Object Design Document

Comp566

Prof. Pierre Elysee

Kyu Park

Friday, March 1, 2013

1. Introduction

1.1 Object Design Trade-offs

1.1.1 Buy vs. Build

A decision can be reached by taking into consideration the following points:

If the software built will provide a competitive advantage by differentiating the company from competitors and if it is practical to build the system, given the limited resources, for less money and fast enough then it is better to build the system instead of buying it.

1.1.2 Space vs. Speed

If the software does not meet response time or throughput requirements then more memory can be expended to speed up the software (e.g. caching, more redundancy, etc.) If the software does not meet memory space constraints then data can be compressed at the cost of speed.

1.1.3 Delivery Time vs. Functionality

If the development runs behind schedule, a project manager can deliver less functionality than specified and deliver on time, or deliver the full functionality at a later time.

1.1.4 Delivery Time vs. Quality

If the testing runs behind schedule, a project manager can deliver the software on time with known bugs (and possibly provide a later patch to fix any serious bugs) or to deliver the software later with more bugs fixed.

1.1.5 Files vs. Databases

A file would be chosen incase of voluminous data (e.g. images), temporary data (e.g. core file), low information density (e.g. archival files, history logs). A database is chosen

in case of concurrent user access, access at fine levels of details, multiple platforms, and multiple application over the same data.

1.2 Interface Documentation Guidelines

Classes are named with singular nouns.

Methods are named with verb phrases, fields, and parameters with noun phrases.

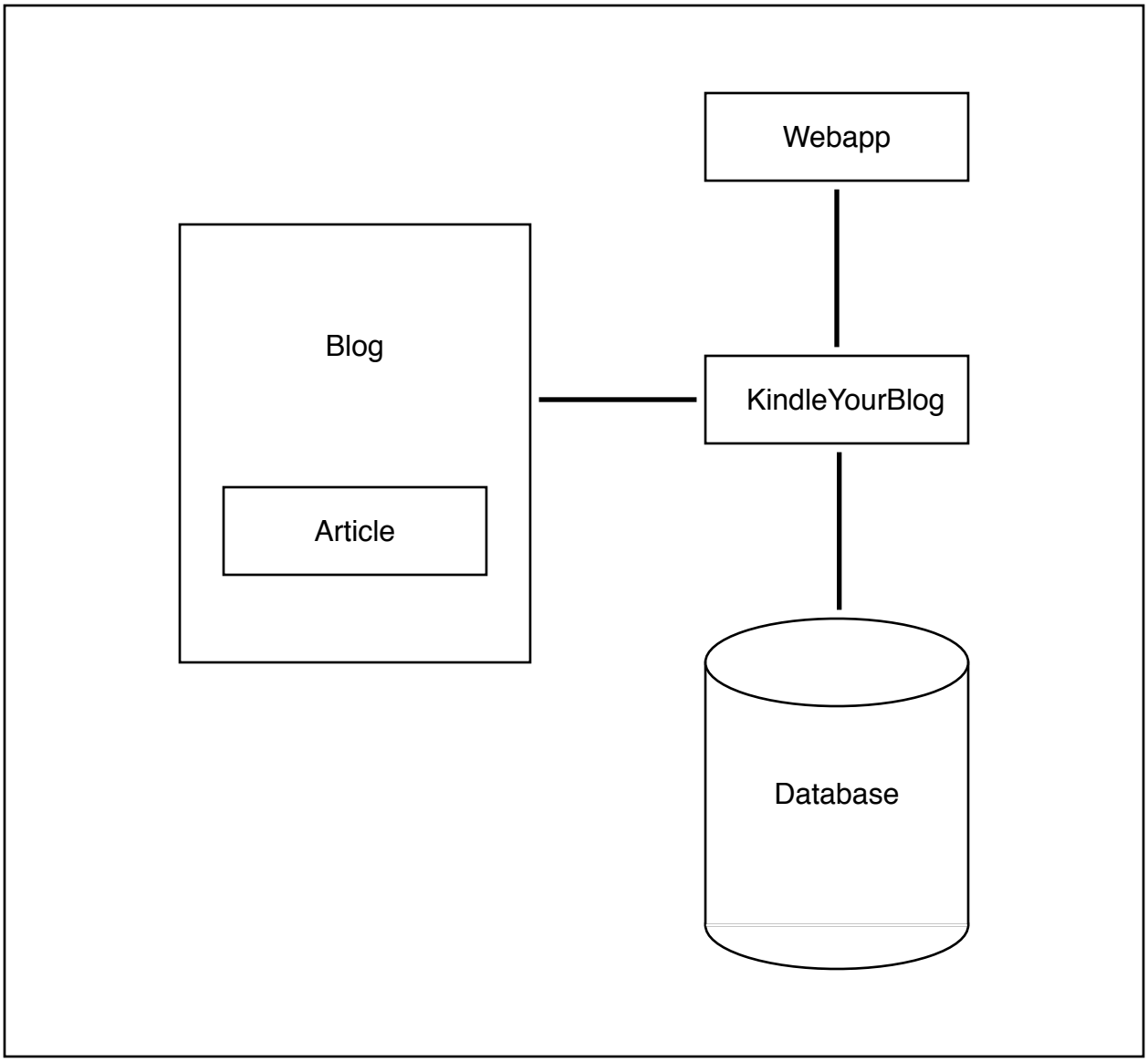
Error status is returned via an exception, not a return value.

1.3 Definitions, acronyms, and abbreviation

Term	Definition
Kindle (mobi)	Kindle is a ebook reader device made by Amazon, and also an Amazon trademark. Kindle also refers to a ebook format, a.k.a. mobi, that can be read on a Kindle device or an app.
KindleGen	The tool made by Amazon that helps convert a book information and metadata into a kindle ebook.
API	An application programming interface (API) is a protocol intended to be used as an interface by software components to communicate with each other.
Blog	A website where people can post a series of writings.

2. Packages

2.1 Package Diagram

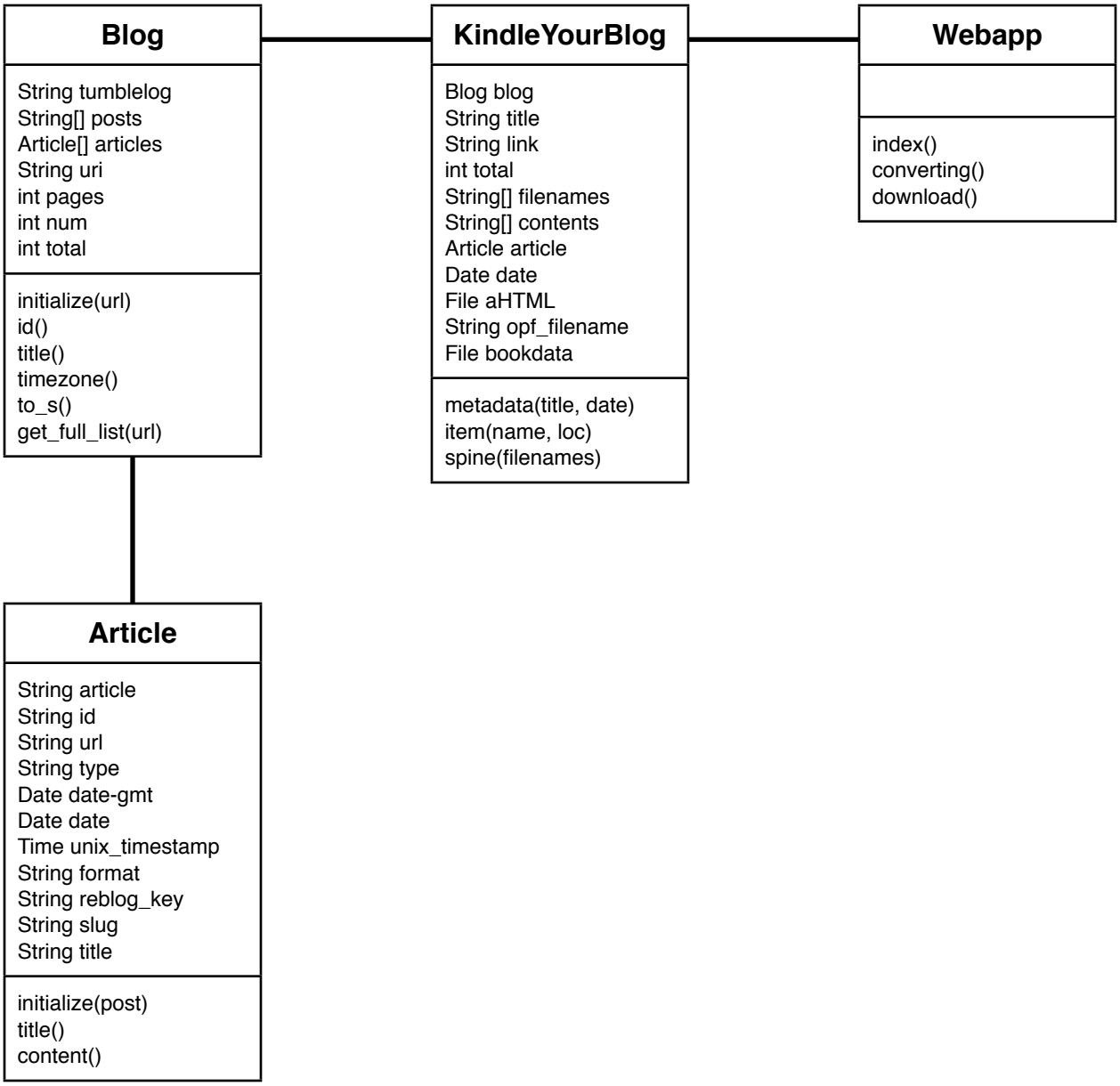


2.2 Package Definition

This software has only one package. This packages contains three main classes and one subclass.

3. Class Interface

3.1 Class Diagram



3.2 Class Definition

3.2.1 KindleYourBlog class (core)

KindleYourBlog
Blog blog String title String link int total String[] filenames String[] contents Article article Date date File aHTML String opf_filename File bookdata
metadata(title, date) item(name, loc) spine(filenames)

The KindleYourBlog class actually converts the blog into a kindle ebook with a command tool called KindleGen. This class just need a Blog object in order to convert into a kindle ebook. The KindleGen command tool requires html files to convert, so this class makes html files based on a Blog object which contains an array of Article objects.

3.2.2 Blog class

Blog
String blog_title String[] posts Article[] articles String uri int pages int num int total
initialize(url) id() title() timezone() to_s() get_full_list(url)

This Blog class helps to organize a blog's information. When an object of Blog class is initialized with a url, the object grabs blog information through the blog API based on the url. The blog API offers 50 articles at a time, so get_full_list method helps to collect all of articles based on int total which can be obtained on first api ping. A blog has an object array of Article class.

3.2.3 Article class

Article
String article String id String url String type Date date-gmt Date date Time unix_timestamp String format String reblog_key String slug String title
initialize(post) title() content()

This Article class helps to organize an article's information. An article has a variety of information, such as, date, time, article format, title, etc.

3.2.4 Webapp class

Webapp
index() converting() download()

The Webapp class contains three methods. In Ruby language, you can use the simple web framework called Sinatra in order to generate dynamic html webpages. Each method offers different pages.