

Q1. Draw a Use Case diagram for Airport Check-In and Security Screening

Business actors are Passenger, Tour Guide, Minor (Child), Passenger with Special Needs (e.g. with disabilities), all playing **external roles** in relation to airport business.

Business use cases are Individual Check-In, Group Check-In (for groups of tourists), Security Screening, etc. - representing business functions or processes taking place in airport and serving the needs of passengers.

Business use cases Baggage Check-in and Baggage Handling extend Check-In use cases, because passenger might have no luggage, so baggage check-in and handling are optional.

Q2. Draw a Use Case diagram for Restaurant

Customer wants to Have Meal, Candidate - to Apply for Job, and Contractor - to fix some appliances. Note, that we don't have such actors as Chef or Waiter. They are not external roles but part of the business we model - the Restaurant, thus - they are not actors. In terms of RUP Chef and Waiter are business workers.

Customer is now connected to **Serve Meal** use case, **Supplier** - to **Purchase Supplies**. We have now new actor **Potential Customer** participating in **Advertise** use case by reading ads and getting some information about restaurant. At the same time, **Contractor** actor is gone because **Repair Appliances** is not a service usually provided by restaurants.

Q3. Draw a Use Case diagram for Online Shopping - Credit Cards Processing

Credit Card Processing System (aka Credit Card Payment Gateway) is a **subject**, i.e. system under design or consideration. Primary **actor** for the system is a **Merchant's Credit Card Processing System**. The merchant submits some credit card transaction request to the credit card payment gateway on behalf of a customer. Bank which issued customer's credit card is actor which could approve or reject the transaction. If transaction is approved, funds will be transferred to merchant's bank account.

Authorize and Capture use case is the most common type of credit card transaction. The requested amount of money should be first authorized by **Customer's Credit Card Bank**, and if approved, is further submitted for settlement. During the settlement funds approved for the credit card transaction are deposited into the **Merchant's Bank** account.

In some cases, only **authorization** is requested and the transaction will not be sent for settlement. In this case, usually if no further action is taken within some number of days, the authorization expires. Merchants can submit this request if they want to verify the availability of funds on the customer's credit card, if item is not currently in stock, or if merchant wants to review orders before shipping.

Capture (request to capture funds that were previously authorized) use case describes several scenarios when merchant needs to complete some previously authorized transaction - either submitted through the payment gateway or requested without using the system, e.g. using voice authorization.

Credit use case describes situations when customer should receive a refund for a transaction that was both successfully processed and settled through the system or for some transaction that was not originally submitted through the payment gateway.

Void use case describes cases when it is needed to cancel one or several related transactions that were not yet settled. If possible, the transactions will not be sent for settlement. If the Void transaction fails, the original transaction is likely already settled.

Verify use case describes zero or small amount verification transactions which could also include verification of some client's data such as address.

Q4. Draw a use-case diagram using StarUML for the scenario given below.

Web Customer actor uses some web site to make purchases online. Top level **use cases** are **View Items**, **Make Purchase** and **Client Register**. View Items use case could be used by customer as top level use case if customer only wants to find and see some products. This use case could also be used as a part of Make Purchase use case. Client Register use case allows customer to register on the web site, for example to get some coupons or be invited to private sales. Note that **Checkout** use case is **included use case** not available by itself - checkout is part of making purchase. Except for the **Web Customer** actor there are several other actors which will be described below with detailed use cases. **View Items** use case is **extended** by several optional use cases - customer may search for items, browse catalog, view items recommended for him/her, add items to shopping cart or wish list. All these use cases are extending use cases because they provide some optional functions allowing customer to find item. **Customer Authentication** use case is **included** in **View Recommended Items** and **Add to Wish List** because both require customer to be authenticated. At the same time, item could be added to the shopping cart without user authentication. **Checkout** use case includes several required uses cases. Web customer should be authenticated. It could be done through user login page, user authentication cookie ("Remember me") or Single Sign-On (SSO). Web site authentication service is used in all these use cases, while SSO also requires participation of external identity provider. **Checkout** use case also includes **Payment** use case which could be done either by using credit card and external credit payment service or with PayPal.

Q5. Draw a use-case diagram using StarUML for the retail Point-of-sale system depicted below.

A retail POS system typically includes a computer, monitor, keyboard, barcode scanners, weight scale, receipt printer, credit card processing system, etc. and POS terminal software.

Checkout **use case** involves Customer, Clerk and Credit Payment Service **actors** and **includes** scanning items, calculating total and taxes, payment use cases.

Checkout use case requires Customer actor, hence the 1 multiplicity of Customer. Clerk can only participate in a single Checkout use case. Credit Payment Service can participate with many Checkout use cases at the same time. Checkout use case may not need Credit Payment Service (for example, if payment is in cash), thus the 0..1 multiplicity.

Checkout use case is an example of a large and complex use case split into several use cases each describing some logical unit of behavior. Note, that including use case becomes incomplete by itself and requires the included use cases to be complete. Payment use case is represented using **generalization** relationship. It means that only one specific type of payment is accepted - either by cash, or by credit, debit, or with check. An alternative to such representation could be to use **include** relationship so that not just single but several forms of payment could be accepted from the same client during checkout.

Q6. Draw a Use Case diagram for **Bank ATM**

An automated teller machine (**ATM**) or the automatic banking machine (**ABM**) is a banking subsystem (**subject**) that provides bank customers with access to financial transactions in a public space without the need for a cashier, clerk, or bank teller.

Customer (**actor**) uses bank ATM to Check Balances of his/her bank accounts, Deposit Funds, Withdraw Cash and/or Transfer Funds (**use cases**). ATM Technician provides Maintenance and Repairs. All these use cases also involve Bank actor whether it is related to customer transactions or to the ATM servicing.

On most bank ATMs, the customer is authenticated by inserting a plastic ATM card and entering a personal identification number (PIN). Customer Authentication use case is required for every ATM transaction so we show it as **include** relationship. Including this use case as well as transaction **generalizations** make the ATM Transaction an **abstract use case**.

Customer may need some help from the ATM. ATM Transaction use case is **extended** via **extension point** called menu by the ATM Help use case whenever ATM Transaction is at the location specified by the menu and the bank customer requests help, e.g. by selecting Help menu item.

ATM Technician maintains or repairs Bank ATM. Maintenance use case includes Replenishing ATM with cash, ink or printer paper, Upgrades of hardware, firmware or software, and remote or on-site Diagnostics. Diagnostics is also **included** in (shared with) Repair use case.

Q7. Draw a Use Case diagram for **Online Library Public Access Catalog**

An **Online Public Access Catalog (OPAC)** is an e-Library website which is part of **Integrated Library System (ILS)**, also known as a **Library Management System (LMS)**, and managed by a library or group of libraries.

Patrons of the library can search library catalog online to locate various resources - books, periodicals, audio and visual materials, or other items under control of the library. Patrons may reserve or renew item, provide feedback, and manage their account.

Q8. Draw a USE-CASE DIAGRAM “**Hospital Reception Subsystem**”

“Hospital Reception Subsystem” supports some of the many job duties of hospital receptionist. Receptionist schedule patient’s appointments with the doctor and also schedule patient hospital admission. If doctor is available and admission to the hospital is possible then receptionist can extend the service to patient registration by collecting the patient information on patient arrival or over the phone. Patient registration is an integral part of patient Hospital Administration use case. Hospital administration use case is further generalized into outpatient hospital admission and inpatient hospital administration. Note that for the patient that will stay in the hospital, he or she should have a bed allotted in a ward. Receptionists might also receive patient’s payments, record them in a database and provide receipts, file insurance claims and medical reports.

Q9. DRAW A USE-CASE DIAGRAM for “MUSIC PORTAL SYSTEM” DEPICTED BELOW.

The following narration describes some of the use cases for “**Music Portal System**”. This system has **web user** as its main actor. The web user can perform first level uses cases namely **SearchAlbum**, **login**, **logout** and **ViewAccount**. **Registration** use case extends login i.e. if the user doesn't have a login and wishes to create a new one, he or she can register to get a login. Moreover the **ViewAlbumDetail** use case is extending the **SearchAlbum** use case. Further the **ViewAlbumDetail** use case is extended by two more services viz. **DownloadAlbum** and **BuyAlbum**. Note that to download or to buy an album the user must be a registered member. Moreover the **buyalbum** and **ViewAccount** is further extended with **Recharge** use case.

Q10. Draw a class diagram for the scenario given below.

This is an example that models University Courses. Assume three classes' such as course, lecturer, student and an interface person. Each course objects maintains a list of student on that course and lecturer who has been assigned to teach that course. The course object has behavior that allows adding and removing student to and from course, assigning the teacher and getting a list of currently assigned student and currently assigned teacher. A teacher may teach several courses but a course only has a single teacher .A lecturer object maintains a list of courses that it teaches, course is attended by 0 or more student and student may attend multiple courses. A person interface will have `getName()` and `getEmailAddress ()` methods both lecturer and student are shown to be the type of person.

Q11. Draw a class diagram for the scenario given below.

This scenario is from system that models companies for a payroll or reporting system. Company object has properties such as name and `employees_list` and `getName` and `getEmployees` as its behavior. Employee object includes employee no, name, salary and manager as its properties `getName ()`, `getEmployeeNo ()` , `getSalary()` and `getManager()` as its methods. `getManager()` accepts object of manager. Company may have one or more employees. A manager object keeps manages as list property and `add TeamMember(employee_list)` and `getTeamMember()` as its behaviors. One or more employee can be managed by manager objects. Some employees are contractual employees who are within a lieu of a contractor object. A contractor object may have `length_ of _contract` as its property and `getLength()` as its behavior.

Q12. Create a class diagram(Use Star UML) for “library management” using the classes with their attributes and operation given below. Also set the appropriate relationship between the classes using the relationship tools from the toolbox following the overview of the system given below. Overview of the system:-

- a) It has a class “Book”. Book has authors so it has an “Author” class.
- b) In order to collect book information it has “BookItem” class which uses some of the properties from book class.
- c) It needs an account for reserving book by the user so it has an “Account class.”
- d) In account class there is an attribute named state which uses an enumeration named “AccountState”.
- e) It also has a class “Library” to manage the account, user and the books.
- f) It has a user class to manage the user detail that has an account in the library and he can borrow and return books to library.
- g) The system also has an interface “Search” where the user searches the book he needed from the “Catalog” class.

Q13. Draw a class diagram using StarUML for the scenario given below.

This scenario shows an inheritance hierarchy of a series of classes and their subclasses. It's for an imaginary application that must model different kinds of vehicles such as bicycles, motor bike and cars. All Vehicles have some common attributes (speed and color) and common behavior (turnLeft, turnRight). Bicycle and MotorVehicle are both kinds of Vehicle and are therefore shown to inherit from Vehicle. To put another way, Vehicle is the superclass of both Bicycle and MotorVehicle. In our model MotorVehicles have engines and license plates. Attributes have been added accordingly, along with some behavior that allows us to examine those attributes. MotorVehicles is the base class of both MotorBike and Car; therefore these classes not only inherit the speed and color properties from Vehicle, but also the additional attributes and behavior from MotorVehicle. Both MotorBike and Car have additional attributes and behavior which are specific to those kinds of object.

Q14. DRAW A CLASS DIAGRAM FOR "ORDER MANAGEMENT"

The Customer object has properties such as CustomerId, CustomerName, Address and Phone and methods such as AddCustomer(), DeleteCustomer() and EditCustomer(). Order object includes OrderId, CustomerId, CustomerName, ProductId, Amount and OrderDate as its property and CreateOrder() and EditOrder(OrderId) as its behavior. A customer can place one or many orders. Further there are SpecialOrder object and NormalOrder object which have same methods CreateOrder(), confirm(), close(), dispatch() whereas the SpecialOrder object also has one property named SpecialDiscount. SpecialOrder and NormalOrder objects are both kinds of order and are therefore shown to inherit from order entity. Moreover the system also has Product entity having attributes such as ProductId, ProductPrice, ProductType and methods such as AddProduct(), ModifyProduct() and SelectProduct(ProductId). Stock object has properties like ProductId, Quality and ShopNo and behavior such as addStock(), ModifyStock(ProductId) and selectStockItem(ProductId). Note that specialOrder and NormalOrder has 1 or more product whereas stock has many products.

Q15. DRAW A CLASS DIAGRAM FOR THE SCENARIO GIVEN BELOW:

This is an example that models "Hospital Management". The ward object of this system has attributes such as name, patient-gender and capacity. Note that patient-gender is a gender type which is an enumeration containing enums male and female. The system also has Patient entity with attributes such as patient_id, admitted, sickness _history, prescriptions, special_reqs and allergies and gender which is again a gender enumeration type. And operations such as getPatient() and deletePatient(Patient_id). Ward is a division of a hospital object having attributes such as name address and phone number. In hospital there are number of wards each of which may be empty or have one or more patient. Each ward has unique name. This ward is shared by patients who need a similar kind of care. Each patient is on a single ward. The system also has Doctor entity which is further classified into Consultant Doctor and Junior Doctor. The doctors in the hospital are organized into Teams entity with attribute team_name. Each team can have two or more doctors. Each patient is under the care of a single team of doctors. A patient may be treated by any number of doctors but all the doctors must belong to same team that cares for the patient. Note that team is own by the hospital.

Q16. DRAW AN ACTIVITY DIAGRAM FOR “BANK ATM MACHINE FOR WITHDRAWING CASH”.

An automated teller machine (ATM) or the automatic banking machine (ABM) is a banking subsystem that provides bank **customers** with access to financial transactions in a public space without the need for a cashier, clerk or bank teller.

Customer uses bank ATM to **check** balances of his/her bank accounts, deposit **funds**, **withdraw** cash and/or transfer funds which are the generalization alternative of ATM transaction use case.

FOR WITHDRAWING CASH

On most bank ATMs, the customer is authenticated by inserting a plastic ATM card and entering a personal identification number (PIN). Bank will then authenticate the customer's pin number. Only authenticated customer can request the system for withdrawing money while the unauthenticated customer will get back his ATM card as the system will reject the card.

Then the system will request the authenticated customer to enter the amount to be withdrawn. The bank will check the balance amount of the customer if it is sufficient bank will provide the requested amount to the customer and debit the respective amount from the balance. The customer will collect or take the amount from the slot. In case of insufficient amount the system will show the balance and reject the card. At the end of all the process the customer will take back his ATM card.

Q17. DRAW AN ACTIVITY DIAGRAM FROM THE NARRATIVE TEXT ON “ONLINE PAPER SUBMISSION SYSTEM”.

The author completes an online form that requests the user to input author name, Correspondence address, email and, title of paper. The system validates this data and, if correct, asks the author to submit the paper. The author then browses to find the correct paper on their system and submits it. Once received and stored, the system returns to the author a reference number for the paper. Authors may submit as many papers as they like to be considered for acceptance to the conference up until the deadline date for submissions. Papers are allocated to referees for assessment. They review each paper and submit to the system their decision. Once the program organizer has agreed the decisions authors are informed by email. Accepted papers are then scheduled to be delivered at a conference. This involves allocating a date, time and place for the presentation of the paper.

Q18. DERIVE AN ACTIVITY DIAGRAM FROM THE CASE GIVEN BELOW ON “ORDER PROCESSING SUBSYSTEM”

Web **Customer** uses some web site to make purchases online. Where customer can search item, View item, **add item to the cart, place order and make payment**. For placing an order the customer first searches the required items from the system. As and when the customer finds the item available in the system he starts adding item to the chart. The System provides facility to the Customer to add any number of items to the chart. Customer can also view his shopping chart containing items. Once the customer finishes his shopping he can place the order by requesting system to confirm the order. The system will then check whether the order is normal order or any special order and according to that the system will generate the bill and request for payment.

After getting the bill the customer can make payment. The bank will validate the credit card number. If the credit card number is valid the system will confirm the order. Otherwise the process will get terminated.