# Software Detail Design

## SDD

* Provide an overview of this project.( cung cấp cái nhìn tổng quan về dự án)
* Providing the details for how the software should be built. (cung cấp chi tiết về cách xây dựng phần mềm)
* Including: class diagram, sequence Diagram, object diagram, collaboration diagram and other supporting requirement information.

## Why do we need SDD

* Evaluate the description of the details for the software and system to be built.
* Understanding that what is to be built ?
* Understanding that how it is expected to built ?

## Class diagram

* Static structure of the classes in the system.
* The classes have relationship between like associated, dependent, specialized and packaged.
* Ingredient: name, attributes and method.

#### Relationship

* Association: sự liên kết giữa 2 lớp khi mà không ai sở hữu ai
* Aggregation: có mối quan hệ sở hữu
* Composition: một phần của class này là một phần của class kia
* Generalization: quan hệ kế thừa

## Sequence diagram

* Simply depicts interaction between objects in a sequential order
* Describe how and in what order the objects in a system function.

#### Ingredient

* Actor or User.
* Object or Class.
* Life line: represents an individual participant in the Interaction.
* Message: defines a particular communication between Lifelines of an Interaction.

#### Message types

* Synchronous Message: need to a request before action
* Asynchronous Message: No need any request.
* Self Message: represents the invocation of message of the same lifeline.
* Return Message: represents the pass of information back to the caller of a corresponded former message.
* Create Message: represents the instantiation of (target) lifeline
* Delete Message: represents the destroying of (target) lifeline.

## Design patterns

* MVC: Model – View – Controller
* MVVM: Model – View Model – View – Controller
* VIPER: View – Presenter – Interactor – Entity - Router