System improvements:

1. Assumptions that will not impact the design:
   1. AcuCafe business will grow and we expect a few hundred drinks and a few dozen topics will be added.
   2. For every drink, we will have a specific selection of topics.
   3. Data sources for the drinks and topics are going to change, but we do not know what type of data source will be.
   4. There is a strong possibility that there will be more channels to inform the barista.
2. Assumptions that will impact the design:
   1. Messages for success and error are going to change. If these are going to change they will be treated as bugs.
3. Design choices
   1. Used Decorator pattern <https://en.wikipedia.org/wiki/Decorator_pattern> we need to decorate Drinks with one or more Topics
      1. Created one ConcreteComponent class that generate SimpleDrink using dynamic data at runtime
      2. Created one Decorator class that generates Topic using dynamic data at runtime
   2. Used single point of truth DrinkDataSource for the Drink and Topic information, additionally system can be extended with more data sources by implementing interfaces.
   3. DrinkDataSource left hardcoded since there are only a few drinks and topics.
   4. AcuCafe class is used only as an orchestrator and system single point of entry.
   5. Used factory classes for Drinks and Topics
   6. Interfaces or virtual public methods exists for all classes in system
   7. Inversion on control applied to all classes used in the system
   8. IoC container registration added
4. AcuCafe main class:
   1. Add Drink OrderDrink(string drinkName, IList<string> topicNames) allow addition of new drinks and topics dynamically, without changing the function signature.
5. Drinks:
   1. Add or modify Drinks or allowed Topics in DrinkDataSource;
   2. Added helper methods to interrogate when it comes to drink names GetDrinkNames() in IDrinkInfoQuery interface;
   3. Added DrinkInfo class to store information about Drinks as aggregation root
   4. Extension Points:
      1. Create new implementation of IDrinkDataSource to change the Drink data source (database, config file …).
      2. Create new implementation of IDrinkInfoQuery if optimization improvements is necessary for new data source
6. Topics:
   1. Added or modify Topics in DrinkDataSource class.
   2. Added helper methods to interrogate when it comes to available topics GetAllowedTopicNames(string drinkName), GetAllowedTopicNames() in IDrinkInfoQuery interface
   3. Added TopicInfo class to store information about Topics.
7. Inform barista:
   1. Added InformBaristaOnConsole class to write information on console. Inherited from IInformBarista.
   2. Added InformBaristaOnFile class to write information on disk file. Inherited from IInformBarista.
   3. Added BaristaFileInfo to store information about file location and name.
   4. Added more information about the errors when they appear.
   5. Added log file to save all information about the system operations.
   6. Extension Points:
      1. Create a new class that implements IInformBarista to create a new messaging option