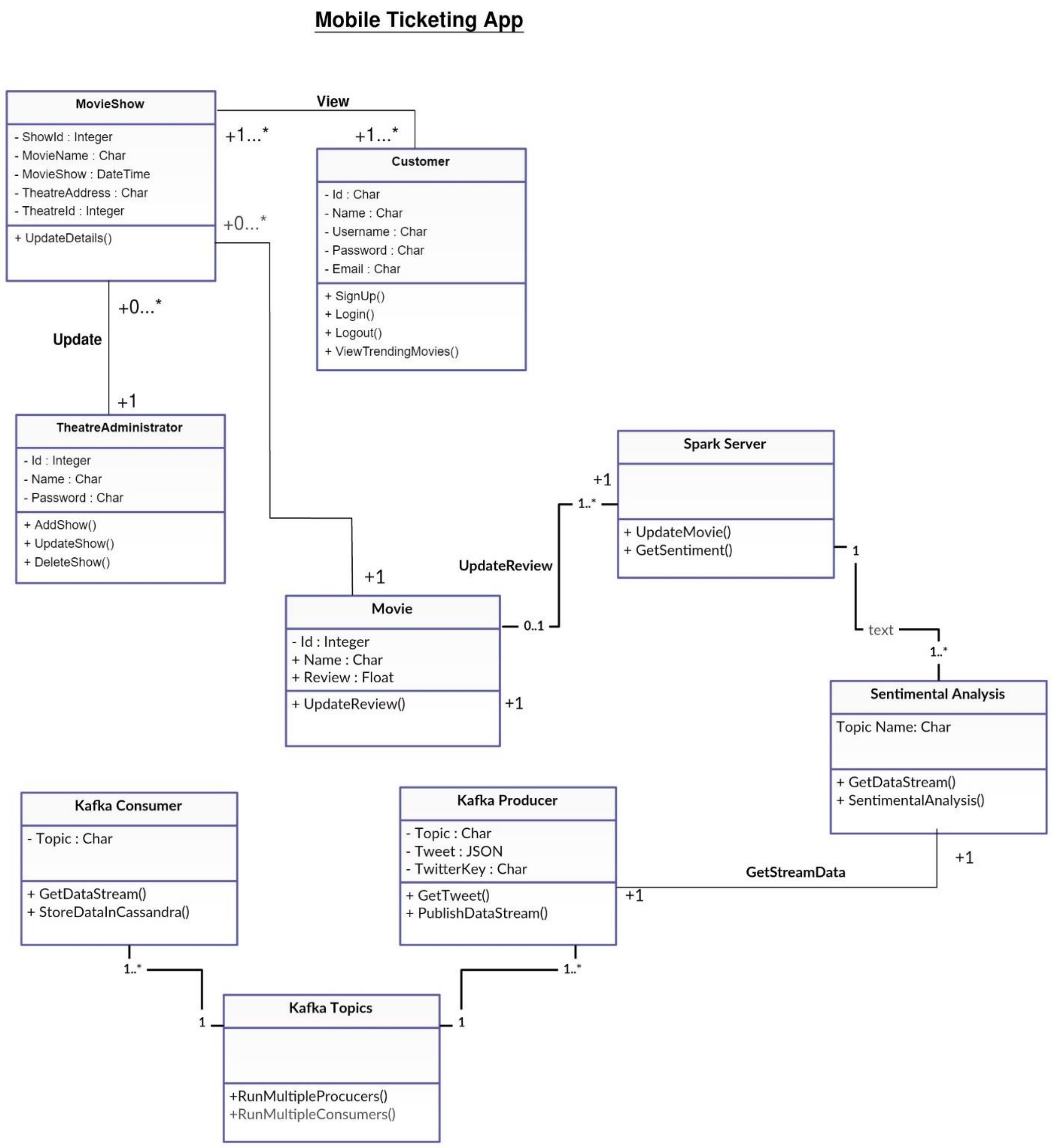
Product Design

Team 10 - Project 3

Nishanth Sharma - Divyesh Jain - Prateek Saini

Design Model

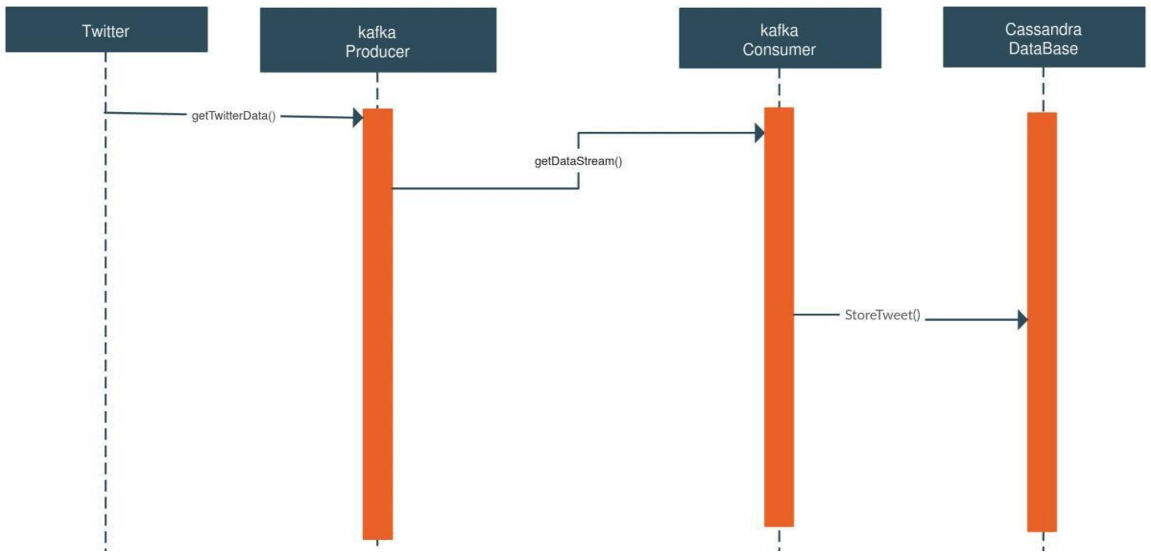


Customer	Class State • Id: Stores unique key for a customer. • Name: Stores unique string for each user. • User Name: Store full user name for each user • Password: Stores hashed password for each user. • Email: Stores email for each user. Class Behaviour • Signup(): Register a user. • Login(): Method to login a user. • Logout(): Method to logout a user. • ViewTrendingMovies(): Method to view trending movies.
Movie	Class State • Id: Stores Id of a movie. • Name: Stores name of movie. • Password: Stores reviews for a movie. Class Behaviour • UpdateReview(): Update reviews for a movie.
TheatreAdministrator	Class State • Id: Stores Id of a theatre. • Name: Stores name of theatre. • Password: Stores password for login into theatre page. Class Behaviour • AddShow(): Add movie shows. • UpdateShow(): Update movie shows. • DeleteShow(): Delete movie shows.
MovieShow	Class State • ShowId: Stores Id of a show. • MovieName: Stores name of the movie being shown in a particular show. • MovieShow: Stores date and time of a particular show. • TheatreAddress: Stores address of the show. • TheatreId: Id of theatre where show is being shown. Class Behaviour • UpdateDetails(): Method to update show.
SentimentalAnalysis	Class State • Topic: Store topic name. Class Behaviour • GetDataStream(): Method to get datastream from kafka server. • SentimentalAnalysis(): Method to determine sentiment of tweet.

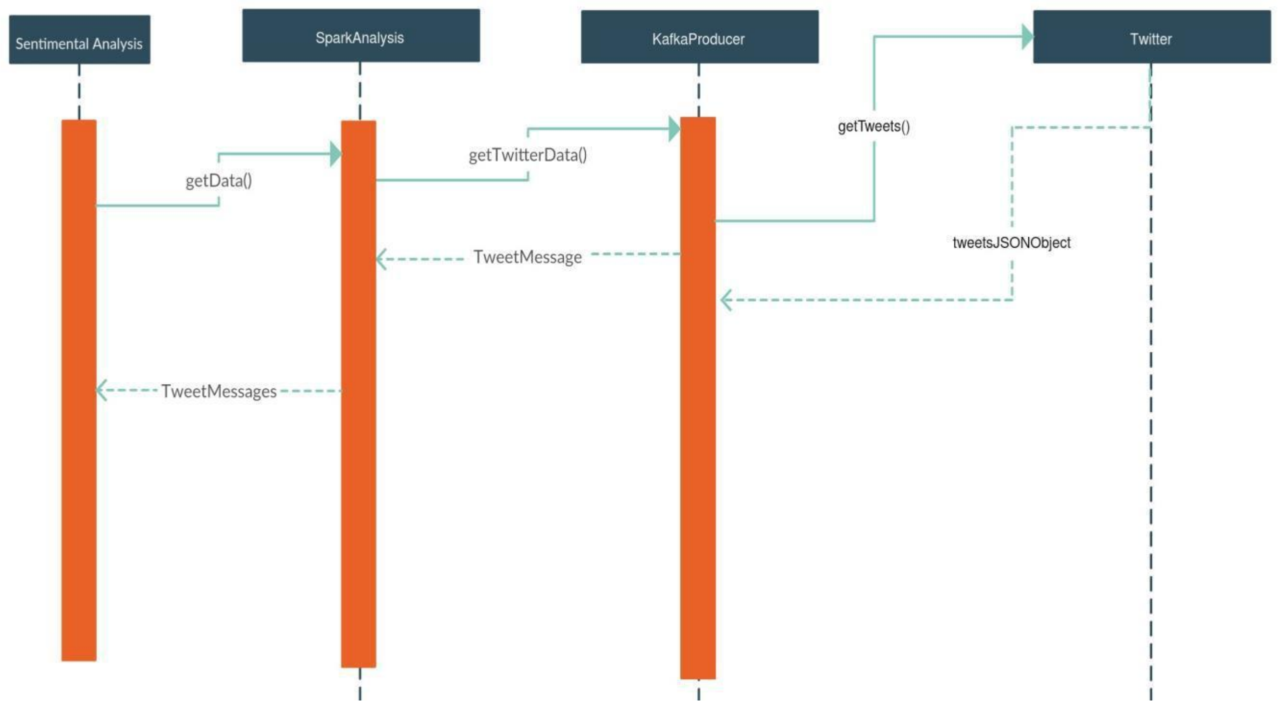
KafkaProducer	Class State • Topic: Stores topic name. • Tweet: Stores tweet. • TwitterKey: Stores authentication for twitter app. Class Behaviour • GetTweet(): Method to get tweets from twitter. • PublishDataStream(): Method to publish data.
KafkaConsumer	Class State • Topic: Store topic name. Class Behaviour • GetDataStream(): Get data from kafka server. • StoreDataInCassandra(): Store tweets in Cassandra database.
KafkaTopics	Class State • No States Class Behaviour • RunMultipleProducers(): Run multiple Kafka Producers. • RunMultipleProducers(): Run multiple Kafka Producers.
SparkServer	Class State Does not have any states. As it is a server which only processes incoming data. Class Behaviour • RunSparkServer() : Run Multiple Spark servers.

Sequence Diagrams

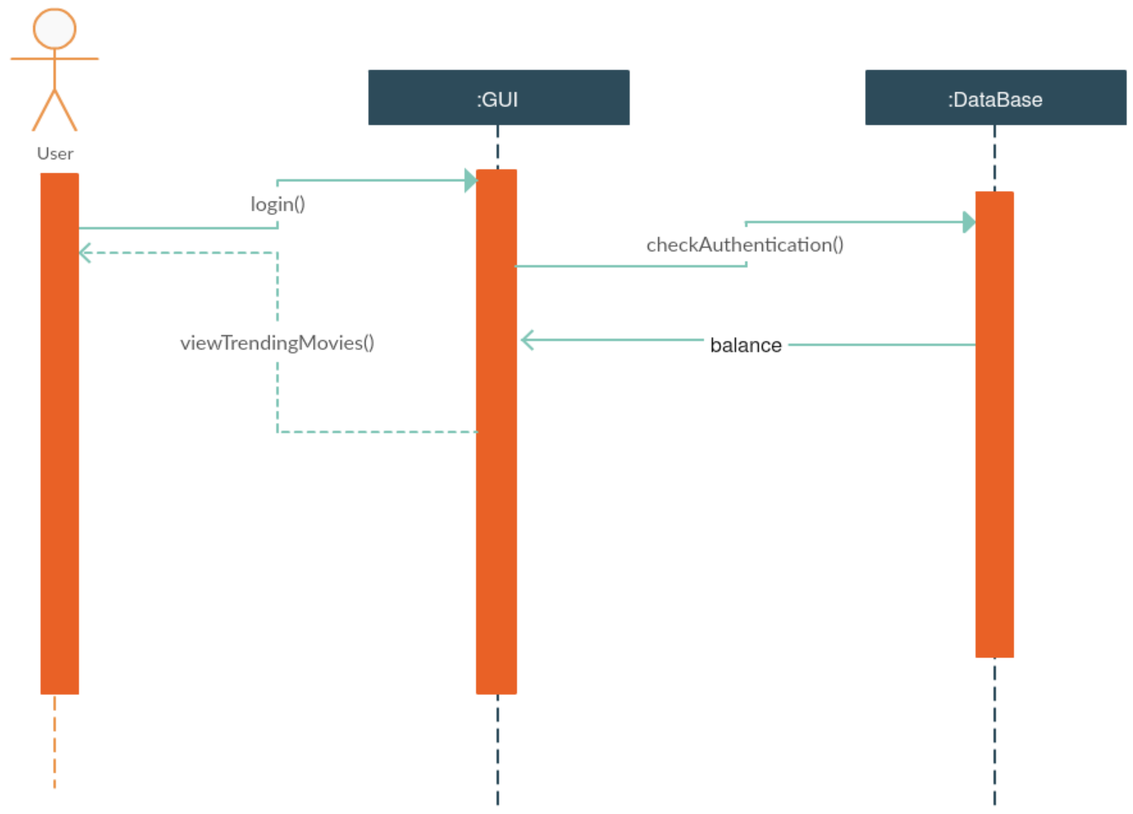
Message Broking



Sentimental Analysis



Suggesting Movies



Design Rationale

Initial design was expected to contain only MovieShow class. But later it was thought that for updating reviews this design was difficult. So, we had to include movie class. Initially design was expected contain class for sentimental analysis. But it is finally included in spark analysis class. As sentimental analysis was too small for class. And was almost like one method. At first, the design contained booking tickets, adding shows and choosing seats. But due to decrease in scope on the instructions of the client, those use cases had to be removed. In the initial design, servers were expected to be run sequentially, but then the execution was changed to parallel. Due to this, 2 more classes had to be included, namely KafkaTopics and SparkServer.