**ACKNOWLEDGEMENT**

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We would like to appreciate Mr. Vinay Pandya for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. We would like to express my gratitude towards our parents & member of SSC for their kind co-operation and encouragement which help us in completion of this project. We would like to express my special gratitude and thanks to all professors for giving us such attention and time.

Our thanks and appreciations also go to our colleague in developing the project and people who have willingly helped us out with their abilities.

Smith Bhojak (156330307008)

Kishan Patel (156330307026)

Bhavin Panchasara (156330307047)

Kandarp Pandya (156330307049)

Pathik Shah (156330307104)

Megh Thakkar (156330307116)

## Table of Contents

ABSTRACT

Chapter 1 Introduction 1

1.1 Need for the New system 2

1.2 Detailed Problem Definition 3

1.3 Viability of the System 3

1.4 Presently Available Systems for the same 4

1.5 Future Prospects 5

Chapter 2 Analysis 6

2.1 Requirement Analysis 6

2.2 Project Model 8

2.3 Schedule Representation 9

2.4 Feasibility Study 10

Chapter 3 Design 12

3.1 Data Flow Diagram 12

3.2 ER-Diagram 22

Chapter 4 System Modeling 27

4.1 Database Design 27

Chapter 5 Technical Specification 36

5.1 Hardware Specification 36

5.1.1 RAM 36

5.1.2 Hard Drive Storage needed 36

5.1.3 Other Hardware requirement 36

5.2 Platform 36

5.2.1 Supported Operating System 36

5.2.2 Programming Server 36

5.2.3 Framework (if any) 36

5.3 Programming Languages used 36

5.3.1 Markup Language 36

5.3.2 Programming Language 36

5.3.3 Scripting Language (If any) 36

5.4 Technical Specification 36

5.4.1 Front-End 36

5.4.2 Back-End 36

5.4.3 IDE 36

5.4.4 UML Tools 36

5.4.5 SRS Tools 36

5.5 Design Layout 37

Conclusion 77

Bibliography 76

## Table INDEX

1 USER\_MASTER 1

2 DEVICE\_MASTER 3

3 OBJECT\_MASTER 17

4 MODEL\_MASTER 17

5 MODEL\_DETAILS 17

6 COLOR\_MASTER 17

7 TYPE\_MASTER 17

8 TEXTURE\_MASTER 17

9 NOTES\_MASTER 17

10 MODEL\_DESIGN 17

**Figure INDEX**

1 Figure 1 About 1

2 Figure 2 About 3

3 Figure 3 About 17

4 Figure 4 About 17

5 Figure 5 About 17

6 Figure 6 About 17

**ABSTRACT**

3D House Modeling in Virtual Reality (VR) is a android application that allows Interior designer to create and design models using objects in 3D space. It is fully dynamic for developing 3D models, user can also modify at any time. It is Portable in the sense that after creating a model it can easily be shared and modified on other devices using the platform. Also the ability to create multiple models and work on them at same time is a feature that is not available on other existing products, and the system provides overall a better user experience and easy usability for the laymen.

**Chapter 1**

**INTRODUCTION**

* 1. **Need for new system**

The need of this system is to allow any interior designer or an architect to create a house model from the interior to the complete exterior. Users can create and design house models in 3D space using Virtual Reality Headset. Through this system user can have enhanced designing experience.

* 1. **Detailed Problem Definition**

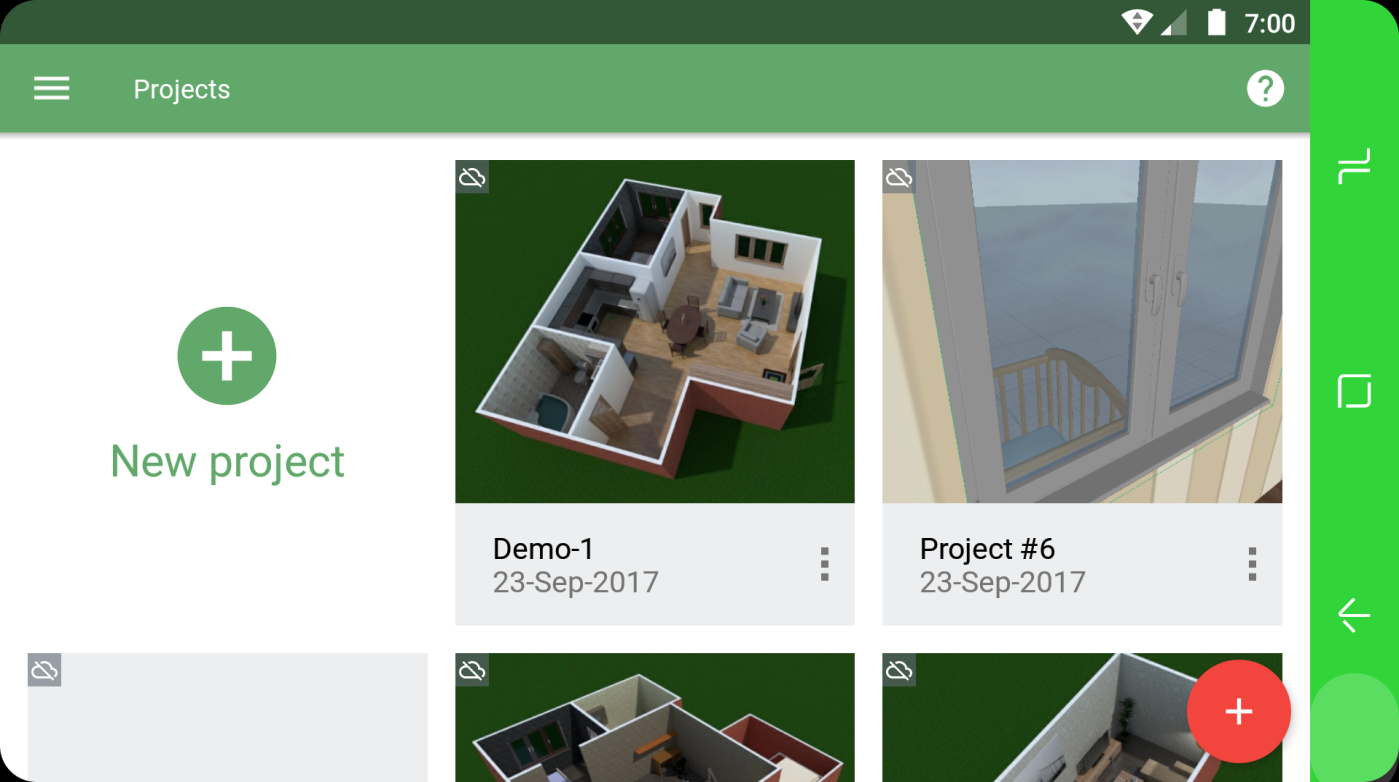
Existing Systems have a Static Architecture, meaning the objects that are created in the environment can only be placed or removed from a specific location in the model, fixed numbers of objects are available and designing of the model is limited. Users’ needs to pay for additional functionalities. Their application lacks a user friendly experience. The cost increases gradually for creating multiple models as newer models are created from scratch and not from reusing resources.

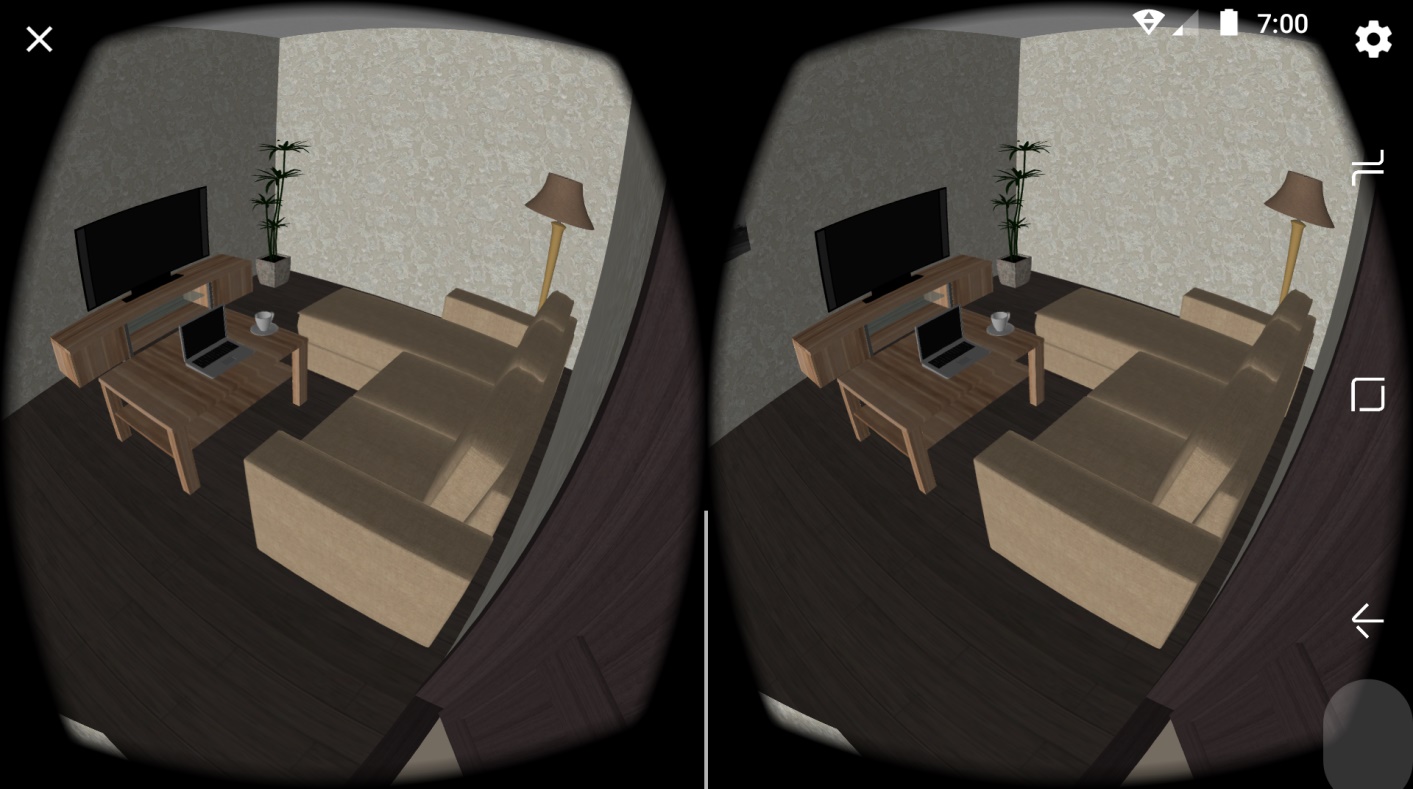
* 1. **Viability of the system**

In this System we are providing Dynamic environment to user to create their particular own design model in it. Objects can be move from one place to another .User can Share their designed models from one system to another and that share model can also be edited .User can work with multiple models at a time .Best user experience so user can use it very easily in it .As it is free so it is affordable in terms of multiple systems development/designing so user do not have to pay to use for multiple systems.

* 1. **Presently Available Systems for the same**

Planner 5D (an android Application) is an existing system which allows a similar functionality.





* 1. **Future Prospects**

The main future prospect is user can use our application to design it will help to design building or skyscraper.

Future updates will allow user more options to create models using different objects in it.

We will provide very simple and easy user interface in it .Will allow to use it at very low costing rates

**Chapter 2**

**ANALYSIS**

**2.1 Requirement Analysis**

In [systems engineering](https://en.wikipedia.org/wiki/Systems_engineering) and [software engineering](https://en.wikipedia.org/wiki/Software_engineering), **requirements analysis** encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting [requirements](https://en.wikipedia.org/wiki/Requirement) of the various [stakeholders](https://en.wikipedia.org/wiki/Stakeholder_(corporate)), *analyzing, documenting, validating and managing* software or system requirements.

**2.2 Project Model**

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software.

***PROTOTYPE model***

**2.3 Schedule Representation**

Program evolution and review techniques (PERT) and critical path method (CPM) are two project scheduling method that can be applied to software development. Both techniques are driven by information already developed in earlier project planning activities:

* Estimate of effort.
* A decomposition of the product function.
* The selection of appropriate process model and task set.
* Decomposition of tasks.

**Table:1 Schedule Representation**

|  |  |  |
| --- | --- | --- |
| **ACTIVITY** | START DATE | FINISH DATE |
| Requirement Analysis |  |  |
| System Analysis |  |  |
| System Design |  |  |
| System Coding |  |  |
| Testing and Integration |  |  |

**2.4 Feasibility Study:**

The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system.

**Chapter 3**

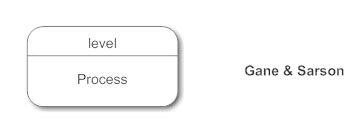
**DESIGN**

**3.1 Data flow diagram**

A data flow diagram (DFD) is a graphical representation of the "flow" of data such as customer names and transaction details through an [information system](https://en.wikipedia.org/wiki/Information_system) like the point of sale (POS) software that would be installed on a register. DFDs can also be used for the visualization [data processing](https://en.wikipedia.org/wiki/Data_processing) (structured design).

**3.1.1 Symbols for Data Flow Diagram**

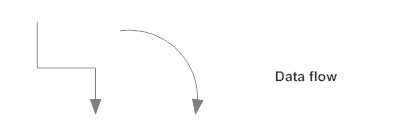
1) **Process Notations:** A process transforms incoming data flow into outgoing data flow.



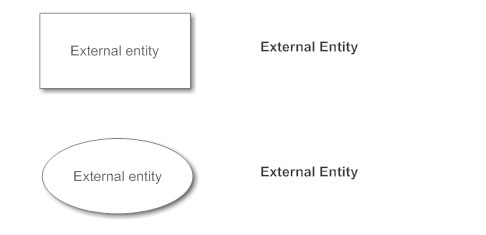
2) **Data store Notations.** Data stores are repositories of data in the system. They are sometimes also referred to as files.

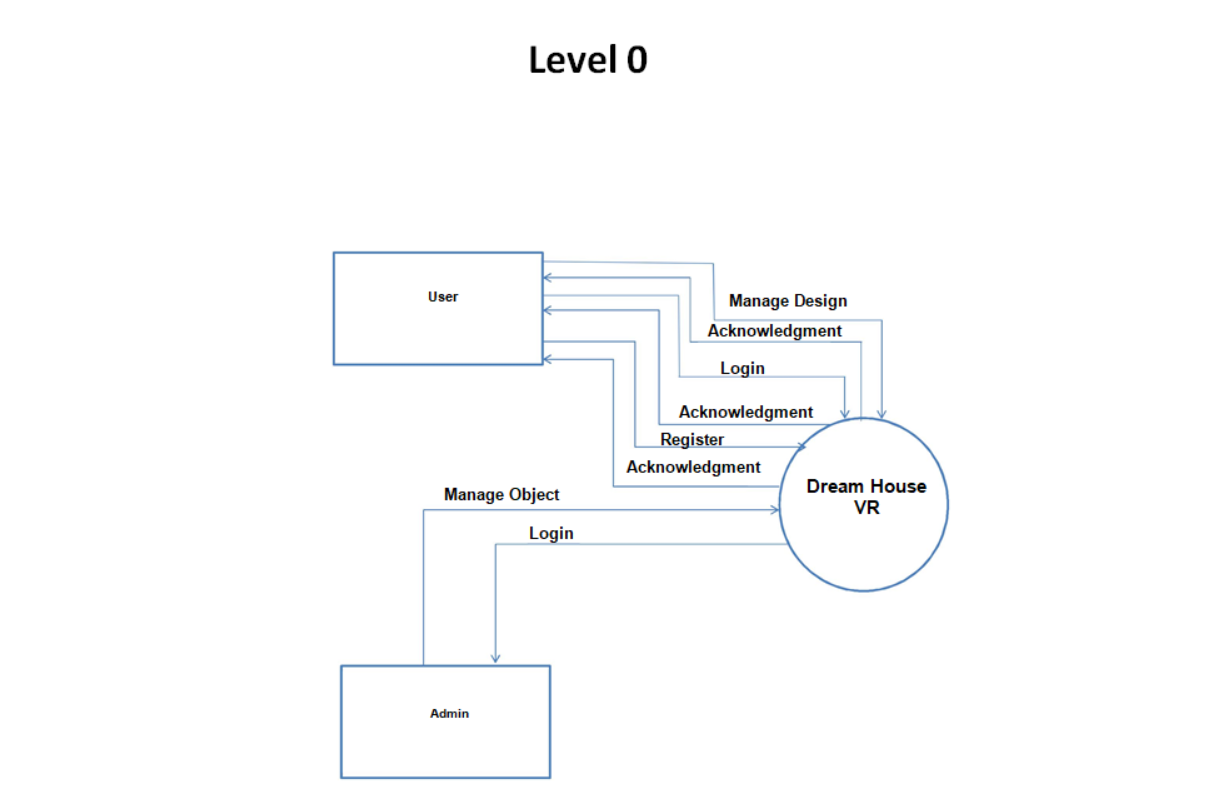


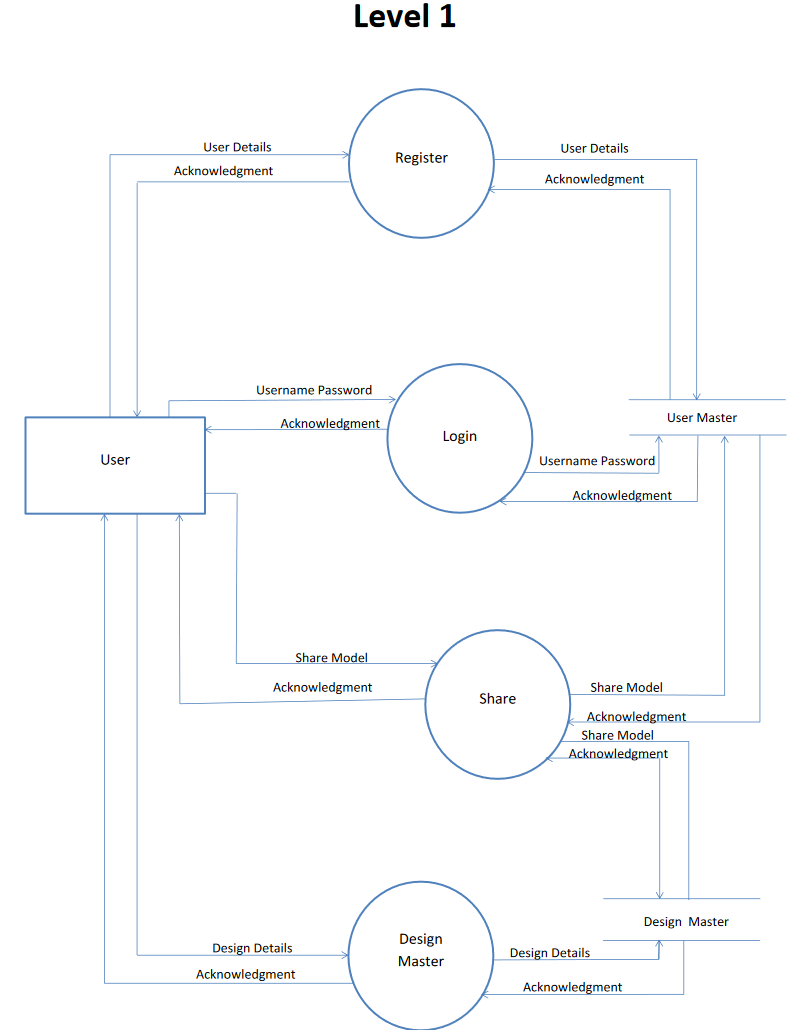
3) **Dataflow Notations .**Dataflow is pipeline through which packets of information flow. Label the arrows with the name of the data that moves through it

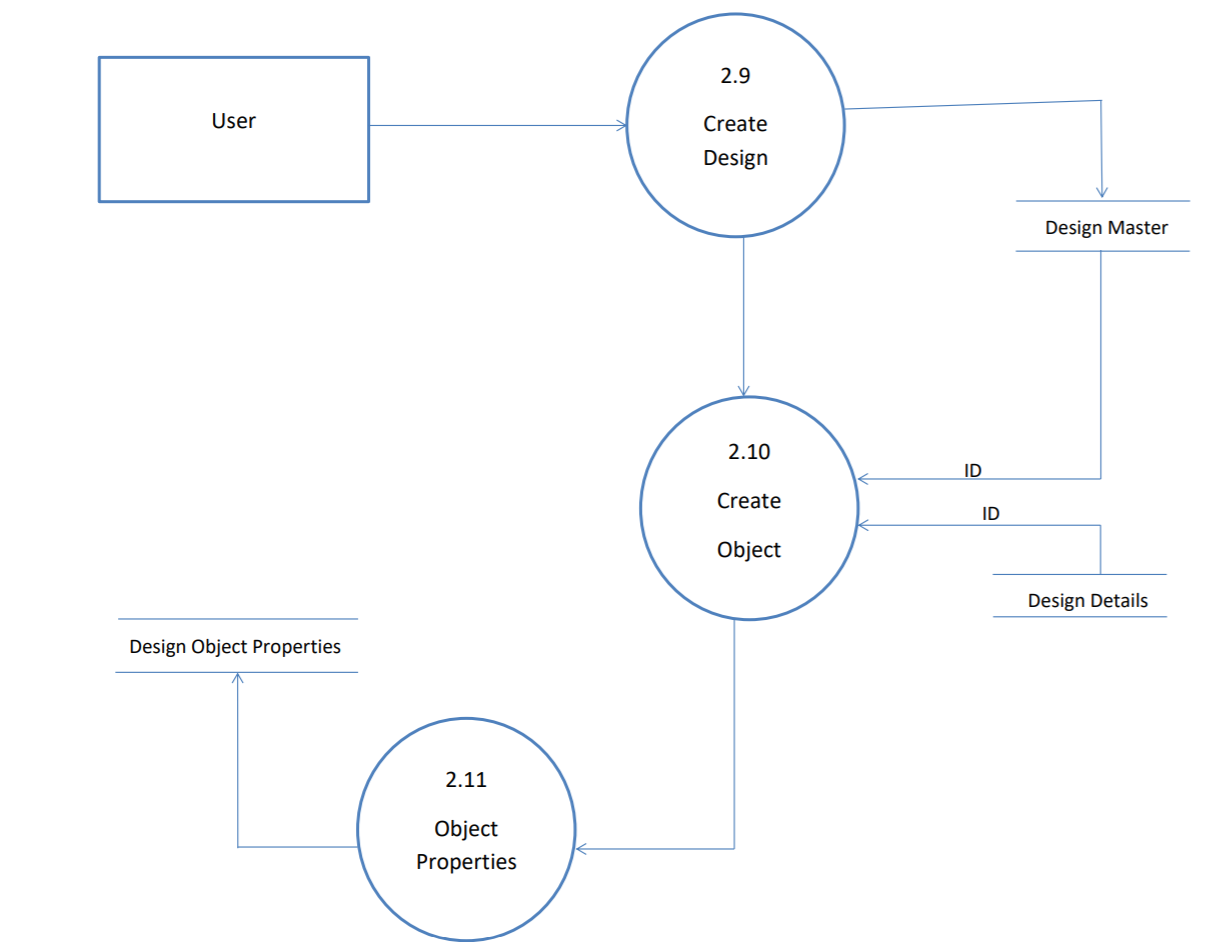
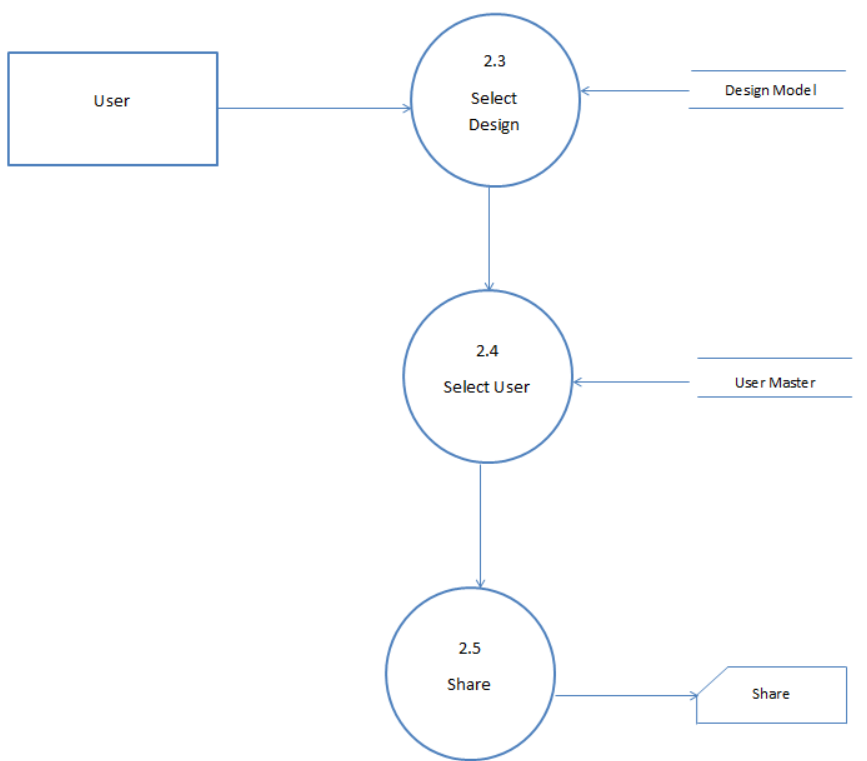
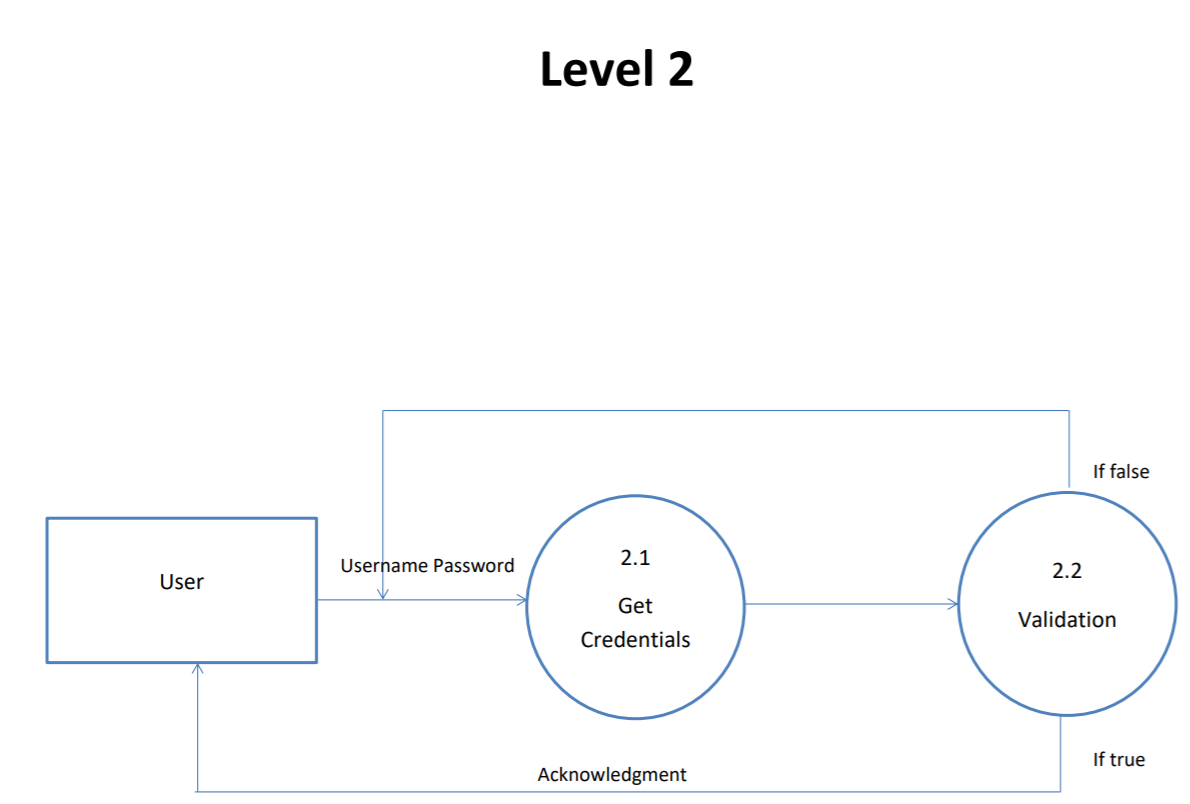


4) **External Entity Notations.** External entities are objects outside the system, with which the system communicates. External entities are sources and destinations of the system's inputs and outputs.









**3.2 Entity Relationship Diagram**

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.

At first glance an entity relationship diagram looks very much like a [flowchart](https://www.smartdraw.com/flowchart/). It is the specialized symbols, and the meanings of those symbols, that make it unique.

**3.2.1 Symbols**

**Entities**: which are represented by rectangles .An entity is an object or concept about which you want to store information



A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.



**Actions**: which are represented by diamond shapes, show how two entities share information in the database.



**Attributes**: which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.



A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values.

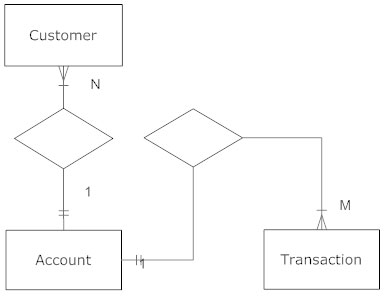


A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.



**Connecting lines**: solid lines that connect attributes to show the relationships of entities in the diagram.

**Cardinality:** specifies how many instances of an entity relate to one instance of another entity. Ordinality is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinality describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and ordinality specifies the absolute minimum number of relationships.



There are many notation styles that express cardinality.  
**Information Engineering Style**





**Chapter 4**

**SYSTEM MODELING**

**4.1 Database Design**

**Database design** is the process of producing a detailed [data model](https://en.wikipedia.org/wiki/Data_model) of [database](https://en.wikipedia.org/wiki/Database). This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a [data definition language](https://en.wikipedia.org/wiki/Data_definition_language), which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. The term database design can be used to describe many different parts of the design of an overall [database system](https://en.wikipedia.org/wiki/Database_system). Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data.

**USER\_MASTER**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| User\_Id | Number(4) | Primary Key | User\_id must be required |
| U\_name | Varchar2(20) | Not null | User name is required |
| Password | Varchar2(10) | Not null | Login password is must be required |
| Firstname | Char(10) | Not null | Firstname of user |
| Lastname | Char(10) | Not null | Lastname of user |
| Email\_id | Varchar2(40) | Unique | Email\_id of User |
| Gender | Char(6) |  | Gender of user |
| Mobile\_no | Number(10) | Unique | Contact Number of User |
| Device\_id | Number(6) | Foreign key | Device\_id from DEVICE\_MASTER table |

**DEVICE\_MASTER**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| Device\_id | Number(6) | Primary key | Device\_id must be required |
| Device\_brand | Char(20) |  | Brand name of device |
| Device\_type | Varchar2(10) | Not null | Device type of user |
| Device\_model | Varchar2(10) |  | Model no. of device |
| Device\_iemi\_no | Varchar2(10) |  | IEMI no. of device |
| Internet\_type | Varchar2(10) |  | Type of internet used in device |
| Device\_OS | Char(30) | Not null | OS used in device |

**OBJECT\_MASTER**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| Obj\_Id | Number(6) | Primary key | Obj\_Id must be required |
| Obj\_name | Varchar2(10) | Not null | Name of the object |
| Img\_path | Varchar2(100) | Not null | Path of the image saved |
| Obj\_Height | Number(4) | Not null | Height of the object |
| Obj\_Width | Number(4) | Not null | Width of the object |
| Obj\_depth | Number(4) | Not null | Depth of the object |
| Obj\_remark | Text |  | Remarks |

**MODEL\_MASTER**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| ID | Number(6) | Primary key | Model\_id must be required |
| User\_Id | Number(6) | Foreign key | Id of user from user\_master table |
| Created\_on | Date |  | Date of creation |
| Created\_by | NUMBER(6) | Foreign Key | ID FROM USER\_MASTER TABLE |
| Modify\_on | Date |  | Date of modification |
| Modify\_by | NUMBER(6) | Foreign Key | ID FROM USER\_MASTER TABLE |
| Mod\_Name | Char(10) | Primary Key | Name of the model |
| Mod\_description | Text |  | Description given of model |

**MODEL\_DETAILS**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| ID | NUMBER(6) | Primary Key | ID MUST BE REQUIRED |
| OBJ\_ID | NUMBER(6) | Foreign Key | ID of object\_ master table |
| Mod\_Id | Number(6) | Foreign key | Id of model from MODEL\_MASTER table |
| Pos\_x | Number(3) | Not null | X-axis of model |
| Pos\_y | Number(3) | Not null | y-axis of model |
| Pos\_z | Number(3) | Not null | z-axis of model |
| Height\_y | Number(3) | Not null | Height of y-axis |
| Width\_x | Number(3) | Not null | Width of x-axis |
| Depth\_z | Number(3) | Not null | Depth of z-axis |
| Mod\_Opacity | Number(3) | Not null | Transparency of model(0 OR 1) |

**COLOR\_MASTER**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| Col\_Id | Number(6) | Primary Key | Id of the color must be required |
| Col\_Name | Char(10) | Not null | Name of color |
| Hex\_code | Varchar2(6) | Not null | Hexadecimal code for color |

**TYPE\_MASTER**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| Type\_Id | Number(6) | Primary Key | Id must be required |
| Type\_Name | Char(10) | Not null | Type of object |
| Type\_remark | Char(500) |  | Remarks |

**PATTERN\_MASTER**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| Patn\_Id | Number(6) | Primary Key | Pattern id must be required |
| Patn\_Name | Char(10) | Not null | Pattern name |
| Patn\_Img | Varchar2(100) | Not null | Image pattern path |

**TEXTURE\_MASTER**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| Tex\_Id | Number(6) | Primary Key | Texture id must be required |
| Tex\_Name | Char(10) | Not null | Texture name |
| Tex\_Img | Varchar2(100) | Not null | Texture image path |

**NOTES\_DES**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| Note\_Id | Number(6) | Primary Key | Id must be required |
| Note\_Des | Char(40) |  | Description of the notes |
| Mod\_Detail\_Id | Number(6) | Foreign Key | Id of model from table MODEL\_DETAILS |

**MODEL\_DESIGN**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD** | **DATATYPE** | **CONSTRAINTS** | **DESCRIPTION** |
| ID | NUMBER(6) | Primary Key | ID MUST BE REQUIRED. |
| Design\_Id | Number(6) | Unique | Id must be required |
| Mod\_Detail\_Id | Number(6) | Foreign Key | Id of model from table MODEL\_DETAILS |
| Side\_No | Number(2) | Primary Key | Different sides of object |
| Col\_Id | Number(6) | Foreign Key | Id from table COLOR\_MASTER |
| Pat\_Id | Number(6) | Foreign Key | Id from table  PATTERN\_MASTER |
| Tex\_Id | Number(6) | Foreign Key | Id from table TEXTURE\_MASTER |
| Type\_Id | Number(6) | Foreign Key | Id of type from table TYPE\_MASTER  TYPE\_ID |

**Chapter 5**

**TECHNICAL SPECIFICATION**

**5.1 Hardware Specification**

**5.1.1 RAM**

-Minimum 2GB is required.

**5.1.2 Hard Drive Storage Required**

-Minimum 100 MB required.

**5.1.3 Other Hardware requirement**

-Virtual Reality headset.

**5.2 Platform**

**5.2.1 Supported Operating System**

-Minimum OS Android 5.1 LOLLIPOP.

**5.2.2 Programming Server**

-Java socket.

**5.2.3 Framework (if any)**

-None

**5.3 Programming Languages used**

**5.3.1 Markup Language**

-eXtensible Markup Language (XML).

**5.3.2 Programming Language**

-Java.

**5.3.3 Scripting Language (If any)**

-None.

**5.4 Technical Specification**

**5.4.1 Front-End**

-Android Studio.

**5.4.2 Back-End**

-Not Decided.

**5.4.3 IDE**

-Android Studio.

**5.4.4 UML Tools**

-Edrawmax

**5.4.5 SRS Tools**

**CONCLUSION:**

## BIBLIOGRAPHY

[1] “Entity-Relationship Model Concepts”,

[*https://www.tutorialspoint.com/dbms/er\_model\_basic\_concepts.htm*](https://www.tutorialspoint.com/dbms/er_model_basic_concepts.htm)

[2] “Entity-Relationship Model Representation”,

[*https://www.tutorialspoint.com/dbms/er\_diagram\_representation.htm*](https://www.tutorialspoint.com/dbms/er_diagram_representation.htm)*,* [*https://www.smartdraw.com/entity-relationship-diagram*](https://www.smartdraw.com/entity-relationship-diagram)

[3] “Data Flow Diagrams”,

[*https://www.lucidchart.com/pages/data-flow-diagram/c?dfd=1*](https://www.lucidchart.com/pages/data-flow-diagram/c?dfd=1) *,*

[*https://www.smartdraw.com/data-flow-diagram*](https://www.smartdraw.com/data-flow-diagram)*,* [*https://www.tutorialspoint.com/programming\_methodologies\_training/data\_flow\_diagram.asp*](https://www.tutorialspoint.com/programming_methodologies_training/data_flow_diagram.asp)

[4] *Kathy Sierra and Bert Bates* “Working with Strings and Arrays”, *ORACLE OCA/OCP Java SE 7 Programmer I & II Study Guide, IZO-803 and IZO-804*

[5] “Java Multithreading”, [*https://www.javatpoint.com/multithreading-in-java*](https://www.javatpoint.com/multithreading-in-java)

[6] “Java Exception Handling”, [*https://www.javatpoint.com/exception-handling-in-java*](https://www.javatpoint.com/exception-handling-in-java)

[7] “Android Application Components”, [*https://www.tutorialspoint.com/android/android\_application\_components.htm*](https://www.tutorialspoint.com/android/android_application_components.htm)

[8] “Android Resources”,[*https://www.tutorialspoint.com/android/android\_resources.htm*](https://www.tutorialspoint.com/android/android_resources.htm)

[9] “Android Activities”,[*https://www.tutorialspoint.com/android/android\_acitivities.htm*](https://www.tutorialspoint.com/android/android_acitivities.htm)

[10] “Android Activity Life-Cycle”,

[*https://www.javatpoint.com/android-life-cycle-of-activity*](https://www.javatpoint.com/android-life-cycle-of-activity)

[11] “Android Activity States”, [*https://www.youtube.com/watch?v=NMDPxN8FgXM&list=PL6gx4Cwl9DGBsvRxJJOzG4r4k\_zLKrnxl&index=9*](https://www.youtube.com/watch?v=NMDPxN8FgXM&list=PL6gx4Cwl9DGBsvRxJJOzG4r4k_zLKrnxl&index=9)

[12] “Android Fragments”,[*https://www.tutorialspoint.com/android/android\_fragments.htm*](https://www.tutorialspoint.com/android/android_fragments.htm) *,* [*https://www.javatpoint.com/android-fragments*](https://www.javatpoint.com/android-fragments)

[13] “Android Widgets”, [*https://www.javatpoint.com/android-ui-widgets-tutorial*](https://www.javatpoint.com/android-ui-widgets-tutorial)

[14] “Android Buttons”, [*https://www.javatpoint.com/android-working-with-button*](https://www.javatpoint.com/android-working-with-button)

[15] “Android Toast”, [*https://www.javatpoint.com/android-toast-example*](https://www.javatpoint.com/android-toast-example)

[16] “Android Text View”,

[*https://www.javatpoint.com/android-autocompletetextview-example*](https://www.javatpoint.com/android-autocompletetextview-example)