Project Report

Music Notation Editor Project Report

1. **Introduction**
    - The Music Notation Editor project aims to develop a user-friendly application for displaying and editing musical notes on a staff using Java Swing.
2. **Features**
    - **Musical Staff Display**: Implemented a graphical panel to display a musical staff using Java Swing components.
    - **Note Selection**: Created toolbar buttons for selecting different note durations (whole, half, and quarter notes).
    - **Note Placement and Editing**: Implemented functionality for placing, moving, and removing notes on the staff panel.
    - **Playback Functionality**: A simple play button that interprets the notes on the staff and plays back the melody using a basic synthesized sound.
3. **Challenges**
    - Designing and implementing the graphical representation of musical notes and staff.
    - Handling user interactions for placing, moving and deleting notes with mouse events.
    - Ensuring responsiveness and smooth performance of the application, especially when handling multiple notes on the staff panel.
4. **Future Improvements**
    - Implement saving and loading functionality to allow users to save and load projects.
    - Enhance the user interface with additional features such as note customization options, playback functionality, and exporting options.
    - Improve error handling and user feedback mechanisms to provide a more seamless user experience.
5. **Conclusion**
    - The Music Notation Editor project has provided a foundation for creating a simple yet functional application for displaying and editing musical notes. Further development and refinement will enhance its usability and feature set.

User Manual

Music Notation Editor User Manual

1. **Introduction**
    - Welcome to the Music Notation Editor! This user manual will guide you through the features and functionalities of our application.
2. **Getting Started**
    - To begin using the Music Notation Editor, simply launch the application by double-clicking the executable file.
    - Upon opening, you will see a graphical user interface consisting of a staff panel and note selection buttons.

3. **Features**
    - **Staff Panel**: This panel displays a musical staff where you can place, move, and remove notes.
    - **Note Selection Buttons**: Use these buttons to select different note durations (whole, half, and quarter notes) before placing them on the staff.
    - **Placing Notes**: Click on the desired note duration button, then click on the staff panel to place the note.
    - **Moving Notes**: Drag existing notes on the staff panel to move them to a new position.
    - **Removing Notes**: To remove a note, simply click on the existing note on a five-line staff panel.
4. **Saving and Loading**
    - The Music Notation Editor does not currently support saving and loading projects. All changes made are temporary within the session.
5. **Troubleshooting**
    - If you encounter any issues or have questions about the application, please refer to the project documentation or contact our support team for assistance.
6. **Feedback**
    - We welcome your feedback and suggestions for improving the Music Notation Editor. Please feel free to contact us with any comments or feature requests.

Design Manual

Music Notation Editor Design Manual

1. **Architecture Overview**
    - The Music Notation Editor follows a Model-View-Controller (MVC) architecture.
    - **Model**: Represents the underlying data structure for musical notes and staff.
    - **View**: Displays the graphical user interface using Java Swing components.
    - **Controller**: Handles user interactions and updates the model accordingly.
2. **Class Structure**
    - **MusicNotationEditor**: Main class responsible for initializing the application and setting up the user interface components.
    - **StaffPanel**: Subclass of JPanel responsible for displaying the musical staff.
    - **NoteButton**: Subclass of JButton representing note selection buttons with icons and names.
    - **MouseAdapter**: Handles mouse events for placing and moving notes on the staff panel.
3. **Graphical Representation**
    - Musical notes are represented as graphical icons displayed on the staff panel.
    - The staff panel uses graphics primitives to draw horizontal lines representing the staff and note positions.
4. **User Interaction**
    - Users interact with the application by clicking on note selection buttons to choose note durations and clicking on the staff panel to place notes.

- Notes can be moved by clicking and dragging them to a new position on the staff panel.

5. **Performance Considerations**
    - The application should be responsive and performant, even when handling multiple notes and user interactions simultaneously.
    - Efficient data structures and event handling mechanisms should be employed to minimize latency and ensure smooth user experience.

Appendices
ChatGPT Logs for Shan He:

See html files under the report folder.