**JabberPoint Problem Report**

*Prepared by Evald Narkevicius (ID: 4981499)*

## Issue 1: "Go to Page" Functionality

**Description:** Users can navigate to any page using the "view" -> "go to" option, even if the page is not part of the presentation. This allows selection of pages outside the presentation's bounds.

**Solution:** Implement a check in the page navigation method to restrict users from accessing pages beyond the presentation's limits. This ensures an error message is displayed if an out-of-bound page is selected.

## Issue 2: LoadDemoPresentation Method and DemoPresentation Class

**Description:** The code for LoadDemoPresentation method and DemoPresentation class is excessive and completely unnecessary, considering the application can load presentations. My suggestion is creating an XML file of the demo presentation in the project directory and accessing it by default instead of generating it every time on launch.

**Solution:** Remove unnecessary methods, save the presentation as an XML file, and update the code to launch the file by default. This will decrease the bloat of the application.

## Issue 3: Double Relationship Between Presentation and SlideViewerComponent

**Description:** Between Presentation and SlideViewerComponent there is a double relationship, making testing difficult and resulting in messy code.

**Solution:** Implement an observer design pattern in the Presentation class to streamline the relationship and improve code organization, testability and maintainability.

## Issue 4: Font Creation Inside SlideViewerComponent

**Description:** SlideViewerComponent should not be responsible for font creation, which adds unnecessary tasks inside the class.

**Solution:** Introduce a separate class to handle font creation, reducing the responsibilities of SlideViewerComponent.

## Issue 5: MenuController and XMLAccessor Complexity

**Description:** MenuController and XMLAccessor have giant methods that are difficult to read and test and have an immense amount of responsibilities.

**Solution:** Break down large methods into smaller, more manageable ones to improve readability and testability.

## Issue 6: System.exit(n) Inside Presentation

**Description:** System.exit(n) is called from a different class, leading to unnecessary complexity.

**Solution:** Remove the class and directly call System.exit(n) from MenuController to simplify the code.

## Issue 7: AboutBox Direct Message Insertion

**Description:** AboutBox.show() directly inserts message and title into the text box, reducing readability and maintainability.

**Solution:** Introduce constants for the title and message to enhance code readability and maintainability.