Term Project Design

Overview

This document describes the final design for a video file query preprocessor as a front end to a multimedia database. Sufficient information (we hope) is presented to suffice as a coding specification.

Background

Development Environment
We have selected the Java 2 Standard Edition (J2SE) as the development environment. The following application programmer interfaces (APIs) will be used:

- Swing for graphic user interface (GUI) implementation.
- Java Database Connection (JDBC) for relational database table manipulation.
- Java Advanced Imaging (JAI) for saving still images.
- Java Media Framework (JMF) for inspecting video clips.

Of these, the last two (JAI and JMF) are optional components that must be installed in addition to the J2SE development kit.

Run Time Interfaces
Since this project makes use of JDBC, a relational database engine and appropriate JDBC driver must be installed to make use of the finished project. A web server, such as Apache, is optional but not required.

Stolen Code
We used two sample applications available on Sun's web site as example Java code. Those sample applications are:

- FrameAccess.java, showing how to use the JMF to play a video and access individual video frames.
- AWTImageSample.java, showing how to encode an image managed by the Java AWT graphics library with the JAI for subsequent writing to disk.

References
Some JDBC code was developed following examples provided by the book, “Teach Yourself Java 2 In 21 Days,” Second Professional Reference Edition. See specifically the code on Page 279 and the examples for Day 11.

GUI Screen Design

The following paragraphs govern the various GUI screens by defining the name and type of all parameters collected for operation or presented for display.

Login Screen
The main screen for connecting to the database via JDBC will accept four parameters:, driver name, URL, user name, and password. These parameters are necessary to establish a Connection object.

The program will display this screen at start up. If connection is successful, the following screen will be displayed. Otherwise an error message will be presented.

Session Parameters Screen
The second screen will accept control parameters that will be used as defaults for the session. These parameters are as follows:

- Source video clip directory: directory where video clips for logging are stored.
- Output image and report directory: directory for writing still images and query reports.
- Samples per video: number of stills to store per video. Seconds per sample will be adjusted based on this parameter and the video duration.
- Initial sample delay: initial position of video. Used to skip blank leader.

The directories will be stored as strings and are required to be in URL format. The other parameters will be stored as double precision floating point numbers.

Clip Ingest Screen
The following parameters will be populated based on calls to the JMF:

- Video format: a string showing the encoding method, frame size, frames per second, and total frames.
- Duration: the clip length in seconds.

The following parameters will be echoed or derived from session parameters:

- Source directory.
- Output directory.
- Initial sample delay.
- Seconds per sample: computed by dividing the total duration by the session parameter of samples per video.

The following parameters will be supplied by the user:

- Clip file name: used with the source directory to load the video from disk.
- Clip title: does not have to match the file name.
- Comments: free form text input.

Database Table Design

In our proof of concept, we discussed the need for two relational database tables: one to store clip related data, and one to store data for each still extracted froma clip. Here is the schema for both of these tables:

Clips Table
The following data items will be collected for each clip in the library:

Column Name	***Description***	***Data Type***
ID	Primary key, auto number format.	Integer
SourceDir	Directory where the original source clip is stored.	String
SourceName	Video clip file name.	String
DestDir	Directory for storing stills.	String
Parameters	Video encoding parameters reported by JMF.	String
Duration	Length of clip in seconds reported by the JMF	Double
Title	User supplied name.	String
Comments	User supplied description.	String

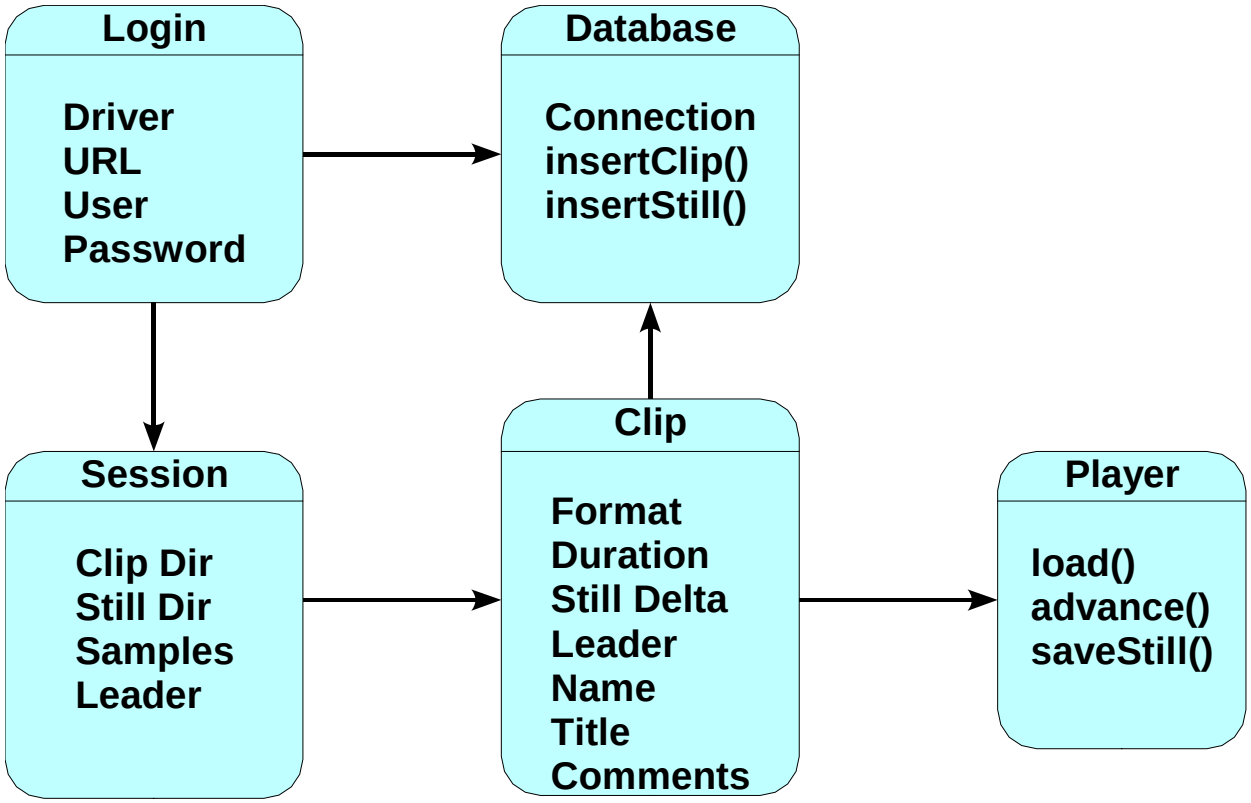
Stills Table
The following data items will be collected for each still from a clip in the library:

Column Name	***Description***	***Data Type***
ID	Primary key, auto number format.	Integer
ClipID	Foreign key into Clip table, 1-M.	Integer
Position	Position from start of clip in seconds from where this still was extracted.	Double
Name	Name of clip, automatically generated by the system by combining the clip SourceName with the Position.	String

Objects

Class Model

The following diagram represents a static model for the custom objects created fro the project. Obviously Java supplies several API objects, but these are not diagrammed:



Sequence Diagram

The following event trace shows the control logic between objects for loading of a video clip and storing stills: