**STEPS FOR ARRIVING AT THE SOLUTION**

1. Understanding the challenge and business logic

* Widening tax net by recruiting new taxpayers operating unregistered businesses

2. Modelling the business logic using OOP concepts

* Objects involved (Models): TaxPayer, FieldOfficer
* Identifying attributes for Models
  + Tax payer
* TPIN
* BusinessCertificateNumber
* TradingName
* BusinessRegistrationNumber
* MobileNumber
* Email
* PhysicalLocation
* Username
  + Field Officer
    - Username
    - Password
    - FirstName
    - LastName
    - email
* Identifying relationships between Models
  + A Field officer manages a tax payer i.e. creates, edit, and deletes a tax payer in the system
* Design input forms for each relationship
* Set up validation for each input form
* Set up a service to persist data from input forms
  + In our case, the service should consume the given webservice
* The service should use http to communicate with the webservice
* The status of the communication will be guided by codes e.g. 200 for success
* The http service will rely on other service to provide it with information
* To avoid tight coupling between services, dependency injection will be observed
* All settings should not be stored in code rather in app settings file