ClassAxis Object Design

Marquette University, Department of Computer Engineering
COEN4610, Object-Oriented Software Engineering

Prepared for:
Dr. James Conigliaro

Prepared by:

Corey Bost	Johnny Fandel	Salome Marquez
Kellene Buchwald	Travis Huhn	Michael Santana
Dexter Del Frate	Andre Lancour	Sam Warmuth

December 10, 2010

Contents

1. Introduction

This document defines the restrictions, trade-offs, and guidelines for the project. It also includes a decomposition of the object design. The packages involved are investigated, as well as the class interfaces.

1.1. Object design trade-offs

The design for ClassAxis is defined by the unusual characteristics of our group. First, the group has nine contributors, which is larger than many of us have ever participated in before. Second, our varied schedules prohibit the entire group from meeting at the same time. Third, and most importantly, the majority of our group has never proceeded in a project like this previously.

1.1.1 Global accessibility versus ease of use

As our group cannot meet in person as frequently as is necessary to share revisions of code, a distributed system is necessary to allow everyone to contribute. However, a large portion of the group is unaccustomed to version control systems, and so a simplified compromise must be reached. These issues were alleviated by using git (for global accessibility), and using github.com as a host, as they provide a mainly graphical interface for repositories! Additionally, the choice of github allows us to use other features, such as issue tracking.

1.2 Interface documentation guidelines

1.2.1 Identifiers, Methods, Local Variables, and Class Variables

Identifiers, methods, class variables, and local variables will use underscore styling for all names. Examples:

- course_number
- current_users

1.2.2 Constants

Constants will use underscore styling and be entirely upper case. Example:

- COUCHDB_SERVER

1.2.3 Classes

All class names must be nouns. They will be capitalized and any additional words are also capitalized (without underscores). Examples:

- Broadcast
- CalendarTheme

1.2.5 Methods

Methods describe either their direct action if the goal is to effect a change, or relate directly to the object that will be returned. Examples:

- User.discussions
- generate_permalink
- logged_in?

1.2.6 Comments

Comments will use the standard ruby commenting procedure: “#” for single line comments, and “=begin” and “=end” for multiline comments. Examples

-
  ```
  user.set_permalink        # username was updated, so we need to regenerate permalink.
  ```

-
  ```
  =begin
  All helpers defined here will be available across all the application.
  @example A helper method for date formatting.
  def format_date(date, format = "%d/%m/%Y")
      date.strftime(format)
  end
  =end
  ```

1.4 References

2. Packages

The project will be organized into 4 overall sections: libraries, static files, configuration, and dynamic files.

- Libraries: external libraries used by the project.
- Static Files: images, javascript, and other non-dynamic files required by the site.
- Configuration: system configuration files.
- Dynamic Files: interpreted code. This falls into:
  - Helper methods

    - Database Model FIles
    - Server Routing Files
    - View Files

3. Class interfaces

- ## Broadcasts
    - Public attributes
        - date
        - type
        - content
        - announce_to_new_users
- ## Users
    - Public Attributes
        - name
        - date
        - calendar_id
        - permalink
        - is_admin
        - broadcast_ids
        - picture_url
- ## Messages
    - Public Attributes
        - sender
        - receiver
        - date
        - subject
        - content
        - unread
- ## Events
    - Public Attributes
        - name
        - location
        - description
        - tags
        - date
        - repeat
        - permalink
- ## Groups
    - Public Attributes

        - name
        - abbreviation
        - is_public
        - date
        - calendar_id
        - parent_id
        - user_ids
        - admin_id
        - wall_id
        - course_number
        - permalink
- **Calendar**
    - Public Attributes
        - date
        - name
        - event_ids
        - creator_id
- **Post**
    - Public Attributes
        - creator_id
        - parent_id
        - topic_id
        - date
        - content
        - permalink
- **Topic**
    - Public Attributes
        - title
        - content
        - tags
        - creator_id
        - date
        - permalink
- **Tags**
    - Public Attributes
        - name
        - date
        - creator_id