Data Flow Diagram Rules:

1. Create a list of activities
2. Construct Context Level "Level-0" DFD (identifies external entities and processes)
3. Construct Level 1 DFD (identify manageable sub process)
4. Construct Level 2 DFD (identify actual data flows and data stores)

DFD Naming Rules:

- External Entity → Noun
- Data Flow → Names of data
- Process → verb phrase
  - ✓ a system name
  - ✓ a subsystem name
- Data Store → Noun
- Symbols → See in the slides.

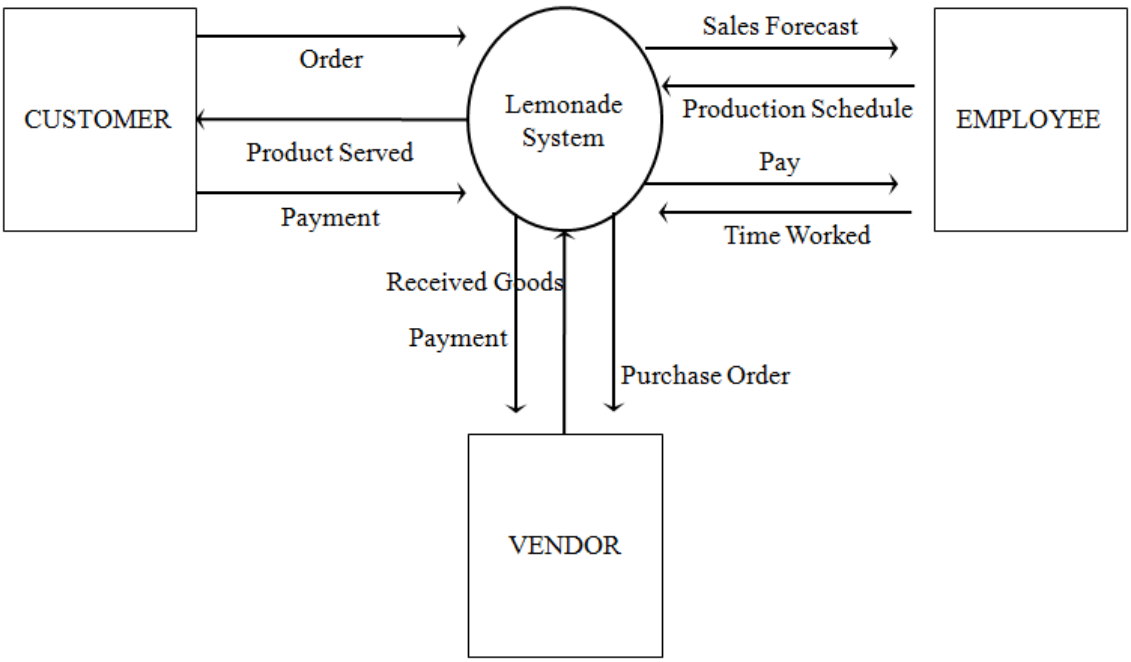
Example1:

Suppose your organization is going to develop a Lemonade Stand system. The system should perform operations such as sale, production, procurement and payroll. System should let the customer to make request for their order and pay for services. When customer requests for an order, system records order of the system, saves it into record and receives payment from customer and save it into payment and serve the order. Employee received the sales forecast from the system and produce and store product if necessary. Details of product should be stored on Inventory database. Also system makes payment to employee according to their working time/hour by checking on time cards database. Vendor should receive order & payment from the system and provides goods. The payment to the vendor should be stored on vendor database and details of the receive goods from vendor should be stored on received items.

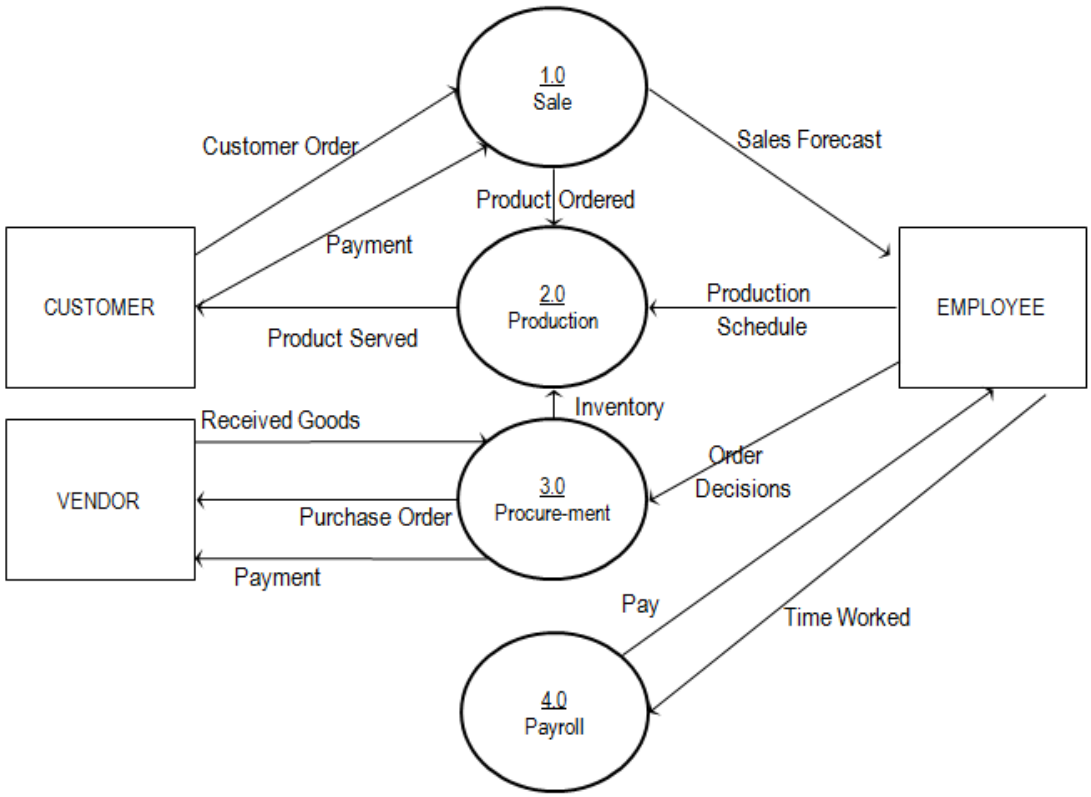
Step1: Create a list of activities

- ✓ Customer Order
- ✓ Serve Product
- ✓ Collect Payment
- ✓ Produce Product
- ✓ Store Product
- ✓ Order Raw Materials
- ✓ Pay for Raw Materials
- ✓ Pay for Labor

Step2: Construct Context Level "Level-0" DFD

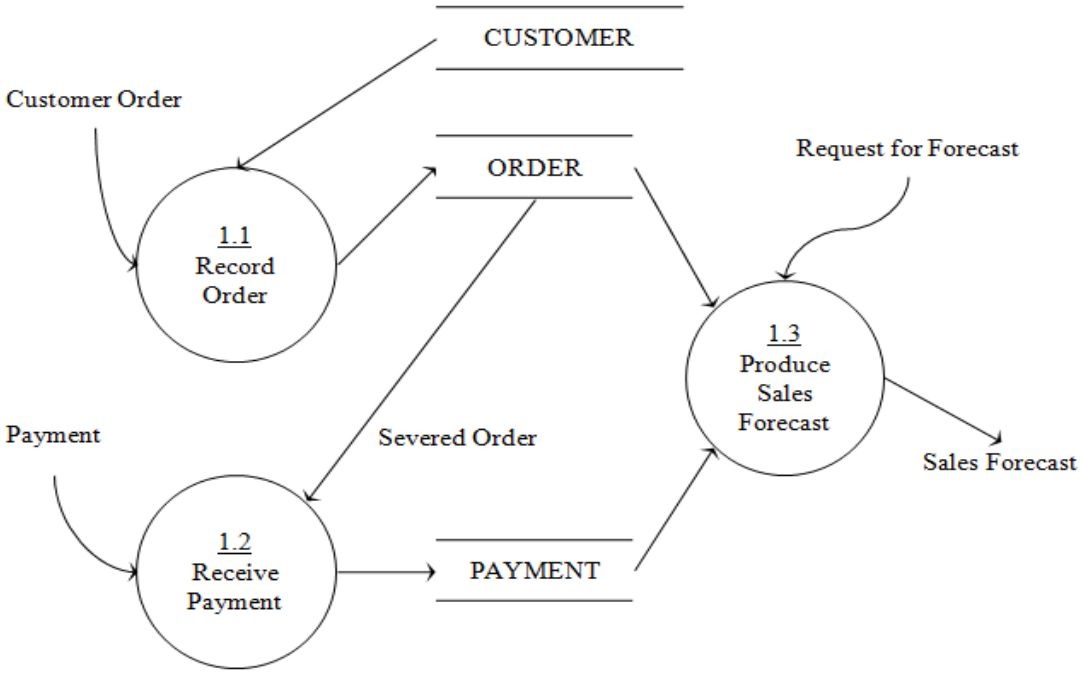


Step2: Construct Context Level-1 DFD **{** identify manageable sub process **}**

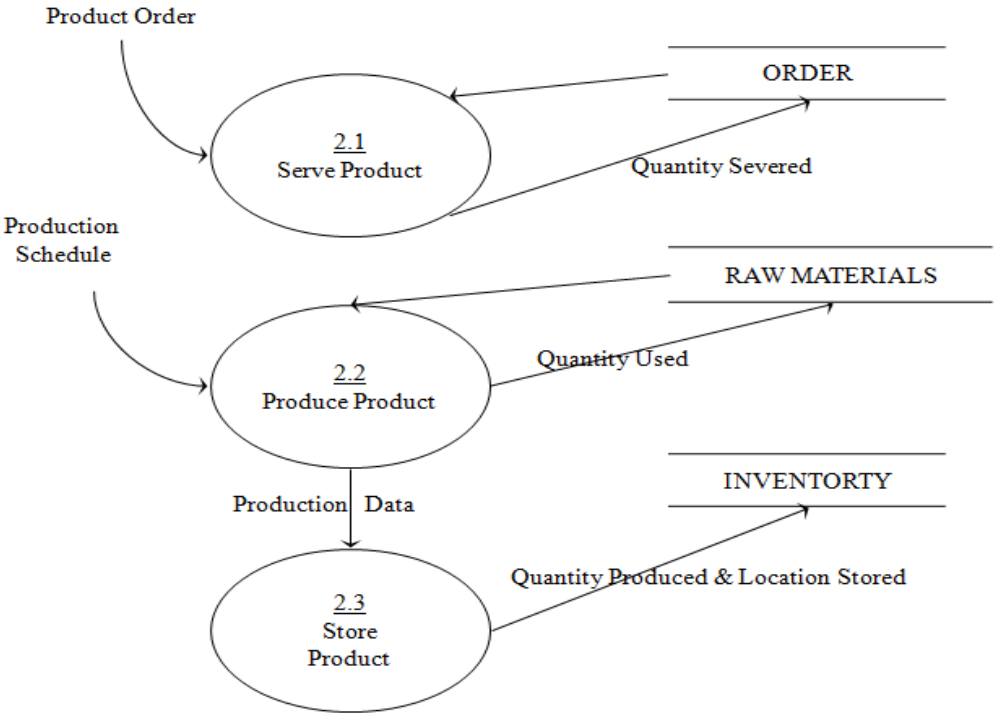


Step3: Construct Context Level-2 DFD {identify actual data flows and data stores}

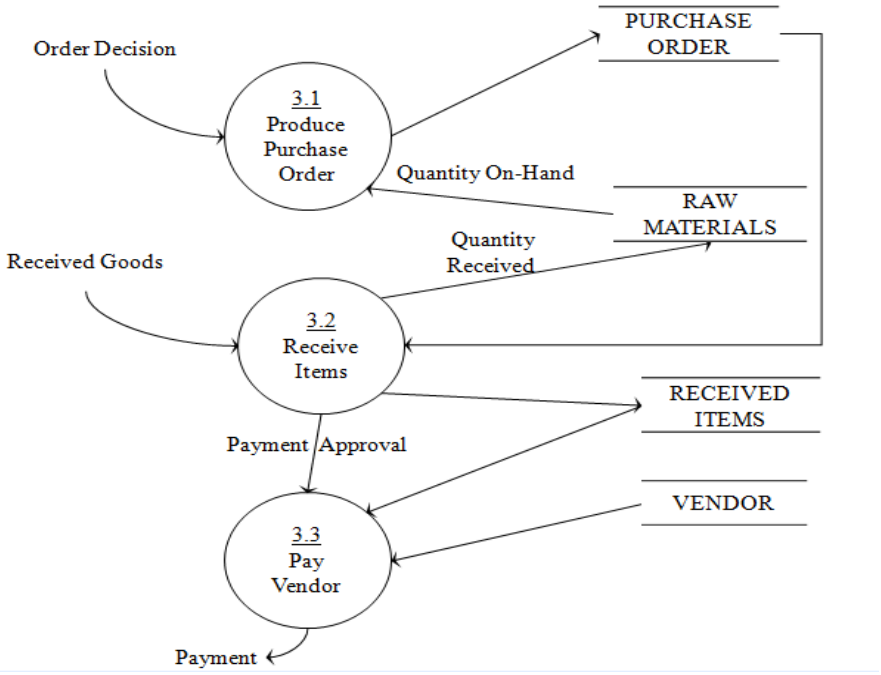
1. Sales



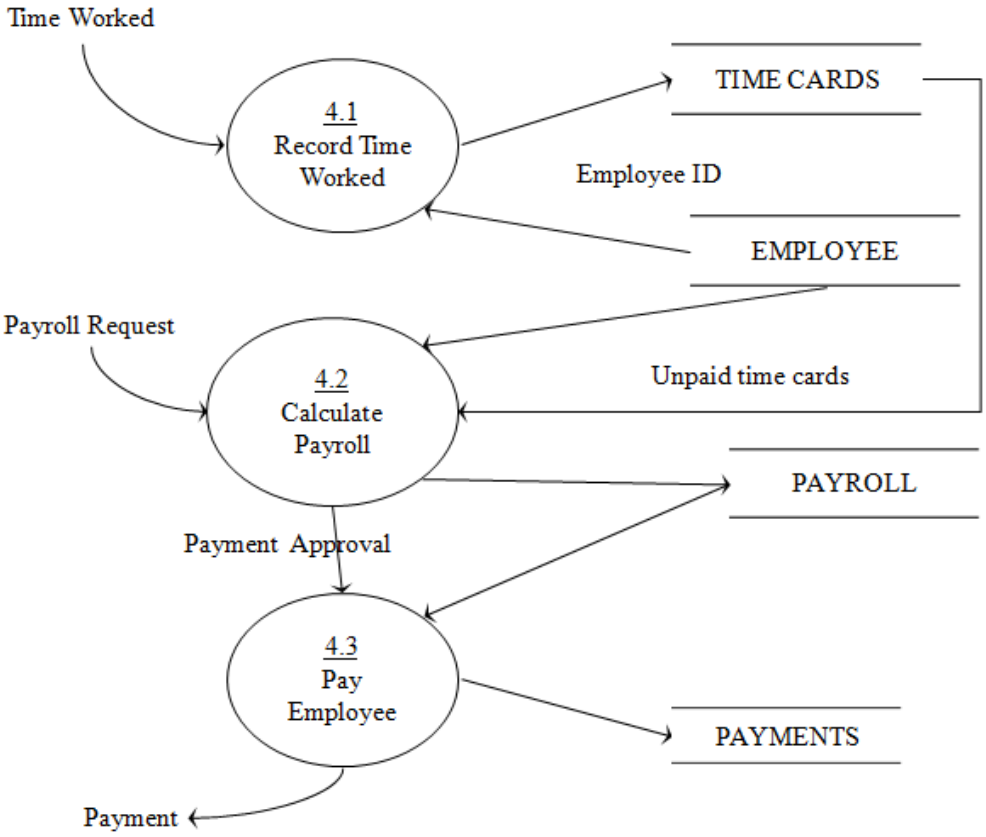
2. Product



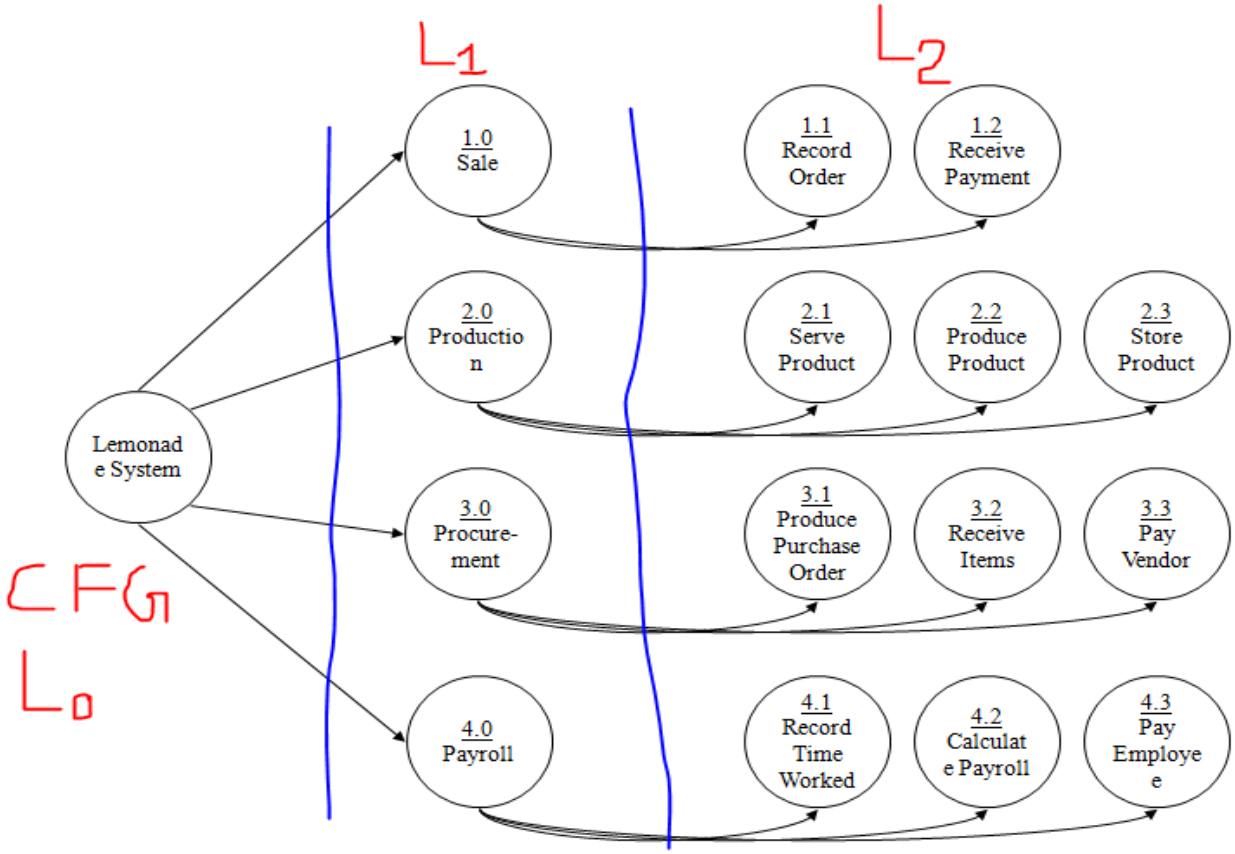
3.


PURCHASE ORDER
Order Decision
3.1 Produce Purchase Order
Quantity On-Hand
RAW MATERIALS
Quantity Received
Received Goods
3.2 Receive Items
RECEIVED ITEMS
Payment Approval
VENDOR
3.3 Pay Vendor
Payment

4.

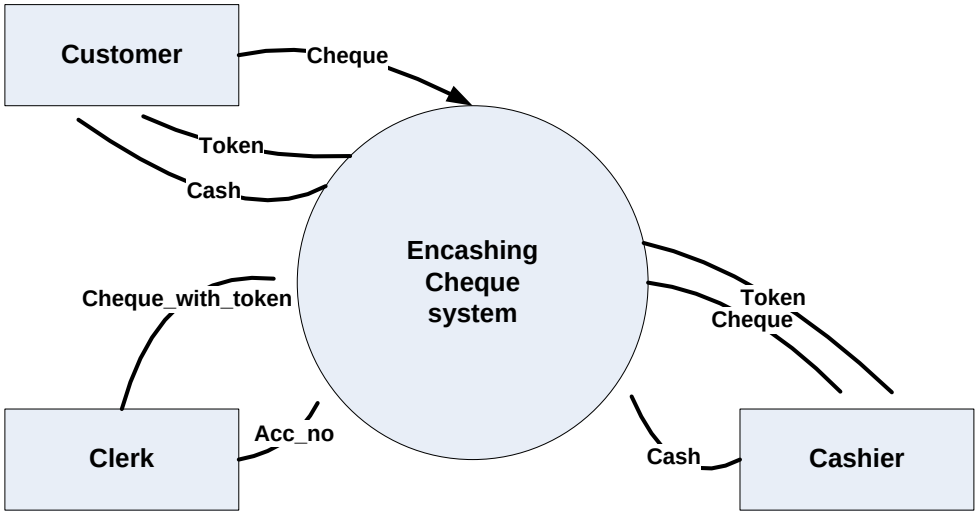

Time Worked
4.1 Record Time Worked
TIME CARDS
Employee ID
EMPLOYEE
Payroll Request
4.2 Calculate Payroll
Unpaid time cards
PAYROLL
Payment Approval
4.3 Pay Employee
PAYMENTS
Payment

Overview

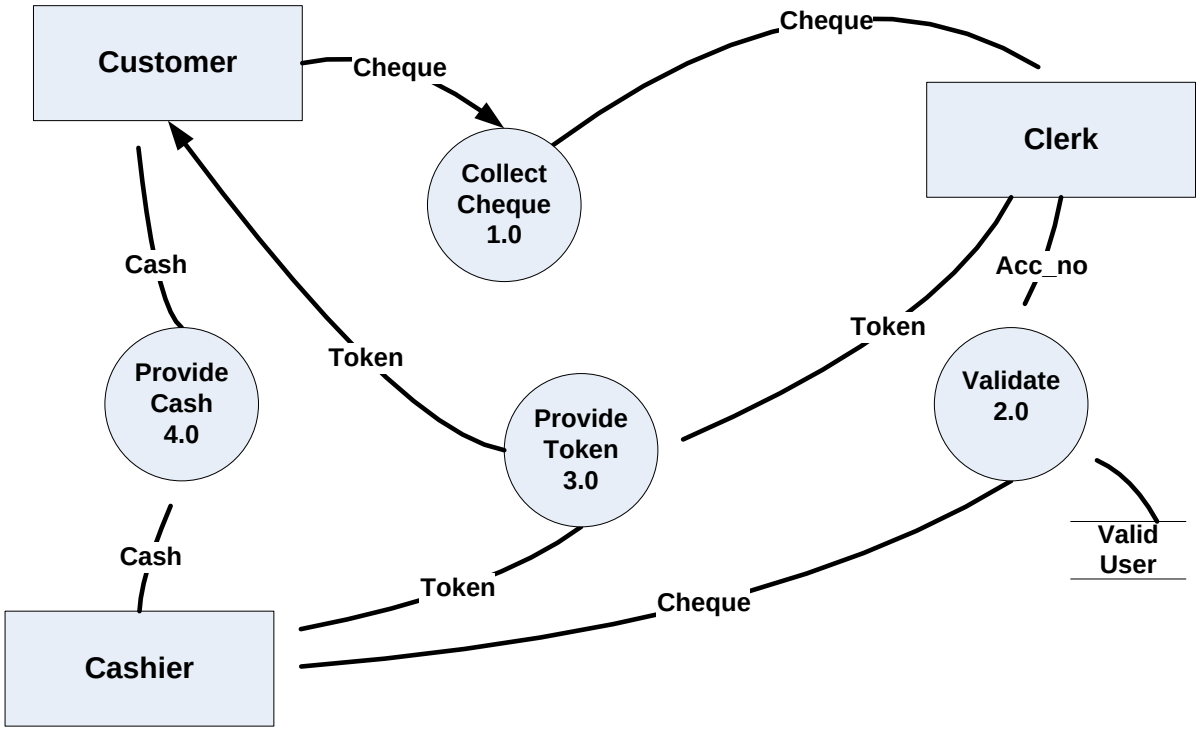


Example2:

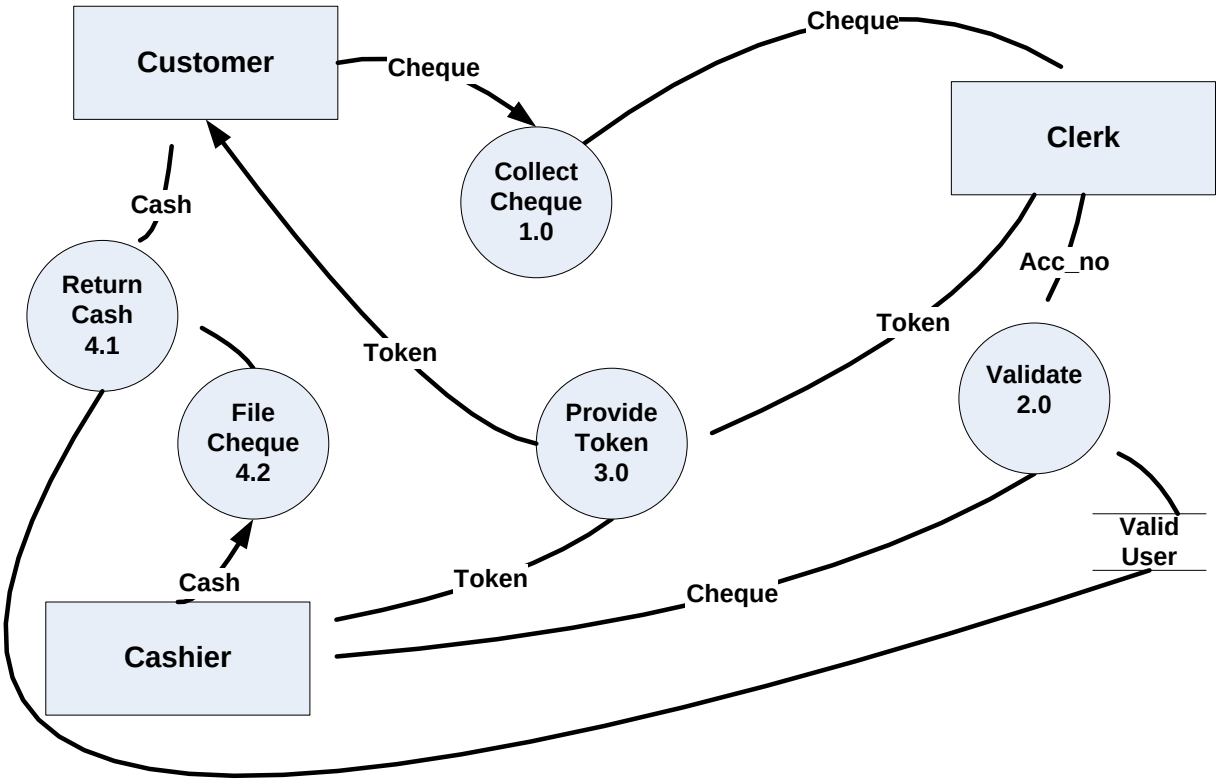
A customer presents a cheque to a clerk. The clerk checks a database containing all account numbers and make sure whether the account number in the cheque is valid, whether adequate balance is there in the account to pay the cheque and whether the signature is authentic. Having done these the clerk gives the customer a token. The clerk also debits the customer account by an amount specified on the cheque. If the cash cannot be paid due to an error on the cheque, the cheque is returned. The token number is retumed on the top of the cheque and it is passed on to the cashier. The cashier calls out returned the token number, takes customer signatures, pays cash, enter cash paid in database and files the cheque.


Customer
Cheque
Token
Cash
Encashing
Cheque
system
Cheque_with_token
Token
Cheque
Clerk
Acc_no
Cash
Cashier

CFG


Customer
Cheque
Cheque
Clerk
Collect
Cheque
1.0
Cash
Acc_no
Token
Token
Provide
Cash
4.0
Validate
2.0
Provide
Token
3.0
Valid
User
Cash
Token
Cheque
Cashier

Level 1


Customer
Cheque
Cheque
Clerk
Collect
Cheque
1.0
Cash
Acc_no
Return
Cash
4.1
Token
Token
Validate
2.0
File
Cheque
4.2
Provide
Token
3.0
Valid
User
Cash
Token
Cheque
Cashier

Level 2

➢ How to list out functional and non-functional requirements

EXAMPLE: HOTEL MANAGEMENT SYSTEM

1. FUNCTIONAL REQUIREMENTS

Functional requirements define the fundamental actions that system must perform. The functional requirements of Hotel management system are divided into three main categories, Reservation/Booking, Food, and Management.

Reservation/Booking

- ✓ *The system shall record reservations.*
- ✓ *The system shall record the customer's first and last name.*
- ✓ *The system shall record the number of occupants.*
- ✓ *The system shall record the room number.*
- ✓ *The system shall display the default room rate.*
- ✓ *The system shall allow the default room rate to be changed.*
- ✓ *The system shall require a comment to be entered, describing the reason for changing the default room rate.*
- ✓ *The system shall record the customer's phone number.*
- ✓ *The system shall display whether or not the room is guaranteed.*
- ✓ *The system shall generate a unique confirmation number for each reservation.*
- ✓ *The system shall automatically cancel non-guaranteed reservations if the customer has not provided their credit card number by 6:00 pm on the check-in date.*
- ✓ *The system shall record the expected check-in date and time.*
- ✓ *The system shall record the expected checkout date and time.*
- ✓ *The system shall check-in customers.*
- ✓ *The system shall allow reservations to be modified without having to reenter all the customer information.*
- ✓ *The system shall checkout customers.*
- ✓ *The system shall display the amount owed by the customer.*
- ✓ *To retrieve customer information the last name or room number shall be used*
- ✓ *The system shall record that the room is empty.*
- ✓ *The system shall record the payment.*
- ✓ *The system shall record the payment type.*
- ✓ *The system shall charge the customer for an extra night if they checkout after 11:00 a.m.*
- ✓ *The system shall mark guaranteed rooms as "must pay" after 6:00 pm on the check-in date.*
- ✓ *The system shall record customer feedback.*

Food

- ✓ *The system shall track all meals purchased in the hotel (restaurant and room service).*
- ✓ *The system shall record payment and payment type for meals.*
- ✓ *The system shall bill the current room if payment is not made at time of service.*
- ✓ *The system shall accept reservations for the restaurant and room service.*

Management

- ✓ *The system shall display the hotel occupancy for a specified period of time (days; including past, present, and future dates).*
- ✓ *The system shall display projected occupancy for a period of time (days).*
- ✓ *The system shall display room revenue for a specified period of time (days).*
- ✓ *The system shall display food revenue for a specified period of time (days).*
- ✓ *The system shall display an exception report, showing where default room and food prices have been overridden.*
- ✓ *The system shall allow for the addition of information, regarding rooms, rates, menu items, prices, and user profiles.*
- ✓ *The system shall allow for the deletion of information, regarding rooms, rates, menu items, prices, and user profiles.*
- ✓ *The system shall allow for the modification of information, regarding rooms, rates, menu items, prices, and user profiles.*
- ✓ *The system shall allow managers to assign user passwords.*

2. NON-FUNCTIONAL REQUIREMENTS

Functional requirements define the needs in terms of performance, logical database requirements, design constraints, standards compliance, reliability, availability, security, maintainability, and portability.

Performance Requirements

- ✓ *The load time for user interface screens shall take no longer than two seconds.*
- ✓ *The log in information shall be verified within five seconds.*
- ✓ *Queries shall return results within five seconds.*

Design Constraints

- ✓ *The System shall be a stand-alone system running in a Windows environment.*
- ✓ *The system shall be developed using Java and an Access or Oracle database.*

Standards Compliance

- ✓ *There shall be consistency in variable names within the system.*
- ✓ *The graphical user interface shall have a consistent look and feel.*

Reliability

- ✓ *Reliability of the software system at time of delivery.*

Availability

- ✓ *The system shall be available during normal hotel operating hours {24 hrs maybe}.*

Security

- ✓ *Customer Service Representatives and Managers will be able to log in to the System.*
- ✓ *CSR will have access to the Reservation/Booking and Food subsystems.*
- ✓ *Managers will have access to the Management subsystem as well as the Reservation/Booking and Food subsystems.*
- ✓ *Access to the various subsystems should be protected by a user log in screen that requires a user name and password.*

Maintainability

- ✓ *The Hotel Management System should be developed in Java.*
- ✓ *The system should be easy to maintain.*

Portability

- ✓ *The Hotel Management System should run in any Microsoft Windows environment that contains Java Run time and the Microsoft Access database.*

Change Management Process

- ✓ *Changes to the document should be made after approval from the project manager and the client approval officer.*

TEST CASE DESIGN EXAMPLE

Scope: Test various field input on the Registration Request Form.
Action: Submit Registration request.
Test Notes and Preconditions: Open the user registration request form from the users tab.
Scenario: Test leaving a required field blank on the new user registration

S. N.	Test scenario	Test cases	Test steps	Expected Results	Actual Results	Pass/ Fail/ Not-executed /Suspended
1	Submit Registration request…	All field filled with valid data	Fill all fields with valid data.	Registration successful		
2	Submit Registration request…	First Name = blank, Other fields valid	Fill all fields with valid data except First Name	Error message should be displayed for invalid First Name.		
3	Submit Registration request…	Last Name = blank, Other fields valid	Fill all fields with valid data except Last Name	Error message should be displayed for invalid Last Name.		
4	Submit Registration request…	Email = blank, Other fields valid	Fill all fields with valid data except email	Error message should be displayed for invalid Email.		
5	Submit Registration request…	Phone number = blank, Other fields valid	Fill all fields with valid data except phone number	Error message should be displayed for invalid Phone number.		
6	Submit Registration request…	Company = blank, Other fields valid	Fill all fields with valid data except company	Error message should be displayed for invalid Last Name.		
7	Submit Registration request…	User ID = blank, Other fields valid	Fill all fields with valid data except user id	Error message should be displayed for invalid User ID.		
8	Submit Registration	Password = blank,	Fill all fields with valid data except	Error message should be		

	request…	Other fields valid	password	displayed for invalid Password.		
9	Submit Registration request…	Confirm Password = blank Other fields valid	Fill all fields with valid data except conform password	Error message should be displayed for invalid confirmation password		
10	Submit Registration request…	All fields blank	Non	Error message should be displayed.		

Scope: Test various field input on the User Login
Action: Verify login
Test Notes & Preconditions: Need a valid registered account for login.
Scenario: Test providing invalid username and password for system login.

S. N.	**Test scenario**	**Test cases**	**Test steps**	**Expected Results**	**Actual Results**	**Pass/ Fail/ Not-executed /Suspended**
1	Verify login	Enter valid username and valid password	Enter username Enter Password Click 'login' button	Login successful		
2	Verify login	Enter valid username and invalid password	Enter username Enter Password Click 'login' button	A message "the username and password you entered don't match" is shown		
3	Verify login	Enter invalid username and valid password	Enter username Enter Password Click 'login' button	A message "the username and password you entered don't match" is shown		
4	Verify login	Enter invalid username and invalid password	Enter username Enter Password Click 'login' button	A message "the username and password you entered don't match" is shown		