**Electron Music Player - Group Project Breakdown**

**Team Structure and Responsibilities**

**Member 1: Project Lead & Core Architecture**

* Set up the initial Electron project structure
* Implement folder selection mechanism
* Configure storage for remembering folder location
* Coordinate integration of components
* Manage GitHub repository setup
* Lead the documentation effort
* Tasks: ~30% of core functionality, 20% of documentation/testing

**Member 2: Media Playback Specialist**

* Develop the audio playback engine
* Implement playback controls (play/pause, skip, volume)
* Create progress bar with seek functionality
* Parse and display track information
* Tasks: ~40% of core functionality, 15% of documentation/testing

**Member 3: UI/UX Developer**

* Design and implement the user interface
* Create responsive layouts
* Implement theme support (light/dark)
* Design and implement playlist display
* Develop keyboard shortcuts
* Tasks: ~80% of user experience, 15% of documentation/testing

**Member 4: System Integration & QA**

* Implement system tray integration
* Add media key support
* Develop notification system
* Write unit and integration tests
* Ensure cross-platform compatibility
* Tasks: ~30% of core functionality, 20% of user experience, 50% of documentation/testing

**Detailed Task Breakdown**

**Member 1: Project Lead & Core Architecture**

1. **Week 1:**
   * Set up Electron project scaffold
   * Configure build system and dependencies
   * Implement folder selection dialog
2. **Week 2:**
   * Implement file system persistence
   * Create file parser for supported audio formats
   * Begin documentation framework
3. **Week 3:**
   * Integration support for other components
   * README and setup documentation
   * Technical report outline

**Member 2: Media Playback Specialist**

1. **Week 1:**
   * Research and select audio playback libraries
   * Implement basic audio playback functionality
   * Build play/pause and skip controls
2. **Week 2:**
   * Develop volume control with mute toggle
   * Create progress bar with seeking capability
   * Implement MP3/audio metadata extraction
3. **Week 3:**
   * Finalize playback engine
   * Document audio subsystem
   * Support integration testing

**Member 3: UI/UX Developer**

1. **Week 1:**
   * Create wireframes for application layout
   * Implement main application window
   * Design playlist view component
2. **Week 2:**
   * Develop theme switching capability
   * Implement responsive design
   * Create playback control UI elements
3. **Week 3:**
   * Add keyboard shortcut system
   * Polish UI components
   * Create user manual for interface features

**Member 4: System Integration & QA**

1. **Week 1:**
   * Set up testing framework
   * Create initial unit tests
   * Research system integration requirements
2. **Week 2:**
   * Implement system tray integration
   * Add media key support
   * Develop notification system
3. **Week 3:**
   * Conduct cross-platform testing
   * Document testing methodology
   * Complete end-to-end tests

**Integration Points**

To ensure the components work together seamlessly, I recommend:

1. **Daily Stand-ups**: Brief meetings to discuss progress and roadblocks
2. **Well-defined APIs**: Clear interfaces between components
3. **Shared Development Standards**: Consistent coding styles and patterns
4. **Regular Integration Testing**: Frequent merging and testing of components
5. **Documentation as You Go**: Document code and design decisions throughout

**Development Timeline**

* **Week 1**: Setup, basic infrastructure, and individual component development
* **Week 2**: Core functionality implementation and initial integration
* **Week 3**: Feature completion, testing, debugging, and documentation
* **Week 4**: Final integration, cross-platform testing, and project submission

**Required Technologies**

* **Electron**: For cross-platform desktop application
* **Node.js**: Backend runtime
* **HTML/CSS/JavaScript**: UI development
* **Howler.js or Web Audio API**: Audio playback
* **Electron Store**: Persistence between sessions
* **Jest/Spectron**: Testing framework
* **music-metadata**: For parsing audio file metadata

**Final Deliverables Checklist**

1. **Code Repository**:
   * Complete source code
   * Build instructions
   * Dependency documentation
2. **Documentation**:
   * Technical README
   * User manual
   * Code documentation
   * API documentation
3. **Technical Report**:
   * Implementation details
   * Challenges and solutions
   * Development reflections
   * Future improvements