# System Design Document – MyBar

In our project we are using the MVC-model as design-pattern, by dividing the classes into packages named “Model”, “View” and “Controller”. By divide the classes it is easier to control what they are doing because they are in a certain package that gives information about what the classes does.

## Packages

### se.turbotorsk.mybar (View)

**AboutBox** – Handles the About-information  
**AddIngredientActivity** – Is used when you want to add an ingredient   
**CollectionActivity** – Handles the Collection-view  
**DrinkAdapter** – Handles the drinks  
**IngredientAdapter** – Handles the ingredients  
**MainActivity** – Makes the main-window with menus  
**MyBarActivity** – Handles the MyBar-view  
**MyFavorites** – Handles the MyFavorites-view  
**Profiles** – Handles the profiles  
**SettingsActivity** – Is used when you want to change settings  
**Share** – Handles the share-function  
**ViewDrinkActivity** – Handles the drink-view

### se.turbotorsk.mybar.controller (Controller)

**Controller** – Handles the communication between model and view  
**DrinkManager** – Handles the drinks you can do with the ingredients you got in the MyBar-view  
**MyBarApplication** – Handles communication between classes

### se.turbotorsk.mybar.model ( Model)

**Data** – Handles the set- and get-methods between database and controller   
 **Drink** – Handles variables for a drink  
 **Ingredient** – Handles the variables for an ingredient  
 **MyBar** – Handles the variables for the content in the MyBar-view

### se.turbotorsk.mybar.model.database

**DrinkTable** – Handles the table for the drinks  
**IngredientTable** – Handles the table for the ingredients  
**MyBarContentProvider** – Handles the communication between database and the Data  
**MyBarDatabaseHelper** - Manages the database  
**MyBarTable** – Handles the table in MyBar-view

### se.turbotorsk.mybar.externaldata

**JsonParse** – Fetch all the external data

## UML

This is the UML for all our packages:

