|  |
| --- |
| Department of computer science & Engineering  University of Nebraska—Lincoln |
| Software Engineering |
| CSCE 361 – Software Engineer Project |
|  |
| **[Cody Bodfield, Jiachun Han, Yan Xin Lee, Toan Nguyen]** |
| **6/17/2014**  **Version** |

# Revision History

[This table documents the various major changes to this document]

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Description of Change(s) | Author(s) | Date |
| 1.0 | Initial draft of this design document |  | 2014/06/17 |
|  |  |  |  |

|  |
| --- |
| This document describes a system design that will allow qualified users to upload pictures they took around campus, share them with other members via placemarkers on Google Maps. |

Contents

[Revision History 1](#_Toc349390336)

[1. Introduction 4](#_Toc349390337)

[1.1 Purpose of this Document 4](#_Toc349390338)

[1.2 Scope of the Project 4](#_Toc349390339)

[1.3 Definitions, Acronyms, Abbreviations 4](#_Toc349390340)

[1.3.1 Definitions 4](#_Toc349390341)

[1.3.2 Abbreviations & Acronyms 4](#_Toc349390342)

[2. Overall Design Description 4](#_Toc349390343)

[2.1 Alternative Design Options 4](#_Toc349390344)

[3. Detailed Component Description 4](#_Toc349390345)

[3.1 Class/Entity Model 4](#_Toc349390346)

[3.1.1 Component Testing Strategy 5](#_Toc349390347)

[3.2 Class/Entity Model 5](#_Toc349390348)

[3.2.1 Component Testing Strategy 5](#_Toc349390349)

[3.3 Database Interface 5](#_Toc349390350)

[3.3.1 Component Testing Strategy 5](#_Toc349390351)

[3.4 Changes & Refactoring 5](#_Toc349390354)

[4. Additional Material 5](#_Toc349390355)

[5. Bibliography 6](#_Toc349390356)

# Introduction

This document describes the design of a system that will allow qualified users to take pictures around main city campus, upload them, and share them with other members through placemarkers on Google Maps.

## Purpose of this Document

The purpose of this document is to outline each step of this project, which includes design, coding and implementation and integration and testing.

## Scope of the Project

This system is developed for qualified users to manipulate photos they take around city campus. It provides four functions. The system checks users’ twitter credentials for login. After qualified users login, they can upload pictures, comment on the pictures and the pictures should be extracted from Google Maps. The system allows users to view photos by selecting markers placed at locations of uploaded photos. Moreover, users can search photos through three conditions. At last, users should be able to delete photos and comments they made.

## Definitions, Acronyms, Abbreviations

### Definitions

[Define any terms that require a definition—domain specific terms, non-standard terms, or terms that are used in non-standard ways]

### Abbreviations & Acronyms

[Define all abbreviations and acronyms used in this document here. This relieves you of the need to define such things within the context of the document itself and provides an easy reference for the reader.]

APA – American Psychological Association

MLA – Modern Language Association (of America)

UAV – Unmanned Aerial Vehicle

# Overall Design Description

In keeping with the OOP paradigm for this application development the use of unique classes is essential. Relevant data, methods, and functionality are built into respective classes based on designing good encapsulation enforcement. The current primary classes include: Comment, Picture, Profile and Results.

The Result class is a generic class. It allows us to push success or error messages from our operations down to the user. Since functions can only return one object at a time, it's hard to know why something didn't work if you only have a True or False coming back to you. The Results class holds (at minimum) at boolean for success or failure, and a string message.

## Alternative Design Options

[If applicable, describe and discuss alternative design options that you considered and discuss why they were not chosen. What advantages and disadvantages do the alternatives provide and what advantage/disadvantages do the chosen design elements provide. Provide some justification for why the chosen elements’ advantages/disadvantages outweighed the alternatives]

# Detailed Component Description

Classes are used to represent instances of given objects. Object creation is handled by constructor methods in the respective classes via provided data. The provided data values are encapsulated and belong to the class they were used to create. By default, all member variables of classes are set to private so that variable interfacing is handled by getter and setter methods contained with the parent classes.

## Database Design

The ER Diagram presented in Figure 1 represents the application’s database schema. This diagram denotes the primary tables with their respective fields and relationships to other tables in the schema. The database schema is loosely based on the Java class entities presented in the application. Careful planning was done to enforce separation of distinct tables and ensure encapsulation of proper information into their respective, distinct tables.

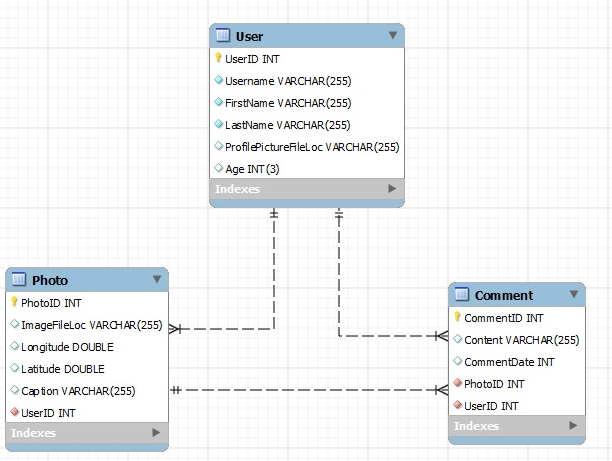


Figure 1: ER Diagram

### Component Testing Strategy

[This section will describe your approach to testing this particular component.]

## Class/Entity Model

This part will be illustrated by a UML diagram at the final version. Following is the first draft description.

There will be four classes for this project, Comment, Picture, Profile, and Result class. The Comment class is used for users to do comment on their photos and on others’ photos. It consists of six variables: \_sConnection, \_sComment, \_sPicture, \_dTime, \_sCommentID, and \_sUersID. The Picture class is used to contain pictures users uploaded. It has variables of \_sConnection, \_sLongititude, \_sLatitude, \_sCaption, \_sPictureID, \_lComment (as list of String), \_sUserID and \_sImagePath. Then we will also have Profile class. It has variables of \_sConnection, \_sUername, \_sFirstName, \_sLastName, \_sUersID, \_nAge, \_sProfilePicturePath, \_lCommentList (as list of Comment), \_lPictureList (as list of Picture). At last, we have Results class. Results class allows us to push success or error messages from our operations down to the user.

Figure 2: UML Diagram

### Component Testing Strategy

[This section will describe your approach to testing this particular component. Describe any test cases, unit tests, or other testing components or artifacts that you developed for this component. What were the outcomes of the tests? Did the outcomes affect development or force a redesign?]

## Database Interface

[This section will be used to detail phase IV where you modify your application to read from a database rather than from flat files. This section will detail the API that you designed—how it conformed to the requirements, how it worked, other tools or methods that you designed to assist, how it handles corner cases and the expectations or restrictions that you’ve placed on the user of the API. In earlier phases this section may be omitted or a short note indicating that details will be provided in a subsequent revision of this document.]

### Component Testing Strategy

[This section will describe your approach to testing this particular component. Describe any test cases, unit tests, or other testing components or artifacts that you developed for this component. What were the outcomes of the tests? Did the outcomes affect development or force a redesign?]

## Changes & Refactoring

[During the development lifecycle, designs and implementations may need to change to respond to new requirements, fix bugs or other issues, or to improve earlier poor or ill-fitted designs. Over the course of this project such changes and refactoring of implementations (to make them more efficient, more convenient, etc.) should be documented in this section. If not applicable, this section may be omitted or kept as a placeholder with a short note indicating that no major changes or refactoring have been made.]

# Additional Material

[This is an optional section in which you may place other materials that do not necessarily fit within the organization of the other sections.]

# Bibliography

[This section will provide a bibliography of any materials, texts, or other resources that were cited or referenced by the project and/or this document. You *must* consistently use a standard citation style such as APA or MLA (good reference: <http://www.cws.illinois.edu/workshop/writers/citation/)>.]

[1] *Citation Styles*. (n.d.). Retrieved December 19, 2012, from [http://www.cws.illinois.edu/workshop/writers/citation/](http://www.cws.illinois.edu/workshop/writers/citation/))

[2] Eckel, B. (2006). *Thinking in Java* (4th ed.). Prentice Hall.