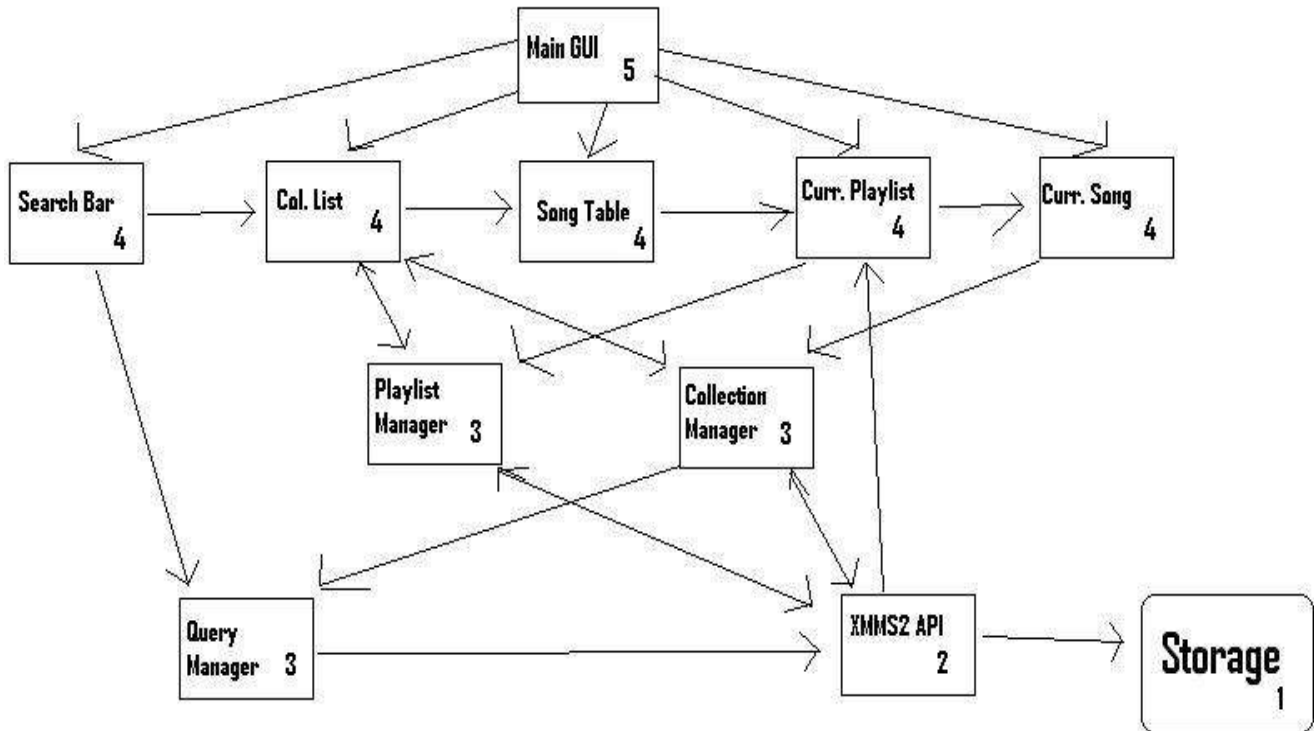


XMMS2 Client Top-Level Design

I. Levelized High-Level Component Diagram



II. External Dependencies

A. XMMS2

1. XMMS2 (X-platform Music Multiplexing System 2) is a newly designed product that plays audio files. It is a redesign of the XMMS audio player.
2. Stable but not bug free. Features change slightly with each build.
3. Client must support multiple instances.
4. Client must integrate well with XMMS2 API.

B. QT

1. QT is a cross-platform development framework commonly used to develop GUI-related software.
2. Uses “signals” and “slots” – a callback system that will easily allow the project to be broken into independent parts.
3. Because QT bindings for XMMS2 not yet available, use of QT will require interfacing through the standard C++ XMMS2 API.

III. Task Breakdown

GUI: Implement, using whatever necessary C++ libraries or toolkits, a smooth UI that follows requirements set forth in specification document.

Back End: Non-GUI related coding in C++ necessary for communicating with XMMS2 API.

Project Manager: Ensure everything is running on schedule and smoothly. Set up meetings. Send regular e-mails. Main contact for information on project status.

Documentation: Responsible for design-related documents (i.e. schematics). Also responsible for drafting non-code-related documents (i.e. user manuals, helpdocs, READMEs, etc...).

Tools: Install and maintain necessary tools (i.e. QT). Also responsible for helping other members with using tools.

Testing: Responsible for ensuring code is well integrated and works smoothly.

Design: Responsible for architectural design, and designing how system communicates with the XMMS2 software.

IV. Group Organization

dlatty – PM, Documentation
ngaylinn – Tools, GUI
bhickey – Back End
colin – Testing
jgan – GUI, Design
kpthomas – GUI, Documentation
lquirk – Testing
ostrain – Back End
sms – Back End
tolson – Back End

V. Schedule

February 26:	Pick Project
March 2:	Finalize Top-Level Design
March 10:	Finalize Interface, Low-Level Design
March 15:	Begin Coding
April 15:	Major Coding Completed
April 15-25:	Code Integration and Testing
April 27:	In-Class Demo
May 7:	Public Demo
May 17:	Final Demo Day

VI. Assumptions Made From Specifications

None