

TABLE OF CONTENTS

1. USAGE SCENARIO	3
2. SCENARIO BASED MODELING	6
4.1 INTRODUCTION	6
4.2 DEFINITION OF USE CASE	6
4.3 USE CASE DIAGRAMS	7
2.3.1 LEVEL-0	7
2.3.2 LEVEL-1	8
2.3.3 LEVEL-1.1	9
2.3.4 LEVEL-1.1.2	11
2.3.5 LEVEL-1.2	12
2.3.6 LEVEL-1.2.2	14
2.3.7 LEVEL 1.2.3	15
2.3.8 LEVEL 1.2.5	16
2.3.8 LEVEL 1.2.7	17
3. ACTIVITY DIAGRAM	18
4. SWIMLANE DIAGRAM	24
5. DATA MODELLING	32
5.1 Data Modelling Concept	32
5.1.1 Data Objects	32
5.1.2 NOUN IDENTIFICATION	32
5.1.3 Potential Data objects:	34
5.1.4 Analysis for finalizing Data Objects	34
5.1.5 Final Data Objects	35
5.2 DATA OBJECT RELATIONSHIPS	35
5.3 ENTITY RELATIONSHIP DIAGRAM	36
5.4 SCHEMA DIAGRAM	36
6. CLASS-BASED MODELING	38
6.1 CLASS BASED MODELING CONCEPT	38
6.2 Noun List	39
6.3 GENERAL CLASSIFICATION	39
Table:1 General Classification	40
6.4 SELECTION CRITERIA	42

Table:2 Selection Criteria	42
6.5 ATTRIBUTE SELECTION	43
Table:3 Attribute Selection	43
6.6 METHOD IDENTIFICATION	44
Table:4 Method Attributes	44
6.7 Analyzing Classes	47
6.8 Class Card	47
Table:5 Class Card for Furniture	47
Table: 6 Class Card for User	48
Table:7 Class Card for Workspace	49
Table:8 Class Card for System	50
Table: 9 Class Card for Database	51
6.9 CRC Diagram	51
7. BEHAVIORAL MODELING	52
7.1 State Diagrams	54
7.2 SEQUENCE DIAGRAM:	57

1. USAGE SCENARIO

Furniture placement in the room can be a very confusing task. For these reasons we intended to build an Augmented reality-based Android application. Our application will provide users the following options:

- ❖ Users can create his/her own account
- ❖ Can select the type of the place(Home, Office) to be decorated
- ❖ The option to drag and drop augmented furniture in their room which they are able to see using their phone's camera.
- ❖ can change the position of the furniture
- ❖ can change the size of the furniture
- ❖ can change the color of the furniture
- ❖ can change the color of the walls and the floor
- ❖ can add customize furniture and
- ❖ can save unfinished work and finish it later.

Authentication:

- At the very beginning, a user will be allowed to create an account or log in.
- If the user has no account, he can create an account. He has to provide a valid email address, username and password.
- The system will check the email and username. If the email or the username already exists, then the system will request the user to re-enter email or username else an OTP will be sent to the given email. Users have to give the code to the system to verify the account. After verification the account will be stored into the database and a notification will be sent to the user's email.
- If s/he is an authorized user then s/he can directly login providing username and password. If the user forgets the password to log in, then he will get a chance to reset the password via email.

Choose Place Type:

After logging into the system, the user will be able to choose which kind of place he wants to design. He can choose House-hold space or Office space for placing furniture.

Drag Augmented Furniture:

There will be some fixed augmented furniture(**Sofa, Chair, Bed**, Almira, Dress Hanger, **Wardrobe, Showcase**, Dressing Table, **Table, Shoe rack**, Refrigerator, Bookshelf, **Desk, Television , Painting**) related to the place user selected in a sidebar of the phone. From there users can drag any furniture and drop on the ground on phones' cameras.

Transposition:

As furniture placement in a room is very confusing, sometimes the user might have to change the position of the furniture. To change position, users have to touch the furniture and have to drop it in the place where they want to place it.

Scalability:

There are rooms and furniture of different sizes, every furniture doesn't suite in every room or more specifically in every side of the room. So it is necessary to have an option of resizing the furniture according to the users' needs. To change the default size of a furniture, users have to touch the furniture with more than one finger on the touch screen and then have to stretch fingers.

Color Modification:

Color choice varies from person to person. It is very much possible that one won't like the default furniture color. If the user doesn't like the color of the furniture he can change it. He can also change the color of the walls and floor.

Add Customize Furniture:

There will be some fixed augmented furniture in the task bar. If the user wants to design rooms with furniture that doesn't exist in the task bar, he can make his own customized furniture for the place and drag the furniture. For making a customized furniture the user has to press a button and he will be redirected to unity asset store and then he needs to give a proper name of the furniture he is looking for and he should download it. After that the furniture will be added in the task bar and it will be stored in his mobile storage.

Save and Load:

Every person is busy in this world. So, if the user doesn't get sufficient time to decorate his place at a time, he can save his current status of his work in the database and can finish it later in his free time. If a user wants to save a state of his work the system will create a log file first then it will convert it into binary file and will save it into database. While loading the system will first collect the binary file then decrypt into a log file and will load into current workspace.

Device Independency :

2.SCENARIO BASED MODELING

This chapter describes the Scenario Based Model for the “InDez”.

2.1 INTRODUCTION

Although the success of a computer-based system or product is measured in many ways, user satisfaction resides at the top of the list. If we understand how end users (and other actors) want to interact with a system, our software team will be better able to properly characterize requirements and build meaningful analysis and design models. Hence, requirements modeling begins with the creation of scenarios in the form of Use Cases, activity diagrams and swim lane diagrams.

2.2 DEFINITION OF USE CASE

A Use Case captures a contract that describes the system behavior under various conditions as the system responds to a request from one of its stakeholders. In essence, a Use Case tells a stylized story about how an end user interacts with the system under a specific set of circumstances. A Use Case diagram simply describes a story using corresponding actors who perform important roles in the story and makes the story understandable for the users. The first step in writing a Use Case is to define that set of “actors” that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators. Every user has one or more goals when using the system. Actors either consume/ produce/ modify information.

Primary Actor

Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and

frequently with the software. The actors who do more than one job(consume/produce/ manipulate information) are our primary actors.

Secondary Actor

Secondary actors support the system so that primary actors can do their work. They either produce or consume information

2.3 USE CASE DIAGRAMS

Use Case diagrams to give a non-technical view of the overall system.

2.3.1 LEVEL-0

Level 0:

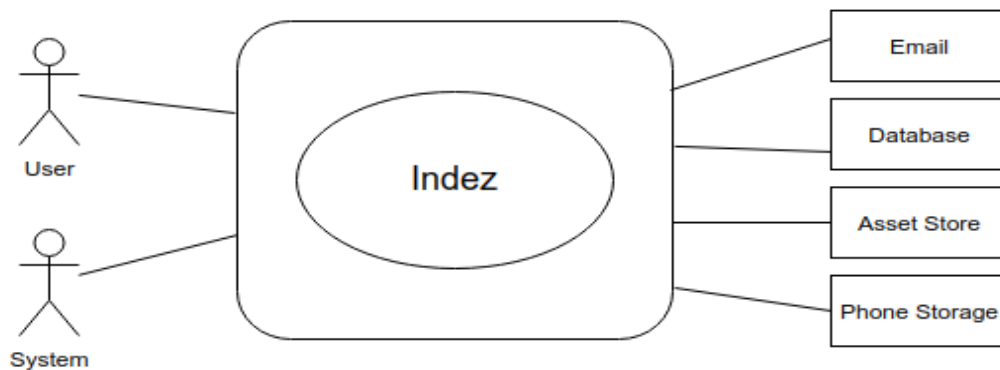


Figure 1: InDez

Description of Use case diagram level-0:

After analyzing the user's story, we found 6 actors that directly or indirectly interacts with the system. Primary actors are those who will play action and get a reply from the system whereas secondary actors only produce or consume information.

Primary Actors: User, Database, System, Phone Storage

Secondary Actors: Email, Asset Store

2.3.2 LEVEL-1

Level 1:

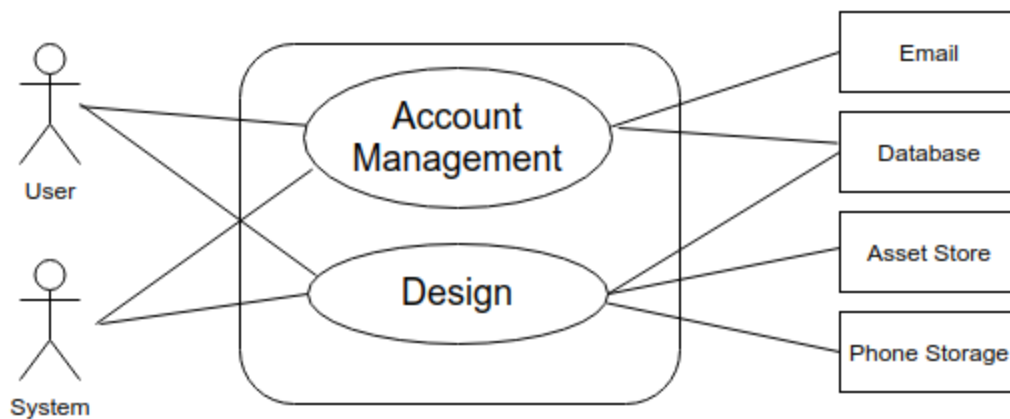


Figure 2: InDez_subsystems

Description of Use case diagram level-1:

There are two subsystems in the system. These are as following:

1. Account Management
2. Design

The subsystems are further decomposed, in level 1.1, 1.2 respectively.

Primary Actors: User, Firebase, System, Phone Storage

Secondary Actors: Email, Asset Store

2.3.3 LEVEL-1.1

Level 1.1:

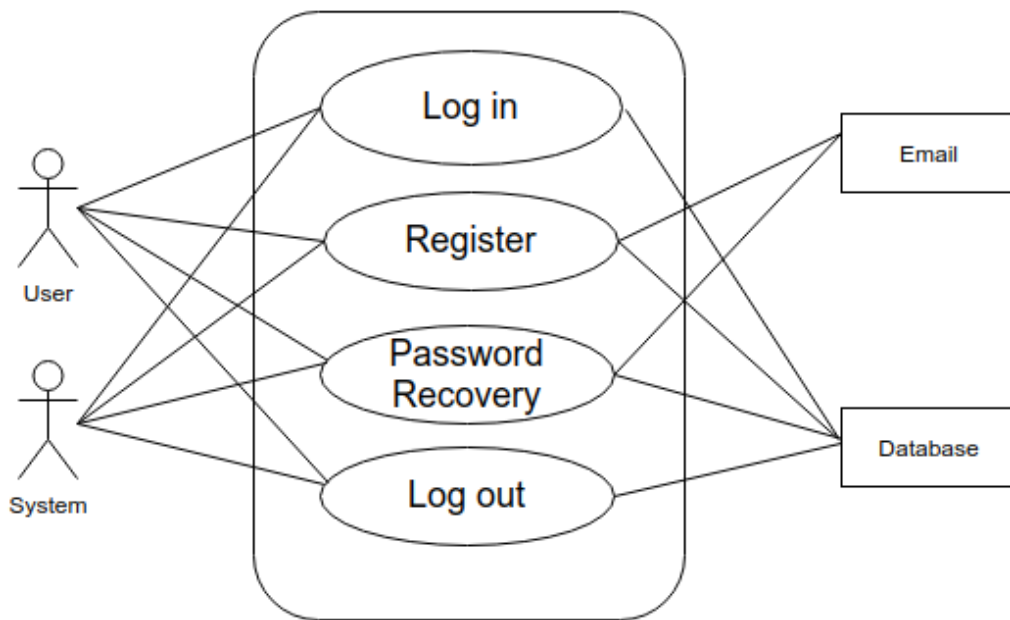


Figure 3: InDez_authentication

Description of Use case diagram level-1.1:

Authentication is a process in which credentials provided are compared to those on files in a database of authorized user's information. The authentication subsystem can be divided into four parts. They are as follows:

1. Log in
2. Register

3. Password Recovery

4. Log out

The Sign Up subsystem are further decomposed in level 1.1.2

Primary Actors: User, Database, System

Secondary Actors: Email

Action Reply :

A1: User provides credentials.

R1: System will check the validity of the given credentials. For valid information, the system will allow the user to log into the account.

A2: User provides invalid credentials.

R2: System will show error messages and allow the user to try again.

A3: User wants to recover password.

R3: Password recovered via email.

A4: User wants to log out from system.

R4: System turns off.

2.3.4 LEVEL-1.1.2

Level 1.1.2:

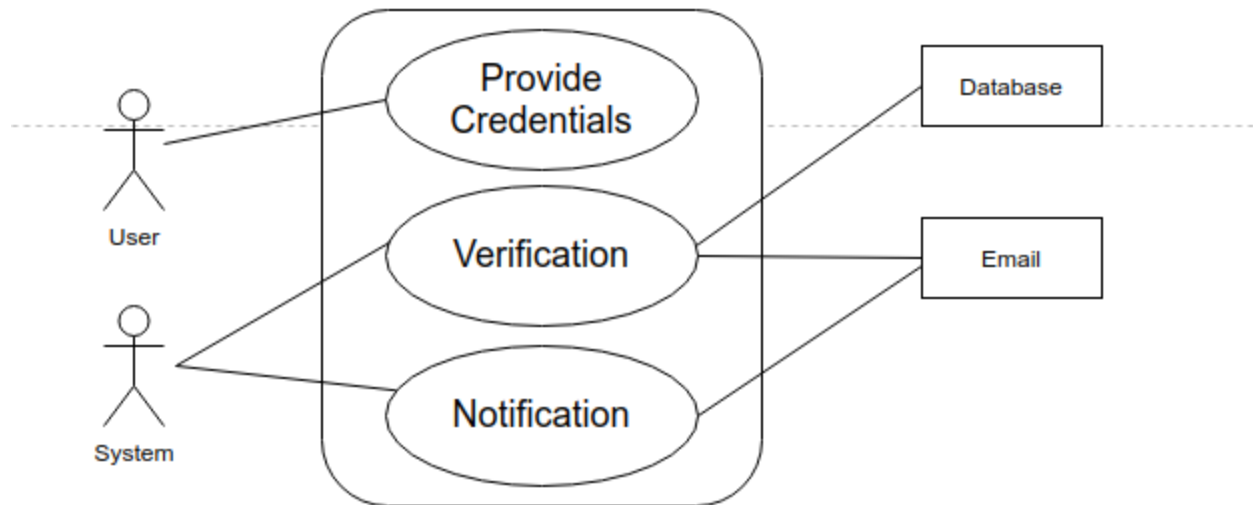


Figure 4: InDez_register

Description of Use case diagram level-1.1.2:

At the time of registration a user has to provide a valid email address, a username and password. This information will be stored in Database.

Primary Actors: User, System, Database

Secondary Actors: Email

Action Reply :

A1: User creates an account filled with valid information.

R1: If the information is valid, the System creates an account and the account information is stored. The user will be notified through email.

A2: User creates an account filled with invalid information.

R2: System allows the user to try again for providing information.

Note: Password should be at least 8 characters at most 32 characters

2.3.5 LEVEL-1.2

Level 1.2:

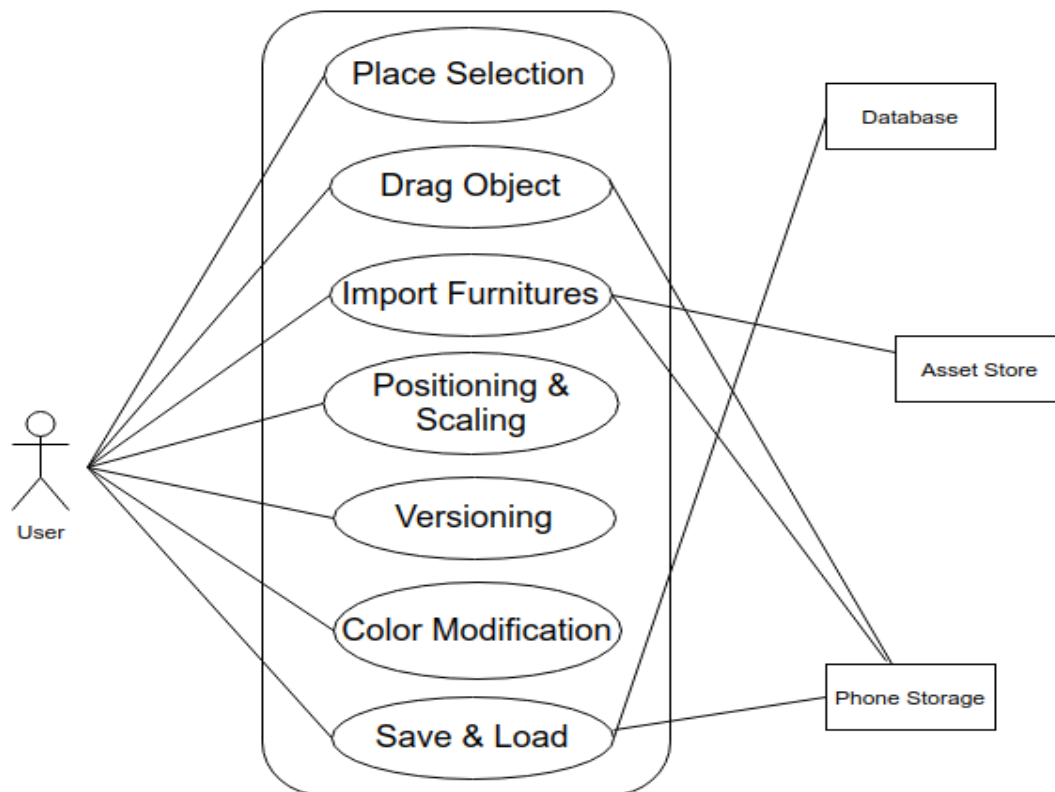


Figure 5: InDez_design

Description of Use case diagram level-1.2:

Anything related to designing and decorating rooms will be handled by this subsystem. This subsystem is divided into eight parts.

1.Place Selection

2. Drag Object

3. Import Furniture

4. Positioning & Scaling

5. Versioning

6. Color Modification

7. Save & Load

Drag object, import furniture, versioning and save & load are described in level 1.2.2, 1.2.3, 1.2.5 and 1.2.7 respectively.

Primary Actors: User, Database, Phone Storage

Secondary Actors: Email, Asset Store

Action Reply :

A1: User wants to select place to design

R1: Place will be selected into system

A2: User wishes to change the position of the furniture

R2: Ground position will be changed and will be saved into the system.

A3: User wishes to change the size of the furniture

R3: Furniture size will be changed and will be saved into the system.

A4: User wishes to change the color of the furniture

R4: Furniture color will be changed and will be saved into the system.

2.3.6 LEVEL-1.2.2

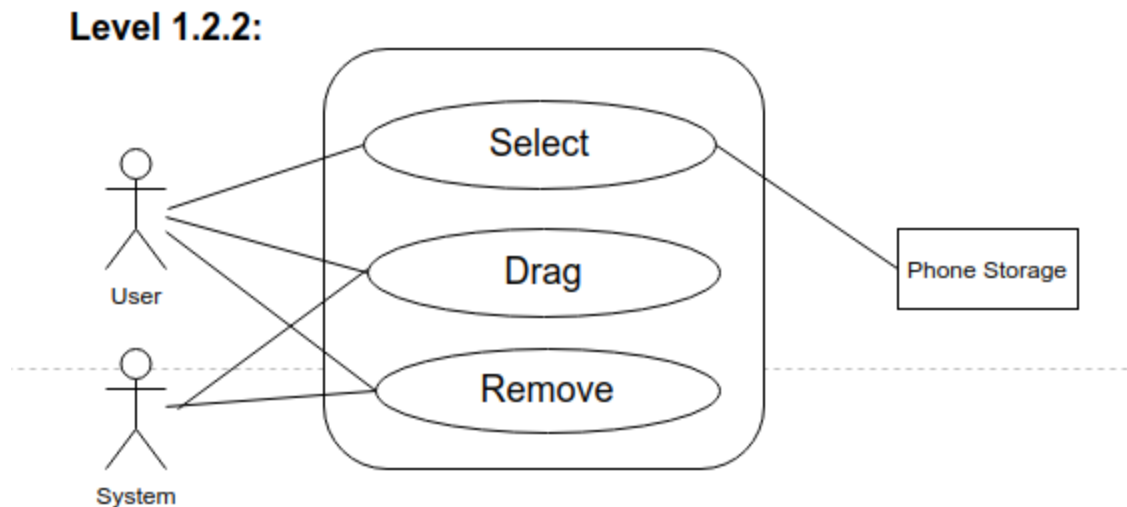


Figure 6: InDez_object_placement

Description of Use case diagram level-1.2.2:

Anything related to dragging augmented furniture from the side task bar and removing furniture from the project will be handled by this subsystem. This subsystem is divided in two parts.

1. Drag Object
2. Remove Object

Primary Actors: User, System, Phone Storage

Secondary Actors:

Action Reply :

A1: User wills to drag furniture from the taskbar.

R1: Selected object will be placed on the ground by the system

A2: User wishes to remove furniture which he/she added in the current room

R2: The furniture will be removed from the room.

2.3.7 LEVEL-1.2.3

Level 1.2.3:

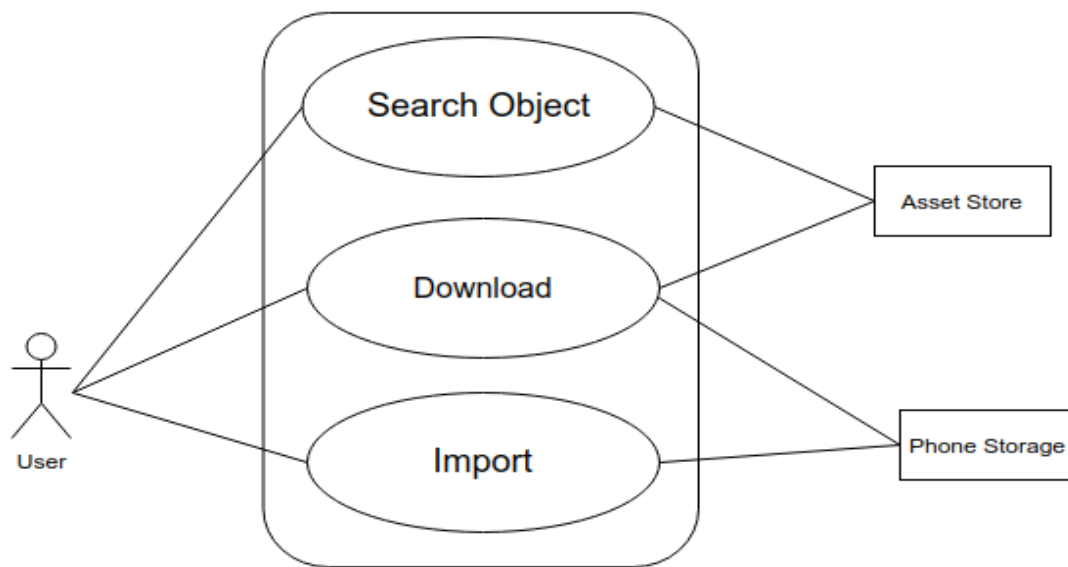


Figure 7: InDez_import

Description of Use case diagram level-1.2.3:

Anything related to importing customized furniture will be handled by this subsystem. This subsystem is divided into three parts.

1. Search Object
2. Download
3. Import

Primary Actors: User, System, Firebase

Secondary Actors: Asset Store, Phone Storage

Action Reply :

A1: User wants to search for the 3D model of the furniture, s/he wants to add in the room decoration.

R1: System will redirect the user to Unity asset store.

A2: User wants to download searched furniture.

R2: Furniture will be downloaded into the phone storage.

A3: User wants to import downloaded Furniture.

R3: Furniture will be imported into the current project from Phone Storage.

2.3.7 LEVEL-1.2.5

Level 1.2.5:

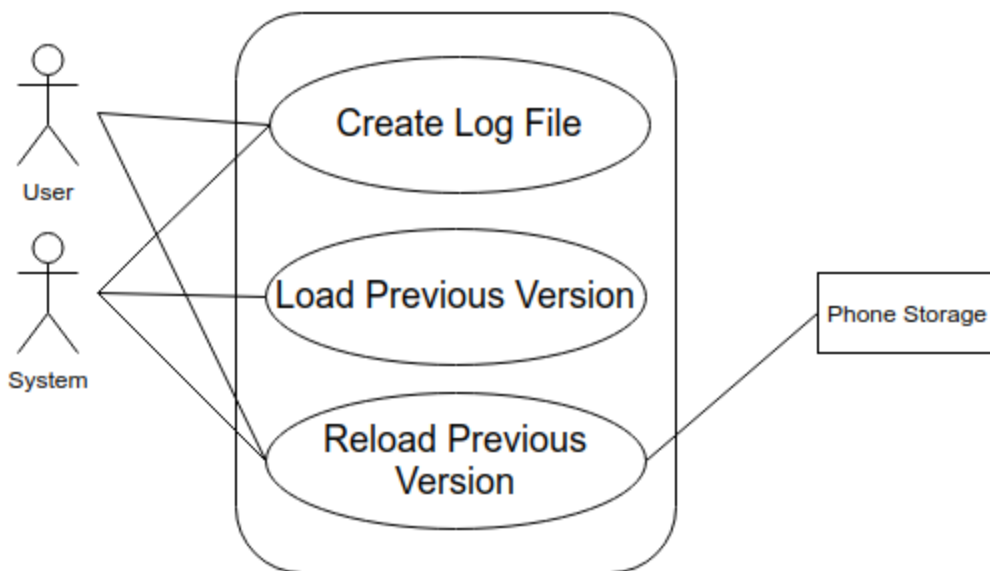


Figure 8: InDez_redo_undo

Description of Use case diagram level-1.2.5:

Anything related to undo and redo work will be handled by this subsystem. This subsystem is divided in three parts.

1. Create Log File
2. Load Previous Version
3. Reload Previous Version

Primary Actors: User, System, Phone Storage

Secondary Actors:

Action Reply :

A1: Users do any work.

R1: System will create a log file and this file will be stored into Phone Storage.

A2: User wants to undo any work.

R2: System will load previously saved log file from Phone Storage..

A3: User wants to redo a work.

R3: System will load previously saved log file from Phone Storage.

2.3.8 LEVEL-1.2.7

Level 1.2.7:

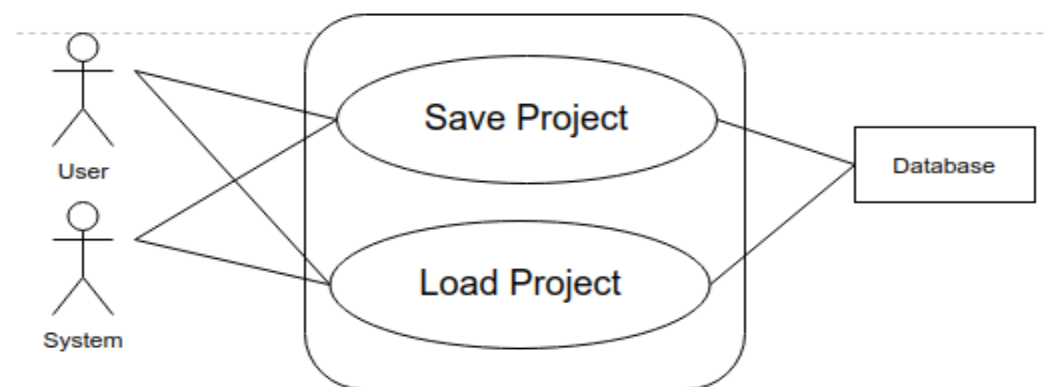


Figure 9: InDez_save_load

Description of Use case diagram level-1.2.7:

Anything related to save and load previously saved project will be handled by this subsystem. This subsystem is divided in two parts.

1. Save Project
2. Load Project

Primary Actors: User, System, Database

Secondary Actors:

Action Reply :

A1: User wants to save current design of the room.

R1: System will create a log file and that file will be stored into users phone storage and into firebase database against users' account.

A2: User wants to load a previously saved project.

R2: System will load the log file from phone storage or from firebase database.

3. ACTIVITY DIAGRAM OF InDez

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

ID: Level-1

Name: indez

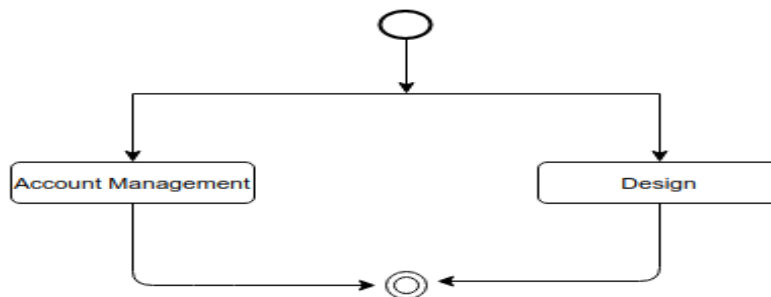


Figure 10: InDez_subsystems

Description: It is visible that the following things can be done in the whole system-

1. Account Management
2. Design

ID: Level-1.1

Name:authentication

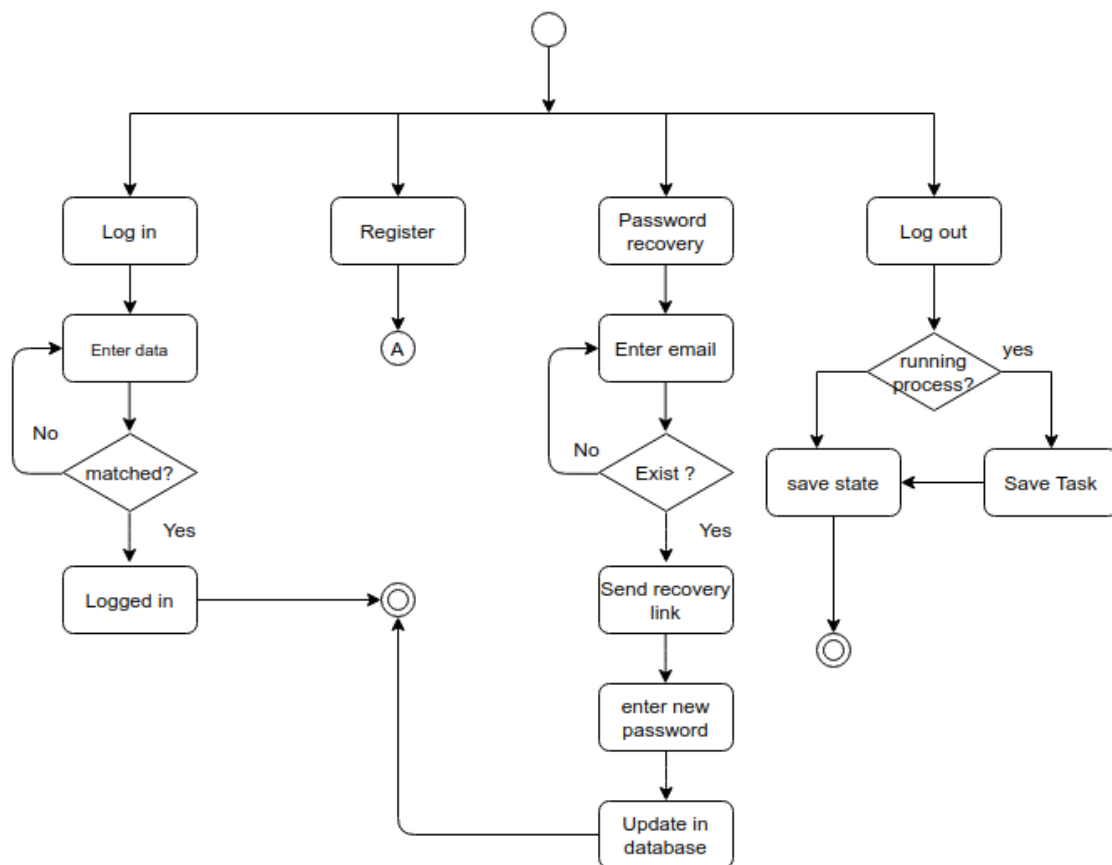


Figure 11: InDez_authentication

Description: It is visible that in Account management the following things can be done.

1. Login
2. Register

3. Password Recovery
4. Log out

it is also visible that username and password has to be entered to sign in. If the username and password matches and is found in the database, then the user will successfully log in. On the contrary, the user will be asked if s/he forgot the password. If the user confirms that s/he forgot the password, an email will be sent to him/her to recover his/her account. If the user has not forgotten his password, s/he will be able to re enter the username and password.

ID: Level-1.1.2
Name: register

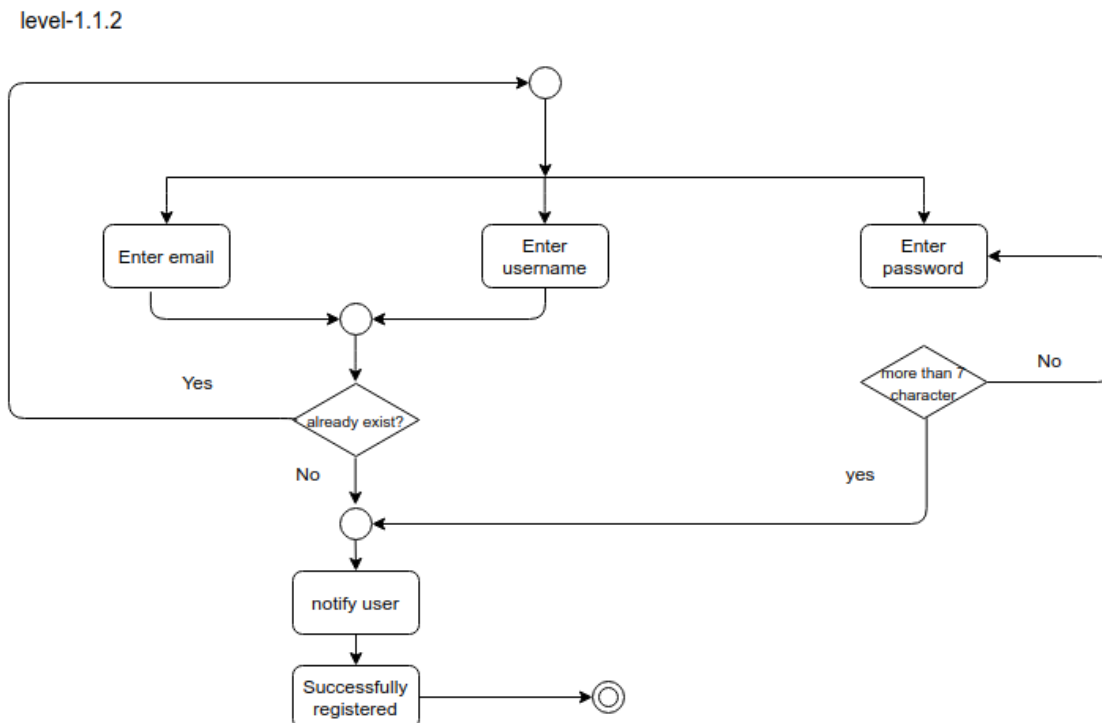


Figure 12: InDez_register

Description: It is visible that the user has to enter email, username and password to register. If the username or email already exists in the database, then the user will be asked to enter a new username. If any of the above information is invalid, s/he will be asked to re enter the

information. After all the information is verified, the user will be notified. With this, the registration will be completed.

ID: Level-1.2

Name: Design

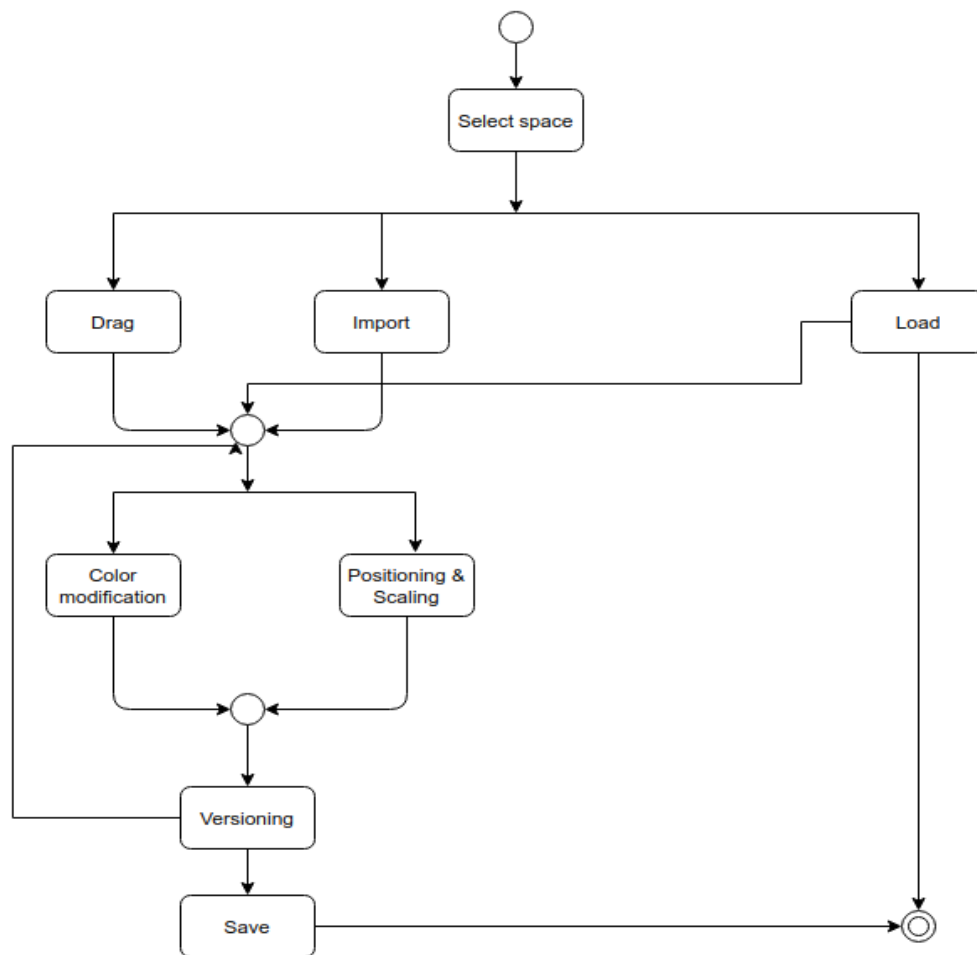


Figure 13: InDez_design

Description: It is visible that after choosing to design the user can select a workspace. After selecting workspace s/he can drag and import objects, he can also load any of previously saved work. After dragging or importing the object user can change its position,color and resize it. After completing design s/he can save the work and make the end of this process

ID: Level-1.2.2
Name: Drag

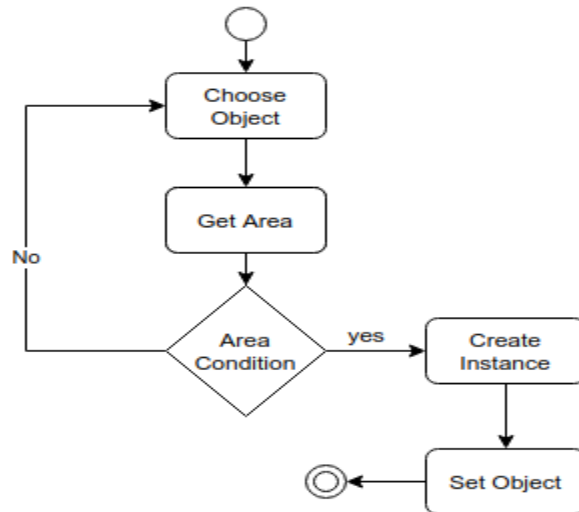


Figure 14: InDez_drag

Description: After selecting the furniture object to drag it into the designing space , the system will first try to detect the ground plane. If there is a ground plane system will create an instance of the furniture object.

ID: Level-1.2.3
Name: Import

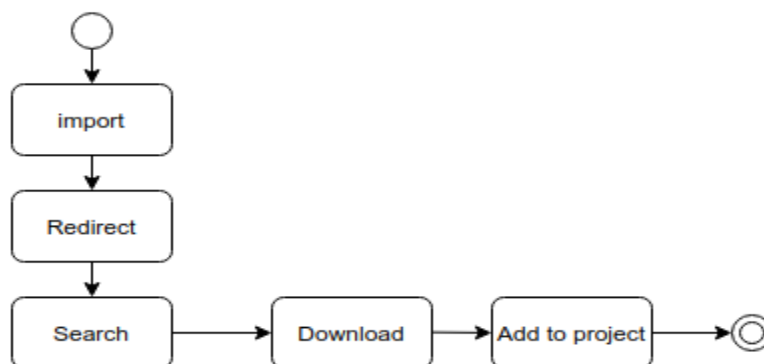


Figure 15: import

Description: It is visible that if a user presses import option s/he will be redirected to assets store by the system automatically. Then s/he will search and download the furniture he wants to import. After the furniture has been downloaded, the system will add it to the project.

ID: Level-1.2.5

Name: Versioning

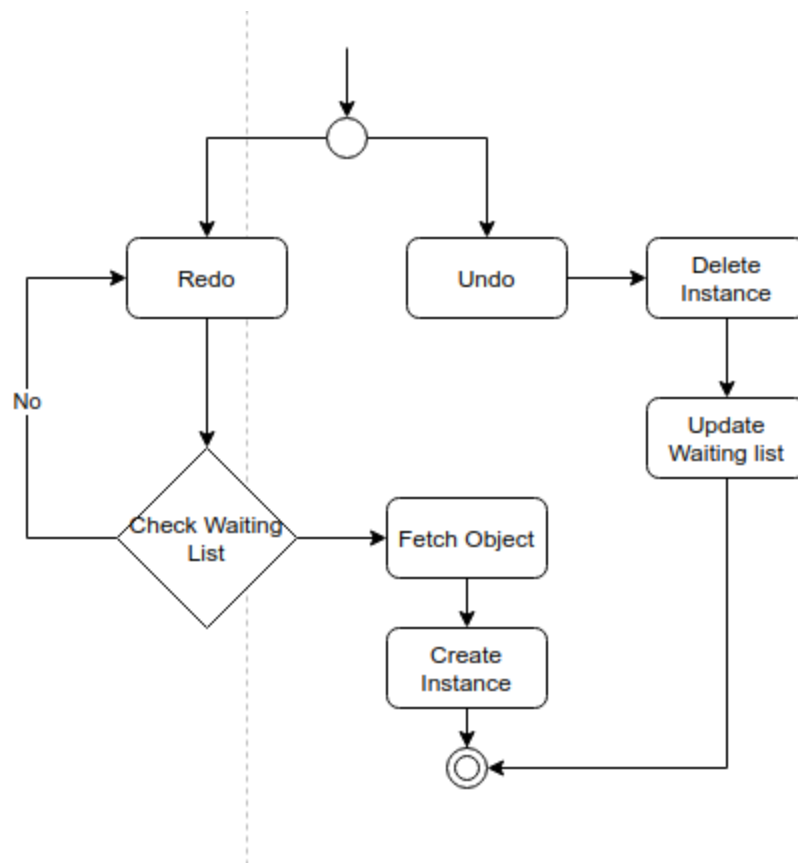


Figure 16: InDez_versioning

Description: After every action the user has taken the system will create a log file. If the user wants to undo his/her action the system will delete the instance and will update the waiting list. If a user wants to redo, a furniture object will be fetched and the system will create an instance of it.

ID: Level-1.2.7
Name:

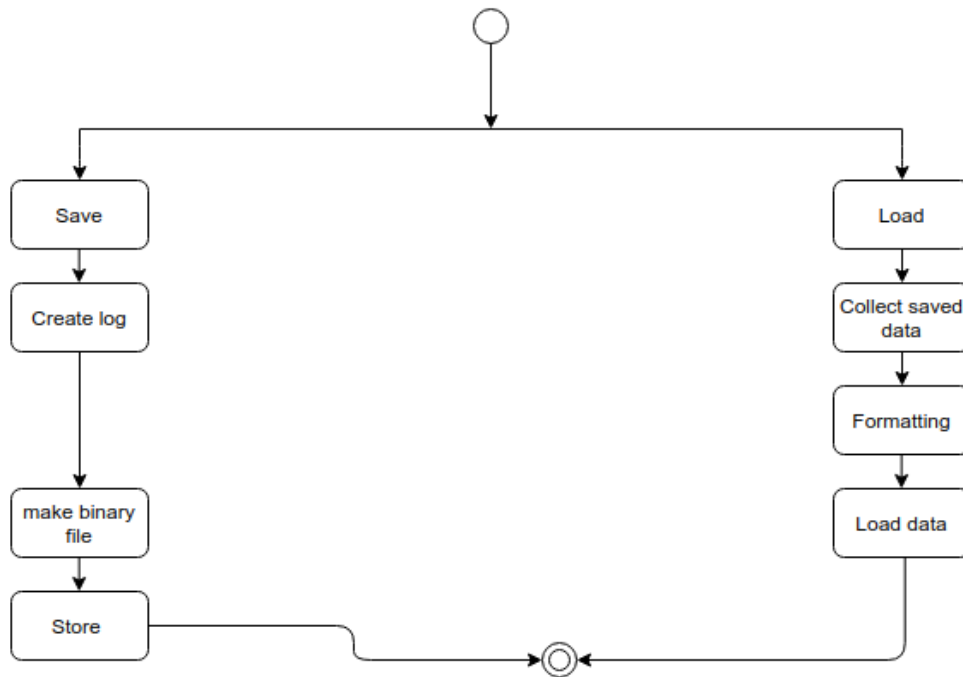


Figure 17: InDez_saveAndLoad

Description: It is visible that while designing a space if a user wants to save his or her current work the system will first create a log file of the working project then will make a binary file and store it in a database. If a user wants to load his or her saved work the system will collect his saved data and after creating the log file system will load the data.

4. SWIMLANE DIAGRAM

A **swimlane** (or **swimlane diagram**) is used in process flow **diagrams**, or flowcharts, that visually distinguishes job sharing and responsibilities for sub-processes of a business process.

Swimlane Diagram 1: Authentication

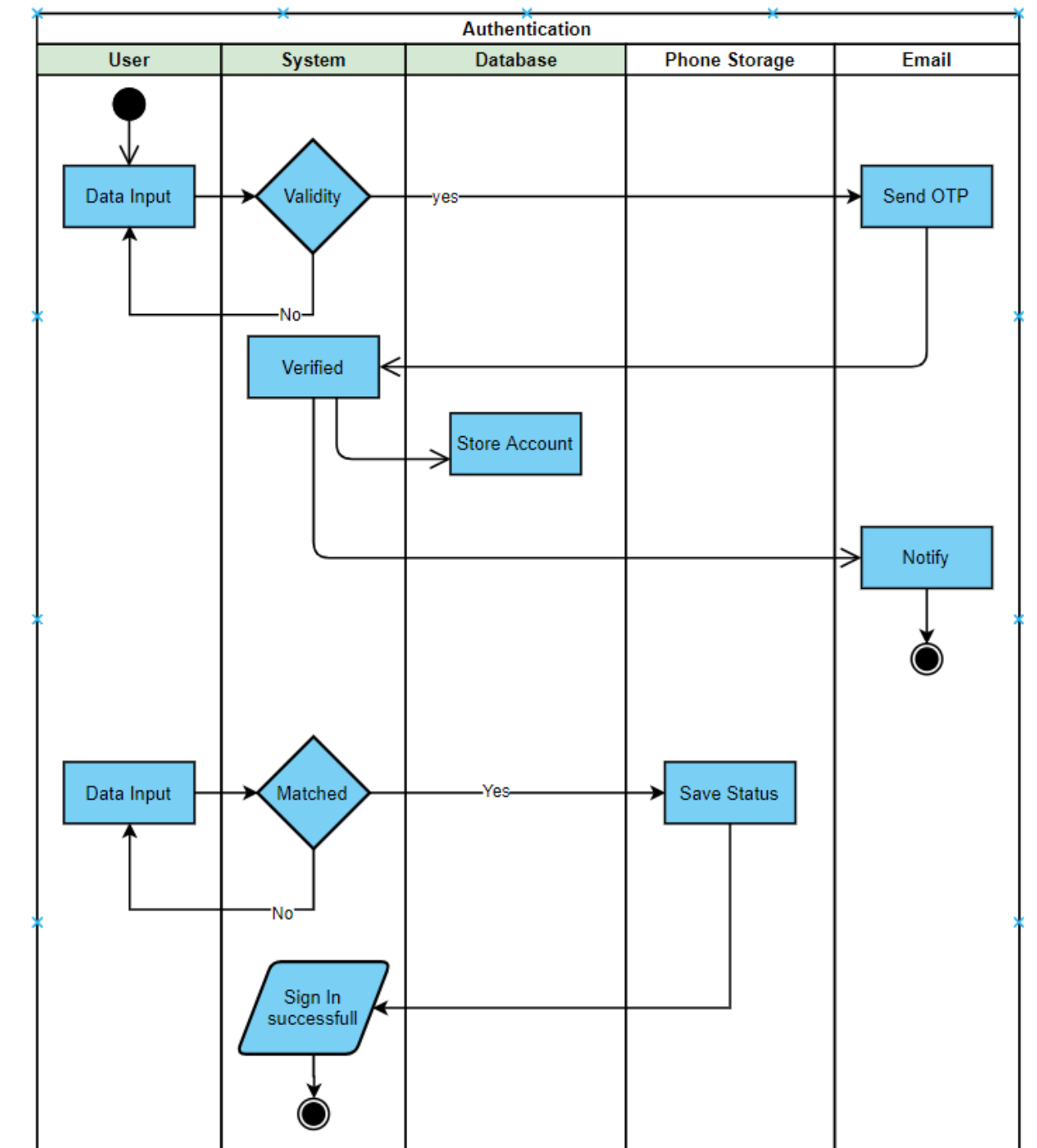


Figure 18: Level 1

Swimlane Diagram 2: Log Out

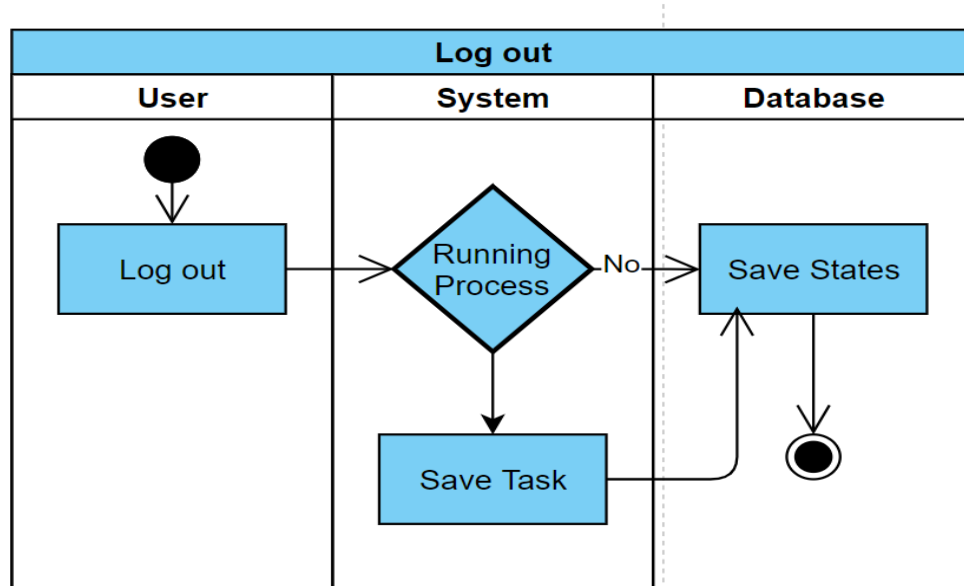


Figure 19: Level 2

Swimlane Diagram 3: Password Recovery

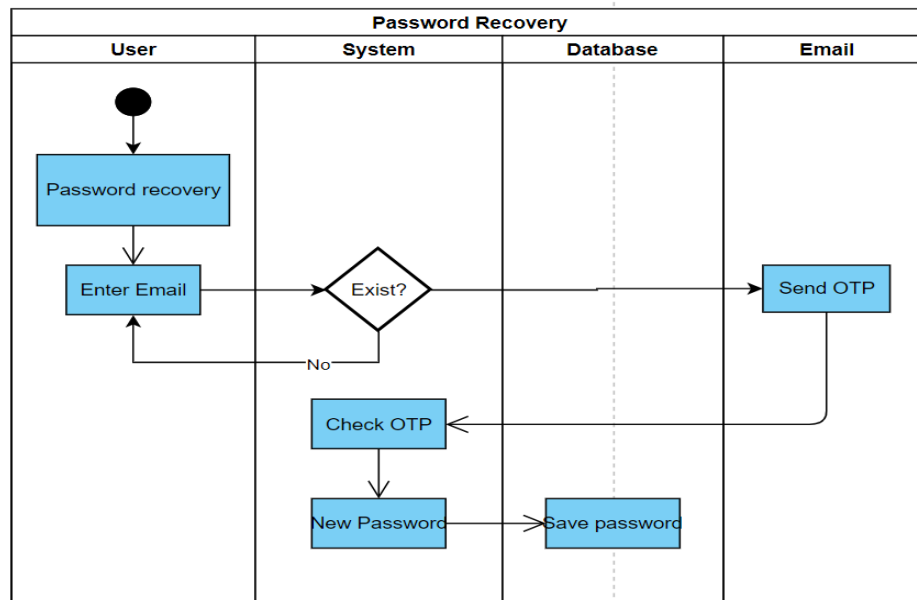


Figure 20: Level 3

Swimlane Diagram 4: Choose Place

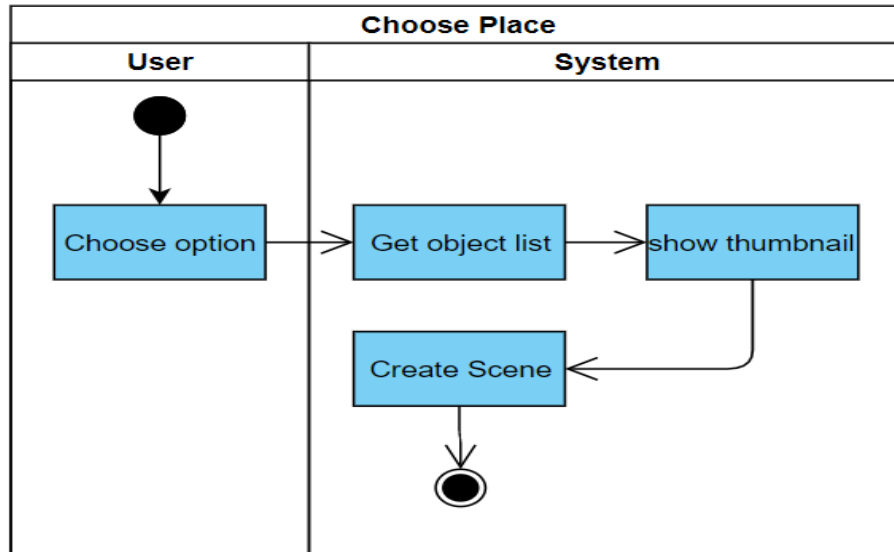


Figure 21: Level 4

Swimlane Diagram 5: Object Placement

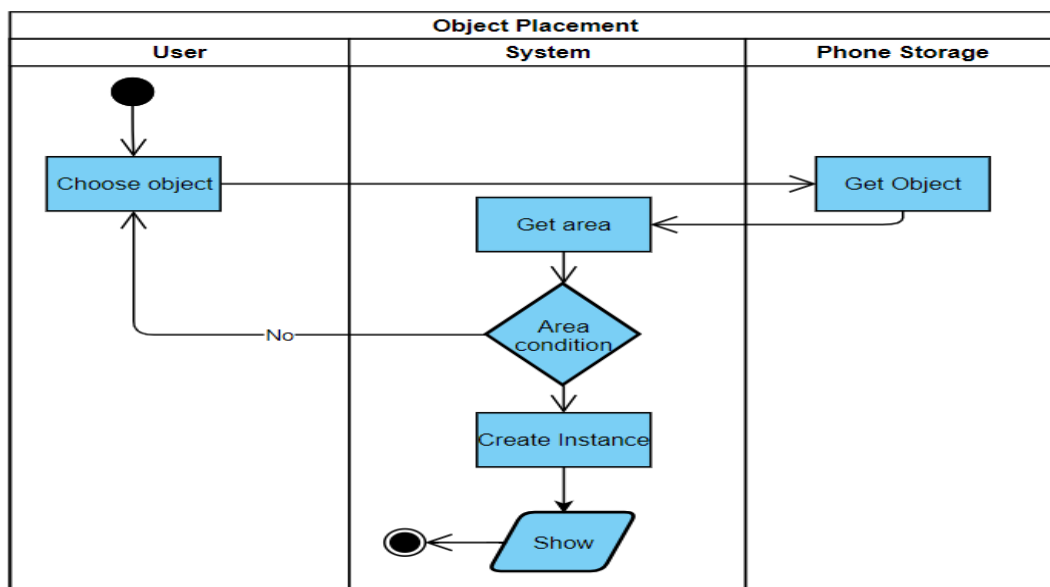


Figure 22: Level 5

Swimlane Diagram 6: Import

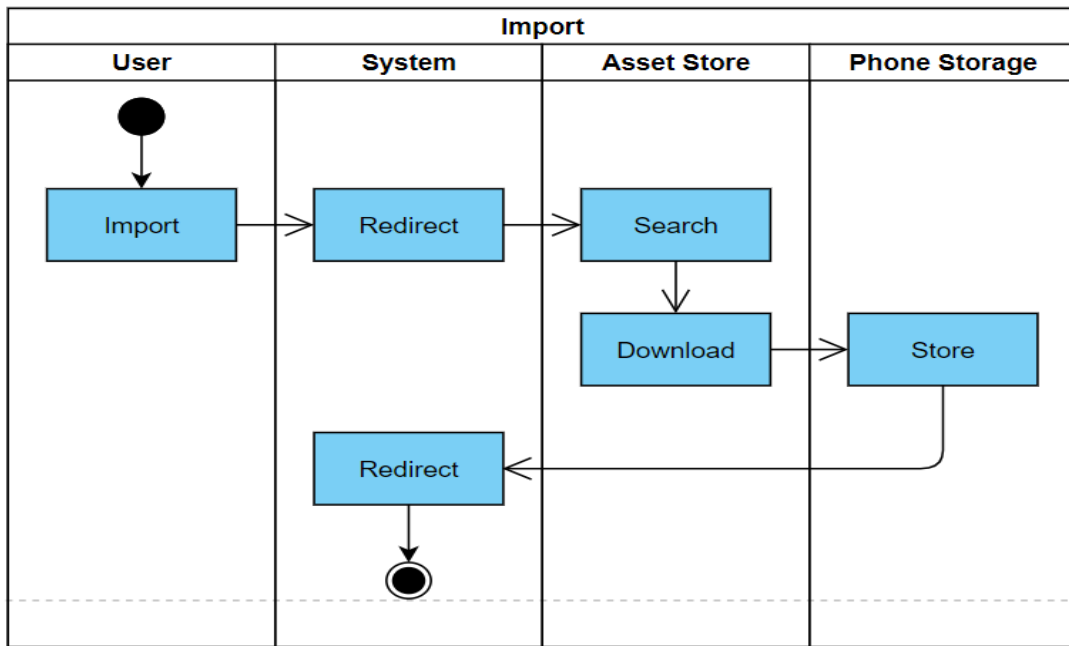


Figure 23: Level 6

Swimlane Diagram 7: Furniture Modify

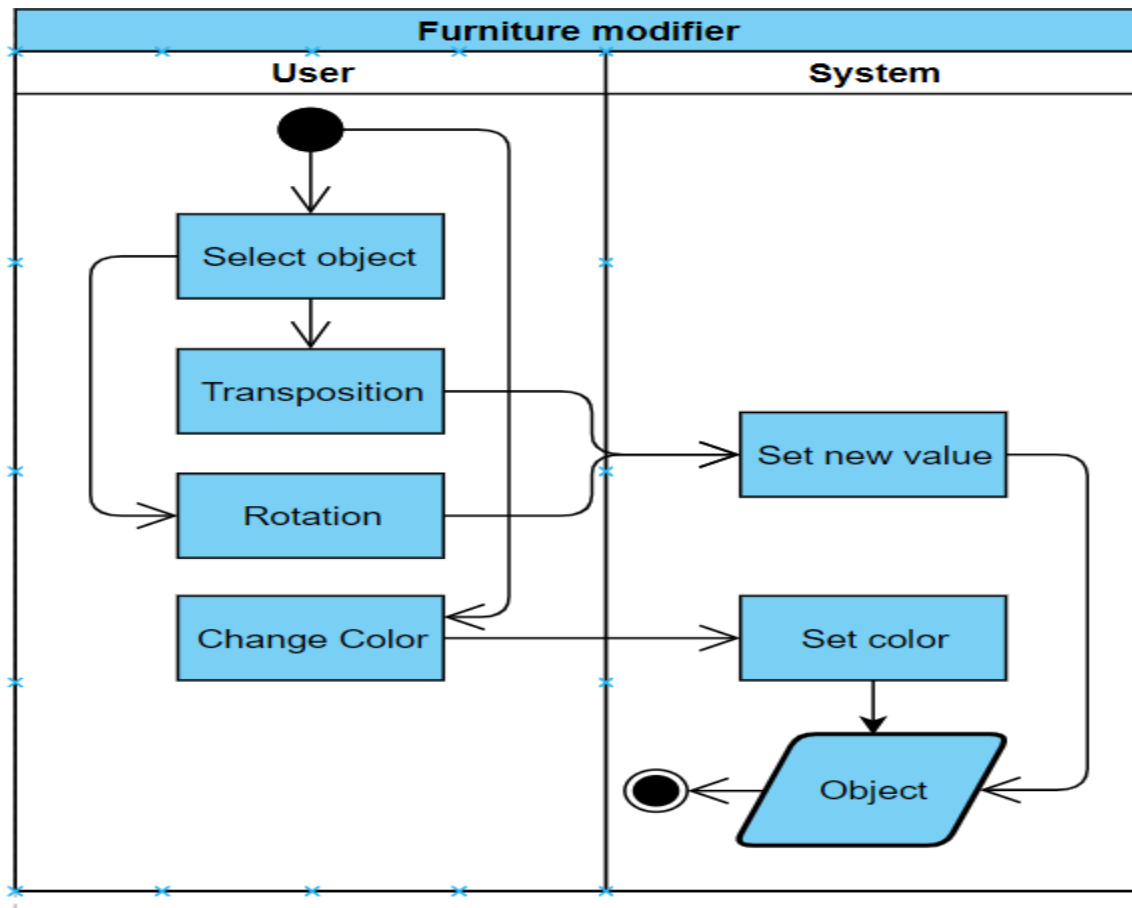


Figure 24:Level 7

Swimlane Diagram 8: Versioning

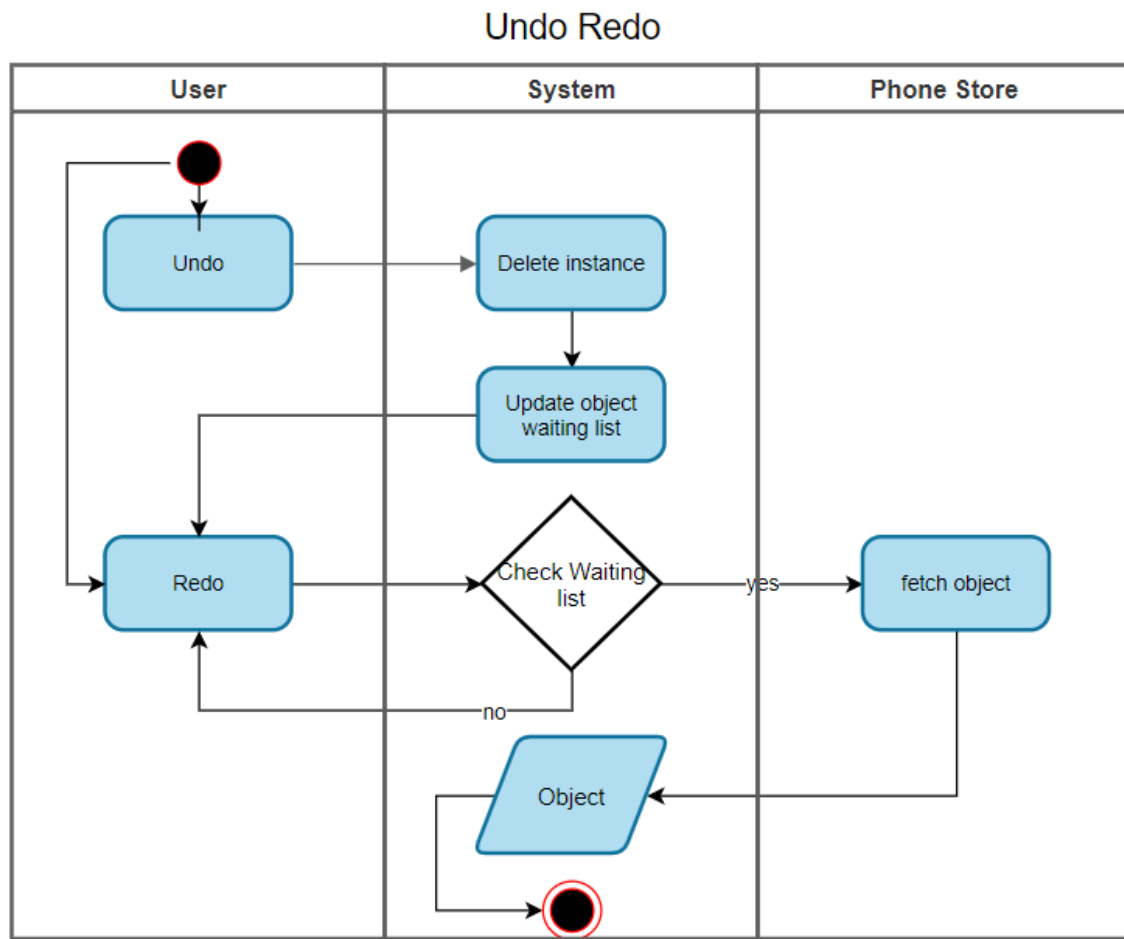


Figure 25: Level 8

Swimlane Diagram 9: Save and Load

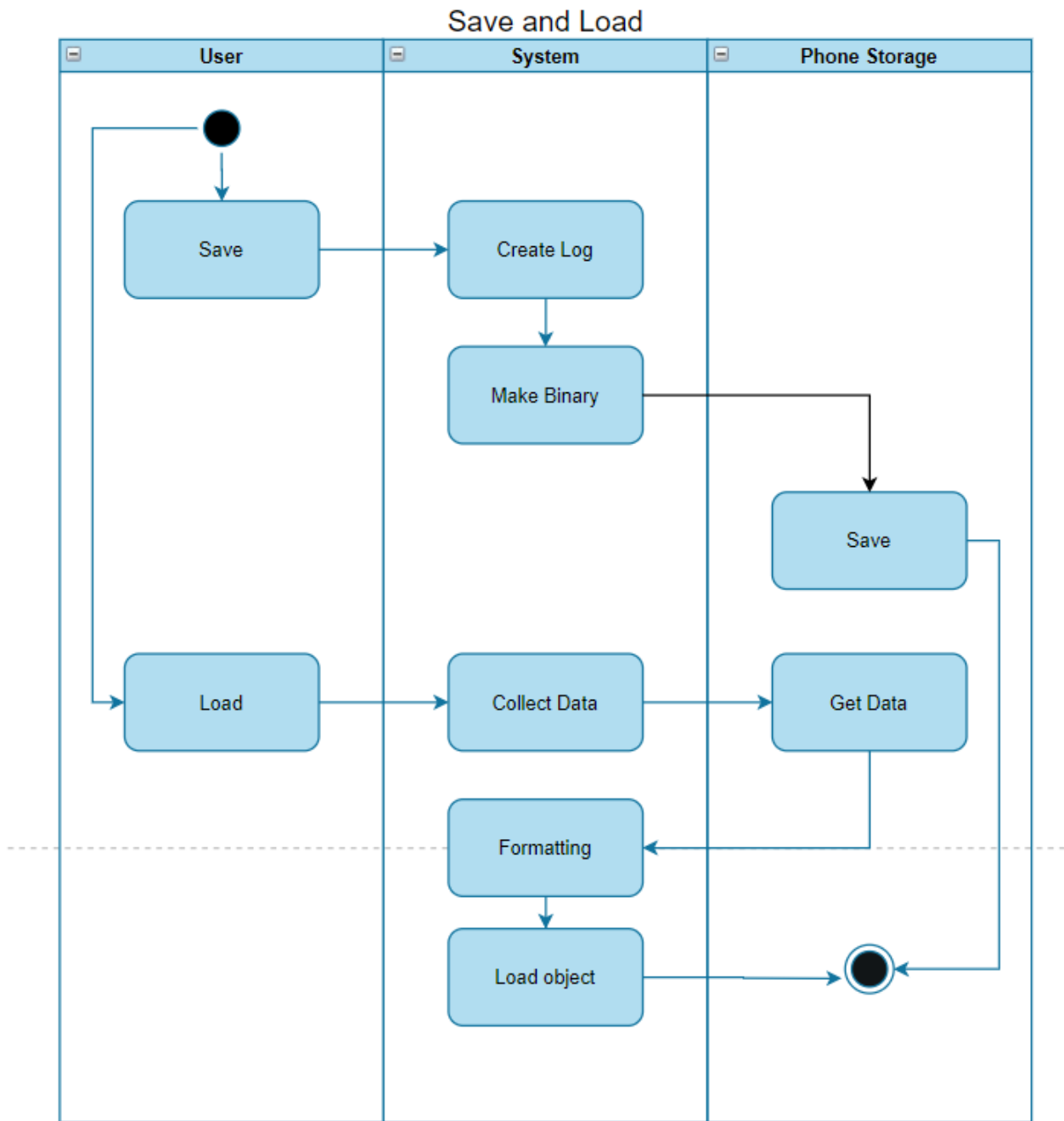


Figure 26: Level 9

5. DATA MODELLING

5.1 Data Modelling Concept

If software requirements include the necessity to create, extend or interact with a database or complex data structures need to be constructed and manipulated, then the software team chooses to create data models as part of overall requirements modeling. The entity-relationship diagram (ERD) identifies all data objects that are processed within the system, the relationships between the data objects and the information about how the data objects are entered, stored, transformed and produced within the system.

There is not much use of database for this project. We are going to save the current working state in the database along with the user profile in the database. Database used for this project is not a relational one, rather it is a database provided by the firebase.

5.1.1 Data Objects

A data object is a representation of composite information that must be understood by the software. Here, composite information means information that has a number of different properties or attributes. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

5.1.2 NOUN IDENTIFICATION

We identified all the nouns whether they are in problem space or in solution space from our story:

Serial no	Noun	Problem/solution space	Attributes
1	Furniture	s	3,11,13,16,22
2	User	s	5,6,7,8,3

3	Work-space	s	1,9,12,17,18,20, 23,24,26,27
4	Account	s	5,6,7,8
5	E-mail	p	
6	email-address	s	
7	username	s	
8	password	s	
9	Customised Furniture	s	3,11,13,16,22
10	System	p	
11	Position	p	
12	Light	p	
13	Color	p	
14	Camera	p	
15	Asset Store	s	9
16	Default size	p	
17	Unfinished work	p	
18	Current status	p	
19	Database	s	
20	Ground Plane	p	
21	Android Application	p	
22	Shadow	p	
23	Sidebar	p	1
24	Taskbar	p	1

25	Mobile Storage	p	
26	Wall	p	
27	Floor	p	
28	Log file	s	
29	Binary file	s	
30	Size		

5.1.3 Potential Data objects:

No	Data Object	Attribute
1	Furniture	3,11,13,16,22
2	User	5,6,7,8,3
3	Workspace	1,9,12,17,18,20,23,24,26,27
4	Account	2,6,7,8
5	Customize Furniture	3,11,13,16,22

5.1.4 Analysis for finalizing Data Objects

- ❑ Both Furniture and Customize Furniture have similar attributes. So their attributes can be stored as **Furniture**.
- ❑ Both User and Account have similar attributes. So their attributes can be stored as **User**.

5.1.5 Final Data Objects

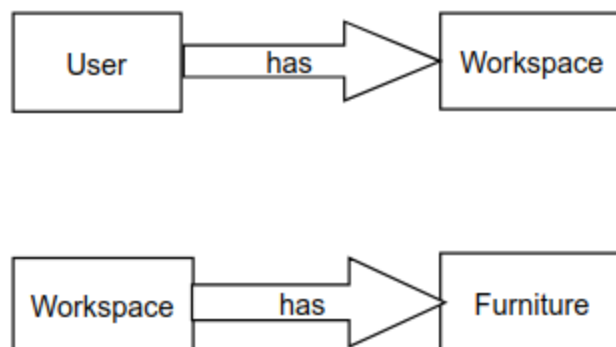
In the following table we finalize the data objects with their attributes. Most of the attributes of the data objects are selected from the usage scenario and some of the attributes are selected to complete the system which are not in the usage scenario but important for the data objects.

Table 3: Final data object

1	User-Account	<u>userID</u> , username, email, password
2	Furniture	<u>furnitureID</u> , size, color, positionX, positionY, positionZ
3	Workspace	<u>workspaceID</u> , userID, log file

5.2 DATA OBJECT RELATIONSHIPS

Data objects are connected to one another in different ways.



5.3 ENTITY RELATIONSHIP DIAGRAM

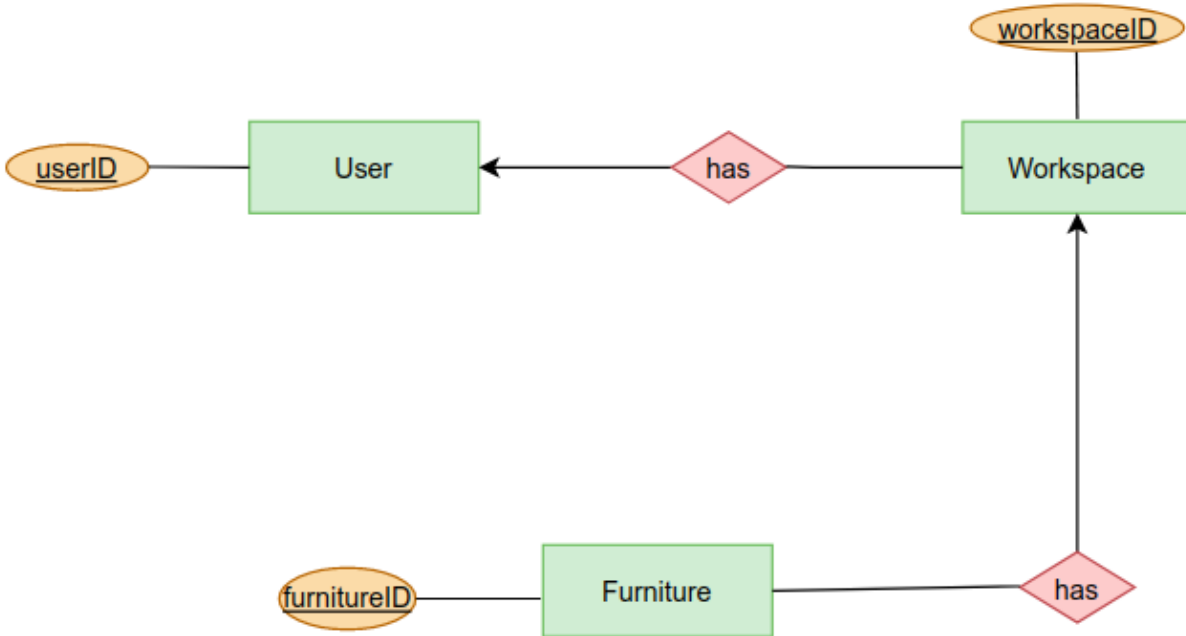


Figure 27: InDez_ERD

5.4 SCHEMA DIAGRAM

Table 4: Schema for user

USER		
Attributes	Type	Size
userID	Numeric	50
username	Varchar	50

email	Varchar	50
password	Varchar	20

Table 5: Schema for Furniture

Furniture		
Attributes	Type	Size
furnitureID	Numeric	50
workspaceID	Numeric	50
color	varchar	100
size	Numeric	50
positionX	Numeric	50
positionY	Numeric	50
positionZ	Numeric	50

Table 6: Schema for Workspace

Workspace		
Attributes	Type	Size
workspaceID	Numeric	50
userID	Numeric	50
log file	Varchar	2000

6. CLASS-BASED MODELING

This Chapter is intended to describe class based modeling of “Course Management System”.

6.1 CLASS BASED MODELING CONCEPT

Class-based modeling represents the objects that the system will manipulate, the operations that will be applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

6.2 Noun List

No	Noun	No	Noun
1	Furniture	26	Wall
2	User	27	Floor
3	Work-space	28	Log file
4	Account	29	Binary file
5	E-mail	30	
6	email-address		
7	username		
8	password		
9	Customised Furniture		
10	System		
11	Position		
13	Color		
14	Camera		
15	Asset Store		
16	Default size		
17	Unfinished work		
18	Current status		
19	Firestore		
20	Ground Plane		
21	Android Application		
22	Shadow		
23	Sidebar		
24	Taskbar		
25	Mobile Storage		

6.3 GENERAL CLASSIFICATION

To identify the potential classes, we have first selected the nouns from the solution space of the story. These were then characterized in seven general classification. The seven general characteristics are as follows:

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

Following are the specifications of the nouns according to the general Classifications:

Table:1 General Classification

No.	Noun	General Classification
1	Furniture	2,5,7
2	User	4,5,7
3	Work-space	2,6,7
4	Account	2,4,7
5	E-mail	1
6	email-address	
7	username	
8	password	
9	Customised Furniture	2,5,7

10	System	2,4,7
11	Position	
12	Light	
13	Color	
14	Camera	1
15	Asset Store	1,2,6
16	Default size	
17	Unfinished work	
18	Current status	
19	Firebase	1,2,4,7
20	Ground Plane	
21	Android Application	
22	Shadow	
23	Sidebar	
24	Taskbar	
25	Mobile Storage	6
26	Wall	
27	Floor	
28	Log file	
29	Binary file	
30		

6.4 SELECTION CRITERIA

The potential classes were then selected as classes by six Selection Criteria. A potential class becomes a class when it fulfills all six characteristics.

1.Retain information

2.Needed services

3.Multiple attributes

4.Common attributes

5.Common operations

6.Essential requirements

Table:2 Selection Criteria

No	Noun	Selection Criteria
1	Furniture	1-5
2	User	1-5
3	Workspace	1-5
4	Account	1-5
5	System	6
6	Database	6
7	Customized Furniture	1-5
8	Asset Store	1,2

6.5 ATTRIBUTE SELECTION

After identifying the classes, we have specified their attributes and methods.

Table:3 Attribute Selection

No	Name	Attribute
1	Furniture	position color size
2	User	email username password
3	Workspace	Furniture Customize Furniture type waitingArray
4	Account	email username password
5	System	profile log file binary file
6	Database	user information log file/binary file
7	Customize Furniture	position color

		size
--	--	------

6.6 METHOD IDENTIFICATION

After identifying the classes, we have specified their methods.

Table:4 Method Identification

No	Noun	Method
1	Furniture	getPosition() setPosition() getColor() setColor() getSize() setSize() drag() drop()
2	User	getEmail() setEmail() getUsername() setUsername() getPassword() setPassword() signup() login()

		resetPassword() selectWorkspace()
3	Workspace	dragFurniture() dropFurniture() setWorkspace() getWorkspace() detectGroundPlane() saveCurrentState() redo() undo()
4	Account	getEmail() setEmail() getUsername() setUsername() getPassword() setPassword() signup() login() resetPassword()
5	System	verifyInformation() createLogfile() createBinaryfile() storeBinaryfile()

		getBinaryfile() loadLogfile() redirect() addToProject()
6	Database	getUserInformation() storeUserInformation() saveBinaryfile() getBinaryfile() storeCurrentState()
7	Customize Furniture	getPosition() setPosition() getColor() setColor() getSize() setSize() getTexture() setTexture() download() delete() drag() drop()

6.7 Analyzing Classes

- User has common methods and attributes with account. So, we merge them as User and add the common and unique attributes and methods to User.
- Furniture has common methods and attributes with Customize Furniture. So, we merge them as Furniture and add the common and unique attributes and methods to Furniture.

6.8 Class Card

After identifying our final classes we have generated the following class cards.

Table:5 Class Card for Furniture

Furniture	
Attribute	Method
position	getPosition()
color	setPosition()
size	getColor()
	setColor()
	getSize()
	setSize()
	drag()
	drop()
	download()
	delete()
Responsibilities	Collaboration

<ol style="list-style-type: none"> 1. Resizing Furniture size 2. Changing Color/Texture 3. Changing Position 4. Dragging Furniture 5. Dropping Furniture 6. Importing Furniture 7. Removing Furniture 	User System Database Workspace
--	---

Table:6 Class Card for User

User	
Attribute	Method
e-mail username password	getEmail() serEmail() getUsername() setUsername() getPassword() setPassword() signup() login() resetPassword()
Responsibilities	Collaboration
<ol style="list-style-type: none"> 1. Creating a new user account. 2. Involving with system operations (example: saving current state of work) 3. Selecting workspace 	System Furniture Workspace Database

Table:7 Class Card for Workspace

Workspace	
Attribute	Method
Furniture Customize Furniture type	dragFurniture() dropFurniture() setWorkspace() getWorkspace() detectGroundPlane() saveCurrentState() loadPrevious() redo() undo()
Responsibilities	Collaboration
1. Setting workspace 2. Getting Furniture 3. Adding and removing light 4. Detecting Ground Plane 5. Adding and removing Furniture 6. Save current state 7. Load previous state	Furniture System Database

Table:8 Class Card for System

System	
Attribute	Method

profile log file binary file	verifyInformation() createLogfile() createBinaryfile() storeBinaryfile() getBinaryfile() loadLogfile() redo() undo() redirect() addToProject()
Responsibilities	Collaboration
1. Verifying Information 2. Creating log file 3. Creating and store Binary file 4. Loading log file 5. Redirecting to asset store	Database Workspace System

Table:9 Class Card for Database

Database	
Attribute	Method

user information current status log file/binary file	getUserInformation() storeUserInformation() saveBinaryfile() getBinaryfile() storeCurrentState()
Responsibilities	Collaboration
1. Store user 2. Saving file 3. Saving current state 4. Giving current state	User System

6.9 CRC Diagram

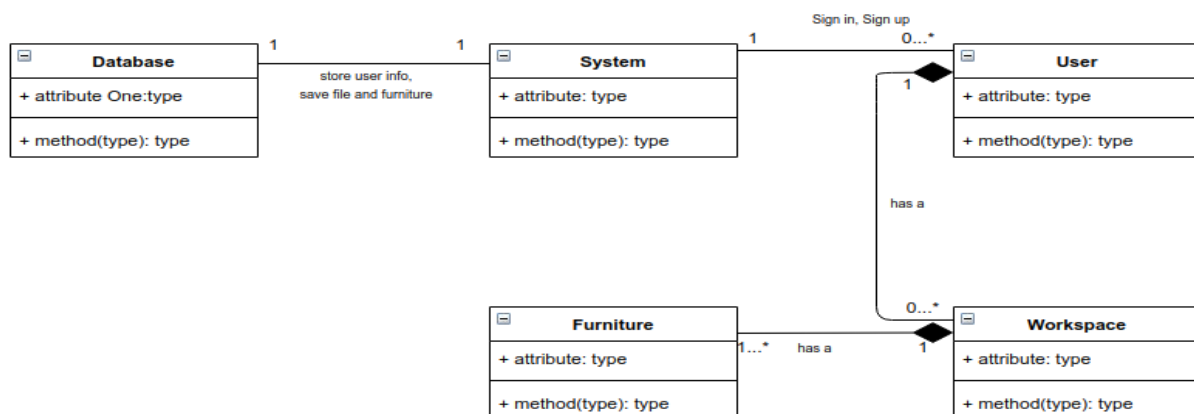


Figure 28: InDez_crc

7. BEHAVIORAL MODELING

Serial NO	Event	Event Name	Initiator	Collaborator
1	Entering into InDez	entry	User	System
2	Creating account	Sign up	User	System, Email, Database
3	Providing information ¹	Providing_info	User	System
4	Verifying information	Verification	System	
	Sending OTP	sent_otp	System	
5	Providing credentials	Sign In	User	System, Database
6	Recovering password via email	Recovering	User	Email, System, Database
7	Selecting workspace	select_workspace	User	
	Load scene	load	System	
8	Dragging Furniture	drag_furniture	User	System, Phone Storage
9	Dropping Furniture	drop_furniture	User	System
10	Changing Size	change_size	User	System
11	Changing Position	change_position	User	System
12	Changing Color	change_color	User	System
13	Detecting Ground	detect_ground	System	
14	Importing Furniture	import_furniture	User	System, Asset Store,
15	Adding to Taskbar	adding_to_taskbar	System	Phone Storage
17	Saving current working state	save_current_state	System	
18	Creating log file	create_log_file	System	
19	Converting into Binary file	make_binary_file	System	
20	Saving Binary file	save_binary_file	System	Phone storage, Database
21	Decrypting binary file	decrypt_binary_file	System	Database
22	Loading previous state	load_previous	User	System, Database

23	Loading log file	load_log_file	System	Database
24	Redirecting to asset store	redirect	System	
26	Undoing work	undo	User	System
27	Redoing work	redo	User	System
28	Converting Log File	Convert_Log	System	
29	Requesting object	Object_request	System	Database
30	Updating Waiting Object	Waiting_object	System	
31	Loading object from waiting list	Load_waiting	System	Database

NOTE

1. To create an account user has to provide information.

7.1 State Diagrams

User

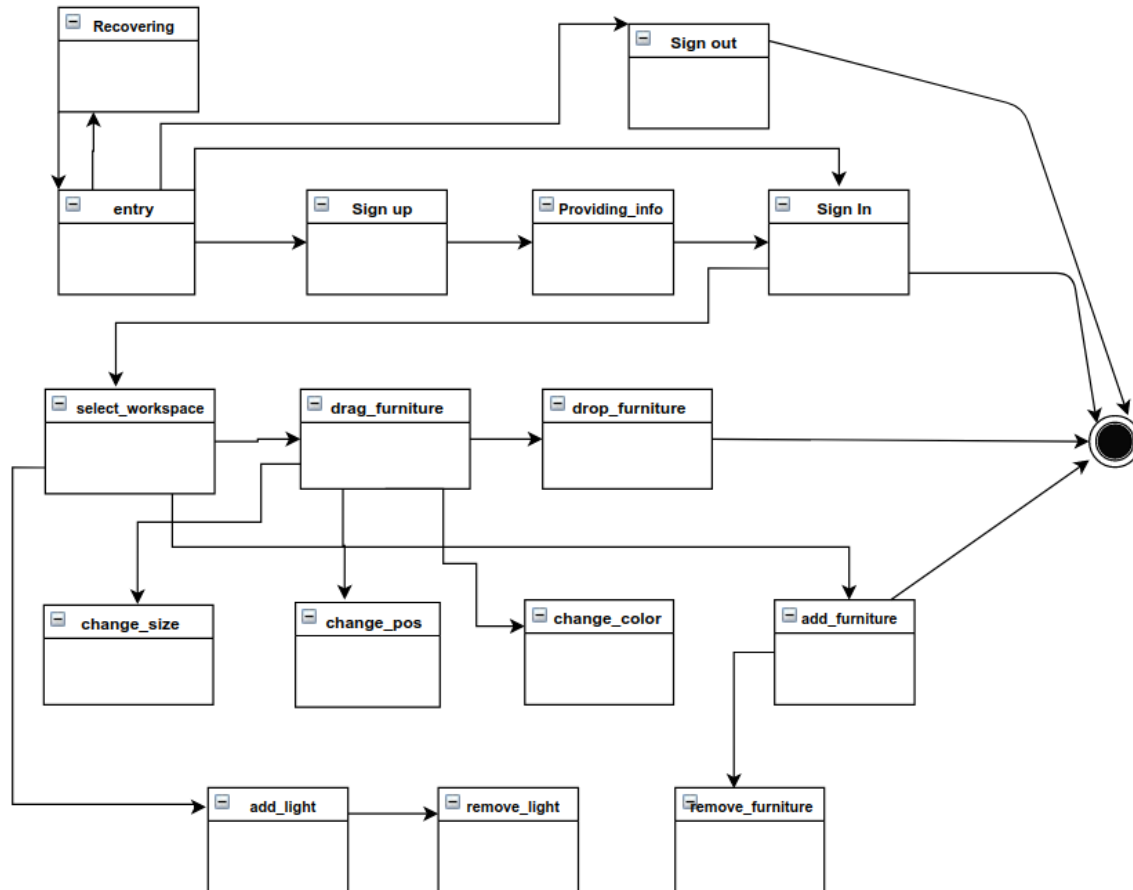


Figure 29: InDez_user

System

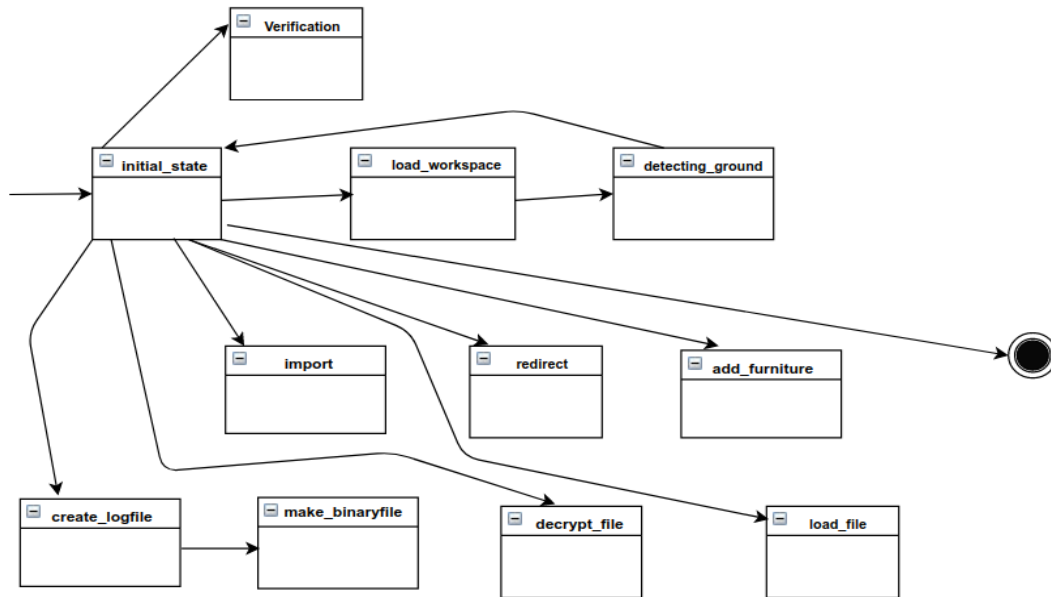


Figure 30: InDez_system

Database

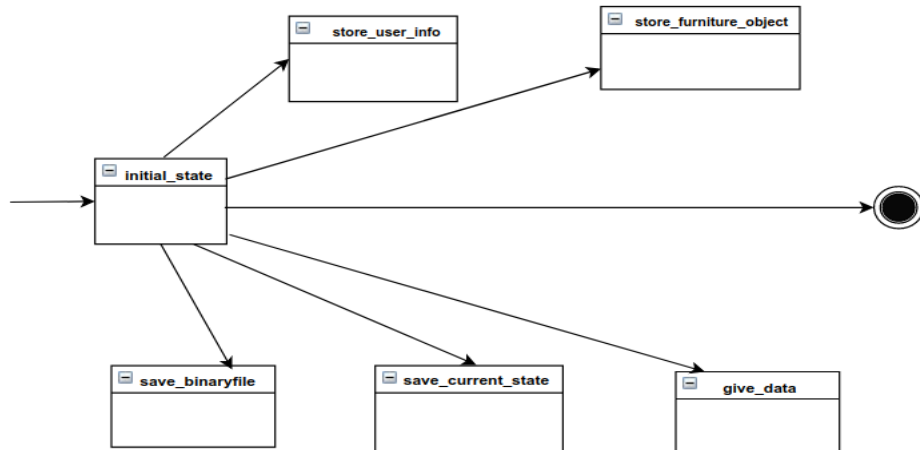


Figure 31: InDez_database

Asset Store

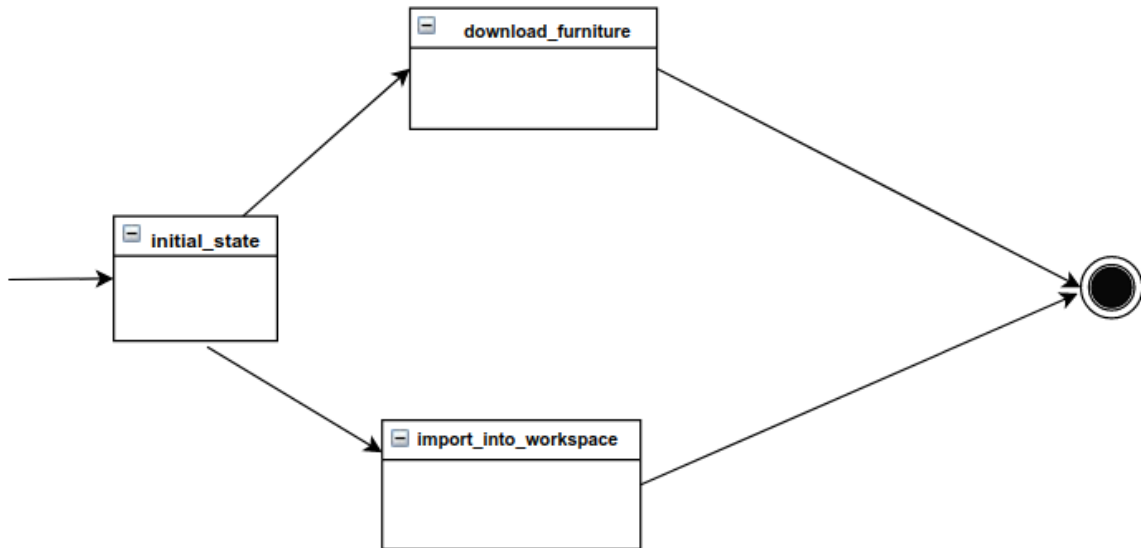


Figure 32: InDez_assetstore

Note: All outward arrows mean system wishes for particular action. After every dead ending state there is an inward arrow to initial state which is omitted for redundancy.

7.2 Sequence Diagram

