* **You use notepad. You can copy text from context menu, main menu, keyboard shortcuts. Can you think which design pattern will you use? Can you design a notepad application? Identify important classes.**
* **Design anti money laundering system.**
  + Configuration of rules
  + Example rule: One person transferring money to several accounts. And those account holders are withdrawing the cash ATM / Bank
  + Execution of these rules at some frequency.
  + Alerts: If any rule satisfies.
  + User of this portal. Some user will have view access. Some users may have access to edit the rules.
  + Transaction data on which these rules will be executed.

Solution

UserService - user login and registration

AlertService -- SMS, email

RuleManagem -- for configuring the rules to detect anomalies in transaction

TransactionService -- will use transaction data to apply the rules

Kafka

Mobile App UserService Kafka

Api Gateway Alert Service

Web App RuleManagementService

TransactionService -- ElasticSearch database- faster search

Transaction

{

"id":

"amount":

"time":

"fromAccount":

"toAccount":

}

* **https://leetcode.com/discuss/interview-question/object-oriented-design/259629/Design-OO-food-delivery-system**
* **https://github.com/jyotiprasadpal/useful-resources/blob/master/Microservices/Assignment%20-%20Lead%20engineer%20\_%20Principal%20Engineer%20role.pdf**
* **Design a system with Order, Product and Payment.**
* **There is a board, two players and holes on the board. The player who puts all four coins on same line will horizontally, vertically or diagonally wins the game. Create a system design for this board game.**
* **Create system design for workflow management system. Workflow is set of tasks to be performed like task1, task2, task3. User should be able to login, create, execute and monitor workflows. Identify the micro services and interaction.**

LoginService --> register and login

WorkflowService -> create, execute workflows

NotificationService -->

* **There are 3 different types of files - json, xml and csv. The incoming data is same while the format is different. The files are continuously written to respective folders. Create a system design to read files and process them.**
* **Design microservice based architecture for stock portfolio app? UI, Microservices, DB**

\*\*--UI---------\*\*

Login component - Authentication service - we can use REST endpoint for user regsitration and login from Authentication service

CustomerDetails component - will show details of user and allow user to edit certain fields like mobile, email, etc. - we can use REST endpoint from customer service

Stock details component: will show stock name, id, price - can get live tick data from stock service endpoint using websocket connection. We need to open

websocket connection and then susbscribe on topic. We will use websocket endpoint for this.

Alert component: will display alert message. This will get datat from server using websocket from alert service.

We need push mechanism like websocket and not http based REST API call which is pull mechanism that will increase network traffic.

Buy/Sell stock component - We will use the trade service to buy/sell stocks

Portfolio component - will display current portfolio of stocks like current profit/loss,

\*\*--Services------------- decompose services by business capability or subdomain\*\*

Authentication/login Service - will expose endpoints to register new user, login user to authenticate - will expose REST endpoint

Customer service - will provide details about customer information - will expose REST endpoint

Stock service - will provide dtails about stock name, id, stock price, etc. -- will use websocket endpoint

Notification/Alert Service - send alerts to UI whenever trade is executed. -- will expose websocket endpoint

Trade Service - will help to buy/sell stocks. stockid, Qty, price, buy/sell indication -- will expose REST endpoint

whenever there is trade, we need to update the portfolio service and need to communicate to portfolio service, and also to notification service.

\*\*--Database------------

we can use kafka k table or event store or some nosql or rdbms solution. \*\*

\*\*Other design considerations\*\*

Logging: we can use centralized logging solutions like ELK stack

Security: we will use token based authentication to secure communication between services.

Deployment: we will use API gateway to talk to the services. We won't talk to individual services directly. We can use containeize our application and deploy in kubernetes cluster.

Interservice communication - we can use message broker like Kafka or RabbitMQ.