



COMSATS University Islamabad (CUI)

**Software Design Description
(SDS DOCUMENT)**

for

InteriorX

Version 1.0

By

Sarah Nasir

CIIT/FA20-BSE-072 /ISB

Warda Rana

CIIT/FA20-BSE-083 /ISB

Supervisor

Sir Atique Ahmed Zafar

Bachelor of Science in Computer Science (2020-2024)

Table of Contents

1. Introduction.....	10
2. Design Methodology and Software Process Model	12
2.1 Design Methodology:.....	12
2.1.1 Rationale behind this approach:	12
2.2 Process Methodology	12
2.2.1 Rationale behind this approach:	12
3. System Overview	13
3.1 Architectural Design	14
3.1.1 Box and Line Diagram:.....	14
3.1.2 Architecture.....	14
4. Design Models	16
4.1 Activity Diagrams	16
4.1.1 Registration	16
4.1.2 Login	17
4.1.3 Search furniture and décor items.....	17
4.1.4 AI powered design recommendation	18
4.1.5 Analyze energy efficiency of items.....	18
4.1.6 Calculate environmental impact.....	19
4.1.7 Collaboration.....	19
4.1.8 Configure Saved designs.....	20
4.1.9 Define user roles and permissions.....	20
4.1.10 Estimate cost of items	21
4.1.11 Explore recycling and upcycling ideas	21
4.1.12 Leave a review and rating	22
4.1.13 Navigate within Virtual Space	22
4.1.14 Optimize design within budget	23
4.1.15 Request guidance from professional interior designers	23
4.1.16 Find Local sustainable suppliers	24
4.1.17 Share Design on social media	24
4.1.18 Save and Manage project	25
4.2 Data Flow Diagrams	25
4.2.1 DFD Level 0	25
4.2.2 DFD Level 1	26
4.2.3 DFD Level 2 Module 1 (User Profile)	27

4.2.4	DFD Level 2 Module 2 (Design Tools)	27
4.2.5	DFD Level 2 Module 3 (Furniture and Décor)	28
4.2.6	DFD Level 2 Module 4 (AI and Designscape)	28
4.2.7	DFD Level 2 Module 5 (Virtual Design Space).....	29
4.2.8	DFD Level 2 Module 6 (Sustainability Information)	29
4.2.9	DFD Level 2 Module 7 (Education and Inspiration)	30
4.2.10	DFD Level 2 Module 8 (Design Project Collaboration)	30
4.3	State Transition Diagram	31
5.	Data Design.....	32
5.1	Data Dictionary	32
6.	Implementation	41
6.1	Algorithm.....	41
6.1.1	SignUp	41
6.1.2	Login:	41
6.1.3	Social Media Integration:	42
6.1.4	Update Profile:	43
6.1.5	Reset Password	43
6.1.6	Expert Articles and advice	44
6.1.7	Design Ideas.....	44
6.1.8	Industry news and updates	45
6.1.9	AI Powered Recommendations:.....	45
6.1.10	Drawing Walls.....	46
6.1.11	3D Rendering of design project:	47
6.2	External APIs/SDK	48
6.3	User Interface.....	49
6.3.1	Login Form	49
6.3.2	Forget Password Page	49
6.3.3	Sign up Page	50
6.3.4	Edit Profile Page	50
6.3.5	Home Page of InteriorX.....	51
6.3.6	Type of project Page	51
6.3.7	Rooms Options Page.....	52
6.3.8	Room Selection	52
6.3.9	Drawing Canvas Page	54

6.3.10	Walls drawn on canvas	54
6.3.11	Doors and windows placement on walls	55
6.3.12	Furniture placement on floor plan	55
6.3.13	3D view of floor plan	56
6.3.14	Textures of walls and floors	56
6.3.15	Education and Inspiration section	57
6.3.16	Articles section	57
6.3.17	Like and Comment on Articles	58
6.3.18	Design Inspiration Ideas section	58
6.3.19	News section	59
6.4	Deployment	59
7.	Testing and Evaluation	60
7.1	Unit Testing	60
	Unit Testing 1:	60
	Unit Testing 2:	61
	Unit Testing 3:	62
	Unit Testing 4:	63
	Unit Testing 5:	63
	Unit Testing 6:	65
7.2	Functional Testing	67
	Functional Testing 1: User Profile verification	67
	Testing Objective:	67
	Functional Testing 2: Design Tools Verification	69
	Testing Objective:	69
	Functional Testing 3: Education and Inspiration verification	70
	Testing Objective:	70
7.3	Scenario Based Testing	71
7.4	Integration Testing	73
	Integration 1:	73
	Testing Objective:	73
	Integration 2:	73
	Testing Objective:	73
	Integration 3:	74
	Testing Objective:	74

Integration 4:	74
Testing Objective:	74
8. Plagiarism Report.....	75

Table of figures

Figure 1 Context Diagram	13
Figure 2 Box and Line Diagram	14
Figure 3 MVC Architecture	15
Figure 4 Registration Activity.....	16
Figure 5 Login Activity.....	17
Figure 6 Search furniture items Activity.....	17
Figure 7 AI recommendations Activity.....	18
Figure 8 Analyze energy efficiency Activity.....	18
Figure 9 Calculate environmental impact Activity	19
Figure 10 Collaboration Activity	19
Figure 11 Configure Saved Designs Activity.....	20
Figure 12 Define roles and permissions Activity.....	20
Figure 13 Estimate cost of items Activity.....	21
Figure 14 Explore recycling ideas Activity.....	21
Figure 15 Leave a review Activity.....	22
Figure 16 Navigate within Virtual Space Activity.....	22
Figure 17 Optimize design within Budget Activity	23
Figure 18 Request Guidance Activity	23
Figure 19 Find local suppliers Activity.....	24
Figure 20 Share Design Activity.....	24
Figure 21 Save and Manage Project Activity.....	25
Figure 22 DFD Level 0 Activity	25
Figure 23 DFD Level 1 (Module 1,2,3,4).....	26
Figure 24 DFD Level 1 (Module 5,6,7,8).....	26
Figure 25 DFD Level 2 User Profile.....	27
Figure 26 DFD Level 2 Design Tools.....	27
Figure 27 DFD Level 2 Furniture and Décor.....	28
Figure 28 DFD Level 2 AI and Designscape	28
Figure 29 DFD Level 2 Virtual Design Space	29
Figure 30 DFD Level 2 Sustainability Information.....	29
Figure 31 DFD Level 2 Education and Inspiration.....	30
Figure 32 DFD Level 2 Design Project Collaboration.....	30

Figure 33 State Transition Diagram VR.....	31
Figure 34 Login Form.....	49
Figure 35 Forget Password Page.....	49
Figure 36 Sign up Page.....	50
Figure 37 Edit Profile Page.....	50
Figure 38 Home Page.....	51
Figure 39 Type of Project Page.....	51
Figure 40 Rooms Options Page	52
Figure 41 Room Selection Page 1.....	52
Figure 42 Room Selection Page 2.....	53
Figure 43 Room Selection Page 3.....	53
Figure 44 Drawing Canvas Page.....	54
Figure 45 Walls drawn on Canvas Page.....	54
Figure 46 Doors and Window placement.....	55
Figure 47 Furniture Placement.....	55
Figure 48 3D view of floor plan.....	56
Figure 49 Texture of walls and floors	56
Figure 50 Education and Inspiration Section.....	57
Figure 51 Articles Section.....	57
Figure 52 Like and Comment on Articles.....	58
Figure 53 Design Inspiration Ideas Section	58
Figure 54 News Section	59
Figure 55 Plagiarism Report	75

Revision History

Name	Date	Reason for changes	Version

Application Evaluation History

Comments (by committee) *include the ones given at scope time both in doc and presentation	Action Taken

Supervised by
Supervisor's Name

Signature_____

1. Introduction

The InteriorX project, utilizing the MERN (MongoDB, Express.js, React.js, Node.js) stack, is making substantial progress in its development phase. As of now, the completed and implemented modules include **User Profile (Module 1)**, covering user registration, login, preferences, history, and social media integration; **Design Tools (Module 2)**, encompassing room and floor plan creation, customizable wall colors and textures, window and door placement, flooring options, and design project management; and **Education and Inspiration (Module 7)**, featuring tutorials, design inspiration, expert advice, industry news, and user-generated content, with access to professional interior designers pending. MongoDB serves as the data store, managing user data and design projects. The MERN stack ensures a robust, scalable, and user-centric web application for virtual interior design, promising a seamless and innovative experience for users seeking to visualize and create their dream home interiors.

The overall modules of our system are as follows:

Module 1: User Profile

- FE-1: User registration and login
- FE-2: User preferences and settings
- FE-3: User history and saved designs
- FE-4: Social media integration
- FE-5: Session Management

Module 2: Design Tools

- FE-1: Room and floor plan creation
- FE-2: Customizable wall colors and textures
- FE-3: Window and door placement
- FE-4: Flooring options
- FE-5: Room layout and furniture placement
- FE-6: Design project management

Module 3: Furniture and Décor

- FE-1: Library of 3D furniture and décor items
- FE-2: Filter and search functionality
- FE-3: Budget optimization
- FE-4: Wishlist and favourites
- FE-5: User Reviews and Ratings

Module 4: AI and Designscape

FE-1: AI based furniture budget prediction

FE-2: 3D rendering of 2D design space

FE-3: AI-powered design recommendations

FE-4: Trend and inspiration boards

Module 5: Virtual Design Space

FE-1: VR look of the 3D design space

FE-2: Moving through the virtual design space

FE-3: Feedback on VR experience

Module 6: Sustainability Information

FE-1: Environmental Impact Calculator

FE-2: Material Eco-Friendliness Ratings

FE-3: Energy Efficiency Analysis

FE-4: Recycling and Up-cycling Ideas

FE-5: Local Sourcing Information

Module 7: Education and Inspiration

FE-1: Tutorials and how-to guides

FE-2: Design inspiration and ideas

FE-3: Expert advice and articles

FE-4: Industry news and updates

FE-5: User-generated content and community features

FE-6: Access to professional interior designers

Module 8: Design projects collaboration:

FE-1: User Accounts and Permissions

FE-2: Real-Time Collaboration

FE-3: Chat/messaging feature for collaborators to communicate

FE-4: Feedback Surveys

FE-5: Annotations and Comments

FE-6: Collaborative Ratings on furniture items

2. Design Methodology and Software Process Model

2.1 Design Methodology:

In developing InteriorX, we've adopted a **procedural approach** as our chosen design methodology.

2.1.1 Rationale behind this approach:

The procedural approach offers tangible advantages for our web application development such as:

- It ensures a clear and organized code structure, facilitating the modular implementation of various design features.
- The step-by-step nature of procedural programming enhances code readability. This readability is valuable for collaboration among developers, especially in larger web development projects.
- New features can be added by developing additional procedures, and existing ones can be modified without overhauling the entire codebase.
- Debugging can be done on a smaller scale, isolating specific procedures for more efficient problem-solving.
- Its resource efficiency and scalability support the ongoing development of InteriorX, aligning with our goal of creating a user-friendly and efficient virtual interior design platform.

2.2 Process Methodology

In developing InteriorX, we've adopted **Agile Scrum** for process methodology.

2.2.1 Rationale behind this approach:

Agile Scrum is like breaking our project into smaller parts, called sprints. It helps us focus on specific goals and use resources wisely. Team members work on these sprints, and all the parts come together to make our complete software.

It will provide us with the following benefits:

- **Transparency through Agile Scrum:**
Agile Scrum ensures that team members have clear and up-to-date information, making it easier to make smart decisions and work together seamlessly.
- **Adaptable and Useful Software:**
By building our software bit by bit, we can easily make changes to keep it useful and in line with what's most important at the time.
- **User-Friendly Design:**
Getting feedback regularly helps us make our software easy to understand and use, creating a product that people find intuitive and friendly.

- **Team Engagement and Ownership:**

Each team member focuses deeply on specific tasks, becoming experts in those areas and feeling a strong sense of responsibility for the success of the project.

3. System Overview

InteriorX is a virtual interior design web application targeting homeowners and design enthusiasts who seek a user-friendly platform to effortlessly design and visualize their dream home interiors. The motivation behind InteriorX stems from the challenges faced by users in existing design apps, including limited customization, inadequate furniture options, and a lack of sustainability information. To address these issues, InteriorX is designed to empower users with extensive customization features, a diverse selection of furniture and décor items, AI-powered design suggestions, and real-time cost estimation. The system facilitates collaboration through real-time project editing and integrated chat, ensuring seamless communication among users. Moreover, InteriorX incorporates virtual reality and visualization techniques to offer precise and realistic representations of designs in actual spaces, mitigating the risk of disappointment. The system also prioritizes sustainability by providing eco- friendliness information and local sourcing details. With a user-friendly interface and comprehensive tutorials, InteriorX aims to simplify the design process, making it accessible and enjoyable for users to create and visualize their ideal home interiors with confidence and ease.

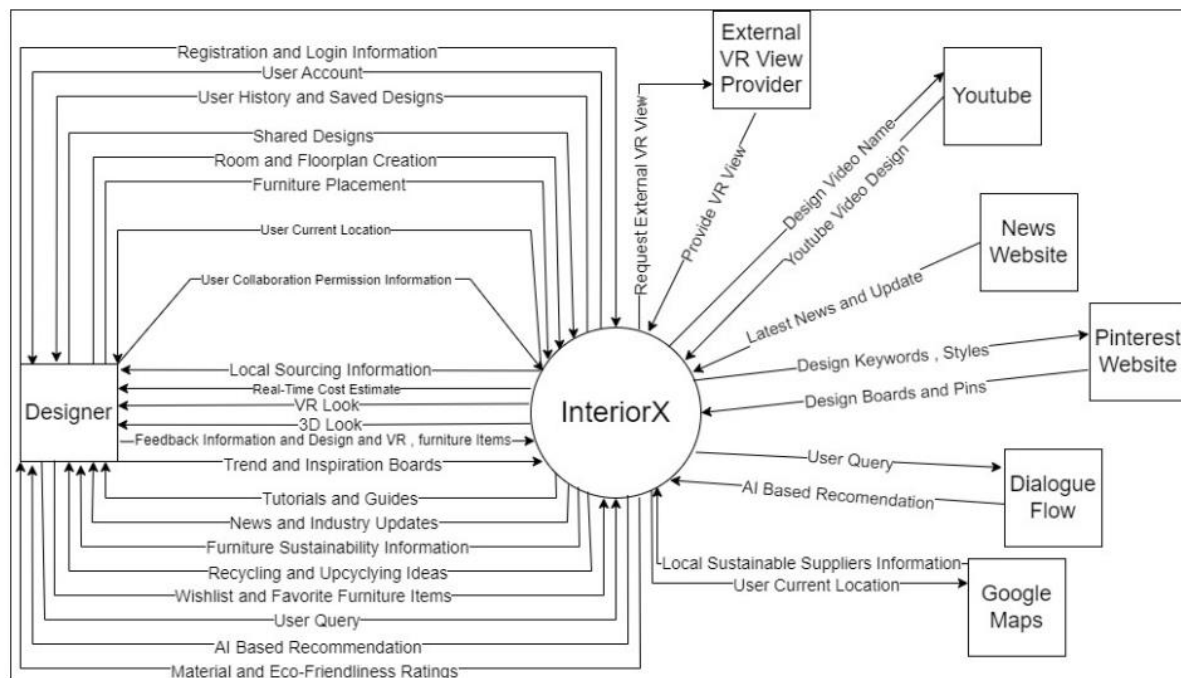


Figure 1 Context Diagram

3.1 Architectural Design

3.1.1 Box and Line Diagram:

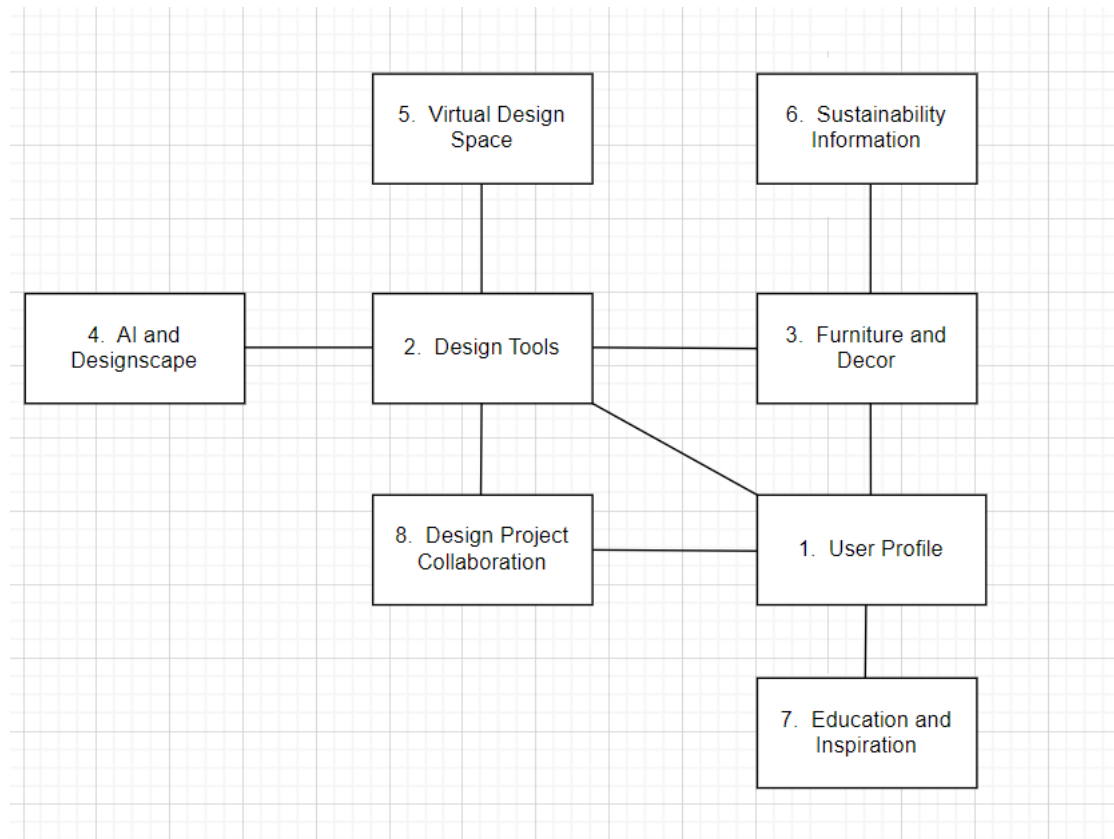


Figure 2 Box and Line Diagram

3.1.2 Architecture

3.1.2.1 MVC:

In building InteriorX, we organized our web app using MVC (Model-View-Controller) architecture. It's like dividing tasks into three parts:

Controller:

The Controller serves as a crucial intermediary, handling user requests and ensuring seamless communication between the Model and View components. Its primary responsibility lies in directing requests to the Model for data-related tasks, maintaining a pivotal role in coordinating the flow of information.

Model:

The Model takes the lead in data operations, assuming responsibility for tasks such as validation, updating, and deleting data. Its primary engagement is with the database, focusing solely on these data-related operations. The Model remains unburdened by concerns related to user interactions, allowing for a dedicated concentration on managing and manipulating data.

View:

After the Model completes data processing, the View collaborates with the Controller to dynamically present information to users. With a primary focus on visual presentation, the View excels at dynamically rendering HTML templates, contributing to a visually pleasing and user-friendly interface that enhances the overall user experience.

Rationale behind Choosing MVC:

We chose the MVC architecture for InteriorX for some good reasons:

- Since we're taking an agile approach to develop the app, changes are expected in the system. The MVC architecture proves to be the best fit for web apps in this scenario because it neatly separates concerns. Particularly, if adjustments are needed in the user interface, MVC supports multiple views without getting tangled up in the business logic, making it adaptable to changes.
- MVC brings in loose coupling by dividing functionality, logic, and the user interface. This separation is handy for collaborative development, allowing multiple developers to work on the same project simultaneously. For instance, one can focus on the Controller, another on the Model, and someone else on the View, streamlining the development process.
- From a testing perspective, MVC's separation of concerns makes the system easy to test. Each component—Controller, Model, and View—can be tested independently, simplifying the testing process and ensuring the robustness of the overall system.

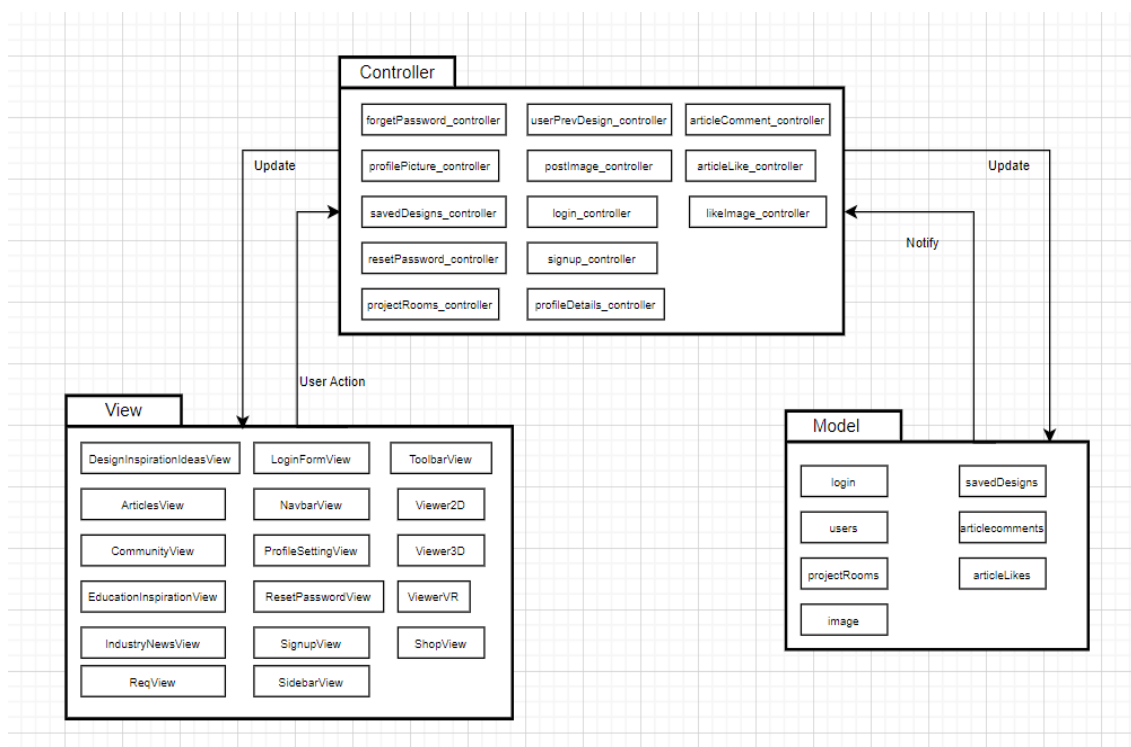


Figure 3 MVC Architecture

4. Design Models

4.1 Activity Diagrams

4.1.1 Registration

This diagram shows the process of registration that how the user will register his account and then the credentials will be checked if valid, then success message will be shown. If invalid, respected error message will be shown.

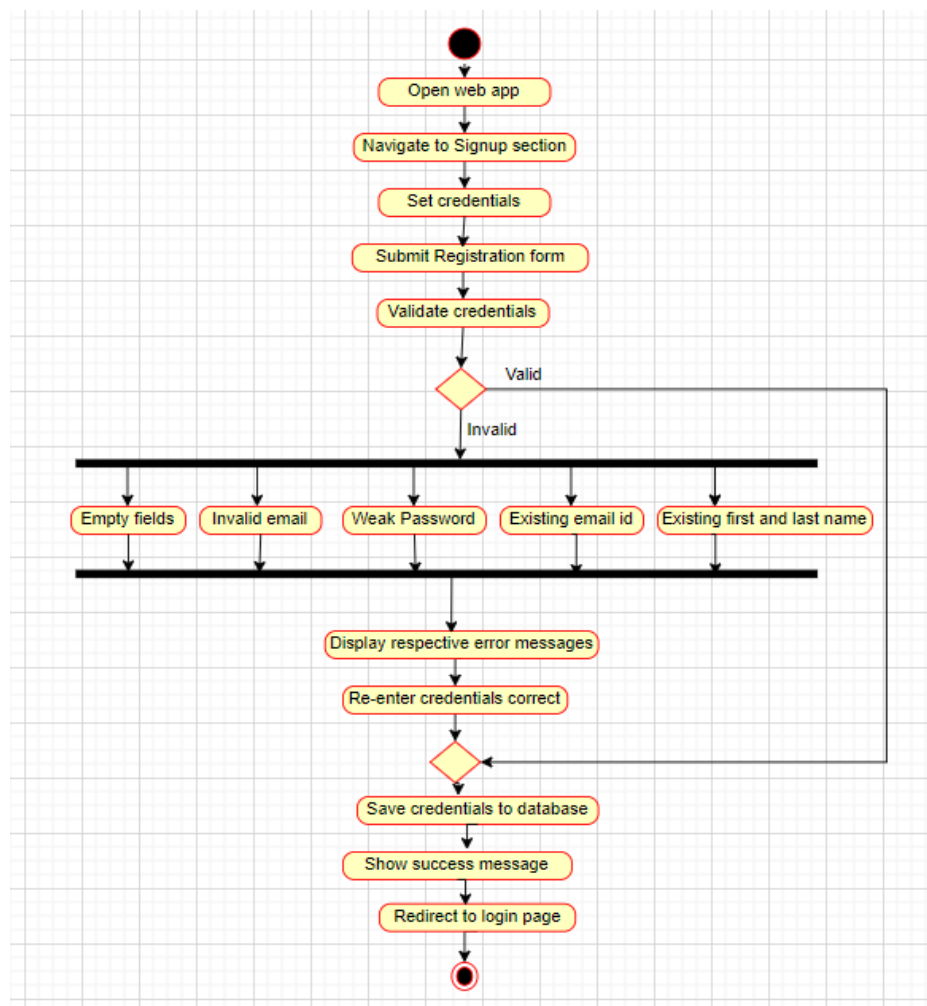


Figure 4 Registration Activity

4.1.2 Login

This diagram shows the process of Login, that how the user logs the system and then the credentials are checked if valid, user will be logged in and JWT token is generated. If invalid, the error message will be shown.

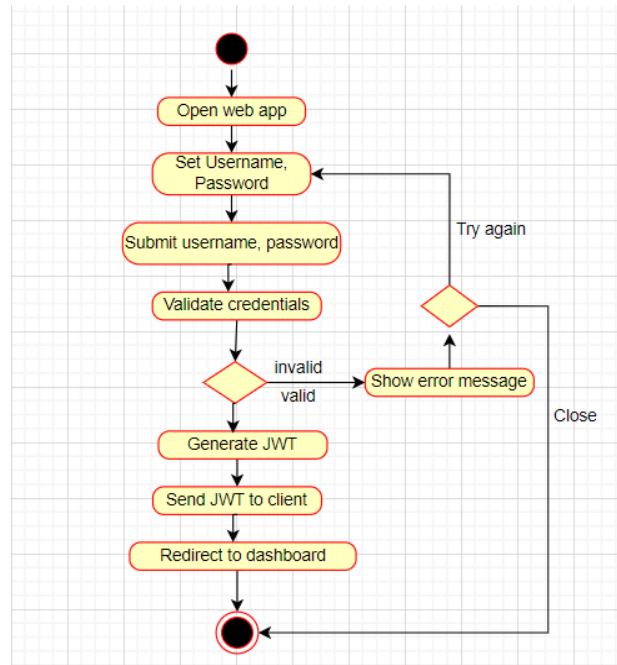


Figure 5 Login Activity

4.1.3 Search furniture and décor items

This diagram shows the process of searching items from the catalog section. The user will enter the item name and then items will be searched based on the input.

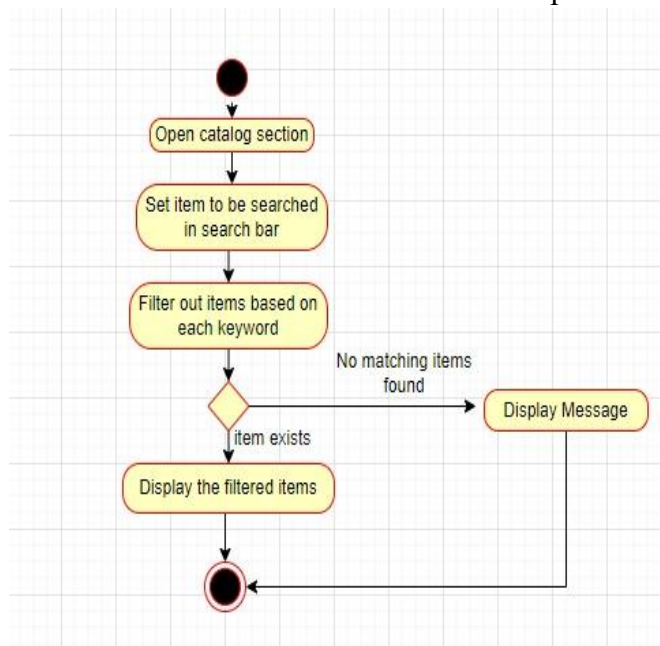


Figure 6 Search furniture items Activity

4.1.4 AI powered design recommendation

This diagram shows the process of AI powered design recommendations, the user will enter the question in the chatbot and then the response will be given to user from dialogue flow.

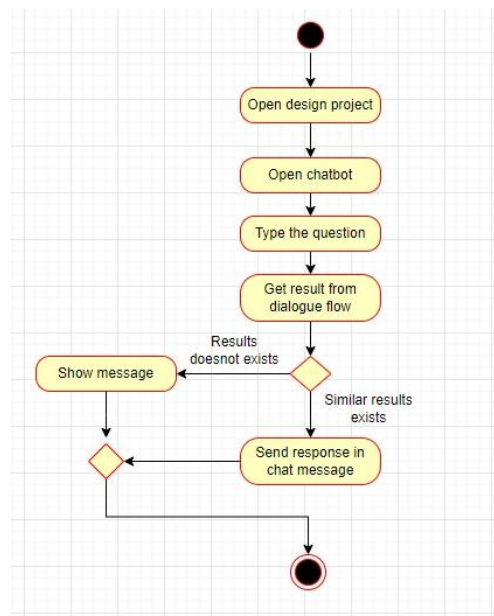


Figure 7 AI recommendations Activity

4.1.5 Analyze energy efficiency of items

This diagram shows the process of analyzing energy efficiency of items. The energy efficiency of items such as electricity appliances or heating/cooling systems placed in the design project is analyzed and reports are generated for each item. Also, the overall energy consumption is displayed.

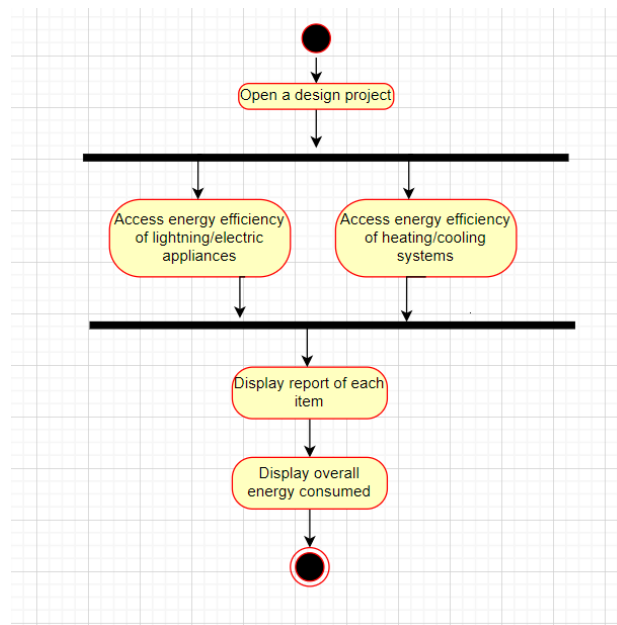


Figure 8 Analyze energy efficiency Activity

4.1.6 Calculate environmental impact

This diagram shows the process of calculating environmental impact, the user will prompt the type and quantity of material and then the environmental impact is calculated, and report is displayed. The user can also save the report.

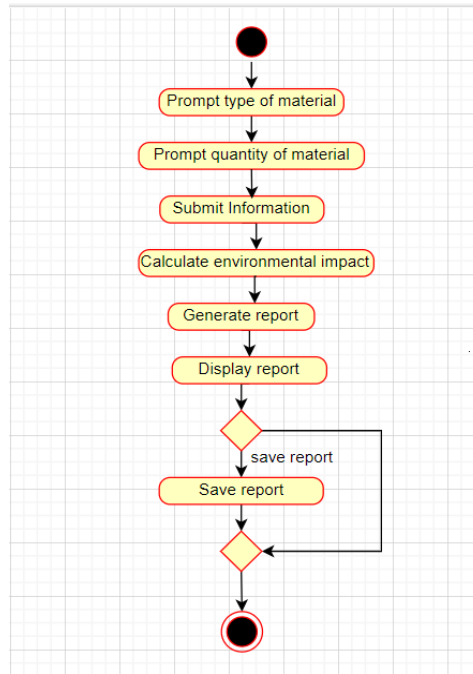


Figure 9 Calculate environmental impact Activity

4.1.7 Collaboration

This diagram shows the process of real-time collaboration, users sets roles and permissions, can add comments to items, provide ratings and communicate with other users. At the end feedback is collected from every user.

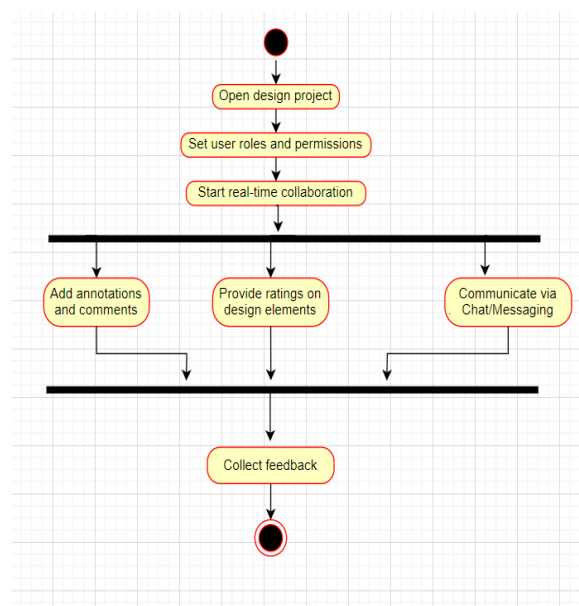


Figure 10 Collaboration Activity

4.1.8 Configure Saved designs

This diagram shows the process of configuring saved designs. User can open the previous designs from the saved design section, can render that design on canvas and can do editing in it.

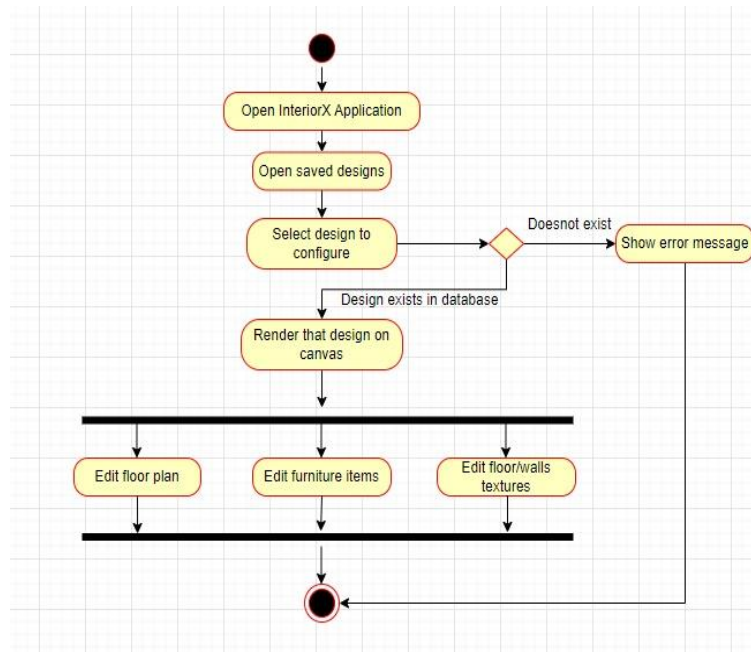


Figure 11 Configure Saved Designs Activity

4.1.9 Define user roles and permissions

This diagram shows the process of defining user roles and permissions. The user can define the roles such as project lead or designer and set permissions such as read, write, read/write.

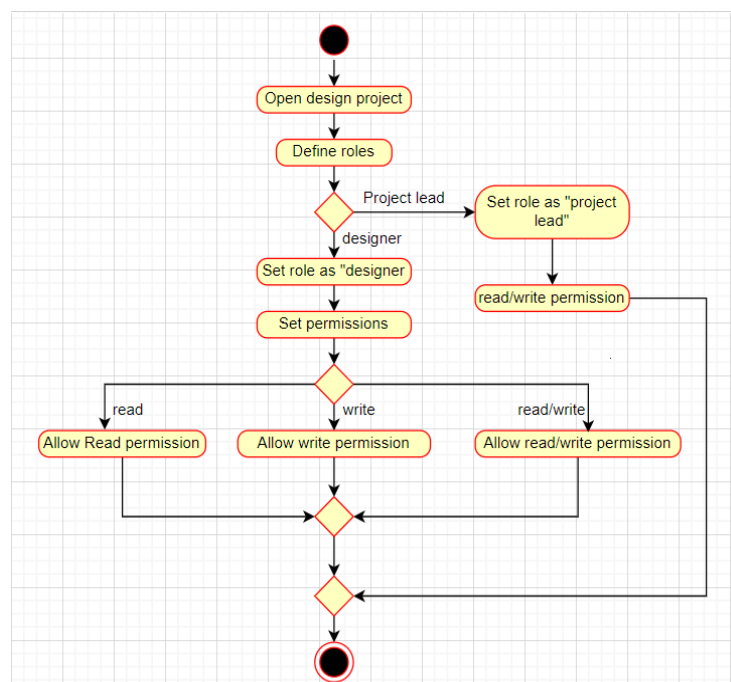


Figure 12 Define roles and permissions Activity

4.1.10 Estimate cost of items

This diagram shows the process of estimating cost of items. The user can provide the furniture description in the furniture prediction section. Then, the price of that item is estimated and displayed.

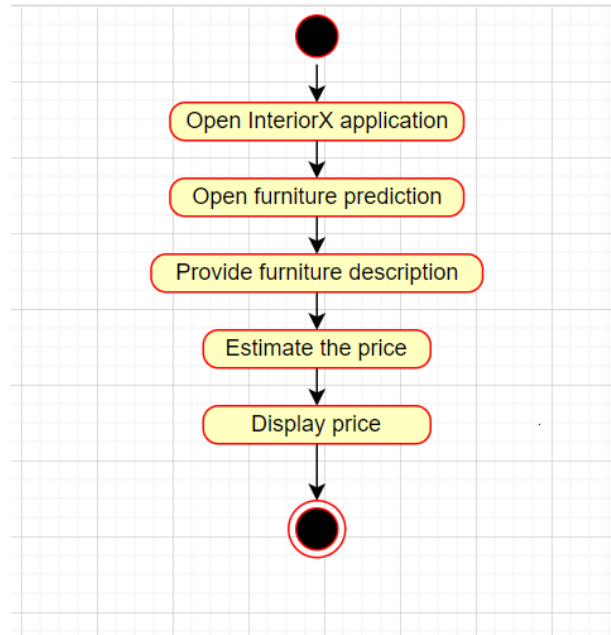


Figure 13 Estimate cost of items Activity

4.1.11 Explore recycling and upcycling ideas

This diagram shows the process of exploring recycling and upcycling ideas. The user can view recycling or upcycling ideas from the catalog section.

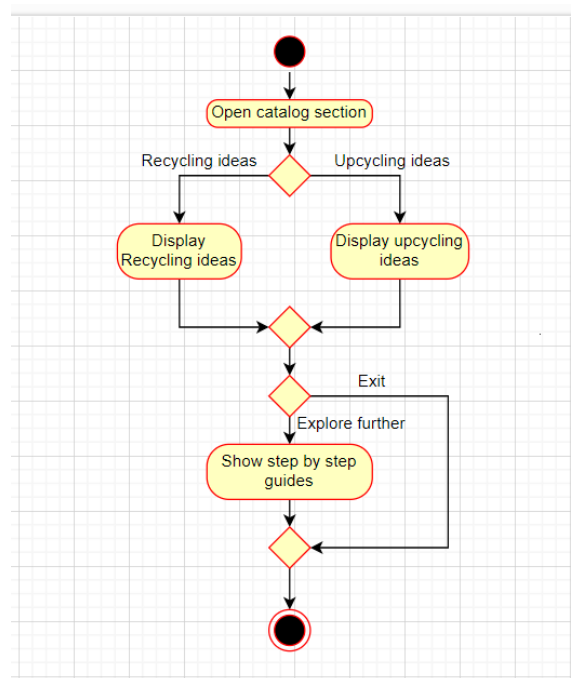


Figure 14 Explore recycling ideas Activity

4.1.12 Leave a review and rating

This diagram shows the process of leaving review or rating on items. The user can add rating or set reviews on the items in catalog section. These are then submitted, stored in the database and also shown under the item.

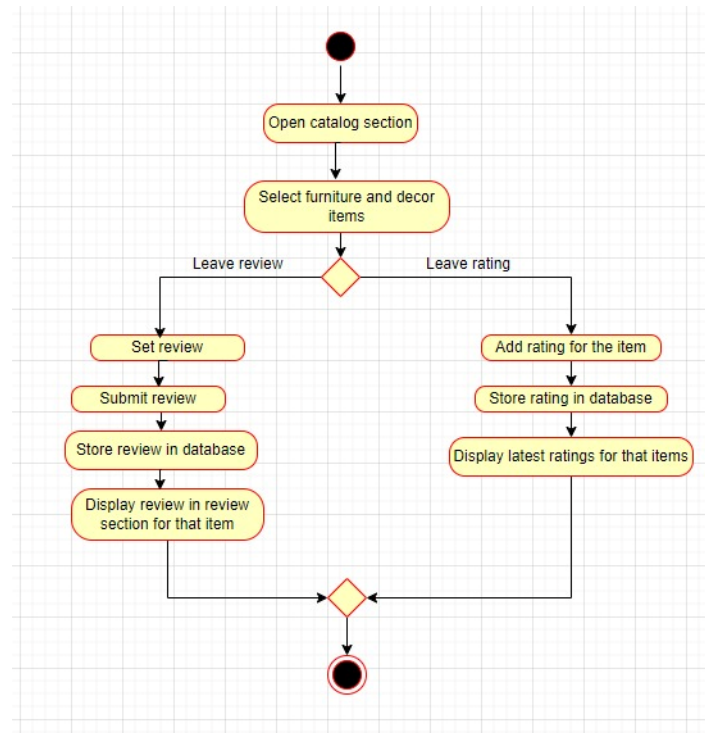


Figure 15 Leave a review Activity

4.1.13 Navigate within Virtual Space

This diagram shows the process of navigation within Virtual Space. The user can enter VR design space and then can move in all directions to view the design clearly.

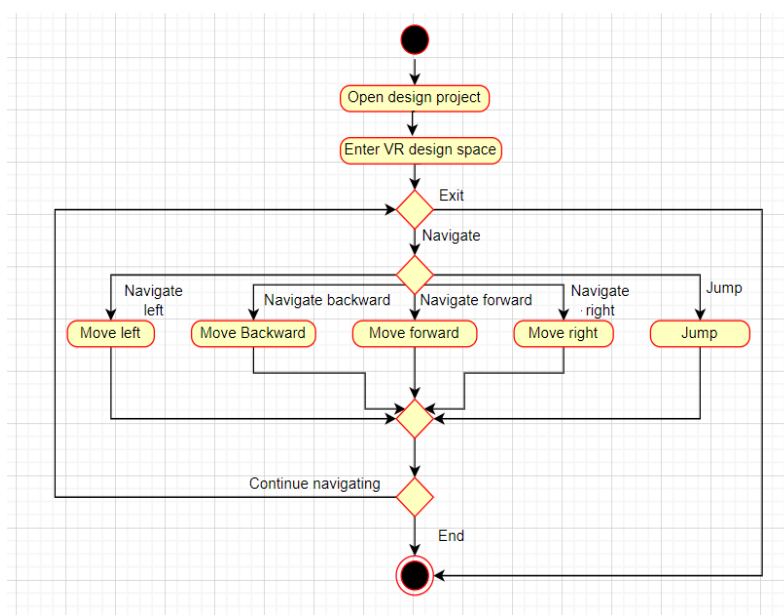


Figure 16 Navigate within Virtual Space Activity

4.1.14 Optimize design within budget

This diagram shows the process of optimizing design within budget. The price of all the items which are placed in the design project are displayed to the user. Also, the total budget of project is displayed.

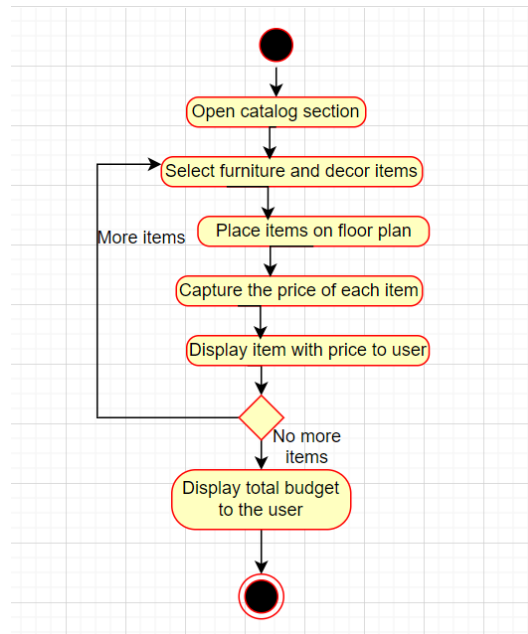


Figure 17 Optimize design within Budget Activity

4.1.15 Request guidance from professional interior designers

This diagram shows the process of requesting guidance from professionals. The user can choose any professional from the list and can contact with him/her via voice call, video call or message.

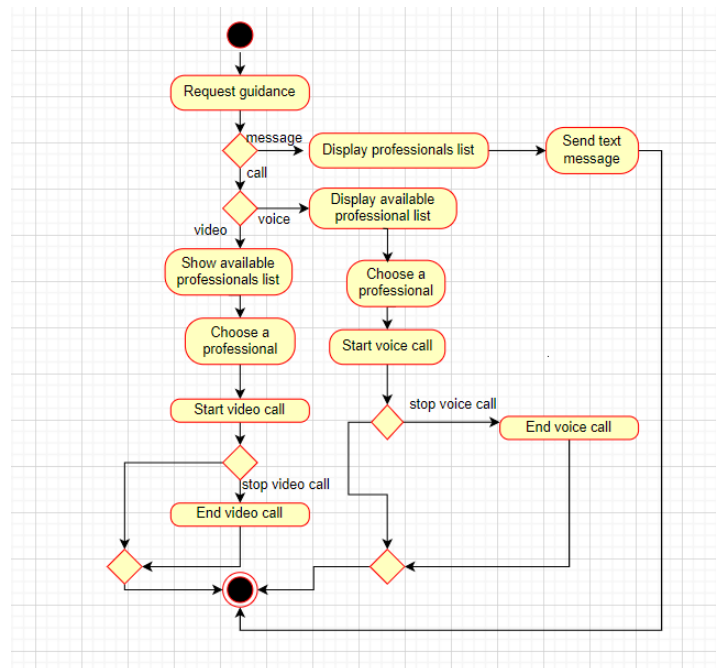


Figure 18 Request Guidance Activity

4.1.16 Find Local sustainable suppliers

This diagram shows the process of finding local suppliers. The system shows the nearby sustainable suppliers based on the user's location. Then, the user can browse details or contact any supplier.

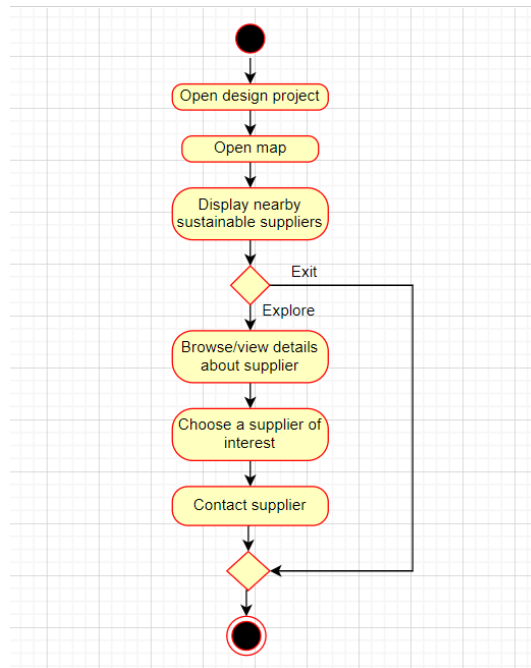


Figure 19 Find local suppliers Activity

4.1.17 Share Design on social media

This diagram shows the process of sharing design on social media. The user can post his/her design in the community section.

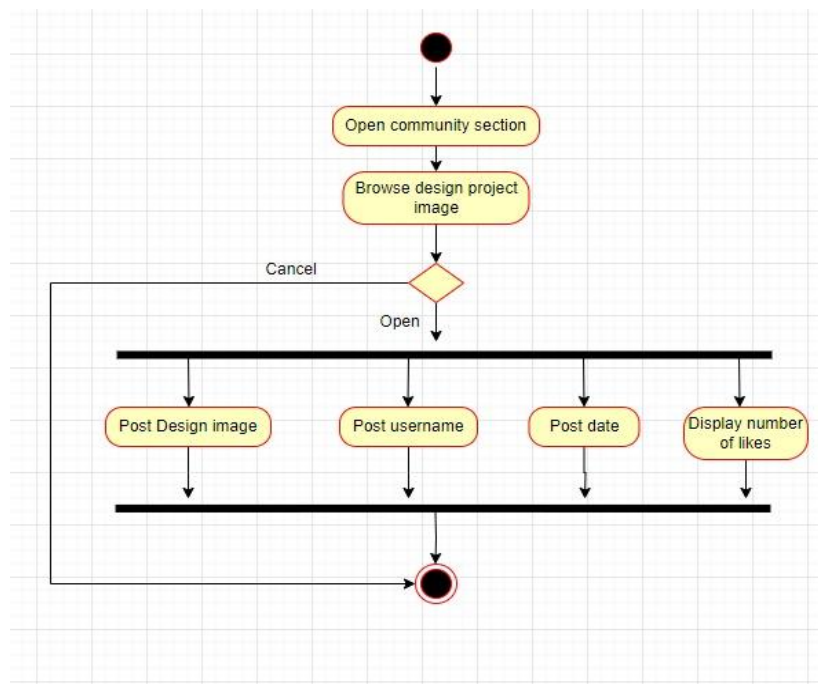


Figure 20 Share Design Activity

4.1.18 Save and Manage project

This diagram shows the process of saving and managing project. The user can save the design project and can load and edit previous design projects from saved designs section.

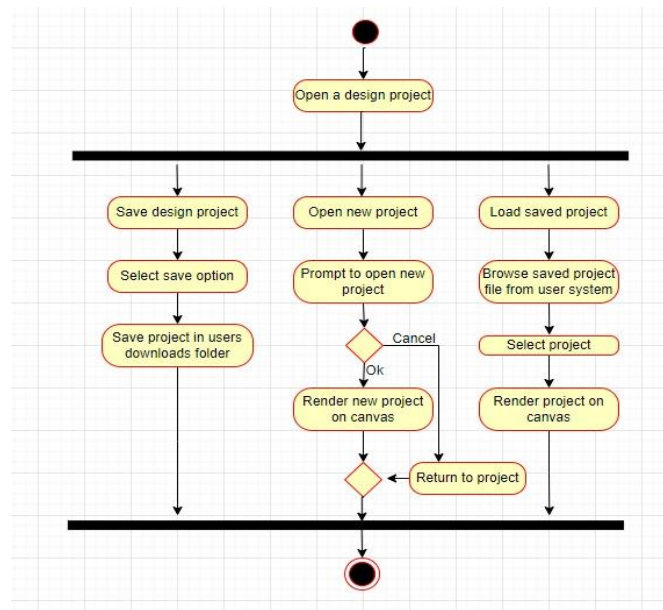


Figure 21 Save and Manage Project Activity

4.2 Data Flow Diagrams

4.2.1 DFD Level 0

The following figure shows the level 0 data flow diagram of the system. It shows the major entities of the system and their dataflow. It tells what data flows in and out of the system and to which entities. It is at higher level of abstraction.

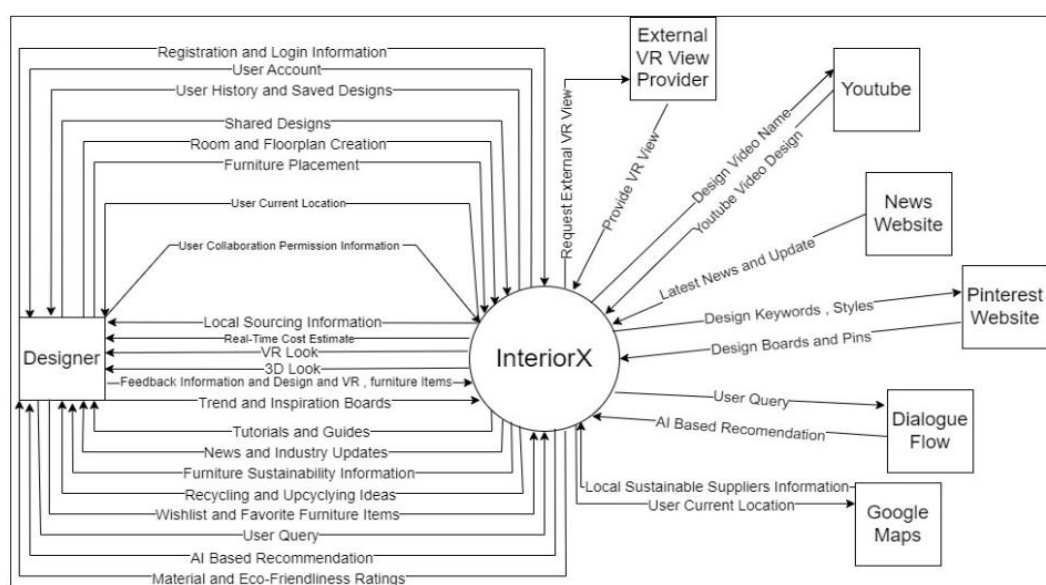


Figure 22 DFD Level 0 Activity

4.2.2 DFD Level 1

The following figures show the level 1 data flow diagrams of the system. They show different modules of the system as processes, entities, the data stores, and data flow. They also show how different modules communicate with each other.

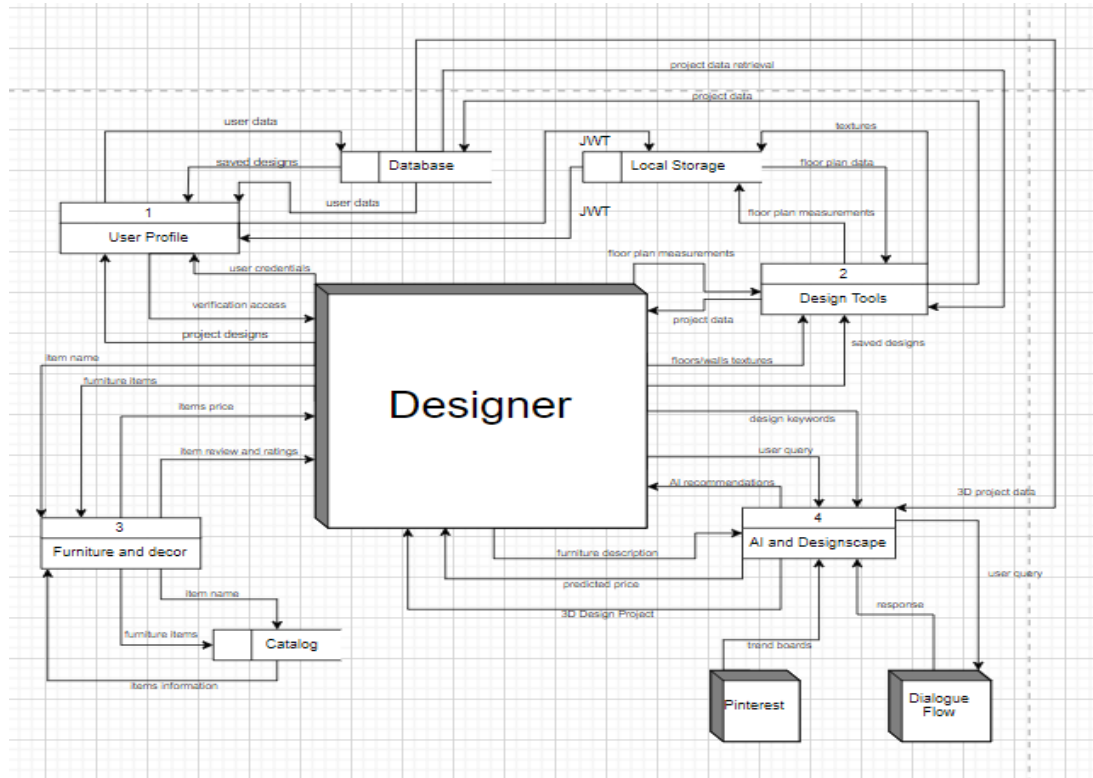


Figure 23 DFD Level 1 (Module 1,2,3,4)

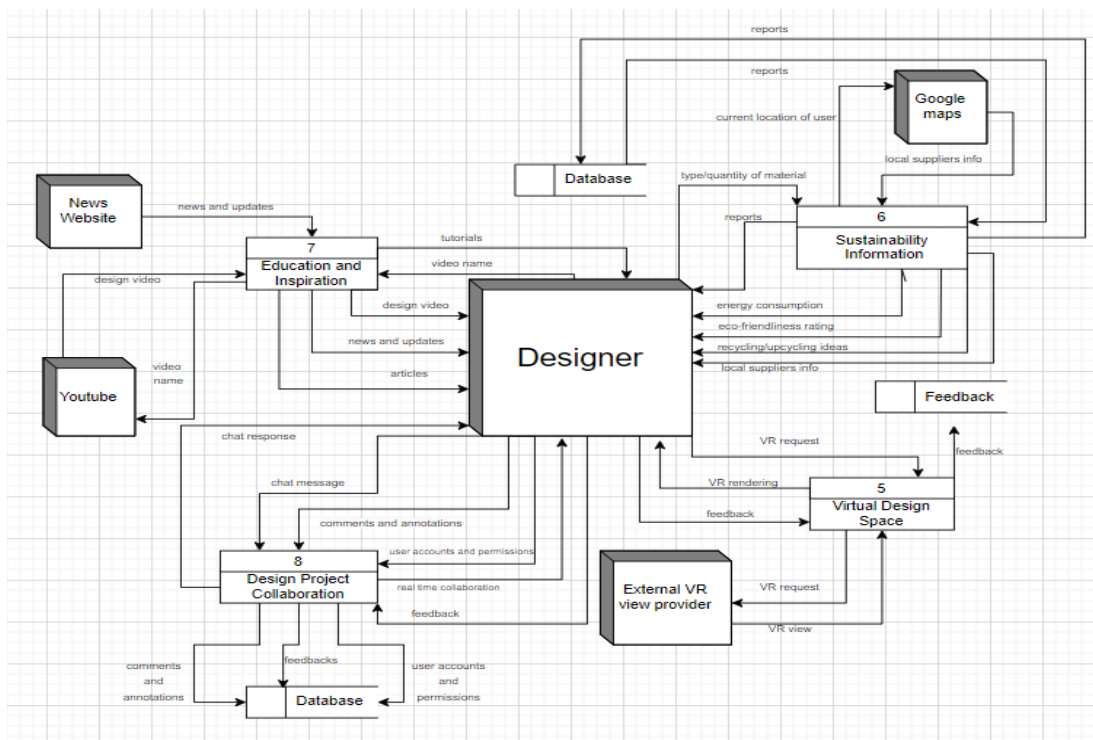


Figure 24 DFD Level 1 (Module 5,6,7,8)

4.2.3 DFD Level 2 Module 1 (User Profile)

The following figure shows level 2 data flow diagram for User Profile module. It shows the processes of the module and how data flows between the processes, datastore and entities.

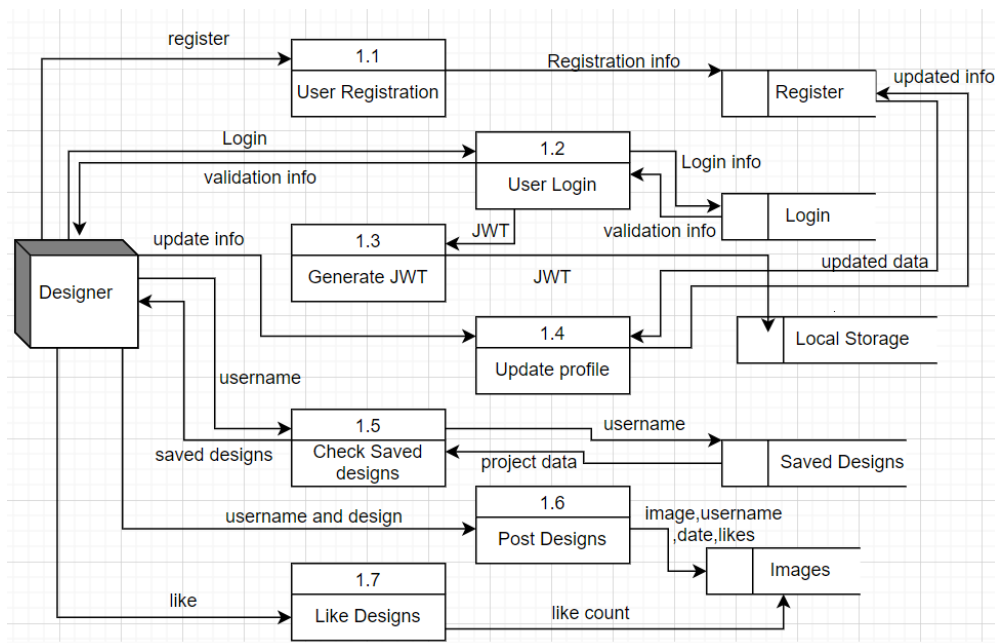


Figure 25 DFD Level 2 User Profile

4.2.4 DFD Level 2 Module 2 (Design Tools)

The following figure shows level 2 data flow diagram for Design Tools module. It shows the processes of the module and how data flows between the processes, datastore and entities.

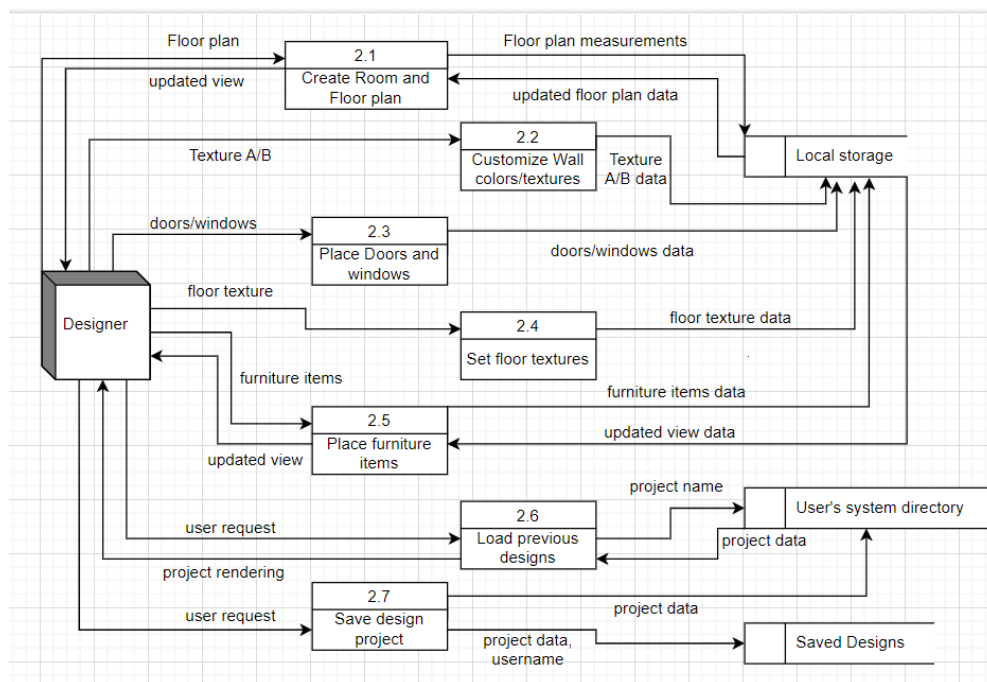


Figure 26 DFD Level 2 Design Tools

4.2.5 DFD Level 2 Module 3 (Furniture and Décor)

The following figure shows level 2 data flow diagram for Furniture and Décor module. It shows the processes of the module and how data flows between the processes, datastore and entities.

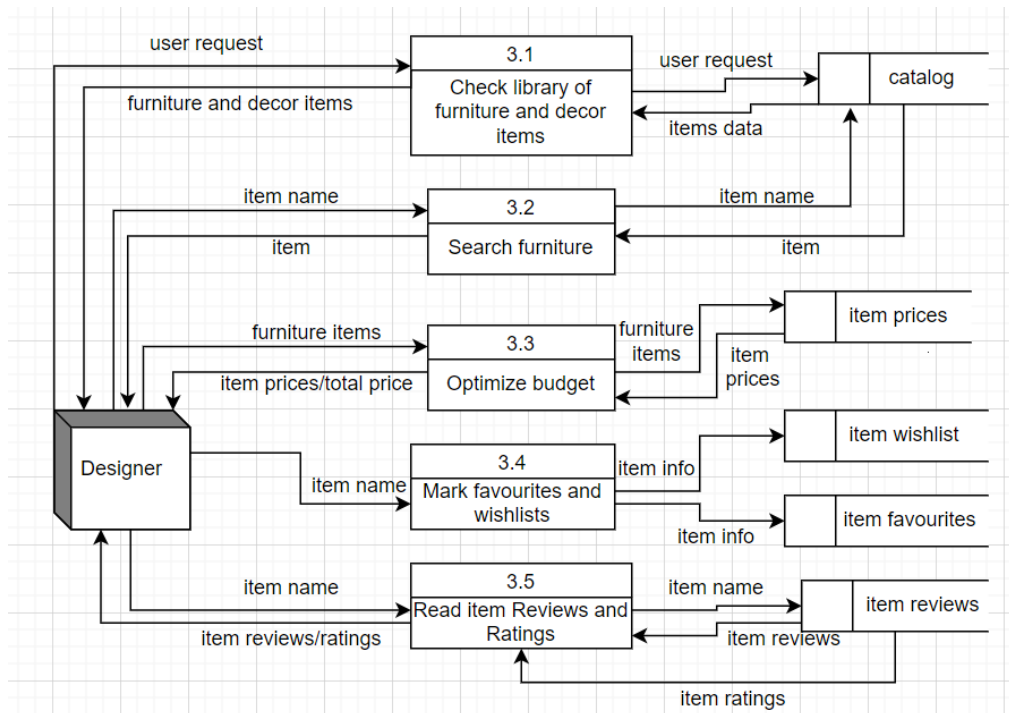


Figure 27 DFD Level 2 Furniture and Décor

4.2.6 DFD Level 2 Module 4 (AI and Designscape)

The following figure shows level 2 data flow diagram for AI and Designscape module. It shows the processes of the module and how data flows between the processes, datastore and entities.

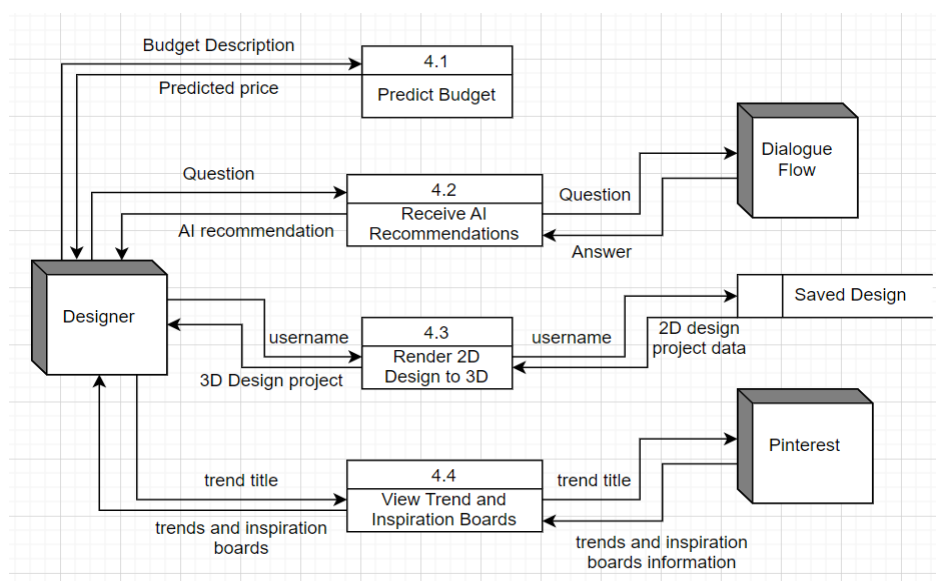


Figure 28 DFD Level 2 AI and Designscape

4.2.7 DFD Level 2 Module 5 (Virtual Design Space)

The following figure shows level 2 data flow diagram for Virtual Design Space module. It shows the processes of the module and how data flows between the processes, datastore and entities.

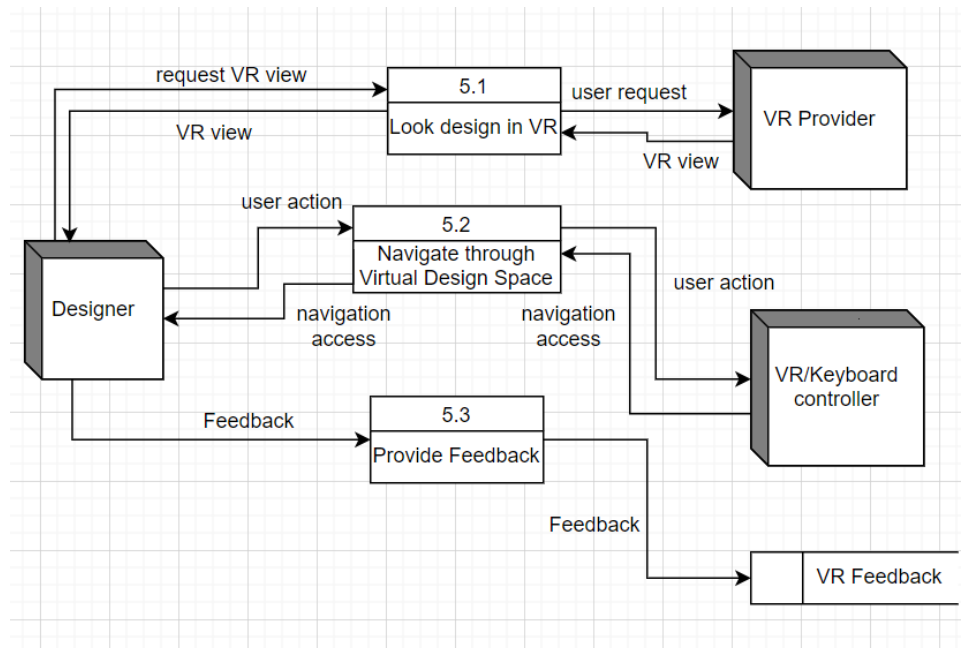


Figure 29 DFD Level 2 Virtual Design Space

4.2.8 DFD Level 2 Module 6 (Sustainability Information)

The following figure shows level 2 data flow diagram for Sustainability Information module. It shows the processes of the module and how data flows between the processes, datastore and entities.

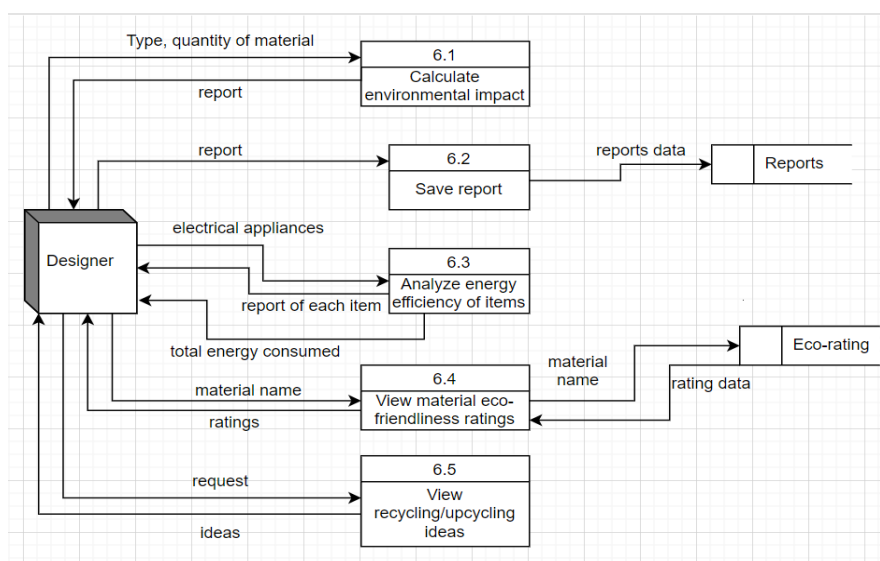


Figure 30 DFD Level 2 Sustainability Information

4.2.9 DFD Level 2 Module 7 (Education and Inspiration)

The following figure shows level 2 data flow diagram for Education and Inspiration module. It shows the processes of the module and how data flows between the processes, datastore and entities.

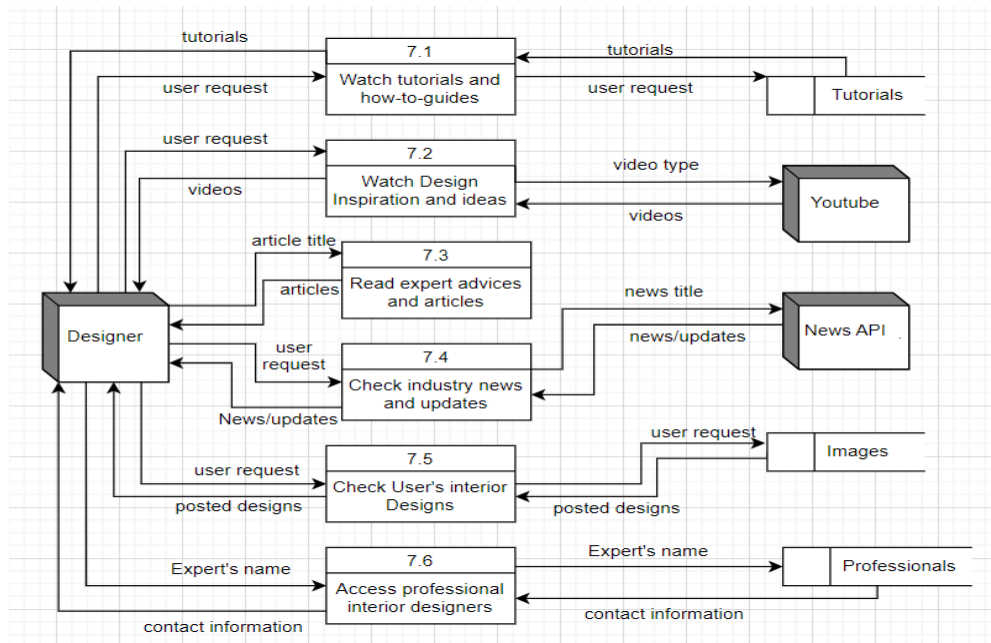


Figure 31 DFD Level 2 Education and Inspiration

4.2.10 DFD Level 2 Module 8 (Design Project Collaboration)

The following figure shows level 2 data flow diagram for Design Project Collaboration module. It shows the processes of the module and how data flows between the processes, datastore and entities.

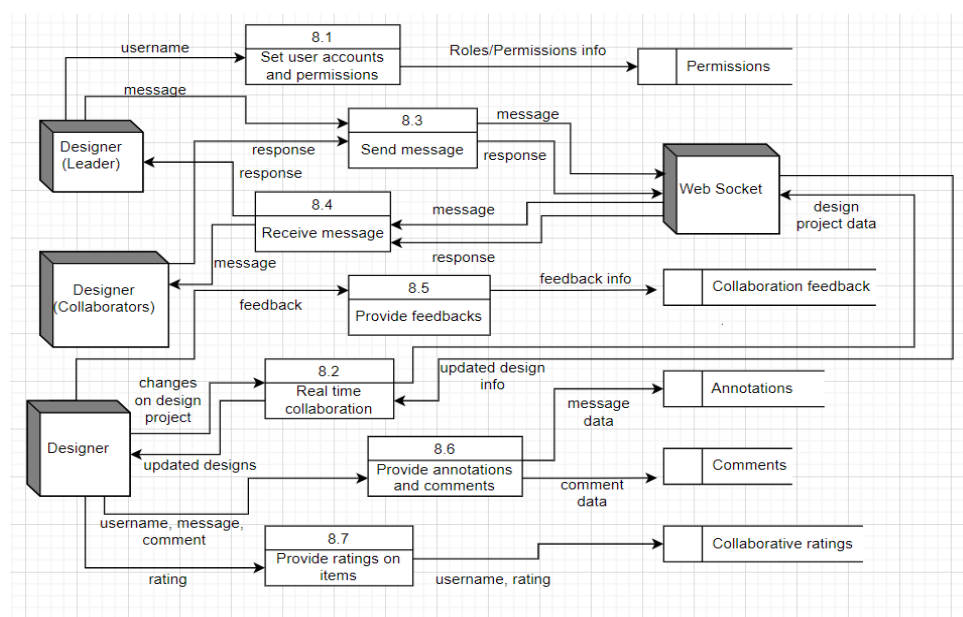


Figure 32 DFD Level 2 Design Project Collaboration

4.3 State Transition Diagram

This diagram shows the transition between the VR events

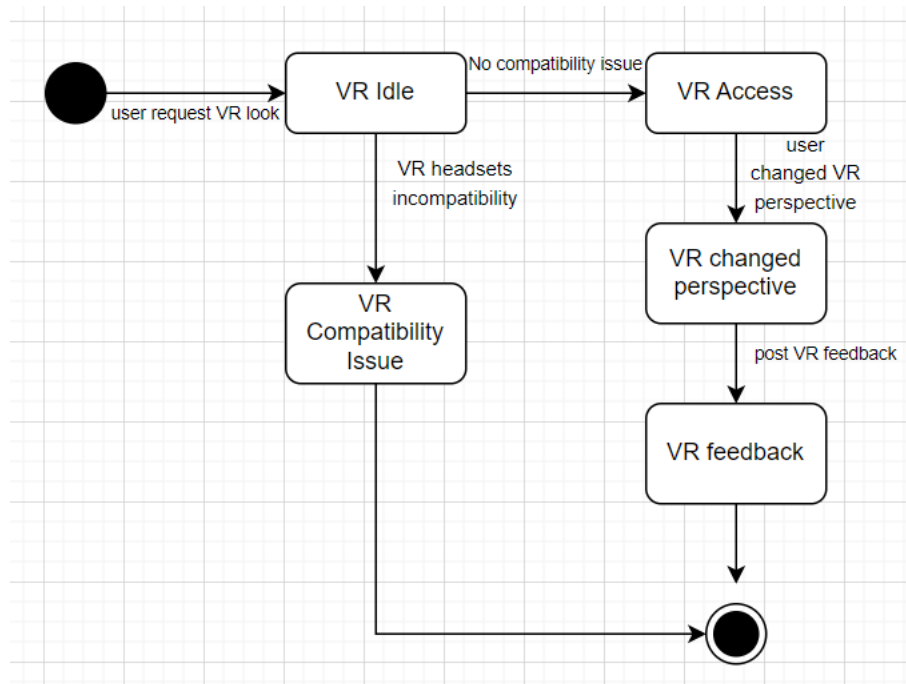


Figure 33 State Transition Diagram VR

5. Data Design

The database used in our application is MongoDB. MongoDB is a cross-platform, documentbased NoSQL database, that facilitates the storage of data in a flexible BSON (Binary JSON) type format. The database of this system is being deployed on MongoDB Atlas, a cloud-based service to deploy and configure database. The database schema is defined by using mongoose in Express.js backend, in JSON format.

5.1 Data Dictionary

This section contains the JSON Schema of the application database. It also describes the data type and description of each attribute.

5.1.1 Collection Schemas:

5.1.1.1 Users Collection:

```
{  username: { type: String, required:
true },    fname: { type: String,
required: true },    lname: { type:
String, required: true },    email: { type:
String, required: true },    password: {
type: String, required: true },
phoneNo: { type: String, required:
false },    address: { type: String,
required: false },    image: { type:
String, required: false },
}
```

Field	Data Type	Description
_id	ObjectId	Unique identifier for the user.
Username	String	User's username. (Required)
Fname	String	User's first name. (Required)
Lname	String	User's last name. (Required)
Email	String	User's email address. (Required)
Password	String	User's password. (Required)
phoneNo	String	User's phone number. (Optional)
Address	String	User's address. (Optional)
Image	String	URL or file path to the user's image. (Optional)

Table 1 Users Collection

5.1.1.2 Logins Collection:

```
{ username: { type: String, required:
true }, password: { type: String,
required: true },
}
```

Field	Data Type	Description
_id	ObjectId	Unique identifier for the login.
Username	String	User's username for authentication. (Required)
Password	String	User's password for authentication. (Required)

Table 2 Logins collection

5.1.1.3 ProjectRooms Collection:

```
{ username: { type: String, required: true },
interiorPlannerv0: { type: String, required:
true }, roomName: { type: String, required:
true },
}
```

Field	Data Type	Description
_id	ObjectId	Unique identifier for the project room.
Username	String	User's username associated with the project room.
interiorPlannerv0	String	All the project data which includes the floor plan measurements and placement of items, holes, lines etc.
roomName	String	Name of the room in the project.

Table 3 ProjectRooms collection

5.1.1.4 SavedDesigns Collection:

```
{ username: { type: String, required:
true }, projectData: { type: String,
required: true },
}
```

Field	Data Type	Description
_id	ObjectId	Unique identifier for the saved design entry.
username	String	User's username associated with the saved design.
projectData	String	Data representing the saved design for a project which includes the floor plan measurements and placement of items, holes, lines etc.

Table 4 SavedDesigns collection

5.1.1.5 Image Collection:

```
{ image: { type: String, required: true
}, username: { type: String, required:
true }, date: { type: String, required:
true }, likes: { type: Number,
default: 0,
},
}
```

Field	Data Type	Description
_id	ObjectId	Unique identifier for the image.
image	String	URL or file path representing the image.
username	String	User's username associated with the image.
date	String	Date associated with the image.
likes	Number	Number of likes for the image (default is 0).

Table 5 Image collection

5.1.1.6 UserComments Collection:

```
{  articleName: { type: String,
required: true },  username: { type:
String, required: true },  comment: {
type: String, required: true },
}
```

Field	Data Type	Description
_id	ObjectId	Unique identifier for the user comment entry.
articleName	String	Name of article.
username	String	User's username associated with the comment.
comment	String	The text content of the user's comment.

Table 6 UserComments collection

5.1.1.7 UserLikes Collection:

```
{  articleName: { type: String, required:
true },  username: { type: String,
required: true },
}
```

Field	Data Type	Description
_id	ObjectId	Unique identifier for the user likes entry.
articleName	String	Name or identifier of the associated article.
username	String	User's username associated with the like.

Table 7 UserLikes collection

5.1.1.8 Areas Collection

```
{
  "title": "String",
  "tag": ["String"],
  "description": "String",
  "image": "String"
}
```

Field	Data Type	Description
title	String	Title of the area.
Tag	Array	An array of strings representing tags associated with the area.
description	String	Description or content associated with the area..
image	String	URL or file path representing an image associated with the area.

*Table 8 Areas collection***5.1.1.9 Holes Collection:**

```
{
  "name": "String",
  "prototype": "String",
  "info": {
    "title": "String",
    "tag": ["String"],
    "description": "String",
    "image": "String"
  },
  "properties": {
    "width": {
      "label": "String",
      "type": "length-measure",
      "defaultValue": {
        "length": 80
      }
    }
  },
}
```

```
"height": {  
  "label": "String",  
  "type": "length-measure",  
  "defaultValue": {  
    "length": 215  
  }  
},  
"altitude": {  
  "label": "String",  
  "type": "length-measure",  
  "defaultValue": {  
    "length": 0  
  }  
},  
"thickness": {  
  "label": "String",  
  "type": "length-measure",  
  "defaultValue": {  
    "length": 30  
  }  
},  
"flip_orizzontal": {  
  "label": "String",  
  "type": "checkbox",  
  "defaultValue": false,  
  "values": {  
    "none": false,  
    "yes": true  
  }  
}
```

```

},
}

```

Field	Data Type	Description
name	String	Name of the hole.
prototype	String	String representing the prototype.
info	Object	Object containing information about the hole, including title, tag, description, and image.
properties	Object	Object containing properties related to the hole, including width, height, altitude, thickness, and flip_orizzontal.
properties.width	Object	Object containing details about the width property, including label, type, and defaultValue.
properties.height	Object	Object containing details about the height property, including label, type, and defaultValue.
properties.altitude	Object	Object containing details about the altitude property, including label, type, and defaultValue.
properties.thickness	Object	Object containing details about the thickness property, including label, type, and defaultValue.
properties.flip_orizzontal	Object	Object containing details about the flip_orizzontal property, including label, type, defaultValue, and values.

Table 9 Holes collection

5.1.1.10 Lines Collection:

```

{
  "title": "String",
  "tag": ["String"],
  "description": "String",
  "image": "String",
  "properties": {
    "catalog": "Boolean",
    "layerElementsVisible": "Boolean"
  },
  "visibility": "Object",
  "textures": {

```

```

"type": "Object",
"properties": {
  "bricks": {
    "type": "Object",
    "properties": {
      "name": {"type": "String"},
      "uri": {"type": "String"},
      "lengthRepeatScale": {"type": "Number"},
      "heightRepeatScale": {"type": "Number"},
    }
  }
}

```

Field	Data Type	Description
Title	String	Title of the line.
Tag	Array	An array of strings representing tags associated with the line.
Description	String	Description or content associated with the line.
Image	String	URL or file path representing an image associated with the line.
Properties	Object	Object containing properties related to the line, including catalog and layerElementsVisible.
properties.catalog	Boolean	Boolean indicating whether the line is a catalog element.
properties.layerElementsVisible	Boolean	Boolean indicating the visibility of layer elements.
Visibility	Object	Object representing visibility properties.
Textures	Object	Object representing texture properties.
textures.type	Object	Object representing the type of textures.
textures.properties	Object	Object containing properties related to textures.
textures.bricks	Object	Object containing properties related to brick textures.
textures.bricks.name	String	Name of the brick texture.
textures.bricks.uri	String	URI or file path for the brick texture.

textures.bricks.lengthRepeatScale	Number	Number indicating length repeat scale for the brick texture.
textures.bricks.heightRepeatScale	Number	Number indicating height repeat scale for the brick texture.

Table 10 Lines collection

5.1.1.11 Permissions:

```
{
  "username": { type: String, required: true },
  "role": { type: String, required: true },
  "permissions": { type: String, required: true },
  "projectData": { type: String, required: true },
}
```

Field	Data Type	Description
username	String	User's username.
role	String	User's role in the system.
permissions	String	String representing the permissions assigned to the user.
projectData	String	Data associated with the user's project.

Table 11 Permissions

The comprehensive use of MongoDB and the defined schemas ensure the systematic organization, processing, and retrieval of data within the interior designing web application. The models align with the principles of NoSQL databases, providing flexibility for the dynamic nature of interior design projects.

6. Implementation

This section contains the implementation details of the system developed so far. It includes the algorithms, external APIs used by the system, user interface design and deployment details.

6.1 Algorithm

Following are the major algorithms used in implementing the functionality of the system developed so far.

6.1.1 SignUp

Algorithm SignUp
Input: Form data including username, fname, lname, email, and password
Output: Success message or error message 1. Initialize formData, message, and validationErrors variables 2. Display the signup form with input fields for username, fname, lname, email, and password 3. When user enters data into the form, update the corresponding values in the formData state 4. When the signup button is clicked: 4.1. Send a POST request to the server endpoint '/signup' with the formData 4.2. Handle the response from the server: 4.2.1. If the response is successful (status 200): 4.2.1.1. Display a success message 4.2.1.2. Clear the form data 4.2.1.3. Redirect the user to the login page 4.2.2. If the response indicates a validation error (status 400): 4.2.2.1. Display an alert with the validation error message 4.2.3. If there is any other error: 4.2.3.1. Display an alert with an error message 5. Implement frontend validation to ensure that all required fields are filled 6. Implement frontend validation for email format using the emailRegex 7. Implement frontend validation for password strength using strongPasswordRegex 8. Display appropriate error messages if validation fails 9. Handle server-side validation for existing users with the same email or name 10. Display relevant error messages for duplicate email or name 11. Use the navigate function to redirect the user to the login page after successful signup

6.1.2 Login:

Algorithm Login
Input: User credentials including username, password, and keepLoggedIn flag
Output: Success message, error message, or password reset form
1. Initialize variables for username, password, isChecked, message, showForgotPassword, and resetEmail 2. Display the login form with input fields for username and password 3. When the user enters data into the form, update the corresponding values in the state variables 4. When the login button is clicked: 4.1. Send a POST request to the server endpoint '/login' with the username, password, and keepLoggedIn flag 4.2. Handle the response from the server: 4.2.1. If the response is successful (status 200):

- 4.2.1.1. Display a success message
- 4.2.1.2. Store the token in local storage if the "Keep me logged in" checkbox is checked
- 4.2.1.3. Redirect the user to the homepage (e.g., '/Req')
- 4.2.2. If the response indicates an authentication error (status 401):
 - 4.2.2.1. Display an error message indicating invalid username or password
- 4.2.3. If there is any other error:

- 4.2.3.1. Display an error message
- 5. Implement frontend validation for the login form
- 6. Implement the "Keep me logged in" checkbox functionality using the isChecked state variable
- 7. Implement the "Forgot password" functionality:
 - 7.1. Display a password reset form when the "Forgot password?" button is clicked
 - 7.2. When the reset password button is clicked:
 - 7.2.1. Send a POST request to the server endpoint '/reset-password' with the resetEmail
 - 7.2.2. Handle the response from the server:
 - 7.2.2.1. If the email is found, navigate to the password reset page with the email as a parameter
 - 7.2.2.2. If the email is not found, display an error message
 - 7.3. Provide a back to login button to switch back to the login form
- 8. Display appropriate error messages for invalid username or password and any other errors
- 9. Use local storage to check if the user is already logged in and redirect them to the desired route

6.1.3 Social Media Integration:

Algorithm Social Media Integration

Input: User-selected image file, username, like button click

Output: error message, posted designs.

1. Initialize state variables for images, selectedImage, username, selectedFile, selectedImageName
2. Display a community page with a list of posted images, like buttons, and an upload form
3. Fetch the initial set of images when the component mounts
4. Set up an interval to fetch images periodically (every 10 seconds)
5. Clean up the interval when the component unmounts
6. Implement a function to handle image uploads:
 - 6.1. Update the state with the selected image file and username
 - 6.2. Create a FormData object and append the selected file and username to it
 - 6.3. Send a POST request to the server endpoint '/api/uploadImage' with the FormData
 - 6.4. Update the state with the new image data returned from the server
7. Implement a function to handle image clicks:
 - 7.1. Set the selectedImage state with the clicked image data
8. Implement a function to handle like button clicks:
 - 8.1. Increment the likes count locally
 - 8.2. Send a POST request to the server endpoint '/api/likeImage/:imageId' to update the likes count
 - 8.3. Update the state to reflect the new likes count
9. Display a modal with the selected image when an image is clicked
10. Ensure error handling for image uploads and like button clicks
11. Display relevant success or error messages to the user

6.1.4 Update Profile:

Algorithm Profile Setting
Input: User details
Output: Updated user details, error message
<ol style="list-style-type: none"> 1. Initialize state variables for user details, user images, selected image, selected file, selected image name, is editing status for each field, notification message, and notification visibility 2. Display a user profile page with user details, profile picture, and editable fields 3. Fetch the user details when the component mounts using the username from local storage 4. Fetch user images and display the profile picture 5. Set up intervals to fetch user images periodically 6. Implement functions to handle edit and save button clicks for each field <ol style="list-style-type: none"> 6.1 Update the user details locally based on user input 6.2 Send a PUT request to the server endpoint '/profile/user/:username' to update the user details 6.3 Show success or error notifications based on the response 7. Implement a function to handle profile picture uploads <ol style="list-style-type: none"> 7.1 Update the selected file state with the chosen file 7.2 Send a POST request to the server endpoint '/profile/user/:username/profile-pic' to upload the profile picture 7.3 Update the user details and images array with the new profile picture path 7.4 Show success or error notifications based on the response 8. Implement functions to handle image clicks and display a modal with the selected image 9. Show a notification for successful updates and hide it after a certain duration

6.1.5 Reset Password

Algorithm Reset Password
Input: Email from URL params, password, confirm password
Output: Updated password, username from state
<ol style="list-style-type: none"> 1. Initialize state variables for email (from URL params), password, confirm password, message, and username 2. Fetch email from URL params using the useParams hook 3. Display a reset password form with input fields for the new password and confirm password 4. Implement a function to handle form submission (handleResetPassword) <ol style="list-style-type: none"> 4.1 Prevent the default form submission behavior 4.2 Check if the password and confirm password match 4.3 If not, set a message indicating that passwords do not match 4.4 Send a POST request to the server endpoint '/reset/:email' with the new password 4.5 Display the server response message and username 4.6 If an error occurs, display an error message

6.1.6 Expert Articles and advice

Algorithm Expert advice and articles
Input: Comments, likes
Output: Displayed article content with like and comment functionality
<ol style="list-style-type: none"> 1. Initialize state variables for comments, newComment, liked, and likesCount 2. Fetch article-related data (likes and comments) on component mount <ol style="list-style-type: none"> 2.1 Use the useEffect hook to call fetchLikes and fetchComments functions 2.2 fetchLikes: Retrieve likes data for the article using a GET request to the '/api/userLikes' endpoint <ol style="list-style-type: none"> 2.2.1 Store the likes count and check if the current user has already liked the article 2.3 fetchComments: Retrieve comments data for the article using a GET request to the '/api/comments2' endpoint <ol style="list-style-type: none"> 2.3.1 Update the comments state with the fetched data 3. Implement like functionality <ol style="list-style-type: none"> 3.1 Display a like button and the likes count 3.2 Use the handleLike function to handle like button clicks 3.3 Check if the user has already liked the article 3.4 If not liked, send a POST request to the '/api/userLikes' endpoint to add a like 3.5 Update the likes count and set the liked state to true 4. Implement comment functionality <ol style="list-style-type: none"> 4.1 Display a comment section with existing comments 4.2 Use the submitComment function to handle new comments 4.3 Send a POST request to the '/api/comments2' endpoint to add a new comment 4.4 Update the comments state with the new comment 4.5 Clear the newComment input field after posting the comment

6.1.7 Design Ideas

Algorithm Design Inspiration and Ideas
Input: Playlists
Output: Displayed page with YouTube videos
<ol style="list-style-type: none"> 1. Initialize state variables for playlists, selectedPlaylistId, and playlistVideos 2. Fetch YouTube playlists on component mount <ol style="list-style-type: none"> 2.1 Use the useEffect hook to call the fetchPlaylists function 2.2 fetchPlaylists: Send a GET request to the YouTube Data API to fetch playlists related to design inspiration <ol style="list-style-type: none"> 2.2.1 Store the playlists in the state variable 'playlists' 3. Fetch playlist videos when a playlist is selected <ol style="list-style-type: none"> 3.1 Use the useEffect hook with a dependency on selectedPlaylistId to call the fetchPlaylistVideos function 3.2 fetchPlaylistVideos: Send a GET request to the YouTube Data API to fetch videos from the selected playlist <ol style="list-style-type: none"> 3.2.1 Update the state variable 'playlistVideos' with the fetched video data 4. Implement playlist selection dropdown <ol style="list-style-type: none"> 4.1 Display a dropdown to select a playlist 4.2 Use the handlePlaylistChange function to update the selectedPlaylistId state when a playlist is selected 5. Display selected playlist videos

- 5.1 Display videos from the selected playlist in a grid
- 5.2 Use an iframe for each video to embed it on the page
- 6. Style the page with appropriate CSS for layout and design
- 7. Add background image to the page
 - 7.1 Use the background-image style property with a specified URL for the background image
 - 7.2 Set background-size to cover, background-attachment to fixed, and margin and padding to 0
- 8. Include a Navbar component for navigation
- 9. Add a Chatbot UI component to the page

6.1.8 Industry news and updates

Algorithm Industry News and Updates
Input: Request for latest news and updates
Output: Displayed page with latest industry news and updates
<ol style="list-style-type: none"> 1. Initialize state variables for news and loading 2. Fetch industry news on component mount <ol style="list-style-type: none"> 2.1 Use the useEffect hook to call the fetchNews function 2.2 fetchNews: Send a GET request to the News API to fetch articles related to interior design, home, and furniture <ol style="list-style-type: none"> 2.2.1 Store the fetched news articles in the state variable 'news' 2.2.2 Set the loading state to false after the news is fetched 3. Style the page with appropriate CSS for layout and design 4. Add background image to the page <ol style="list-style-type: none"> 4.1 Use the background-image style property with a specified URL for the background image 4.2 Set background-size to cover, background-attachment to fixed, and margin and padding to 0 5. Display the latest industry news <ol style="list-style-type: none"> 5.1 Display a loading spinner while the news is being fetched 5.2 Use a conditional rendering to check if loading is true or false 5.3 If loading is true, display a loading spinner 5.4 If loading is false, map through the 'news' array and display each news article in a styled box <ol style="list-style-type: none"> 5.4.1 Use a unique background color for each news box, calling the getRandomColor function 5.4.2 Display the news article title, description, source, date, and a "Read More" link 5.4.3 Link the "Read More" link to the article URL, opening it in a new tab 6. Include a Navbar component for navigation 7. Add a Chatbot UI component to the page

6.1.9 AI Powered Recommendations:

Algorithm Chatbot
Input: User question
Output: Displayed chat interface with user and bot responses
<ol style="list-style-type: none"> 1. Initialize state variables for messages, inputText, and chatVisible 2. Fetch initial chat messages from the server on component mount <ol style="list-style-type: none"> 2.1 Use the useEffect hook to call initializeChatbot function 2.2 initializeChatbot: Send a GET request to the server endpoint for initializing the chatbot <ol style="list-style-type: none"> 2.2.1 Set the fetched messages in the state variable 'messages'

3. Define event handlers for input change and form submission
 - 3.1 handleInputChange: Update the 'inputText' state on input change
 - 3.2 handleSubmit: Send a POST request to the server endpoint for sending a user message
 - 3.2.1 Include the input text in the request body
 - 3.2.2 Update the 'messages' state with the user message and the corresponding bot response
 - 3.2.3 Clear the 'inputText' after submitting the message
4. Define a function to toggle the visibility of the chat window
 - 4.1 toggleChatVisibility: Update the 'chatVisible' state to show/hide the chat window
5. Render the chat interface
 - 5.1 Display the chat container
 - 5.2 Use conditional rendering to show the chatbot UI only if 'chatVisible' is true
 - 5.3 Map through the 'messages' array to display each message in the chat window
 - 5.4 Display the user and bot messages with appropriate styling
 - 5.5 Render an input form for typing messages
 - 5.5.1 Include an input field to type messages (controlled by 'inputText' state)
 - 5.5.2 Include a submit button to send messages
 - 5.6 Display a chat icon with an onClick event to toggle the chat visibility

6.1.10 Drawing Walls

Algorithm Drawing walls on canvas
Input: state, layerID, x, y
Output: Updated state
<ol style="list-style-type: none"> 1. Initialize snapElements using SnapSceneUtils.sceneSnapElements <ol style="list-style-type: none"> 1.1 Set snapElements to the result of SnapSceneUtils.sceneSnapElements (using the current scene from state, an empty list, and the snapMask from state) 1.2 Initialize snap to null 2. Check if snapMask is not empty <ol style="list-style-type: none"> 2.1 If snapMask is not empty, find the nearest snap point using SnapUtils.nearestSnap (using snapElements, x, y, and state.snapMask) 2.2 If a snap point is found, update x and y to the snap point coordinates 2.3 Modify snapElements <ol style="list-style-type: none"> 2.3.1 Create horizontal and vertical lines using GeometryUtils.horizontalLine and GeometryUtils.verticalLine 2.3.2 Add line snaps to snapElements for the created lines 3. Create drawingSupport using the current state's drawingSupport with updated layerID <ol style="list-style-type: none"> 3.1 Set drawingSupport to the result of updating the drawingSupport layerID 4. Unselect all elements in the specified layer using Layer.unselectAll <ol style="list-style-type: none"> 4.1 Set state to the result of Layer.unselectAll with the specified layerID 5. Create a new line using Line.create with the specified layerID, drawing type, and initial coordinates <ol style="list-style-type: none"> 5.1 Set { updatedState: stateL, line } to the result of Line.create 5.2 Set state to stateL 5.3 Select the newly created line using Line.select <ol style="list-style-type: none"> 5.3.1 Set state to the result of Line.select with the specified layerID and line ID

6. Update the state with the drawing mode, snapElements, activeSnapElement, and drawingSupport
 - 6.1 Set state mode to MODE_DRAWING_LINE
 - 6.2 Set state snapElements to the computed snapElements
 - 6.3 Set state activeSnapElement to the snap point or null
 - 6.4 Set state drawingSupport to the computed drawingSupport
7. Return { updatedState: state }

6.1.11 3D Rendering of design project:

Algorithm 3D View of Design Project
Input: props (width, height, state), Three (3D graphics library), OrbitControls
Output: 3D scene
<ol style="list-style-type: none"> 1. Initialize class properties in the constructor <ol style="list-style-type: none"> 1.1 Set initial values for lastMousePosition, width, height, and renderingID 1.2 Create or reuse a WebGLRenderer instance and store it in the class property 'renderer' 2. Implement componentDidMount lifecycle method <ol style="list-style-type: none"> 2.1 Fetch necessary actions from the context (areaActions, holesActions, itemsActions, linesActions, projectActions) 2.2 Extract scene data from props 2.3 Create a new Three.Scene instance and set the clear color for the renderer 2.4 Add plan geometry and grid to the scene 2.5 Set up a PerspectiveCamera, set its position based on bounding box, and add it to the scene 2.6 Add ambient light and spotlights to the scene 2.7 Set up object picking using Raycaster 2.8 Attach mousedown and mouseup event listeners for object interaction 2.9 Create and append the renderer's DOM element to the canvas wrapper 2.10 Create an OrbitControls instance for camera manipulation 2.11 Define a render function that updates controls, lights, and renders the scene 2.12 Call the render function recursively using requestAnimationFrame 2.13 Set class properties for camera, scene3D, and planData 3. Implement componentWillUnmount lifecycle method <ol style="list-style-type: none"> 3.1 Cancel the animation frame to stop rendering when the component is unmounted 3.2 Dispose of the OrbitControls 3.3 Remove event listeners and clean up the scene 3.4 Set class properties to null and dispose of render lists 4. Implement componentWillReceiveProps lifecycle method <ol style="list-style-type: none"> 4.1 Extract width, height, state from nextProps 4.2 Update camera aspect ratio and projection matrix 4.3 Check for changes in scene props using immutablediff 4.4 Call updateScene function to handle changes in the scene 4.5 Update renderer size with new width and height 5. Implement render method <ol style="list-style-type: none"> 5.1 Return a div with a ref attribute pointing to 'canvasWrapper' 6. Define PropTypes for the component (state, width, height) 7. Define contextTypes for required context actions and catalog

6.2 External APIs/SDK

The below table shows the external APIs used by the system.

Name of API and version	Description of API	Purpose of Usage	API Endpoint/Function/Class
News API	The News API provides a programmatic way to access and retrieve news articles and headlines from various sources across the internet.	We are using this API to fetch the industry news and updated related to interior designing.	File: EducationInspirationNews.js API endpoint: 'https://newsapi.org/v2/everything?q=(interior%20design%20OR%20home%20OR%20furniture)&apiKey=\${apiKey}'
YouTube API	The YouTube API allows to access and interact with YouTube's vast video content, channels, and user data.	We are using this API to access YouTube "Hashtag Décor" channel playlist for design inspiration and ideas.	File: DesignInspirationIdeas.js API endpoint: 'https://www.googleapis.com/youtube/v3/playlistItems?key=AIzaSyBnE21LnVyZULOKv_VVhNL3gH6Vu5E33Q&playlistId=\${selectedPlaylistId}&part=snippet&maxResults=10'
Pinterest API	This API is used to access Pinterest user data, boards, and pins, as well as perform actions like creating or updating boards and pins. It enables the integration of Pinterest features into third-party applications.	We are using this API to retrieve and display the trend and inspiration boards	File: TrendBoards.js
Dialogflow API	This API allow developers to integrate and manage natural language understanding and conversation flow in their applications, making it easier to create interactive and intelligent chat experiences.	We are using this API to provide user with AI powered recommendations related to interior designing.	File: ChatbotUI API endpoint: await axios.get('http://localhost:5000/api/init')
Google Maps API	This API is used to embed interactive maps into their websites or applications and leverage location-based services.	We are using this API for displaying maps and locations within the application.	File: Suppliers.js

Table 12 External APIs

6.3 User Interface

6.3.1 Login Form

The following image shows login form. User can enter his/her username and password to log into the InteriorX web application and access its features.

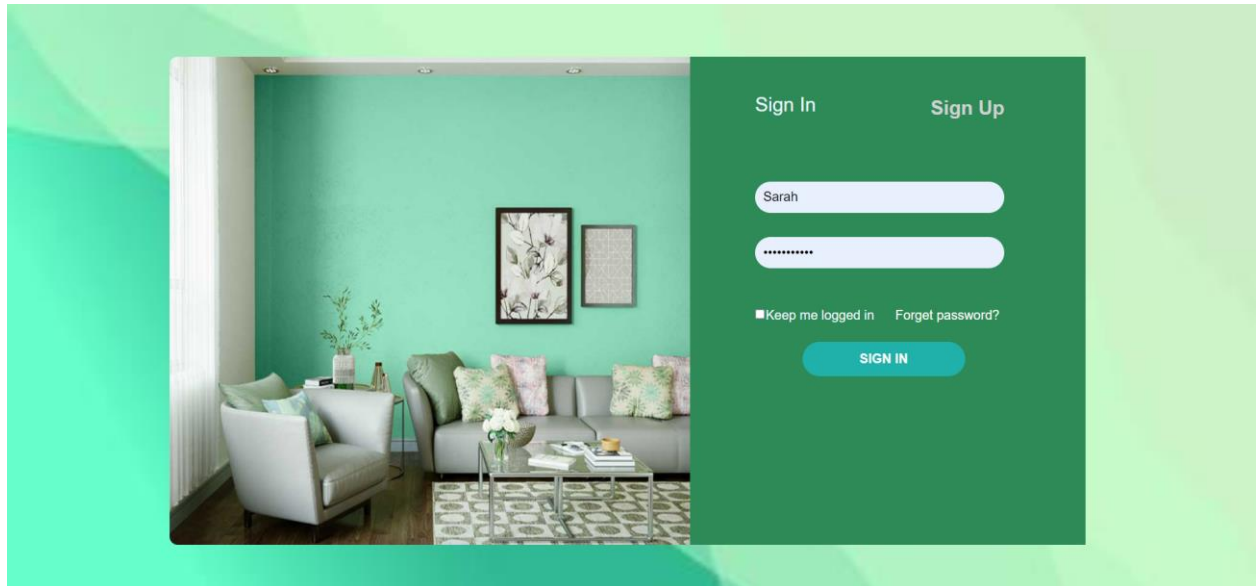


Figure 34 Login Form

6.3.2 Forget Password Page

The following image shows forget password page, in case if the user forgets his password then he can enter his email and the password will be provided to the user along with username.

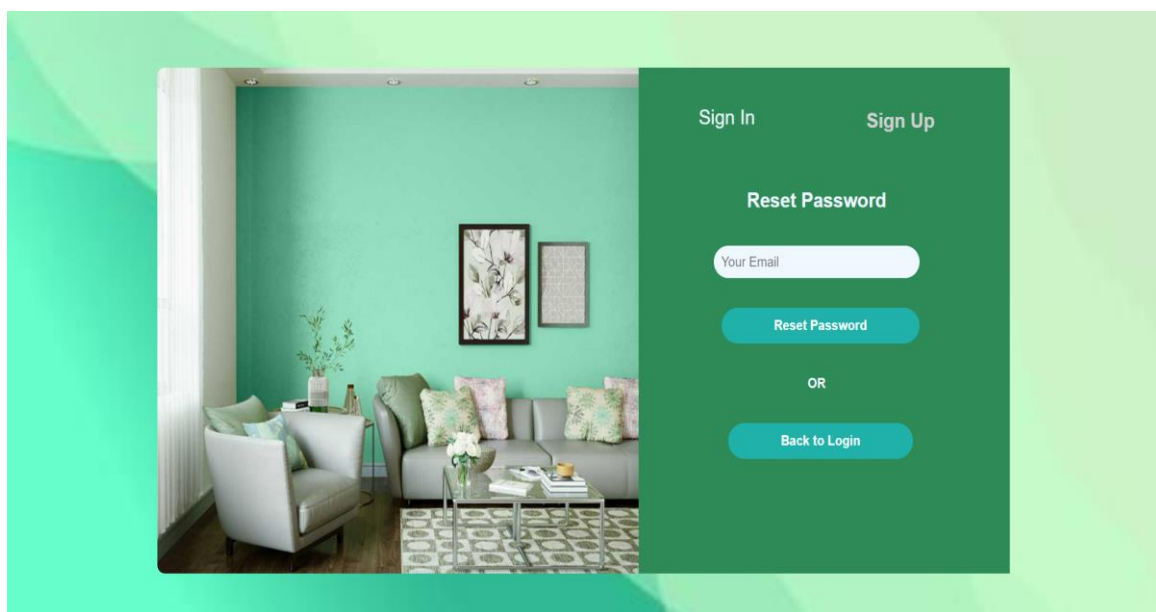


Figure 35 Forget Password Page

6.3.3 Sign up Page

The following image shows registration form. In order to access InteriorX application, user has to make account by providing his/her username, password, first, last names, and email.

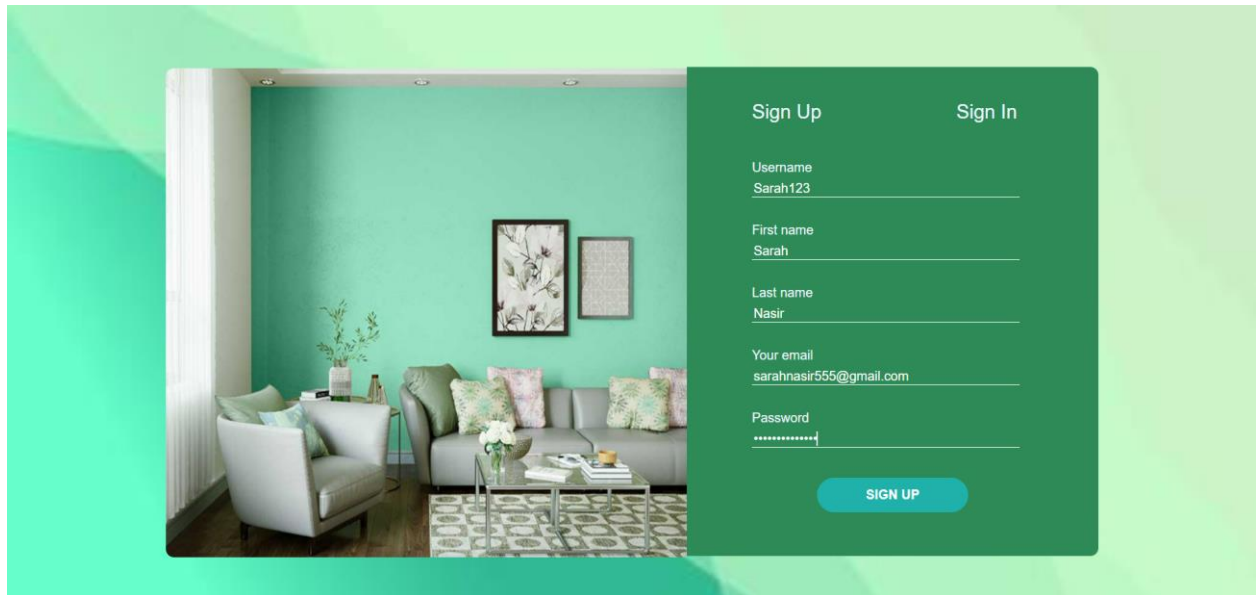


Figure 36 Sign up Page

6.3.4 Edit Profile Page

The following image shows the profile setting page. User can update his/her credentials and set profile picture etc.

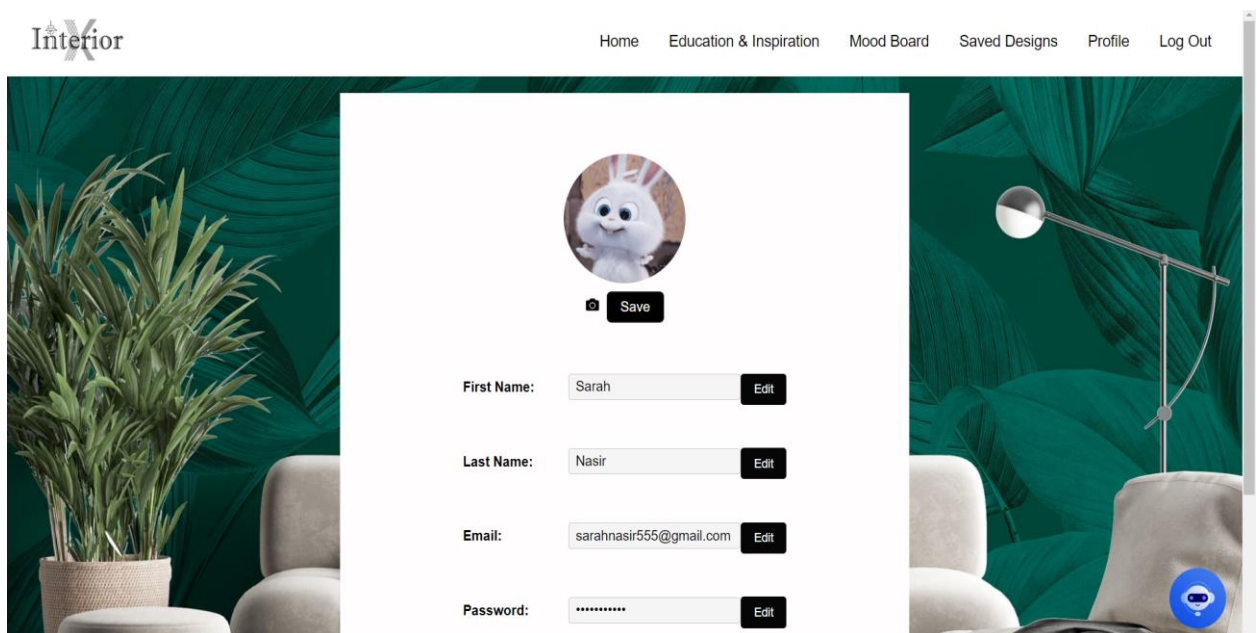


Figure 37 Edit Profile Page

6.3.5 Home Page of InteriorX

The following image shows the Homepage of application. User can click on “Get started” option to start designing his/her room space.

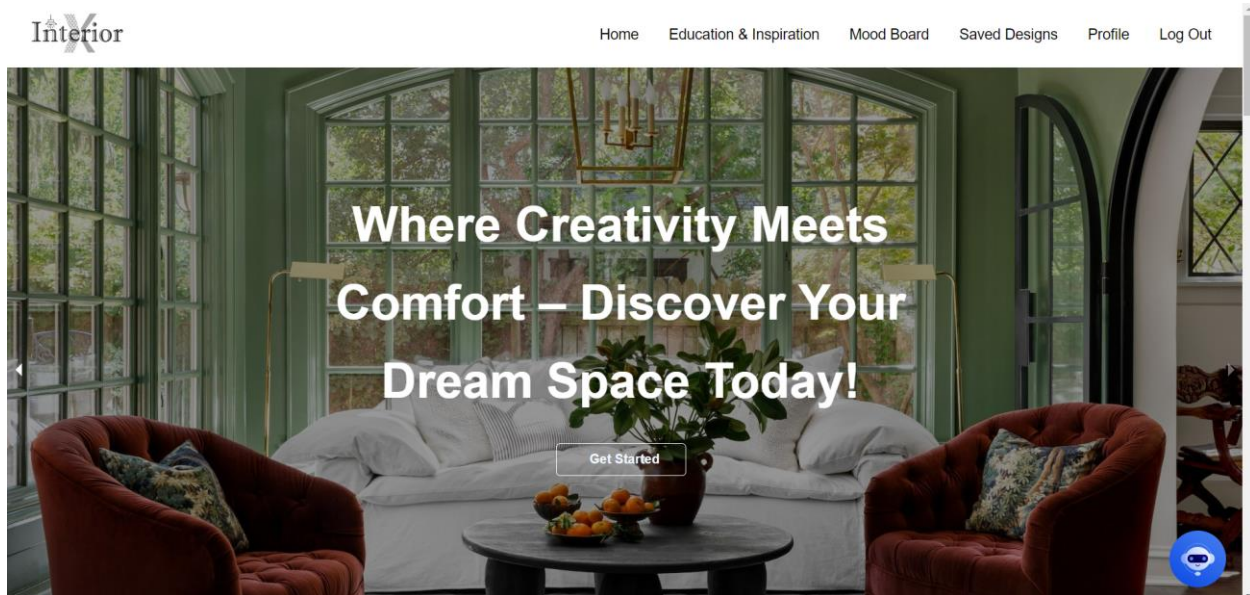


Figure 38 Home Page

6.3.6 Type of project Page

The following image shows the type of project that user wants to create either a House, Single room, Flat or Studio. On these basis, some rooms will get automatically selected that will compose the project.

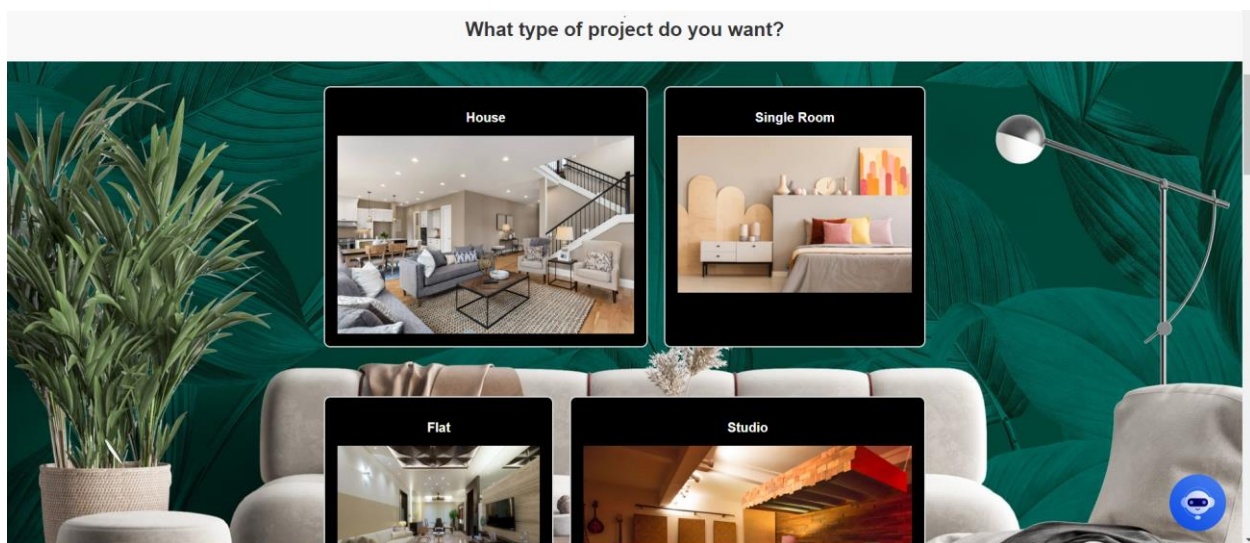


Figure 39 Type of Project Page

6.3.7 Rooms Options Page

The following image shows that the user can select the rooms in his/her project further if he/she wants to.

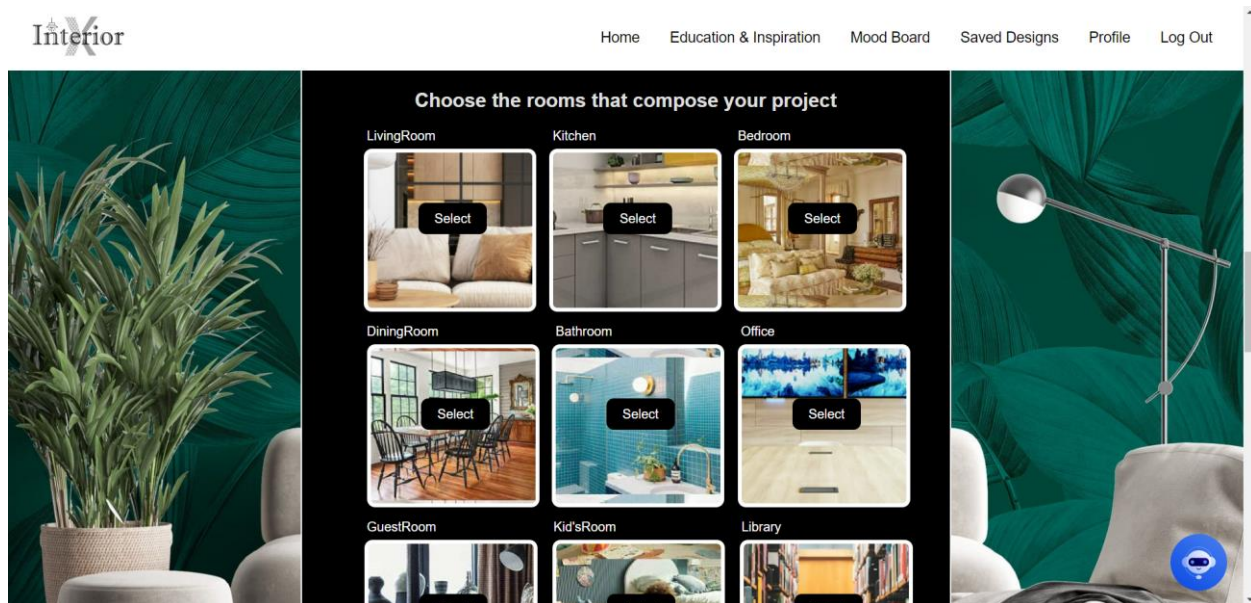


Figure 40 Rooms Options Page

6.3.8 Room Selection

The following images shows the types of rooms that user selects for the project.

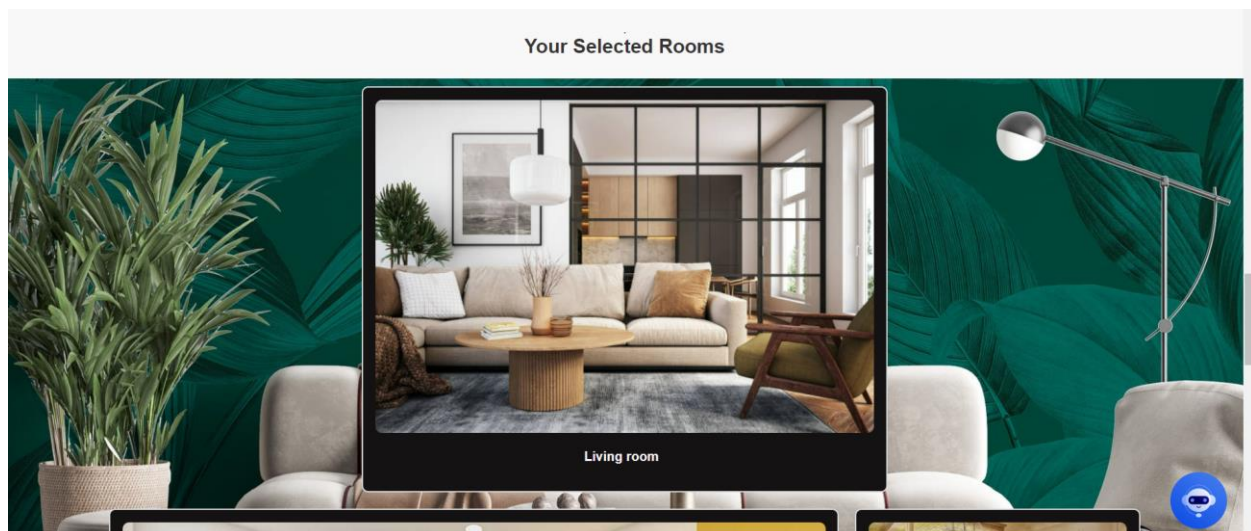


Figure 41 Room Selection Page 1

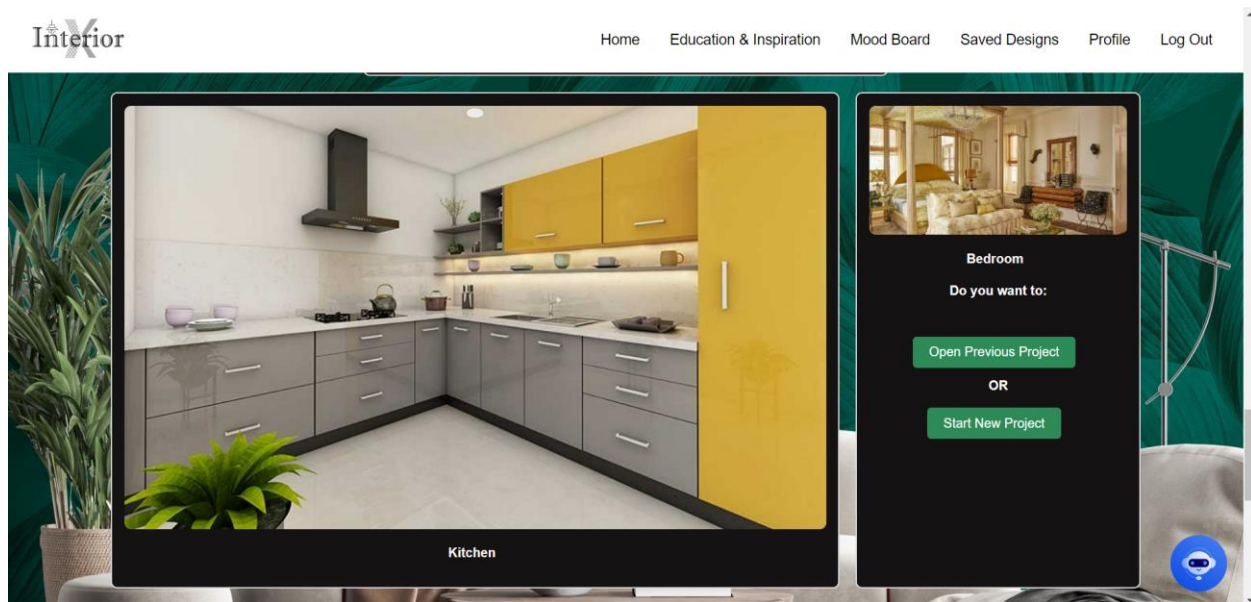


Figure 42 Room Selection Page 2

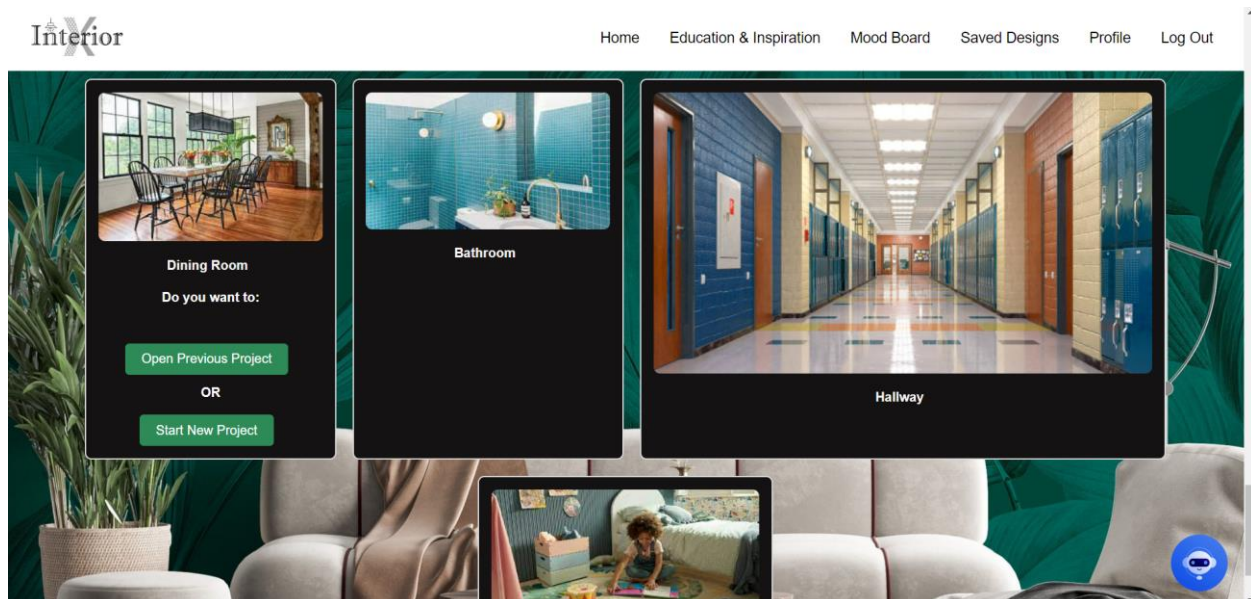


Figure 43 Room Selection Page 3

6.3.9 Drawing Canvas Page

The following image shows the drawing canvas once the user clicks on a particular room to design it.

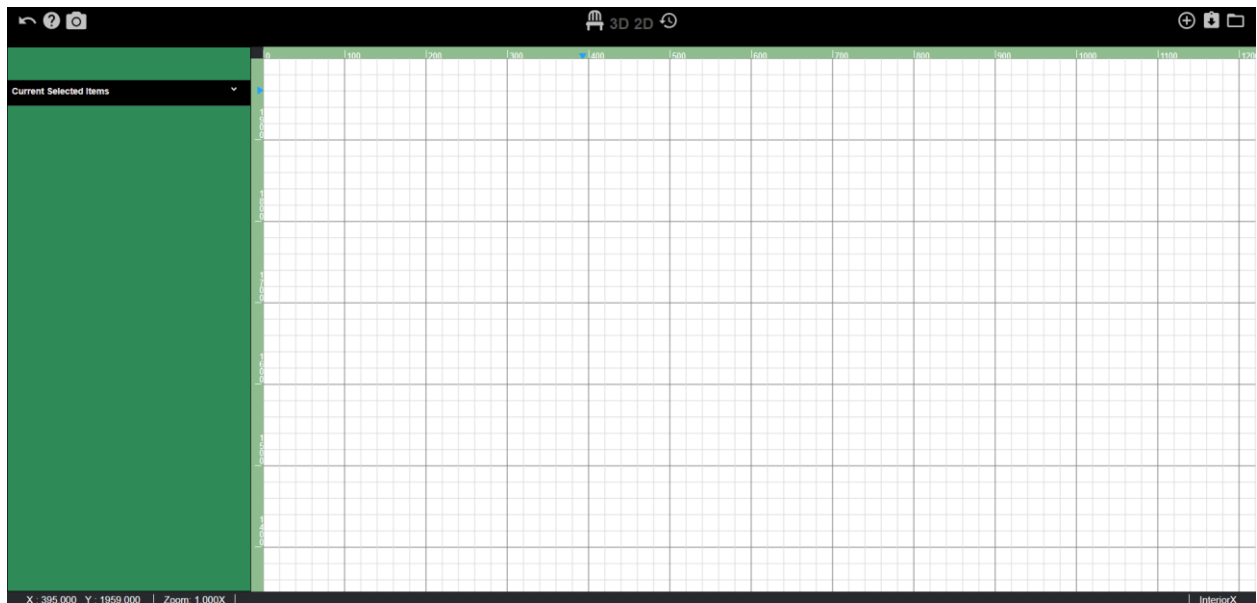


Figure 44 Drawing Canvas Page

6.3.10 Walls drawn on canvas

The following image shows the walls on the canvas that user has drawn. He/she can adjust the size of wall, edit it, remove it etc.

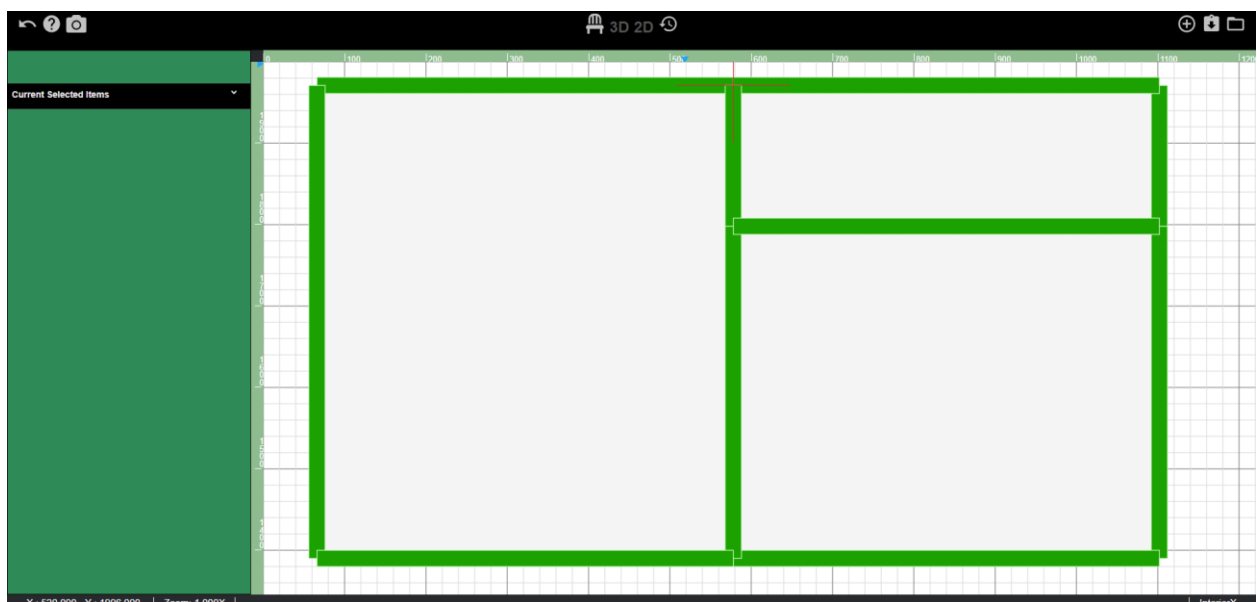


Figure 45 Walls drawn on Canvas Page

6.3.11 Doors and windows placement on walls

The following image shows the placement of doors and windows on the walls.

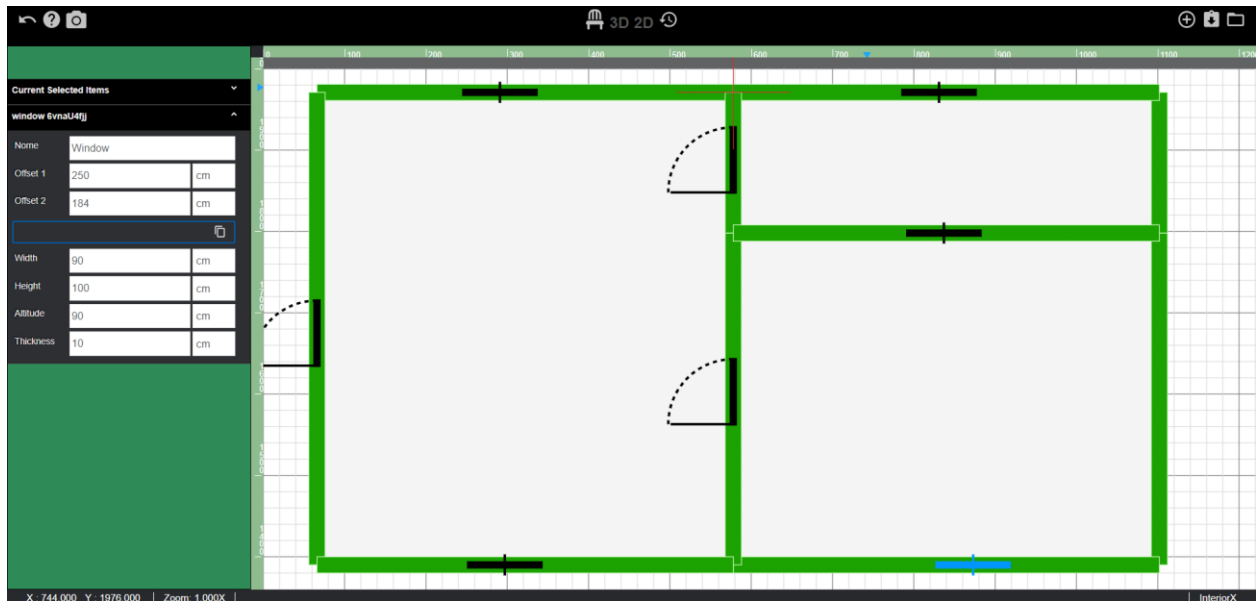


Figure 46 Doors and Window placement

6.3.12 Furniture placement on floor plan

The following image shows the placement of furniture on 2D floor plan.

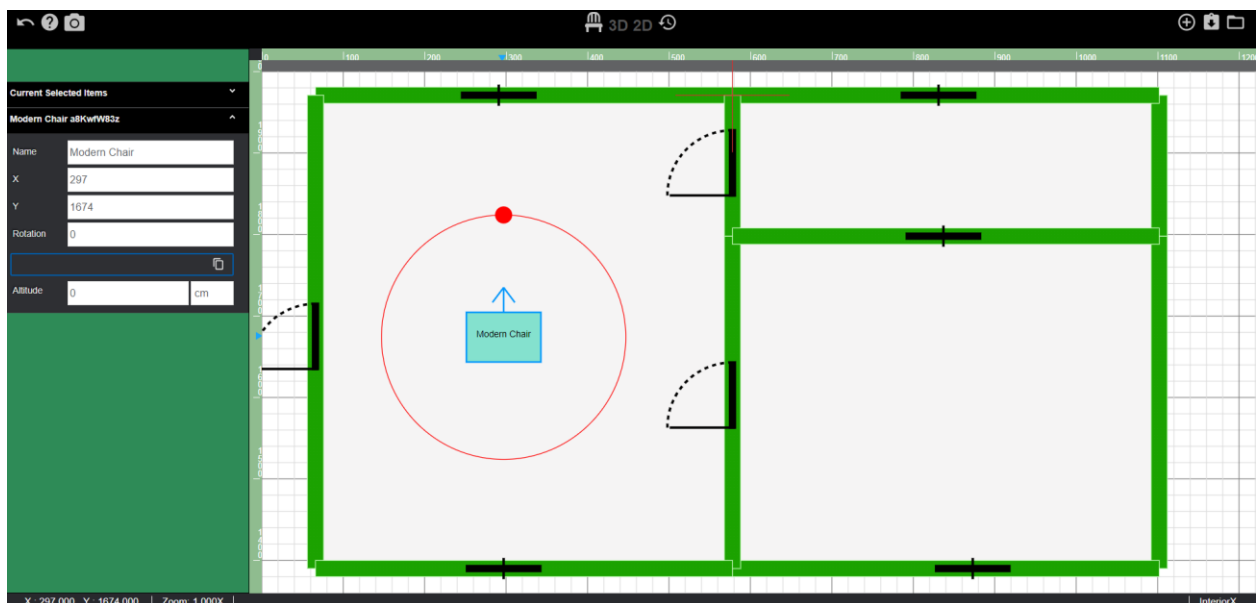


Figure 47 Furniture Placement

6.3.13 3D view of floor plan

The following image shows the floor plan and furniture items in 3D view.

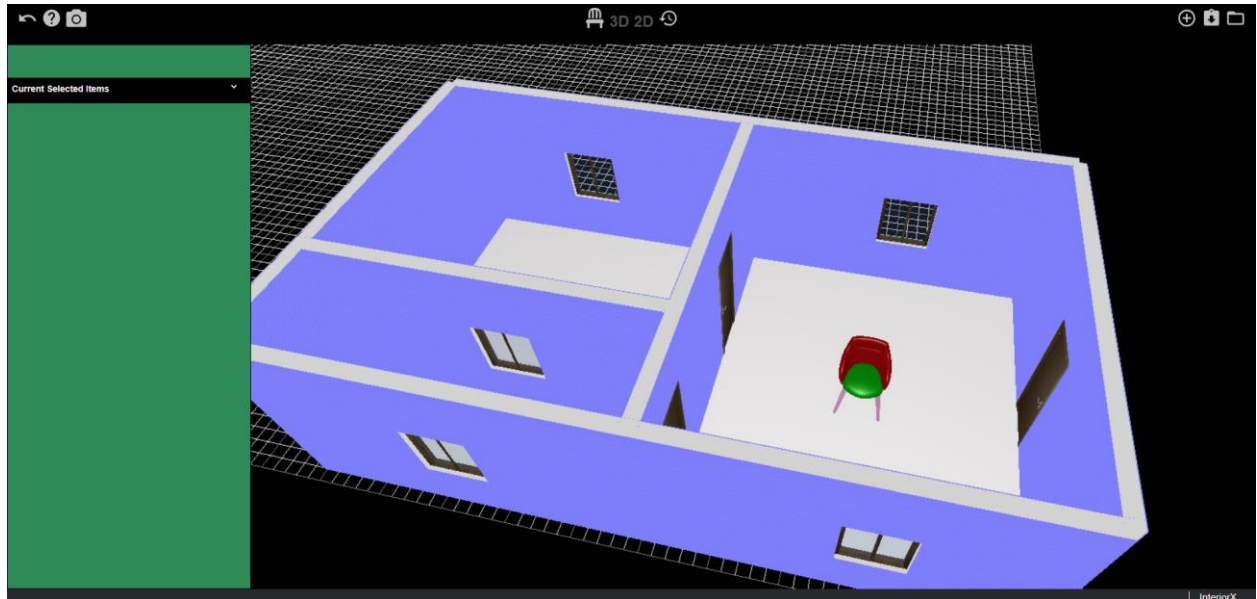


Figure 48 3D view of floor plan

6.3.14 Textures of walls and floors

The following image shows the different textures for walls and floors.

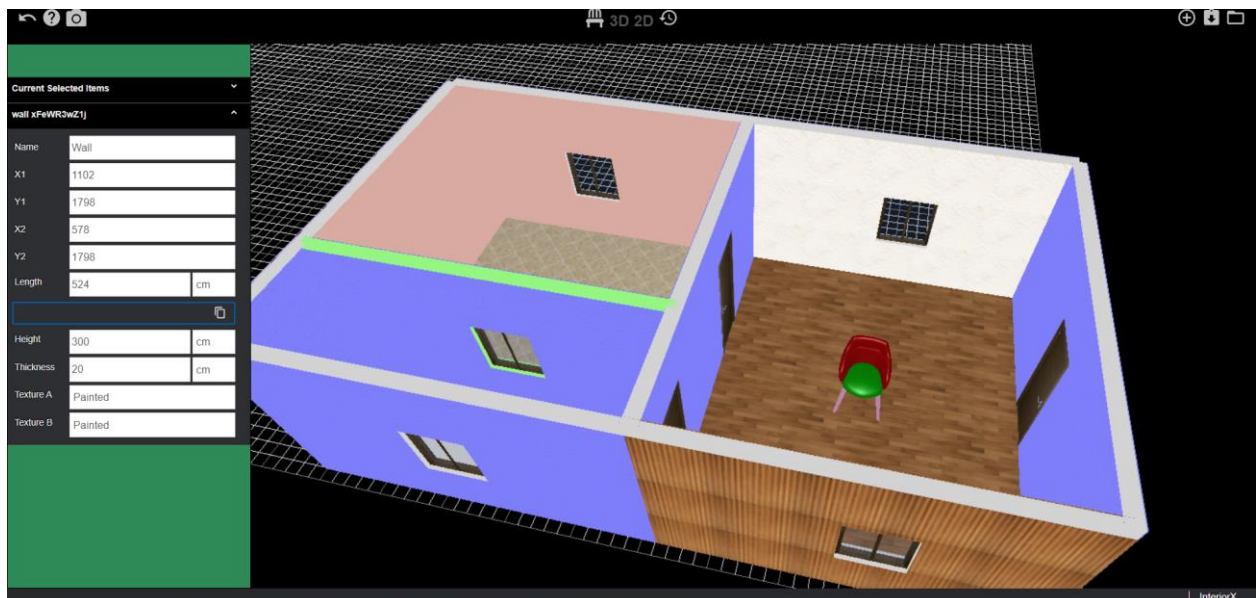


Figure 49 Texture of walls and floors

6.3.15 Education and Inspiration section

The following image shows the education and inspiration section. It has four options available that can help the user in creating professional look of their room spaces. These are articles, design ideas, design news, and community section where user can check other people room space designs.

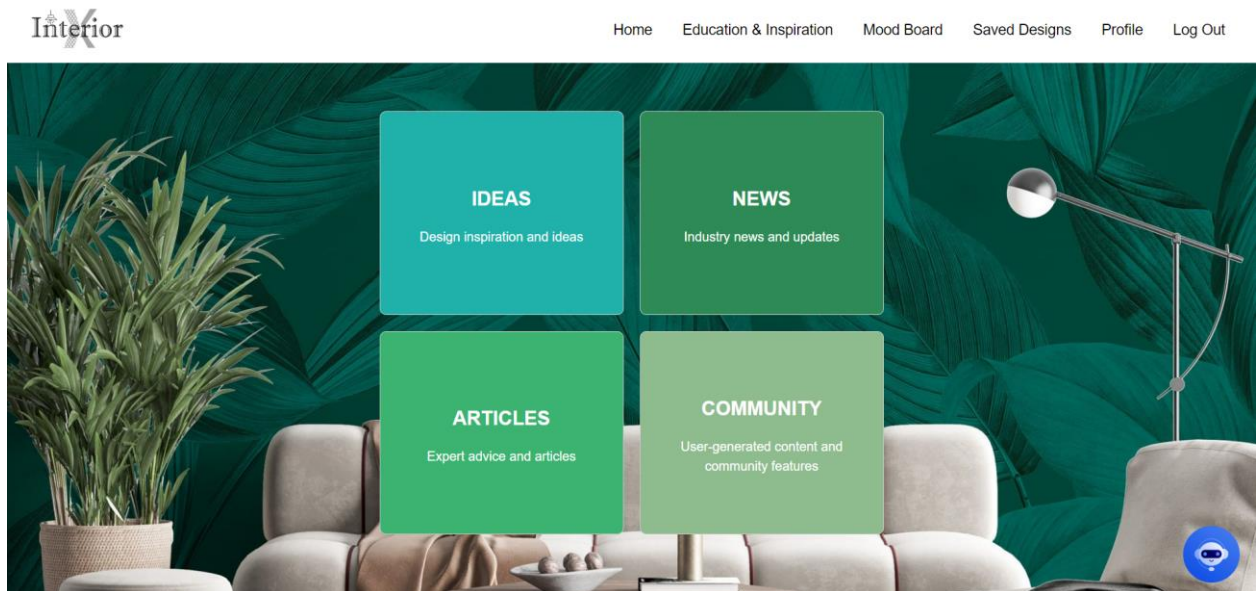


Figure 50 Education and Inspiration Section

6.3.16 Articles section

The following image shows the articles section.

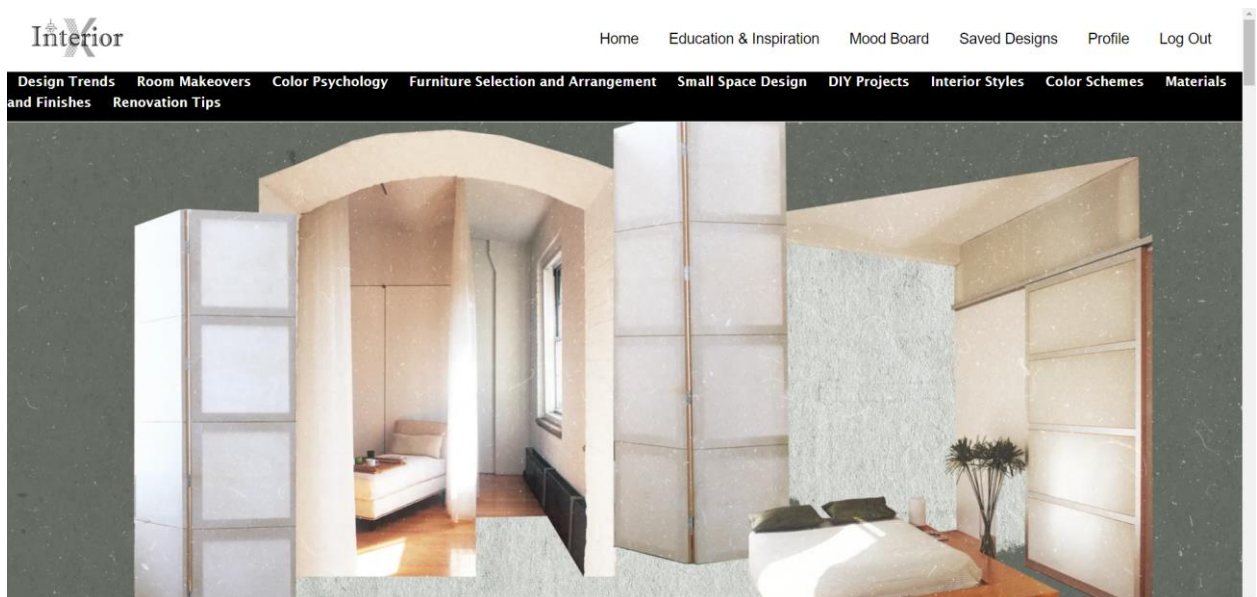


Figure 51 Articles Section

6.3.17 Like and Comment on Articles

The following image shows article section and users can like and comment on these articles.

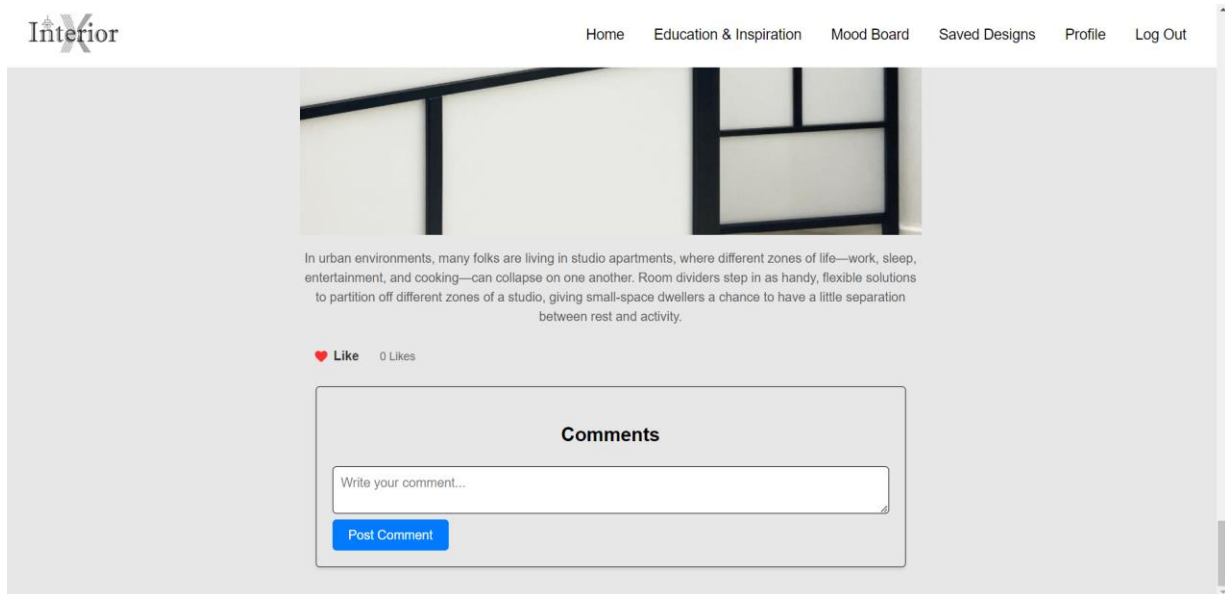


Figure 52 Like and Comment on Articles

6.3.18 Design Inspiration Ideas section

The following image shows design inspiration ideas section. User can select the type of videos that he wants and then the videos will get displayed to the user.

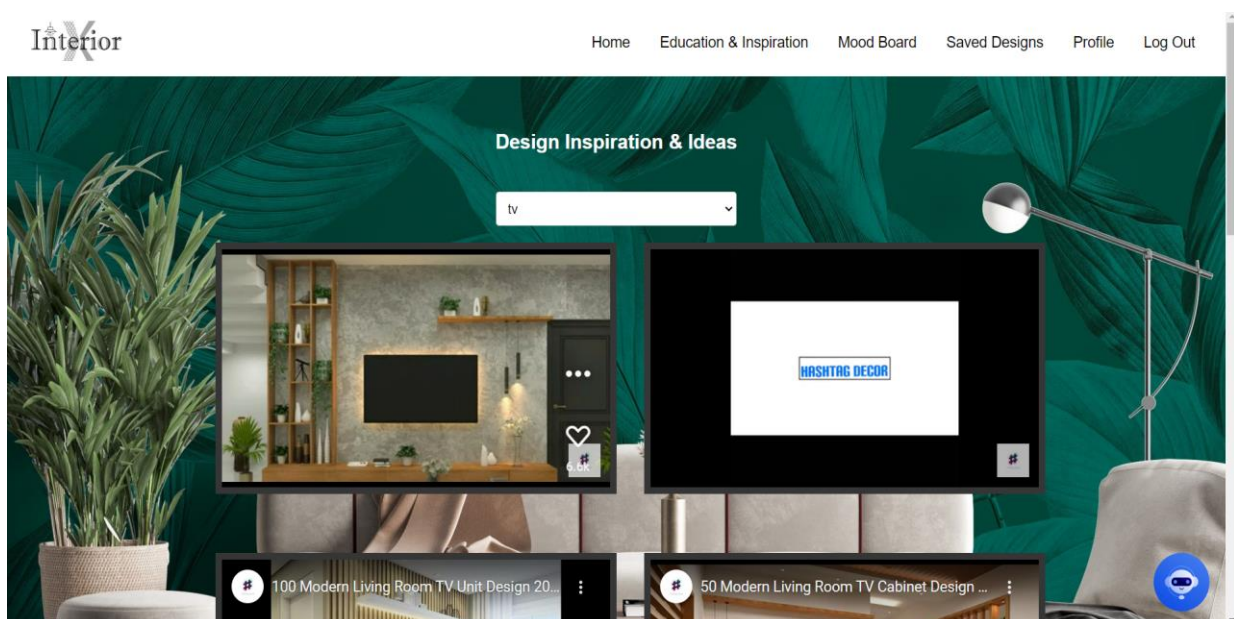


Figure 53 Design Inspiration Ideas Section

6.3.19 News section

The following image shows latest industry news and updates section.

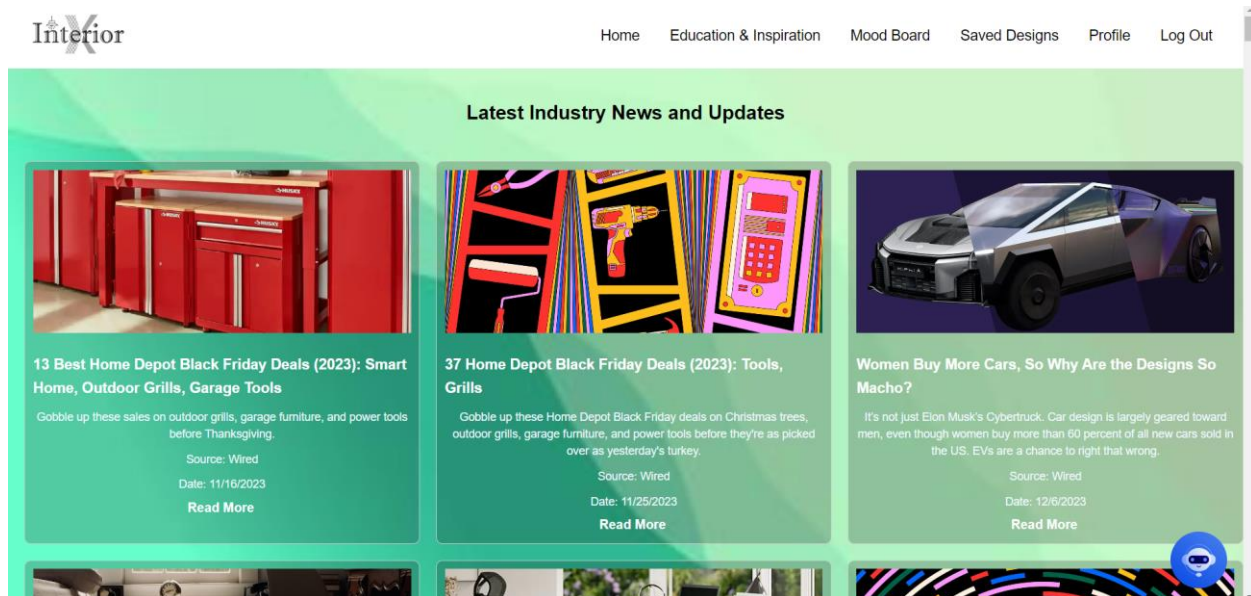


Figure 54 News Section

6.4 Deployment

The application is still in production environment. It is not deployed yet.

7. Testing and Evaluation

This section contains the test cases for the system developed so far. It includes unit tests, functional tests, scenario-based testing and integration testing.

7.1 Unit Testing

The purpose of unit testing is to validate that each unit of the software performs the desired functionality.

Unit Testing 1: Register with valid and invalid credentials.

Testing Objective: To ensure that the register functionality is working correctly with both valid and invalid inputs.

No.	Test case/Test script	Attribute and value	Expected result	Result
1	To verify that all fields take input or not	Leave any field empty	Error message displayed that all fields must be filled	Pass
2	To verify that the sign up button works correctly	Click on sign up button	The sign up button redirects user to the signin page	Pass
3	To verify that the email field accepts email if it is not already registered	Email: wardarana083@gmail.com	No error is displayed, email is valid	Pass
4	To verify that the email field displays error on an already registered user	Email: wardarana082@gmail.com	Error message is displayed that user with this email already exists	Pass
5	To verify that the email field displays error on invalid format	Email: wardarana083gmail.com	Error message is displayed that invalid email format	Pass
6	To verify that the First name field accepts name if it is not already registered	First name: Warda2	No error is displayed, username is valid	Pass
7	To verify that the First name field displays error on an already registered user	First name: Warda	Error message is displayed that user with this name already exists	Pass
8	To verify that the password field accepts password if valid	Password: wardaRana2@02	No error is displayed, password is valid	Pass
9	To verify that the password field displays error on invalid format	Password: warda20	Error message is displayed that : "Password must contain at least 8 characters, including one uppercase letter, one lower case letter, one number, and one special character"	Pass

Unit Testing 2: Login with valid and invalid credentials and Reset Password functionality.

Testing Objective: To ensure that the login and reset Password functionalities are working correctly with both valid and invalid inputs.

No.	Test case/Test script	Attribute and value	Expected result	Result
1	To verify that the username field shows error if user is not registered.	Username: Warda123	Error message is displayed "Incorrect username or password"	Pass
2	To verify that password field displays error if it does not match with the existing one.	Password: wardaRanaaa2@02	Error message displayed "Incorrect username or password"	Pass
3	To verify that the sign in button works correctly	Click on Sign in button	Sign in button redirects user to the home page of InteriorX	Pass
4	To verify that the "keep me logged in" checkbox can be checked	Check on the Keep me logged in button	Keep me logged in button checked	Pass
5	To verify that the "Forgot Password" link works correctly	Click on Forgot Password link	Forgot Password link redirects user to the page with Reset password	Pass
6	To verify that the email field in the Reset Password Page shows error if email is not written in correct format	Email: warda	Error message is displayed "Please include an @ in the email address"	Pass
7	To verify that the email field in the Reset Password Page shows error if email does not match with existing one	Email: wardarana082@gmail.com	Error message is displayed "Email not found"	Pass
8	To verify that the user is directed to the next page if the email entered in the email field of Reset Password Page is correct	Email: wardarana083@gmail.com Click on Reset Password button	User is directed to page with fields new password and Confirm password	Pass
9	To verify that the error is displayed in the Reset Password Page when the New password and Confirm Password does not match	New Password: Helloworld!12 Confirm Password: Helloworld	Error is displayed "Password does not match"	Pass
10	To verify that the error is displayed when passwords are not written in correct format	New Password: hello Confirm Password: Hello	Error is displayed "Password must contain at least 8 characters, including one uppercase letter, one lower case letter, one number, and one special character"	Pass

11	To verify that the success message is displayed when the new password and confirm password in the Reset Password Page are correct	New Password: Helloworld!12 Confirm Password: Helloworld!12	Success Message displayed "Passwords updated successfully"	Pass
12	To verify that the Back to Login link on the Reset Password Page works correctly	Click on Back to Login link	User is redirected to the Login Page	Pass

Unit Testing 3: Verify Edit Profile functionality.

Testing Objective: To ensure that the information can be updated and profile picture can be setup.

No.	Test case/Test script	Attribute and value	Expected result	Result
1	To verify that the Profile button on the InteriorX home page works properly	Click on Profile button	User is redirected to the edit profile page.	Pass
2	To verify that the first name can be edited by using Edit button.	Click on Edit button in front of first name field	The first name field becomes editable and Edit button changes to Save button	Pass
3	To verify that the last name can be edited by using Edit button.	Click on Edit button in front of last name field	The last name field becomes editable and Edit button changes to Save button	Pass
4	To verify that the Email can be edited by using Edit button.	Click on Edit button in front of email field	The email field becomes editable and Edit button changes to Save button	Pass
5	To verify that the Password can be edited by using Edit button.	Click on Edit button in front of password field	The password field becomes editable and Edit button changes to Save button	Pass
6	To verify that the Phone Number can be edited by using Edit button.	Click on Edit button in front of phone number field	The phone number field editable and Edit button changes to Save button	Pass
7	To verify that the Address can be edited by using Edit button.	Click on Edit button in front of address field	The address field becomes editable and Edit button changes to Save button	Pass
8	To verify that success message is displayed when the save button in any field is clicked	Click on Save button	Success Message displayed "Updated successfully"	Pass

9	To verify that the camera icon below profile picture shows Choose File option when clicked.	Click on camera icon under the profile picture	Choose File option is displayed.	Pass
10	To verify that the Choose File button opens the dialog box from which user can choose picture	Click on Choose File button	The dialog box is displayed from which user can choose picture	Pass
11	To verify that the picture is displayed in the circle when save button is clicked	Choose any picture Click on save button	The profile picture chosen is displayed in the circle	Pass

Unit Testing 4: Verify the saved designs and user history functionalities.

Testing Objective: To ensure that the previous designs of user are saved from where he can edit them.

No.	Test case/Test script	Attribute and value	Expected result	Result
1	Verify that the Saved designs button on toolbar is working properly	Click on Saved Design button	The page with previously saved designs of user are displayed.	Pass
2	Verify that when any design is clicked, it opens that project	Click on a previously saved design project	Opens that respective project where user can edit it	Pass

Unit Testing 5: Designing Tools functionalities.

Testing Objective: To verify that design is created using different designing tools.

No.	Test case/Test script	Attribute and value	Expected result	Result
1	To verify that catalog section is opened upon clicking on chair icon in the design screen	Click on chair icon on the top of design screen	The catalog section is opened	Pass
2	To verify that the items can be clicked from the catalog section	Click on item	The item is selected and user is directed to the design screen to place or draw that item	Pass
3	To verify that user can draw the walls using the mouse	Drawing the walls on the design project using dragging or clicking	The wall is drawn on the design screen when user is dragging or clicking using mouse	Pass
4	To verify that the wall details are shown at the left sidebar of design screen	Click on any wall	The details are displayed on the left side bar	Pass

5	To verify that the item can be entered in Search bar of catalog section	“Windows”	The input is displayed in search bar	Pass
6	To verify that the items are displayed in the Last selected bar in catalog section	Different furniture items	The items which are previously selected are shown here	Pass
7	To verify that user can place any door in the design project	Door	The door is placed where desired and the details of door are shown on left sidebar	Pass
8	To verify that user can place any window in the design project	Window	The window is placed where desired and the details of window are shown on left sidebar	Pass
9	To verify that user can place any furniture item in the design project	Furniture item	The furniture item is placed where desired and the details are shown on left sidebar	Pass
10	To verify that floor detail fields are shown on the left sidebar when floor is clicked.	Click on floor	The floor detail fields are shown on the left sidebar	Pass
11	To verify that the color of floor can be customized when the floor is clicked in 2D or 3D view	Floor Color	The color of floor is set and shown in the view	Pass
12	To verify that the texture of floor can be selected from drop down menu when the floor is clicked in 3D view	Click on texture field Ceramic tile texture	The texture of floor is set and shown in the view	Pass
13	To verify that the textures of wall can be selected from drop down menu when the wall is clicked in 3D view	Click on texture A field Bricks texture Click on Texture B field Painted texture	The texture of walls are changed and shown in view	Pass
14	To verify that the screenshot is captured upon clicking the camera icon on the design screen	Click on Camera Icon	The screenshot of the design project is saved in the device	Pass
15	To verify that the save project button works properly	Click on the save project icon on the design screen	Design project is saved in the downloads folder and database	Pass
16	To verify that the new project button works properly	Click on the new project icon on the design screen	New project is created and empty design screen is displayed	Pass

17	To verify that the load project button works properly	Click on the Load project icon on the design screen	Allows user to select the saved project from his/her own system directory, and then shows that design project on the screen	Pass
----	---	---	---	------

Unit Testing 6: Verification of Education and Inspiration features.

Testing Objective: To ensure that the features of Education and Inspiration are working properly.

No.	Test case/Test script	Attribute and value	Expected result	Result
1	To verify that the Question mark button on design screen works properly	Click on Question mark icon on design screen	The user is directed to Tutorials and Guides Page	Pass
2	To verify that all the information (text and videos) is displayed in the Tutorials and Guides Page	See the page of Tutorials and Guides	All the information about tutorials and guides is shown on screen	Pass
3	To verify that the Tutorial video can be opened	Click the video of "View in 3D"	The video is opened and shown to user	Pass
4	To verify that the Back button on Tutorials and Guides Page works properly	Click on Back button	The user is redirected to the design screen	Pass
5	To verify that the Education and Inspiration button is working properly	Click on Education and Inspiration button on the home page	The page is displayed with 4 options: Ideas, News, Articles, Community	Pass
6	To verify that the Ideas button on the Education and Inspiration Page is working properly	Click on Ideas in the Education and Inspiration Page	The user is directed to the Design Inspiration and Ideas Page	Pass
7	To verify that the playlist can be selected from dropdown menu on Design Inspiration and Ideas Page	Click on "Select a playlist" dropdown menu Select playlist "TV"	The related videos are shown based on the playlist selected from dropdown menu	Pass
8	To verify that the Articles button on the Education and Inspiration Page is working properly	Click on Articles in the Education and Inspiration Page	The Articles page is shown	Pass
9	To verify that the information is shown based on the category selected	Click on category "Room Makeovers"	The related category information (text and pictures) is shown to user	Pass
10	To verify the like functionality on the article	Click on heart icon under Room Makeover article	Like count is increased and number of likes are shown on screen	Pass

11	To verify that all the comments on the article are shown to the user	Click on category "Room Makeovers" Scroll to bottom of page	Comments of users are shown	Pass
12	To verify that the comment can be entered in the "Write a comment" field under article	"Amazing Article"	Comment is entered and displayed in the comment field	Pass
13	To verify the Post comment button functionality	Click the Post comment button	Comment which was written in comment field is shown in the comment section.	Pass
14	To verify that the News button on the Education and Inspiration Page is working properly	Click on News in the Education and Inspiration Page	Latest news and updates are shown	Pass
15	To verify that the latest news and updates are shown in the News Page	Click on News in the Education and Inspiration Page	The latest and trending news about interior design industry are shown to the user	Pass
16	To verify the Read More link functionality	Click on Read More link of "Design Award Winner 2023" news	The user is directed to the website of the related news where the information is shown in detail	Pass
17	To verify that the Community button on the Education and Inspiration Page is working properly	Click on Community in the Education and Inspiration Page	The community section is displayed	Pass
18	To verify that the users' posted designs are displayed in Community Section	Click on Community Section Page	All the designs posted by different users are displayed in this section	Pass
19	To verify that the posted designs in Community section can be liked	Click on Like button on the posted design	The like count is incremented and likes are displayed on the design	Pass
20	To verify that the Image can be selected in Community section	Click on select Image Image "Design1" selected	The image is selected and ready to post	Pass
21	To verify that the Selected Design Image can be posted	Click on Post button	The image is successfully posted in the Community section	Pass

7.2 Functional Testing

The purpose of functional testing is to test the functionality of each module. It ensures that the developed system is according to the specifications.

Functional Testing 1: User Profile verification

Testing Objective: To ensure that the user profile features works properly.

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Verify the signup functionality with valid credentials	Username: NewUser First name: New Last name: User Your email: newuser123@gmail.com Password: CorrectPassword!23	The user is registered successfully and is redirected to sign in page	Directed to Sign In page	Pass
2.	Verify the signup functionality with invalid credentials	Username: ExistingUser First name: Existing Last name: User Your email: existinggmail.com Password: incorrectformat2	Respective error messages display to the user	Error messages displayed	Pass
3.	Verify the login functionality with valid credentials	Username: ExistingUser Your Password: existingPassword!23	The user is redirected to the Home Page of InteriorX	User redirected to the Home Page	Pass
4	Verify the login functionality with invalid credentials	Username: NewUser YourPassword: newpassword12	InteriorX Home Page is not displayed Respective error messages shown on screen	User is not directed to Home page Error messages displayed	Pass

5	Verify that the data can be edited in Edit Profile	<p>First Name: Edited</p> <p>Last Name: Name</p> <p>Email: updatedemail@gmail.com</p> <p>Password: newpasswOrd!23</p> <p>Phone Number: 03321122345</p> <p>Address: abc, city</p> <p>Click on Edit button in front of every field</p> <p>Click on save button after writing new data</p>	Success Message displays to user	Success Message displayed	Pass
6	Verify that profile picture can be set up	<p>Click on Edit button under profile picture</p> <p>Profile picture "MyPic.png" selected</p> <p>Click on Save Button</p>	Profile picture displays to the user	Selected profile picture displayed	Pass
7	Verify the Saved Designs functionality	Click on Saved Designs button on toolbar	The previously saved designs of user are displayed	The previously saved designs of user are displayed	Pass

Functional Testing 2: Design Tools Verification**Testing Objective:** To verify the design creation functionality.

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1	To verify that the items can be selected from the catalog section and placed in the project	Click on “walls” from catalog section Click and drag on design canvas	The items can be placed anywhere in the project	Item placed in the project where desired	Pass
2	To verify that the properties of item are shown on left sidebar when item is selected in design project	Click on windows	The properties of selected item are shown in left sidebar	The properties of selected item are shown in left sidebar	Pass
3	To verify that floor design can be customized when floor is clicked in 3D view	Click on floor in 3D view Choose Color : (blue) Texture: Ceramic Tile	The floor design is updated and shown in the 3D view	The floor design is updated and shown in the 3D view	Pass
4	To verify that the texture of wall can be changed when wall is clicked	Click the wall Choose Texture A: Painted Texture B: Bricks	The wall texture is changed and shown in the 3D view	The wall texture is changed and shown in the 3D view	Pass
5	To verify the Save Project functionality	Click on Save Project icon on the design screen	The project is saved in the database and user system directory.	The project is saved in the database and user system directory.	Pass
6	To verify the New Project functionality	Click on New Project icon on the design screen	The New project opens, empty design screen is shown	The empty design screen is displayed	Pass
7	To verify the Load project functionality	Click on Load Project icon on the design screen Project “ designproject1” selected from user system directory.	The selected project is loaded and displayed on the design screen	The selected project is loaded and displayed on the design screen	Pass

Functional Testing 3: Education and Inspiration verification

Testing Objective: To verify the functionality of Education and Inspiration features.

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	To verify the Tutorials and Guides functionality	Click on Question Mark icon on the design screen	All the tutorial videos are displayed	All the tutorial videos are displayed	Pass
2.	To verify that the design videos are displayed according to the playlist selected	Select “ tv” from dropdown menu.	The related videos are displayed.	The related videos are displayed.	Pass
3	To verify that the articles are displayed according to the category selected	Click on category “Room Makeovers” on Articles Page	The related information is displayed	The related information is displayed	Pass
4	To verify the News functionality in Education and Inspiration	Click on News on Education and Inspiration Page	The latest design industry news and updates are displayed	The latest design industry news and updates are displayed	Pass
5	To verify the Community functionality in Education and Inspiration	Click on Community on Education and Inspiration Page	The Community section is displayed which has all the designs posted by other users	The Community section is displayed which has all the designs posted by other users	Pass
6	To verify the like functionality on Users’ Designs	Click on heart icon on the posted design	The like count is incremented and likes are shown	The likes are shown on the design	Pass
7	To verify the Post functionality in Community Section	Select image “Design1.png” Click on Post button	The design image is posted in Community Section	The design image is posted in Community Section	Pass

7.3 Scenario Based Testing

Scenario-based testing uses realistic user interactions to assess software functionality and usability. It is to ensure that the software meets user expectations in real-world scenarios.

Our module Design Tools requires scenario-based testing for verifying the functionality of design creation in the InteriorX.

No.	Test case/Test script	Test Steps	Attribute and value	Expected result	Result
1	Verify the draw walls functionality	1) Login in the InteriorX 2) Select rooms 3) Start new or previous project 4) Click on catalog (chair icon) 5) Select walls Draw 6) the walls on design screen using mouse 7) Edit the properties of walls which appear on left sidebar.	Walls drawn on canvas	The walls drawn on the screen and properties can be changed	Pass
2	Verify the place doors functionality	1) Login in the InteriorX 2) Select rooms 3) Start new or previous project 4) Click on catalog (chair icon) 5) Search doors in search bar 6) Select “double door” 7) Place door on the design project anywhere. 8) Edit properties of door which appear on left sidebar	Door placed on canvas	The doors placed on the screen and properties can be changed	Pass

3	Verify the place windows functionality	<ol style="list-style-type: none"> 1) Login in the InteriorX 2) Select rooms 3) Start new or previous project 4) Click on catalog (chair icon) Search windows in search bar 5) Select “Double Glazed window” 6) Place window on the design project anywhere. 7) Edit properties of window which appear on left sidebar 	Windows placed on canvas	The windows are placed on the screen and properties can be changed	Pass
4	Verify the place furniture items functionality	<ol style="list-style-type: none"> 1) Login in the InteriorX 2) Select rooms 3) Start new or previous project 4) Click on catalog (chair icon) 5) Search beds in search bar 6) Select “Double Beds” 7) Place item on the design project anywhere. 8) Edit properties of item which appear on left sidebar 	Furniture item placed on canvas	The furniture items are placed on the screen and properties can be changed	Pass
5	Verify that the floor appearance can be customized	<ol style="list-style-type: none"> 1) Click on 3D view icon on the design screen. 2) Click on floor. The properties of floor are displayed on the left sidebar. 3) Change color ,thickness, texture of floor. 4) 	<p>Click on floor in 3D view</p> <p>Color : blue</p> <p>Texture: Ceramic Tiles</p>	The floor appearance is changed and shown in 3D View	Pass

7.4 Integration Testing

Integration testing is used to test the interface between different modules. It ensures that the data is correctly passed from one module to another.

Integration 1: Floor plan and saved designs

Testing Objective: To verify integration between user profile module and design tools module

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Verify that when the room is selected and project is started, then the design screen is displayed.	Selected room Design screen	The design screen is opened when the user selects the room and starts the project	The design screen opened for designing the selected room	Pass
2	Verify that the saved designs will be displayed to user according to logged in username.	Click on Saved Designs button on the toolbar section.	The saved designs of user who has logged in the account are displayed to the user	The saved designs of user who has logged in the account are displayed to the user	Pass

Integration 2: User Profile and Education Inspiration

Testing Objective: To verify integration between user profile module and education and inspiration module.

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Verify that the Education and Inspiration features are accessible when the user logs in the account	Log in Click on Education and Inspiration button	The Education and Inspiration features: Ideas, Community, Articles, News are displayed to the user.	The Education and Inspiration features: Ideas, Community, Articles, News are displayed to the user.	Pass
2.	Verify that the posted designs in the Community Section will get the username of the user who posts the designs.	Username of the person who post the design	The username of user is displayed with the design image posted	The username of user is displayed with the design image posted	Pass

Integration 3: Education , Inspiration and YouTube

Testing Objective: To verify integration between education and inspiration module and YouTube.

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Verify that the related videos are fetched from YouTube channel when the playlist is selected from dropdown menu of Design Inspiration and Ideas Page	Playlist “tv” selected from dropdown menu	The related YouTube channel videos are displayed to the user	The related YouTube channel videos are displayed to the user	Pass

Integration 4: Education , Inspiration and News Website

Testing Objective: To verify integration between education and inspiration module and News Website

No.	Test case/Test script	Attribute and value	Expected result	Actual result	Result
1.	Verify that the news are fetched from News Website when the user clicks News button on Education and Inspiration Page	Click on News button on Education and Inspiration Page	The latest industry news and updates are fetched from News Website and shown to user.	The latest industry news and updates are fetched from News Website and shown to user.	Pass

8. Plagiarism Report

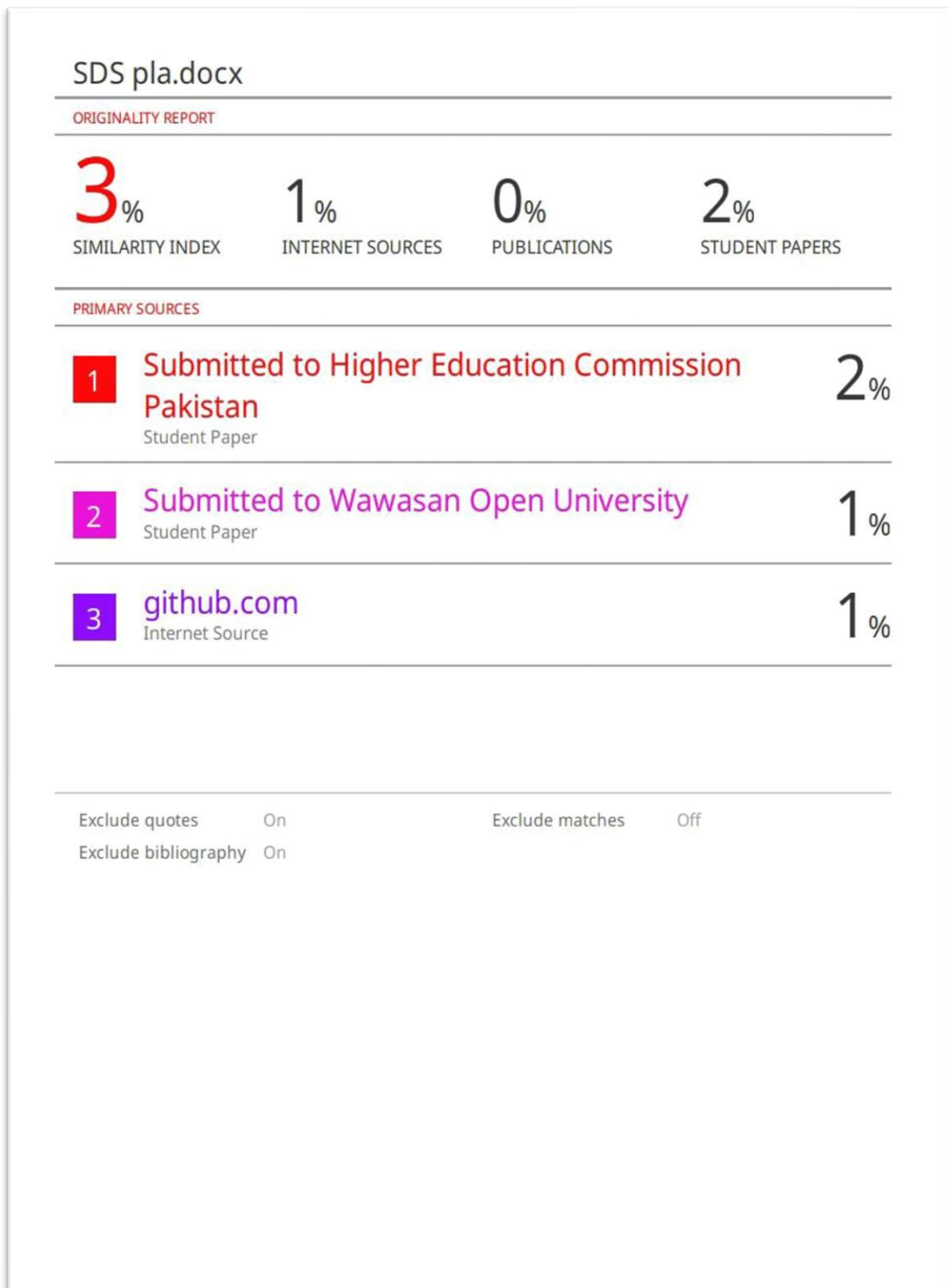


Figure 55 Plagiarism Report