This diagram shows the overall software architecture of our software. From the outside the client sees only the user interface. The interface then uses a global settings class that handles the data transfer from the UI to the “backend.” The UI has been deisgned to use the MVP archetecture in order to allow extensibility. As a note, the SettingsDialog has many tabs even though we only have one tab for now. After release one, we plan on adding more functionality, through espn and twitter, that will require more tabs. This is to allow for extensibility. The backend code, (Audio Control) handles the audio interface exposed through Java to use the picked devices and play audio through them. The AudioControl also handles any of the audio caching that needs being done.

Class: Audio Control

Handles javax.sound api to obtain devices and play from them

<<Not exposed>>

Class: Settings

Acts as model for MVP

<<Exposed through settings page but not explicitly changeable>>

Client

Inherits

Has Many

Interface,MVP: ITab

Is a tab

**MVP: SettingsDialog**

Holds a list of iTabs and hanldes the saving of settings and the updating of the UI afterword

MVP: AudioSettingsTab

Handles the settings for Audio

**MVP: Application**

Handles overall execution and main page UI

Application

SWT User Interface

MVP architecture, All MVP objects have a view and a presenter. The view handles the display, presenter handles all business logic