MyTimeline SRS

Prepared for: Student Government, College of Imaging Arts and Sciences, Timeline Information Services
Prepared by: Chris Cowdery-Corvan, Editor In Chief

Last Revised: April 5, 2009 (Revision 1.1.0)

Table of Contents

Project Summary

Objective

To deliver to students, alumni, faculty and parents a rich, custom-tailored publication to look back of their time at RIT.

Solution

Create a system leveraging our existing suite Timeline PHP web applications to provide users an innovative personal publishing suite accessible over the Internet. Users will be able to create their own custom publications by selecting what events they want, in addition to uploading their own photos and inserting their own text to create a truly custom experience. The users of MyTimeline will have the opportunity to share their book in print (printed directly from the HUB at RIT) or over the net.

MyTimeline will integrate with Timeline photo libraries as well as the users. By leveraging Application Programming Interfaces (API) provided by Facebook, Flickr and Picasa we will be able to include the user's custom photos to further enhance the personalized effect of the publication.

By combining the manpower from the College of Imaging Arts and Sciences (CIAS) and Timeline Information Services (TIS) and our existing code base, we plan to have a release candidate ready in time for Imagine RIT and perfect it for public use by Week 9 of Spring Quarter, 2009.

Also, events like Commencement (which occur after the release of the book) can be added effortlessly without the need for a new publication. For these reasons, it makes overwhelming sense to use the web as a publishing medium.

We also plan to improve the system as time progresses and look forward to the exciting new features next year!

Revision History

Revision	Notes
0.1	Initial Revision
0.2	Added Objective, Solution
0.3	Added Preliminary Schedule
0.4	Stubbed classes
1.0	Added details for each class
1.0.1	Added diagrams and revised documentation where appropriate Revised schedule
1.1	Remodeled classes after design meeting with developers

Schedule

Date	Task
April 5	Assign tasks to developers, develop HTML mockup of element
April 8	Meeting to check progress of HTML mockups, start backend development Establish database model
April 12	Review backend progress, collaborate on current status,attempt integration of components
April 15	Fix & address outstanding integration issues for R1, address any feature backlog
April 19	Begin R2 development, prepare demo plan for ImagineRIT
April 22	Review progress, start beta testing, start marketing PR blitz
April 26	Address any outstanding issues, finalize uploading of all photo and copy assets
April 29	Address any outstanding issues
May 2	Imagine RIT Demo
May 3	Fix up any issues, concerns presented at ImagineRIT
May 6	Grand unveiling event to student body

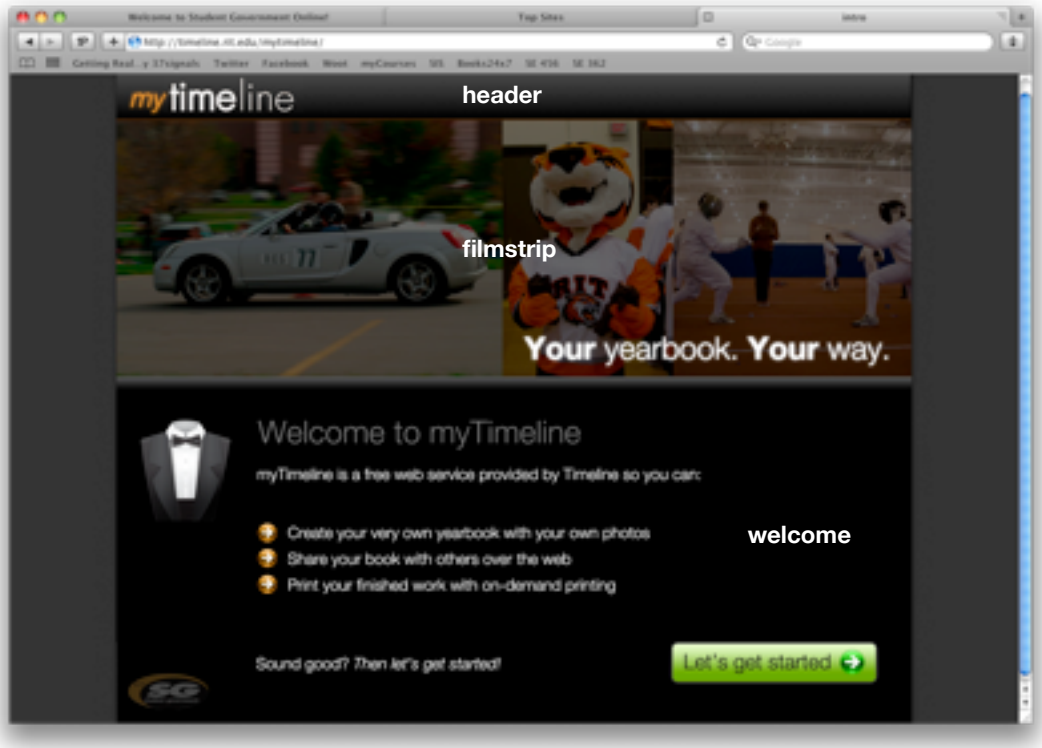
View: Welcome Page

Requirements and Specifications

- The user shall be presented with a “Lets get Started” option
    - Clicking this leads to the myTimeline passport creation
- The user shall be presented with a “Log In” option

Non-functional requirements

- The system shall use the CodeIgniter PHP framework.
- Libraries and helpers can be added as necessary, provided no licenses are broken.



Welcome Subsystem

Requirements and Specifications

- The welcome page shall be the default/index page for the application.
- The user shall be presented with a brief description of the intent of the application.
- The user shall be presented with brief instructions on how to use the application.
- The user shall be presented with two actions:
    - Login
        - Redirect to the MyTimeline login controller
    - Let's Get Started
        - Redirect to the MyTimeline account creation controller

Non-functional requirements

- The system shall use the CodeIgniter PHP framework.
- Libraries and helpers can be added as necessary, provided no licenses are broken.

Login Subsystem

Requirements and Specifications

- The system will have two forms of accounts, RIT accounts and guest accounts.
- The system shall use SSL/HTTPS when credentials are transmitted.
- The system shall present the user to select which type of account they own (RIT, Guest)
    - RIT accounts
        - The system shall prompt the user for their RIT DCE username and password
        - The system shall authenticate the inputted credentials over LDAP to the RIT Active Directory
        - The system then shall check to see if the user has a MyTimeline account associated with the DCE
            - If no DCE is found in the database, the system shall redirect the user to the Let's Get Started controller
            - If the DCE is found in the database, the system shall redirect the user to the Main controller
    - Guest Accounts
        - The system shall prompt the user for their email address and password
        - The system shall authenticate the inputted credentials against the MyTimeline guest database
            - If no user is found in the database, the system shall redirect the user to the Let's Get Started controller
            - If the user is found in the database, the system shall redirect the user to the Main controller

Non-functional requirements

- The system shall use the CodeIgniter PHP framework.
- Libraries and helpers can be added as necessary, provided no licenses are broken.

Let's Get Started Subsystem

Requirements and Specifications

- The system shall prompt the user if the user would like to use their RIT DCE or a MyTimeline guest account to login.
- The system shall use SSL/HTTPS when user credentials are transmitted.
    - If the user selects RIT DCE as a login system:
        - The system will prompt the user for their DCE username and password via HTTP authentication
        - The system shall write the information recorded in the browser session to the database and write the login information to a table for DCE accounts.
    - If the user selects MyTimeline as a login system:
        - The system will prompt the user for the following information:
            - Required Fields:
                - Email Address
                - Password
            - Optional Fields:
                - Name
                - Address
        - The system will write the information to a table containing the guest login information.
- The system shall send a confirmation to the user of account creation when completed.

Non-functional requirements

- The system shall use the CodeIgniter PHP framework.
- Libraries and helpers can be added as necessary, provided no licenses are broken.

Header Subsystem

Requirements and Specifications

- The header shall consist of one section and shall include the following at all times:
    - MyTimeline logo
- When a user is logged in, the user shall see extra options:
    - The user name
    - Logout
        - When activated, initiates logoff operation
    - Save
        - When activated, invokes the save operation
    - Share to Web
        - When activated, invokes the save to web operation
    - Send to Printer
        - When activated, invokes the send to printer operation

Non-functional requirements

- The system shall use the CodeIgniter PHP framework.
- Libraries and helpers can be added as necessary, provided no licenses are broken.

Sidebar

Requirements and Specifications

- The sidebar shall consist of two sections.
    - The first sidebar section shall display the current listing of events selected by the user.
        - This section will be referred to as the “My Selected Events.”
    - The second sidebar section shall display a list of possible events to select.
        - This section will be referred to as the “Event Drawer.”
- The user shall be able to add and remove events to “My Selected Events” from the “Event Drawer.”
    - The user will be able to drag-and-drop events from the “Event Drawer” to the “My Selected Events” drawer.
    - The user will be presented with a preview of the event if clicked.
- The sidebar must notify the MyTimeline observer when the following actions occur:
    - Event added to “My Selected Events”
    - Event removed from “My Selected Events”

Non-functional requirements

- The system shall use the CodeIgniter PHP framework.
- Libraries and helpers can be added as necessary, provided no licenses are broken.

Photo Viewer Subsystem

Requirements and Specifications

- The user shall be presented with two columns, a “Photo Library Selector” and a “Preview Pane”
  - Photo Library Selector
    - The user shall be presented with a listing of possible services to view photos from. These services include:
      - Timeline Photos
      - Facebook
      - Flickr
      - Picasa
      - Photos On My Computer
    - When each option is selected the switch library operation will be executed, affected systems will be notified accordingly.
  - Preview Pane
    - The user shall be presented with an album selector for the library selected.
      - Timeline photo albums will be designated by each event in the Timeline PhotoDB
      - Facebook’s photo albums will be designated by each album
      - The album selector will sort albums in reverse-chronological order.
    - The user shall be presented with thumbnail representations of the album selected in the pane
    - If the number of thumbnails for the given size goes over the window dimensions, a scrollbar will be presented, allowing the user to scroll through thumbnails
    - Clicking the thumbnail will notify all affected subsystems of selection and close.

Non-functional requirements

- The window shall open as a pop-up window when an PageElement of the Photo type is pressed
- Libraries and helpers can be added as necessary, provided no licenses are broken

PageCollection Subsystem

Requirements and Specifications

- PageCollection shall contain a PHP array of Page objects.
    - The index of the Page object in the array shall denote the page number.
- The Page objects shall be able to be reordered in any way requested.
- Page objects can be added, removed, updated or deleted.
- Page objects will be returned for any index asked and will return an exception if a page is not found.

Non-functional requirements

- The system shall use the CodeIgniter PHP framework.
- Libraries and helpers can be added as necessary, provided no licenses are broken.

Template Subsystem

Requirements and Specifications

- Template objects shall contain a collection of uniquely addressable PageElement objects.
- Template objects shall use XSLFO for layout information.

Non-functional requirements

- The system shall use the CodeIgniter PHP framework.
- Libraries and helpers can be added as necessary, provided no licenses are broken.

Page Subsystem

Requirements and Specifications

- Page objects shall have three forms of initialization:
    - Page objects shall accept a Template object as a constructor parameter.
        - The system shall generate the requested PageElements contained within the Template object
        - Note: This will be invoked for the creation of blank, new pages.
        - The system shall mark the Page as editable.
    - Page objects shall accept a URI to a pre-compiled PDF document as a constructor parameter.
        - Note: This will be invoked for the inclusion of pre-compiled pages.
        - The system shall include the PDF and return the PDF when asked to render.
        - The system shall mark the Page as unable to edit.
    - Page objects shall accept a MyTimeline User ID and a Page number as constructor parameters.
        - The system shall pull corresponding assets from the database and add the template and PageElements as stated from the database.
        - The system shall mark the Page as editable.
        - Note: This will be invoked when the user is attempting to open an existing publication
- Page objects must import any assets needed from the Template object (i.e. background images)
- A Page object's PageElements must be accessible and accept a common PageCollection iterator for all editable Pages.
- When asked to render, the system shall present a JDF-compliant PDF representation of the pa

Non-functional requirements

- The system shall use the CodeIgniter PHP framework.
- Libraries and helpers can be added as necessary, provided no licenses are broken.

PageElement Subsystem

Requirements and Specifications

- PageElement shall be an interface dictating the content of a page.

Non-functional requirements

- **PageElements shall adhere to the Virtual Proxy design pattern.**
- The system shall use the CodeIgniter PHP framework.
- Libraries and helpers can be added as necessary, provided no licenses are broken.

Copy Subsystem

Requirements and Specifications

- Copy shall extend the PageElement interface.
- Copy shall contain a String of the text desired.
- The system shall write any changes to the database when they occur.

Non-functional requirements

- The system shall use the CodeIgniter PHP framework.
- Libraries and helpers can be added as necessary, provided no licenses are broken.

Image Subsystem

Requirements and Specifications

- Image shall extend the PageElement interface.
- Image shall contain a URI to the image referenced.
- The system shall write any changes to the database when they occur.

Non-functional requirements

- **All images will be of the same resolution, DPI and format (JPG).**
- The system shall use the CodeIgniter PHP framework.
- Libraries and helpers can be added as necessary, provided no licenses are broken.

Book Subsystem

Requirements and Specifications

- A book object shall be initialized for each web session of MyTimeline.
- Book shall contain the following objects for each instance:
    - PageCollection
    - AssetLibrary

Non-functional requirements

- The system shall use the CodeIgniter PHP framework.
- Libraries and helpers can be added as necessary, provided no licenses are broken.

Main Subsystem

Requirements and Specifications

- A main controller will contain the following objects:
    - A Book object for the current user
    - A Header
    - A Sidebar
    - A Iterator
        - The system shall present the user with controls to go forward or backwards in the Pages collection.
        - The system will invoke the render method on each page when requested and replace the images where necessary.
- The system shall also set up any session variables requested by other subsystems.

Non-functional requirements

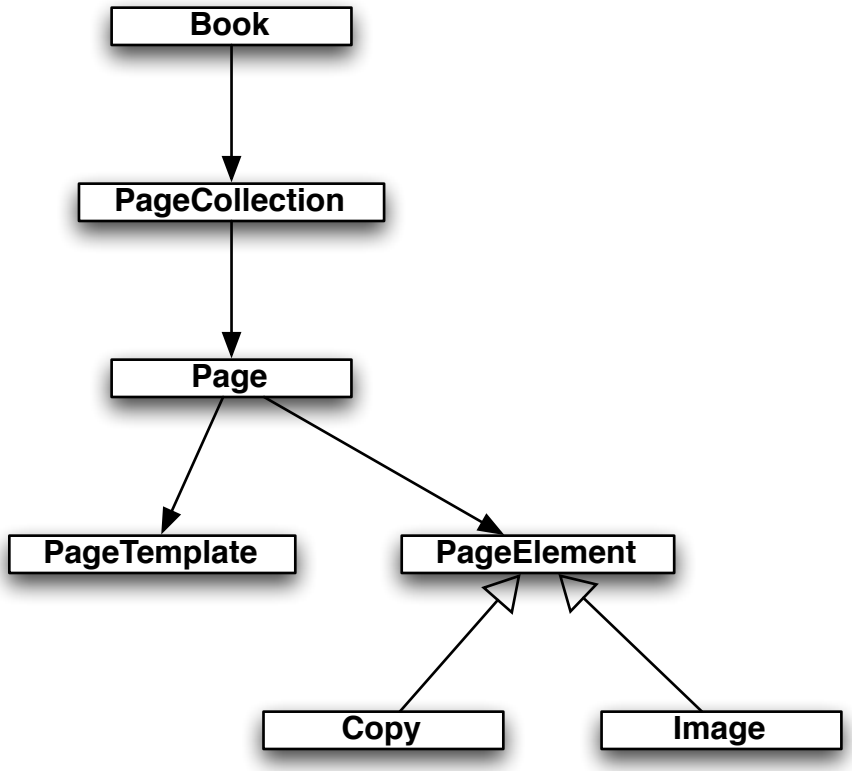
- The system shall use the CodeIgniter PHP framework.
- Libraries and helpers can be added as necessary, provided no licenses are broken.

Diagrams & Figures

Note: These UML diagrams reflect the current stage in the design process are rough and will be refined as standards have been agreed upon.

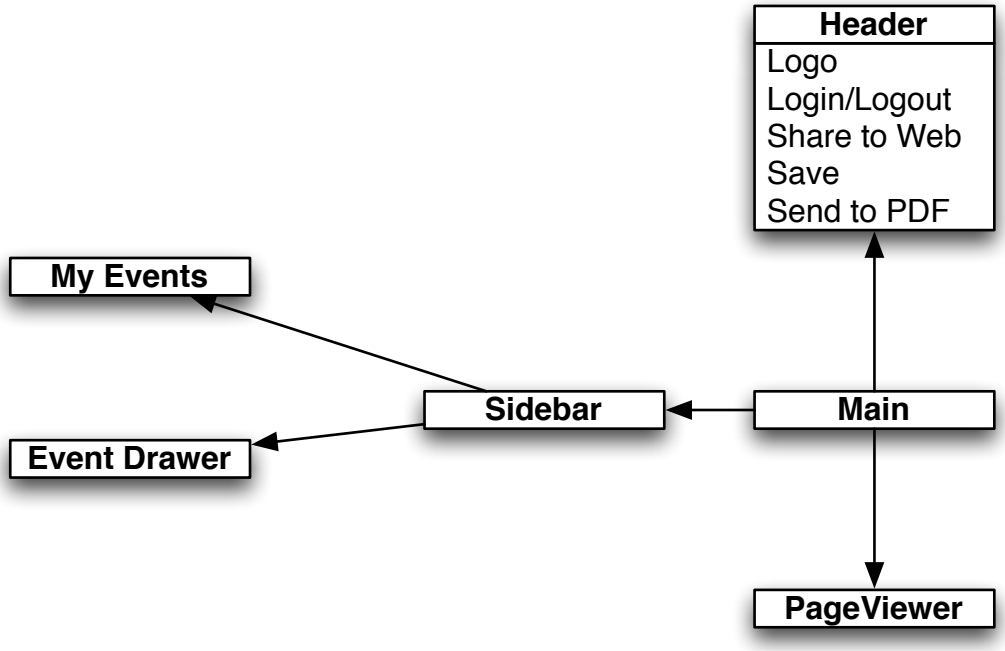
Book Subsystem UML

Developer: Chris Cowdery-Corvan (TIS)



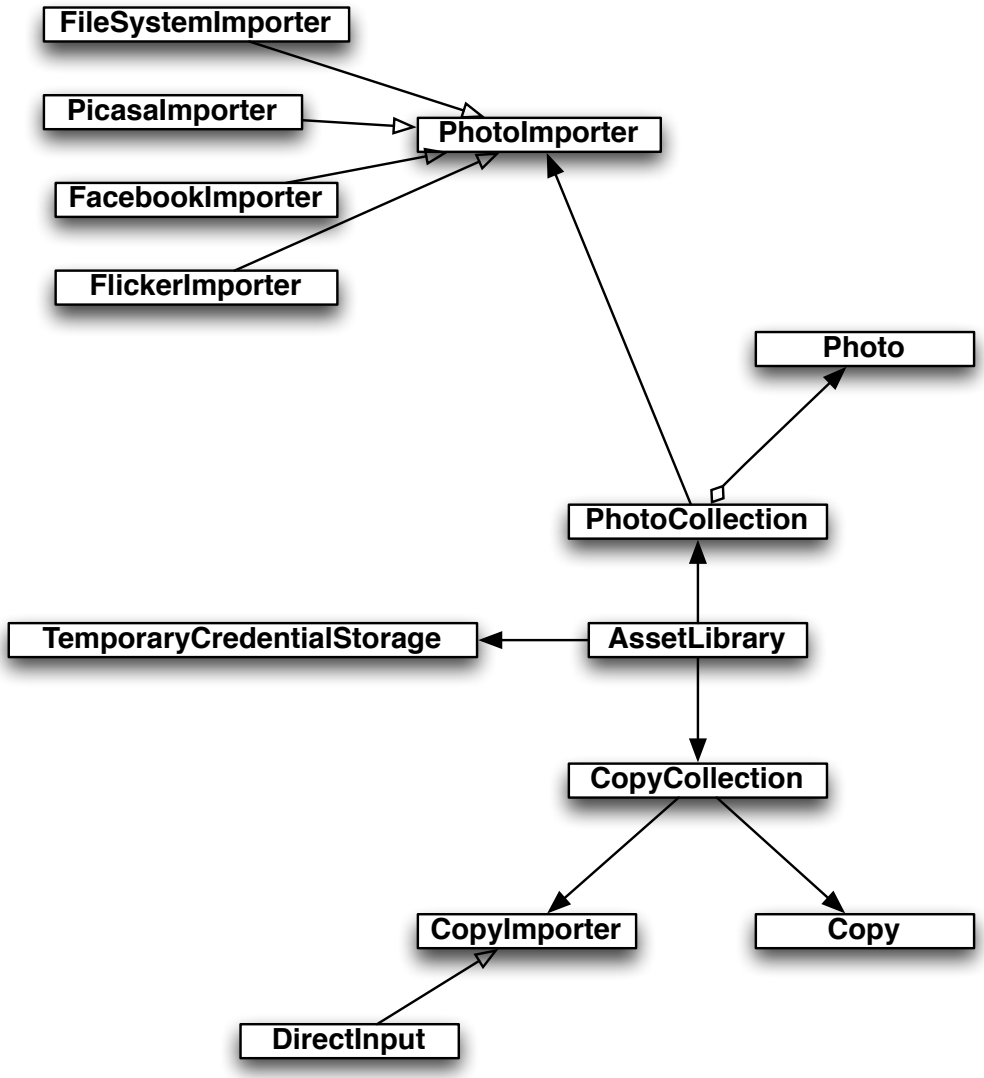
View Subsystem UML

Developer: Mike Bruckner (TIS)



AssetLibrary Subsystem UML

Developer: Justin Gerace (TIS)



Example Interface