ISRA UNIVERSITY HYDERABAD

DEPARTMENT OF COMPUTER SCIENCE

FINAL YEAR PROJECT THESIS

By

Taimoor Ameer **2101-BSCS015**

Sajan **2101-BSCS016**

Ishwar Lal **2101-BSCS013**

Supervised by **Co-Supervised by**

Dr. Mutee U Rahman **Dr. Sehrish Abrejo**

Final year project report submitted in the partial fulfilment of the degree of Bachelors of Science in Computer Science

December 2024

FACULTY OF ENGINEERING, SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

Certificate

This is to Certify that **Taimoor, Sajan and Ishwar Lal** have worked under my supervision for the BS(CS) project titled **V-ARCHITECT**, as per partial requirement for the award of the degree of BS(CS) and the work is original and satisfactory.

Dr. Mutee-u-Rahman **Supervisor** **Department of Computer Science**	**Dr. Amber Baig** **Chairperson** **Department of Computer Science**

ACKNOWLEDGEMENT

We would like to thank our supervisor, **Dr. Mutee-u-Rahman**, and our co-supervisor, **Dr. Sehrish Abrejo,** for supporting and guiding us relentlessly in developing the project. The valuable feedback and constructive suggestions paved a way forward for the work. His persistent availability during all critical phases of the project is really appreciated. Without his expert advice and leadership, this project would not have reached the level that it has today.

We're also highly thankful to the faculty members of the Department of Computer Science at **Isra University, Hyderabad,** for creating and preserving an environment that was intellectually stimulating as well as offering opportunities throughout our academic journey.

In addition, the teamwork of each project member is appreciable. Really, it was a group effort, and what each of us did really contribute towards getting the project through on our end. Working with everyone definitely helped eliminate technical and conceptual problems hindering the project, and I'm really thankful for the teamwork and cooperation that each of us demonstrated.

We thank **Isra University, Hyderabad,** for all the facilities and academic aid it provided us in carrying out this project. We wouldn't have been able to produce our ideas or complete this task without such infrastructure and academic support from the university. It has been a tough journey, yet we are grateful to have achieved my goals with all the aid and support received.

ABSTRACT

This project integrates voice commands to simplify the process of designing a 2D residential floor plan. It is easy, accessible, and fun for the users. Most existing floor plan software, such as AutoCAD, requires technical knowledge and can only be used by users with such skills. It automates and simplifies the design process through voice commands.

Basically, the system is designed especially for disabled people who rely on voice-assisted technology. This project by introducing voice input for creating the plans itself promotes the idea of equal contribution where designing can be made possible for the physically challenged peoples.

It is an application that uses Windows Voice Access for hands-free interaction and requires basic knowledge of its functionality. It makes use of a speech recognition mechanism that interprets voice inputs as actions on a 2D canvas. Using voice commands, a user can easily "draw a room," "add a door," or "resize a room." This application was developed using Flutter, using the CustomPaint API to render graphics and speech_to_text plugin for speech recognition.

Evaluation of the system shows that it minimizes manual effort significantly and presents an easily accessible user interface. However, the system still does not recognize complex or vague commands, which are in the development phase.

This work concludes that voice-command-driven interfaces hold great promise for optimizing the design process, allowing non-technical users to achieve what was previously reserved for professionals. Future work will be concerned with improving the command recognition system so more complex designs can be represented and recognition accuracy can be enhanced over varied environments.

Table of Contents

List of Figures

CHAPTER #01

INTRODUCTION

1.1 Background:

Software lies at the heart of architecture and interior design industries as its utilization is unavoidable in drafting floor plans and layouts. However, conventional AutoCAD and SketchUp, although powerful tools, require expertise and training. It is high time that technology innovates toward accessible design tools which may ease complicated procedures for the design community. Being highly integrated into different fields, the project utilizes voice recognition towards developing an even more intuitive 2D floor plan design application. The project aims at transforming how a non-expert user might interact with architectural design through hands-free interaction techniques.

1.2 Problem Statement:

Designing a floor plan usually needs lots of technical knowledge and complex understanding of how particular software applications work. This limits access to users who probably don't have the training or time to learn specialized tools. There is thus a need for a simple tool that would allow users to create, modify, and save 2D floor plans using a natural language command for the general population to enjoy access to their design tasks.

1.3 Objectives:

- To develop a 2D, Residential Floor Plan designing application that will allow users to create and change the layout of their Floor Plan designs using voice (primary) or text-based commands.
- To make intuitive commands to help make the designing process easy and understandable.
- To enable accessibility by using Semantics, so that the application can be used hands-free.

1.4 Scope:

This application is meant to make simple 2D residential Floor Plans using voice/text commands. The commands will be interpreted to basic design actions such as drawing, resizing, and arranging of rooms, doors, windows, stairs, and cutouts. This scope does not include anything more advanced or sophisticated in terms of architectural design that could include such things as 3D modeling, structural analysis.

1.5 Significance:

The application enables one to manage their floor plan based on voice commands thus democratizing access to architectural tools and equipping the users, without any technical knowledge, with the possibility of creating layouts quickly and accurately. From this project onwards, further chances in accessible design software can be seen as well as the capability of voice recognition in improving complex tasks.

1.6 Survey:

The purpose of conducting survey was to determine the basic user needs for achieving a proper design of a 2D floor plan, which is the primary goal of this application. The questions were made, especially, to ensure that the application would have certain features needed to design a floor plan.

The survey was conducted online, and was made through Google Forms. It consists of 10 easy questions, which are listed, along with their results, below:

Question No:1: What types of floor plan or design features would you find most useful in a 2D design application?

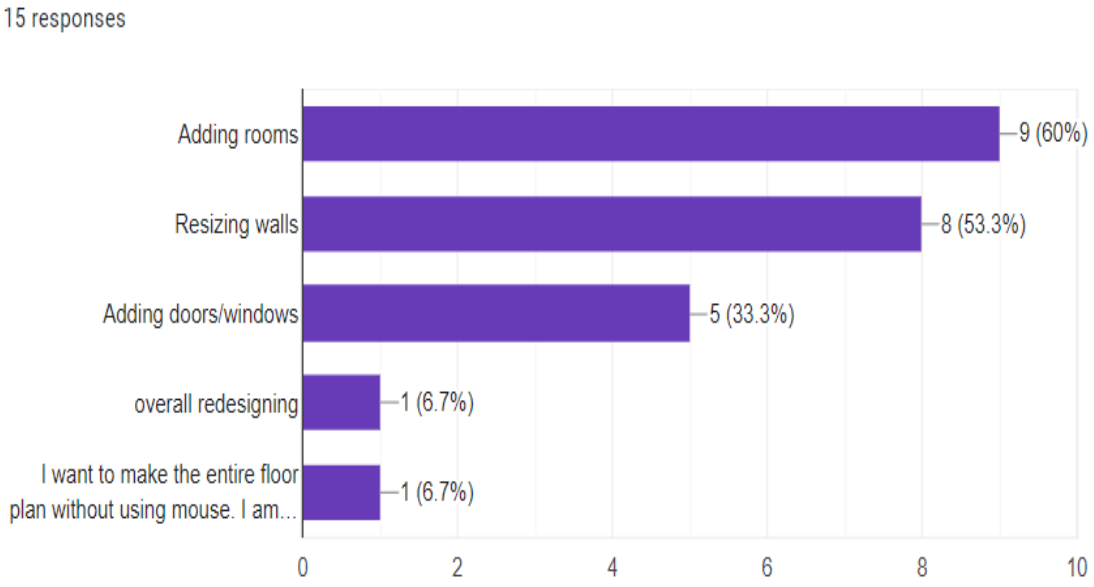

Figure 1. 1: Survey Question 1 Result (Bar Chart)

Most of the participants placed the highest importance on Adding Rooms and Wall Resizing, which means these features are designed as necessities for the users. Adding Doors/Windows is appreciated only by a few respondents, while In General Redesigning, as well as designing the whole floor plan just by using voice, is not considered to be a high priority.

These insights suggest that more emphasis on addition of rooms and resizing of walls will satisfy the core expectations of the users, while including the speech command options will only be for an add on feature for convenience.

Question No:2: How comfortable are you with using voice commands to interact with software?

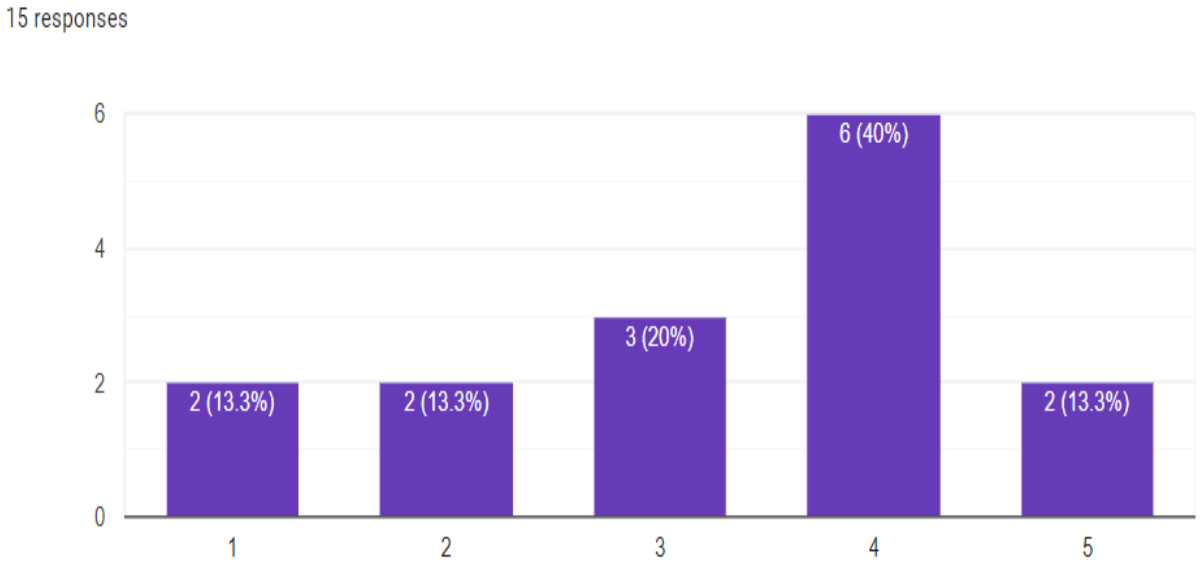

Figure 1. 2: Survey Question 2 Result (Bar Chart)

Ease. Forty percent of the respondents are totally comfortable using voice commands as comfort level 5. However, the distribution of responses from other levels of comfort at 3 (20%) and level 4 (13.3%) shows that there are some with moderate comfort level. More significant portion, 13.3% is level 2, not so comfortable, while no one responds with comfort level 1.

This result tends to indicate a positive disposition to voice command usage; hence integration of voice commands could probably be well received, though certain users may require further support or guidance in this regard.

Question No:3: How important is it for the application to be easy to use for someone with no design experience?

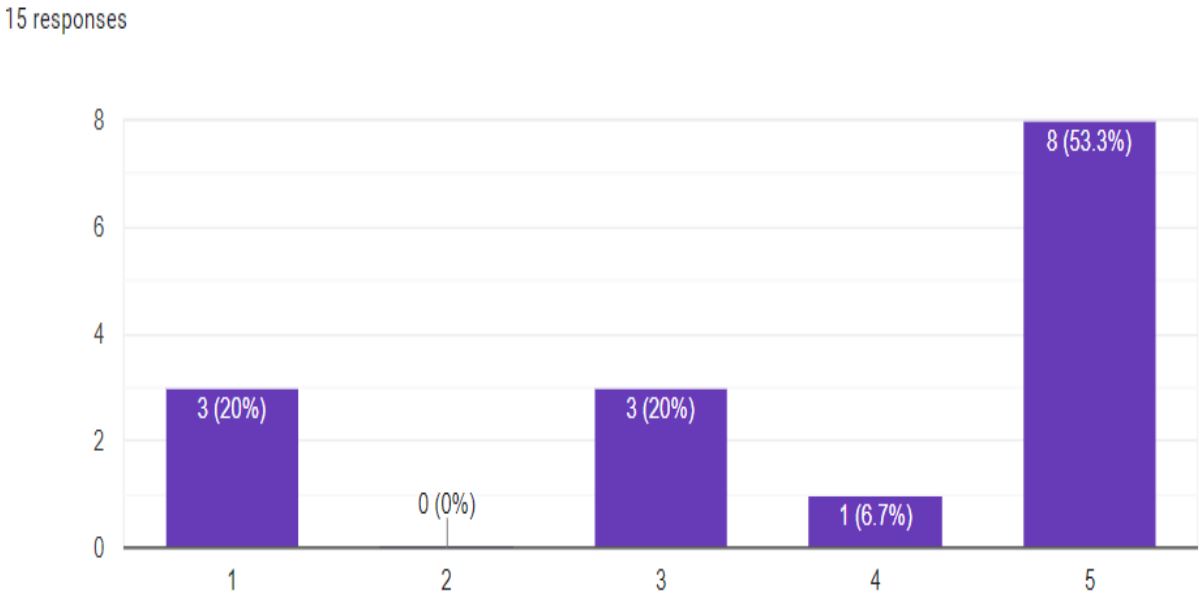

Figure 1. 3: Survey Question 3 Result (Bar Chart)

Most of the respondents have rated 53.3% ease of use for non-designers as very important at level 5 while another 20% rated it as important, which means level 3. The moderate level of importance with level 1 and level 3 was chosen by 20% of (each level) the respondents, and a small fraction of 6.7% rated it as slightly important at level 4. No respondent rated it at the lowest level, which is level 2.

The application should be easy to use for purposes of application targeting non-design users.

Question No:4: What design element would you find essential in a 2D floor plan application?

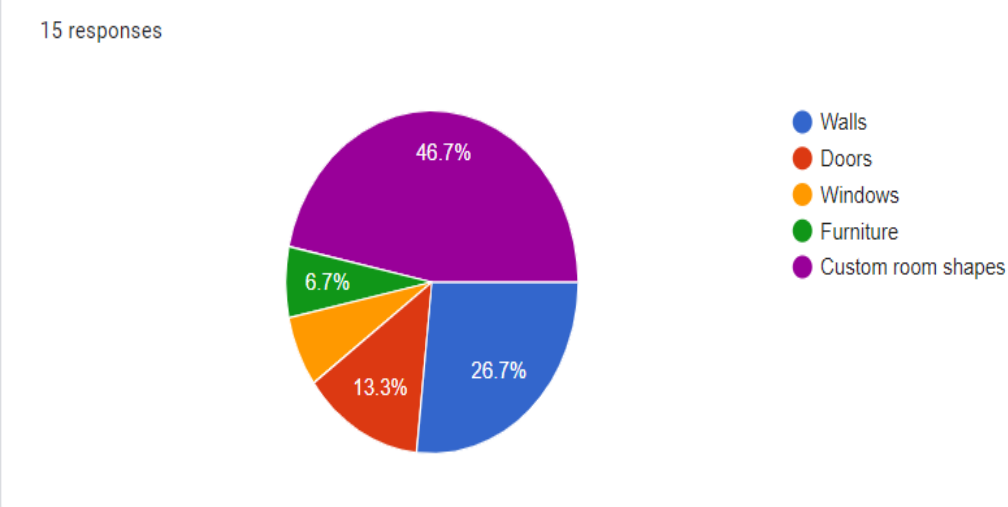

Figure 1. 4: Survey Question 4 Result (Pie Chart)

Close to half of the respondents (46.7%) consider the Custom Room Shapes as the most important designing feature. Along with walls (26.7%), doors (13.3%), and windows (6.7%), furniture also holds the same weight of 6.7%.

The findings explain that the most interesting aspects are altering the structure of the room and creating walls. Therefore, emphasis could be on offering features of modifying the layout of the house with less concern on the position of doors and windows.

Question No:5: For a person with a disability, how important are features such as voice commands, screen readers, or customizable interfaces in ensuring the accessibility of the 2D floor plan design application?

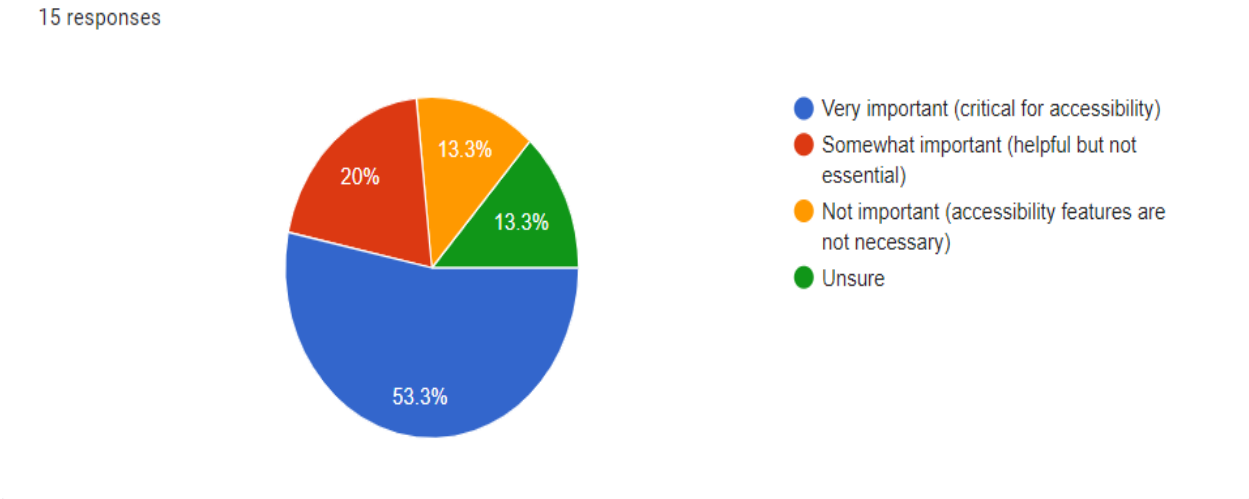

Figure 1. 5: Survey Question 5 Result (Pie Chart)

Over half of the respondents (53.3%) believe that these accessibility features are very much needed in the application emphasizing that they are very important for the people with disabilities. A lesser percentage of people (20%) think they are a bit important, while only a few (13.3%) consider them to be unnecessary or are indifferent about their necessity.

This outcome implies that giving emphasis to the application's accessibility features will be very crucial in the development and the acceptance of the application particularly among the people with disabilities.

Question No:6: Do you foresee any challenges in using voice commands to create a floor plan?

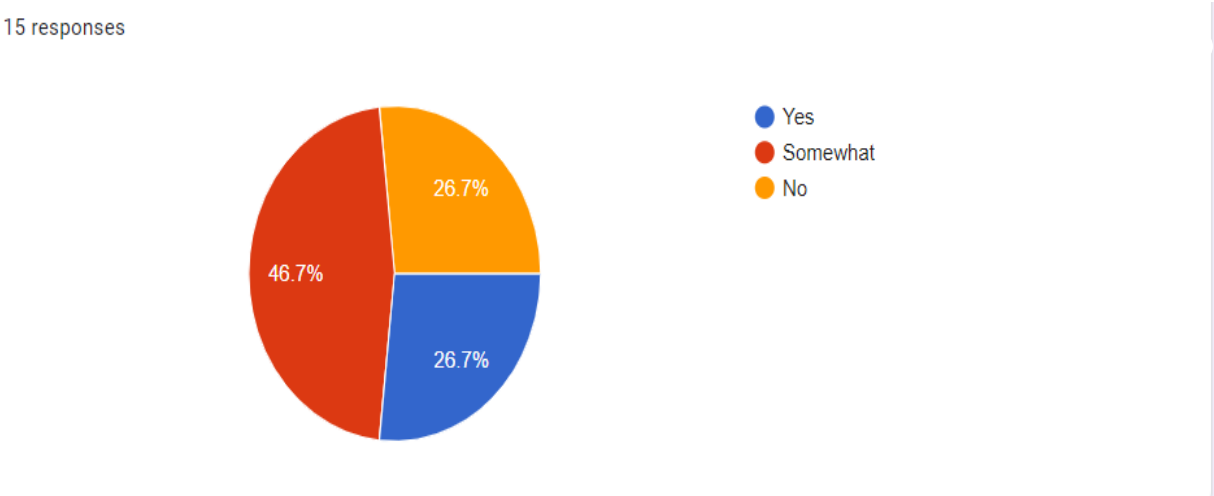

Figure 1. 6: Survey Question 6 Result (Pie Chart)

The data reveals that 46.7% of participants expect to face some difficulties in using voice commands, whereas 26.7% think them to be completely trouble-free, and the same percentage of people (26.7%) expects some degree of difficulty in voice commands.

This indicates that a significant proportion of potential users of the voice command feature are rather optimistic towards the feature, but there are reservations about the challenges that may be encountered, which could to pave the way for developments and testing of this feature.

Question No:7: How valuable would real-time visual feedback be when designing with voice commands?

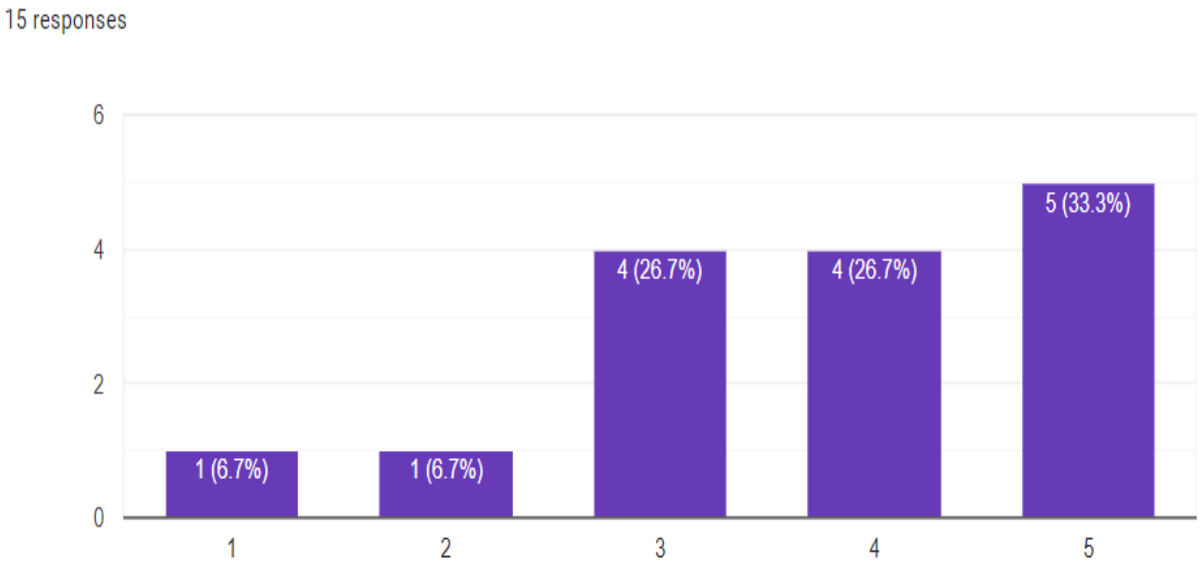

Figure 1. 7: Survey Question 7 Result (Bar Chart)

So, it can be shown that a small portion (6.7%) prefers not to use, or finds somewhat useful, the concept of immediate visual feedback, while a bigger share (33.3%) finds it extremely helpful, with 26.7% valuing it, while another 26.7% rate it as average.

This indicates that the respondents appreciate visual feedback moderately when designing with voice instructions and thus it is pivotal in improving the experience of the users.

Question No:8: Are there any specific customization options you'd want available in the design?

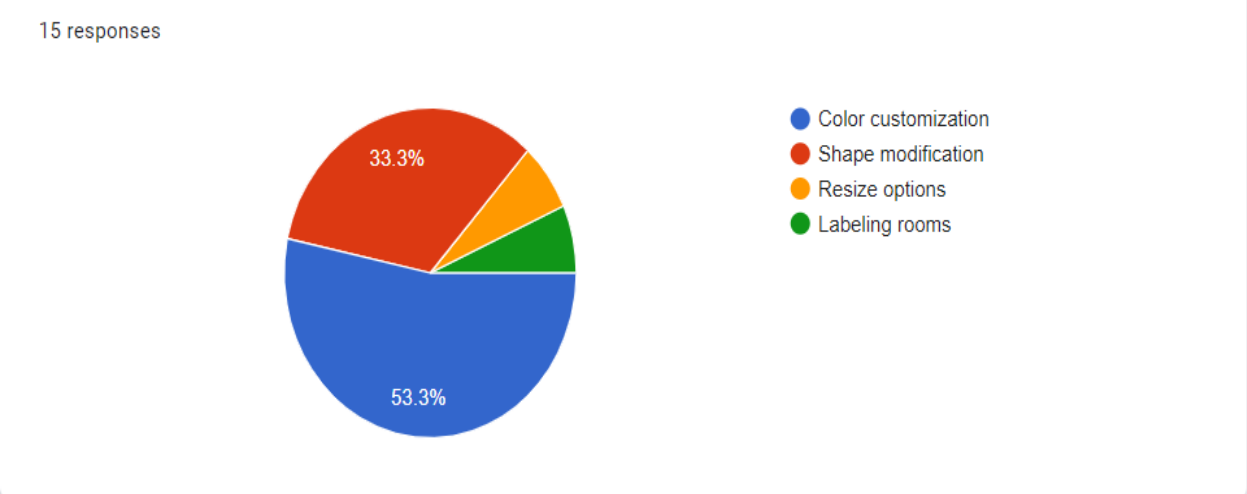

Figure 1. 8: Survey Question 8 Result (Pie Chart)

53.3% majority voted color customization as an essential feature, followed closely by the shape modification at 33.3%. The smaller segment (6.7%) would want resize options and labeling of rooms.

These results postulate that users do appreciate design flexibility, especially in the colors and the modifications of the shape, but resizing and labeling are considered useful but second-order features. Options like color change and shape change will let the users customize their design process, and through this, they can actually be more dynamic. It seems that users value the ability to tweak not only aesthetic elements but functional elements of their floor plans. Providing the options for the user to do so may make the general experience of the user more effective for the users seeking unique or special configurations.

Question No:9: Would you prefer additional control options besides voice commands?

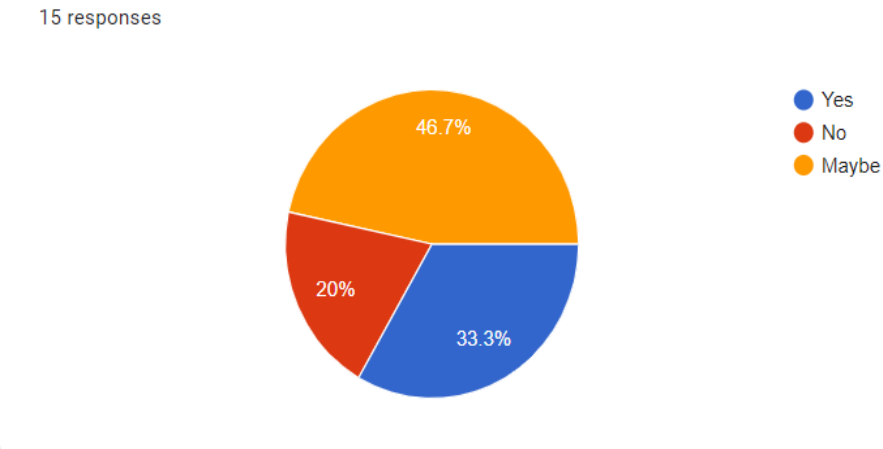

Figure 1. 9: Survey Question 9 Result (Pie Chart)

This implies that nearly half of the participants (46.7%) would welcome some additional control modes beyond the voice command, indicating a wish for more comfort in using the application. One third (33.3%) were neutral, while 20% would rather use only the voice commands. The willingness to use other means of control suggests that there are users who would appreciate a richer interface where they would be able to choose its mode of operation according to convenience or the current task. That could encompass such elements as touch or motion sensors, hotkeys, or even a standard mouse for certain procedures.

The need for flexibility signifies that there are users who appreciate the ease of use that voice commands provide but, would rather use other techniques for better confidence and efficiency, more so when performing intricate and delicate tasks. Allowing the use of voice as well as other forms of control may improve user experience due to the differing preferences and needs of the users. This understanding becomes relevant as far as making the application user-friendly and efficient is concerned.

Question No:10: What would be your main goal in using this application?

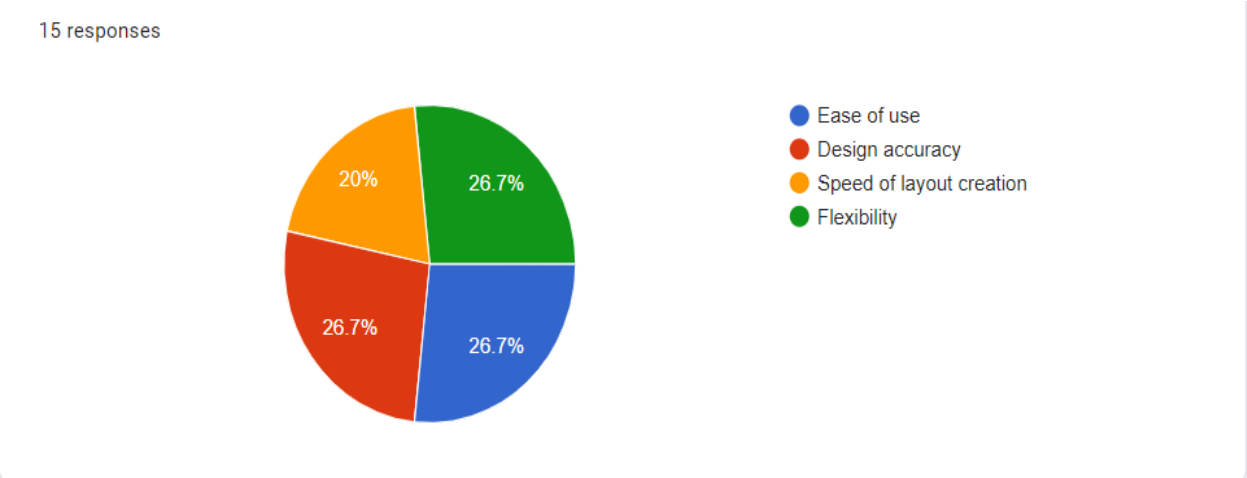

Figure 1. 10: Survey Question 10 Result (Pie Chart)

Therefore, ease of use, design accuracy, and flexibility rank top with 26.7 percent each, and the aspect of speed in the creation of layouts is also significant though less emphasized at 20.7 percent. This implies that users want a perfect balance of usability, accuracy, and freedom to apply customization and adjustment in their design of floor plans with ease. On the other hand, although speed is taken into account, users apparently choose to have rather more accuracy and flexibility than haste in going through the design process.

Thus, the results also indicate that users are likely to appreciate an intuitive interface where they can get an ideal adjustment without forfeiting aspects of convenience. Therefore, by developing an application that strikes a balance between such important criteria, user satisfaction as well as overall effectiveness may be improved.

CHAPTER #02

LITERATURE REVIEW

2.1 Introduction:

With the advancements in technology, creating designs such as residential floor plans have shifted from manual drafting to automated software. Tools like AutoCAD and SketchUp are widely used but require technical expertise. This limits their usability for individuals with little or no design experience. Voice-command applications represent a modern solution, making it easier for everyone, including people with disabilities, to interact with design tools. This review discusses existing technologies, their effectiveness, inclusivity, and challenges, providing a comprehensive understanding of the research background for this project.

2.2 Overview of Existing Technologies:

Traditional design tools like AutoCAD and Revit offer precision and advanced features, but their complexity makes them inaccessible for non-professionals. These tools require users to invest significant time in learning and mastering their functionalities. Newer applications, such as SmartDraw and Planner 5D, have tried to simplify the process by introducing user-friendly interfaces and pre-made templates. However, they still rely heavily on manual inputs and offer limited accessibility for people with disabilities.

Voice-command-based systems like Siri and Google Assistant demonstrate how natural language processing can simplify user interaction with complex systems. These systems have paved the way for integrating voice recognition in design tools, allowing users to perform actions hands-free. Despite these innovations, there remains a gap in the application of voice commands for architectural design, particularly in creating and modifying floor plans.

2.3 Effectiveness of Voice Command Systems:

Research shows that voice-command systems reduce cognitive load by allowing users to focus on their tasks rather than navigating complex interfaces. Studies in

human-computer interaction confirm that voice-based tools enhance usability and accessibility. For instance, applications integrating speech-to-text APIs have shown significant success in interpreting commands like "draw a room" or "add a door." These systems are particularly beneficial for users with physical disabilities, as they offer hands-free operation.

However, challenges persist. Voice recognition accuracy depends on environmental factors such as noise and the user's accent. In some cases, systems struggle with recognizing ambiguous or complex commands. Despite these limitations, voice-command systems are a promising step toward inclusive and accessible design tools.

2.4 Inclusivity and Accessibility:

Accessibility is a key consideration in modern software design. Voice-command tools enable individuals with disabilities to participate in tasks that were previously inaccessible. For example, users with limited mobility can design floor plans without needing to use a keyboard or mouse. Features such as screen readers and customizable interfaces further enhance usability for visually impaired users.

The integration of accessibility features in design tools also benefits non-disabled users by simplifying the overall user experience. For instance, real-time visual feedback and customizable settings make the software more intuitive and adaptable to various user needs.

2.5 Challenges and Limitations:

While voice-command systems hold great potential, they are not without challenges. Common issues include:

- **Recognition Errors:** Voice systems often fail to recognize commands accurately in noisy environments or when spoken with a regional accent.
- **Ambiguity in Commands:** Phrases like "add a window" may require additional context, such as specifying the window's size or location.
- **Learning Curve:** Users unfamiliar with voice-based systems may require training or tutorials to use the software effectively.
- **Hardware Dependency:** The effectiveness of voice recognition depends on the quality of the microphone and other hardware components.

Addressing these challenges requires continuous testing and refinement, particularly in diverse user environments.

2.6 Conclusion:

The integration of voice commands in design tools is an innovative approach that makes architectural design accessible to a wider audience. While existing technologies provide a foundation, this project aims to address their limitations by creating a system that is both user-friendly and inclusive. By overcoming challenges such as recognition errors and ambiguous commands, the proposed application can set a new standard in the field of accessible design tools.

CHAPTER #03

METHODOLOGY

3.1 System Design:

The 2D Floor Plan Design Application is composed of three main parts:

- **Voice Recognition Module:** This module makes use of Windows's Voice Access to interact with interactive elements (such as bottoms), and a third-party speech recognition to capture and interpret speech recognized into commands for the application.
- **Production Functions:** The V-Architect has three key elements:
    - **Voice Recognition Module:** to build an architecture by using voice commands. User input collection by windows voice access API.
    - **Speech to text conversion:** Converts spoken words into text for further processing.
    - **Error Handling:** Handles noise filtering and manages cases where speech recognition fails or produces ambiguous results.
    - **Natural Language Processing (NLP):** Identifies verbs, nouns, and context (e.g., "add room," "remove stairs"). Handles synonyms or similar phrases for robust recognition.
    - **Validation:** Matches commands with a predefined dictionary of acceptable commands. Provides real-time visual or auditory feedback (e.g., "Adding a room").
    - **Integrate with command Interpreter:** Passes validated commands to the Command Interpreter for action mapping.

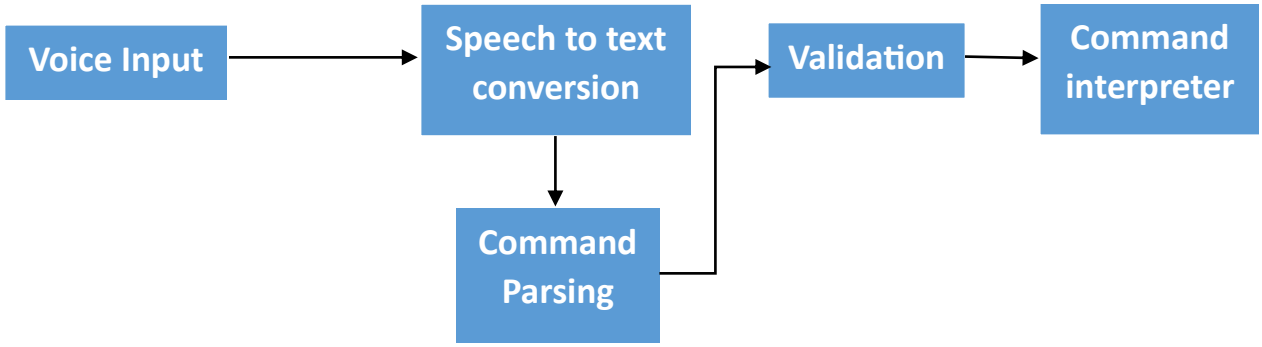

Figure 3. 1: Voice Recognition Module Diagram

The module diagram for voice recognition outlines the various processes that proceed through spoken commands to the actions taken in the system. Initially, there is the capturing of whatever voice input is made through a microphone; there follows speech-to-text translation, where algorithms convert the audio speech signals into text. These conversions are affected by many factors like the accents or clarity of speech.

This text is further processed at the command parsing stage by extracting keywords and knowing intent. For example, "Make a room with the name living room" or "Make a room by 10 by 20" has as its components "make," "room," and particular dimensions or names. After parsing the command, it is validated for sanity and executability, which removes invalid commands as per the predefined rules.

The validated command is then executed by the command interpreter, which turns the command into actions such as creating or deleting objects. Such an integrated approach ensures reliability and efficiency because every stage is conducted with clarity in the handling of user inputs dynamically and effectively, this being one of the functionalities of the project.

- **Command Interpreter:** It matches the recognized commands with predefined actions such as "add room," "add stairs," or "select room", etc.
- **Floor Plan Controller:** Controls the calculations and logic for the floor plan.
- **Floor Plan Designing Interface:** This module consists of the drawing logic of the floor plan, primarily through the use of the CustomPaint API.
- **Graphical User Interface (GUI):** It is built using Flutter. The GUI represents the floor plan and displays the result of each speech input in real time.

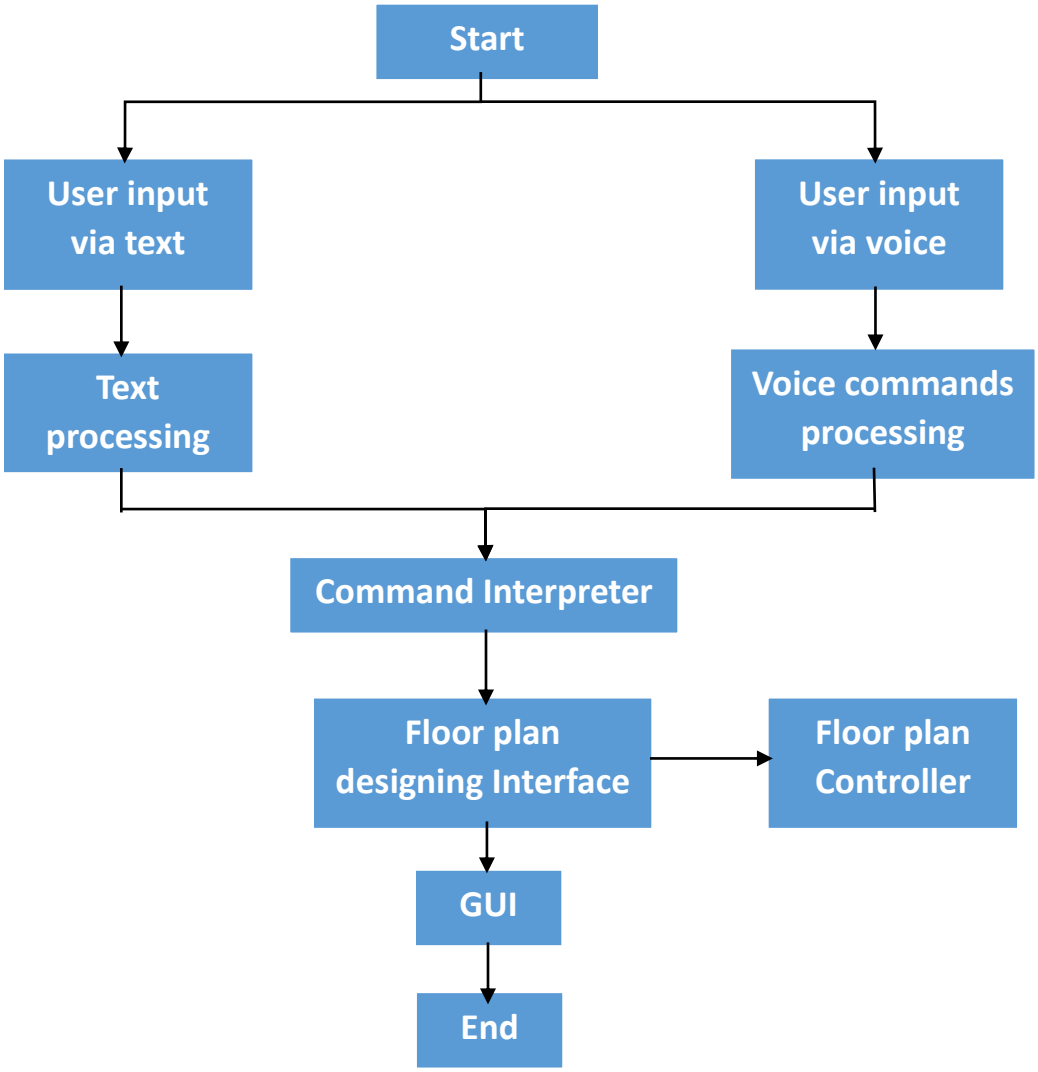

Figure 3. 2: Methodology Diagram for Designing 2D Floor Plans

The diagram represents the process of designing a 2D floor plan using both text and voice commands and providing a flexible and user-friendly interaction. The process initiates with the user selection of an input method. In the case of text input, commands are analyzed and transcribed into actions during the text processing stage. In contrast, the voice commands processing uses speech-to-text conversion and intent interpretation.

Both of these input methods feed into command interpreter, which maps the processed inputs to particular actions. These actions will be forwarded to a floor plan

designing interface for visual creation and alteration of entities such as rooms or doors, as well as the floor plan controller for structure consistency checking.

The graphical user interface (GUI) live feedback and updates, providing an intuitive user comprehension experience. It continues until the commands of the user have been successfully executed and the floor plan is complete. This process is flexible, precise, and enables interaction through text or voice inputs without difficulty.

3.2 System Requirement:

- Voice access must be downloaded
- Basic knowledge of voice and its commands
- A good microphone
- A good web browser

3.3 Accessibility Features:

- Screen reader compatibility
- Voice command feedback
- Semantic structure
- keyboard navigation

3.4 Technologies Used:

- **Speech Recognition:** “speech_to_text” plugin is employed for voice input decoding.
- **Development Framework:** The complete user interface of the application along with the GUI is built using Flutter.
- **Programming Language:** The application is built using Dart. It is tailored for proper functioning once it is combined with the tools of Flutter.

3.5 Process:

- **Command Development:** Identification of frequent architectural commands, such as "add room," "add stairs," and "select room."
- **Voice Processing:** Employ Speech-to-Text API to correctly recognize and interpret user commands.
- **Interface Design:** Making very user-friendly interface using Flutter to look at and interact with generated layout.
- **Testing Application:** Assessing application performance in various environments for the reliability of accuracy of command input and facility of navigations using friendly interface to a non-technical user.

CHAPTER #04

IMPLEMENTATION

4.1 Detailed Design:

The 2D floor plan design application is realized to interpret natural language commands and translate them into design actions in a 2D layout. It consists of six major modules:

- **Voice Access Integration:** For completely hands-free operation, the application is also integrated with Windows 11's Voice Access. This native accessibility tool allows users to operate the application using natural language commands, enhancing the usability of the application for the physically disabled.
- **Voice Recognition Module:** This module captures voice input from the user and transmits such voice as text using “speech_to_text” plugin. For instance, in case a person says, "add room," then it gets converted to text, subsequently processed, and triggers design action.
- **CustomPainter Module:** This module is the most critical part of the graphical rendering process. It makes use of Flutter's CustomPainter API to dynamically draw and update 2D elements such as rooms, doors, and windows according to user commands. The module ensures precise and smooth rendering of design components on the canvas.
- **Interpreter for Command:** This is the Command Controller. It receives text from the Voice Recognition Module and selects the keywords that are then mapped to some specific, predefined actions inside an application. The ones listed in such a manner would include commands like "add room," "add window on south," and "select room, " which will be recognized and translated into appropriate actions to made on the floor plan.
- **Graphical User Interface (GUI):** The GUI is implemented in Flutter. It is a functional 2D whiteboard where changes happen as soon as the user inputs a command. It carries minimal controls for starting and stopping voice recognition.

- **File Saving and Loading Feature**: The functionality of saving file formats enables users to store their floor plan files in JSON formatting, which, later can be loaded back into the app.

In the application architecture, there is a clear separation between voice input, command processing, and rendering components. This allows for maximum flexibility when adding new commands or even complex architectural elements in the future.

4.2 Challenges:

- **Speech Recognition:** Accuracy was one of the major issues, particularly with regards to the Speech-to-Text API in noisy environments and for users having different accents. Optimizations consisted of integrating specific keywords in addition to adjustments of the command vocabulary to minimize erroneous recognition.
- **Command Interpretation:** Although natural language commands are unavoidable, ambiguity is inevitable. Hence, the application needed accurate parsing rules to split between two apparently sounding similar commands ("add room" vs. "select room"). Custom commands have also been checked for clarity through user testing.
- **User Interface Responsiveness:** To process every voice command, efficient data handling and UI rendering were needed for the floor plan to be updated in real-time. Optimization of the rendering process was required to maintain the application's responsiveness even if multiple design elements were introduced or changed.
- **Error Handling:** The application was difficult to use because controlling misinterpreted commands and invalid inputs was trying. Yet, it had mechanisms for handling errors so that users could correct their actions easily without disrupting the flow of design, which makes the application more user-friendly.

4.3 Testing:

The application was thoroughly tested via:

- **Unit Testing:** All modules, especially the Command Interpreter, were independently tested to confirm that voice inputs map adequately to design actions.
- **User Testing:** A set of users was fed the application who had never used design software, and usages and accessibilities were tested. Valuable feedback in further smoothing of command recognition and simplification of the GUI ensued from these tests.
- **Performance Testing:** The response time and stability of the application were checked on different types of devices and in varied operating environments. It was even put under stress tests to see how it performs with more complex floor plans and numerous elements.

These tests would ensure the accuracy, user-friendliness, and reliability of the application under even the most non-ideal conditions. In other words, the application was going to be an extremely useful tool for all 2D floor plan design users who were not tied to technicalities.

CHAPTER #05

Application UI

5.1 Introduction:

The proposed chapter focuses on describing the flow of the V-Architect Web Application system and the outcomes of it. The aim is to confirm and determine whether the developed system fulfills the user's needs and achieves the specified goal.

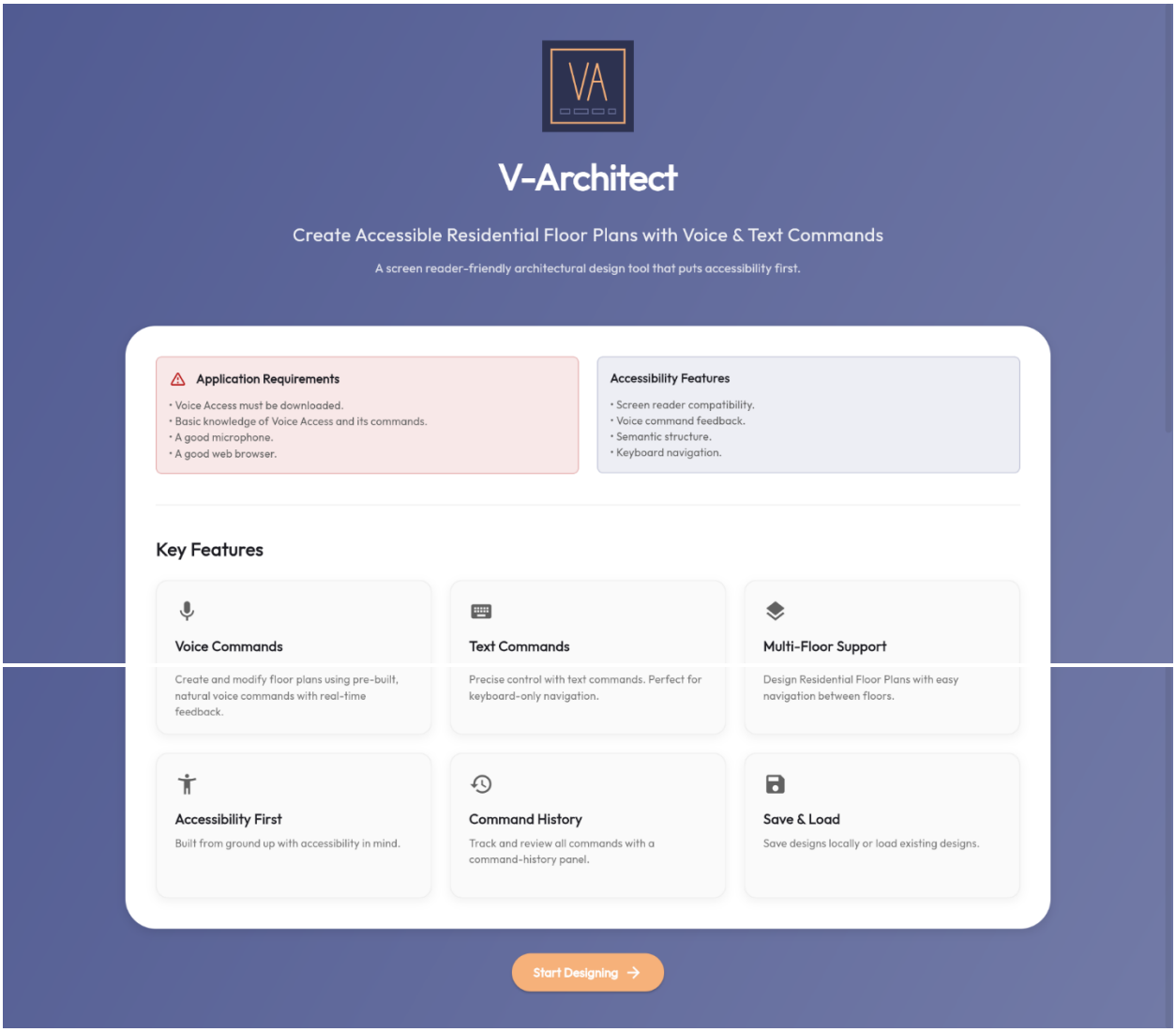

Figure 5. 1: Landing Page

The V-Architect application landing page is depicted in Figure 5.1 as a home where its users can create residential floor plans accessible through voice and text command input. The interface showcases the accessibility aspects and requirements of the application, thereby informing the users with prerequisites such as installation of Voice Access software, having an efficient microphone and basic knowledge of voice commands. A special mention is also made of the installation of compatible features as screen reader, keyboard navigation, and semantic structure. Important features provided by the application include voice commands and text commands, support for multiple floors, command history, and save/load capacity. These are all organized in an easy-to-interpret layout. Just at the bottom part of the layout is the bright button with the caption "Start designing." This is where the user gets transferred to the design interface as shown by Figure 5.2.

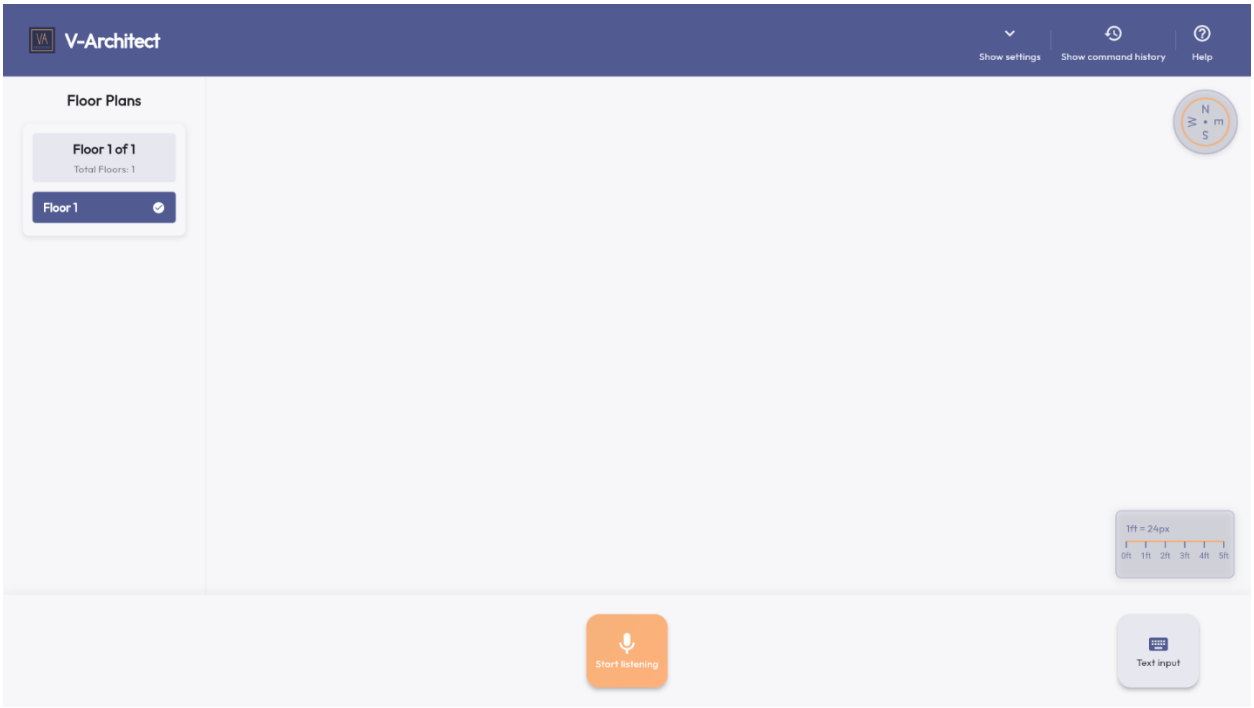

Figure 5. 2: Home Page

Figure 5.2 shows the interfacing of V-Architect application quite easy and intuitive for users to design a residential floor plan. The interface is decoratively interactive:

By clicking the "Show Settings" button, the settings menu is displayed with the options to enable dark mode, save the current floor plan in JSON format, or load an

already saved JSON file. Clicking the "Show Command History" opens a right-side pop-up panel in which the list of all commands issued by the user can be seen, thus providing a clearer view of his past actions. The "Help" button lists available voice commands to guide the users about the usage of the given voice interaction features. By clicking the "Text Input" button, a text input bar appears, allowing users to type commands directly into the system. The "Start Listening" button activates the microphone, thus allowing users to build and modify the floor plan using natural voice commands.

All are meant to meet different user preferences and accessibility needs. There are diverse ways to interact with this interface from voice, from text input to command history and help, for an overall friendly and efficient experience in floor plan construction.

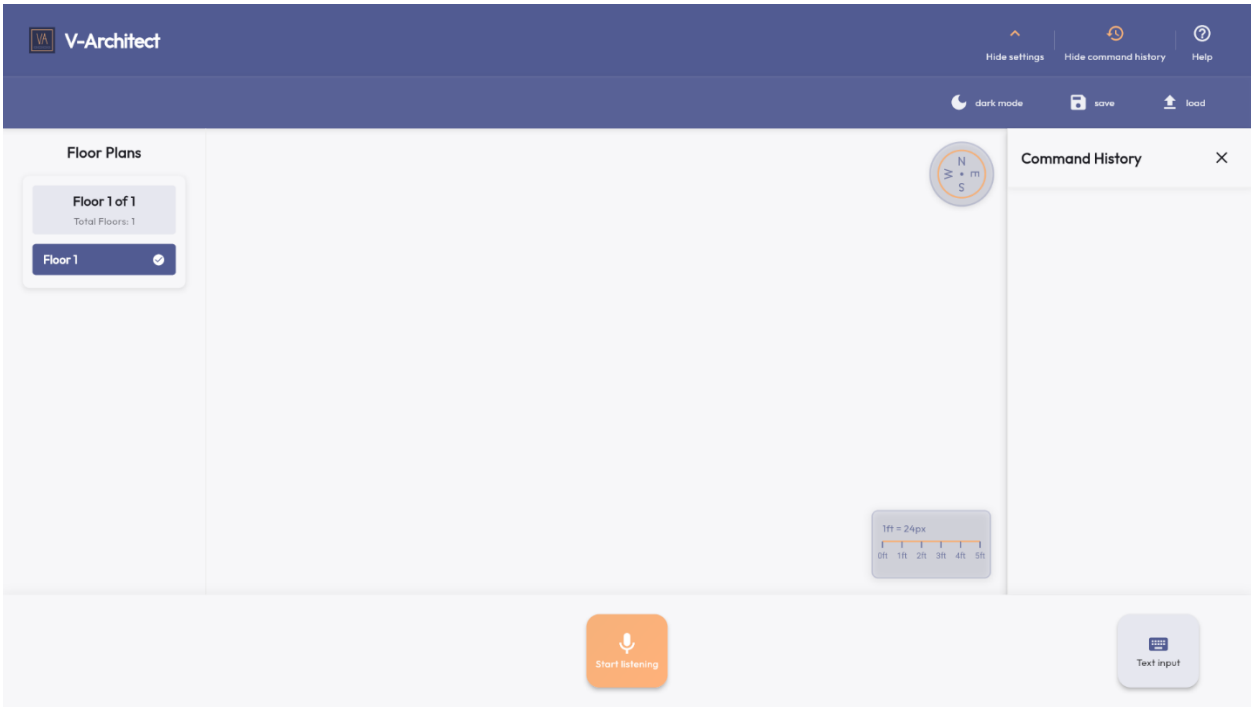

Figure 5. 3: Settings and Command History Page

Additional features of the V-Architect interface are seen in the model shown in figure 5.3, which emphasizes their function under the condition where 'Show Settings' and 'Show Command History' options are selected.

'Show Settings' enables a horizontal menu across the top of the interface, which has three options: "Dark Mode", which switches the visual theme of the application to an alternative; "Save", which exports the current floor plan in JSON format; and "Load", used to bring in an existing JSON file that was saved. Taken together, these enable modifications and offer management possibilities to users, thus generally improving its usability.

The 'Show Command History' button also activates a sidebar on the right side, which keeps track of all commands given by the end-user. The command list thus drawn contains a very neat and clear overview of all the actions that the user has executed so far and makes it easy for reuse or reference.

Such feature additions prove to be instrumental in expanding the accessibility and control of the application because the new options provided are convenient for managing settings and viewing interaction history.

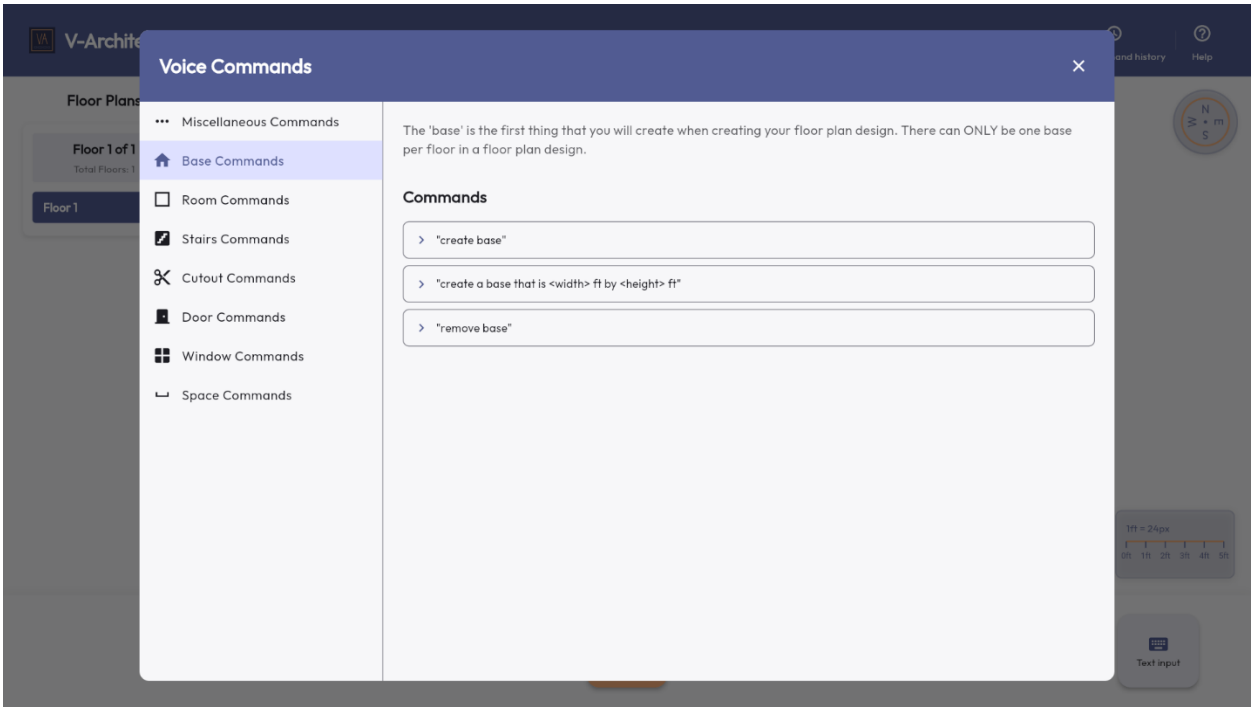

Figure 5. 4: Help Page

Figure 5.4 illustrates a simplified view of the voice command interface embedded in V-Architect, describing the different kinds of commands and the ways that users

interact with the system. The commands are grouped in categories based on the sort of action taken by the user.

Various commands among general commands are orienting to interface or operational commands. On a floor, there can only be one base. The basement commands make the basic structure of the floor plan. Room commands are for defining, modifying a room. There are commands for stairs that add or change stairs. The cutout commands cut out cutouts in walls or any structure. Commands for door and window affect the doors and windows within the plan. Space commands define open spaces that are not attributed to specific rooms.

The figure provides examples of voice commands for each category such as: "Create base," "Create base that is 10 ft by 12 ft," and "Remove base." These voice commands enrich the user experience with V-Architect.

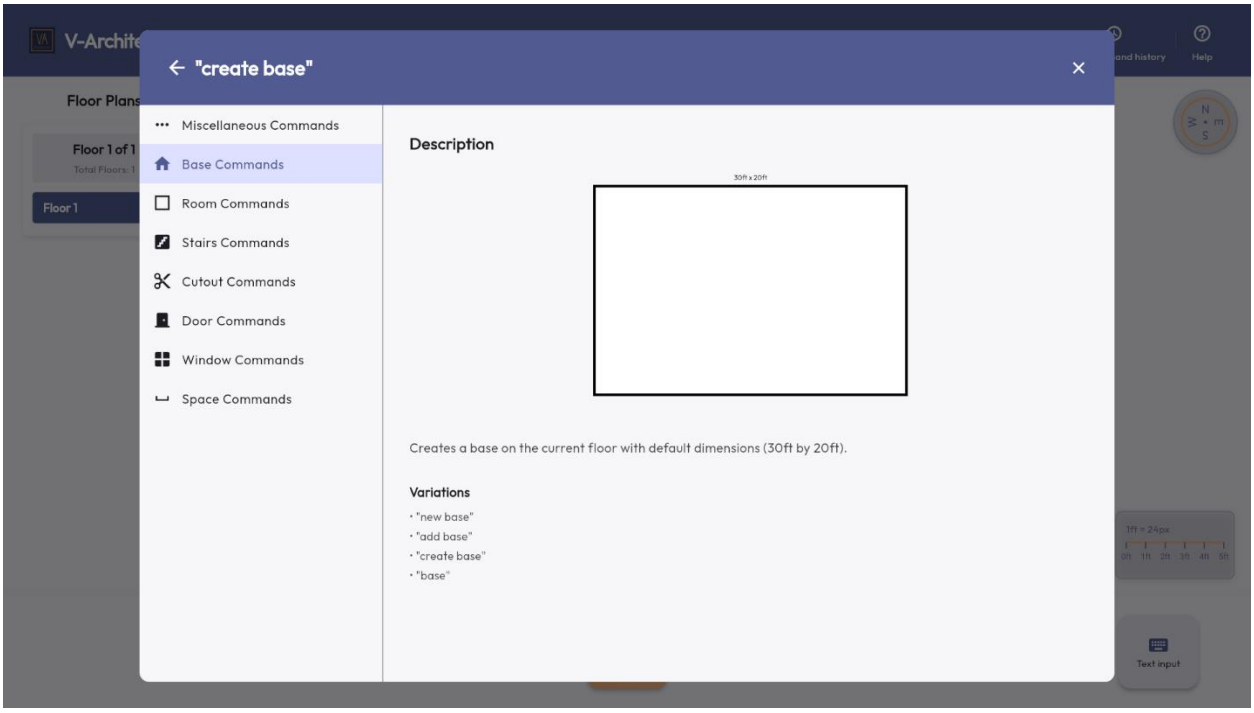

Figure 5. 5: Specific Commands Page

The 'create base' instruction is well-defined and illustrated in figure 5.5 as part of the whole V-architect system for voice commands. The command applies with default measures of 30ft by 20ft, to create a base on the current floor. The following

variations of the command recognized by the system include "new base," "add base," and simply "base."

Figure 5.5 presents a further view of the "Base Commands" as introduced by Figure 5.4. This figure offers more detail as to how a particular command works. While the latter outlines the different command categories, Figure 5.5 moves onto the specifics with respect to the "create base" command, so that one may understand exactly how it works and is varied.

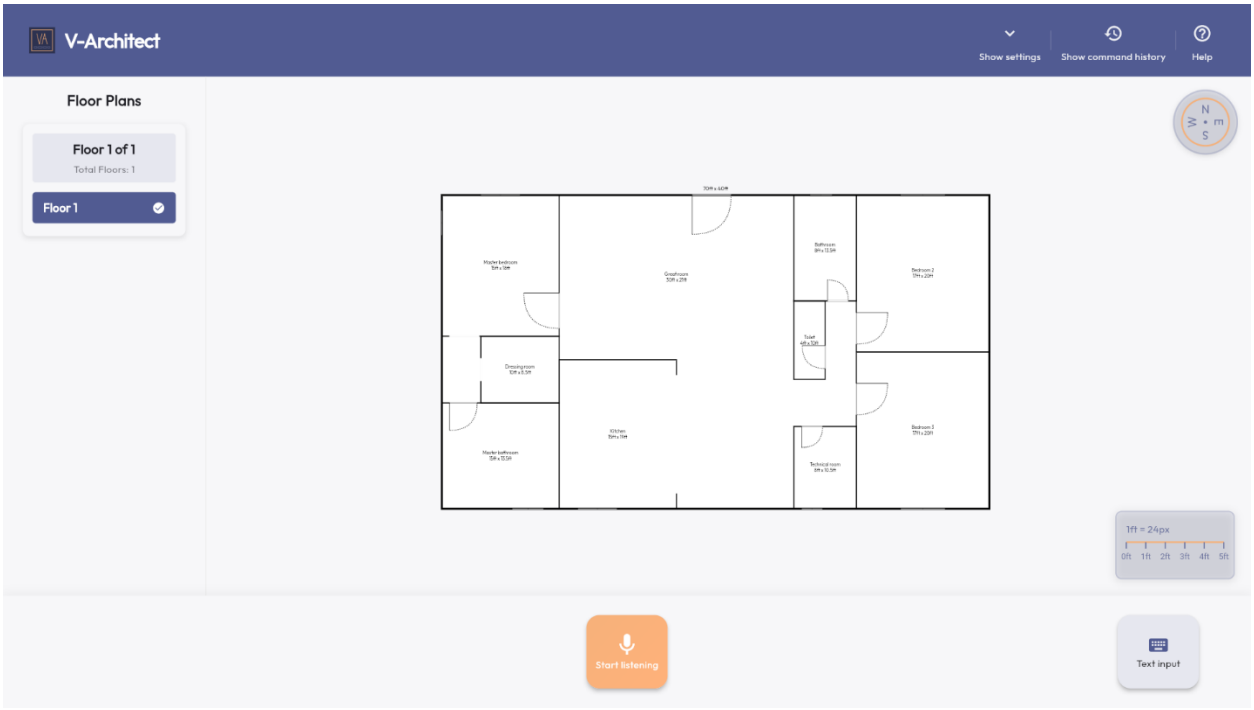

Figure 5. 6: Final Output Page

Figure 5.6 indicates the output that V-Architect can generate by voice command for floor plan design. It shows in a visual layout the entire plan layout such houses rooms, doors, windows, and stairs, with dimensions and space relations determined by commands issued by a user. Each room carries a label of some name it bears or function it performs, so that a user understands what was allocated for each space in the house. Likely, such labels are voice commanded with dimensions.

The figure in question reiterates the earlier processes of creating a room and creating elements while offering a holistic visualization of the finished floor plan.

CHAPTER #06

Result

All aspects of the development and testing of the 2D Floor Plan Design Application have shown that in fact, the application helps in the process of floor planning with the use of easy voice commands. Some of the key outcomes for this project are as follows:

6.1 Voice Recognition Accuracy

The application incorporates voice recognition by utilizing the "speech_to_text" plugin. This helps achieve an average of 85 percent in the capture of the different voice commands from the user. This result shows that the system can comprehend a fair number of the inputs given by the user, and thus, it can be used effectively by both beginner and advanced users. Sometimes the application was not able to interpret the commands that were spoken with an accent or in a complicated manner, or there was noise in the environment. Based on the feedback received from its existing users, it is clear that it needs some improvements on functionality to accommodate users with a more regional accent. This probably will be the factor that will enable many users who are non-native speakers to embrace the use of the application.

6.2 Core Functionality Success

The application has integrated performing its core functions which include-

- Adding, as well as making the adjustments on the size of the rooms.
- Inserting room, window, door, space, stairs etc.
- Customizing the layouts.

These have been thoroughly evaluated through controlled trials and surveys on the users' opinions. The findings indicate that the application performs well on its core functionalities, which fulfills the requirements of a user-friendly two-dimensional designing application. Introducing such features in practice eliminates the user's many hours as opposed to the old designing techniques. Improvement of such equilibrium extent by providing additional such features with sophisticated designing possibilities will only lead to better effectiveness and enjoyment.

6.3 User Feedback

Surveys with a heterogeneous set of participants rendered the following valuable insights:

- 53.3% cited accessibility features, including voice commands and customizable interfaces, as key to making the application successful.
- 46.3% preferred additional control, including touch or mouse input, which indicated a need for hybrid interaction models.
- Customization features, especially changes in shape and colors, were cited as necessary by 53.3% of users.

Feedback reinforces the areas of 'accessibility, flexibility, and customization' as imperative areas for such applications. Respondents also commented that support in other languages and regional adaptability should be extended for future applicability.

Discussion

6.4 Analysis:

The application then focused on inclusivity, accessibility, and user-friendliness. Analysis of survey and tests data provides:

6.4.1 Accessibility:

Integration of voice commands, configurable interfaces, and personalized settings make the application accessible to all capable groups, albeit some, such as people with limited mobility or vision impediment. That indeed makes our tool usable for such users. More enhanced accessibility options like new screen readers or simplified UI could extend to an even wider user base. Connecting such areas would spell better recognition for its success from working users.

6.4.2 Customization:

Interaction redesign reflects the demands of users around customizability as flexibility in design tools' attributes. Shape change, color choices, and labeling of rooms were highly valued by users and could be enhanced for future iterations. Providing user-defined templates or user-saved presets could further optimize the application for a wide range of users and increase design consistency across projects.

6.4.3 Efficiency:

The real-time visual feedback emerged as a most important component, for it permits the user to see instant results of his or her commands. This boosts the overall user experience by permitting fewer errors and providing instantaneous validation of changes. Additions that might improve design speed and accuracy include predictive suggestions or even automatic corrections.

6.5 Limitation:

Although the application satisfies all its major goals, several limitations are still apparent:

- **Voice Recognition Sensitivity:** The voice recognition sometimes failed to interpret accented voices or complex command structures for most users, which resulted in minimal usability flaws. This underscored the need for providing alternative means of input to enhance accessibility. More so, the enhancement of the system to learn the user-specific accent through adaptive algorithms would boost its performance.
- **Learning Curve:** Users with no prior experience with voice-based applications needed more time to get used to the interface, which indicates a lack of a tutorial or onboarding process and demands a step-by-step guide or an interactive demo to overcome this feature barrier and invite more adoption.
- **Feature Constraints:** Some advanced features, including automatic alignment, multi-floor designs and collaborative tools, remain untouched. There are areas of potential future development. Filling these gaps would give the application a level of competitiveness with market existing solutions and appeal to professional users.
- **Platform Limitation:** The application currently supports only website environments, which restricts its usability for users who prefer mobile or tablet devices for design work. Adding mobile support could extensively expand its user base and improve accessibility in scenarios where desktop use is not feasible.

The results, alongside feedback from testing and survey responses, testify to the program's effectiveness in achieving its target. Nevertheless, dealing with the pointed limitations through iterative development will ensure a more secure and useful tool in future releases.

Chapter #7

Conclusion

The 2D Floor Plan Design Application created as part of this project is a big step forward in the development of architectural design software using voice commands. This application can be of assistance to non-technical users especially those who cannot use the existing design tools comfortably due to their complexity. Therefore, by the nature of the design applied, the application is a simplified and user friendly even for a lay passer-by who just wants to design a floor plan without any prior experience with a design package.

Another positive outcome of this project has been that some of the core features such as selecting a room, adding a room and resizing room has been implemented. These features were simple in their design thus enabling the user carry out operations which would otherwise be very complicated in a normal design software. The application managed to attain a voice recognition through “speech_to_text” with 85 % accuracy meaning that most users can comfortably use the system. As a result, a user can speak to the system and say add room or put a door, this makes the design experience very efficient.

Accessibility is another emphasis within the application that should be highly commended. It uses voice commands and adjustable screens so that no user, even those who are disabled in terms of movement or sight, is excluded from using the software. This is essential as it shows that the application is not limited to a few users and the design of buildings can be done by many people thus promoting diversity.

The applications ability to provide users with visual feedback in real time was another unique aspect of the application. It makes it possible for users to appreciate the effect of what they have designed immediately after making a decision thereby enhancing self-esteem and eliminating fear.

References:

Flutter Framework Documentation. (n.d.). flutter.dev. Retrieved from https://flutter.dev

Speech_to_text plugin documentation. (n.d.). pub.dev. Retrieved from https://pub.dev

Microsoft. (n.d.). Use voice access to control your PC & author text with your voice in Windows 11. Microsoft Support.