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

Table INDEX

Figure INDEX

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

1.1 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.

1.2 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.

1.3 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.

1.4 Presently Available Systems for the same

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

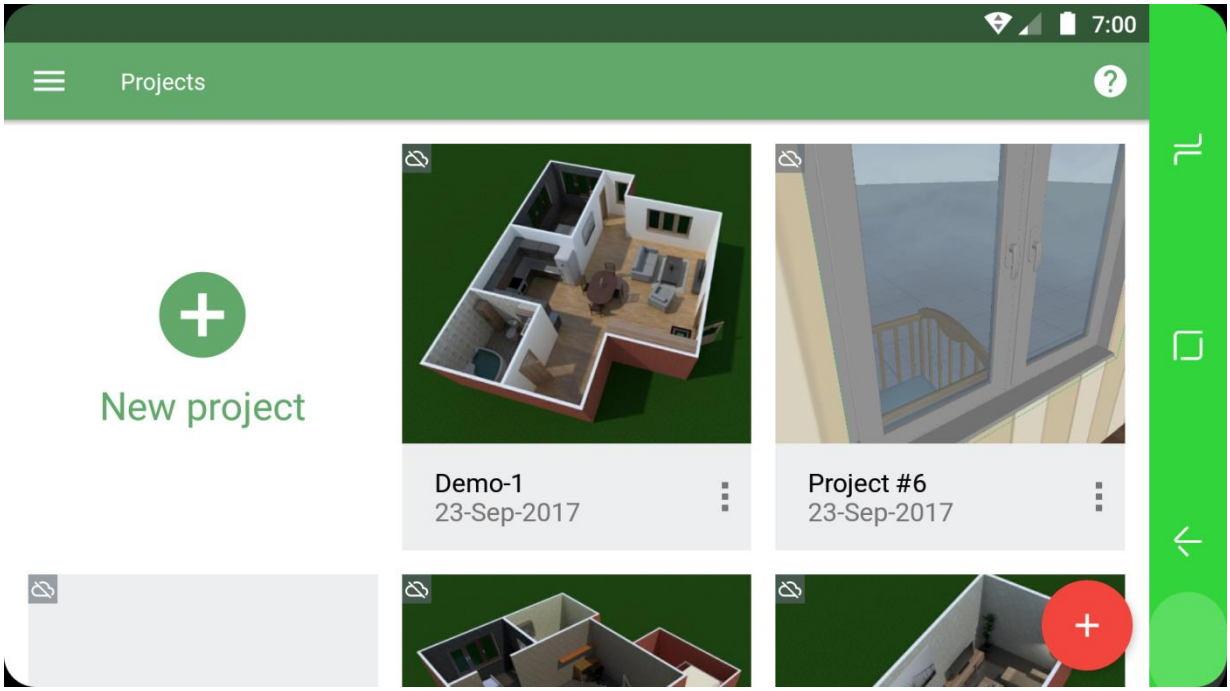


FIG: PRESENTLY AVAILABLE SYSTEM

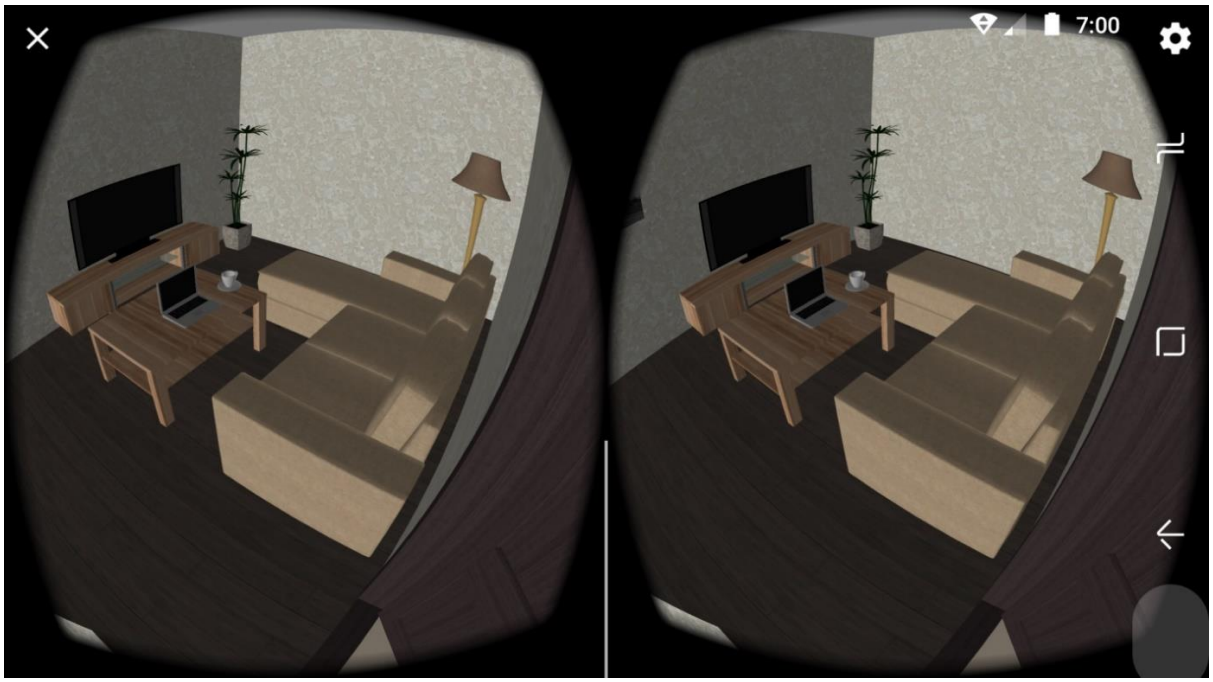


FIG: PRESENTLY AVAILABLE SYSTEM IN VR

1.5 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 and 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 of the various stakeholders, *analyzing, documenting, validating and managing* software or system requirements.

There is no such existing platform that provides exact functionalities which allows user to immerse into system for enhanced designing.

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.

We will be using prototype model for the development of this system.

In this model we will be creating the prototype of the system before creating the actual system.

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.

2.4.1 Technical Feasibility:

The technical issue usually raised during the feasibility stage of the investigation includes the following:

Does the necessary technology exist to do what is suggested?

- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

2.4.2 Operational Feasibility:

Proposed projects are beneficial only if they can be turned out into information system. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following:

- Is there sufficient support for the management from the users?
    - Will the system be used and work properly if it is being developed and implemented?
    - Will there be any resistance from the user that will undermine the possible application benefits

2.4.3 Economical Feasibility:

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. There is nominal expenditure and economic feasibility for certain**.**

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 like the point of sale (POS) software that would be installed on a register. DFDs can also be used for the visualization 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.

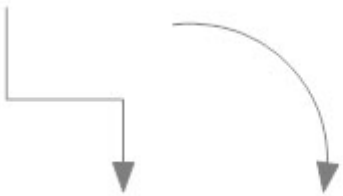
Gane & Sarson

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

Yourdon & Coad

Gane & Sarson

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



Data flow

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.

External Entity

External Entity

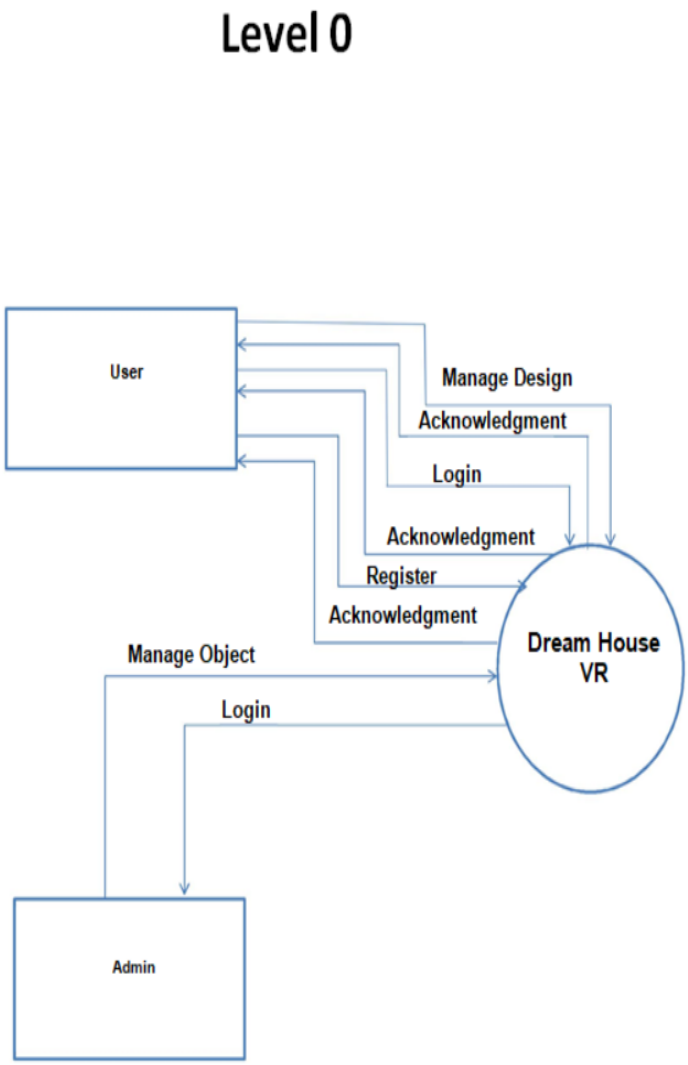


FIG: LEVEL 0 DATA FLOW DIAGRAM

Level 1

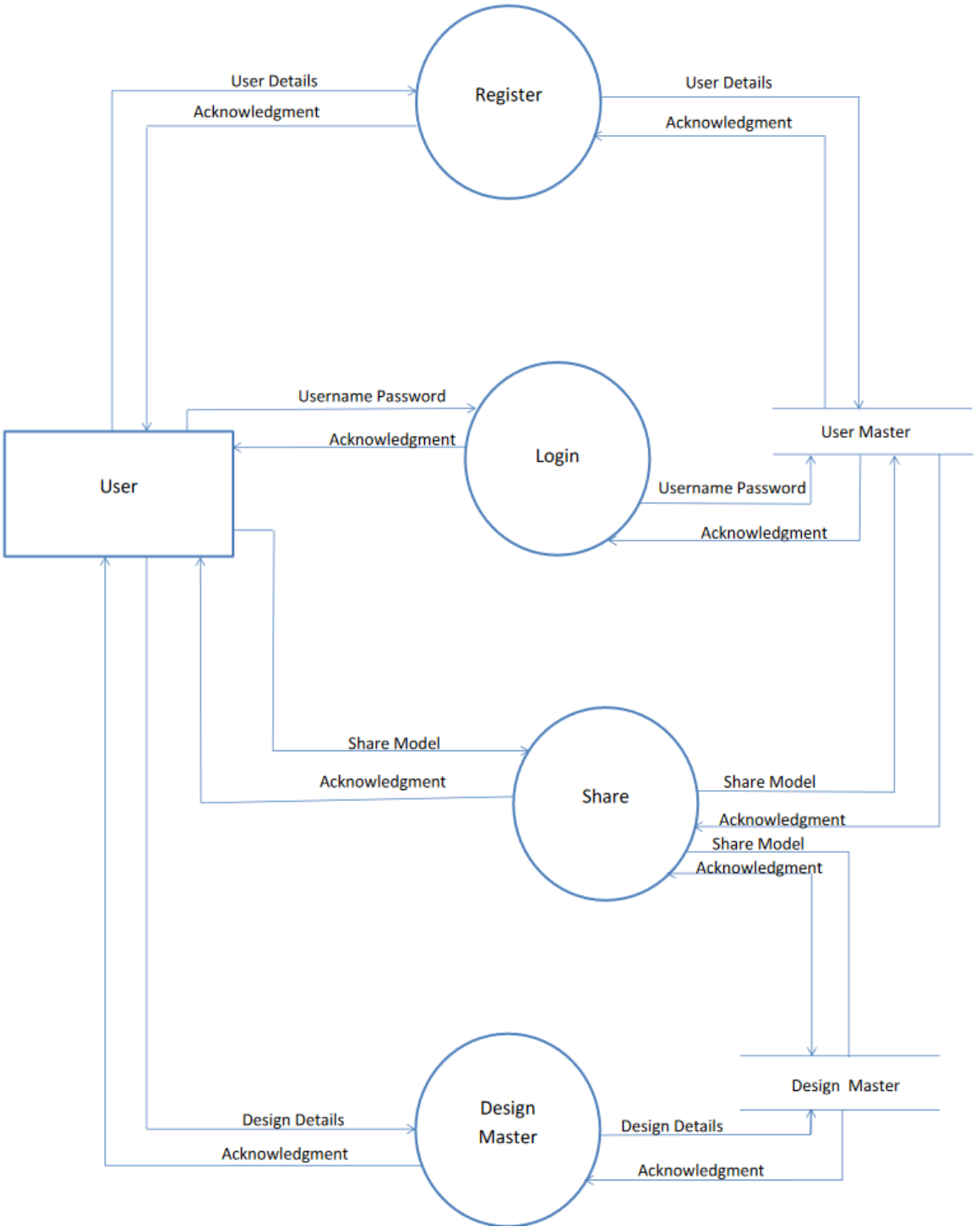


FIG: LEVEL 1 DATA FLOW DIAGRAM

Level 2

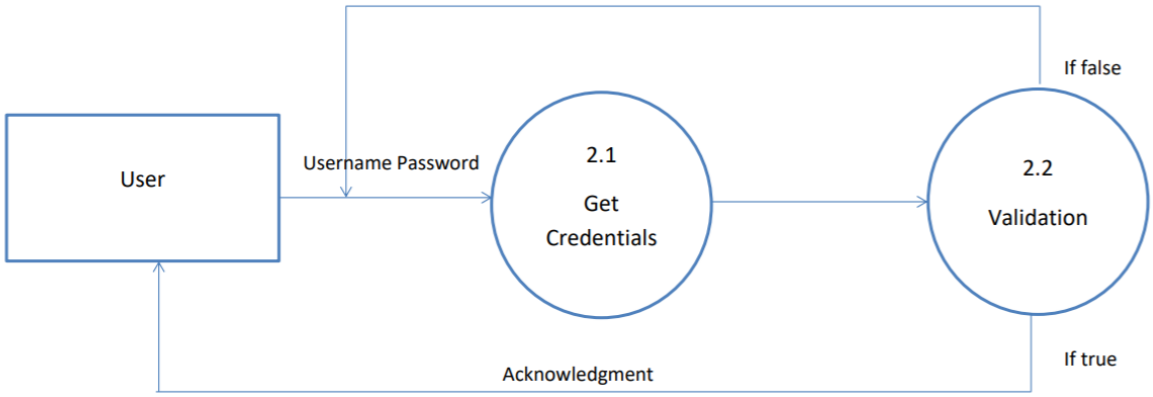


FIG: LEVEL 2 DATA FLOW DIAGRAM

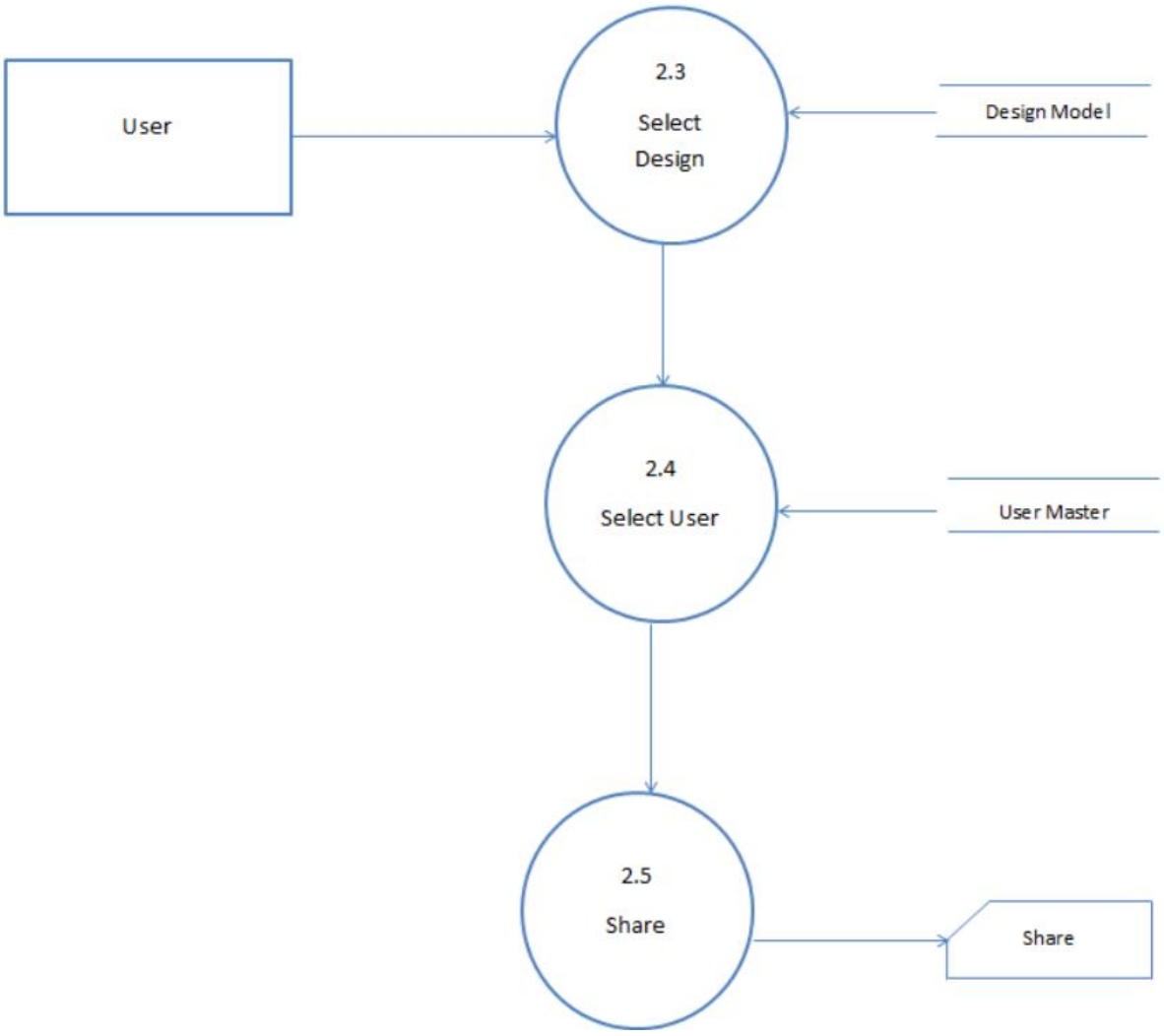


FIG: LEVEL 2 DATA FLOW DIAGRAM

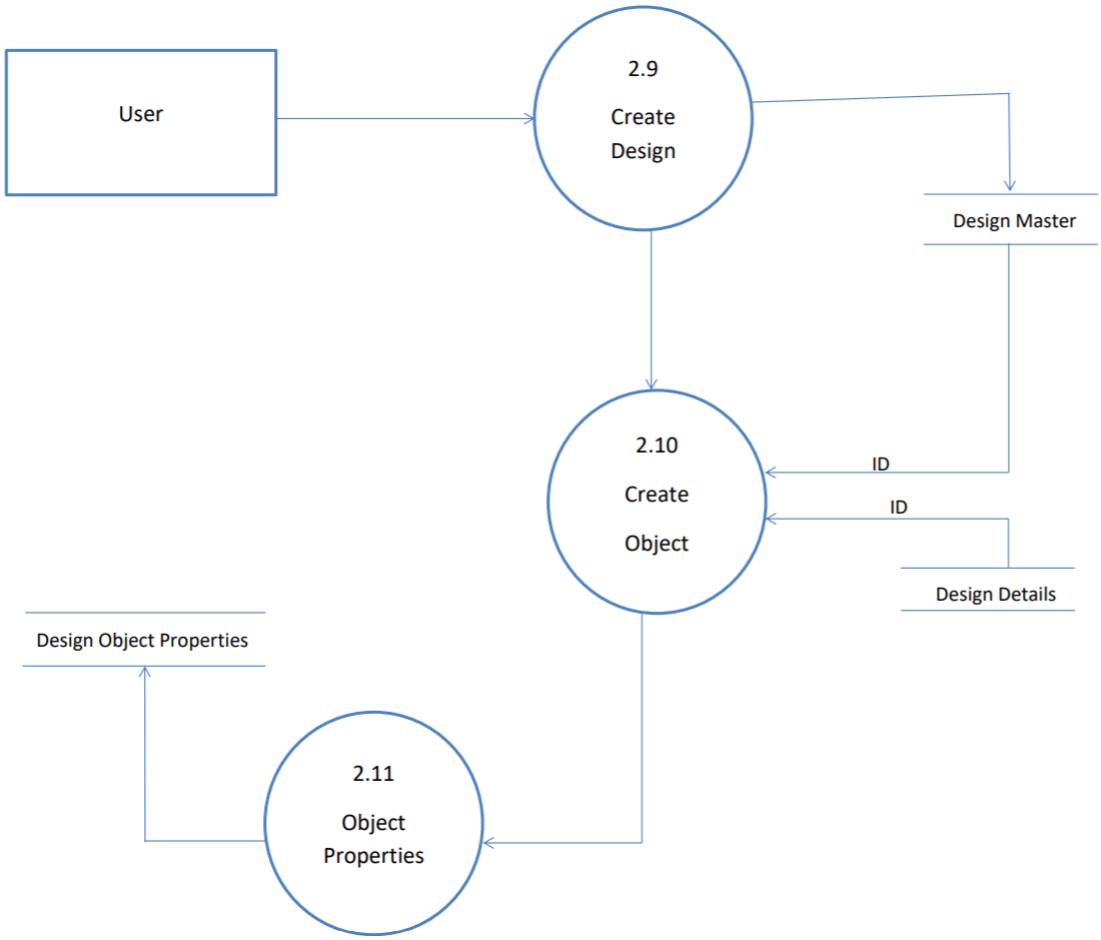


FIG: LEVEL 2 DATA FLOW DIAGRAM

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. 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.

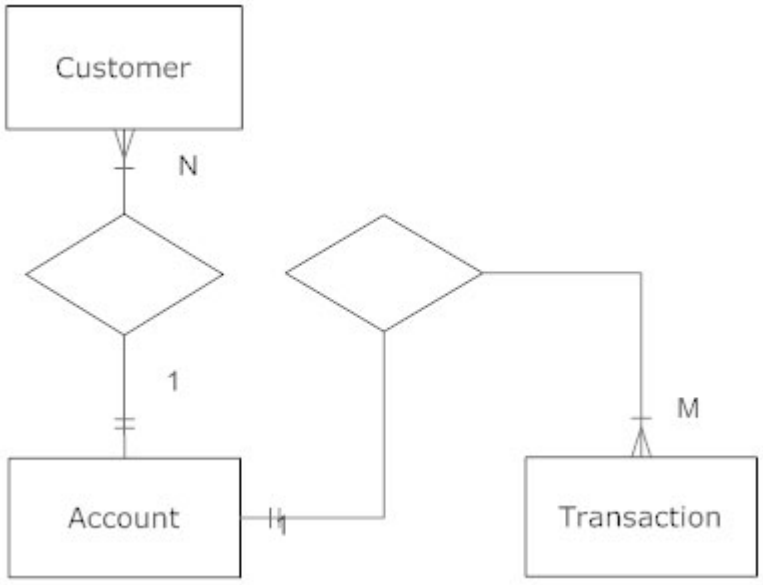


FIG: EXAMPLE OF RELATIONSHIPS

There are many notation styles that express cardinality.

Information Engineering Style

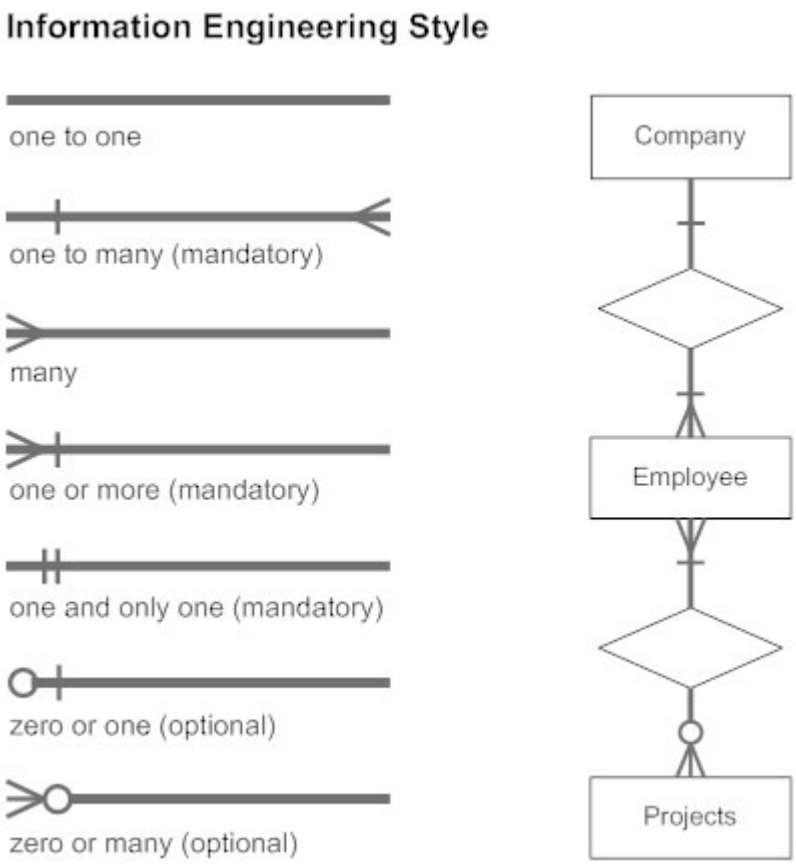


FIG: RELATIONSHIPS

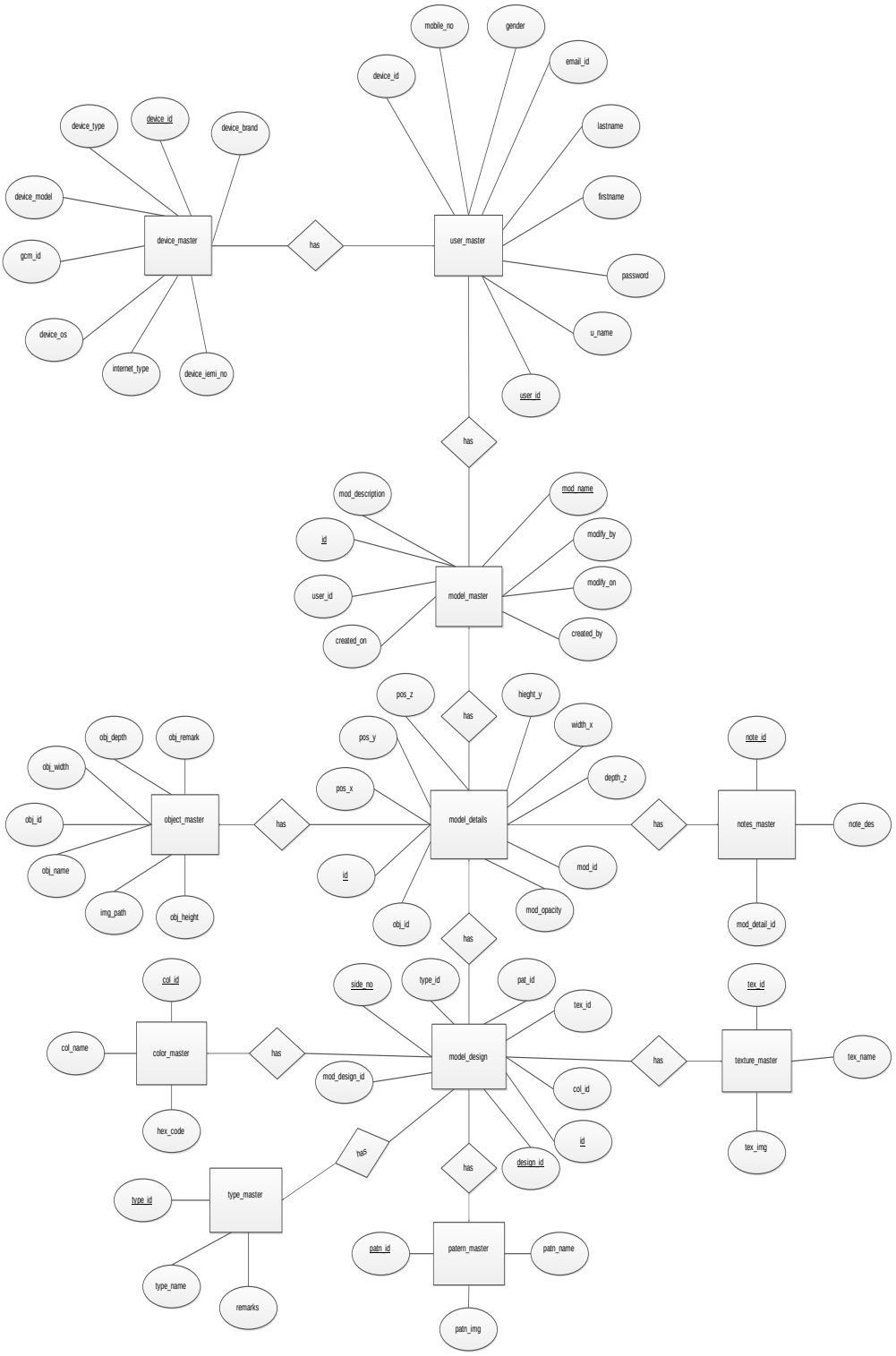


FIG: ENTITY-RELATIONSHIP DIAGRAM

Chapter 4

SYSTEM MODELING

4.1 Database Design

Database design is the process of producing a detailed data model of 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, 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. 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

-SQLite

5.4.3 IDE

-Android Studio.

5.4.4 UML Tools

-Edrawmax

5.4.5 SRS Tools

CONCLUSION:

From this project we learnt about how to create database and we also learnt about various diagrams like data flow diagram and entity relationship diagram. Also we learnt about the programming language core java.

BIBLIOGRAPHY

[1] “Entity-Relationship Model Concepts”,

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.smartdraw.com/entity-relationship-diagram*

[3] “Data Flow Diagrams”,

https://www.lucidchart.com/pages/data-flow-diagram/c?dfd=1 ,

https://www.smartdraw.com/data-flow-diagram, *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*

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

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

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

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

[10] “Android Activity Life-Cycle”, *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*

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

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

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

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

[16] “Android Text View”,

https://www.javatpoint.com/android-autocompletetextview-example