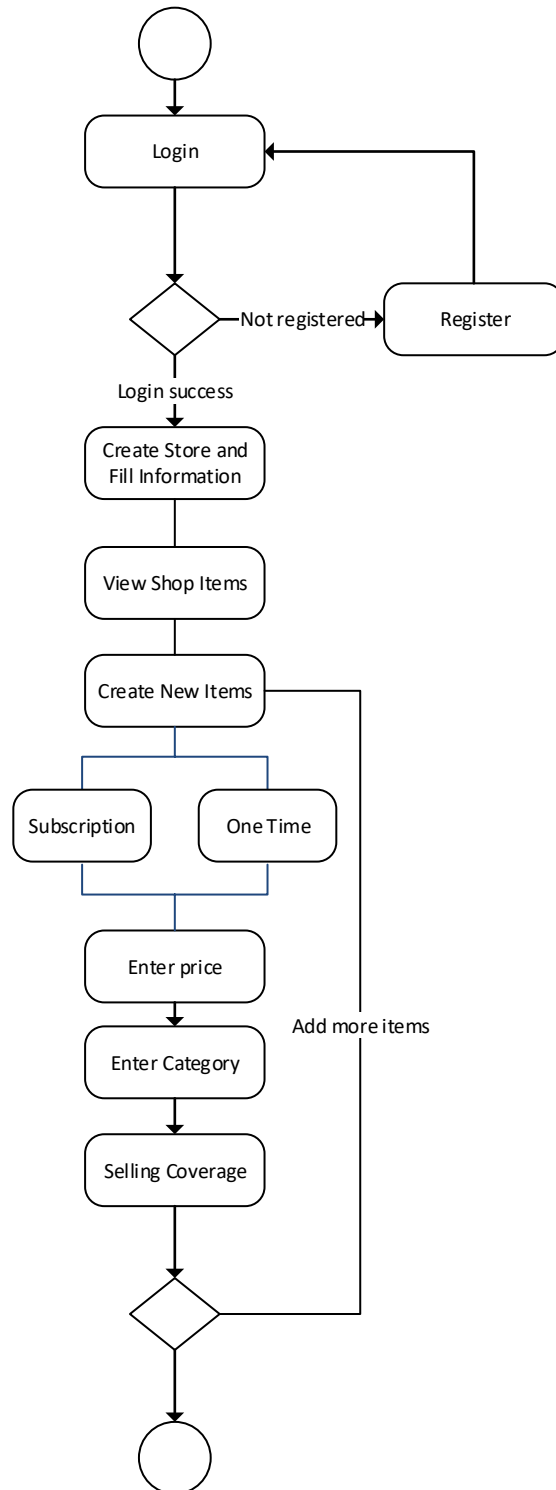
do not: transfer data with format other than text

## Basic Coding

### User Side Activity Diagram



## Supplier Side Activity Diagram



Algorithm

Answer in separate compressed files