

# Chapter 1: Introduction

## Overview

The Interactive Floor Plan Designer is a Java Swing-based application that empowers users to creatively design and manipulate floor plans. It is tailored for architects, interior designers, and enthusiasts who wish to visualize space layouts efficiently.

## Project Goals

The primary objectives of the project were:

- **Intuitive Design Interface:** To develop a graphical user interface that is both intuitive and user-friendly, allowing users of all skill levels to easily navigate and utilize the tool's functionalities.
- **Flexibility in Design:** To enable users to create, modify, and visualize floor plans in a dynamic 2D space, offering a wide range of customizable elements such as walls, furniture, and fixtures.
- **Efficiency and Productivity:** To enhance the efficiency of designing floor plans by offering features like templates, drag-and-drop capabilities, and easy manipulation of design elements.
- **Collaboration and Sharing:** Although earmarked for future updates, the goal to facilitate collaboration among users and the ability to share designs easily remains a core long-term objective.

## Scope

The scope of the Interactive Floor Plan Designer was defined to encompass the following:

- **Core Functionalities:** The project focused on delivering essential features necessary for designing basic to moderately complex floor plans. This includes the placement and adjustment of walls, furniture, doors, windows, and other architectural elements within a 2D space.
- **Customization and Flexibility:** The tool was designed to offer a high degree of customization, allowing users to tailor dimensions, styles, and other attributes of floor plan elements to fit their specific needs.
- **User Experience:** Priority was given to ensuring a seamless user experience, with a clean and straightforward interface,

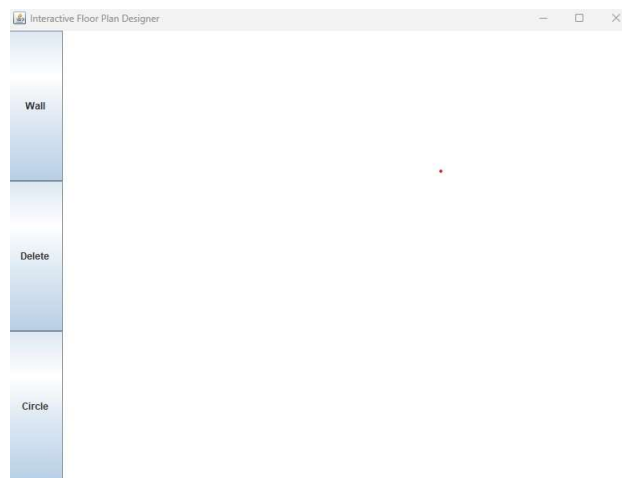
# Chapter 2: Project Report

## 2.1 Project Timeline

This timeline provides a week-by-week breakdown of the key development activities, achievements, and deliverables for the Interactive Floor Plan Designer project. Each week marks a distinct phase in the project's lifecycle, from initial setup to final preparations for demonstration.

### Week 1: Project Kickoff and Basic Setup

- **Initial Setup:** Configuration of the development environment, including setting up the IDE, version control, and project repository.
- **Design Elements Palette:** Development of the UI component that allows users to select various design elements (walls, furniture, etc.) to include in their floor plans.
- **Drawing Canvas:** Creation of the central canvas area where users can draw and visualize their floor plans.



*Figure 1. Week 1 progress.*

### Week 2: Core Functionality Development

- **Placing Design Elements:** Implementation of the functionality allowing users to place selected design elements from the palette onto the canvas.
- **Selecting and Manipulating Elements:** Development of the ability to select individual elements within the canvas and manipulate them (move, resize, rotate).

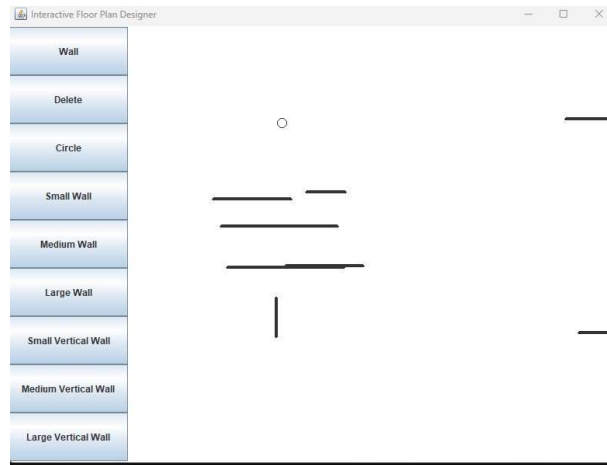


Figure 2. Week 2 progress.

### Week 3: Advanced Features and User Interaction

- **Save and Load Capabilities:** Adding functionality for users to save their floor plan designs and load them back into the application for further editing or review.
- **User Interaction Refinement:** Enhancing the interaction model to make the manipulation of design elements more intuitive and efficient, including improvements to element selection, dragging, and property editing.

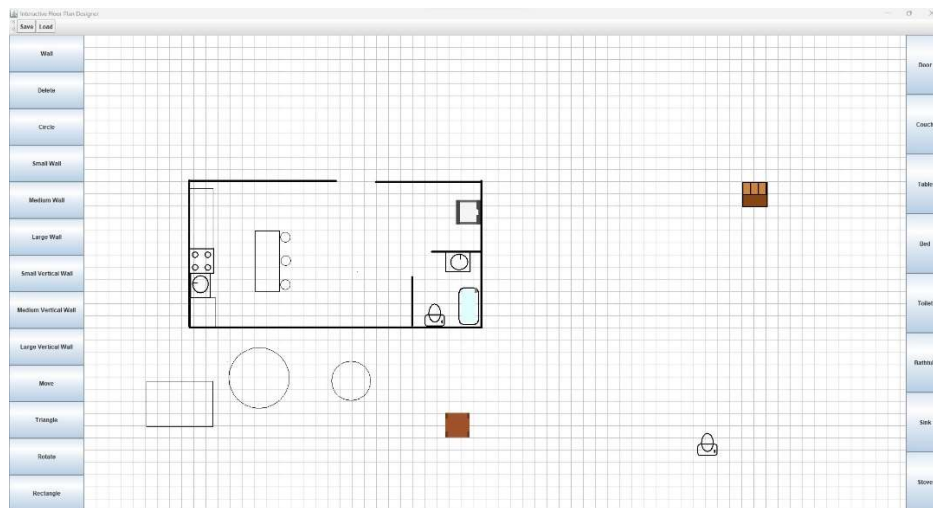


Figure 3. Week 3 progress.

### Week 4: Finalization and Demonstration Preparations

- **Final Testing:** Conducting comprehensive tests to identify and resolve any remaining issues, ensuring the application's reliability and usability.

- **Documentation:** Compilation of essential documentation, including a user guide detailing how to use the application and a developer guide outlining the project's architecture and codebase.
- **Demonstration Materials:** Preparation of materials for project demonstration, such as a presentation video showcasing the application's features and a live demo setup for potential users to explore.

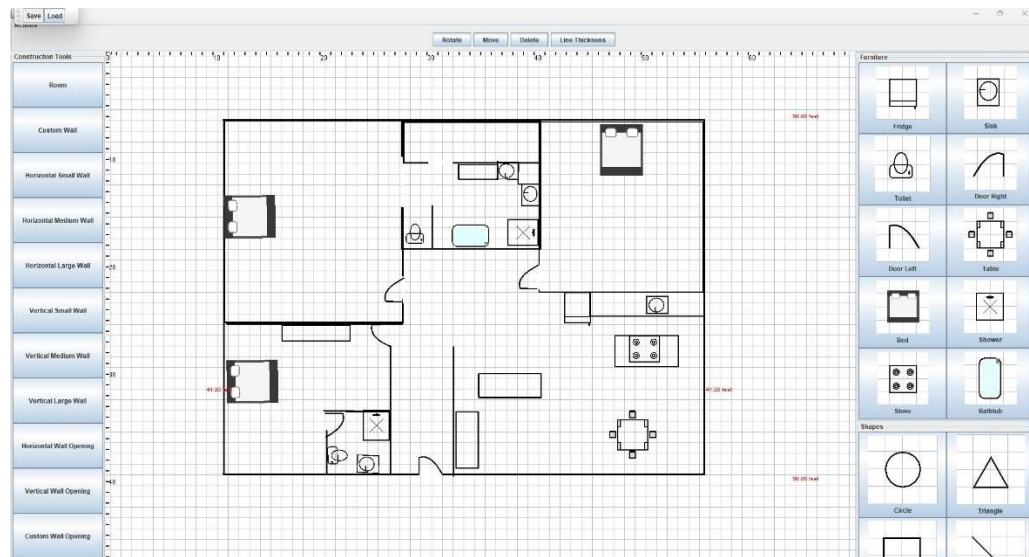


Figure 4. Week 4 progress.

## 2.2 Implementation Details

The implementation of the Interactive Floor Plan Designer involved a methodical approach, divided into key development phases, with a careful selection of technologies tailored to meet the project's requirements. This section provides an overview of these phases, the technologies employed, and some of the notable challenges encountered along with the solutions devised to overcome them.

### Key Development Phases

1. **Requirement Analysis and Planning:** The project commenced with a thorough analysis of requirements, identifying the necessary features, user interface design, and user experience goals. This phase included creating wireframes and prototypes for the application's UI, establishing a foundation for the subsequent development stages.
2. **Design and Architecture:** With the requirements in place, the next phase focused on designing the application's architecture, adopting the Model-View-Controller

(MVC) pattern to ensure a clear separation of concerns. This architectural pattern facilitated easier maintenance and scalability of the codebase.

3. **Development:** The core development phase involved implementing the planned features, starting with the foundational components like the canvas for drawing floor plans and extending to more complex functionalities such as the drag-and-drop interface for adding and manipulating elements.
4. **Testing and Iteration:** Following development, rigorous testing was conducted to identify and fix bugs. This phase also involved usability testing with potential users to gather feedback, leading to several iterative cycles of refinement to enhance functionality and user experience.

## 2.3 Methodology

The methodology section outlines the structured approach, development practices, and the set of tools that were utilized in the creation of the Interactive Floor Plan Designer. This comprehensive approach ensured the project's success by maintaining a high standard of code quality, facilitating collaboration among team members, and enabling efficient project management.

### Development Methodologies

1. **Agile Development:** The project adopted Agile principles, focusing on iterative development, frequent delivery of functional software. This approach allowed for flexibility in responding to changing requirements and user feedback throughout the development process.

### Tools Used

1. **Java Swing for UI Design:** Java Swing was chosen as the primary toolkit for building the graphical user interface (GUI) due to its extensive library of widgets and its flexibility in customizing components, which was crucial for creating the intuitive and interactive design environment of the application.
2. **Git for Version Control:** Git was utilized for version control, allowing the team to track and manage changes to the project codebase efficiently. This tool was essential for coordinating work among multiple developers and for maintaining a history of the project's evolution.
3. **Maven for Dependency Management:** Maven was used to manage project dependencies, automate builds, and ensure consistency across development environments. This tool simplified the management of library dependencies and streamlined the build process.

4. **Eclipse IDE:** Eclipse was used by the development team to enhance productivity. Features like code completion, intelligent refactoring, and integrated debugging tools were invaluable throughout the development process.

By combining these methodologies and tools, the project team was able to develop a high-quality software application that met the defined objectives and was adaptable to user needs and feedback. This structured approach not only facilitated the efficient completion of the project but also ensured that the final product was robust, user-friendly, and met the high standards set by the team.

## 2.3 Results

The Interactive Floor Plan Designer project culminated in a robust application capable of creating detailed and customizable floor plans. The project successfully met its primary objectives, offering a user-friendly interface for designing floor plans and enabling the addition, removal, and modification of various elements. Key achievements include:

- **Intuitive UI:** The application boasts an intuitive drag-and-drop interface, allowing users to effortlessly place and manipulate architectural elements.
- **Advanced Zoom Functionality:** Implementing zoom centered on the mouse cursor enhances detail work, allowing users to focus on specific areas without losing context.
- **Dimension Display for Objects:** The feature to select objects and view their dimensions directly contributes to precision in design, facilitating accurate space planning.
- **Efficient Save and Load Operations:** These features ensure users can preserve their work and revisit or revise designs as needed, enhancing the application's usability.

## 2.4 Conclusion

The Interactive Floor Plan Designer project stands as a testament to the successful application of Java Swing in developing a comprehensive and user-friendly floor planning tool. It achieves its core objective of simplifying the floor plan design process, providing users with a suite of powerful features wrapped in an intuitive interface.

The project's journey from conception to realization has been rich with learning opportunities, from tackling UI design challenges to implementing complex functionalities like zoom and object manipulation. The decision-making process regarding feature inclusion was guided by a balance between ambition and practicality, ensuring the project remained focused and manageable.

Looking ahead, the foundation laid by this project opens numerous possibilities for expansion and refinement. Future versions could explore advanced visualization techniques, collaborative features, and integration with other design tools, further solidifying the application's value to its users.

In conclusion, the Interactive Floor Plan Designer embodies a significant step forward in making floor plan design more accessible and enjoyable, paving the way for future innovations in the field.

## Chapter 3: Software Design

The Software Design chapter serves as a comprehensive manual for understanding the architectural and detailed design of the Interactive Floor Plan Designer. It elucidates the system's structure, design choices, and the rationale behind these decisions, providing insights into how the application fulfills its requirements while maintaining scalability, maintainability, and usability.

### Architectural Overview

The application is structured following the Model-View-Controller (MVC) architectural pattern, a decision that promotes separation of concerns, thereby enhancing modularity and facilitating easier maintenance and testing.

- **Model:** This layer manages the application's data and business logic. It defines the data structures for floor plan elements (walls, furniture, etc.), and it encapsulates the logic for manipulating these elements, such as adding, deleting, or modifying their properties.
- **View:** The View layer is responsible for rendering the graphical user interface (GUI). It observes the Model for any changes and updates the UI accordingly. This layer is built using Java Swing, taking advantage of its rich set of components and flexibility to create a highly interactive and intuitive design interface.
- **Controller:** Acting as the intermediary between the Model and View, the Controller layer handles user input, processes it (possibly updating the Model), and updates the View. It contains the logic for user actions, such as dragging to draw walls, selecting elements, and applying transformations like resizing or rotating.

### Component Design

The application comprises several key components, each designed to handle specific aspects of the floor plan design process:

- **Canvas Component:** At the heart of the application is the Canvas, a dedicated area where users create and interact with their floor plan. It manages the rendering of shapes and handles user inputs like clicks and drags.
- **Shape Components:** The fundamental building blocks of floor plans are the Shape components, including lines (for walls), rectangles (for rooms or furniture),

and other geometric shapes. These components are designed with extensibility in mind, allowing for easy addition of new shapes in the future.

- **Toolbar and Tool Components:** The Toolbar provides users with quick access to tools for drawing, selecting, and manipulating elements within the floor plan. Each tool in the Toolbar is a separate component, designed to perform specific actions, ensuring modularity and ease of extension.
- **Property Panel Component:** This component displays and allows editing of the properties of selected elements, such as dimensions, color, and other customizable attributes. It dynamically updates based on the current selection, providing users with contextual controls.

## Design Patterns

Several design patterns are employed throughout the application to solve common design problems and enhance the code's modularity and flexibility:

- **Factory Pattern:** Used for creating Shape objects, the Factory Pattern allows for the instantiation of shapes based on specified parameters, facilitating the addition of new shape types without modifying existing code.
- **Observer Pattern:** To maintain synchronization between the Model and View, the Observer Pattern is used. The View layers register as observers to the Model, ensuring they are updated whenever the Model changes, promoting a reactive UI.
- **Composite Pattern:** For handling complex designs involving grouped elements, the Composite Pattern treats single and composite elements uniformly. This allows users to manipulate grouped elements as if they were a single entity, simplifying operations like moving or resizing.



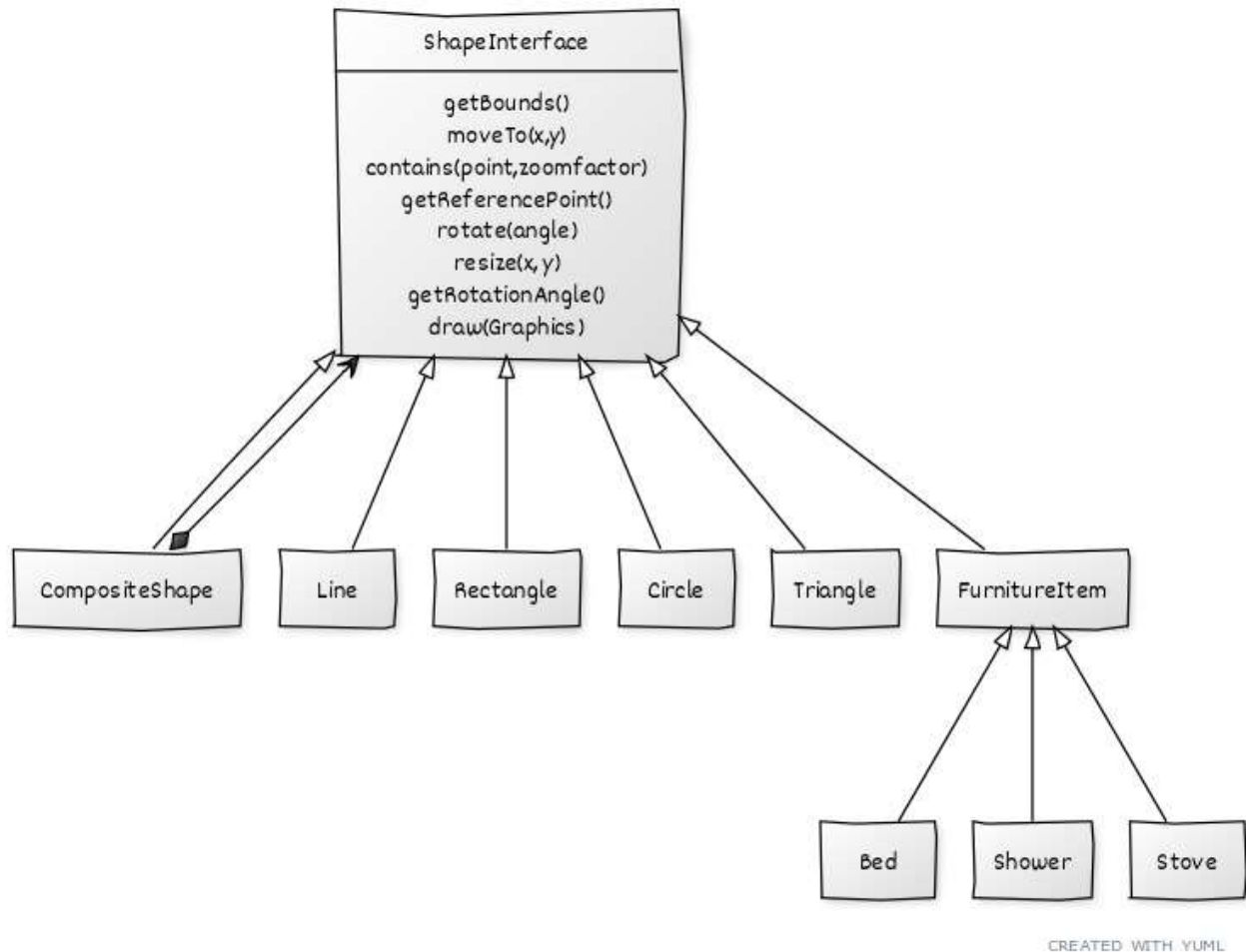


Figure 5. UML Diagram.

## Chapter 4: User Manual

### Getting Started

#### Opening the Application

Upon launching, the application will present a welcoming interface, featuring a menu bar at the top, a toolbar with design tools on the side, and a blank canvas in the center where your designs will take shape.

### Basic Operations

#### Creating Walls

1. **Select the Wall Tool:** Click on the wall icon in the toolbar to activate the wall drawing tool.
2. **Draw a Wall:** Click and drag on the canvas where you want the wall to start and end. Release the mouse button to finalize the wall segment.

## Creating Rooms

1. **Select the Room Tool:** Click on the room icon in the toolbar to activate the room drawing tool.
2. **Draw a Room:** Click and drag on the canvas where you want the room to start and end. Release the mouse button to finalize the room segment.

## Creating Opening

1. **Select the Opening Tool:** Click on the opening icon in the toolbar to activate the opening drawing tool.
2. **Draw a Room:** Click and drag on the canvas where you want the opening to start and end. Release the mouse button to finalize the opening segment.

## Adding Furniture

1. **Choose Furniture:** Access the furniture library by clicking the furniture icon in the toolbar. A list of available furniture items will appear.
2. **Place Furniture:** Drag and drop your chosen furniture item onto the canvas. Position it as desired by clicking and dragging.

## Customizing Elements

- **Select an Element:** Click on any element on the canvas to select it. A bounding box with handles will appear around it.
- **Rotate:** Use the rotation handle (usually at the top of the selection box) to rotate the element.
- **Move:** Click on any element and move it using the move tool.
- **Delete:** Drag and delete everything in the rectangle.
- **Line thickness:** Choose the desired thickness.
- **Select Tool:** Drag and select everything in the rectangle. You can now move it as a single object, until something else is selected with the tool.

## Advanced Features

### Zooming In Where the Mouse Is

1. **Activating Zoom:** Use mouse wheel.
2. **Zooming In:** To zoom in on a specific area, position your mouse cursor over the desired point on the canvas. Scroll the mouse wheel forward. The view will zoom in centered on the cursor's location.

3. **Zooming Out:** To zoom out, position the cursor and scroll the mouse wheel backward.

### Selecting Room or Wall Objects to View Dimensions

1. **Selecting an Object:** Click on a room or wall object to select it. A bounding box or selection outline will appear around the selected object.
2. **Viewing Dimensions:** With the object selected, look for dimensions around the object.

### Saving Your Design

1. **Accessing the Save Function:** Click on the 'Save' button at the top of the application window.
2. **Choosing a Save Location:** A dialog box will appear prompting you to choose a location on your computer to save the file. Enter a name for your file and select the desired save location.

### Loading a Previously Saved Design

1. **Accessing the Load Function:** Click on the 'Load' button at the top of the application window.
2. **Finding Your File:** In the dialog box that appears, navigate to the location where your design file is saved. Select the file you wish to open.
3. **Opening the File:** With the file selected, click 'Open' to load your design onto the canvas. Your previously saved work will appear, ready for further editing or review.

## Appendix

### ChatGPT Logs

Logs can be found in the ChatGPTLog Folder.

0215\_ChatGPTLog.mhtml

0216\_ChatGPTLog.mhtml

0229\_ChatGPTLog.mhtml

0301\_ChatGPTLog.mhtml

0306\_ChatGPTLog.mhtml

0307\_ChatGPTLog.mhtml