FYP-II Final Report

F22-103-D-IntelleTect

Momin Shahzad 19I-1652

Ali Raza 19I-1660

Mohammad Umer 19I-1787

Supervisor: Ms. Noor Ul Ain



Table of Contents

[1. Introduction: 5](#_Toc136808972)

[2. Project Vision 5](#_Toc136808973)

[2.1. Problem Statement: 5](#_Toc136808974)

[2.2. Business Opportunity 6](#_Toc136808975)

[2.3. Objectives 6](#_Toc136808976)

[2.4. Project Scope: 7](#_Toc136808977)

[2.4.1. Project Components: 7](#_Toc136808978)

[2.4.2. Work Division: 7](#_Toc136808979)

[2.5. Constraints 8](#_Toc136808980)

[2.6. Stakeholder Description: 8](#_Toc136808981)

[2.7. Stakeholder Summary: 9](#_Toc136808982)

[3. Use Diagram & Requirements Analysis 10](#_Toc136808983)

[3.1. Use-Case Diagram: 10](#_Toc136808984)

[3.2. List of Features: 10](#_Toc136808985)

[3.3. Functional Requirements: 11](#_Toc136808986)

[3.4. Non-Functional Requirements: 11](#_Toc136808987)

[3.5. Quality Attributes: 11](#_Toc136808988)

[4. High-Level Use Case: 11](#_Toc136808989)

[5. Expanded Use-Cases: 13](#_Toc136808990)

[6. Project Diagrams 20](#_Toc136808991)

[6.1. Domain Model: 20](#_Toc136808992)

[6.2. System Sequence Diagram: 21](#_Toc136808993)

[6.3. SSD for Authorization: 22](#_Toc136808994)

[6.4. SSD for Floor Plan Generative Model: 22](#_Toc136808995)

[6.5. SSD for Storing Designs 23](#_Toc136808996)

[6.6. Component Diagram: 23](#_Toc136808997)

[6.7. Architectural Diagram 24](#_Toc136808998)

[6.8. Activity Diagram 25](#_Toc136808999)

[7. Package Diagram: 26](#_Toc136809000)

[8. Implementation timeline: 27](#_Toc136809001)

[8.1. Iteration 1 (Sep 2022 – Oct 2022) 27](#_Toc136809002)

[8.2. Iteration 2 (November 2022 – Dec 2022) 27](#_Toc136809003)

[8.3. Iteration 3 (November 2022- February 2023) 27](#_Toc136809004)

[8.4. Iteration 4 (March 2023 – April 2023) 27](#_Toc136809005)

[8.5. Iteration 5 (April 2023 – May 2023) 27](#_Toc136809006)

[9. Implementation Details 28](#_Toc136809007)

[9.1. Methodology: 28](#_Toc136809008)

[9.2. Gathering and processing of dataset: 28](#_Toc136809009)

[9.3. Training and Testing of Models: 28](#_Toc136809010)

[9.4. Training Files. 29](#_Toc136809011)

[9.4.1. Data-Loading Part. 29](#_Toc136809012)

[9.4.2. Training 30](#_Toc136809013)

[9.5. Model After Training: 32](#_Toc136809014)

[9.5.1. Details of the above used functions 33](#_Toc136809015)

[9.6. Design and Development of a Website: 35](#_Toc136809016)

[10. Iteration Plan: 37](#_Toc136809017)

[10.1. Iteration 1 (Sep 2022 – Oct 2022) 37](#_Toc136809018)

[10.2. Iteration 2 (November 2022 – Dec 2022): 37](#_Toc136809019)

[10.3. Iteration 3 (Dec 2022- Jan 2023): 37](#_Toc136809020)

[10.4. Iteration 4 (Feb 2023 – March 2023): 37](#_Toc136809021)

[10.5. Iteration 5 (April 2023 – May 2023): 37](#_Toc136809022)

[11. User manual. 38](#_Toc136809023)

[11.1. Login or Signup. 38](#_Toc136809024)

[11.2. User Commands. 39](#_Toc136809025)

[11.3. Test Cases. 40](#_Toc136809026)

[12. Bibliography 43](#_Toc136809027)

Anti-Plagiarism Declaration:

This is to declare that the above FYP report produced under the:

Title: IntelleTect

is the sole contribution of the author(s) and no part hereof has been reproduced on as it is basis (cut and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is determined.

Date: October 23, 2022

Student 1

Name: Momin Shahzad

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student 2

Name: Ali Raza

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student 3

Name: Muhammad Umer

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Supervisor

Name: Noor Ul Ain

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Introduction:

IntelleTect is a web-based application that generate floor plans using the textual/voice description. It is an input dependent software that allows user to enter the description through voice or text and based on the input passed it create floor plans. The way it works is, the description passed as input is converted into graph, then passed to neural networks which then generate the floor plan that user can view.

Furthermore, with the output generated you can get the idea of how the floor looks like and if you are not satisfied with the output, you can customize the description to get the desired output. The output is not just a single output, there would be multiple options generated by the model, out of which you select the most viable option for your case.

We discovered after doing a lot of research that there are already applications out there that does similar jobs compared to ours. Two types of applications are available i.e., Web/Desktop apps and Mobile apps. The problem with web/desktop applications is they require technical knowledge to use them. On the other side, while current mobile applications are much simpler to use, but they get less functional as a result. So, using IntelleTect the user does not any superior knowledge of design, they can just pass the design requirement in text form and IntelleTect will generate the design of floor plans.

# Project Vision

## Problem Statement:

Home designing is a complex task that normally requires architects and people of field to finish with their professional skills and tools. Imagine a worker who has no prior knowledge to what a floor plan is and what tools (like AutoCAD) to use to drag and drop and match the much complex strategies used by an architect to design a floor plan for a house, can generate a whole house design just by giving its description which he desires.

Even if the floor plan is generated by an architect can result in dissatisfaction which is not a good thing. Our software can impact directly for people having financial reasons or lack of knowledge of different tools. Also, our software can provide assistance to the architect as well to make process of generating floorplans easier.

A potential solution to the problems discussed is our software IntelleTect does not require any prior experience of tools to design it. You can directly provide the description to generate designs. Moreover, it can also help architect to make the process smoother and less contempt.

## Business Opportunity

Considering the fact that there are multiple similar software’s out there available in the market that user can use to build the floor architecture/plan, but as discussed earlier they require the knowledge of those software. Also, the fact that every normal person has a dream to build his own house in his lifetime and first step towards building house his making the floorplan which is a prototype to see how everything looks. So, this is like almost a never-ending loop until all the land on earth is occupied. This is where IntelleTect will come into play build different floorplans to select the one desired from generated designs.

* 1. Objectives:

The objective of this project is to provide an interface to users to put their envisioned design in front of themselves without the need of outside assistance and with none to least expense spared by them. They do not need to learn new techniques tools or even pay thousands of rupees out of their pocket to architects or designing companies to let them see their house design up to some match of what they originally want but to have an exact image of their mind portrayed onto the screen itself.

It also eliminates the risk of unsatisfactory service, financial difficulty and layman illiteracy on the respective tools that are meant to design a project like this in real time. Helping both the user and also the company whoever incorporates IntelleTect in their designing phase.

## Project Scope:

### Project Components:

1. Web Application that takes in audio/textual input and generates 2D house design.
2. Save/Download feature on the web app that helps store old projects.
3. Customize your description if you don’t like the output.
4. Deep learning models that take in input and generates a 2D house design.
5. Web application for visualizing and viewing designs and taking user input.

### Work Division:

#### Member#1**:** Momin Shahzad – i191652

1. Creating/training the language conditioned texture generative (LCT-GAN) model
2. Testing the output generated by the HPGM (house plan generative model).
3. Front end Development and integration of the model with it.

#### Member#2: Ali Raza – i191660

1. Taking input from the website and integrating it to the model.
2. Mapping the data onto the model.
3. Graph conditioned layout prediction network (GC-LPN) creation/training.

#### Member#3**:** Muhammad Umer – i191787

1. Accuracy evaluation based on the IOU.
2. Using the Stanford scene parsing software to map the description in the dataset.
3. Mapping the output and generating evaluation on the basis of realism, compatibility and diversity.

## Constraints

1. Intelletect does not generate a 3d floor plan
2. 3d plan requires a different type of dataset which has the 360 views of every floor but in our case single we have to process the text and then generate it will take much time to train and process.
3. Intelletect will process the size of room in number. It is easier for the model to understand the number than words like small, large etc.
4. The material of the rooms if mentioned in description, would be shown in the design generated if the model has seen that type of material while training, which means if available in the dataset.
5. Users have to verify yourself from related department about the design you have selected whether it is allowed or not.
6. Users would not be able to customize the floorplan once generated by the software. They will need to customize the input to see the different floorplans.

## Stakeholder Description:

The stake holders consist of

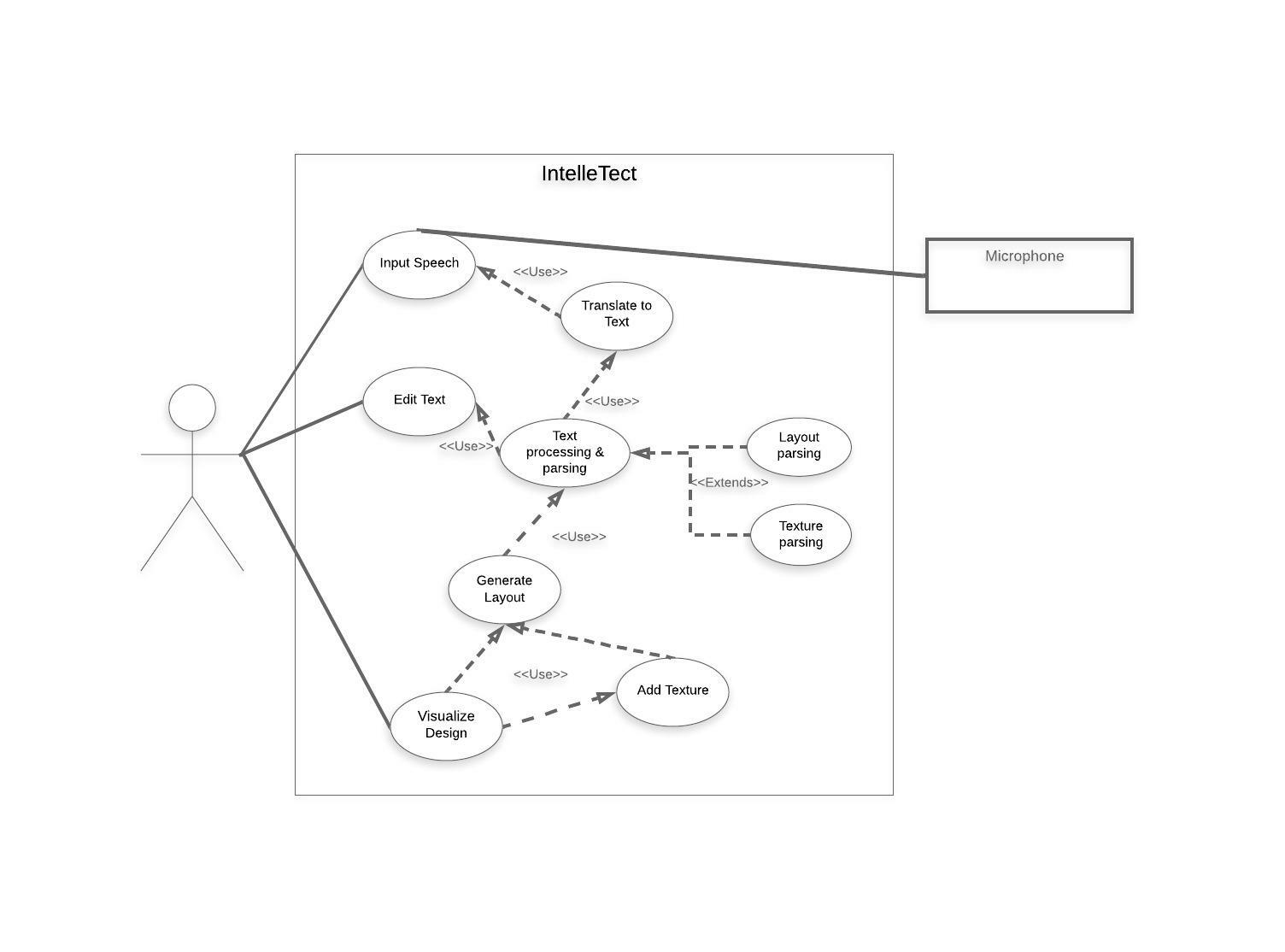
* End-users that will utilize the system. Can be a free user or a company that incorporates such design in the designing phase of the construction of the property they are working on.
* This also includes the development team. Back end, front end and the testing team.
* Project manager can also be included as a stakeholder.

## Stakeholder Summary:

|  |  |  |
| --- | --- | --- |
| Name: | Representation: | Role: |
| Actor  Architect, non-professional, designer | End-User | * Inputs data for floor plan generation. * Uses the product for generating and using the product output. |
| Developers | Development Team | * Stakeholder will be responsible for delivering the finished web-application. * Stakeholder is responsible for maintaining the product documentation. |
| Supervisor | Project Manager | * The Stakeholder is   responsible for managing the development team.   * Stakeholder is Responsible for the approving of documentation. |

# Use Diagram & Requirements Analysis

## Use-Case Diagram:



## List of Features:

* Speech to text translation
* Text input preprocessing
* Text parsing
* Text to Layout generation
* Training of Deep Learning models
* Analysis of Deep Learning models
* Output image generation

## Functional Requirements:

* System should take voice input
* System should take text input
* System should generate design layout
* System should record the output generated

## Non-Functional Requirements:

* System should allow editing of the text input after voice recognition to allow change in descriptions.
* System should recognize voice delay (2 sec) to stop speech input.
* System should display the generated image on a web-application
* System should allow voice input after user trigger/button press

## Quality Attributes:

* Reliable
* Correctness
* Efficiency

# High-Level Use Case:

|  |  |
| --- | --- |
| UC01 |  |
| Use Case: | Input Speech |
| Actors: | Client, Architect |
| Type: | Primary |
| Description: | The client speaks and describes the desired floor plan and what it should include. |

|  |  |
| --- | --- |
| UC02 |  |
| Use Case: | Translate to Text |
| Actors: | System |
| Type: | Primary |
| Description: | Obtained Speech will be translated into text form. |

|  |  |
| --- | --- |
| UC03 |  |
| Use Case: | Edit text |
| Actors: | Client, Architect |
| Type: | Primary |
| Description: | Edit and append translated text to add descriptions. |

|  |  |
| --- | --- |
| UC04 |  |
| Use Case: | Text parsing and processing |
| Actors: | System |
| Type: | Primary |
| Description: | Descriptions of the floor plan will be extracted from the text. |

|  |  |
| --- | --- |
| UC05 |  |
| Use Case: | Generate layout |
| Actors: | System |
| Type: | Primary |
| Description: | Using the linguistic descriptions obtained a layout or mapping of the design will be produced. |

|  |  |
| --- | --- |
| UC06 |  |
| Use Case: | Add texture |
| Actors: | System |
| Type: | Primary |
| Description: | Using the linguistic descriptions obtained to create textures for the layout produced and finalize design. |

|  |  |
| --- | --- |
| UC07 |  |
| Use Case: | Visualize Design |
| Actors: | System |
| Type: | Primary |
| Description: | Display the finalized output and process it into an image. Output will be an image that the client can view. |

# Expanded Use-Cases:

|  |  |
| --- | --- |
| UC01 | |
| Use Case: | Input Speech |
| Scope: | IntelleTect Input process |
| Level: | User-goal |
| Description: | The user speaks and describes the desired floor plan and what it should include. |
| Primary Actor: | Client, Architect |
| Stakeholders: | * Client/ Architect: Wants to input and describe the floor design. * Developers: Want to be able to take descriptive features to run models. |
| Preconditions: | Voice input Trigger/button is pressed |
| Success Guarantee: | Speech and voice recording is taken in real-time. Data is sent to Text translation and converted to text. |
| Main Success Scenario: | Actor Action:   1. Client opens web-application 2. Speech input button is pressed |
|  | System Responsibility:  3. Voice recording is saved  4. Translates text |
| Tech | * Microphone |

|  |  |
| --- | --- |
| UC02 | |
| Use Case: | Translate to Text |
| Scope: | IntelleTect data processing |
| Level: | User-goal |
| Description: | Obtained Speech will be translated into text form. |
| Primary Actor: | System |
| Stakeholders: | * Developers: Want to be able to take descriptive features to run models. |
| Preconditions: | Voice input has been taken. |
| Success Guarantee: | Speech has been converted and translated into textual form |
| Main Success Scenario: | Actor Action: |
|  | System Responsibility:  1. Voice recording is saved  2. Speech is Translated to text  3. Text is displayed |

|  |  |
| --- | --- |
| UC03 | |
| Use Case: | Edit text |
| Scope: | IntelleTect Input process |
| Level: | User-goal |
| Description: | The user speaks and describes the desired floor plan and what it should include. |
| Primary Actor: | Client, Architect |
| Stakeholders: | * Client/ Architect: Wants to input and describe the floor design. * Developers: Want to be able to take descriptive features to run models. |
| Preconditions: | * Web-Application is opened. * Voice input is translated to text. |
| Success Guarantee: | When the Text translation output is appended. |
| Main Success Scenario: | Actor Action:  1.Client enters text into displayed text output |
|  | System Responsibility:  2. New descriptions and text is recorded  3. Text is sent for Parsing |

|  |  |
| --- | --- |
| UC04 | |
| Use Case: | Text parsing and processing |
| Scope: | IntelleTect data processing |
| Level: | User-goal |
| Description: | Descriptions of the floor plan will be extracted from the text. |
| Primary Actor: | System |
| Stakeholders: | * Developers: Want to be able to take layout and texture descriptions to allow inference. |
| Preconditions: | * Voice input is taken. * Text descriptions have been finalized. |
| Success Guarantee: | When layout and texture descriptions are separated and extracted. |
| Main Success Scenario: | Actor Action: |
|  | System Responsibility:  1. Extract layout descriptions of floor design  3. Extract Texture descriptions of floor design |

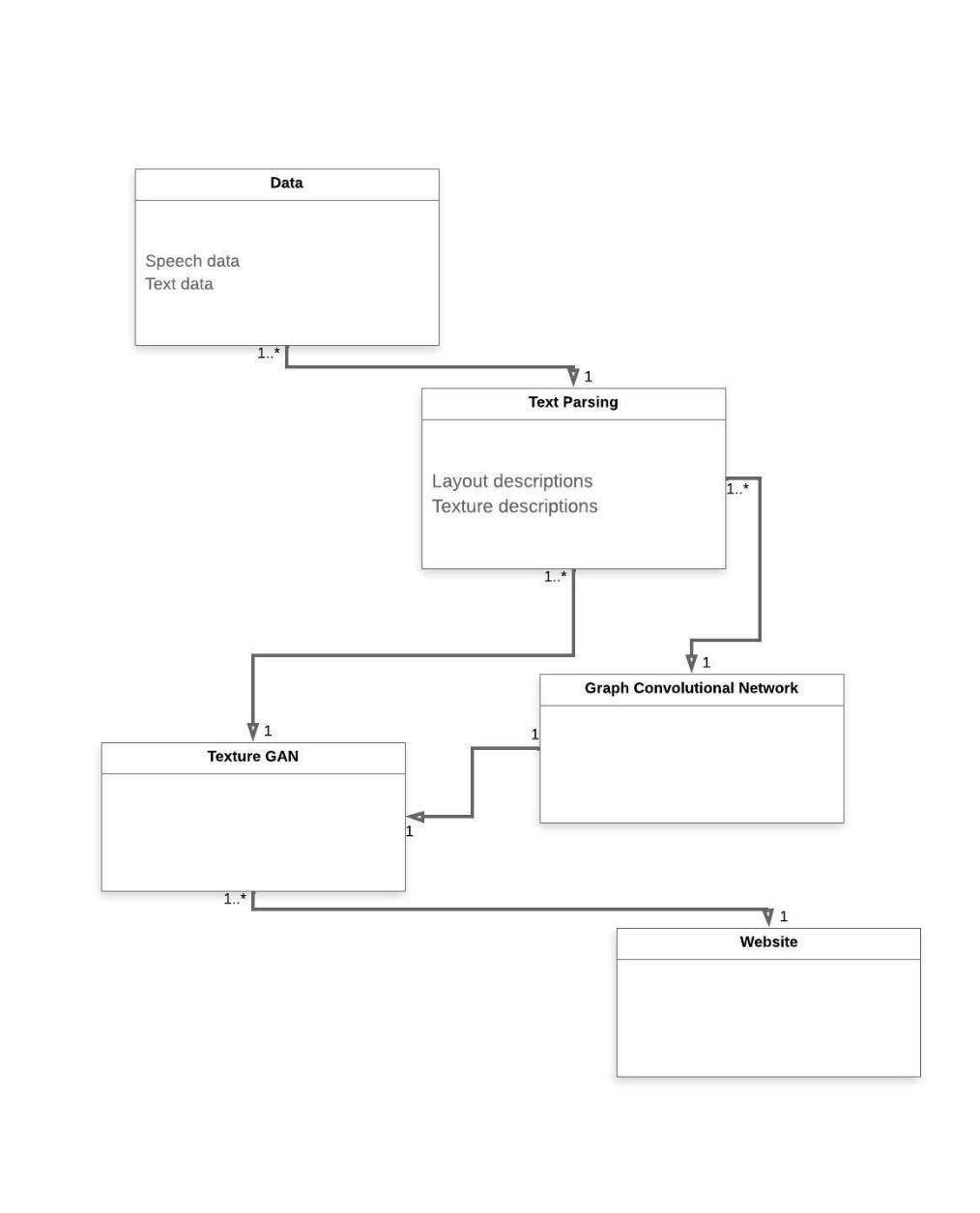
|  |  |
| --- | --- |
| UC05 | |
| Use Case: | Generate layout |
| Scope: | IntelleTect Model processing |
| Level: | User-goal |
| Description: | Using the linguistic descriptions obtained a layout or mapping of the design will be produced. |
| Primary Actor: | System |
| Stakeholders: | * Client/ Architect: Wants the structure and map of the design. * Developers: Want to be able to create the mapping of the linguistic descriptions. |
| Preconditions: | * Voice input / Text editing * Text translation * Text processing |
| Success Guarantee: | Layout of the design is generated. |
| Main Success Scenario: | Actor Action: |
|  | System Responsibility:  1. Parsed text is input in model  2. Model generates layout |

|  |  |
| --- | --- |
| UC06 | |
| Use Case: | Add texture |
| Scope: | IntelleTect Model processing |
| Level: | User-goal |
| Description: | Using the linguistic descriptions obtained to create textures for the layout produced and finalize design. |
| Primary Actor: | System |
| Stakeholders: | * Client/ Architect: Wants the structure and map of the design. * Developers: Want to be able to create the mapping of the linguistic descriptions. |
| Preconditions: | * Layout has been generated |
| Success Guarantee: | Textures have been added to the layout. |
| Main Success Scenario: | Actor Action: |
|  | System Responsibility:  1. Parsed text is input in model  2. Model generates layout |

|  |  |
| --- | --- |
| UC07 | |
| Use Case: | Visualize Design |
| Scope: | IntelleTect Model processing |
| Level: | User-goal |
| Description: | Display the finalized output and process it into an image. Output will be an image that the client can view. |
| Primary Actor: | System |
| Stakeholders: | * Client/ Architect: Wants the floor design. * Developers: Want to be able generate output of the models. |
| Preconditions: | * Layout has been generated * Textures has been mapped to layout |
| Success Guarantee: | Image of the final floor design is displayed. |
| Main Success Scenario: | Actor Action: |
|  | System Responsibility:  1. merge texture and layout  2. Generate Image |

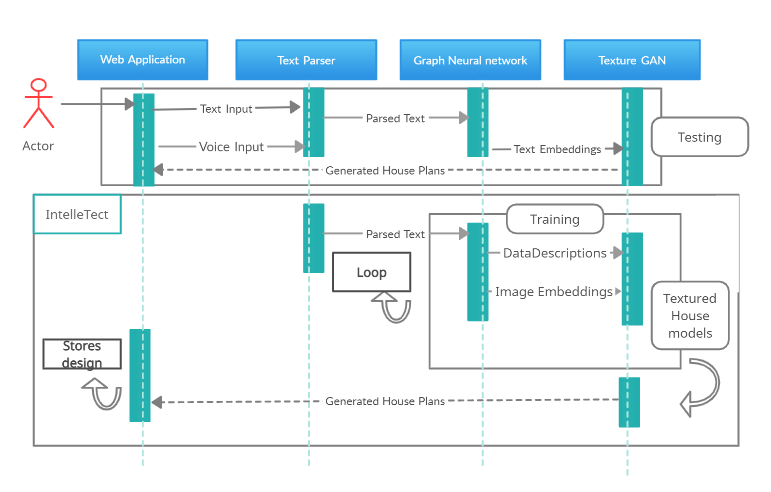
# Project Diagrams

## Domain Model:

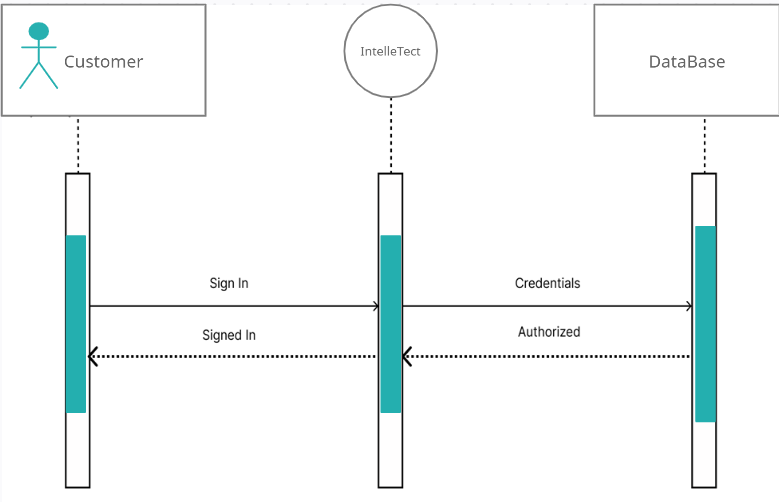
This domain model shows the objects present in our domain and the relationships they have with one another.

## System Sequence Diagram:

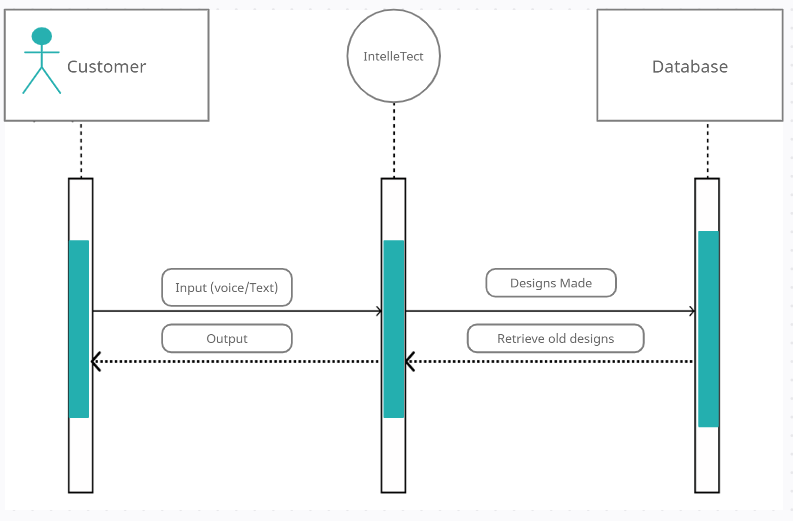
The sequence diagram shows that the user size of the top side, which uploads the floorplan and some details after which the system on the bottom side recognizes the floor plan, applies post processing, and generate floor plan. After this, the recognized floorplan with the cost estimate and information is returned. The user can also view 2d model which the system generates.



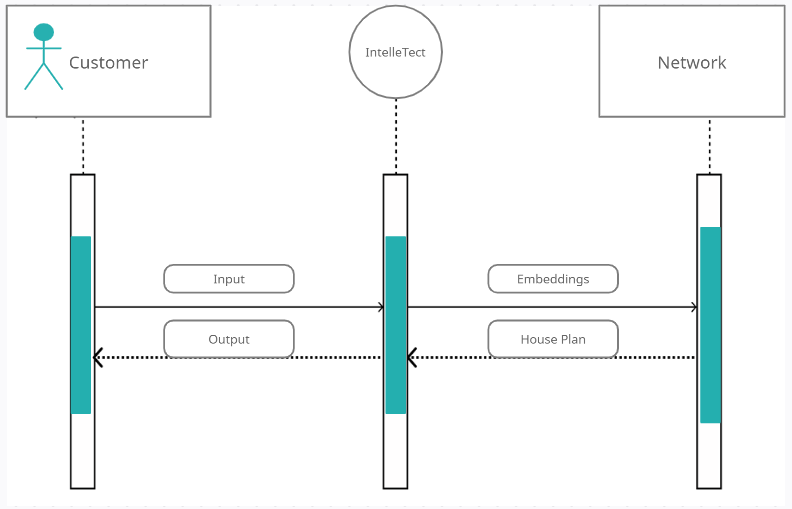
## SSD for Authorization:



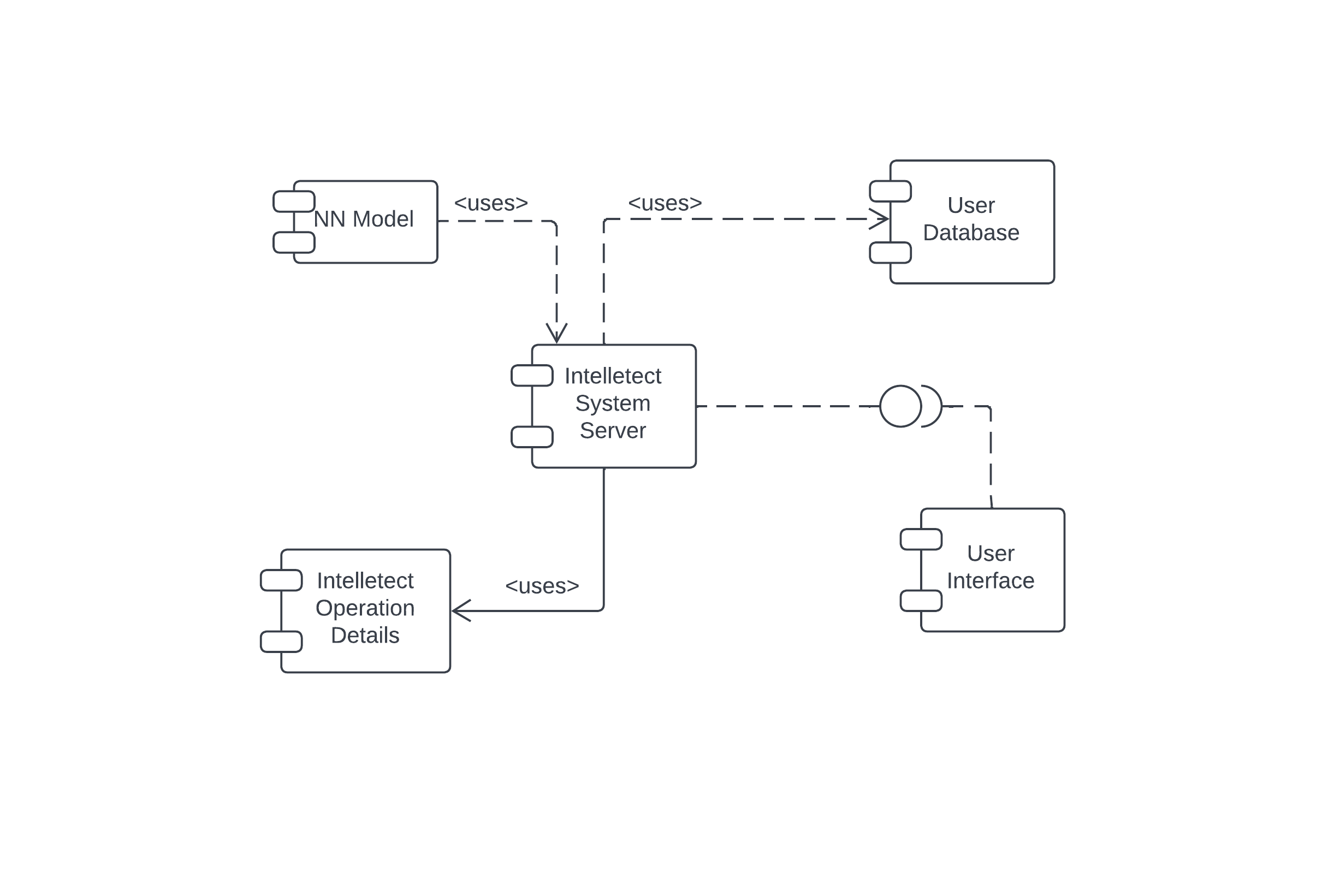
## SSD for Floor Plan Generative Model:



## SSD for Storing Designs

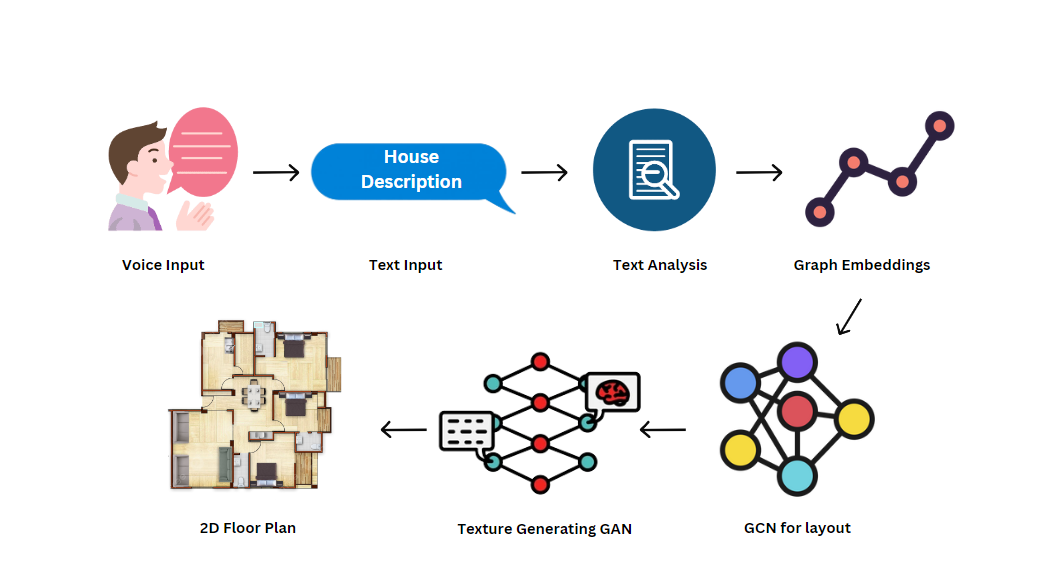


## Component Diagram:

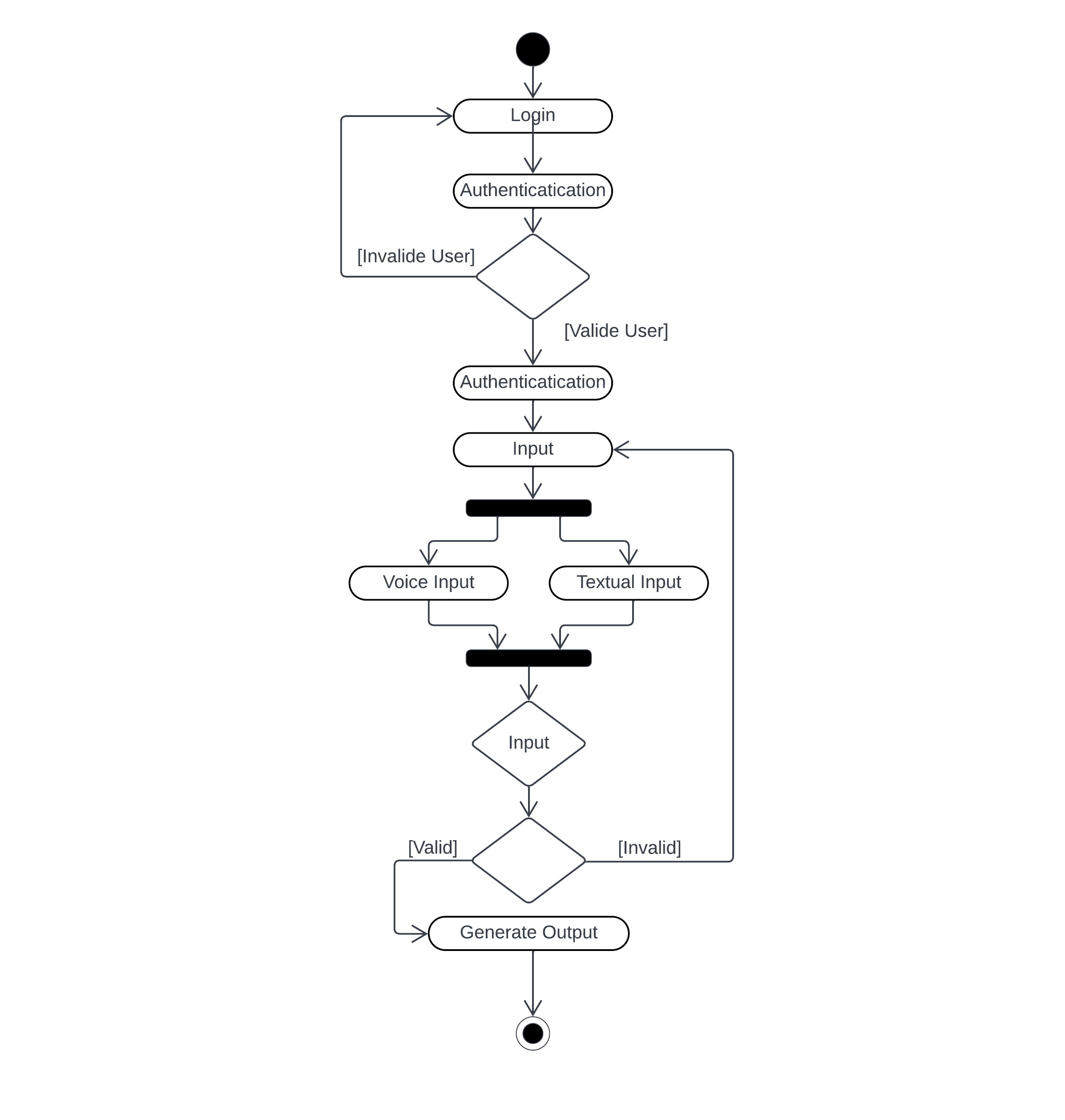


## Architectural Diagram

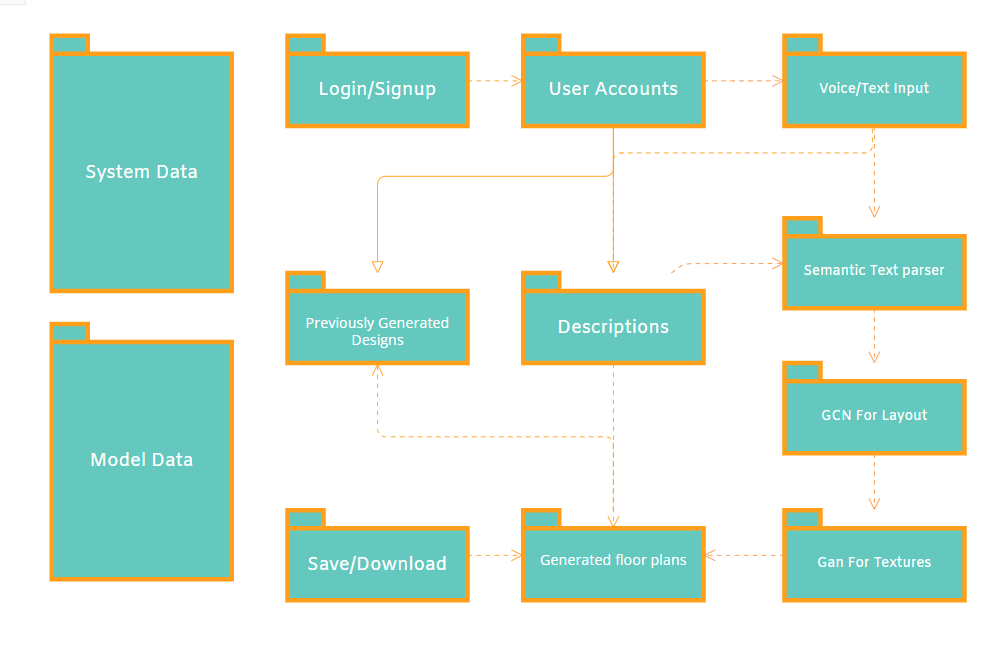
(Client / Server Architecture) The architecture that our project follows is primarily a Client/Server model.The web app acts as a client as it sends the text description to the server that processes the text using the machine learning model and returns generated image as response. A 2d image generation module is part of the app in the form of a component. This will communicate with the server, and generate the 2D image with in the app. Furthermore, the central repository will provide the registered users a platform to interact on.

****

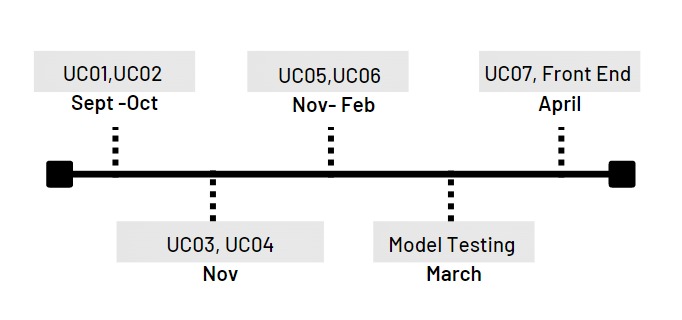
## Activity Diagram

****This is how our activities are coordinated to provide a service which can be at different level of abstraction.

# Package Diagram:

The Below Diagram shows the structure of the front and backend of the software, it highlights the core responsibilities of the front end i.e., the website of software and what services is it willing to provide for its users, and also the models at the backend that are responsible for the front end to work.

# Implementation timeline:

****

## Iteration 1 (Sep 2022 – Oct 2022)

1. Input Speech
2. Translate to Text
3. Annotation

## Iteration 2 (November 2022 – Dec 2022)

1. Edit text
2. Text parsing and processing

## Iteration 3 (November 2022- February 2023)

1. Generate layout

## Iteration 4 (March 2023 – April 2023)

1. Add texture
2. Design of final GUI,.

## Iteration 5 (April 2023 – May 2023)

1. Testing and deployment on website.

# Implementation Details

## Methodology:

1. Gather all the dataset required for this project.
2. Parsing the dataset and text input.
3. Trying to get the maximum accuracy using Generative models.
4. Designing of GUI to show our user their desired house design.

## Gathering and processing of dataset:

First, we gather data of multiple house plans and parse it to find out different labels of the rooms and details of the house. The challenge will be to get textual information out of the visual data, but for that we will use the scene graph parser.

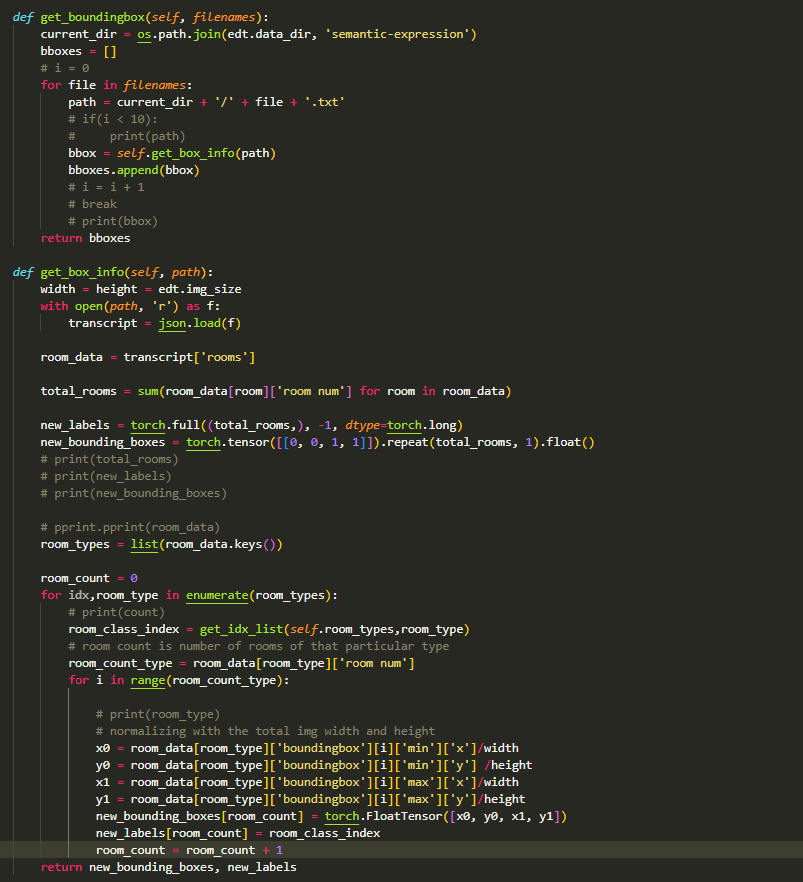
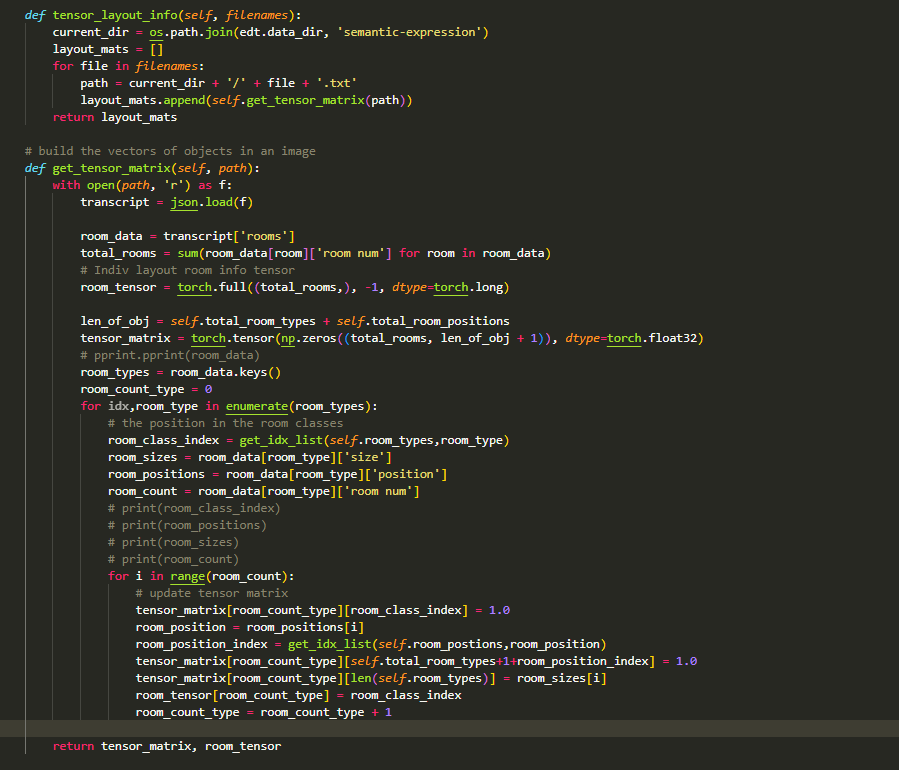
## Training and Testing of Models:

For this we used a house Plan generative model including the following components.

1. A graph convolutional network which takes in image embeddings and maps them with the text input.
2. A language conditions GAN to interpret those embeddings and provide a house plan according to the language descriptions given.
3. The house plan is generated on the bases of bounding box regression each bounding box contains vector values containing the embedding s from the user input.
4. Th user input describes the length width and direction of the bounding box hence those embedding make a clear shape pf the room and where it is.
5. Multiple rooms are then compiled this way and paced on their respected positions, overlapping is eliminated by measuring the distance of each overlap form the center of the bounding box and the over lapped region.
6. The windows and doors are added on the basis of a simple criteria of what wall is the biggest has the window and what wall is facing inwards has a door on it.
7. After the generation, an f1 score is maintained to check the designs on the basis of its precision and recall.

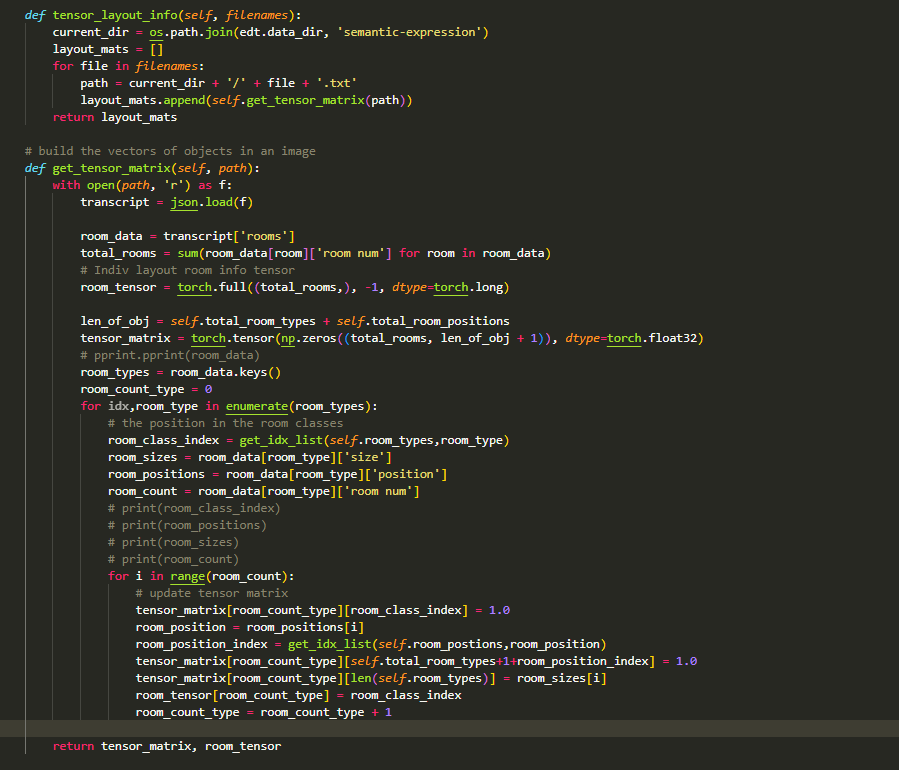
## Training Files.

### Data-Loading Part.

* Total Training Data: 1600 (textual content-information with their layouts) – These are randomly selected from general data, that is 2000 houses with layout descriptions.
* Matrix storing connections among nodes (rooms). It is accomplished for every layout.
* The matrix shops connections among nodes or rooms in a format. This matrix is created for every layout, that means that each layout has its very own corresponding matrix. The connections among nodes or rooms can be used to investigate the format and discover patterns or relationships between specific factors in the layout.
* Bounding Box data of every format. A bounding box is a square field that surrounds an object or institution of gadgets in a photograph. In the context of layouts, the bounding field information provides the scale and position of the rectangle that surrounds the whole layout. These records can be used to investigate and examine distinct layouts, or to crop and resize the layouts for numerous applications.
* Information about layouts (positions/wide variety of rooms and many others.) of every room. This line refers back to the facts available approximately the format of every room in a given space. These facts can encompass the location of each room inside the format, as well as the range of rooms inside the format.

### Training

#### GCN NETWORK.

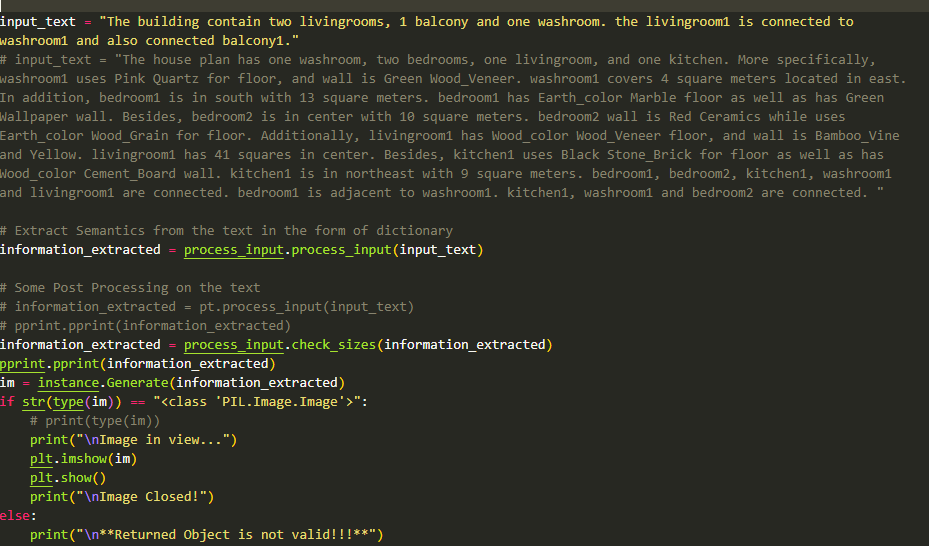


#### **Text Description automatically generated**Bounding Box Network

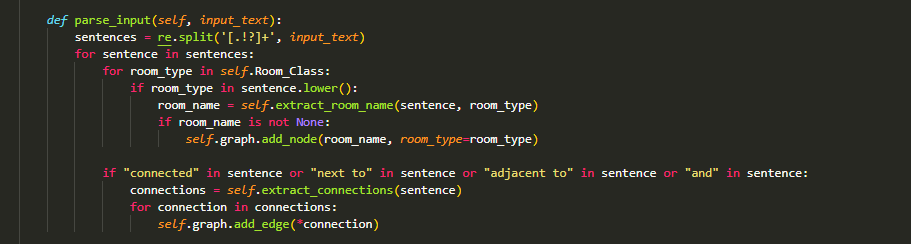
#### **Text Description automatically generated**Training Function:

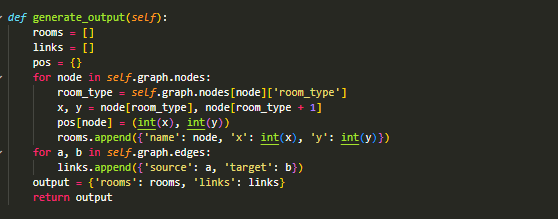
#### **Text Description automatically generated** Training class:

## Model After Training:



### Details of the above used functions

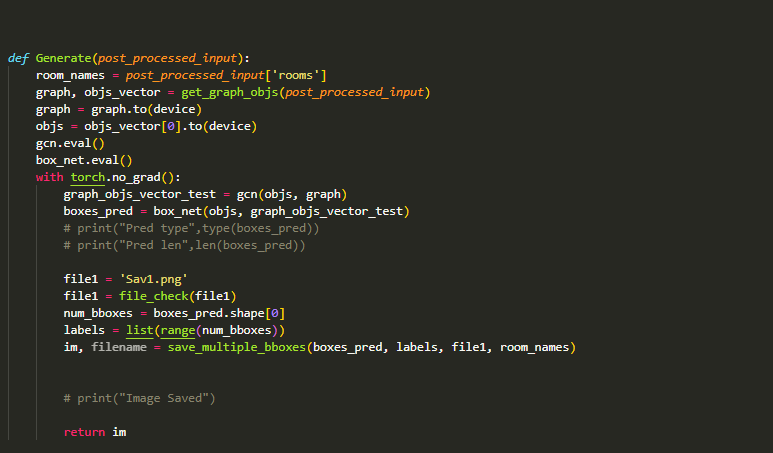


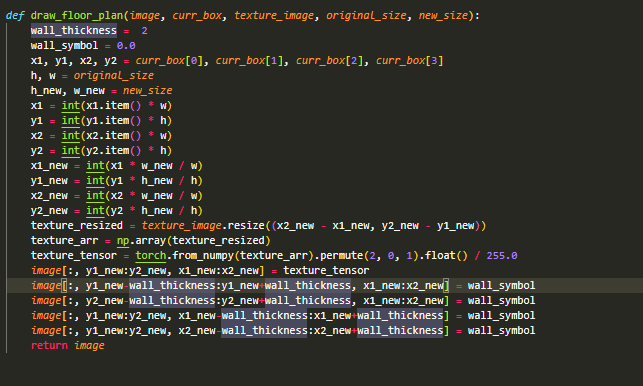


Generate function takes the dictionary containing the semantics extracted from the text and uses the trained models to generate the output.

The function takes as input a dictionary containing semantics extracted from a text, and uses trained models to generate an output. The function uses natural language processing (NLP) techniques to extract the semantics from the text, which might include identifying keywords, entities, and relationships between different elements of the text, in our case the layout.

The trained models used by the have been trained on a large dataset of similar texts, and are able to recognize certain patterns or structures within the text. These models include language models, which are trained to predict relationships pf the entities with each other.

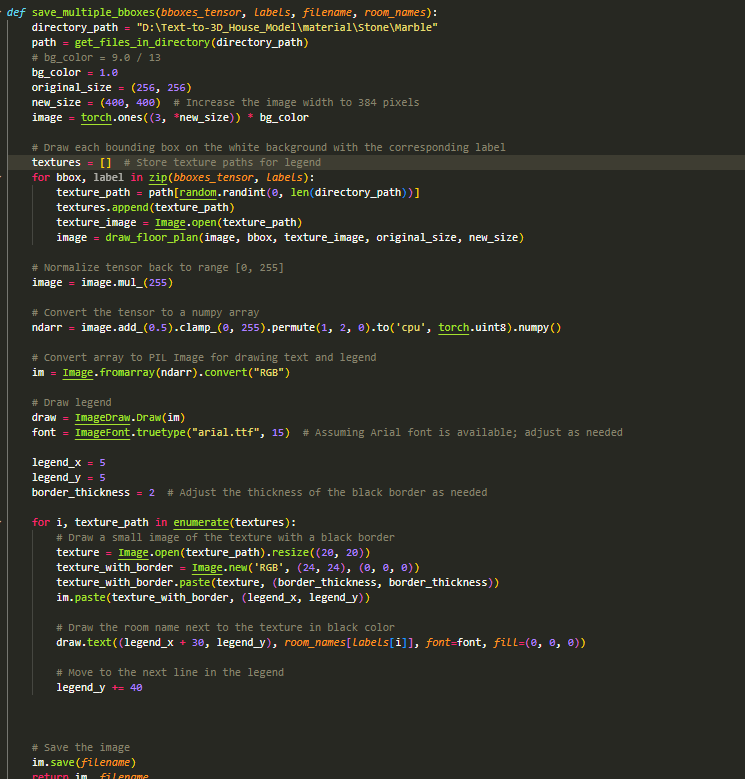
Once the input dictionary has been processed by the trained models, the function generates an output based on the results of the models. This output includes a synthesized version of the input text in other words a visual a summary or abstract of the input text.

The below two functions are used to put the details of bounding box and texture to be filled in the bounding box and on the image.

The two functions, "Draw\_floor\_plan" and "save\_multiple\_bboxes", are used to put details of bounding boxes and textures on an image of a floor plan.

The first function, "Draw\_floor\_plan", takes as input a floor plan image and a set of bounding box coordinates, which define the location and size of various objects within the floor plan. Using this information, the function draws the bounding boxes on the floor plan image, typically using a solid color or a dashed line to indicate the boundaries of the objects.

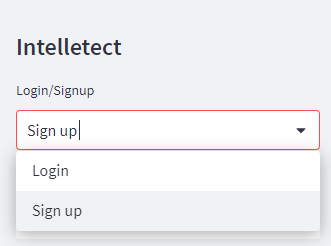
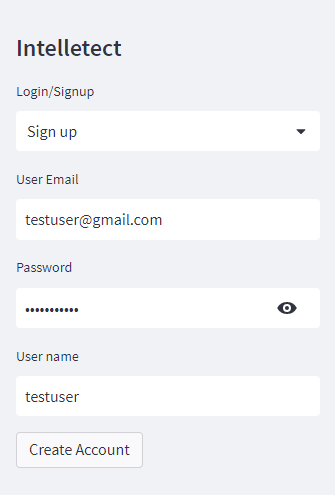
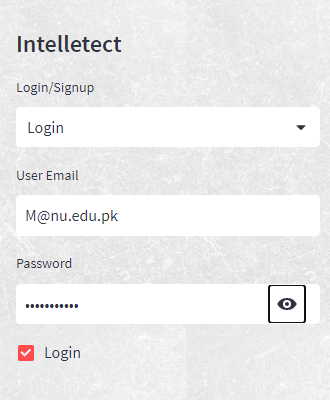
The second function, "save\_multiple\_bboxes", takes as input a set of images, each of which contains a floor plan with bounding boxes drawn on it. The function then saves each of these images as a separate file, typically in a specified directory or folder. This function is useful when working with large sets of floor plan images, as it allows multiple images to be processed and saved at once.



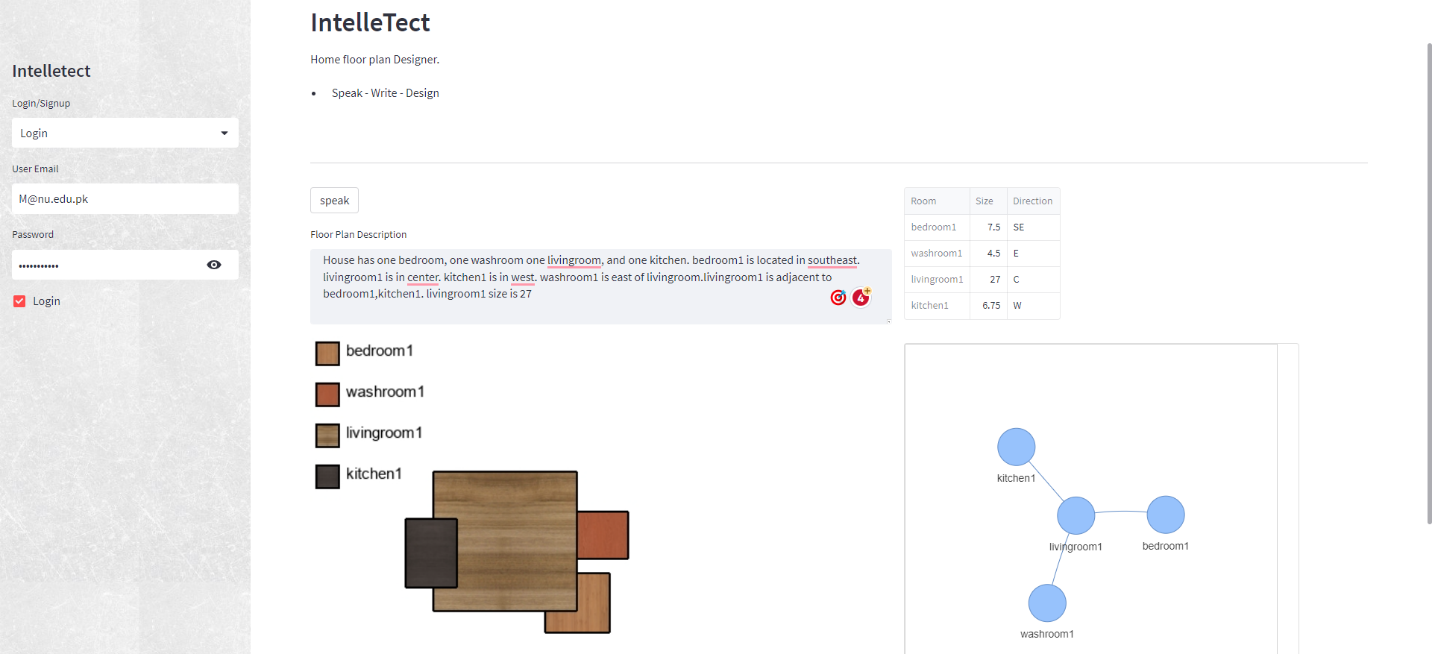
## Design and Development of a Website:

We designed a website in ‘StreamLit’ for our project. The user will enter voice /text input in the provided web interface and the provided website will generate a house plan for them according to their descriptions also storing them for later consultation.

#### Login and Signup pages.



#### The Working front End



## Iteration Plan:

## Iteration 1 (Sep 2022 – Oct 2022)

This iteration focuses on the initial processing of the input speech. The tasks for this iteration include.

1. Translating the speech to text.
2. Annotating the text to identify key elements.

## Iteration 2 (November 2022 – Dec 2022):

In this iteration, the focus is on editing and processing the text generated in Iteration 1. The tasks for this iteration include.

1. parsing and processing the text to extract key information.

## Iteration 3 (Dec 2022- Jan 2023):

The third iteration involves generating a layout based on the processed text from Iteration 2. The tasks for this iteration include.

1. Designing and generating an initial layout.

## Iteration 4 (Feb 2023 – March 2023):

In this iteration, the focus is on adding texture to the layout generated in Iteration 3. The tasks for this iteration include.

1. Designing and adding texture to the layout
2. Designing the final user interface.

## Iteration 5 (April 2023 – May 2023):

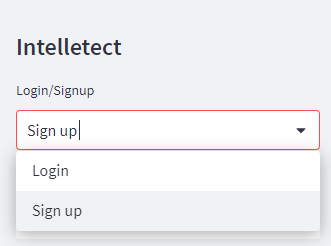
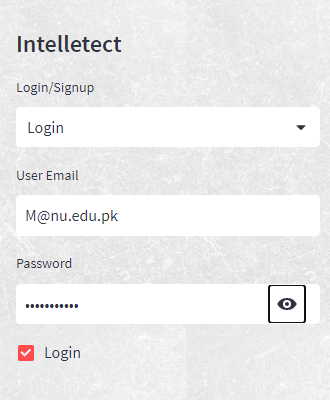
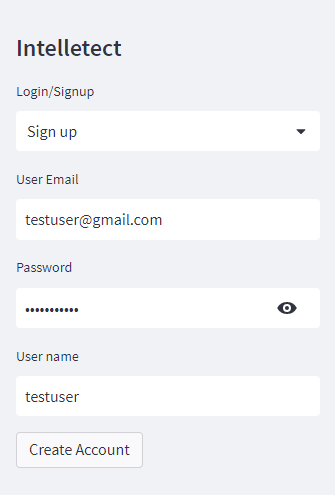
The final iteration focuses on testing and deployment of the system on a website. The tasks for this iteration include testing the system for functionality and usability, and deploying the system on a website for public use.

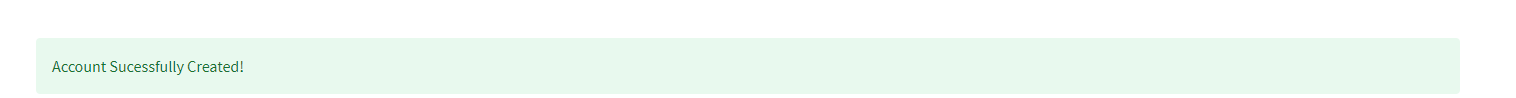
## User manual.

The user interface is kept very simple and user friendly, it is your basic website containing a login page where you can create an account and be a member of our community, after that. We have included the design your house page where the designs are made, an inventory option to help rediscover your previous designs and your saved/liked designs.

## Login or Signup.

* From the menu you can either select login or signup for your new user.
* You can login if you’re an existing user.
* Or you can sign up if you’re new.

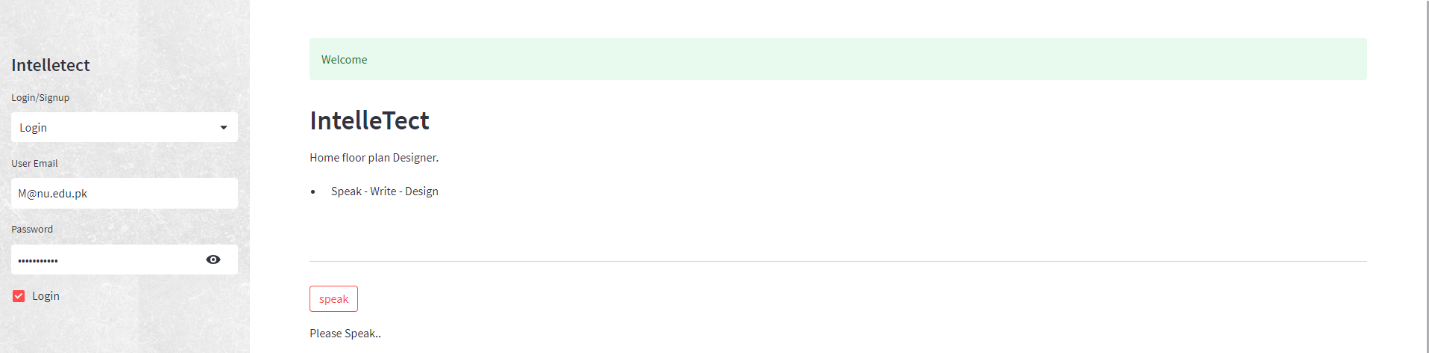




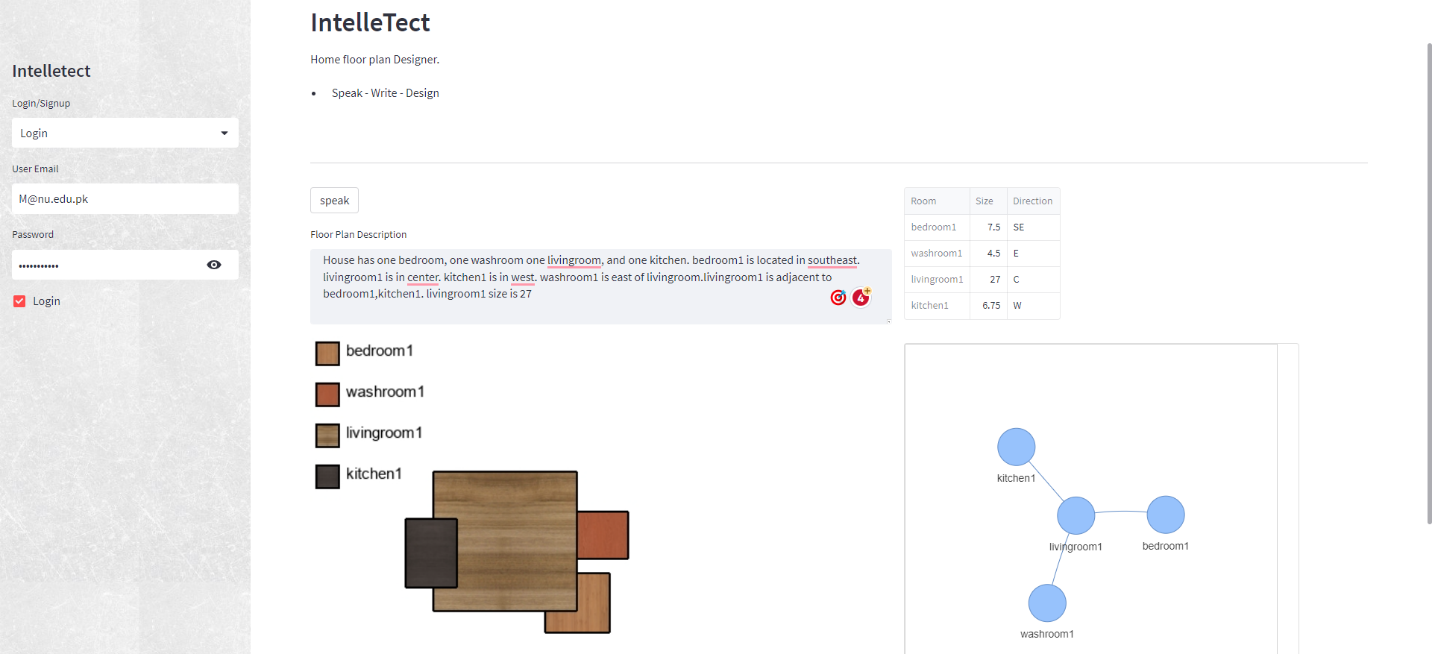
Account was created successfully

## User Commands.

The user can either speak in the microphone to put in his description of the house that he/she wants or write in the text box given below providing the different rooms, locations sizes and number of rooms desired in his house of choice, the system will take in his/her input and will generate an output according to the input provided as shown below.



Login Complete!



Room Legend

Positions & Sizes

Adjacency and Connections

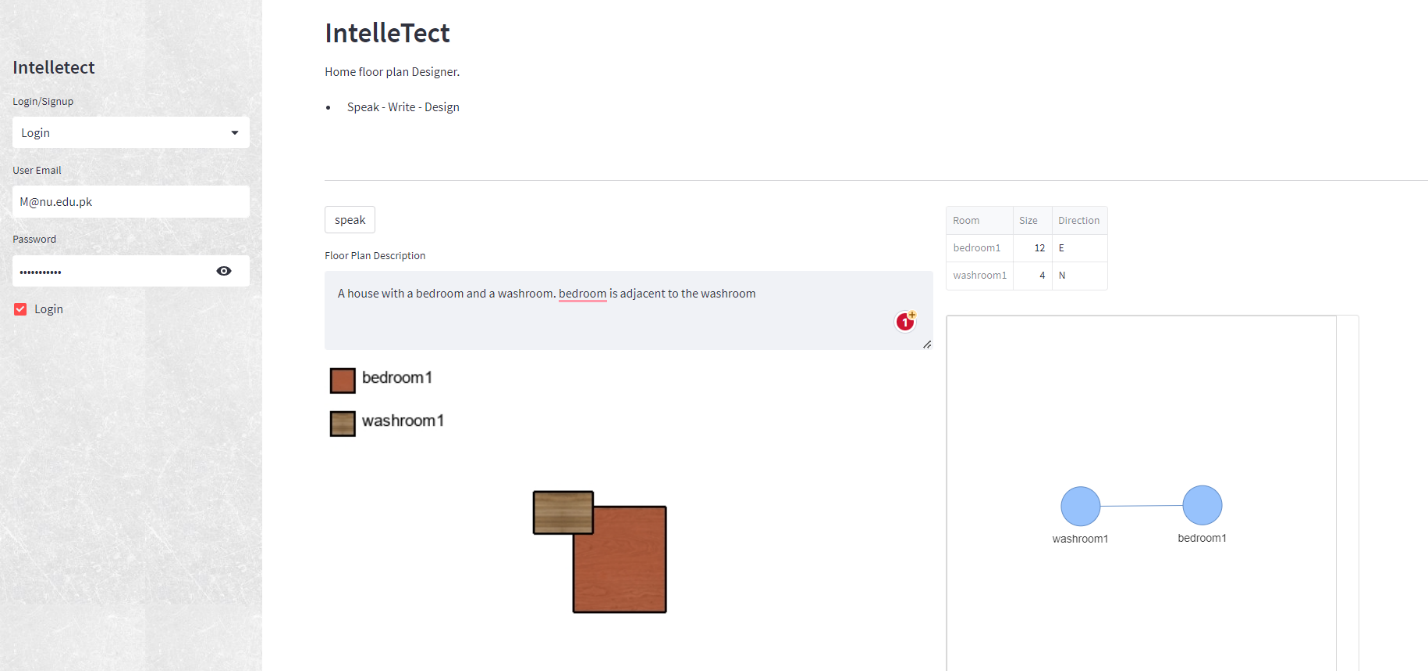
Voice Input!

TEXT Input!

## Test Cases.

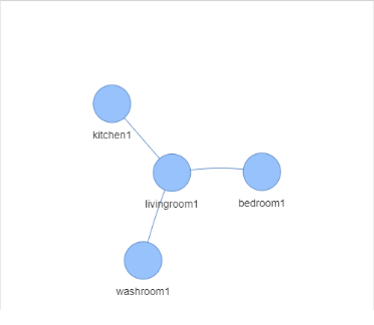
#### What if the input is small?

The system will still give put the desired output according to however small the input is.



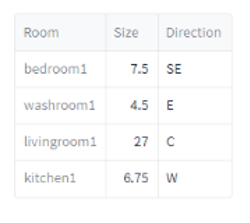
#### What about the connections of the room?

The system will give the connections of rooms in a form of visual graph to show linkage and adjacency, also providing the information for location.

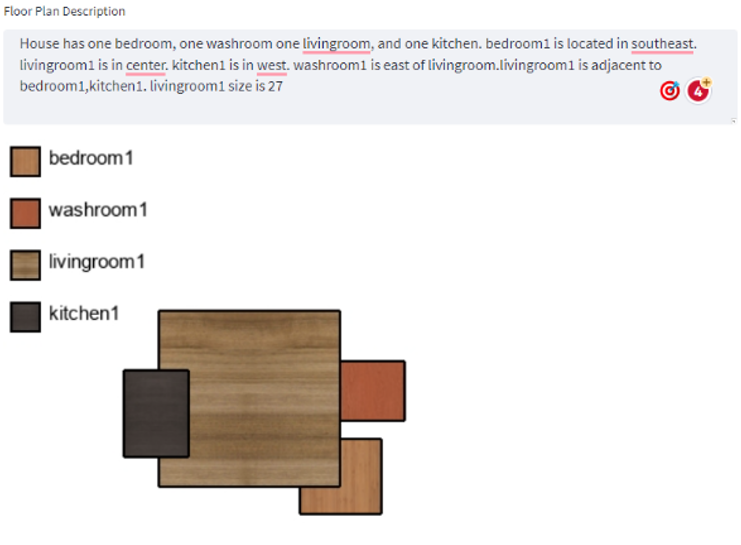


#### What about the locations of the rooms?

The system will give the locations and sizes of rooms in a form of a table providing the information for location and sizes.



#### Can you edit the input?

The input can be edited through a text bar that takes in the new input and provides you with the edited house plans.

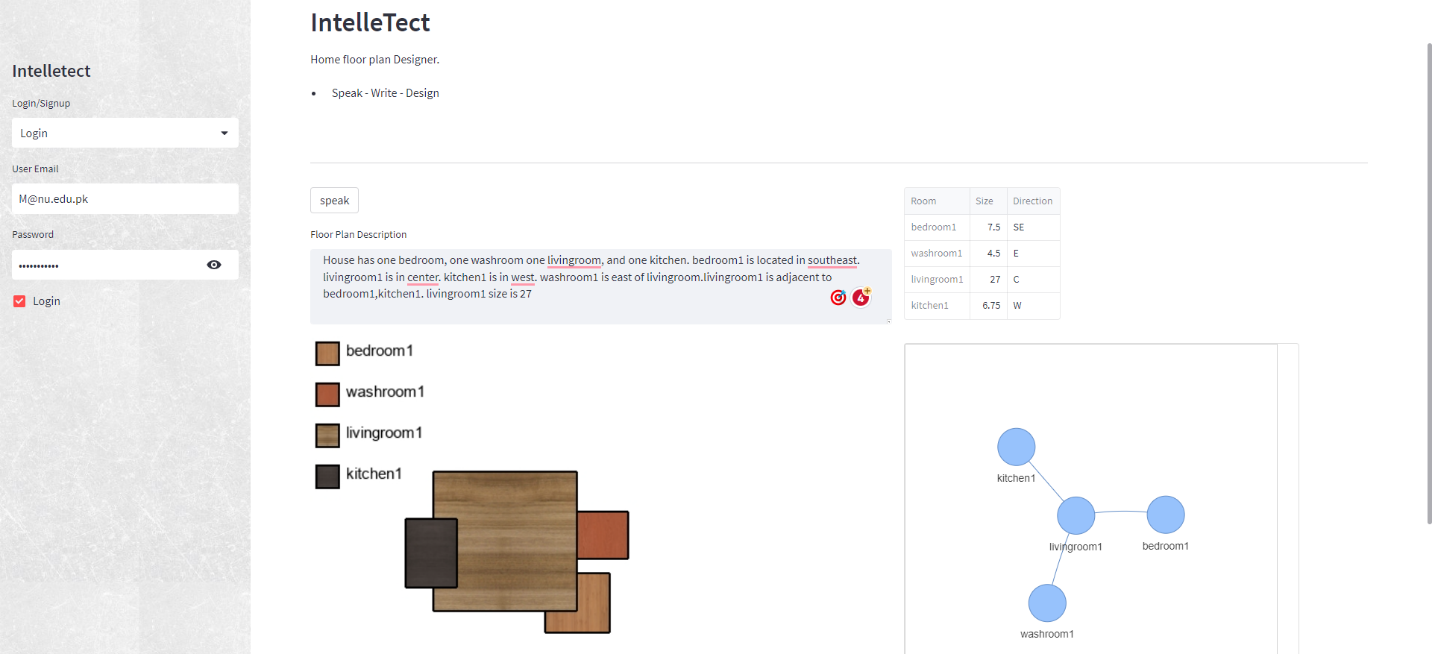
#### Can you tell which room is which?

The system gives you an automatic legend that describes what room has what sort of flooring and what texture is applied to what room, hence the user can differentiate the different rooms easily.



#### Does the output show correct positions and dimensions?

The system gives you an output that is the correct translation of the input provided, what room has what positions and is the size given in the description is matched by the output.



# Bibliography

Chen, Q., Wu, Q., Tang, R., Wang, Y., Wang, S., & Tan, M. (2020). Intelligent Home 3D: Automatic 3D-House Design from Linguistic Descriptions Only. 14.

Nauata, N., Chang, K.-H., Cheng, C.-Y., Mori, G., & Furukawa, Y. (2020, march 16). Networks for Graph-constrained House Layout. 17.

Nauata, N., Hosseini, S., Chang, K.-H., Chu, H., Cheng, C.-Y., & Furukawa, Y. (2020). House-GAN++: Generative Adversarial Layout Refinement Network towards Intelligent Computational Agent for Professional Architects. 10.