KMM COLLEGE OF ARTS & SCIENECE, THRIKAKKARA

KVCOP

REVIEW2

Submitted By

Shabas V S

Yedhu Prasad

Muhsin Ibrahim Submitted To

Sheema Rajan

Date : 09-09-2021

INTRODUCTION

The KVCOP web interface is a futuristic replacement for the existing vehicle penalty management system of the Kerala Motor Vehicle Department. This interface allows users to know the details such as a number of charges and penalties imposed upon a vehicle or an individual license holder. Integrating many existing modules into a single web portal provides ease of use for both administration and civilians. The interface allows guest users to access basic details about a vehicle, while it allows privileged users to access advanced details about a vehicle or an individual license holder and impose penalties on the vehicle or the license holder with visual evidence and details. Some of the main factors that have to be noted using the design of the system are:

Practicability

The system must be capable of being operated over a long period and must have ease of use.

Efficiency

Make better use of available resources. Efficiency involves accuracy, timeliness, and comprehensiveness of system output.

Cost

Aim of minimum cost and better results.

Security

Ensure physical security of data

INPUT DESIGN

Input design is the process of converting user-oriented input to a based format. Inaccurate input data are the most common cause of errors in the data processing. Errors entered by data entry operators can be controlled by input design.

The goal of designing input data is to make data entry as easy, logical, and free from errors. When we approach input data design; we design the data source documents that capture the data and then select the media used to enter them into the computer.

User-friendly screen format can reduce the burden on end-users, who are not highly proficient in computers. An important step in the input design stage is the design of the source document. The source document is the form in which the data can initially capture. The next step is the design of the document layout. The layout organizes the document by placing information, where it will be noticed and establishes the appropriate sequence of items.

OUTPUT DESIGN

Computer output is the most important and direct source of information to the user. Efficient and intelligent output design improves the system’s relationship and helps user decision-making.

In the output design, it is determined how the implementation is to be played for immediate need and also the hardcopy output. A major form of input is a hard copy from the printer. Printouts should be designed around the output requirement of the user. Printers, CRT screen displays are examples for providing computer-based output. The output design associated with the system includes the various reports of the table generations and query executions.

Output design is one of the, most important features of the information system. The logical design of an information system is analogous to an engineering blueprint of an automobile. It shows the major features and how they are related to one another. The outputs, inputs, and databases are designed are in this phase

DATABASE DESIGN

Database design, The most important part of the system design phase. In a database environment, data available are used by several users instead of each program managing its data, authorized users share data across the application with the database software managing the data as an entity.

The primary key is one of the candidate keys that are chosen to be the identifying key for the entire table.

Normalization is a process of organizing the data in the database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

Here are the most commonly used normal forms:

● First normal form(1NF)

● Second normal form(2NF)

● Third normal form(3NF)

● Boyce & Codd normal form (BCNF)

First normal form (1NF)

As per the rule of the first normal form, an attribute (column) of a table cannot hold multiple values.

It should hold only atomic values.

Second normal form (2NF)

A table is said to be in 2NF if both the following conditions hold:

● The table is in 1NF (First normal form)

● No non-prime attribute is dependent on the proper subset of any candidate key of the table.

An attribute that is not part of any candidate key is known as a non-prime attribute.

Third Normal form (3NF)

A table design is said to be in 3NF if both the following conditions hold:

● The table must be in 2NF

● Transitive functional dependency of non-prime attribute on any super key should be removed.

An attribute that is not part of any candidate key is known as a non-prime attribute.

In other words, 3NF can be explained like this: A table is in 3NF if it is in 2NF, and for each

functional dependency X-> Y at least one of the following conditions hold:

● X is a super key of the table

● Y is a prime attribute of the table

An attribute that is a part of one of the candidate keys is known as a prime attribute.

Boyce Codd normal form (BCNF)

It is an advanced version of 3NF that’s why it is also referred to as 3.5NF. BCNF is stricter than

3NF. A table complies with BCNF if it is in 3NF and for every functional dependency X->Y, X

should be the super key of the table.

TABLE DESIGN

1. Login

|  |  |  |  |
| --- | --- | --- | --- |
| FIELDS | DATATYPES | SIZE | DESCRIPTION |
| Login\_id | int | 15 | Used for login id (primary key) |
| username | varchar | 30 | Used for username |
| password | varchar | 30 | Used for password |
| usertype | varchar | 30 | Used for usertype |

2. POLICE

|  |  |  |  |
| --- | --- | --- | --- |
| FIELDS | DATATYPES | SIZE | DESCRIPTION |
| Police\_id | int | 15 | Used for piloce id (primary key) |
| Login\_id | int | 15 | Used for login id (foreign key) |
| Station\_name | varchar | 30 | Used for station name |
| Phone | int | 15 | Used for phone |
| email | varchar | 30 | Used for Email |
| place | varchar | 30 | Used for Place |
| pincode | int | 15 | Used for Pincode |

3. USER

|  |  |  |  |
| --- | --- | --- | --- |
| FIELDS | DATATYPES | SIZE | DESCRIPTION |
| User\_id | int | 15 | Used for user id (primary key) |
| Login\_id | int | 15 | Used for login id (foreign key) |
| First\_name | varchar | 30 | Used for first name |
| Last\_name | varchar | 30 | Used for last name |
| location | varchar | 15 | Used for location |
| Address | varchar | 15 | Used for Address |
| pincode | int | 15 | Used for Pincode |
| Gender | varchar | 30 | Used for gender |
| DOB | date |  | Used for date of birth |
| photo | varchar | 150 | Used for photo |
| phone | int | 12 | Used for phone |
| email | varchar | 30 | Used for Email |

4. Document\_type

|  |  |  |  |
| --- | --- | --- | --- |
| FIELDS | DATATYPES | SIZE | DESCRIPTION |
| doc\_type\_id | int | 15 | Used for document id (primary key) |
| doc\_type\_name | varchar | 30 | Used for document name |
| doc\_type\_description | varchar | 30 | Used for document description |

5. punishment\_types

|  |  |  |  |
| --- | --- | --- | --- |
| FIELDS | DATATYPES | SIZE | DESCRIPTION |
| punishment\_type\_id | int | 15 | Used for document id (primary key) |
| punishment\_type\_name | varchar | 30 | Used for punishment name |
| punishment\_type\_description | varchar | 30 | Used for punishment description |
| penalty\_description | varchar | 30 | Used for penalty description |
| penalty\_amount | int | 15 | Used for penalty amount |

6. Vehicles

|  |  |  |  |
| --- | --- | --- | --- |
| FIELDS | DATATYPES | SIZE | DESCRIPTION |
| vehicle\_id | int | 15 | Used for vehicle id (primary key) |
| user\_id | int | 15 | Used for user id  (foreign key) |
| register\_no | int | 15 | Used for register number |
| manufacture\_year | date |  | Used for manufacture year |
| Brand | varchar | 30 | Used for brand |
| Model | varchar | 30 | Used for model |

7. Documents

|  |  |  |  |
| --- | --- | --- | --- |
| FIELDS | DATATYPES | SIZE | DESCRIPTION |
| document\_id | int | 15 | Used for document id (primary key) |
| document\_type\_id | int | 15 | Used for document id(foreign key) |
| user\_id | int | 15 | Used for user id |
| document\_file | varchar | 150 | Used for document file |
| doc\_number | int | 15 | Used for document number |

8. Punishments

|  |  |  |  |
| --- | --- | --- | --- |
| FIELDS | DATATYPES | SIZE | DESCRIPTION |
| Punishment\_id | int | 15 | Used for punishment id (primary key) |
| punishment\_type\_id | int | 15 | Used for punishment type id(foreign key) |
| user\_id | int | 15 | Used for user id |
| police\_id | int | 15 | Used for police id |
| punishment\_date | date |  | Used for punishment date |
| punishment\_status | varchar | 30 | Used for punishment status |

9. Payments

|  |  |  |  |
| --- | --- | --- | --- |
| FIELDS | DATATYPES | SIZE | DESCRIPTION |
| payment\_id | int | 15 | Used for Payment id (primary key) |
| user\_id | int | 15 | Used for User id  (foreign key) |
| Amount | int | 15 | Used for Amount |
| payment\_date | date |  | Used for Payment date |

10. insurance

|  |  |  |  |
| --- | --- | --- | --- |
| FIELDS | DATATYPES | SIZE | DESCRIPTION |
| Insurance\_id | int | 15 | Used for insurance id (primary key) |
| Insurancename | varchar | 30 | Used for insurance name |
| Description | varchar | 30 | Used for description |
|  |  |  |  |

11. insurance company

|  |  |  |  |
| --- | --- | --- | --- |
| FIELDS | DATATYPES | SIZE | DESCRIPTION |
| Company\_id | int | 15 | Used for company id (primary key) |
| Login\_id | int | 15 | Used for login id  (foreign key) |
| Company name | varchar | 30 | Used for company |
| Place | varchar | 30 | Used for place |
| Phone | int | 10 | Used for phone |
| Email | varchar | 25 | Used for email |

12. Request insurance

|  |  |  |  |
| --- | --- | --- | --- |
| FIELDS | DATATYPES | SIZE | DESCRIPTION |
| Request\_id | int | 15 | Used for request id (primary key) |
| Insurance\_id | int | 15 | Used for insurance id(foreign key) |
| Vehicle\_id | int | 15 | Used for vehicle id |
| Requestdate | date |  | Used for request date |
| Fromdate | date |  | Used for from date |
| Todate | date |  | Used for to date |
| Status | varchar | 30 | Used for status |

13. Insurance claim details

|  |  |  |  |
| --- | --- | --- | --- |
| FIELDS | DATATYPES | SIZE | DESCRIPTION |
| Claim\_id | int | 15 | Used for request id (primary key) |
| Insurance\_id | int | 15 | Used for insurance id(foreign key) |
| Claim\_date | date |  | Used for claim date |
| Parts\_changed | varchar | 30 | Used for parts changed |

14. complaints

|  |  |  |  |
| --- | --- | --- | --- |
| FIELDS | DATATYPES | SIZE | DESCRIPTION |
| Complaint\_id | int | 15 | Used for complaint id (primary key) |
| Sender\_id | int | 15 | Used for sender id |
| Complaint | varchar | 30 | Used for complaint |
| Reply | varchar | 30 | Used for reply |
| date | date | date | Used for date |

DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through

an information system, modeling its process aspects. A DFD is often used as a preliminary step

to create an overview of the system without going into great detail, which can later be

elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kind of information will be input to and output from the system, how the

data will advance through the system, and where the data will be stored. It does not show

information about process timing or whether processes will operate in sequence or parallel,

unlike a traditionally structured flowchart that focuses on control flow, or a UML activity

workflow diagram, which presents both control and data, flows as a unified model.

● RULES FOR DRAWING DATA FLOW DIAGRAM

Each data store should have at least one data flow in and one data flow out.

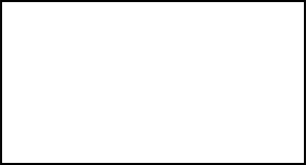
Data stored in a system must go through a process.

All processes in a DFD go to another process or a data store.

● BASIC DFD SYMBOLS

The primitive symbols used in the DFD are:

External entity



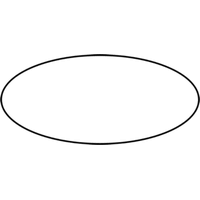
The external entities are essentially those physical entities external to the software system which

interact with the system by inputting data to the system or by consuming the data produced by

the system. For example, the user of a system, Entities supplying data are known as sources and

those that consume data are sinks.

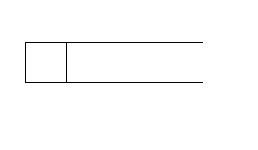
Process

  
The functions are using circles. Bubbles are annotated with the names of the corresponding functions. They convert data into information.

Data Flow

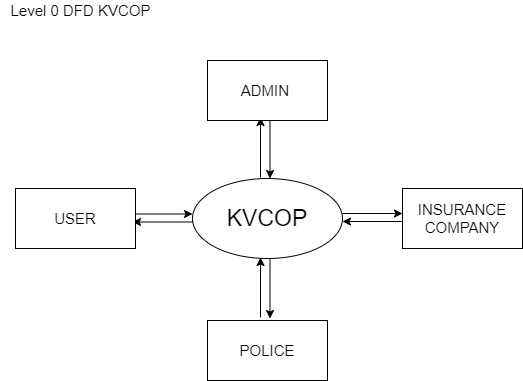
A directed arrow or arc is used as a data flow symbol that represents the data flow occurring between two processes, or between an external entity and a process.

Data Store

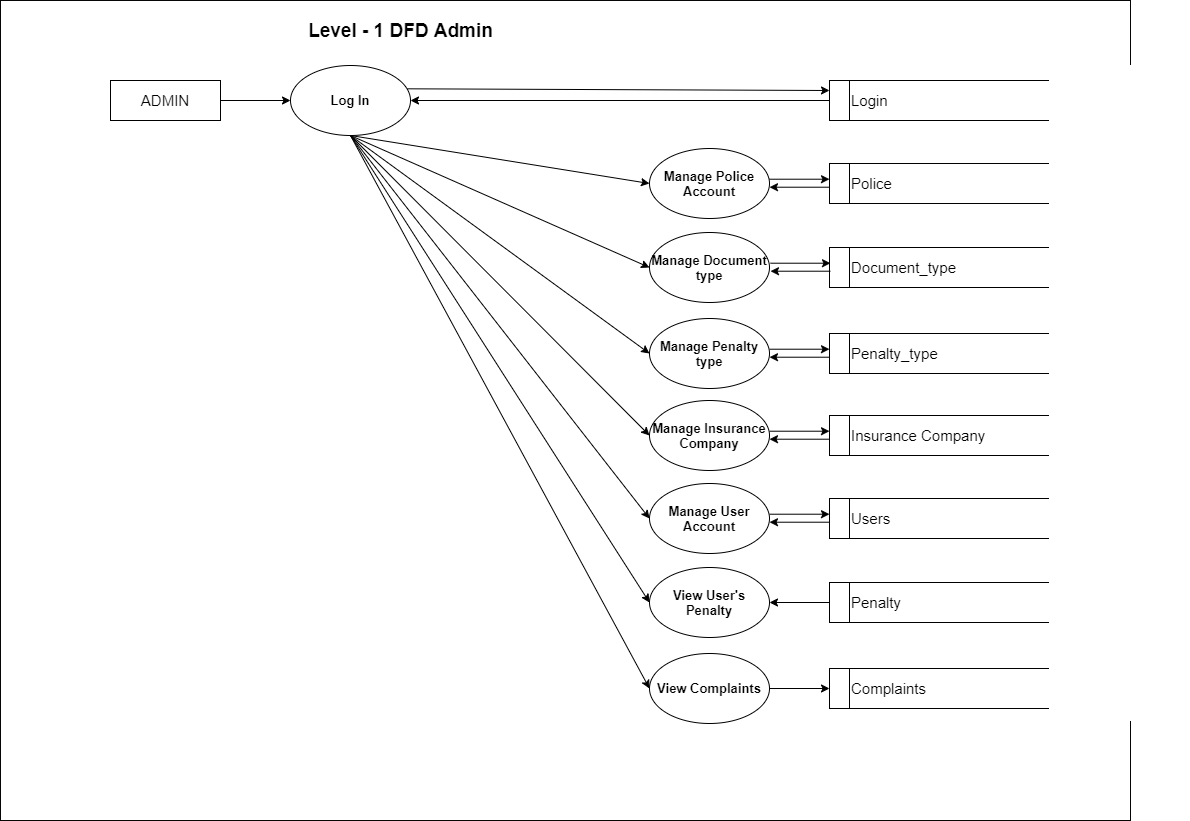


A data store represents logical files. Each datastore is connected to a process using a data flow symbol.

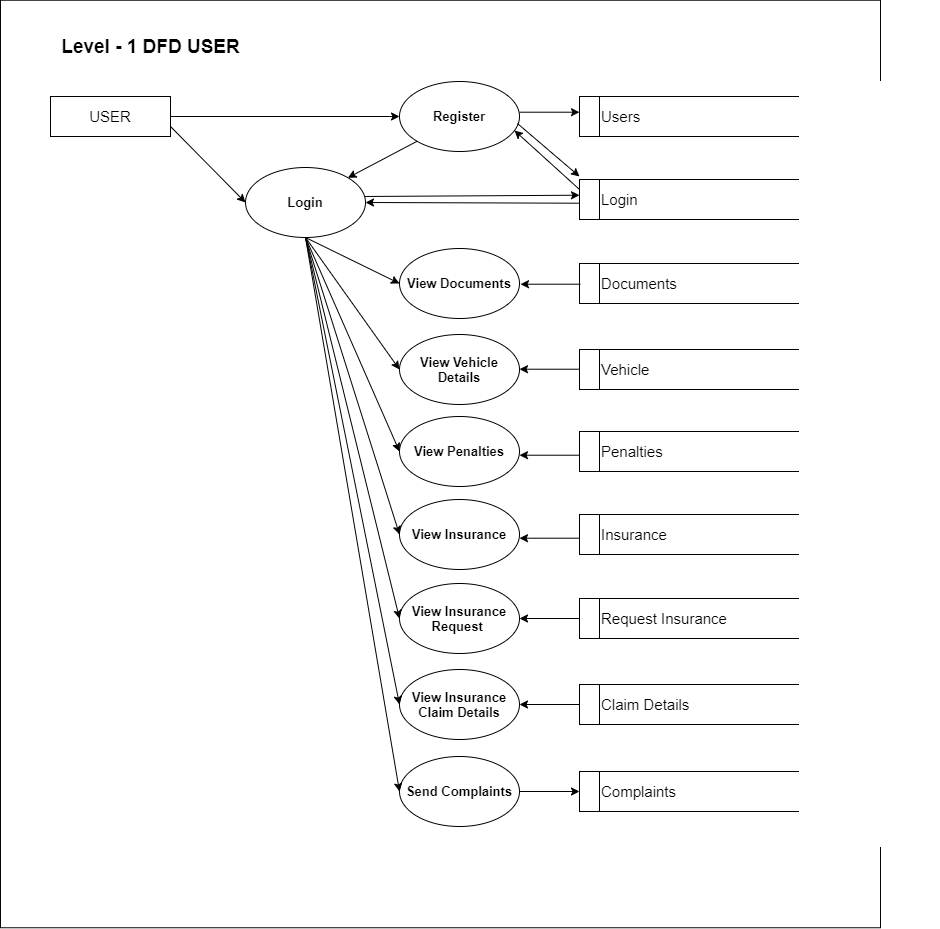
LEVEL 0 (CONTEXT DIAGRAM)



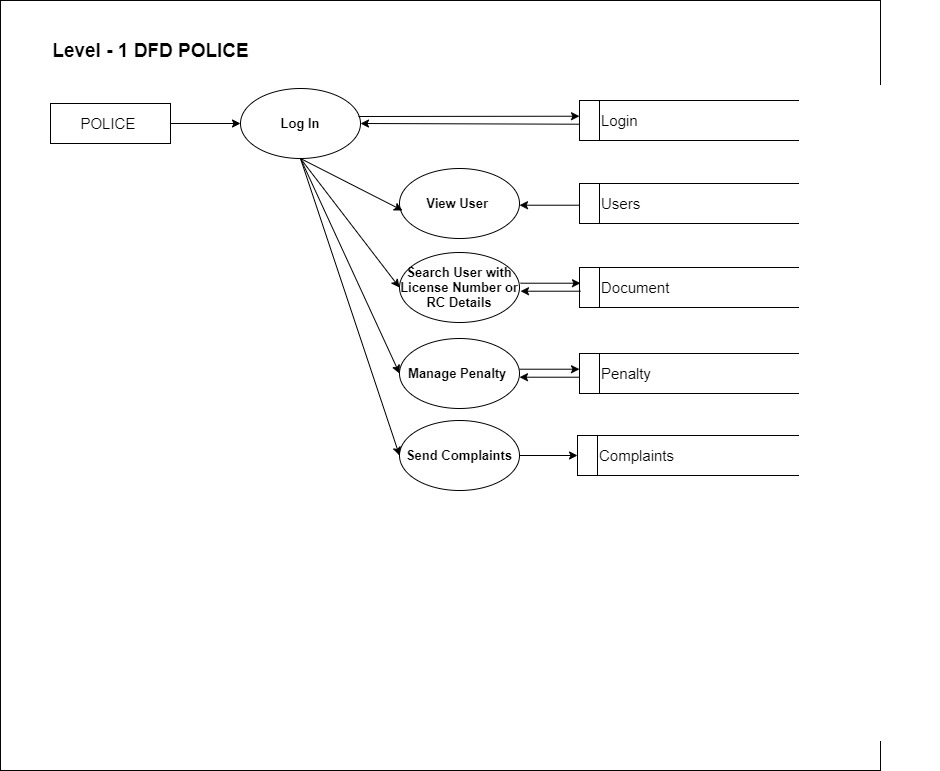
LEVEL 1 DFD FOR ADMIN



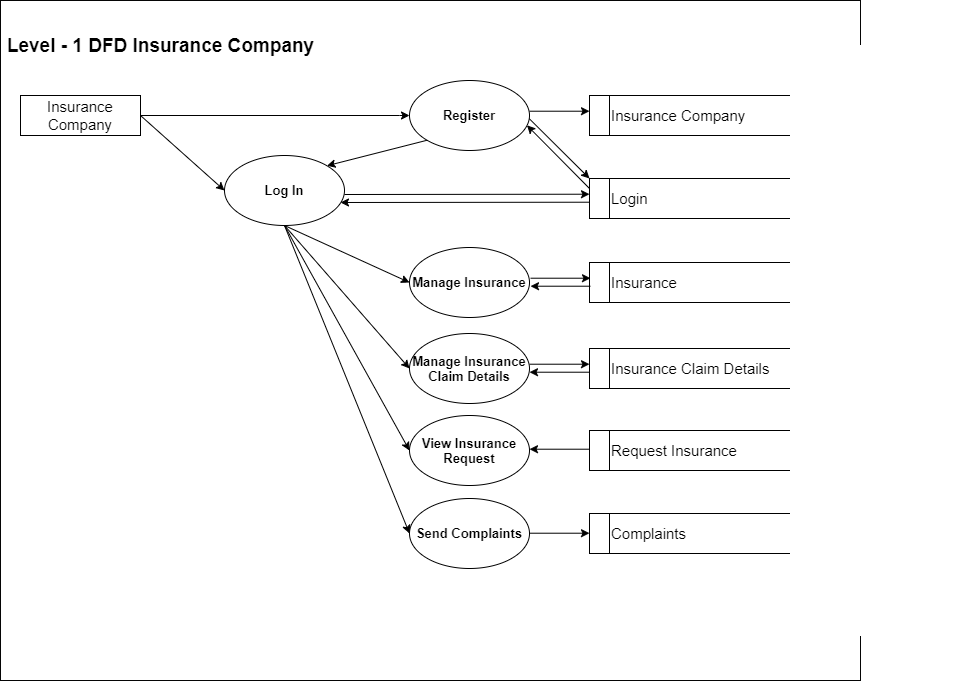
Level 1 DFD For user



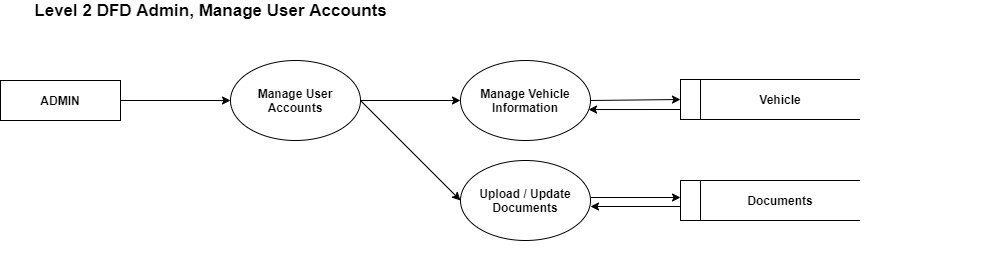
Level 1 DFD For Police



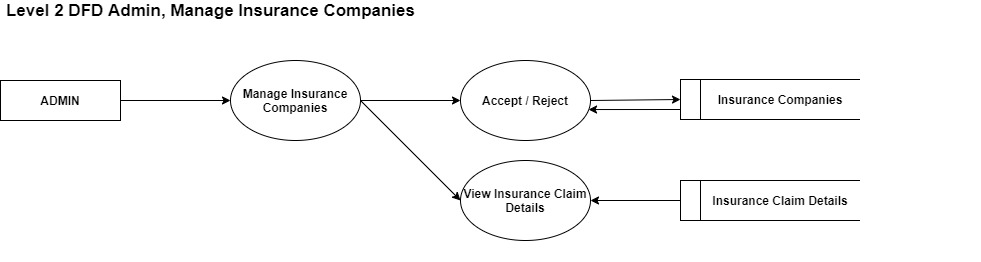
Level 1 DFD For Insurance Company



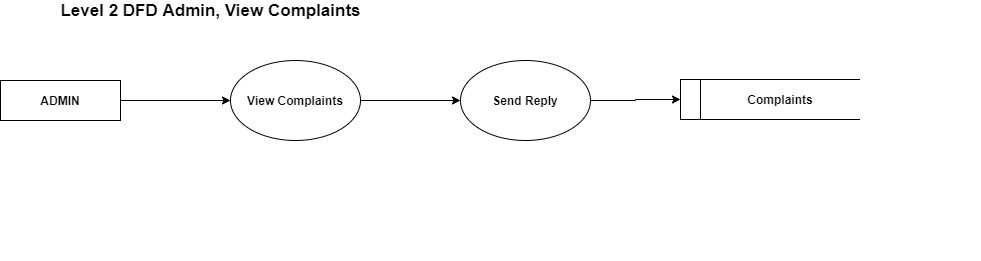
Level 2 DFD ADMIN, Manage User Accounts



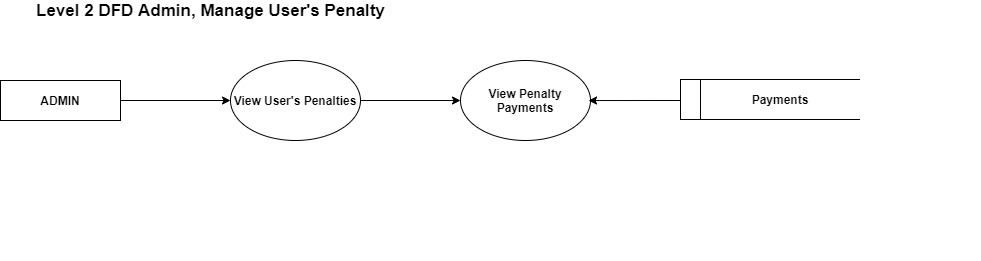
Level 2 DFD ADMIN, Manage Insurance Companies



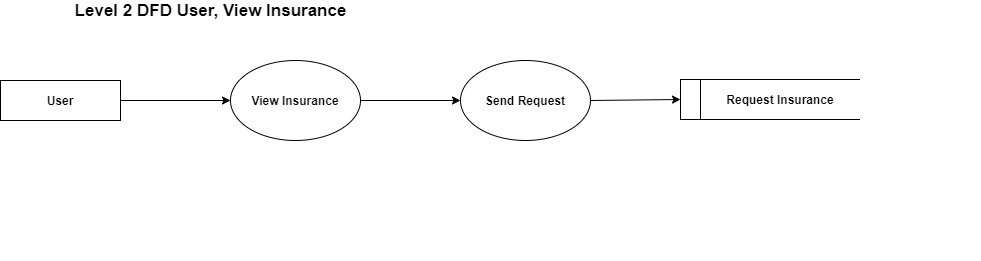
Level 2 DFD ADMIN, View Complaints



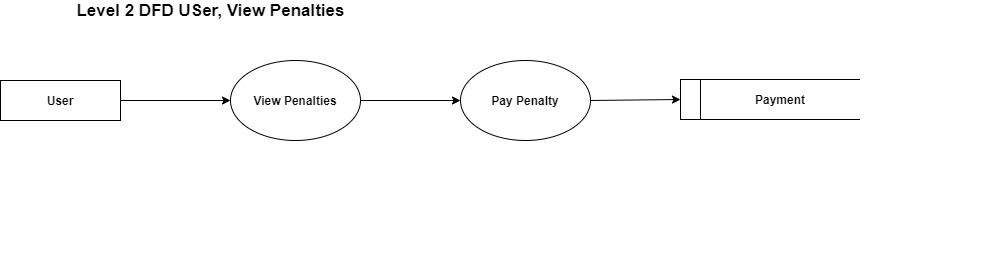
Level 2 DFD ADMIN, Manage User’s Penalty



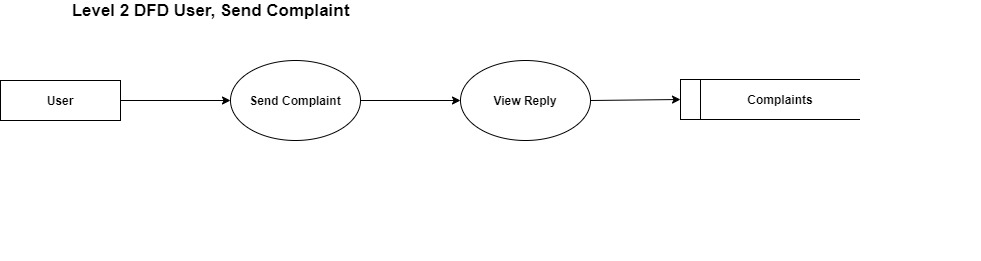
LEVEL 2 DFD USER, View Insurance



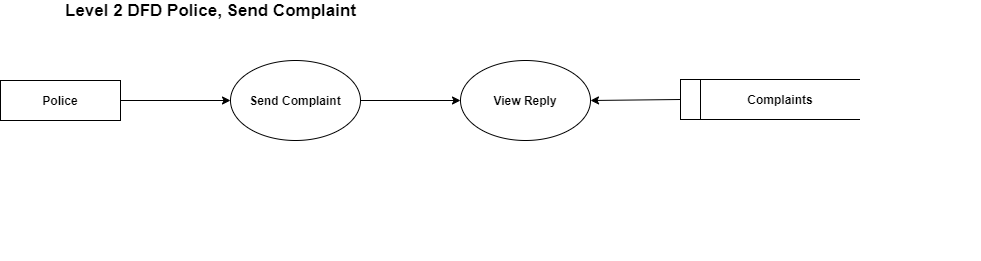
LEVEL 2 DFD USER, View Penalties



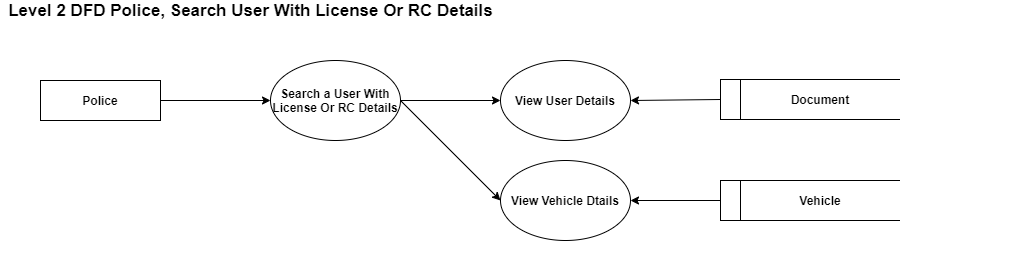
LEVEL 2 DFD USER, Send Complaints



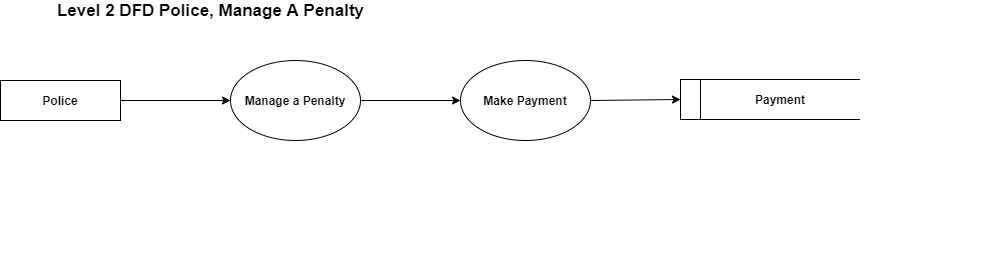
LEVEL 2 DFD POLICE, Send Complaints



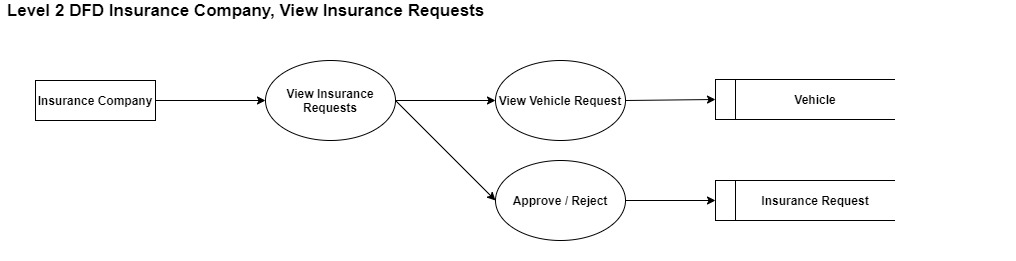
LEVEL 2 DFD POLICE, Search User with License Or RC Details



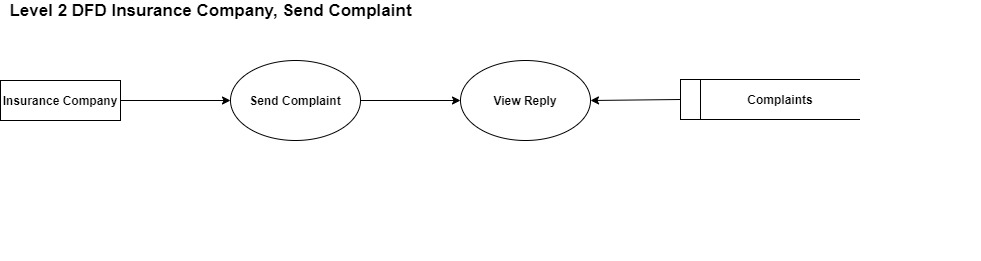
LEVEL 2 DFD POLICE, Manage a Penalty



LEVEL 2 DFD INSURANCE COMPANY, View Insurance Request



LEVEL 2 DFD INSURANCE COMPANY, Send Complaint

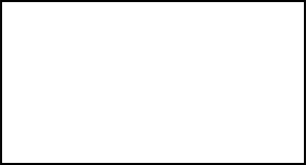


ENTITY-RELATIONSHIP DIAGRAM

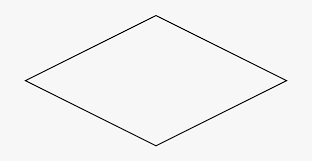
The ER diagram is a graphical representation of entities and their relationships to each other, typically used in computing regarding the organization of data within database or information systems. An entity is a piece of data an object or concept about which data is stored. A relationship is how the data is shared between entities.

ER-SYMBOLS

ENTITY

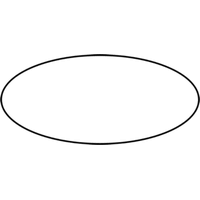


An entity is defined as a thing that is recognized as being capable of independent existence and which can be uniquely identified. An entity is abstract from the complexities of a domain. When we speak of an entity, we normally speak of some aspect of the real world that can be distinguished from other aspects of the world.

RELATIONSHIP

The relationship is a meaningful association between or among entities. A relationship provides useful information that could not be discerned with just the entity types.

ATTRIBUTES



Attributes are characteristics of an entity, a many-to-one relationship, or a one-to-one relationship.

● ER DIAGRAM



USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user’s interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. While a use case itself might drill into a lot of detail about every possibility, a use-case diagram can help provide a higher-level view of the system. It has been said before that” Use case diagrams are the blueprints for your system”. They provide a simplified and graphical representation of what the system must do.

GUIDELINES FOR DRAWING USE CASE DIAGRAM

A use case describes how a user uses a system to accomplish a particular goal. A use case diagram consists of the system, the related use cases, and actors and relates these to each other to

visualize: what is being described? (system), who is using the system? (actors) and what do the actors want to achieve? (use cases), thus, use cases help ensure that the correct system is developed by capturing the requirements from the user’s point of view.

• Give meaningful business-relevant names for actors – For example, if your use case interacts with an outside organization it’s much better to name it with the function rather than the organization name. (Eg: Airline Company is better than PanAir)

• Primary actors should be to the left side of the diagram – This enables you to quickly highlight the important roles in the system.

• Actors model roles (not positions) – In a hotel both the front office executive and shift manager can make reservations. So, something like “Reservation Agent” should be used for the actor's name to highlight the role.

• External systems are actors – If your use case is send-email and if interacts with the email management software then the software is an actor to that particular use case.

• Actors don’t interact with other actors – In case actors interact within a system you need to create a new use case diagram with the system in the previous diagram represented as an actor.

• Place inheriting actors below the parent actor – This is to make it more readable and to quickly highlight the use cases specific for that actor.

USE CASES

• Names begin with a verb – A use case models an action so the name should begin with a verb.

• Make the name descriptive – This is to give more information for others who are looking at the diagram. For example, “Print Invoice” is better than “Print”.

• Highlight the logical order – For example, if you’re analyzing a bank customer typical use cases include open account, deposit, and withdrawal. Showing them in the logical order makes more sense.

• Place included use cases to the right of the invoking use case – This is done to improve readability and add clarity.

• Place inheriting use case below parent use case – Again this is done to improve the readability of the diagram.

BASIC USE CASE DIAGRAM SYMBOLS

The primitive symbols used in the USE CASE are:

• System

System name

Draw your system’s boundaries using a rectangle that contains use cases. Place actors outside the system’s boundaries.

• Use Case

Draw use cases using ovals. Label the ovals with verbs that represent the system’s functions.

• Actor

Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype. An actor is a person, organization, or external system that plays a role in one or more interactions with your system (actors are typically drawn as stick figures on UML Use Case diagrams).

• Relationship

<<include>>

<<extend>>

< ---------------

Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either” uses” or” extends.” A” uses” relationship indicates that one use case is needed by another to perform a task. An” extends” relationship indicates alternative options under a certain use case.

• USE CASE

