



BackCast Fly Inventory Tool

Project Requirements

Project:	BackCast
Author:	Michael Hodgson
Version:	0.1
Date:	November 5, 2008
Status:	First Draft

Contents

1.0	Introduction	3
2.0	Document Purpose	3
3.0	Application Overview	3
3.1	Objectives	3
4.0	Project Scope	4
4.1	Out of Scope	4
4.2	Project Phases	4
4.3	Source Code	4
5.0	User Profile	5
5.1	Demographic Information	5
5.2	User Goals	5
6.0	Artefacts and Content	5
7.0	Use Cases	6
7.1	Use case 1 – View pattern	6
7.2	Use case 2 – Searching	6
7.3	Use case 3 – Add new pattern (or edit existing one)	7
7.4	Use case 4 – Download pattern(s) from the server	7
8.0	Features	7
8.1	Pattern Search	7
8.2	View Pattern	8
8.3	Play Media	8
8.4	Create Pattern	8
8.5	Print Pattern	8
8.6	Server Connection	8
8.7	Search Server Patterns	8
8.8	Download Patterns	8
9.0	Application Panel Layout	9
10.0	Database design	9

1.0 Introduction

Recently Adobe has released a new application environment called AIR (Adobe Integrated Runtime) which allows developers/designers to build desktop applications using Action Script (the scripting language of Flex). This means that a developer should be able to build a sexy interface without having to develop piles of library code.

Since I learn best by doing, I have decided to pick a large(ish) project with many different interface aspects and attempt to build something useful (sort of). Having thought about all of the possibilities, and made up a couple of new ones, I decided that I would go with a Fly Tying Inventory tool, code named BackCast.

In this case the application is not the important thing, but rather it is a vehicle for the learning of the software. As such the application must have elements that lend themselves to multimedia (video, text, graphics and audio), offline capabilities (such as a local database), an engaging user interface, etc. Basically the application will be a storage and search tool for a specific type of multimedia.

2.0 Document Purpose

The Requirements Document establishes the operational framework and performance baseline for the BackCast software. This document shall be considered the baseline for the implemented requirements of the system and will specify the needs for both the client and server side components as well as how they will interact.

Evaluation of criteria that are not detailed in this document is beyond the scope of the software project and will not be used when evaluating the success of the proof.

3.0 Application Overview

3.1 Objectives

The following is a list of the high level functionality that will be demonstrated in the proof of concept:

- Demonstrate how Adobe AIR can be used to provide an engaging user experience
- Should make use of some of AIR's special talents - otherwise why build it in AIR?
 - online/offline access - this implies that there is at least some kind of a back end system
A central database of patterns that can be used to update the local software will be very useful
 - local database
 - have update capabilities for new versions of the software
 - platform independence

- multimedia presentation

4.0 Project Scope

The BackCast proof of concept will include:

- An engaging user interface that would include:
 - Display of pattern information
 - Access to pattern multimedia elements
 - Search capabilities to locate patterns
 - Creation of new patterns including the import of multimedia elements
- A local database of all multimedia elements and pattern data
- A server side database of patterns that may be added to the local database upon request

4.1 Out of Scope

The following functionality is beyond the scope of the current proof of concept

- User login and security
- Virus checking on uploaded files
- Multimedia manipulation (editing of multimedia files)
- Integration to other existing backend systems – security, printing services, etc.

4.2 Project Phases

The proof described in this document will be delivered in one phase. If other requirements are appended to the project they may be released in subsequent phases.

4.3 Source Code

All source code will be made available to the public for the purpose of evaluation, experimentation and general knowledge sharing. Code will be considered the property of Adobe Systems, Inc. with no warranty or guarantee what so ever.

5.0 User Profile

5.1 Demographic Information *

Age	30-50 Years (median age – 46.6 years)
Income	\$84, 600 average individual \$113,000 average household
Sex	84% Male
Employment	Professional, managerial and administrative
Education	82% attended/graduated college 41% attended post graduate program
Family Status	78.1% Married

5.2 User Goals

The user's main goal, as it pertains to this project, is to quickly find a desired pattern that can be easily read and understood. Finding the desired fly requires a degree of searching capability based on:

- Fish species - salmon, trout, bass, etc.
- Fly tying materials – material type, color, size, etc.
- Fly type or application – streamer, dry fly, terrestrial, etc.

Secondary goals include the ability to store new patterns so they can be quickly found and to obtain new patterns for a specific use.

6.0 Artefacts and Content

Central to the tool is the concept of a fly tying pattern (or just the “pattern”). The pattern consists of an image of the fly and a recipe. The recipe consists of several part names and the materials used to create the part. In general, the order of recipe items is very important as it indicates the order in which the materials are applied to the hook. Patterns may also include the tier's name, the name of the files originator, some notes about tying or fishing the fly and perhaps other images or videos the technique for tying the fly.

* Business Plans That Work By Joan Gillman, Sarah White (based on 1990 National Sports and Recreation Study)

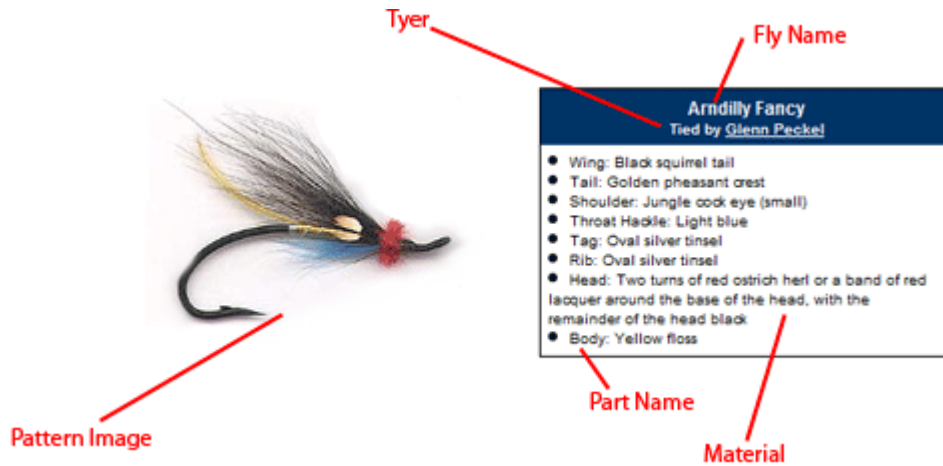


Figure 1: Pattern Example

Most users find patterns in pattern books or magazines where a fairly standard format exists (although the exact layout may differ). Other sources of information, including the web pages, stick to this standard format but may add additional multimedia elements.

7.0 Use Cases

The following use case is a generalization and is only meant to illustrate the user interaction with the software. Other user scenarios may be used for evaluation as long as they test the requirements stated in this document.

7.1 Use case 1 – View pattern

1. The user launches the application
2. User is presented with a list of available patterns
3. User selects a pattern from the list
4. Complete pattern is shown on the screen
 - Multimedia is displayed
 - Pattern parts are listed

7.2 Use case 2 – Searching

1. The user launches the application
2. User is presented with several possible search options – such as species, type and material
3. User can combine searches (for example a specific species and material)
4. List of available pattern narrows to those that match the search

7.3 Use case 3 – Add new pattern (or edit existing one)

1. The user launches the application
2. User selects to add a new fly pattern
3. User can add pattern multimedia elements
4. User can enter pattern data (such as the name)
5. User can select the relevant search tags and pattern details (species, type and material)
6. User can save the pattern to their local database for later use

7.4 Use case 4 – Download pattern(s) from the server

1. The user launches the application
2. User selects to download new patterns from the server
3. Software checks to see if user is online and if a connection can be made to the server
4. User is presented with a list of available patterns, patterns that are in local database are tagged with an identifier
5. User can search for patterns as in Use Case 2
6. User selects which patterns to download
7. Patterns are downloaded to local database for later viewing

8.0 Features

8.1 Pattern Search

Users will have the ability to search the local collection of patterns by:

- Pattern name or keyword in the pattern name
- Type – dry, wet, streamer, nymph, etc.
- Fish Species – patterns may apply to more than one species
- Material – the material used to construct the pattern

Search criteria is cumulative, meaning that a user can search by more than one criteria and the results will be narrowed using an “AND” operation. For example a user can search for dry bass patterns.

The result list will be presented to the user for selection. Selected patterns will be displayed in the “view panel”.

8.2 View Pattern

Selected patterns will appear in a “view panel”. The “view panel” will display:

- The fly name
- The fly tyer’s name (optional)
- An image of the fly
- The fly “recipe” – a list of parts and the materials used for them. It is important to note that the order of parts in the recipe as they reflect the order that they will be applied to the fly.
- Instructions – step by step instructions, usually including text and images, may be displayed. (optional)

8.3 Play Media

If media (video or sound) is attached to the pattern, the user will have standard controls to play the media. Media will be shown in the view panel.

8.4 Create Pattern

Users will have the ability to create a new pattern by adding information for the view panel ([see section 8.2](#)). Users will be able to drag and drop images and media as well as to enter data for the search criteria.

8.5 Print Pattern

Users will be able to print a simple hard copy of the pattern. Of course this will not include the media elements.

8.6 Server Connection

Users will be able to alter the address of the central server that acts as a pattern repository. This may include changing the user’s name and password if that is required by the server. Visual feedback will indicate to the user whether they are connected to the server or are acting in an “offline” mode.

8.7 Search Server Patterns

This is similar to the [Pattern Search](#) requirement except that the user is able to search and preview patterns that are stored on the server.

8.8 Download Patterns

Users will be able to download selected patterns from the central server to their local database for later offline use.

9.0 Application Panel Layout

The application will be laid out as several screens or panels that relate to a specific user task. As such all features that relate to that user task will be available directly from that panel. Features related to other tasks will be hidden from view.

The following diagram demonstrates how the major features relate to the various panels.

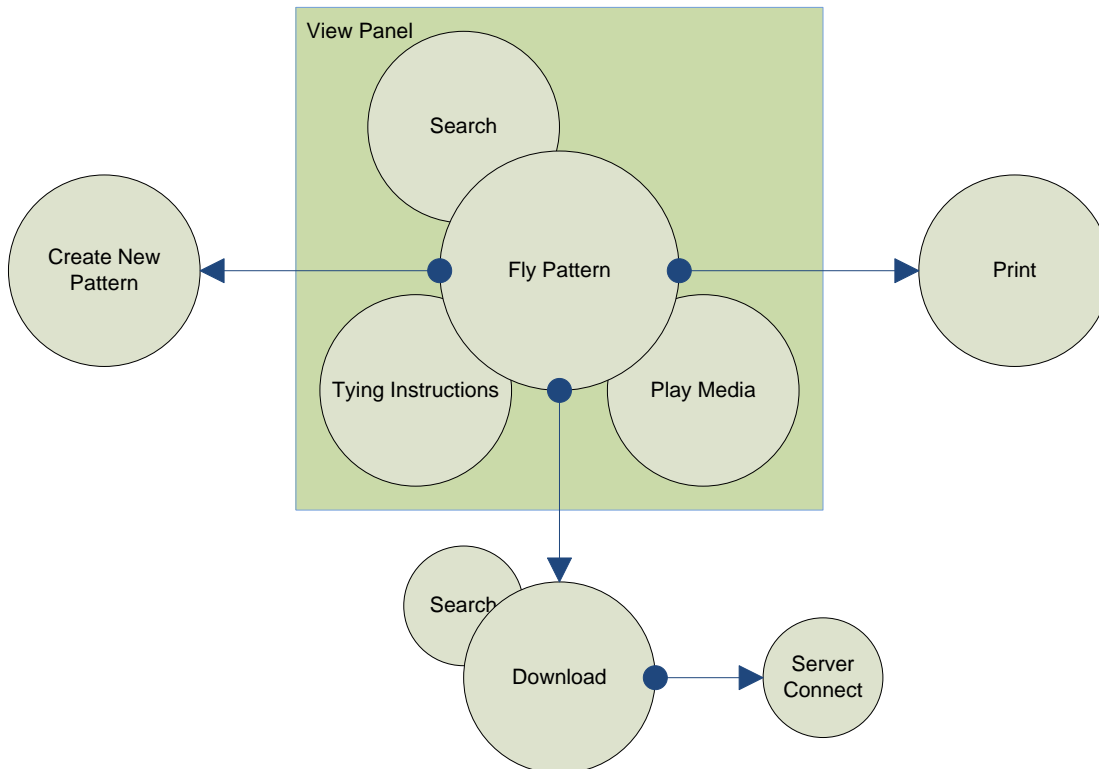


Figure 2: Structural diagram – screen interconnection

10.0 Database design

It is expected that all pattern records will be stored in a local database. The development environment will be Adobe AIR which requires a unique object for each database record retrieved. In other words, the database design will drive many of the data structures inside the AIR application.

The following database structures are based on a normalized (to second normal form) structure. Underlined items indicate the primary key, bold items indicate required data:

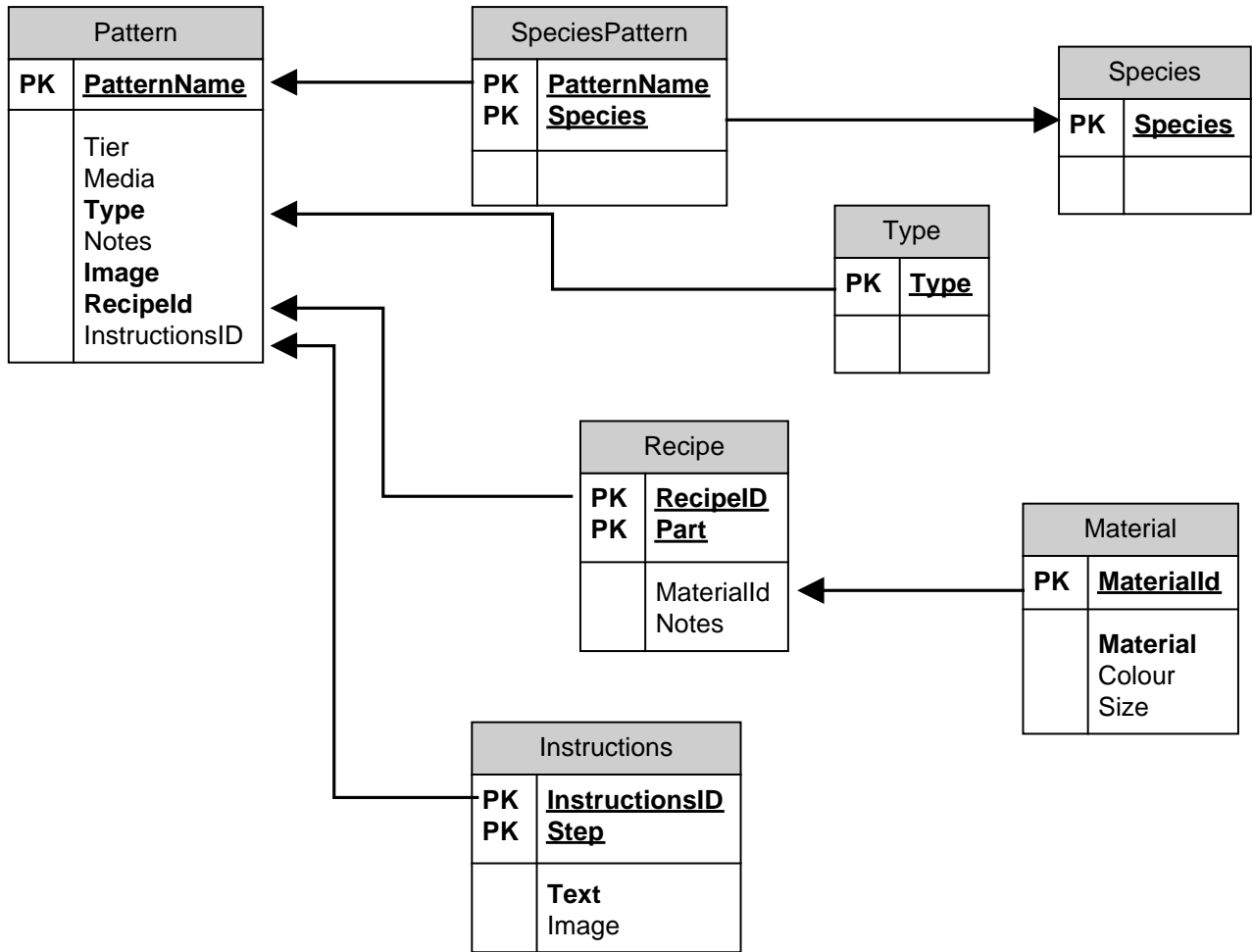


Figure 3: Database Design