

Practical :- 1

Aim :- Define the Problem statement of Software development and identify the Requirements based on the requirement engineering tasks.

Theory:

❖ Phases in software development project:

Software Development Life Cycle is well defined and structured sequence of stages that is used in software engineering to build high quality and intended software. SDLC provide series of steps to be followed to build and design software product efficiently.

Steps in SDLC:

1) Requirement Gathering and Analysis:

This is the initial step of life cycle model. In this step all the requirement of user for software are collected from users. They are provided document called business requirement specification or System Requirement Specification. All there requirements are discussed in detail with client/costumer of software in this phase. Team of software development holds meeting to stakeholders to solve the query about requirement and try to get more and more detail from stakeholder. Requirements are segregated into functional requirement and non-functional requirement.

2) Feasibility Study:

After gathering requirement, team come out with rough plan of software. At this step team analyses if software can be made that fulfil all the requirement or not. It is found out that the software will be ready in given financial condition and in given time deal line. There are many algorithms available that help team to study feasibility study of software.

3) System Analysis:

At this step team decide roadmap of their plan and choose best software model to develop software. System analysis include analysis of software limitation, software related changes to be done on existing software, identifying and addressing the impact of software on organization and personal etc. The project team analyses the scope of the software and plans the schedule and resources accordingly.

4) Software Design:

In this step, team bring down whole knowledge of requirement gathering on the desk and design software product. The input from user and information gathered in step of requirement gathering is the input at this stage. The output of this stage is come in form of two design:

- Logical design
- Physical design

Team prepare meta data, data dictionaries, logical diagram, use case diagram, dataflow diagram and in some cases pseudo codes.

5) Coding:

This step is also known as programming stage. The implementation of software design starts in terms of writing program code in suitable programming language and deploy error free and executable program efficiently.

6) Testing:

At estimate 50% of software development process should be tested. Error may ruin the software from critical condition to its removal. Software testing is done while coding and testing experts test software at various level like module testing, program testing, in-house testing and testing at end-user side. Early discovery of model and their remedy is the key for reliable software.

7) Integration:

Software may need to integrate with various library, database and other programs. This stage of SDLC include integration of software program to outer world entities.

8) Implementation:

In this step installation of software on user machine will take place. In this step software needs post install configurations at user end. Software is tested for portability and adaptability and integration related issues are to be solved at this step.

9) Operation and maintenance:

In this step confirms the software operation in terms of efficiency and les errors. If required users are trained for this and/or documentation provide for easy use of software and keep software in operational state. Software maintenance take place by updating code accordingly user's requirement and change made at user end and update software with new technologies. This stage may face problem of hidden bugs and real-world unidentified problems.

10) Disposition:

As time elapsed, the software may decline on the performance front. It may be completely obsolete or may require intense up gradation. Hence pressing need to eliminate major portion of the system arise. This phase may include achieving data and required software component, closing down system, planning disposition activity and terminating system at end of appropriate system time.

- **Background/Preparation:**

Title :- REAL-STATE MANAGEMENT SYSTEM

Our team put serious efforts to understand the problem and then we collectively defined the problem statement for the software to be developed. The requirement engineering task were performed to gather the requirements shown below.

- **Tools/material needed:**

Hardware: Computer, Pen, Paper

Software: M.S word

- **Procedure:**

Problem statement:

For buying a new property, users needs to travel. Users have to spend more time and money for travelling or searching the desired property. So we come up with our project which reduces the searching time and travelling cost and helps users to find their desired property.

Functional and non-functional requirements :

- Error handling
- Viewing sales statistics
- Updating details
- Performance
- Reliability
- Security
- Availability

Practical :- 2

AIM: Prepare the Software Requirement Specification (SRS) document

1.0 Introduction:

1.1 Purpose :

The purpose of this document is to present a detailed description of the Real-estate management system. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate.

1.2 Scope of project :

This project helps the users to make good decisions regarding buying or selling of valuable property. Prior to this online system this process involved a lot of travelling costs and searching time. Due to this system the user now does not have to travel much and can look for the property it is searching for, online according to its requirements. This system includes property details like Address, space measurement (sq ft), number of BHKs, Floor, Property Seller name and its contact number plus email-id. The user can search property depending on the area that it wants in, number of wash rooms, bedrooms, halls and kitchen. The system contains an algorithm that calculates loan that the user can take plus 20%-30% cash that the user has to pay. This system allows the admin to enter details about any property that it is wishing for. The admin can even delete the property details.

Thus this system can be used by any user looking to buy a property. Even the property sellers can use this application and reach out to large number of buyers looking for property. Due to this system the user now does not have to travel much and can look for the property it is searching for, online according to its requirements. This system eliminates cost to a great extent and also reduces searching time.

Provide a superior real estate experience by making it easier, secure, faster and more accurate to find buyers for valuable property. We give our customers ease

to use and also we will maintain good relationship with the customers, brokers, sellers for properties.

1.3 Objective:

The project is to bring the real estate industry online and enabling real estate industry participants to benefit from the internet. Site act as an interface between individuals, brokers and realtors. Here the user can advertise his property for buying or selling.

1.4 Overview of document :

This system eliminates cost to a great extent and also reduces searching time. With the help of this system the user can get the property details depending on its preferences. Thus this system also helps to maintain good relationship between the buyers and the sellers of the property.

1.5 Problem statement:

For buying a new property, users needs to travel. Users have to spend more time and money for travelling or searching the desired property. So we come up with our project which reduces the searching time and travelling cost and helps users to find their desired property.

2.0 Overall description:

2.1 Modules:

- Admin
- Sellers (Builders/Brokers/Direct sellers)
- Buyers

2.2 Features:

- Signup/Login
- Add property
- Update property information
- Delete property
- Search property
- View property
- Loan calculation
- Direct contact to seller
- Add property to wish list
- Give/view feedbacks

2.3 Advantages:

- This system reduces searching time and eliminates travelling cost.
- This system gives details of the property to the buyers depending on the preferences entered.

2.4 Limitations:

- If server fails, system may not work properly.

2.5 Assumption and dependencies:-

- Well created flexible database
- High connectivity
- Sellers
- Buyers

3.0 Requirement specification:

3.1 Functional requirement:

1) Admin

- Manage users:

Admin can manage users in the system.

- Manage security:

Admin can manage security of the system.

2) Sellers

- Add property:

Sellers can add properties.

- Manage quotation:

Sellers can manage quotation of the property.

- Update details:

Sellers can update property details.

- Delete property:

Sellers can delete property.

- View feedback:

Sellers can view feedback given by the users.

3) Buyers

- View property:

Buyers can view property.

- View quotation:

Buyers can view quotation.

- Calculate loan:

Buyers can calculate loan.

- Contact to sellers:

Buyers can contact to sellers.

- Give feedback:

Buyers can give feedback.

3.2 External interface requirements :

- **Sellers interface**

The RMS screen displays an interface to add, update and delete the properties.

- **Buyers interface**

The RMS screen displays an interface to view property and property details and book the property.

- **Admin interface**

The RMS display the interface to manage users and security of the system.

3.3 Non-functional requirement :-

- **Performance**

The system must be interactive and the delays involved must be less. So in every action-response of the system, there are no immediate delays.

- **Reliability**

As the system provide the right tools for discussion, problem solving it must be made sure that the system is reliable in its operations and for securing the sensitive details.

- **Safety**

Information transmission should be securely transmitted to server without any changes in information.

- **Availability**

If the internet service gets disrupted while sending information to the server, the information can be send again for verification.

→ **Requirements with their input, output and processing :**

Requirement No (Module No. Req No.)	Description	Type System /User	Nature Functional /Non Functional
M1.R1	Error handling The RMS may not be able to connect to the server due to error in network connection, in the case of which transaction is not possible. 1.user-admin 2.priority-immediate	System	Functional
M1.R2	Viewing sales statistics R2.1- Input <ul style="list-style-type: none"> • Item identification parameter (such as product ID or name). • Time period or duration. R2.2 – Processing <ul style="list-style-type: none"> • The RMS looks into the database, the cost and selling price of the particular property for every transaction in that period and generates the profit statistics in the requested format. R2.3 - Outputs <ul style="list-style-type: none"> • The profit statistics are displayed in the requested format for the manager, which he prints for his convenience. 	User	Functional
M1.R3	Updating the prices/details R3.1 – Inputs	User	Functional

	<ul style="list-style-type: none"> • The product identification parameter (such as product ID or name). • New Price for the property. <p>R3.2 – Processing</p> <ul style="list-style-type: none"> • The RMS looks into the database and shows the property information. • It updates the database with the new price or new details. <p>R3.3 - Outputs</p> <ul style="list-style-type: none"> • The property information with updated price is shown. <p>1.User-sellers 2.Priority-high</p>		
M1.R4	<p>Updating the details</p> <p>R4.1 - Inputs</p> <ul style="list-style-type: none"> • The product ID <p>R4.2 - Processing</p> <ul style="list-style-type: none"> • The RMS looks into the database, if the property ID already exists in the inventory database, otherwise new property information has to be added to the database. <p>R4.3 - Outputs</p> <ul style="list-style-type: none"> • A message is displayed confirming the update regarding the property ID and amount. <p>1.User-sellers 2.Priority-high</p>	User	Functional

M2.R1	Performance High level of performance requires high speed network and high level of connectivity.	System	Non functional
M2.R2	Reliability The available server must be reliable and the network connectivity should be proper for smooth flow of all operations and data.	System	Non functional
M2.R3	Security Every user of the software is provided a unique login ID and a password which is stored in the database hashed by SHA2 algorithm	User	Non functional
M2.R4	Availability The software is available for use anytime.	System	Non functional

3.4 Other requirements :

3.4.1 Software Requirements:

- Windows Xp, Windows 7(ultimate, enterprise)
- Sql 2008
- Visual studio 2010

3.4.2 Hardware Components:

- Processor – i3
- Hard Disk – 5 GB
- Memory – 1GB RAM

3.4.3 Tools and technologies:

- Operating system: Windows XP/2000/Vista/7/8
- Software requirements for development: Wamp/Wordpress/edit++
- Front-end technology: HTML/JavaScript/CSS
- Back-end technology: PHP
- Database server: MySQL

3.5 References :

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998.

Practical :- 3

Aim :- Prepare the software management plan (SPMP) including following :

- Estimation of Size, Cost, Duration, Effort
- Prepare the Schedule, Milestone
- Identification of Risk, Risk management

1) Estimation of Size, Cost, Duration, Effort

❖ Estimation of Size:

❖ External inputs:

- Login to the system and manage all the functionalities of Real-estate management system.
- Add the records of Property, Purchasing, Payment and Login.
- Edit the records of Property, Purchasing, Payment and Login.
- Delete the records of Property, Purchasing, Payment and Login.
- View the records of Property, Purchasing, Payment and Login.
- Manage all the details of User.
- Manage all the details of Sales.
- Manage all the details of Property.
- Track the detailed information of the User.
- Track the detailed information of the Purchasing.
- Track the detailed information of the Sales.
- Track the detailed information of the Payment.

❖ External outputs:

- Generate reports of Property.
- Generate reports of User.
- Generate reports of Purchasing.
- Generate reports of Sales.
- Generate reports of Payment.

❖ External enquiry:

- Search the details of Users.
- Search the details of Payment.
- Search the details of Property.
- Apply different levels of filters on report of Sales.
- Apply different levels of filters on report of Payment.

❖ External interface files:

- Wireless network interface.

❖ Internal logical file:

- User table
- Login table
- Property table
- Sales table
- Payment table

		Function Levels			
Components	Count	Low	Average	High	Total
External Input	12	1 x 3	7 x 4	4 x 6	55
External Output	5	1 x 4	2 x 5	2 x 7	28
External Enquiries	5	0 x 3	2 x 4	3 x 6	26
Internal Logical Files	5	5 x 7	0 x 10	0 x 15	35
External Interface Files	1	0 x 5	0 x 7	1 x 10	10
Total Unadjusted FP (UFP):					154

[Unadjusted function point table]

• Degree of Influence (DI) :

Characteristic	Description	Degree of Influence (DI)
1. Data Communication	Are specialized data communications required to transfer information to or from the application?	4
2. Distributed Data Processing	Are there distributed processing functions?	2
3. Performance	Is performance critical?	3
4. Heavily Used Configuration	Will the system run in an existing, heavily utilized operational environment?	0
5. Transaction Rate	Does the online data entry require the input transaction to be built over multiple screens or operations?	3
6. Online Data Entry	Does the online data entry required?	4

7. End-user Efficiency	Is the application designed for end-user efficiency?	5
8. Online Update	Are the ILFs updated online?	3
9. Complex Processing	Are the inputs, outputs, files, inquiries or internal processing complex?	2
10. Reusability	Is the code designed to be reusable?	5
11. Reliable Backup and Recovery	Does the system require reliable backup and recovery?	5
12. Operational Ease	How effective and/or automated are start-up, back-up, and recovery procedures?	3
13. Multiple installation	Is the system designed for multiple installations in different organizations?	0
14. Facilitate Change	Is the application designed to facilitate change and for ease of use by the user?	4
Total Degree Of Influence (TDI):	43	

[Degree of Influence]

- Scale criteria:

Rating	Degree of Influence
0	Not present, or no influence
1	Incidental influence
2	Moderate influence
3	Average influence
4	Significant influence
5	Strong influence throughout

❖ Size estimation:

Three Formulas for Size Estimation of the FIS;

→ Total Degree of Influence (TDI) = Sum of the Influence Factors= TDI=43

→ Value Adjustment Factor (VAF) = $0.65 + (0.01 * \text{TDI})$
= $0.65 + (0.01 * 43)$
= 1.08

→ Function Points (FP) = $\text{VAF} * \text{unadjusted FP} = 1.08 * 154$
= 166.32

→ Line of Code = $(\text{SLOC per FP for Java}) * \text{FP} = 30 * 166.32$
= 4,989.6

→ KLOC = 4.9896

❖ Effort Estimation:

Software Projects	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

→ Real estate Application is an Organic Project.

→ Effort = $a * (\text{KLOC})^b$
= $2.4 * 4.9896^{1.05}$
= $2.4 * 5.40$
= 12.96 (Man per month)

❖ Duration Estimation:

$$\begin{aligned}\text{Development Time (T)} &= c * (\text{Effort Applied})^d \\ &= 2.5 * (12.96)^{0.38} \\ &= 2.5 * 2.64 \\ &= 6.6 \text{ (Months)}\end{aligned}$$

$$\text{Staff} = \text{Effort} / \text{Duration} = 12.96 / 6.6 \approx 1.96 \text{ persons}$$

❖ Cost Estimation:

The salary of each group member is 40000Rs./p.m @ 152Hrp.m, therefore the estimated cost of the Project is:

$$12.96 \text{ man per month} = 12.96 \times 152 \text{ (hours in a month)} = 1969.92 \text{ man per hours}$$

$$\text{Cost} = \text{Effort} \times \$ \text{ (the hourly salary)}$$

$$= 1969.92 \text{ man per hours} \times (40000/152) \text{ man per hours}$$

$$= 518400 \text{ Rs (Eight Lacs One Thousand)}$$

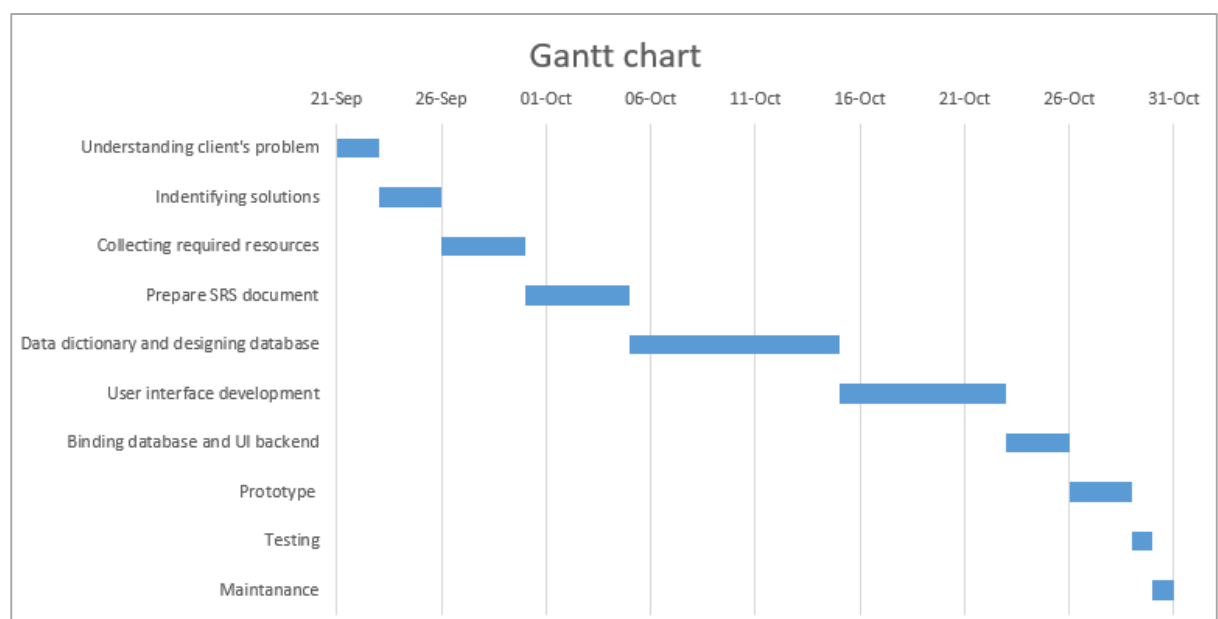
2) Prepare the Schedule, Milestones

❖ Scheduling tasks :

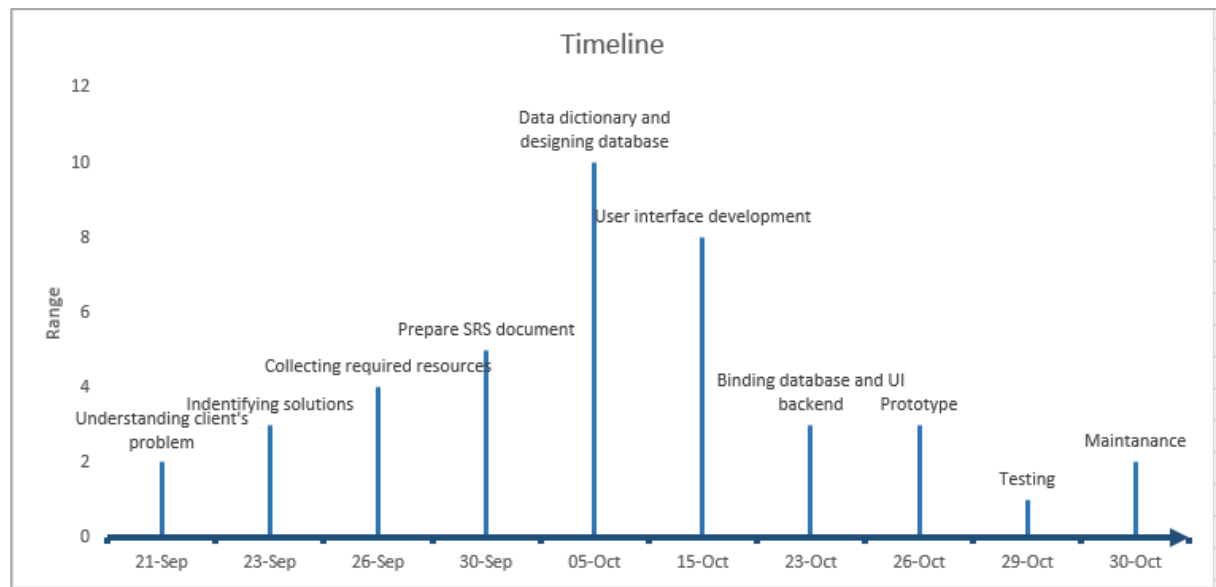
Tasks	Status	Starting date	Ending date
Understanding client's problem	Completed	21-Sep	22-Sep
Identifying solutions	Completed	23-Sep	25-Sep
Collecting required resources	Completed	26-Sep	29-Sep
Prepare SRS document	In process	30-Sep	04-Oct
Data dictionary and designing database	In process	05-Oct	14-Oct
User interface development	In process	15-Oct	22-Oct
Binding database and UI backend	Pending	23-Oct	25-Oct
Prototype	Pending	26-Oct	28-Oct
Testing	Pending	29-Oct	30-Oct
Maintenance	Pending	30-Oct	31-Oct

❖ Gantt chart :

- ➔ A Gantt chart is a horizontal bar chart showing the start and end dates of each task within a project. It shows the tasks on the vertical axis and time on the horizontal axis. The tasks are shown sequentially.



❖ Timeline :



3) Identification of Risk, Risk Management

Identifying risk is one of most important or essential and initial steps in risk management process. By chance, if failure occurs in identifying any specific or particular risk, then all other steps that are involved in risk management will not be implemented for that particular risk.

To manage risk, project team or organization are needed to know about what risks it faces, and then to evaluate them.

➔ Budget Risk :

Budget related risks refers to the monetary risks mainly it occurs due to budget overruns. Always the financial aspect for the project should be managed as per decided but if financial aspect of project mismanaged then there budget concerns will arise by giving rise to budget risks. So proper finance distribution and management are required for the success of project otherwise it may lead to project failure.

Some reasons for Budget risks –

- Wrong/Improper budget estimation
- Unexpected Project Scope expansion
- Mismanagement in budget handling
- Cost overruns
- Improper tracking of Budget

➔ **Operational Risks :**

Operational risk refers to the procedural risks means these are the risks which happen in day-to-day operational activities during project development due to improper process implementation or some external operational risks.

Some reasons for Operational risks –

- Insufficient resources
- Improper management of tasks
- No proper planning about project
- Less number of skilled people
- Lack of communication and cooperation
- Lack of clarity in roles and responsibilities
- Insufficient training

➔ **Technical Risks :**

Technical risks refers to the functional risk or performance risk which means this technical risk mainly associated with functionality of product or performance part of the software product.

Some reasons for Technical risks –

- Frequent changes in requirement
- Less use of future technologies
- High complexity in implementation
- Improper integration of modules

➔ **Programmatic Risks :**

Programmatic risks refers to the external risk or other unavoidable risks. These are the external risks which are unavoidable in nature. These risks come from outside and it is out of control of programs.

Some reasons for Programmatic risks –

- Rapid development of market
- Running out of fund / Limited fund for project development
- Changes in Government rules/policy
- Loss of contracts due to any reason

Practical :- 4

AIM: Prepare the following components of Data Flow Model:

- Data Dictionary
- Data Flow Diagram
- Structure Chart

- **Theory:**

Data Dictionary:

A data dictionary is a file or a set of files that includes a database's metadata. The data dictionary holds records about other objects in the database, such as data ownership, data relationships to other objects, and other data. The data dictionary is an essential component of any relational database. Ironically, because of its importance, it is invisible to most database users. Typically, only database administrators interact with the data dictionary.

The data dictionary, in general, includes information about the following:

- Name of the data item
- Aliases
- Description/purpose
- Related data items
- Range of values
- Data structure definition/Form

Data Flow Diagram

Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both

It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a

whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

In Software engineering DFD (data flow diagram) can be drawn to represent the system of different levels of abstraction. Higher-level DFDs are partitioned into low levels-hacking more information and functional elements. Levels in DFD are numbered 0, 1, 2 or beyond. Here, we will see mainly 3 levels in the data flow diagram, which are: 0-level DFD, 1-level DFD, and 2-level DFD.

0-level DFD: It is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

1-level DFD: In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses.

2-level DFD: 2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system's functioning.

Structured Chart:

- Structured analysis is used to carry out the top-down decomposition of a set of high-level functions depicted in the problem description and to represent them graphically. During structured analysis, functional decomposition of the system is achieved.
- Background / Preparation: Team did deep study of the solution and it's components to derive attributes of data dictionary, data flow diagram and its different levels and structure diagram.
- Tools / Material Needed:
 - o Hardware: Computer
 - o Software: Draw.io, M.S. word

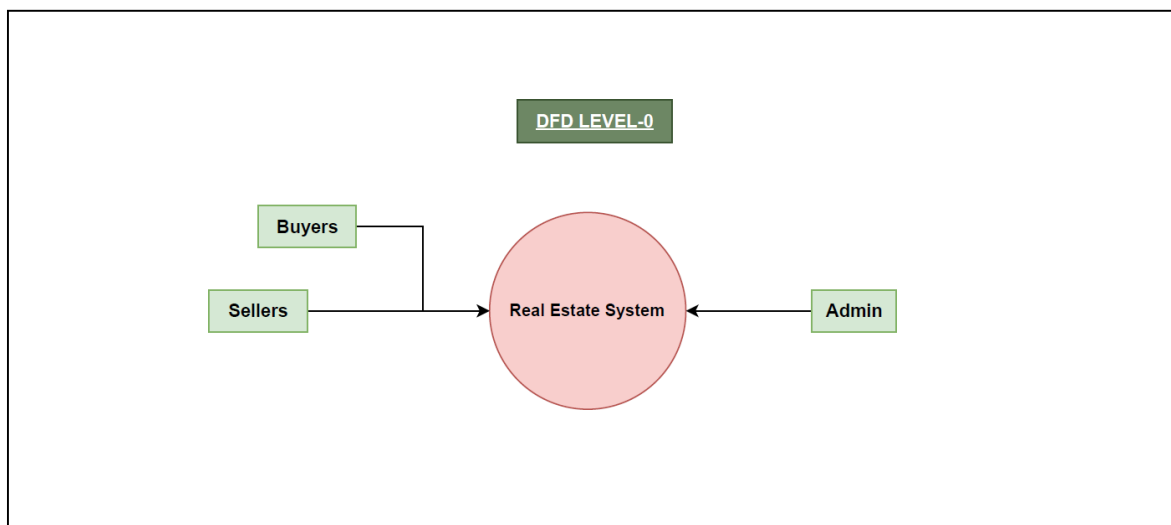
❖ Procedure:

1. Data flow Diagrams :

0 Level Diagram :

High Level Entities and process flow of Real Estate Sysyem:

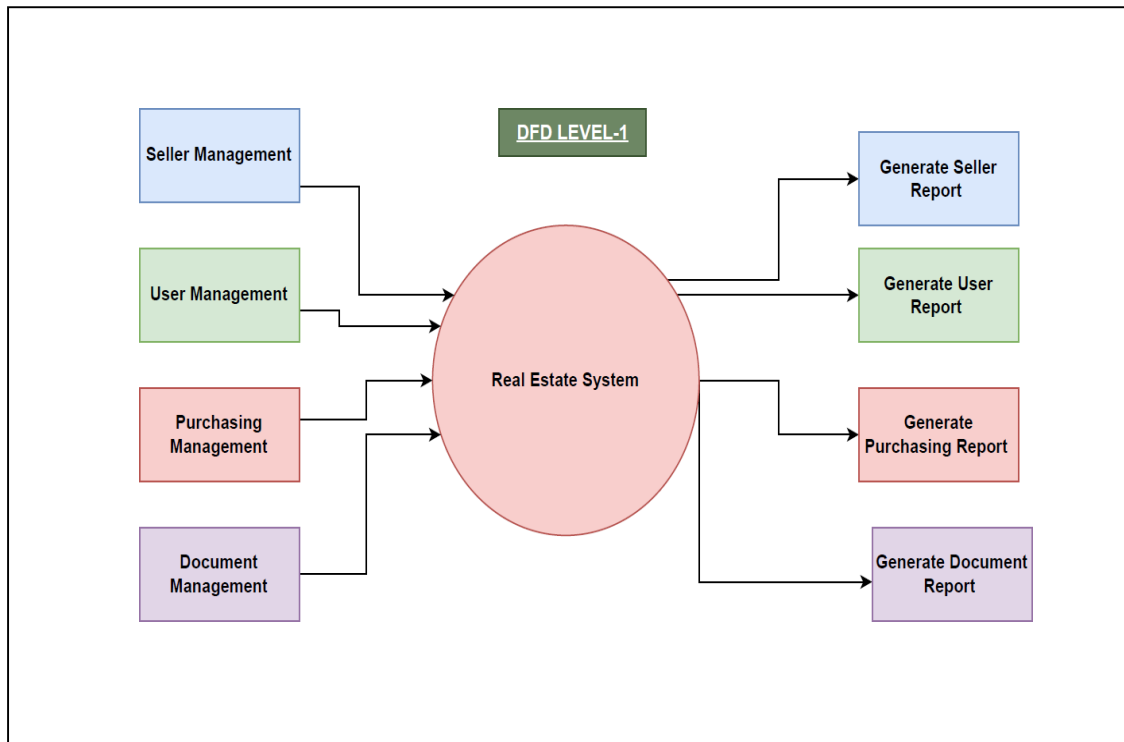
- Managing all the Properties
- Managing all the Buyers
- Managing all the Purchasing
- Managing all the Sales
- Managing all the Payment
- Managing all the Login



b. 1st Level Diagram :

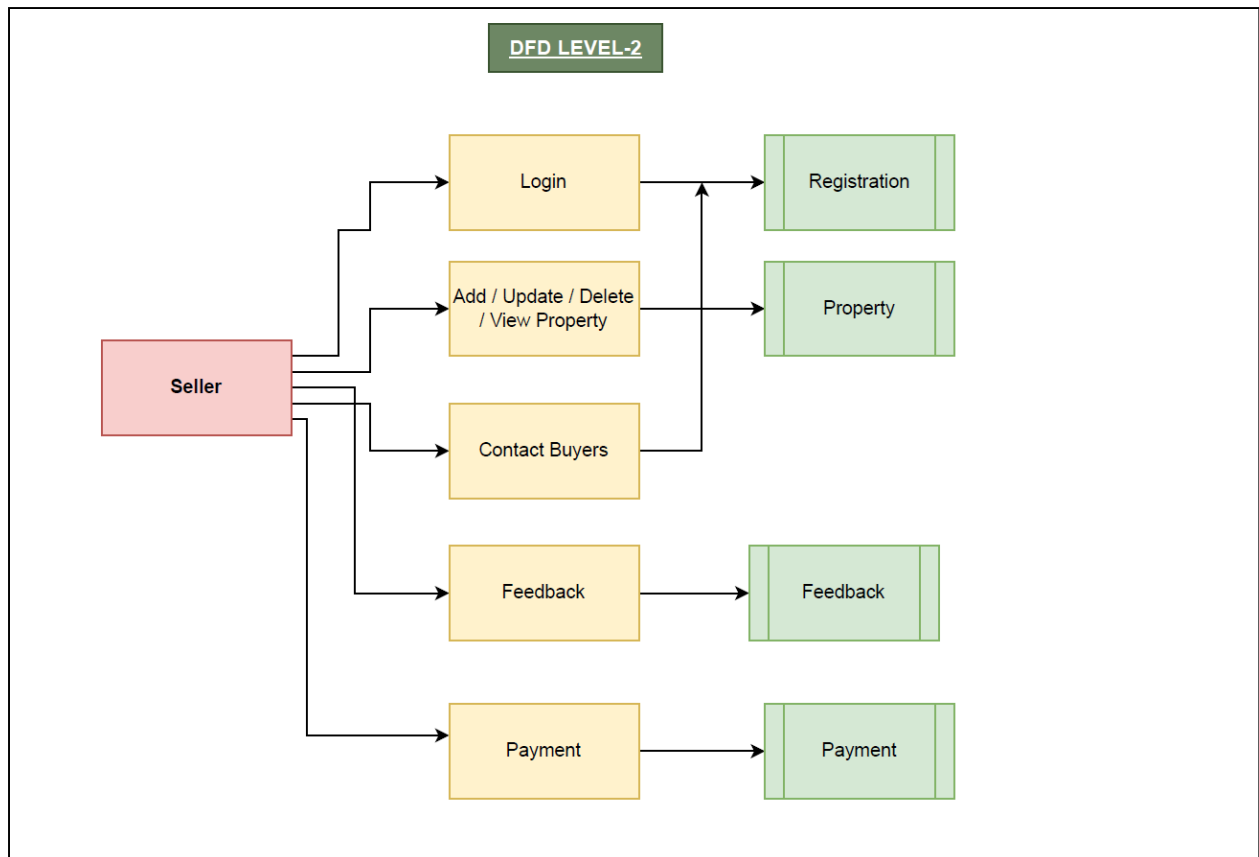
- Main Entities and output of First Level DFD:
- Processing Seller records and generate report of all Seller
- Processing Customer records and generate report of all Customer
- Processing Purchasing records and generate report of all Purchasing
- Processing Payment records and generate report of all Payment

- Processing Login records and generate report of all Login



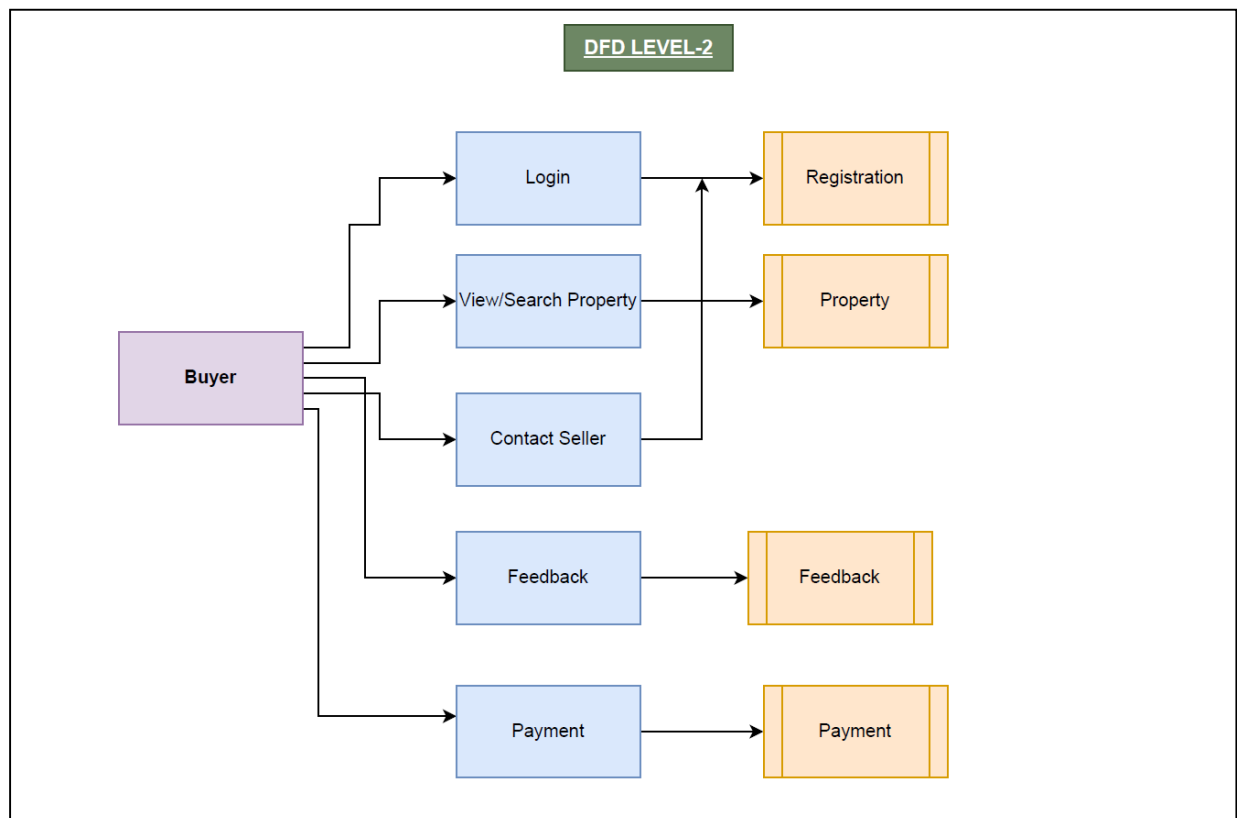
2nd Level Diagram:

- **Low level functionalities of Sender(Real Estate System):**
- Seller logs in to the system and manage all the functionalities.
- Seller can add, edit, delete and view the records of Property, Payment
- Seller can contact all the Customer.
- Seller can also generate report of Customer, Sales, Payment,
- Seller can search apply different level of filters the details of Customer, Payment, Property.



- **Low level functionalities of Buyer (Real Estate System):**

- Buyer logs in to the system and manage all the functionalities.
- Buyer can delete, view the records of Property and Payment.
- Buyer can contact all the Customer.
- Buyer can also generate report of Property and Payment.
- Buyer can search apply different level of filters the details of Seller, Payment, Property.



❖ **Data Dictionary :**

Table 1: User

Will be used as Registration table, Login table and Contact-us table.

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
user_mail	Varchar	primarykey	It stores e-mail id of user
user_pswd	Varchar		It stores e-mail id of user
user_name	Varchar		It stores name of user
user_mobile	Number	uniquekey	It stores user's mobile number

user_add	Varchar		It stores address of user
User_bio	Varchar		It Stores a small bio of user

Table 2: Admin

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
admin_mail	Varchar	uniquekey	It stores e-mail id of admin
admin_pswd	Varchar		It stores e-mail id of admin
admin_name	Varchar		It stores name of admin
admin_id	Number	primarykey	It stores user's mobile number

Table 3: Property

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
Property_image	Blob		It stores photos of Property
Property_id	Varchar	primarykey	It stores id of Property
Property_type	Varchar		It stores type of Property
Property_price	Number		It stores price of Property

Property_place	Number		It stores place of Property
Other_details	Varchar		It stores other details of Property

Table 4: Payment

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
Payment_id	Number	primarykey	It stores payment id
Payment_amt	Varchar		It stores amount
Payment_dt	Date and Time		It stores date and time of payment.
Payment_from	Number	Foreign	It stores user id, who is doing payment of Property
Payment_to	Number	Foreign	It stores user id, to whom payment is done.
Other_details	Varchar		It stores other details of Payment

❖ Structure chart :

