**Classes Explanation:**

The scope of this document is to list our critical classes involved in the design and the purpose of them.

Application is **started/booted** by **Application.java** which loads records from mongodb at the startup time.

**Application configuration** is maintained in a class called **AppConfig.java** , this class can be used in future to maintain any new properties added to **application.properties** which has a prefix of app.

The shuffle algorithm is defined in a property called app.algorithm = PairWise

**MainController.java** => Exposes rest methods for /PUT, /POST, /GET and /DELETE

This class is responsible for handling the http requests and pass to downstream classes.

**IDeckCardDataService.java** => Interface for Create Deck,Shuffle deck,Deal Card and Delete deck operations which is **implemented** by **DeckCardDataService**.

**DeckCardDataService.java** => Core service implementation which creates a deck,shuffles,deals and deletes a card, It uses in memory hazelcast map and updates mongodb asynchronously.

**MongoDeckRepository.java** => **JPA CRUD** interface which adds **asynchronous** behaviour.

**Deck.java** => Model entity to be persisted in **MongoDB**.

**ServerStartupLoader.java** => Loads records from MongoDB to in memory map during server startup.

**IDeckShuffler.java** => Interface for shuffle algorithms. This gives us advantage of new shuffling algorithms being added easily without change to the code thats making use of shuffle.

**PairWiseCardShuffler.java** => implements **IDeckShuffler** with a pairwise swapping being used for swapping, Algorithm selects two random cards and swaps them and repeats this process for 5 times.

**SupportedCards.java,SupportedSuits.java** => Enum for cards and suits(e.g Spade,Diamond etc.,)

**DeckOfCards.java** => logical representation of deck of cards arrangement, contains methods for creating a deck of 52 cards, dealing a card.