System Design Document

for

Family Photo Album

**Authors:**

Zyad Ahmed,

Rostam Boroumand Rad,

Alina Momin,

Wes Robinson,

Thomas Wang,

Huzaifa Zia

UMBC

10/05/2022

**Table of Contents:**

[**1. Introduction**](#_heading=h.1fob9te) **3**

[**1.1 Purpose of the system**](#_heading=h.3znysh7) **3**

[**1.2 References**](#_heading=h.2et92p0) **3**

[**2. System Architecture**](#_heading=h.tyjcwt) **3**

[**2.1 Overview**](#_heading=h.3dy6vkm) **3**

[**2.2 Subsystem decomposition**](#_heading=h.1t3h5sf) **3**

[**2.3 Software/hardware mapping**](#_heading=h.4d34og8) **4**

[**2.4 Access control and security**](#_heading=h.2s8eyo1) **5**

[**2.5 Global software control**](#_heading=h.17dp8vu) **6**

[**2.6 Boundary conditions**](#_heading=h.3rdcrjn) **6**

[**3. Persistent Data Design**](#_heading=h.26in1rg) **6**

[**3.1 Database Descriptions**](#_heading=h.lnxbz9) **6**

[**4. Subsystem architectures**](#_heading=h.1ksv4uv) **7**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Rostam B. Rad | 10/5/22 | Wrote foundation for SDD | 1.0 |
| Alina Momin | 10/6/22 | Added Software/Hardware Architecture diagrams. Added SBD and descriptions tables/figures. | 1.1 |
| Alina Momin | 11/2/22 | Design clarification and updates, includes any project pivots | 1.2 |

# Introduction

## Purpose of the system

The purpose of this system is to allow users to store and organize photos based on “tags”. The system will automatically add generic tags that sort based on date, location and color. However, users will also be able to add custom tags to be able to customize their photos any way they want. This system design document will explain how the application will be structured. The system architecture will include a diagram of the logical architecture, a top level overview of the subsystems used in this application, any software components that will be used, as well as security measurements taken when designing the family photo app. In addition to the system architecture, this document will cover the database that will be used to store user data and a complete overview of the subsystem architecture.

## References

We will be using various Python libraries and APIs to create our frontend and backend.

The system is composed of multiple subsystems that consist of the graphical user interface and the database. The graphical user interface will be created using a lightweight flask framework. To implement the design, Flask documentation will be reviewed.

To create the database, we will be using SQL and will be following documentation to understand the structures that would best fit our data.

# System Architecture

## Overview

The Gizmo: Gizzy Gallery is a virtual photo album that allows users to search, sort and view their photos in the gallery. Users will be able to access the User Interface to interact with the system. Users will create an account to upload files to the system. The user data and file metadata will be stored in the Database.

## Subsystem decomposition

**Gizmo System:** Responsible for maintaining how the graphical user interface and database communicate with each other.

* **Graphical Interface:** Responsible for providing an intuitive/enjoyable experience to the user to be able to interact with the system. The GUI is able to receive and make calls to the database to access information.
  + **Photo Gallery:** Responsible for displaying user’s photos in an attractive way. Users will be able to scroll through stored photos with ease and be able to categorize based on tags.
  + **File Transfer:** Responsible for allowing users to share photos with other users and to upload existing photos to be stored in the database.
  + **Login Prompt:** Responsible for authenticating users accounts and displaying users stored data upon authentication.
* **Database:**  Responsible for storing user accounts, user photos, and user tags. Able to make and receive calls from the GUI.
  + **Photo Metadata:** Responsible for containing text information pertaining to an image. Includes tags.
  + **User Information:** Includes user account information (username and password), and stored photos
* **Users:** Responsible for interacting with the graphical user interface

## Software/hardware mapping

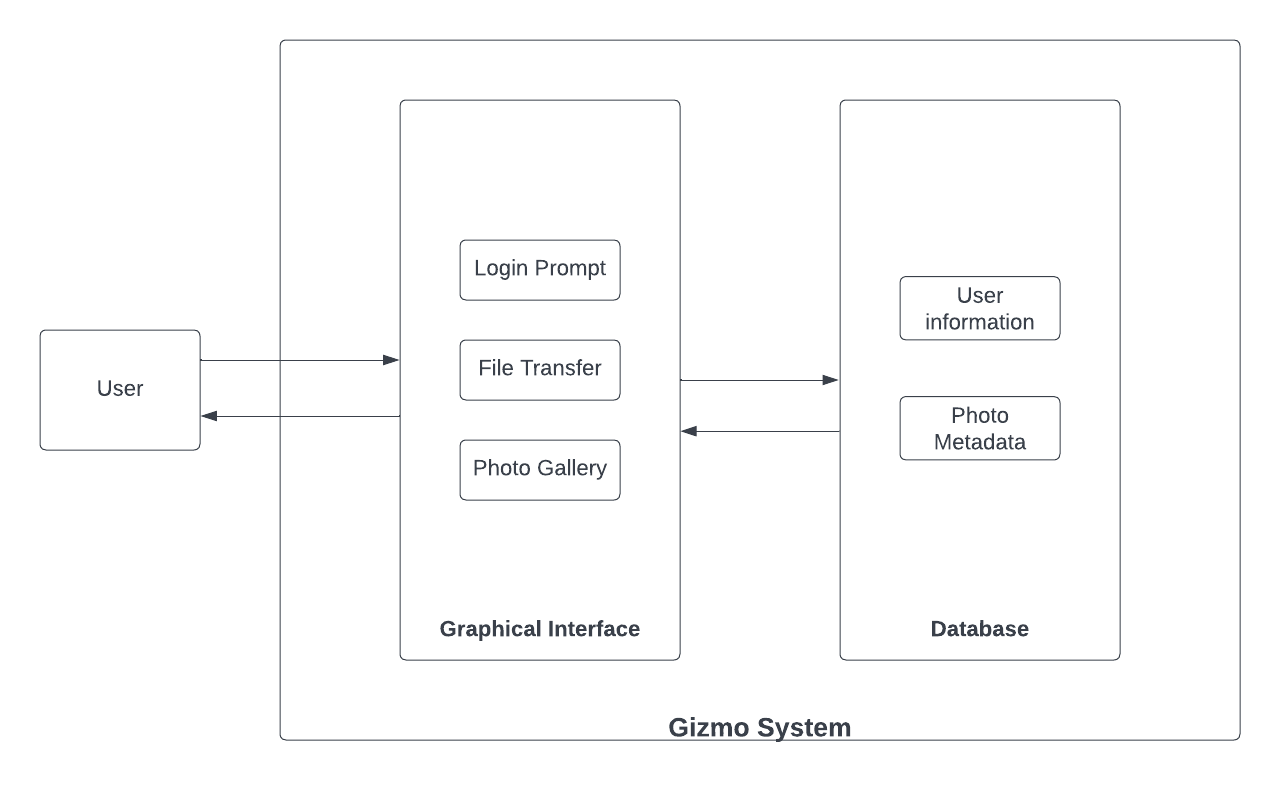


Figure 1: System Boundary Diagram

The diagram above shows a top-level software architecture for the system under development. The system boundary diagram illustrates the internal and external systems for the Gizmo system. There are two main systems illustrated below. The Graphical interface is the central hub that users will interact with. Here they will have login options for the user. After login, the user is able to transfer files and access their photo gallery. All of the user information and photo metadata lives in the database and can be viewable in the user accessible photo gallery.

Subsystems are assigned to various hardware, specifically a graphical user interface. Software subsystems include the database.

*Table 1: Hardware List*

| Hardware | Description | Subsystem |
| --- | --- | --- |
| Graphical Interface Display | A display, on a computer system that can visualize the user’s interaction with the Photo Gallery login prompts and file transfer functions. | Graphical User Interface |

*Table 2: Software Architecture*

| File | Class/Function |
| --- | --- |
| file\_transfer.py | file\_metadata |
| graphical\_user\_interface.py | user\_login |
| user\_metadata |
| photo\_database\_metadata |
| user\_information.py | user\_database |
| photo\_metadata.py | photo\_database |

## Access control and security

Access to the albums will be given to the software administrators as well as the user who is tied to the album through the created account. With the inclusion of other users and their own individual accounts, it is our responsibility to ensure that the privacy of one’s photos are kept intact and permissions are set in place to not allow other users to intrude on other users’ albums. This is mainly handled through the sign in process which will make it so that each user creates a unique username and password for access to their personal album. Administrators are also given universal access to allow them to solve any issues that might arise with the application.

## Boundary conditions

On startup, the user will be prompted to log into an existing account or to create a new account. Once logged in users will be able to upload photos to the system and error handling will ensure the correct file type is being uploaded. The uploaded photos will then be stored in our database and kept safe upon shutdown so the user will be able to come back and view their previously uploaded photos.

# Persistent Data Design

## Database Descriptions

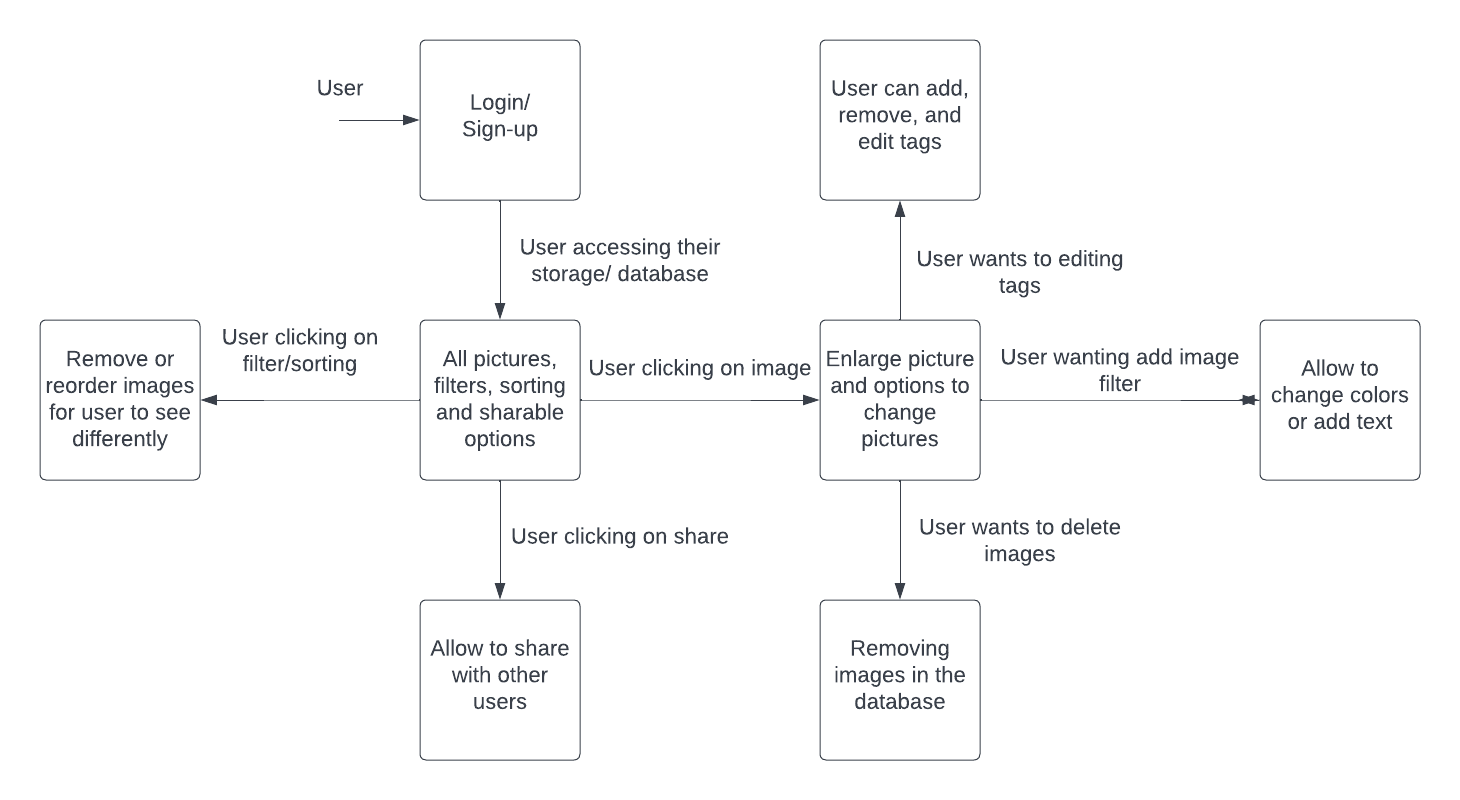


Figure 2: Data Flow Diagram

Figure 2, above, shows the system data flow structure within the SQL database. Various tags will be associated with the Photo that is being uploaded.

Figure 2, above, shows the system data flow structure within the SQL database. Various tags will be associated with the Photo that is being uploaded. The user will be able to login and sign-up for an account. Their Gizzy Gallery account will allow users to upload and store photos in the gallery. The user will have the ability to tag and sort images. The Gizzy Gallery will allow users to add, remove and edit tags while also allowing users to view their photo gallery.

# Subsystem architectures

The Gizmo: Gizzy Galley consists of two main subsystems, the Database and the User Interface. The database management system is where user information, photos, as well as metadata information about the photos are stored. The User interface is the method by which the user interacts with the Gizmo: Gizzy Gallery. The graphical interface will allow the user to login, upload/view files, and tag files.

The user interface will allow the user to create an account or login to an account via the React webpage. The account will be Authenticated using the user's Google account. Once the account is created, the user will be able to login and access their gallery. The gallery will allow users to upload photos. The metadata associated with the photos will be stored in our database. The users will also be able to tag the photos with arbitrary values, the tags will also be metadata values.