**Concept Document for WYSIWYG Configurator Application**

**Purpose of the Document**

This document provides a comprehensive overview of the WYSIWYG Configurator application, outlining its design, implementation, and alignment with client requirements. It highlights:

1. The deliverables completed across all days, including drag-and-drop functionality, element selection, dynamic property editing, and advanced customization features.
2. The modular architecture that ensures scalability and maintainability.
3. Key classes, properties, and methods with their purpose and interactions.

This document aims to summarize the functionality implemented and confirm that all deliverables have been met as per the proposal and client requirements.

**Key Features Delivered**

1. **Drag-and-Drop Functionality**:
   * Implemented for Rectangle, Ellipse, and TextBlock elements.
   * Elements can be dragged from the Toolbox and dropped into the Design Area.
2. **Element Selection**:
   * A visual overlay appears when an element is selected.
   * Selection highlights the element and enables property modification.
3. **Dynamic Property Editing**:
   * Properties like width, height, text, font size, color, and opacity can be modified dynamically.
   * Property changes are immediately reflected in the Design Area.
4. **Advanced Features**:
   * Real-time updates to properties for selected elements.
   * Error handling for invalid inputs, such as invalid colors or out-of-range opacity values.
   * Modularized code structure for extensibility.

**List of Classes, Enums, Properties, Methods, and Exceptions**

**1. MainWindow**

* **Description**: Orchestrates interactions between the Toolbox, Design Area, and Property Box.
* **Properties**:
  + \_selectedElement: Tracks the currently selected element in the Design Area.
  + \_selectionOverlay: A Rectangle used as a visual overlay to indicate selection.
  + Toolbox: A ListBox for managing draggable items.
  + DesignArea: A Canvas serving as the workspace for elements.
  + PropertyBox: A StackPanel dynamically populated with editable property fields.
* **Methods**:
  + Toolbox\_PreviewMouseDown: Initiates drag-and-drop operations from the Toolbox.
  + DesignArea\_Drop: Processes dropped elements and adds them to the Design Area.
  + SelectElement: Manages element selection and updates the visual overlay and Property Box.
  + UpdatePropertyBox: Dynamically populates the Property Box with fields for the selected element.
  + AddPropertyField: Adds dynamic fields to edit properties such as color, opacity, etc.

**2. ToolboxContainer**

* **Description**: Manages tools and their drag-and-drop functionality.
* **Properties**:
  + Items: A collection of available tools (Rectangle, Ellipse, TextBlock).
* **Methods**:
  + StartDrag: Initiates a drag operation for a selected tool.

**3. DesignAreaContainer**

* **Description**: Serves as the workspace where elements are placed, manipulated, and arranged.
* **Properties**:
  + \_elements: A collection of all objects in the Design Area.
  + \_selectedElement: Tracks the currently selected object.
  + \_selectionOverlay: A Rectangle that visually indicates the selected element.
* **Methods**:
  + AddElement: Adds elements (Rectangle, Ellipse, or TextBlock) to the Design Area.
  + SelectElement: Updates the visual overlay for the selected element.

**4. PropertyContainer**

* **Description**: Displays and manages the editable properties of the selected object.
* **Methods**:
  + DisplayProperties: Populates the Property Box with property fields.
  + AddPropertyField: Adds fields for editing properties.
  + AddColorPicker: Enables color editing for elements.
  + AddOpacityProperty: Enables opacity editing for elements.

**Purpose of Each Method**

1. **Toolbox\_PreviewMouseDown**: Initiates drag-and-drop operations from the Toolbox.
2. **DesignArea\_Drop**: Adds dropped elements into the Design Area.
3. **SelectElement**: Highlights the selected element and updates the Property Box.
4. **UpdatePropertyBox**: Displays editable properties for the selected element.
5. **AddPropertyField, AddColorPicker, and AddOpacityProperty**: Allow modification of various properties dynamically.

**Detailed Class Interactions**

1. **ToolboxContainer to DesignAreaContainer**:
   * Drag-and-drop operations are initiated from the Toolbox and processed in the Design Area.
2. **DesignAreaContainer to PropertyContainer**:
   * Properties of selected elements are sent to the Property Box for editing.
3. **PropertyContainer to DesignAreaContainer**:
   * Updates made in the Property Box are reflected in the Design Area in real-time.

**Extensibility and Future Plans**

* **Support for Additional Elements**:
  + The current architecture allows easy addition of new elements to the Toolbox.
* **Enhanced Alignment Features**:
  + Snapping and alignment tools can be integrated into the Design Area.
* **Export and Save**:
  + Saving the Design Area layout to a file format for future use.
* **Improved Property Management**:
  + Adding advanced data validation and property grouping for better user experience.

**Conclusion**

The WYSIWYG Configurator application is now fully functional, meeting all client requirements across all five days of deliverables. Key features such as drag-and-drop, element selection, property editing, and dynamic updates have been implemented successfully. The modular design ensures scalability and extensibility for future enhancements.

This application is ready for client review and deployment.