# Design Pattern

Considering our application is used by hotel stuff in different scenarios such as making reservations, creating orders, etc. Different boundary classes are created for different use cases. Therefore, Factory Design Pattern is implemented in the top layer of our system.

The benefits and OO concepts we implemented Factory Design Pattern are:

1. Use polymorphism for boundary object creation and encapsulate the boundary creation from clients.
2. Easy to extend, which satisfies the open-closed principle. For example, we can extend our application to support checking employee information of the hotel by adding the Employee Boundary with the corresponding entity and control classes without affecting other objects and classes.
3. Support lazy initialization and singleton pattern, which the returned boundary object is based on the selection of user.

From entity layer to boundary layer, Entity Control Boundary (ECB) pattern is implemented to support different use cases for hotel stuff. In our application, one type of entity class has a corresponding control and boundary class, and all the communications between the boundary class and entity class are processed in control classes.

The benefits we implemented ECB Pattern are:

1. Separate the data layer and user interface layer to reduce the coupling.
2. One type of entity class has one control class, single responsibility with high cohesion.