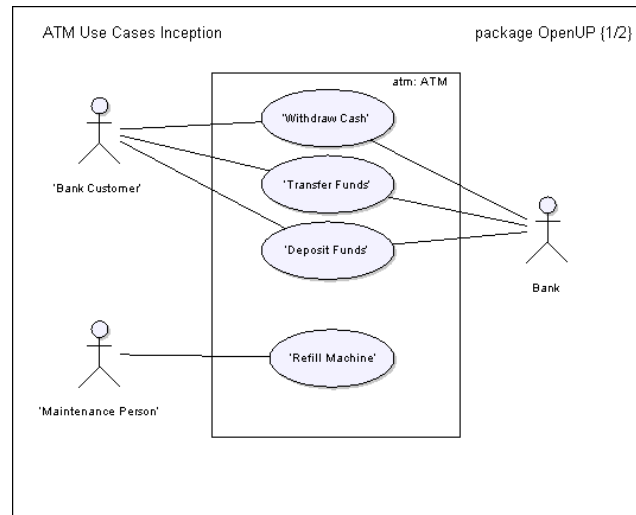


Typical exam problems structure: you are given some text describing a business case and you are asked to do the following:

1. Identifying actors and use cases; including detailed use case description and use case diagram.

Example:



Use Case Name:	Withdraw Cash
Actor(s):	Customer (primary), Banking System (secondary)
Description:	Allows any bank customer to withdraw cash from their bank account.
Pre-Condition:	The bank customer has a card to insert into the ATM The ATM is online properly
Post-Condition(s):	The bank customer has received their cash (and optionally a receipt) The bank has debited the customer's bank account and recorded details of the transaction
Basic Path:	<ol style="list-style-type: none"> 1. The customer enters their card into the ATM 2. The ATM verifies that the card is a valid bank card 3. The ATM requests a PIN code 4. The customer enters their PIN code 5. The ATM validates the bank card against the PIN code 6. The ATM presents service options including "Withdraw" 7. The customer chooses "Withdraw" 8. The ATM presents options for amounts 9. The customer selects an amount or enters an amount 10. The ATM verifies that it has enough cash in its hopper 11. The ATM verifies that the customer is below withdraw limits 12. The ATM verifies sufficient funds in the customer's bank account 13. The ATM debits the customer's bank account 14. The ATM returns the customer's bank card 15. The customer takes their bank card 16. The ATM issues the customer's cash 17. The customer takes their cash
Alternative Paths:	<ol style="list-style-type: none"> 2a. Invalid card 2b. Card upside down 5a. Stolen card 5b. PIN invalid

2. Picking an architectural pattern from the course (except Layers), explain why you picked it and draw a block diagram for it in the context of the business case.

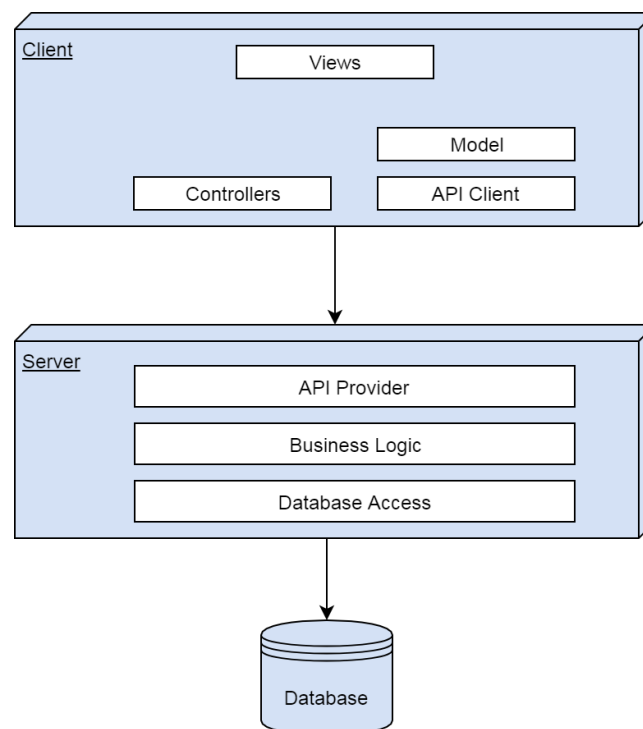
I would recommend focusing on thinking about the following:

- Client – server
- MVC (or variants)
- Micro-services

And as a last resort think of the following:

- Broker
- Pipes and filters
- Plugin (“microkernel”)

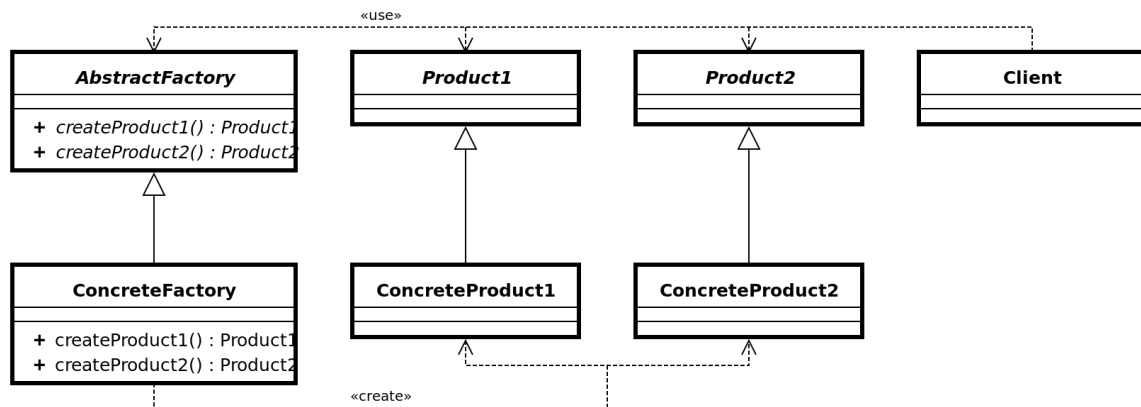
Example diagram:



3. Pick one or two design patterns (not singleton), apply them in the business case, explain why you chose them and draw the class diagram highlighting the design patterns.

I would recommend focusing on thinking about the following GoF patterns:

- Abstract factory,
- Factory method,
- Observer,
- Command,
- Composite,
- Builder,
- Strategy.



4. Draw the database diagram – it is very important to specify: the primary keys, the foreign keys, the more important attributes, the cardinality between tables.
5. Draw a sequence diagram – it is important to have the sequence diagram start from the user interaction until you get to the database layer.
6. Writing code on paper... Usually for the design patterns or the data access layer.

Example exam (2018):

Banca ClujBank doreste sa lanseze propriul site de internet banking si lanseaza o competitie pentru cel mai bun design propus.

Scenariile pe care ClujBank le doreste implementate pe site sunt urmatoarele:

- clientul se poate inregistra pe site avand un username si parola
- clientul poate edita informatiile personale (adresa de resedinta, email, telefon)
- clientul poate vedea conturile deschise la banca, poate sa vada tranzactiile intre anumite perioade, poate sa transfere bani dintr-un cont in altul (inclusiv intre conturile lui), poate sa isi deschida un nou cont
- clientul poate sa aiba conturi in EURO, DOLARI, RON si poate sa faca operatii de schimb valutar intre conturile sale
- clientul poate primi un premiu din partea bancii daca este inscris la tombola organizata de banca si va fi notificat printr-un email
- banca ofera lunar premii pentru clientii sai astfel incat un client se poate inregistra din contul lui sa fie inclus la tombole, iar clientii castigatori vor fi notificati printr-un email ca au castigat un premiu

1. Identificati actorii si realizati diagramele cazurilor de utilizare pentru actorii identificati. Descrieti pasii unui caz de utilizare.
2. Prezentați un pattern arhitectural (exceptand Layers) pentru implementarea sistemului si argumentati alegerea facuta.
3. Descrieti 2 design patterns (excluzand Singleton) folosite in implementarea sistemului si realizați diagrama de clasa strict pentru design patterns alese. Argumentati alegerea facuta.
4. Realizati diagrama bazei de date. Pentru orice informatii lipsa puteti sa va folositi imaginatia.
5. Realizati diagrama de secvente pentru cazul de utilizare "Transfer de bani între 2 conturi"