Smarchitect: Evolutionary approach to automated floor plan generation and visualization

Salar Abbas — **FAST NUCES**

Muddassir Lateef — **FAST NUCES**

Shayan Amir Zuberi — **FAST NUCES**

Abstract

Smarchitect is an innovative AI-driven tool designed to automate the process of generating floor plans for residential construction, offering a streamlined solution for architects and individuals lacking access to professional architectural services. By leveraging a genetic algorithm, *Smarchitect* enables users to create customized 2D floor plans based on specific inputs, which can then be converted into 3D models for enhanced visualization. The application addresses the limitations of traditional floor plan generation, reducing the time from several weeks to minutes and cutting down associated costs. This research paper explores the development process, architectural design and methodologies employed to create *Smarchitect*, highlighting its unique role in advancing AI applications within the architectural industry.

---

1. Introduction

Background and Motivation

As artificial intelligence (AI) continues to permeate diverse fields, the architectural industry has yet to fully adopt AI's potential to streamline processes and enhance productivity. Traditional methods of floor plan generation are typically manual, time-consuming, and costly, with architects often taking several weeks to produce preliminary plans. This approach poses challenges, particularly for individuals who may not have access to architectural firms due to financial constraints or geographical limitations. Recognizing the potential of AI to automate and democratize this aspect of architectural design, *Smarchitect* was conceived as a solution to expedite floor plan creation, reduce costs, and make architectural planning more accessible.

Problem Statement

The conventional process of floor plan generation is labor-intensive and time-consuming, requiring architects to spend significant time drafting and revising layouts based on client specifications. This process can be prohibitively costly for individuals without easy access to architectural firms. Furthermore, there is a lack of AI-based tools specifically tailored for floor plan generation that can provide users with quick, customizable results. *Smarchitect* aims to address these issues by developing a platform that leverages AI to generate accurate, user-specified 2D floor plans and convert them into 3D models within minutes, thereby filling a notable gap in the industry.

Objectives of *Smarchitect*

The primary objective of *Smarchitect* is to provide a fast, cost-effective, and user-friendly tool for generating customized residential floor plans. The system is designed to meet the following specific goals:

- Automatically generate 2D floor plans based on user-defined specifications, such as dimensions and room requirements.
- Enable easy modification and resizing of floor plans to suit users' changing needs.
- Convert 2D floor plans into 3D models to improve spatial visualization.
- Minimize floor plan generation time, setting a benchmark of under 10 seconds for both 2D and 3D outputs.

Target Audience

Smarchitect is designed to serve two key groups:

1. **Professional Architects**: The tool aims to support architects by providing a rapid solution for generating preliminary floor plans, allowing them to save time and effort in the initial design stages.
2. **Individuals without Access to Architectural Services**: For users who cannot afford or access professional services, *Smarchitect* provides an affordable and intuitive platform for creating custom floor plans without requiring specialized knowledge or experience.

Scope of the Study

This study examines the development, implementation, and testing of *Smarchitect*, exploring its technical architecture, functionality, and usability. The paper will delve into the genetic algorithm employed for 2D floor plan generation, the methodology behind 3D model conversion, and the iterative development process. Additionally, the study evaluates the application's performance through testing and user feedback, offering insights into its impact on the architectural industry and potential areas for future enhancement.

2. Literature Review

2.1 Current State of AI in Architecture

Artificial Intelligence (AI) has increasingly penetrated the architectural field, with applications spanning from design visualization to generative layout planning. However, the extent of AI's influence in architecture remains limited compared to its adoption in other industries. The architectural design process has traditionally relied on the expertise of architects to optimize space for functionality and aesthetics. Emerging AI-based approaches have introduced generative designs that leverage algorithms to automate some of the early-stage layout configurations. Despite this, current AI tools primarily support decision-making rather than independently automating complex tasks such as floor plan generation. The challenge remains in aligning AI with the nuanced requirements of architectural design, where subjective aesthetics and contextual relevance are paramount.

2.2 Existing Solutions for Floor Plan Generation

Floor plan generation today is largely manual, facilitated by CAD software like AutoCAD, SketchUp, and Revit, which offer only limited automation features. While some platforms, such as RoomSketcher and Planner 5D, allow users to create basic layouts, these tools remain constrained by their reliance on predefined templates and user adjustments, lacking genuine generative capabilities. The development of automated solutions that can generate floor plans on-demand based on specific user inputs is still in its infancy. Existing tools also often fail to incorporate the complex spatial relationships and dependencies that are central to architectural layouts, particularly in residential or custom housing.

2.3 Limitations of Traditional Methods

Traditional floor plan generation is often a labor-intensive and costly process. Architects typically spend weeks crafting layouts that balance spatial efficiency with design aesthetics. This process also requires significant expertise, limiting access to high-quality floor plans for individuals without professional resources. CAD software, while powerful, requires specialized knowledge to navigate and utilize effectively, making it less accessible to non-experts. Furthermore, manual layout creation is prone to errors and may struggle with replicating consistent design patterns.

2.4 Opportunities for AI Integration

AI introduces a transformative opportunity for automating the floor plan generation process through evolutionary approaches, which can optimize layout solutions iteratively. This study leverages the principles proposed by Martin Stacey, who explored the use of evolutionary algorithms to understand and replicate spatial qualities from existing architectural designs. By analyzing attributes like proportion, connectivity, and adjacency within a database of floor plans, an evolutionary approach can identify consistent traits and apply them to new designs. This method reduces reliance on explicit hardcoding and enables a flexible adaptation of layout elements based on learned patterns.

Such an approach allows AI to not only generate layouts quickly but also tailor them to specific user constraints, such as room dimensions and types. Additionally, integrating 2D-to-3D conversion capabilities provides users with immersive visualization options, enhancing the usability and accessibility of floor plan design tools. This AI-driven approach holds significant potential for making floor plan generation both efficient and accessible to a broader audience, including those without architectural training.

3. Project Overview

3.1 Project Vision

The project, *Smarchitect*, aims to leverage Artificial Intelligence to streamline the creation of residential floor plans. By automating the floor plan generation process, the application allows users to input specific constraints (e.g., room dimensions, number of rooms) and obtain multiple layout options within minutes. The vision is to make high-quality architectural planning accessible and cost-effective for both professionals and non-experts, particularly those without easy access to architectural services. *Smarchitect* also features a 3D visualization module, converting 2D layouts into 3D models for a more immersive design experience. By reducing the time and expertise required to create floor plans, *Smarchitect* aims to revolutionize the initial stages of residential construction planning.

3.2 Functional Requirements

The system's primary functional requirements include:

- **2D Floor Plan Generation**: Users can create 2D floor plans by specifying constraints such as the number of rooms, dimensions, and layout preferences.

- **Customizable Floor Plan Properties**: Users can adjust various properties of the generated floor plans, such as room size and placement, to meet specific spatial requirements.

- **Entity Addition**: Users can add new entities (e.g., furniture, walls, windows) to the 2D floor plans for enhanced customization.

- **2D to 3D Conversion**: The system allows users to convert the finalized 2D floor plans into a 3D model, providing an immersive visualization of the design.

3.3 Non-Functional Requirements

The non-functional requirements focus on performance, usability, and security:

- **Speed**: The system must generate 2D floor plans and convert them to 3D models within one minute, as this is a key selling point.
- **Consistency and Accuracy**: Floor plans must meet predefined design constraints and produce consistent, reliable outputs across multiple generations.
- **User Accessibility**: The system interface should be user-friendly, allowing non-experts to interact with the platform intuitively without prior architectural knowledge.

4 Implementation

4.1 Genetic Algorithm for Floor Plan Generation

Genetic algorithms (GAs) are widely used for solving complex design optimization problems, such as generating floor plans that balance multiple constraints. In *Smarchitect*, the GA is employed to generate efficient and customized floor plans by evolving an initial set of layouts. Each potential layout is represented as a "chromosome," or a configuration of rooms and spaces. Through iterative processes of selection, crossover, and mutation, the GA continuously refines these layouts based on a defined fitness function that assesses how well each configuration meets the design criteria.

4.1.1 Factoradic System for Permutation Encoding

A key challenge in the GA is to explore a vast space of possible layouts in an efficient and systematic way. To achieve this, *Smarchitect* leverages the factoradic (or factorial number system) for generating the n-th permutation of room arrangements. This method, commonly used in combinatorial generation and permutation algorithms, allows the GA to directly compute specific permutations rather than sequentially generating all prior permutations.

This approach is rooted in the work of Donald E. Knuth, particularly in *The Art of Computer Programming* (Volume 4: Combinatorial Algorithms), where Knuth discusses the application of the factoradic system for indexing permutations. Using the factoradic system, a specific arrangement of rooms can be encoded as a permutation index, making it possible to generate unique floor plan configurations efficiently within the GA's mutation and crossover processes. This systematic permutation generation enhances the diversity of layouts explored by the algorithm, ensuring that each iteration introduces new and varied designs for evaluation.

4.1.2 Fitness Function of the Genetic Algorithm

The fitness function in *Smarchitect* is a crucial component that evaluates each generated layout based on several design criteria:

- Room Size and Proportion: The function checks each room against minimum area and dimension requirements. Room types (e.g., bedroom, bathroom) must adhere to size constraints, and penalties are applied if rooms fall outside these limits.
- Adjacency Requirements: Logical adjacency constraints, such as placing a kitchen near a dining area, are enforced through penalties in the fitness score for plans that do not meet these adjacency requirements.
- Total Area and Space Utilization: The function evaluates the layout's footprint relative to target dimensions, rewarding efficient space utilization and penalizing excessive or inadequate usage.
- Proportionality and Aspect Ratios: Each room's aspect ratio is checked against acceptable bounds to ensure balanced and functional shapes. Rooms with awkward proportions are penalized, encouraging layouts with well-shaped rooms.

The overall fitness score is a weighted sum of these factors, and designs with lower penalties (or higher rewards) are selected for reproduction. By leveraging the factoradic system, the GA efficiently generates diverse permutations and converges on optimized floor plan solutions, balancing user constraints with architectural standards.

4.2 Conversion of 2D Floor Plans to 3D Models

In addition to 2D layout generation, *Smarchitect* provides a 3D visualization module that enables users to better understand the spatial flow and aesthetics of their floor plans. This module utilizes a 3D rendering engine, such as Unity, which interprets the data from 2D layouts to build a 3D environment. The conversion process involves defining the 3D properties of walls, windows, and other architectural elements based on the 2D layout structure. By translating flat designs into three-dimensional spaces, this feature enhances user engagement and provides an immersive visualization experience that aids in layout decision-making.

5. Results and Evaluation

5.1 Performance of Floor Plan Generation

The floor plan generation module in *Smarchitect* was evaluated for speed, accuracy, and alignment with user-defined constraints. Across 200 test cases, the module consistently met the target generation time of **10 seconds** per layout. However, due to current limitations, the module achieved **70–80% accuracy** in satisfying specific room dimension and adjacency requirements. This accuracy is partly due to the system's focus on single-floor layouts, which simplifies generation but may restrict adaptability for multi-level or more complex designs.

While *Smarchitect* may not be ideal for highly customized or intricate layouts, it significantly reduces the time required for initial drafts, providing users with a quick starting point. The generated layouts offer a functional level of organization, particularly for single-floor houses, making it suitable for users aiming to design their own homes without in-depth architectural knowledge.

5.2 Efficiency of 2D to 3D Conversion

The 2D to 3D conversion module was assessed for rendering speed and quality, with conversion times averaging **8 seconds** on high-performance systems and **10 seconds** on standard devices, meeting the targeted performance frame.

User feedback on the quality of 3D models was positive, with spatial accuracy and detail rated **4.7 out of 5**. The conversion process accurately preserved the layout from 2D to 3D, with clear room dimensions and placements. Users noted that the 3D view enhanced their understanding of room relationships and spatial flow, offering a practical preview for those less familiar with architectural plans.

5.3 User Feedback and Usability Evaluation

User feedback was gathered from a group of 50 participants, including architects, designers, and non-professional users. The evaluation focused on usability, control clarity, and overall satisfaction with the generated outputs.

Participants rated the floor plan generation module **4 out of 5** for usability. Both professional and non-professional users appreciated the fast generation of initial layouts, particularly those designing their homes independently. The generated plans generally met user expectations, providing practical layouts with minimal setup. Non-professional users, in particular, valued the straightforward interface and found the tool effective for creating basic, functional floor plans without requiring extensive design knowledge.

The 2D to 3D conversion feature also received favorable feedback, with a **4.6 out of 5** usability rating. Users found the interactive 3D visualization useful for exploring and understanding spatial arrangements. Suggestions for future improvements included additional customization options for finishes, lighting, and furniture in the 3D view, especially for users interested in more personalized visualizations. Overall, *Smarchitect* was praised for enabling users to achieve a workable initial layout quickly, reducing the need for time-consuming manual drafting.

6 Discussion

6.1 Challenges and Limitations

The development and implementation of *Smarchitect* presented several challenges, particularly in achieving a high degree of accuracy in meeting room constraints within the targeted 10-second generation frame. Currently, the algorithm's accuracy in satisfying specific room dimensions and adjacency requirements is approximately **70–80%**. This limitation stems from the focus on single-floor layouts, which restricts the tool’s ability to handle multi-level designs or layouts requiring intricate spatial arrangements. Additionally, while the genetic algorithm enables rapid generation of viable layouts, it may struggle with highly specific or conflicting user constraints, where flexibility is limited by the genetic algorithm’s inherent prioritization of simpler solutions over complex optimizations.

6.2 Comparison with Traditional Methods

Compared to traditional manual drafting or CAD-based design approaches, *Smarchitect* offers considerable advantages in speed and accessibility. Traditional floor plan creation often requires architectural expertise and significant time investment, particularly in the early drafting stages. In contrast, *Smarchitect* allows users to generate a basic, functional layout within seconds, making it a valuable tool for those without design experience. However, while the application provides a solid starting point, it lacks the precision and customizability that a skilled architect or designer can achieve. Unlike traditional methods, which allow for in-depth customization and iterative refinements, *Smarchitect* focuses on producing quick drafts, making it more suitable for preliminary layouts or single-floor residential projects rather than complex or multi-functional spaces.

6.3 Potential for Future Development

Smarchitect has strong potential for further development, particularly in improving accuracy and expanding functionality to handle multi-level and more intricate designs. Enhancing the genetic algorithm to better accommodate complex constraints could increase the accuracy in meeting user specifications, especially for users with specific spatial requirements. Additional features, such as multi-floor support and customizable room shapes, would expand the tool’s applicability beyond basic residential layouts. Moreover, incorporating a learning component to adapt the algorithm based on user feedback could make the tool more intuitive and responsive over time, ultimately enhancing its utility for a broader audience.

7. Conclusion

Smarchitect was developed to address the need for a quick, accessible tool to generate floor plans, providing an alternative to traditional architectural design methods for single-floor residential layouts. Testing indicated that the tool achieves its primary objective by generating functional floor plans within a **10-second** timeframe, with **70–80% accuracy** in meeting user-defined room constraints. User feedback confirmed that the tool effectively reduces initial drafting time, making it particularly beneficial for users without architectural expertise.

7.1 Contributions to the Field

The development of *Smarchitect* contributes to the architectural and design fields by providing a prototype for rapid, AI-driven floor plan generation. It showcases the potential of genetic algorithms in space planning, demonstrating how AI can assist in preliminary layout creation by balancing speed with moderate accuracy. The tool's approach democratizes access to architectural design, offering non-professional users an easy entry point into the planning process, which could inspire further applications of AI in residential design.

7.2 Future Work and Recommendations

To enhance *Smarchitect*'s functionality, future work should focus on improving algorithm accuracy and expanding the tool's versatility to support multi-floor layouts and more complex spatial configurations. Integrating features for finer control over room dimensions and adjacencies would allow users to create more customized layouts. Additionally, incorporating an adaptive learning system based on user feedback could refine the tool's responses over time, making it increasingly accurate and user-friendly. Finally, exploring collaborations with CAD software for seamless integration could provide users with more advanced editing options, bridging the gap between rapid drafting and professional-grade design refinement. In addition, if the dataset is available DL based approaches show promise as well.

Appendix

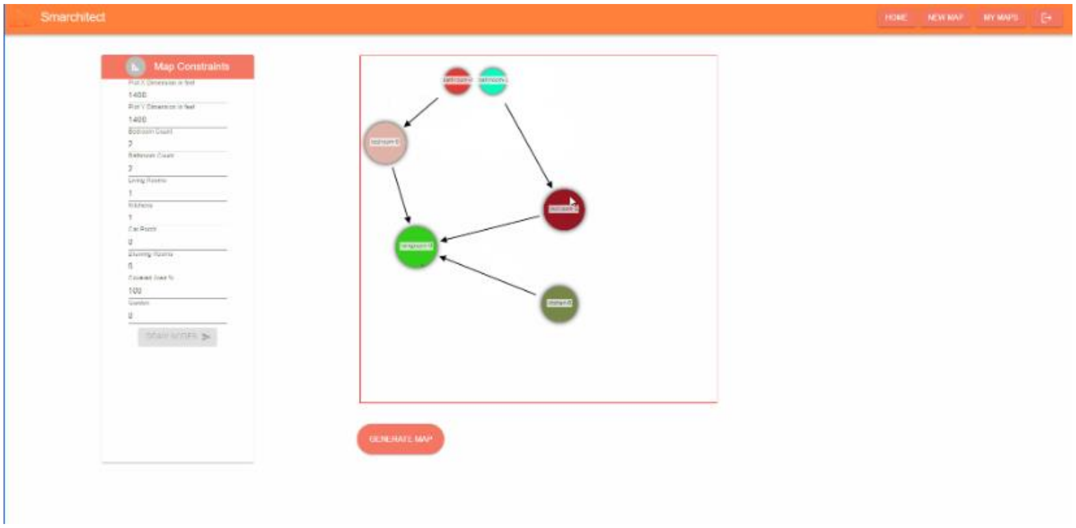
User Interface

Floor Plan Generation Screen

The Floor Plan Generation screen serves as the starting interface where users specify constraints for generating a customized 2D floor plan. Here, users enter basic layout requirements, including the number of rooms, room types (e.g., bedrooms, bathrooms, kitchen), and approximate dimensions.

Once these inputs are submitted, the system generates a graph-based visualization representing the specified rooms as nodes. Each node corresponds to a room and can be connected to other nodes to define desired room adjacencies. This interactive adjacency tool allows users to directly map relationships between rooms, specifying, for instance, that a bedroom should be adjacent to a bathroom or that the living room should be centrally connected to multiple rooms.

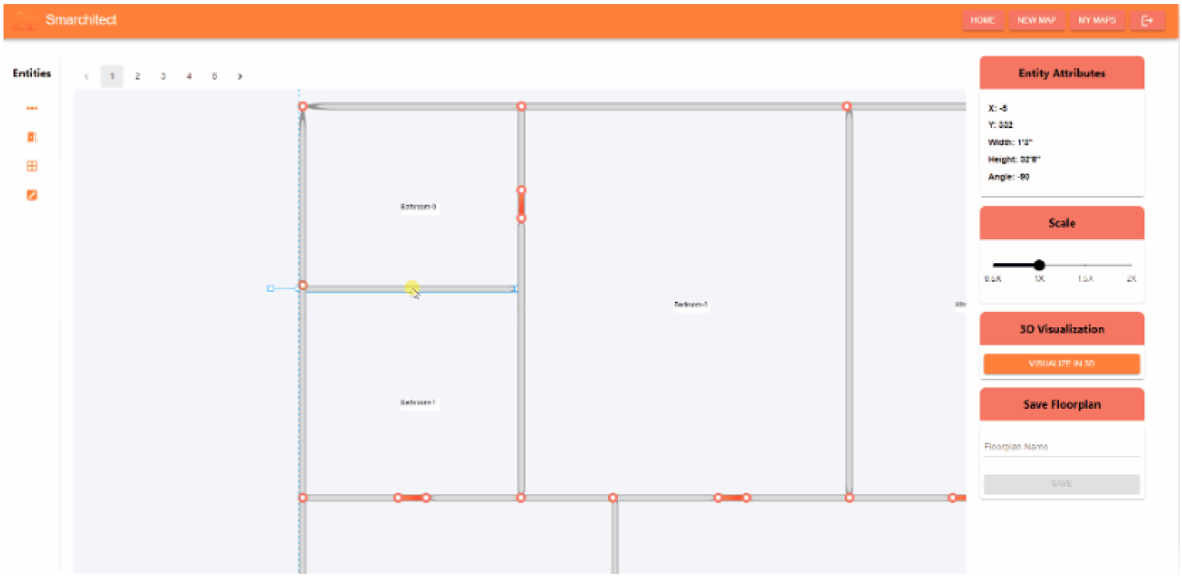
The system uses these adjacency connections as constraints in the evolutionary algorithm, ensuring that generated layouts honor the user-defined spatial relationships. After the adjacency graph is finalized, the system displays three optimized 2D floor plans that meet the user's specifications, including room connections, sizes, and proximities.



This interactive approach provides users with greater control over the spatial relationships in their layout, creating a more customized and functional design.

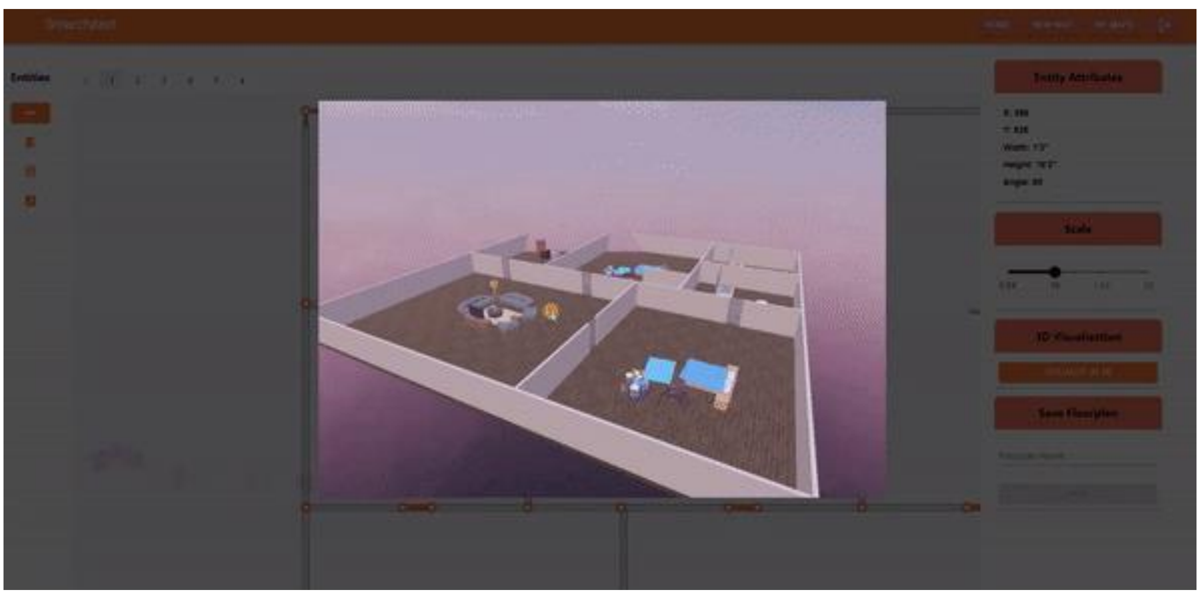
2D Floor Plan Tweaking Screen

The 2D Floor Plan Tweaking screen provides an interactive interface for users to adjust and customize the generated floor plans. Users can drag and drop entities (such as furniture and walls), resize rooms, and modify layout elements directly. The screen also provides visual cues and validation to ensure that adjustments remain within the defined constraints, offering a seamless and intuitive customization experience.



3D Generation Screen

The 3D Generation screen allows users to view a converted 3D model of their floor plan. This screen provides various viewing angles, zoom capabilities, and a real-time walkthrough feature, enabling users to explore their designs from different perspectives. The 3D Generation screen enhances user experience by offering an immersive visualization of the layout, helping users make more informed decisions about their designs.

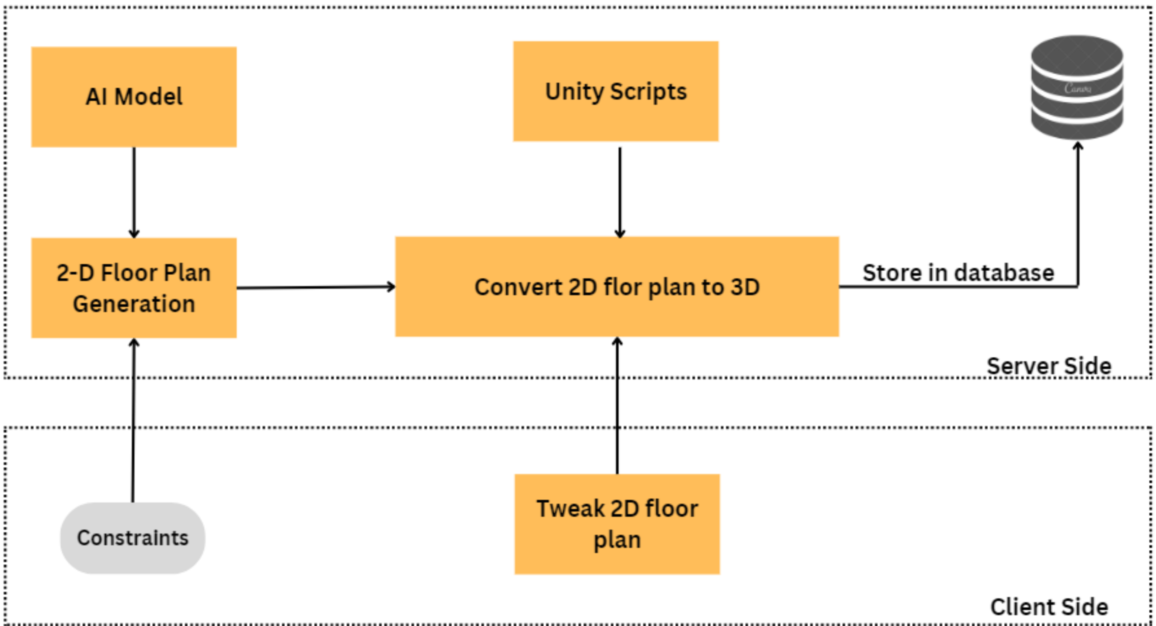


Technology Stack

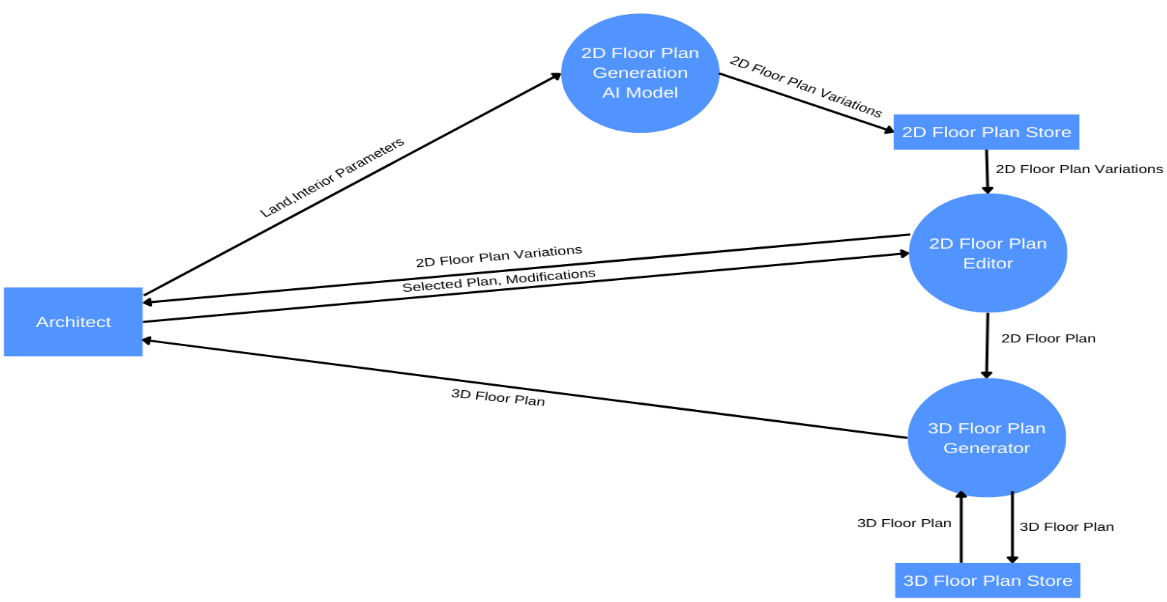
The Smarchitect application integrates a **React (JavaScript) frontend** with a **Django (Python) backend** for responsive UI and real-time AI-driven layout generation, while **Unity3D (WebGL)** enables interactive 3D floor plan visualization. A **custom evolutionary algorithm**, supported by **NumPy, Pandas, and Scikit-Learn**, optimizes floor plans based on user-defined spatial constraints.

Full code for the whole project is available on github at: https://github.com/Salar24/Smarchitect

Architectural Diagram



Data Flow Diagram (DFD)



Pseudocode:

1. Initialize Parameters
   - Set constants for rewards and thresholds: DOOR_LENGTH, ADJACENCY_REWARD, PERCENTAGE_REWARD, PROPORTION_REWARD.
   - Define room names, desired proportions, and area percentages.
   - Define width and height of the layout area.

2. Generate All Possible Full Binary Trees
   - Input: Number of rooms (Nr)
   - Output: List of all possible binary trees for structuring room layouts
   - Formula: Total nodes in a full binary tree = 2 * Nr - 1
   - Call `allPossibleFBT(2 * Nr - 1)` to generate possible tree structures.

3. Define Genetic Algorithm (GA) Parameters
   - Set `population_size`, `number_of_generations`, and `mutation_rate`.
   - Initialize population using `gen_population(population_size, Nr, total_trees)`.
   - Each chromosome represents a possible layout configuration:
     - `chromosome[0]`: Tree structure index
     - `chromosome[1]`: Factoradic index for room permutation
     - Remaining genes: Splitting information for tree nodes (axis and ratio)

4. Run Genetic Algorithm for Specified Number of Generations
   - For each generation:
     a. Crossover
     b. Mutation
     c. Evaluate Fitness
        - For each chromosome in the population:
          - Generate a tree layout from the chromosome by factoradic method to get nth permutation.
          - Convert the tree structure to room coordinates.
          - Calculate the chromosome’s fitness with:
            - Adjacency Check: Ensure specified rooms are adjacent.
            - Percentage Area Match: Compare room areas to target percentages.
            - Proportion Match: Check room proportions to desired ratios.
          - Assign fitness score based on adherence to constraints.
     d. Selection
        - Sort population by fitness score.
        - Retain the top chromosomes to form the next generation.

5. Convert Top Chromosome to Room Layout
   - Select the best chromosome from the final generation.
   - Generate a room layout tree.
   - Generate room coordinates from the tree structure.
   - Convert room coordinates to boundary lines.

6. Normalize and Add Additional Elements
   - Normalize boundaries to avoid overlapping walls `.
   - Insert doors between adjacent rooms based on adjacency requirements.

7. Output Final Layout
   - Format the final layout as JSON for visualization.
   - Include room boundaries, door placements, and room centers in the output.

8. End.

References

1. **Knuth, Donald E.** "The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1." Addison-Wesley Professional, 2011.

   - This book explores various combinatorial algorithms, including the use of factoradic (factorial-based) numbering systems for permutation indexing, a key technique used in this project for generating unique room arrangements.

2. **Michalewicz, Zbigniew.** "Genetic Algorithms + Data Structures = Evolution Programs." Springer Science & Business Media, 1996.

   - This text provides foundational concepts in genetic algorithms, explaining operations like crossover, mutation, and selection, which form the basis for the layout generation algorithm used in *Smarchitect*.

3. **Stacey, Martin.** "Automatized Space Layout Planning from Preceding Typologies." Master’s Thesis, Bartlett School of Graduate Studies, University College London, 2017.

   - Stacey’s work introduces an algorithmic approach to generating floor plans based on traits extracted from existing designs. His research on automating space layout planning through genetic algorithms and knowledge transfer informed key aspects of the *Smarchitect* project’s methodology, particularly in encoding spatial qualities and adjacency rules derived from architectural typologies.