Public Library Geographical Photo Archive (PLGPA)

A Look Back At Your Neighborhood

Software Design Specification

Prepared by
Mohammed Al Rawi,
Angel Orduno,
Nam Pham,
Jose Rivas Jr,
Taisei Terasaka
Section 1: Introduction
This document will outline in detail the software architecture and design for the Public Library Geographical Photo Archive (PLGPA). This document will provide several views of the system's design in order to facilitate communication and understanding of the system. It intends to capture and convey the significant architectural and design decisions that have been made for the PLGPA.

1.1 Scope
This document provides the architecture and design of the PLGPA. It will show how the design will accomplish the functional and nonfunctional requirements detailed in the PLGPA Software Requirements Document, (SRD) document.

1.2 Intended Audience
This document is written on a technical level to address Medtronic’s technical department and CSULA Computer Science department.

1.3 References
Public Library Geographical Photo Archive (PLGPA) Functional Software Requirements Document (SRD).

Section 2: System Overview
The County of Los Angeles (County) Public Library (Library) collects historical images of areas, residents and other artifacts, those images are relevant to specific locations across the County. The Library acquired a Software as a Service (SaaS) solution called CONTENTdm to host, categorize and publish its archive. CONTENTdm offers web publishing tools to easily create pages showcasing digital collections, additionally, CONTENTdm offers a WorldCat global categorization making digital more discoverable to people around the world. CONTENTdm also provides a cloud-based with an Active-Active disaster recovery solution to maintain digital collection’s availability.

With the massive footprint and population, the County’s digital archive is anticipated to increase exponentially. With the current publishing layout of a single page and digital collection grouped by area names and listed alphabetically, user experience will be affected as it will be difficult to find collections that are of the user’s interest. To address this issue, the Library partnered with California State University - Los Angeles, College of Engineering, Computer Science, and Technology to implement a the Public Library Geographical Photo Archive (PLGPA), a mobile-friendly web application to showcase its
collections with an interactive map to enrich user experience on both stationary and hand-held digital devices.

2.1 Context Diagram (DFD Level 0 Data Flow Diagram)

PLGPA’s architecture is summarized in the Context Diagram (DFD Level 0) given below. A more detailed Functional Description is given in Section 2 of this document.

![Context Diagram (DFD Level 0)](image)

**Figure 1-1. Context Diagram (Level 0 Data Flow Diagram)**

2.2 Context Diagram (DFD Level 1 Data Flow Diagram)

![Context Diagram (DFD Level 1)](image)
Section 3: Design Considerations

3.1 Goals
- Our goals for this web application is to make it mobile friendly.
- No matter what device you are using to view the site, it will be very responsive
- The app will have an integrated map where it will depict the user’s location and also nearby historical archives
- It will be a parallax scrolling page
- It will also have County services on the map, which would be a helpful resource
- Social media integration for user interactivity
- Social media share counters which will give insight to what recent users have been looking at
- If users want to download and share an image the options will be available
- We will reference contentdm and use some of the meta data to fill the image bubbles
- If users want to find more information, they will be redirected to the current county contentdm site
- Links to other County sites
- A form will be available to users who want to share old pictures with the county

3.2 Development Methods
- We will have an agile/scrum approach to our development
- every group member will have a demo site to their styling and then we will collaborate and choose the best parts to have in our main application
- we will be using Openstreet maps, CONTENTdm, leaflet,HTML5, bootstrap, java servlets, JSPs, AngularJS, Javascript and JQuery to program the site
- The development process is formal with document and code reviews.
- Once a final product is achieved we will run plenty of unit and bug tests, then we will contact the county and proceed with Go-Live process.

Section 4: Architectural Strategies

4.1 Use of a particular type of product (programming language, database, library, etc.)
PLGPA’s client-side is being coded and developed using a combination of HTML 5, CSS3, AngularJS and JQuery. Server-side will use Java MVC using Servlets and JSPs. A local SQL server will be implemented to synchronize with CONTENTdm’s database to maximize performance.

4.2 Reuse of existing software components to implement various parts/features of the system
PLGPA utilizes mobile-friendly HTML layout for proper display on mobile devices as proper component of the application.

4.3 Future plans for extending or enhancing the software
Future plans for PLGPA include:
- Developing appealing web design for all mobile screen sizes that resize properly.
- Implementing additional features such as report function or update lists
- Tracking user visits on developer side

4.4 User interface paradigms (or system input and output models)
PLGPA client-side is being developed using a combination of HTML 5, CSS3, AngularJS and JQuery. Server-side will use Java MVC using Servlets and JSPs. A local SQL server will be implemented to synchronize with CONTENTdm’s database to maximize performance. PLGPA requires Internet for users to get their desired output. PLGPA will output relevant historical imagery.

4.5 Error detection and recovery
Constant backups will be handled via CONTENTdm as well as Los Angeles Public
Library backing up all web content. Recovery will be handled by fixing source error or reestablishing website by using backups created.

**4.6 External databases**
CONTENTdm’s databases will be used to synchronize with a local SQL database for maximum availability and performance.

**4.7 Distributed data or control over a network**
Control for users will be provided by general access to PLGPA webpage via network access. Admin control over network will utilize ContentDM for database control and Java Servlet for HTTP handling/server side coding. Proper authentication is required by username and password for admin control while general users will not.

**4.8 Generalized approaches to control**
Control for users will require network access to PLGPA. Admin will require authentication to ContentDM and Java Servlet control.

**4.9 Concurrency and synchronization**
PLGPA will synchronize with ContentDM. Google Maps, HTML 5.0

**4.10 Communication mechanisms**
Communications to PLGPA will use internet network access and World Wide Web use.

**Section 5: System Architecture**
*This section should is to provide a high-level overview of how the functionality and responsibilities of the PLGPA system were partitioned.*

**5.1 component-1: Server-Side Implementation:**
PLGPA’s Server-side will be implemented on a Linux-based, Apache TomCat server. Main programming language on the server-side will be Java using Java MVC framework using Servlets and JSPs. Additionally, the server-side component will include a local SQL server to be implemented to synchronize with CONTENTdm’s database, this will reduce latency and reduce calls to CONTENTdm’s API.

**5.2 component-2: Client-Side Implementation:**
PLGPA’s client-side implementation will be developed using a combination of HTML 5, CSS3, AngularJS and JQuery.
Section 6: Detailed System Design

6.1 Module 1 - Main Control:

6.1.1 - The Main Control Module (MCM) is the Model view controller (MVC) for PLGPA. It contains the central hub of the server side implementation of PLGPA. The MVC will serve as the main controller for the user interface, OpenStreetMap, LeafLet, Social Media, CONTENTdm’s Modules. The MCM is also the primary communicator to social media platform such as Facebook, Twitter, and Instagram using their respective API’s.

6.2 Module 2 - User Interface:

6.2.1 - The User Interface Module (UIM) provides the GUI and functionality for the user-friendly interface to navigate Public Library digital archives relevant to users’ geolocation. The UIM contains tools such as search bar for user-defined location, pinpoints of digital archives containing thumbnails and a brief historical information. The UIM will also provide ways of interactivity such as email and social media sharing as well as download and the capability to request ultra high resolution photos.

6.3 Module 3- CONTENTdm:

6.3.1 - The CONTENTdm Login Module is a part of CONTENTdm administration web interface. CONTENTdm Server Administration is accessed via a Web interface using a browser.

6.3.2 - The CONTENTdm API, AKA dmwebservices, provides several method calls to get data from the CONTENTdm Server for use in customizing PLGPA application. dmwebservices is a simple wrapper around the CONTENTdm Server API functions, with a few exceptions to improve error handling.

6.4 Module 4- Leaflet - OpenStreetMaps API

6.4.1 - OpenStreetMaps API Module: Leaflet is an open-source JavaScript library for mobile-friendly interactive maps. Weighing just about 33 KB of JS, it has useful mapping features most developers ever need. Leaflet It works efficiently across all major desktop and mobile platforms, can be extended with lots of plugins, has a well-documented API.

6.5 Module 5- Los Angeles County Service Locator API:

6.5.1 - Los Angeles County Service Locator API Module was implemented to the legwork for County residents. Users can find the county services available in their areas and get contact information for local government representatives as well as additional information for all county services and facilities, driving directions to and from LA County facilities.
6.6 Module 6 - Social Media API

6.6.1 - The Facebook API will allow PLGPA application to use social connections and profile information to make the application more involving, and to publish activities to news feeds and profile pages of Facebook, subject to individual users privacy settings. With the API. The API uses RESTful protocol and responses are in JSON format.

6.6.2 - Twitter API Module: The Twitter micro-blogging service includes two RESTful APIs. The Twitter REST API methods allow access to core Twitter data. This includes update timelines, status data, and user information. The Search API methods give developers methods to interact with Twitter Search and trends data. The API presently supports the following data formats: XML, JSON, and the RSS and Atom syndication formats, with some methods only accepting a subset of these formats.

6.6.3 - Instagram API Module: The Instagram API Platform will be used to enable individuals share PLGPA’s content on Instagram. This API will help making PLGPA more discoverable, get digital rights to media, and share media with proper attribution.

Section 7: Graphical User Interface Design

GUI Design.

Section 8: Glossary

An ordered list of defined terms and concepts used throughout the document.

- AKA - Also Known As
- API - Application Program Interface
● DFD – Data Flow Diagram
● JSON - JavaScript Object Notation
● LA – Los Angeles
● MCM- The Main Control Module
● MVC - Model View Controller
● PLGPA – Public Library Geographical Photo Archive
● REST or RESTful - Representational State Transfer
● RSS - Rich Site Summary
● SRD – Software Requirements Document
● UIM - User Interface Module
● XML - Extensible Markup Language