INTRODUCING MARSHALL 2008

Jason Maddocks, Tim Ward, Andy Goh, Matt White

Marshall

This presentation is designed to give you an overview of our project, its aims, its technologies and its impact on a well established business.

- Client Background
- Project Overview
- Technologies
- Design

CLIENT BACKGROUND

Client Overview

Aero IT – World Class Aviation Software

Software development company specialising in the creation of operational Aviation Software.

Established in 2007, group developing since 1999.

Created by professionals from both the Software Development and Aviation Industries for the Aviation Industry.

Business Aims

Motivation

"Many General Aviation operators find it difficult to operate their business, incorporate new regulations, Aircraft Directives, Manage Aircrafts and Crew, analysing operational costs, CASA audits, win customers and students"

Aviation Management System

Concentrates on Operational Aviation Management for Charter Flight Management and Flight Training operations.

Management for:

- Aircraft Maintenance
- Pilots
- Customers
- Student and Supplier Management
- Flight Operations Analysis
- Payment Processing
- Extensions for CRM and Accounting

Current Situation

AeroIT currently faces documentation problems:

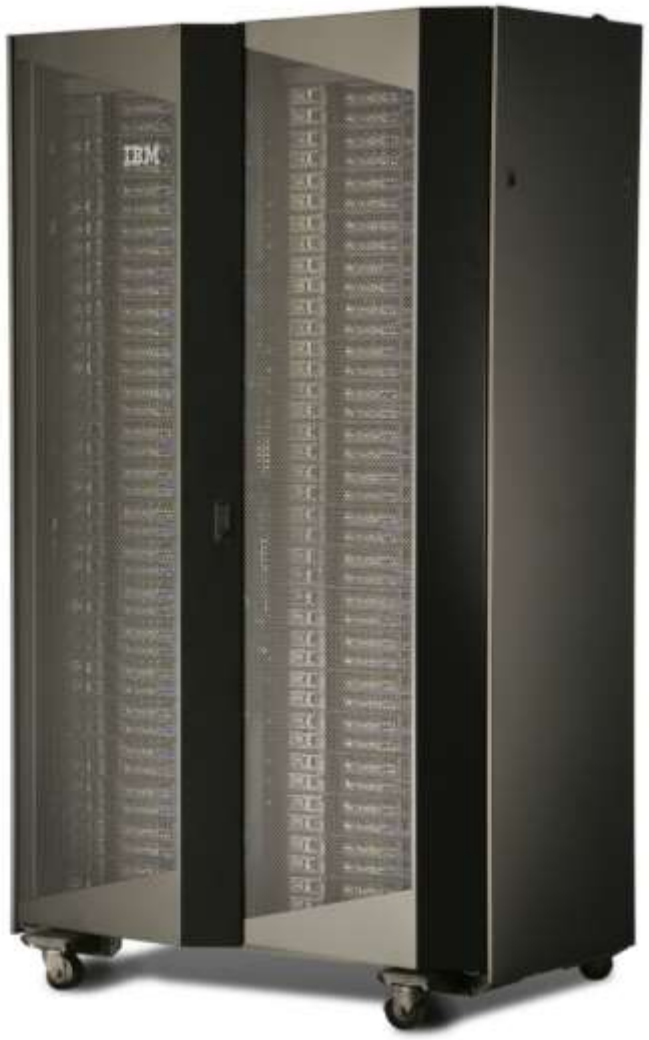
- Too many applications
- No central documentation system

Long-Term Business Goals

Moving to Web Services architecture.

Hosting all services for clients.

Integration with other Aviation related services. E.g. Airport fuel cost services.



PROJECT OVERVIEW

Project Purpose

AeroIT is looking for a solution to store their documents in a centralised location.

Allowing for:

- Easy Documentation Management
- Centralised access
- Expanded Documentation Format

Business Position

A new method to publish documentation

Replaces printed documentation

Web-based solution to manage documentation

Business Benefits

Advantages:

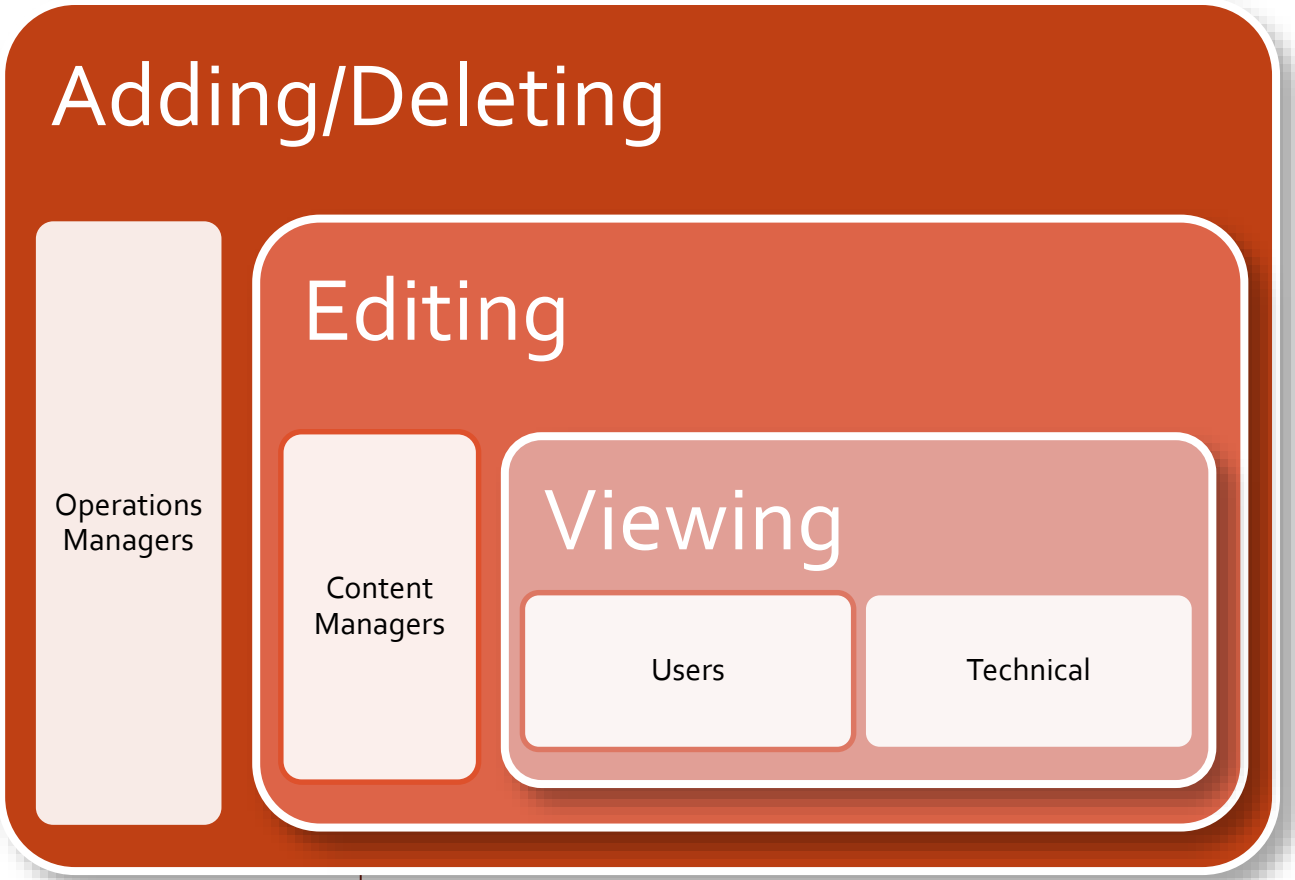
- Easier management of documentation
- Web-based system
- Easy utilisation of documentation by users
- Various methods for documentation delivery
- Lower documentation cost
- Environmentally friendly

Project Requirements

The client specified a range of general requirements that had to be satisfied:

- Web-based system to manage published documentation
- Database independent
- Operating system independent
- Hardware independent
- Intuitive user interface
- Built around THE Web 2.0 standpoint

User Overview



Users:

- Users
- Technical Users
- Content Managers
- Operation Managers

TECHNOLOGIES & TOOLS

WEB 2.0

What is Web 2.0?

There is no formal definition for Web 2.0, but the general ideas are:

Community driven

Information sharing

Collaboration

Rich + User friendly interface based on AJAX

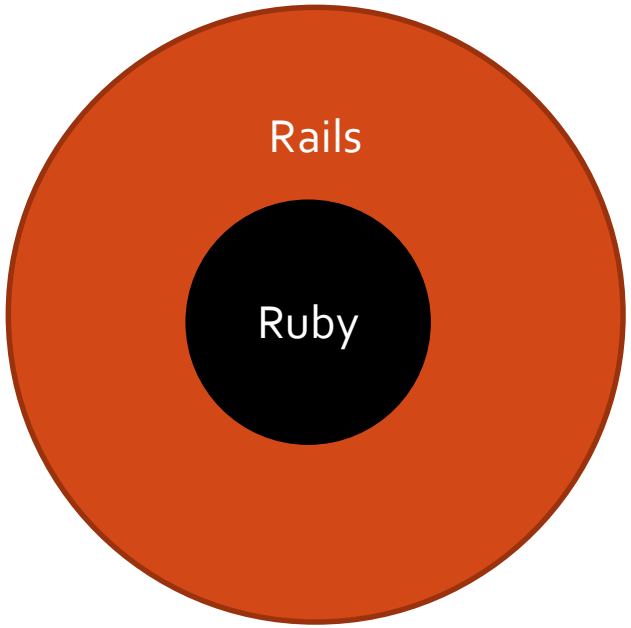
Tools that facilitate Web 2.0 production:

- AJAX
  - JavaScript
  - XML/XHTML
- Cascading Style Sheets
- Separation of data from business processing

RUBY ON RAILS

History

- First released to public in July of 2004
- Was created through the work of 37 signals on a web application called BootCamp
- Included in Max OS X
- Its essentially a framework sitting on top of ruby
- Often shortened to *ROR* or just *Rails*



Ruby

- Conceived in Japan in the mid 90's
- Very High Level
- Multiple Programming Paradigms
  - Functional
  - OO
  - Imperative
  - Reflection
- Interpreted language

AJAX & Web 2.0

Rails based around:

- Web 2.0
- AJAX

Rails Principles

"Convention over Configuration"

Developer only needs to specify unconventional things about an application

Advantages

- Less coding
- Encourages good naming conventions

- **"Don't repeat yourself"**
- Everything is located in a single unambiguous location.

- **Advantages**
  - Encourages abstraction
  - No repeated code

Scaffolding

Every web application wants the same basic CRUD functionality

Create

- Allows for the creation of a new object with its associated fields
- e.g Creating a new recipe for making cookies, with a title and description

Read

- Viewing and displaying of these new objects and there fields
- e.g Viewing our cookie recipe and the steps to make it

Update

- Editing and updating the fields of a previous object
- e.g Changing the title on our cookie recipe to "Hash Brownies"

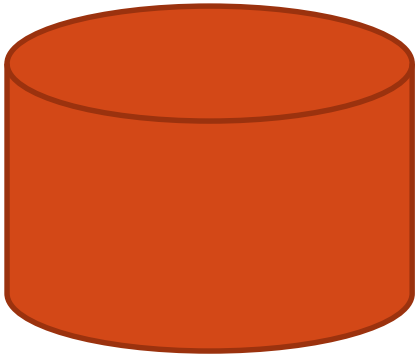
Delete

- Removing an object from our database
- e.g Deciding we don't want to leave incriminating evidence and deleting our recipie

Active Record

"Active Record connects business objects and database tables to create a persistable domain model where logic and data are presented in one wrapping."

Object-Relational Mapping (ORM) put on Rails



Features

- Automated mapping
- Associations and aggregations
- Validation
- Call-backs
- Observers
- Inheritance hierarchies
- Tree and list data types
- Transaction support
- Reflections
- Direct manipulation
- Logging support

Migrations

- As your application grows, the database is going to grow and change with it
- Migrations, is a tool to support this
- Allows for the evolution of the database
- Through the specification of "migrations"

Migration 1
- Add Recipe Table

Migration 2
- Add Date added column

Migration 3
- Change a column title

Gems – Plug-in

ROR supports the use of packages and plug-ins.

Great for building applications quickly.

Gems is the package manager for rails. It allows for the installation of 3rd party plug-ins for rails and ruby.

Rails Tools

There 2 main ways ruby on rails applications can be developed:

- Using the console and a text editor
- Or a ROR supported IDE

One of the better IDE's is Rad Rails

http://www.aptana.com/radrails

TOOLS

Aptana

Open-Source IDE for AJAX projects, based on Eclipse.

Available under the GNU General Public License (GPL), or the Aptana Public License (APL)

Plug ins for not natively supported languages (e.g.: RadRails)

SVN

Provided by Monash University

SVN Server + Trac (coming soon to a cinema near you)

Version Control

Collaboration of work

Configuration Management

Management of software and hardware setup.

TortoiseSVN

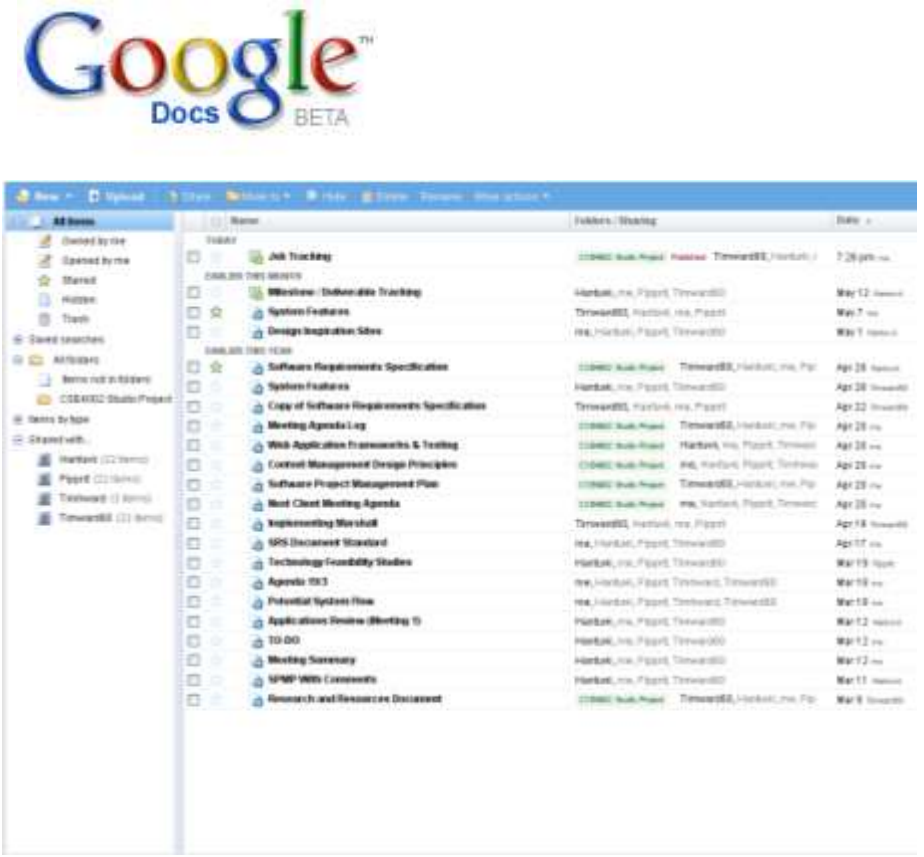
Google Documents

On line Google based system

Incorporates traditional document editors into a web-based system, with real-time web chat with other co-owners of documents.

Useful for simultaneous editing of documents from separate locations.

Aids in communication while working.



DESIGN

Design Overview

Web2.0-Based

Using web 2.0 ideas to build the system around a Client Server Architecture

Model-View-Controller Architecture

Allows us to separate the business functionality of the system and display

Rapid-Application Development

Concentrating on implementing the key features as early as possible

Expandability

Design with future expandability in to Web Service Architecture in mind

Design:

- System is structured centrally around a common entity. **Document**
- Data Model is based around this, and designed to be easily account for additions and modifications.
- Concentration on Presentation and Interface. Support for Multimedia Documentation
- Interface with existing user authentication system.

Design Goals

Our Goals:

- Platform Independent
- Database Independent
- Easily Maintainable
- Quality over Quantity
- Intuitive Interface
- Stylish

Web 2.0 & Our Project

Idea is for our project to fall under this umbrella of technologies

Documentation has always been considered a collaborative exercise

By using this style of web development we aim to create a:

- Fast
- Efficient
- Good Looking
- Useful

System

Client-Server Architecture

Client makes a service request.

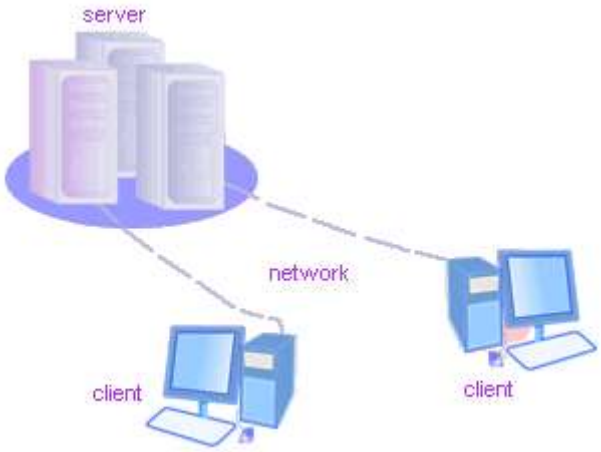
Server fulfils the request.

Examples:

Searching for documents.

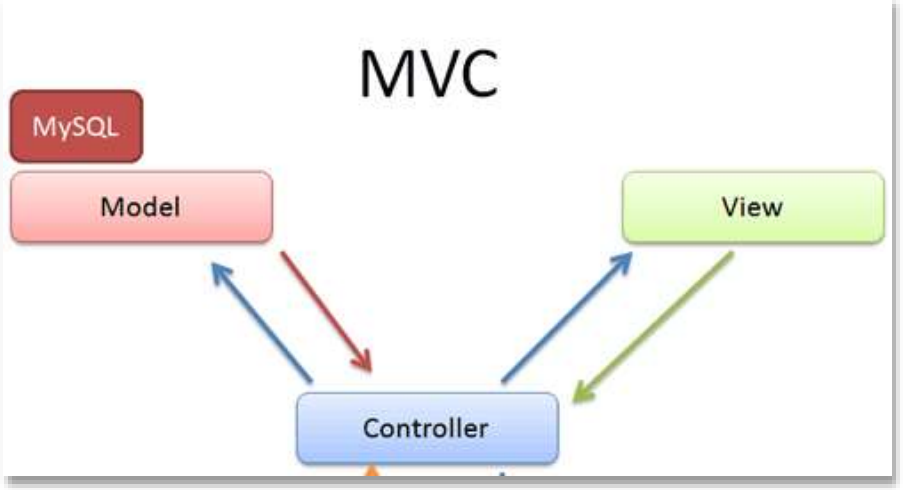
Browsing for Documents.

Viewing documents.



Model View Controller Architecture

- Software Engineering Design Philosophy
- About separating the display and the business functionality
- Used in a lot of frameworks
  - Java Swing
  - CakePHP
  - **Ruby On Rails**

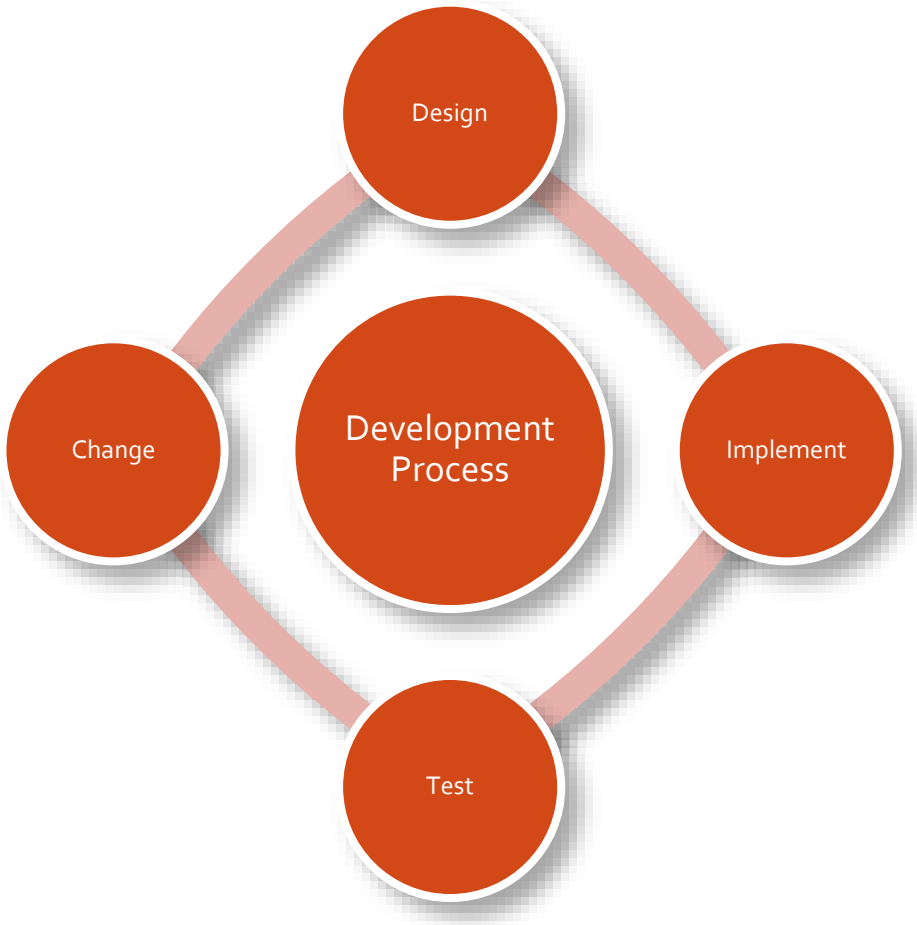


Rapid Application Development

Quick development and distribution of applications

Ruby on Rails supports this through a number of tools:

- Scaffolding
- Generators
- Migrations
- Rake

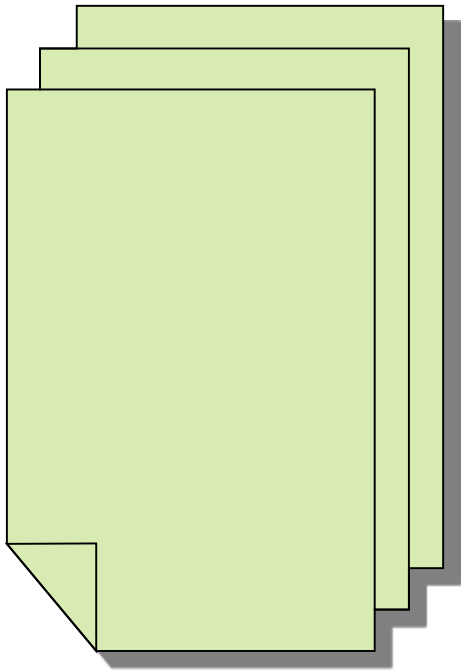


Documents

Documents make up the core of our system.

Documents include:

- Processes
- Reports
- Screens
- Fields
- Tables
- Databases



Applications and Categories

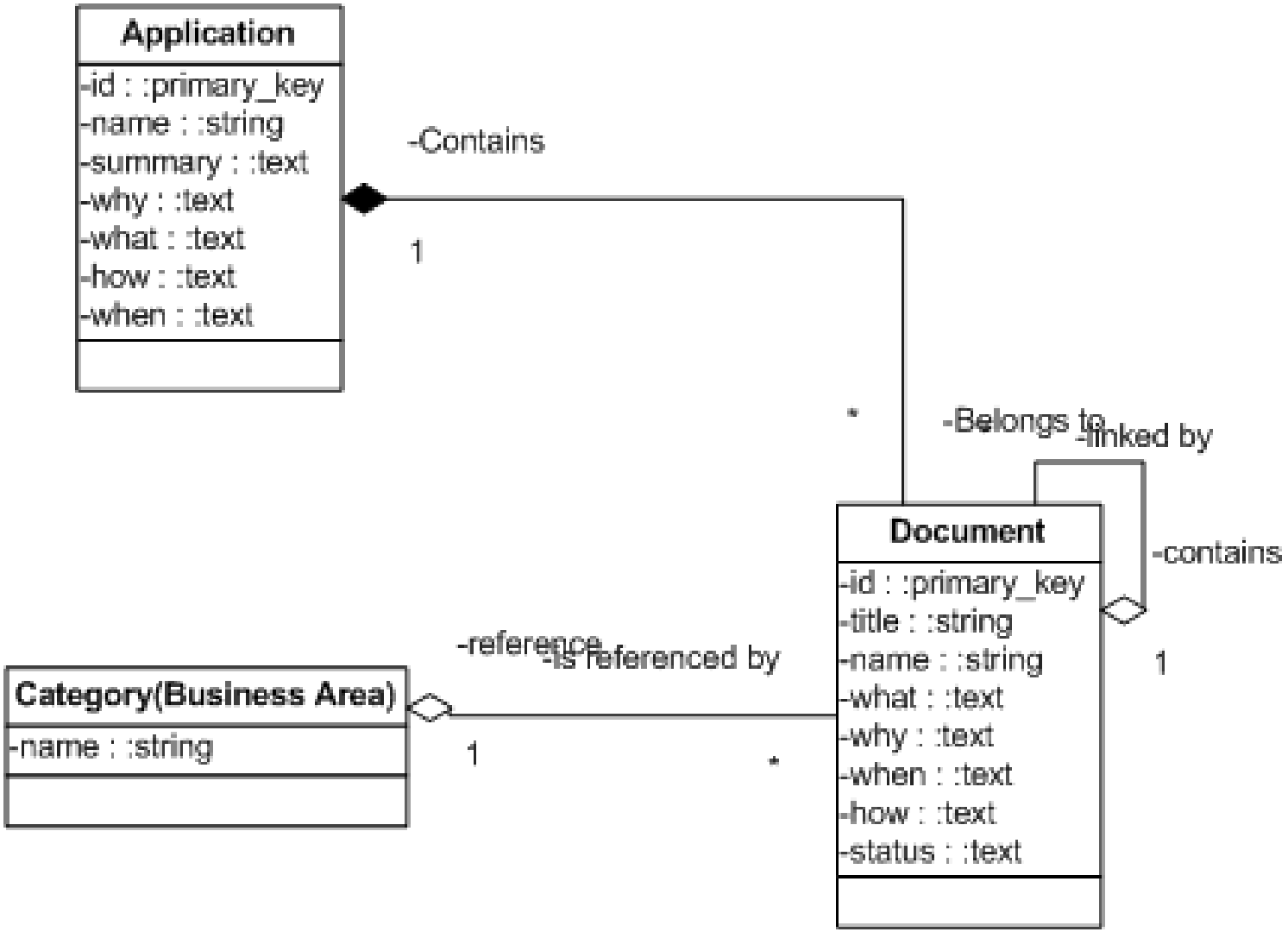
Documents belong in Applications and can further be refined into Categories.

Applications refer to what AMS Application the Documents relate to.

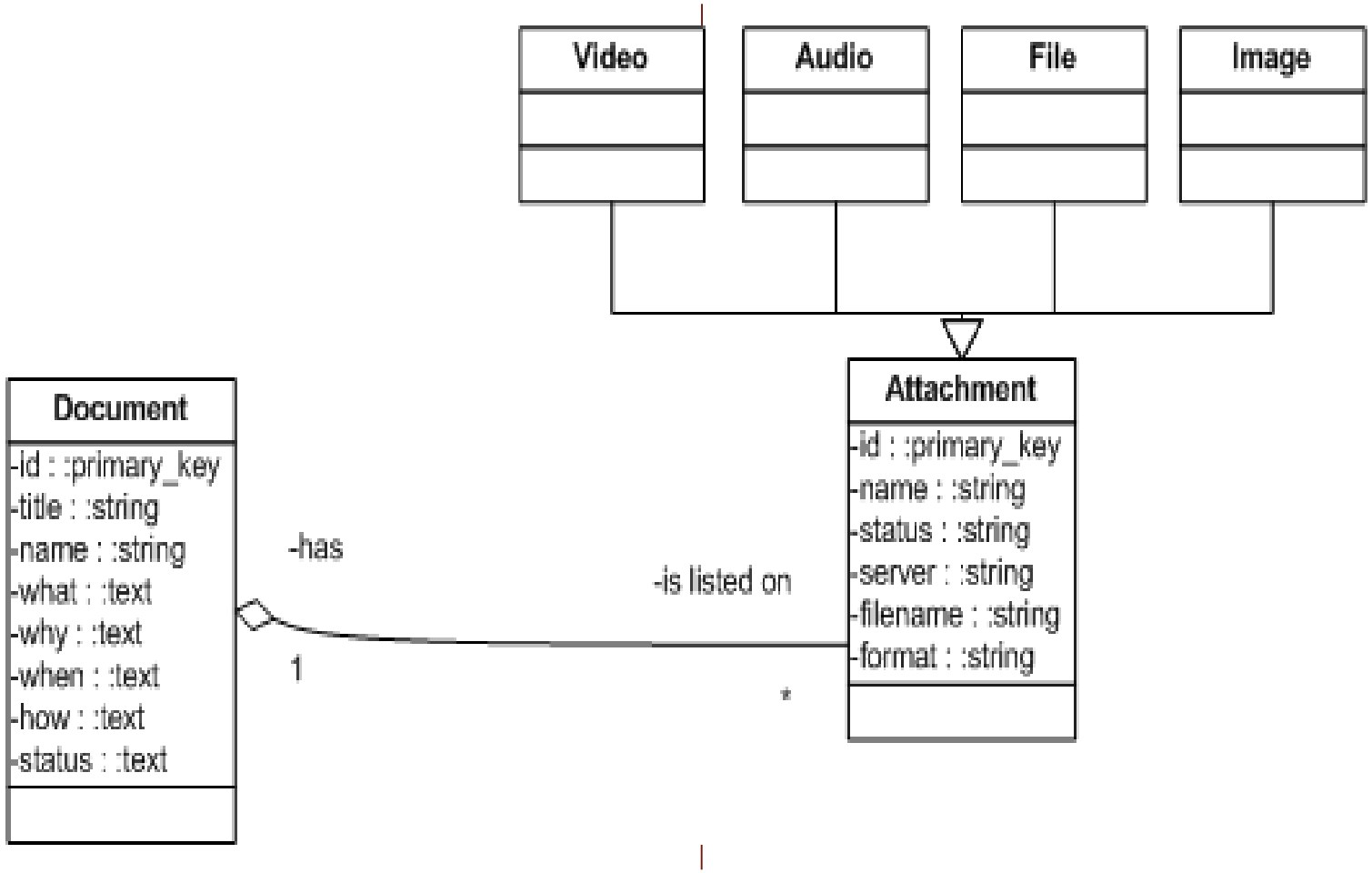
Applications have many categories.

Categories refer to what Business Areas the Documents relate to.

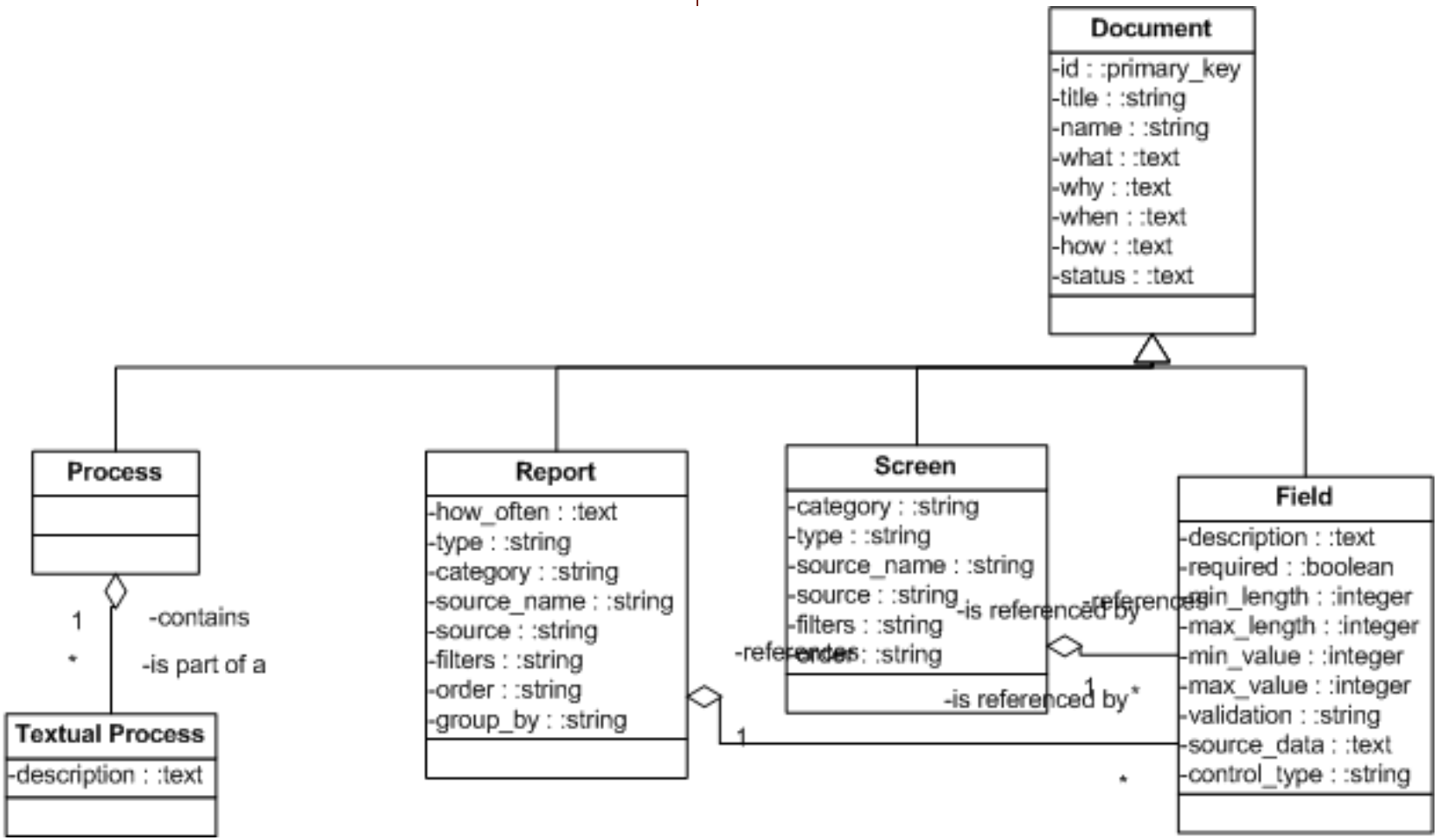
Class Structure



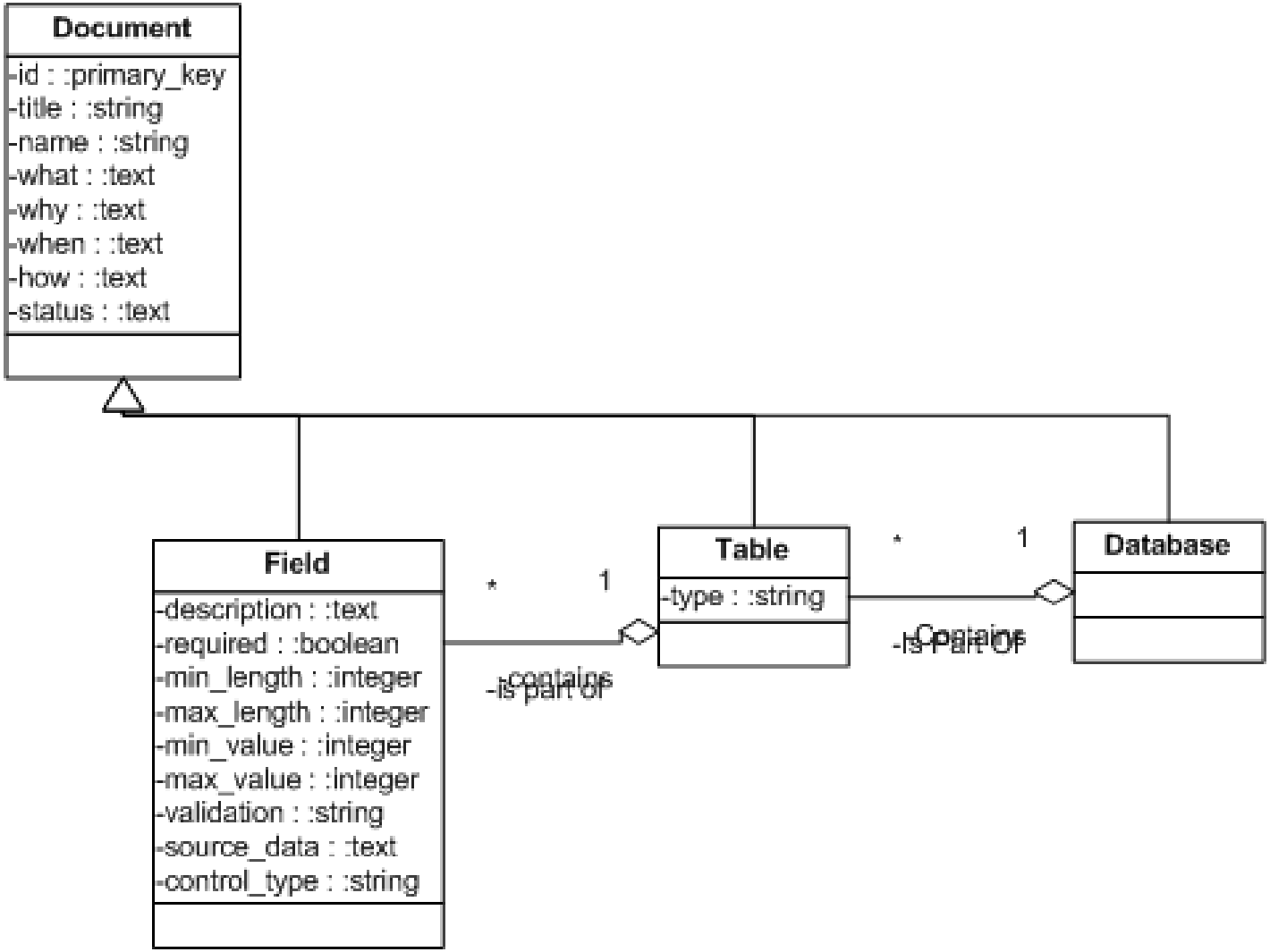
Class Structure (Cont.)



Class Structure (Cont.)

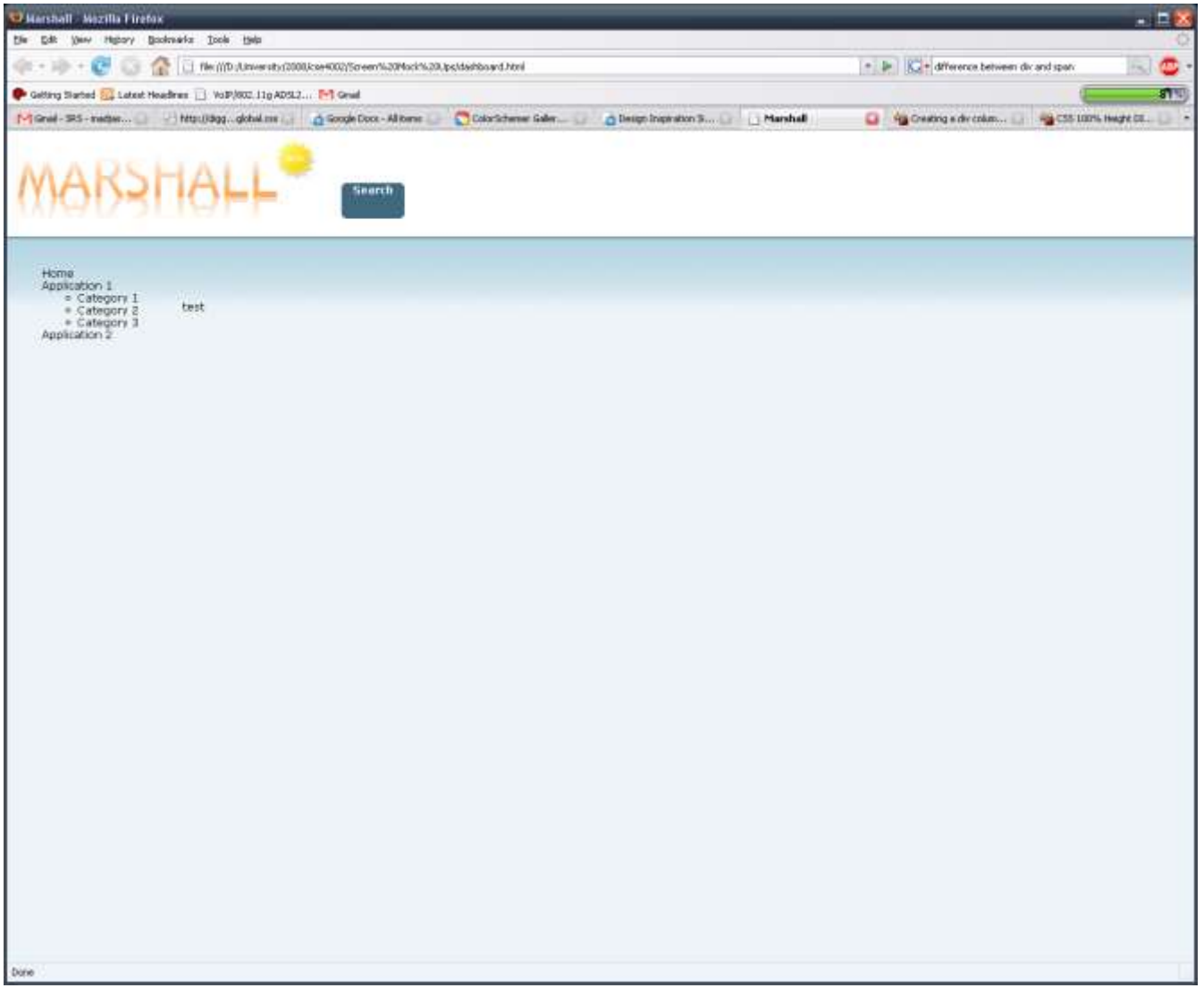


Class Structure (Cont.)

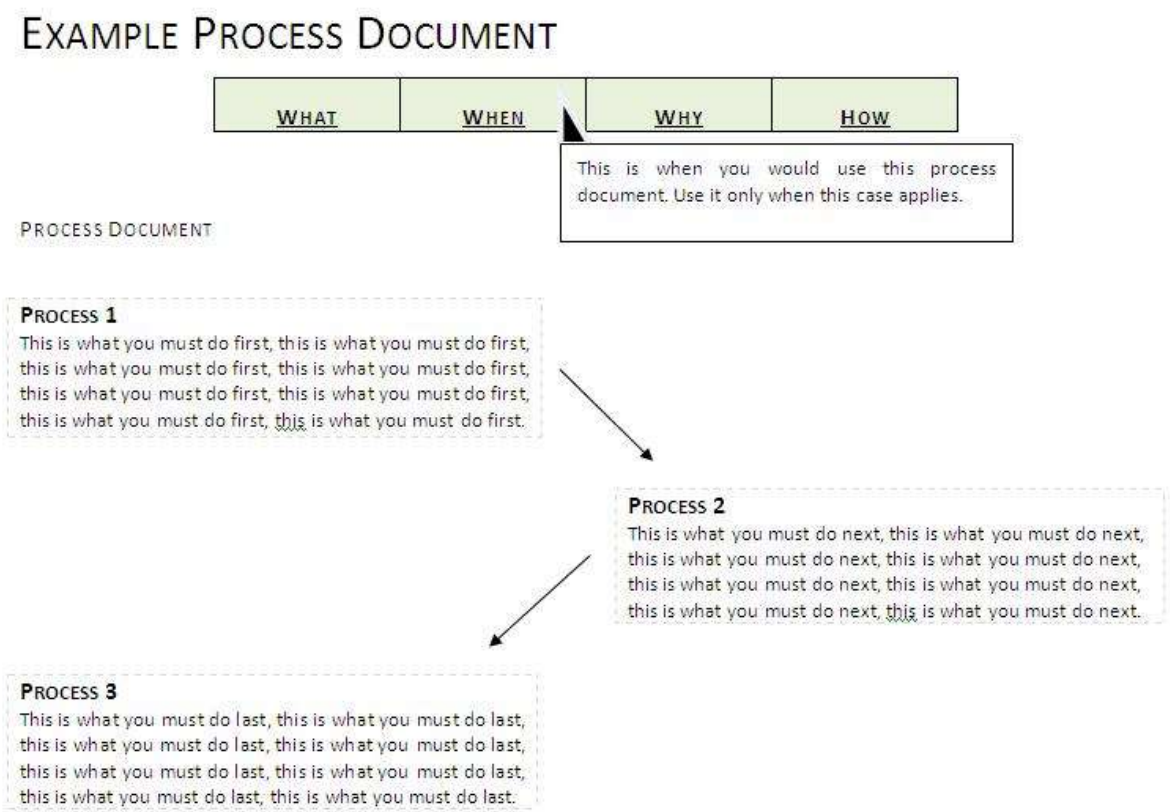


DESIGN MOCKUPS

Dashboard


Marshall - Mozilla Firefox
MARSHALL
Search
Home
Application 1
+ Category 1
+ Category 2
+ Category 3
Application 2
test
Done

Process Document



Screen Document

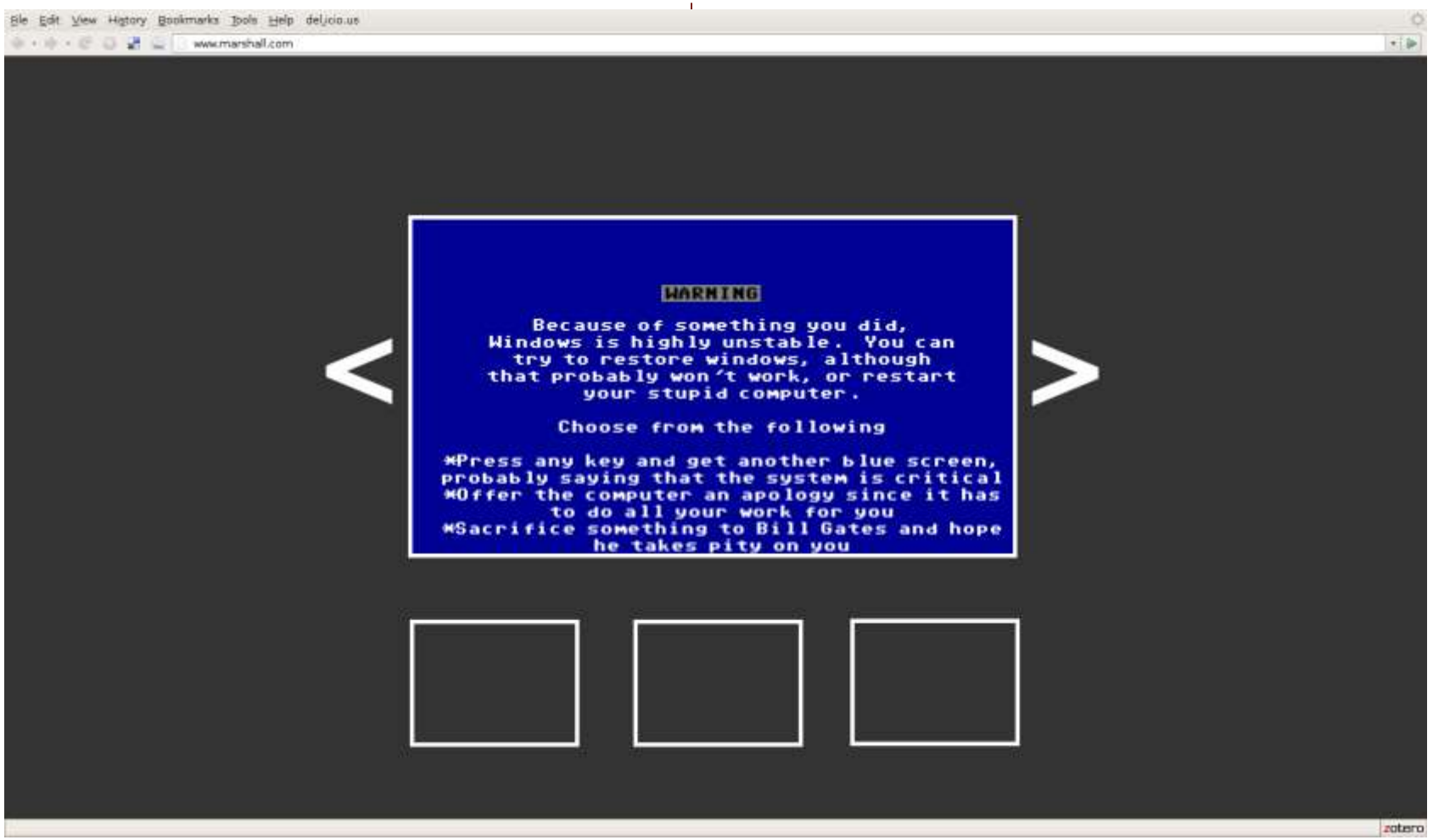
Example Screen Document

What	When	Why	How

Screen Document

This Screen is for some application and process.

Multimedia Gallery



IMPLEMENTATION PROTOTYPE DEMONSTRATION

CONCLUSION

Thankyou for your time...