**Additional Notes**

The first class was the PlayListTableModel. This had the properties that used the Constants from the PlayListControlsPanel class. That led to the discovery of the sortPopup object displaying the choices. We determined that we were going to follow the "Show Genres" pattern and put the Show Duration choice right after it

The PlayListController class posed a slightly different challenge. We needed to determine the column widths and the calculation for this. We also discovered the unary ("immediate if") statement was being used for coding. This was again a familiarity with C# comparison.

Followed to the PlayListControlsController to add a listener, then the semi-complex series of commands related to having a similar behavior for a column on the panel.

Because there was this new behavior we needed to add the preservation of state to the application.

Because during some earlier testing, one of us had the Atunes chinese language switch on and it took a while to figure out what was happening. Then it became obvious that the properties files had the constants for the languages.

Just for fun we used Google translate for the German, French, Spanish, and Italian languages. We saw that for the simplified and traditional Chinese properties file used the Unicode strings, but we were concerned that we were running out of time for completing the assignment, so we did not investigate that step further.

**Class references**

|  |
| --- |
| public class **PlayListTableModel** implements TableModel  private ArrayList<AudioFile> songs;  private ArrayList<TableModelListener> listeners;  private boolean trackVisible = true;  private boolean artistVisible = true;  private boolean albumVisible = true;  private boolean genreVisible = true;  private boolean durationVisible = true;  private enum PlayListColumn {FAVORITE, TRACK, TITLE, ARTIST, ALBUM, GENRE, DURATION}  private static PlayListColumn[] headers = {PlayListColumn.FAVORITE, PlayListColumn.TRACK, PlayListColumn.TITLE, PlayListColumn.ARTIST, PlayListColumn.ALBUM, PlayListColumn.GENRE, PlayListColumn.DURATION};  private PlayListColumn[] currentHeaders;  private static HashMap<PlayListColumn, Class> classes;  private static HashMap<PlayListColumn, String> columnNames;  public PlayListTableModel()  public void addTableModelListener(TableModelListener l)  private void setCurrentHeaders()  private PlayListColumn getColumn(int colIndex)  public Class<?> getColumnClass(int colIndex)  public int getColumnCount()  public String getColumnName(int colIndex)  public int getRowCount()  public Object getValueAt(int rowIndex, int colIndex)  public boolean isCellEditable(int rowIndex, int columnIndex)  public void removeTableModelListener(TableModelListener l)  public void setValueAt(Object aValue, int rowIndex, int columnIndex)  public void addSong(AudioFile song)  public void removeSongs(int[] rows)  public void removeSongs()  public void moveToTop(int[] rows)  public void moveUp(int[] rows)  public void moveDown(int[] rows)  public void moveToBottom(int[] rows)  public void refresh()  public void refresh(int pos)  public AudioFile getFileAt(int pos)  public void setTrackVisible(boolean trackVisible)  public void setArtistVisible(boolean artistVisible)  public void setAlbumVisible(boolean albumVisible)  public boolean isEmpty()  public boolean isAlbumVisible()  public boolean isArtistVisible()  public void setGenreVisible(boolean genreVisible)  public void setDurationVisible(boolean durationVisible)  public boolean isGenreVisible()  public boolean isDurationVisible()  public boolean isTrackVisible() |

|  |
| --- |
| public class **PlayListControlsPanel** extends JPanel {  private static final long serialVersionUID = -1966827894270243002L;  private PopUpButton sortPopup;  private JCheckBoxMenuItem showTrack;  private JCheckBoxMenuItem showArtist;  private JCheckBoxMenuItem showAlbum;  private JCheckBoxMenuItem showGenre;  private JCheckBoxMenuItem showDuration;  private JMenuItem sortByTrack;  private JMenuItem sortByTitle;  private JMenuItem sortByArtist;  private JMenuItem sortByAlbum;  private JMenuItem sortByGenre;  private JMenuItem sortByDuration;  private JButton savePlaylistButton;  private JButton loadPlaylistButton;  private JButton artistButton;  private JButton albumButton;  private JButton topButton;  private JButton upButton;  private JButton deleteButton;  private JButton downButton;  private JButton bottomButton;  private JButton infoButton;  private JButton clearButton;  private PopUpButton favoritePopup;  private JMenuItem favoriteSong;  private JMenuItem favoriteAlbum;  private JMenuItem favoriteArtist;  public PlayListControlsPanel()  private void addContent()  private void setButton(JButton button, GridBagConstraints c)  public JButton getBottomButton()  public JButton getDeleteButton()  public JButton getDownButton()  public JButton getInfoButton()  public JButton getTopButton()  public JButton getUpButton()  public JButton getClearButton()  public JButton getLoadPlaylistButton()  public JButton getSavePlaylistButton()  public JMenuItem getFavoriteAlbum()  public JMenuItem getFavoriteArtist()  public JMenuItem getFavoriteSong()  public PopUpButton getFavoritePopup()  public JMenuItem getSortByTitle()  public JMenuItem getSortByAlbum()  public JMenuItem getSortByArtist()  public JCheckBoxMenuItem getShowTrack()  public JCheckBoxMenuItem getShowArtist()  public JCheckBoxMenuItem getShowAlbum()  public JButton getAlbumButton()  public JButton getArtistButton()  public JMenuItem getSortByTrack()  public JMenuItem getSortByGenre()  public JMenuItem getSortByDuration()  public JCheckBoxMenuItem getShowGenre()  public JCheckBoxMenuItem getShowDuration() |

|  |  |
| --- | --- |
| public class **PlayListController** extends PanelController implements PlayListColumnClickedListener  public PlayListController(PlayListPanel panel)  public void finish()  protected void addBindings()  protected void addStateBindings()  protected void notifyReload() {}  public void columnClicked(String column)  public void updatePositionInTable(int pos)  public void addSongsToPlayList(ArrayList songs, int selectedSong)  public void setSelectedSong(int song)  private void setSelectedSongOnTable(int song)  public void adjustColumnsWidth()  public void playSelectedSong()  public void setTrackNumber()  public void setGenre()  public void moveToTop()  public void moveUp()  public void moveDown()  public void moveToBottom()  public void deleteSelection()  public void setAsFavoriteSongs(ArrayList<AudioFile> songs)  public void setAsFavoriteAlbums(ArrayList<AudioFile> songs)  public void setAsFavoriteArtists(ArrayList<AudioFile> songs)  public void disablePlayListItems(boolean disable)  private ArrayList<AudioFile> getSelectedSongs(JTable table)  public ArrayList<AudioFile> getSelectedSongs()  public JScrollPane getMainPlayListScrollPane()  public PlayListTable getMainPlayListTable()  public void setArtistAsPlaylist()  public void setAlbumAsPlaylist()  public void showPlaylistControls(boolean show) |  |

|  |
| --- |
| public class **PlayerControlsController** extends PanelController  public PlayerControlsController(PlayerControlsPanel panel)  protected void addBindings()  protected void addStateBindings()  protected void notifyReload()  public void setTime(long time, long totalTime)  public void setMaxDuration(long maxDuration)  protected void setVolume(int value)  public void setShuffle(boolean shuffle)  public void setRepeat(boolean repeat)  public void setMute(boolean mute)  public void setPlaying(boolean playing) |

|  |
| --- |
| public class **PlayerControlsListener** extends MouseAdapter implements ActionListener, ChangeListener  private PlayerControlsPanel panel;  private PlayerControlsController controller;  protected PlayerControlsListener(PlayerControlsPanel panel, PlayerControlsController controller)  public void mouseClicked(MouseEvent e)  public void mouseReleased(MouseEvent e)  public void stateChanged(ChangeEvent e)  public void actionPerformed(ActionEvent e) |

|  |
| --- |
| public class **ApplicationState** implements Serializable  private static final long serialVersionUID = 4771296638947290335L;  private boolean showNavigationPanel = true;  private boolean showNavigationTable = true;  private boolean showSongProperties;  private boolean showStatusBar = true;  private boolean showOSD;  private boolean shuffle;  private boolean repeat;  private boolean sortDeviceByTag;  private boolean showSystemTray;  private boolean showTrayPlayer;  private int navigationView;  private String language;  private Search defaultSearch;  private boolean useAudioScrobbler = true;  private boolean multipleWindow;  private int version;  private boolean showPlaylistControls = true;  private ProxyBean proxy;  private boolean showTrackInPlayList = true;  private boolean showArtistInPlayList = true;  private boolean showAlbumInPlayList = true;  private boolean showGenreInPlayList = true;  private boolean showDurationInPlayList = true;  private String theme = LookAndFeelSelector.DEFAULT\_THEME;  // Constant properties editable in edit preferences  public boolean SHOW\_TITLE = true;  public int OSD\_DURATION = 2; // In seconds  public int AUTO\_REPOSITORY\_REFRESH\_TIME = 60; // In minutes  public boolean ANIMATE\_OSD = true;  public boolean SHOW\_FAVORITES\_IN\_NAVIGATOR = true;  public boolean SAVE\_PICTURE\_FROM\_AUDIO\_SCROBBLER = true;  public boolean SHOW\_ALBUM\_TOOLTIP = true;  public int ALBUM\_TOOLTIP\_DELAY = 1; // In seconds  // Next attributes are not always consistent since are changed when storing state  private Point windowLocation;  private boolean maximized;  private Dimension windowSize;  private Point multipleViewLocation;  private Dimension mutipleViewSize;  private int volume = 50;  public ApplicationState()  public boolean isMaximized()  public void setMaximized(boolean maximized)  public boolean isRepeat()  public void setRepeat(boolean repeat)  public boolean isShowNavigationTable()  public void setShowNavigationTable(boolean showNavigationTable)  public boolean isShowOSD()  public void setShowOSD(boolean showOSD)  public boolean isShowSongProperties()  public void setShowSongProperties(boolean showSongProperties)  public boolean isShuffle()  public void setShuffle(boolean shuffle)  public boolean isSortDeviceByTag()  public void setSortDeviceByTag(boolean sortDeviceByTag)  public boolean isShowSystemTray()  public void setShowSystemTray(boolean showSystemTray)  public boolean isShowTrayPlayer()  public void setShowTrayPlayer(boolean showTrayPlayer)  public Point getWindowLocation()  public void setWindowLocation(Point windowLocation)  public int isNavigationView()  public void setNavigationView(int navigationView)  public int getNavigationView()  public String getLanguage()  public void setLanguage(String language)  public Search getDefaultSearch()  public void setDefaultSearch(Search defaultSearch)  public boolean isShowNavigationPanel()  public void setShowNavigationPanel(boolean showNavigationPanel)  public boolean isMultipleWindow()  public void setMultipleWindow(boolean multipleWindow)  public int getVersion()  public Dimension getWindowSize()  public void setWindowSize(Dimension windowSize)  public boolean isShowPlaylistControls()  public void setShowPlaylistControls(boolean showPlaylistControls)  public boolean isUseAudioScrobbler()  public void setUseAudioScrobbler(boolean useAudioScrobbler)  public ProxyBean getProxy()  public void setProxy(ProxyBean proxy)  public boolean isShowStatusBar()  public void setShowStatusBar(boolean showStatusBar)  public Point getMultipleViewLocation()  public Dimension getMutipleViewSize()  public void setMultipleViewLocation(Point multipleViewLocation)  public void setMutipleViewSize(Dimension mutipleViewSize)  public boolean isShowAlbumInPlayList()  public void setShowAlbumInPlayList(boolean showAlbumInPlayList)  public boolean isShowArtistInPlayList()  public void setShowArtistInPlayList(boolean showArtistInPlayList)  public String getTheme()  public void setTheme(String theme)  public boolean isShowGenreInPlayList()  public boolean isShowDurationInPlayList()  public void setShowGenreInPlayList(boolean showGenreInPlayList)  public void setShowDurationInPlayList(boolean showDurationInPlayList)  public int getVolume()  public void setVolume(int volume)  public boolean isShowTrackInPlayList()  public void setShowTrackInPlayList(boolean showTrackInPlayList) |