Richard Compton

Zachary Denny

Samuel Khalil

# **Database Project: Phase 2**

## Problem Statement

We are constructing an application that will serve as a resource for students, professors, and visitors to university or college campuses across Michigan. The app’s main function is to assist campus-goers in discovering the most interesting, provoking, or action-packed locations on a university or college campus. One database or more is being constructed to accomplish the following:

* Store university or college data, such as name, address, population, mascot, et cetera.
* Hold names, addresses, descriptions, and pictures of specific locations on a school campus.
* Store user information such as name, age, occupation, user status, et cetera.

If a database was not utilized, this application would be cluttered, difficult to navigate, and painfully slow. Utilization of a database also allows this application to be dynamic real time allowing users to get the most up-to-date info about a campus. Thus, at least a single database is necessary to avoid these issues. As of the end of Phase 2, we have settled on a name for the application: MyCollege.

## System Requirements

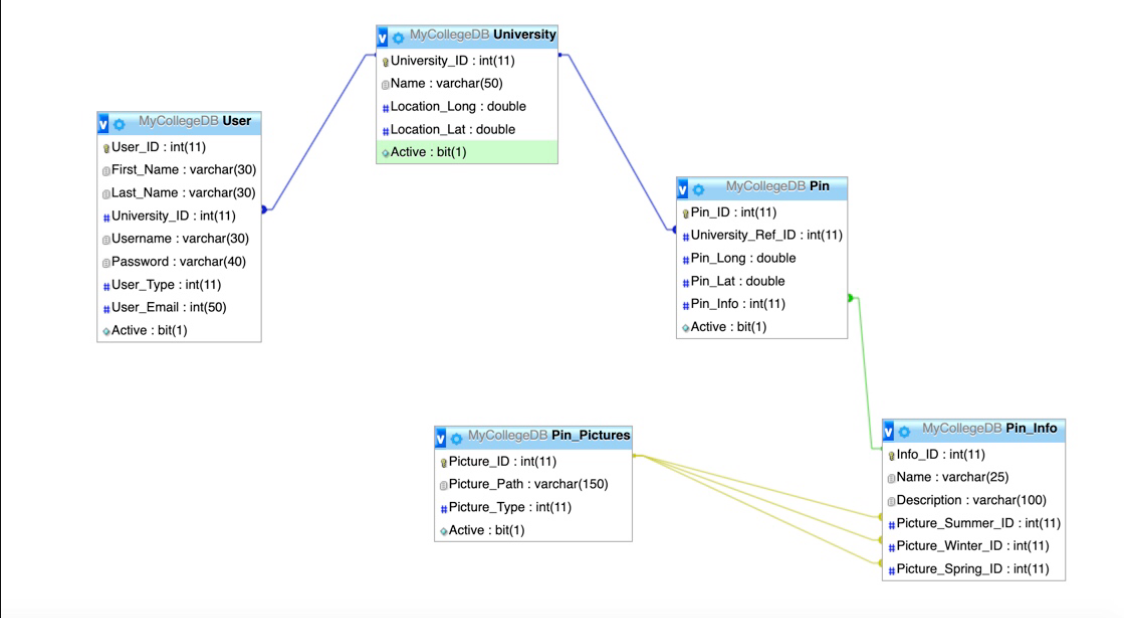
The application will be accessible on the Apple Application Store. It will be free and downloadable for users of Apple products such as an iPhone or iPad. Users will then be able to open the application on their technological devices and utilize all of its features. Input data will be added through a web portal for university admins to add their campuses key locations and hot spots. The following features are planned to be available:

* The application shall allow users to discover interesting locations on a university’s campus.
* The application shall utilize GPS technology.
* The application shall allow universities or colleges to control their own accounts.
* The application shall have an administrator that approves of all locations in the database.

The application will not serve as a substitute to other already available services like Google or Apple Maps because it will not show users how to navigate to certain locations. Instead, the app will advertise and make students, professors, and visitors aware that these unique locations exist.

## Conceptual Database Design

This is the EER Diagram for the application:



## Functional Requirements

The application frequently interacts between different entities. Many retrievals and updates are made, entities are accessed, and modifications are made. Here are a few example database transactions:

* The database updates when users enter information about universities, locations, or themselves.
* The database may or may not update when a user submits a LOCATION\_DESCRIPTION, pending approval from the admin.
* UNIV\_ID will be accessed when searching for locations of in the area.
* Specific users will be connected to specific universities using a UNIV\_ID.
* Every time a user logs in to the application, all of the databases will be accessed. The user entity will be required for login and the university and location entities will be required to populate the application’s data.

This list is by no means exhaustive. The database will constantly be altered and accessed while the application runs, leading to a plethora of other transactions.

## Logical Database Design

#### University Entity

**Primary Key**: University\_ID

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| University\_ID | int | Unique identification number for a university. |
| Name | varchar | Name of a university. |
| Location\_Long | double | Longitudinal location of a university on a map. |
| Location\_Lat | double | Latitudinal location of a university on a map. |
| Active | bit | To set record inactive or active to keep record. |

#### User Entity

**Primary Key**: User\_ID

**Foreign Key**: University\_ID

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| User\_ID | int | Unique identification number for a user. |
| First\_Name | varchar | First name of a user. |
| Last\_Name | varchar | Last name of a user. |
| University\_ID | int | Unique identification number for a university. |
| Username | varchar | Username of a user. |
| Password | varchar | Password of a user. |
| User\_Type | int | Type of user. |
| User\_Email | int | Email of user. |
| Active | bit | To set record inactive or active to keep record. |

#### Pin

**Primary Key**: Pin\_ID

**Foreign Key**: University\_Ref\_ID

**Foreign Key**: Pin\_Info

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| Pin\_ID | int | Unique identification number for a pin. |
| University\_Ref\_ID | int | Unique reference number for a university. |
| Pin\_Long | double | Longitudinal location of pin on a map. |
| Pin\_Lat | double | Latitudinal location of pin on a map. |
| Pin\_Info | int | Information attached to a pin. |
| Active | bit | To set record inactive or active to keep record. |

#### Pin Info

**Primary Key**: Info\_ID

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| Info\_ID | int | Identification number for a pin. |
| Name | varchar | Name of a pin. |
| Description | varchar | Description of a pin. |
| Picture\_Summer\_ID | int | Identification number for summer. |
| Picture\_Winter\_ID | int | Identification number for winter. |
| Picture\_Spring\_ID | int | Identification number for spring. |

#### Pin\_Pictures

**Primary Key**: Picture\_ID

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Type** | **Description** |
| Picture\_ID | int | Identification for a picture. |
| Picture\_Path | varchar | Path for a picture. |
| Picture\_Type | int | Type of picture. |
| Active | bit | To set record inactive or active to keep record. |

## Application Programs Design

MyCollege performs a myriad of transactions between its various parts. Here is a complete specification of them.

**IOS:**

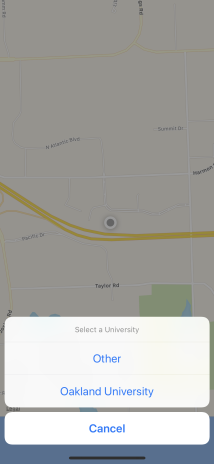
* When a user loads the app, the application retrieves all active universities from the university table in the database.
* When users select a university, the pins for that university is loaded onto the map from the pin table along with that pins linked info from Pin\_Info and Pin\_Pictures tables to be used for a Pin’s data.

**Web Portal:**

* The web portal will be accessed through the internet by a university admin and they can log in with their assigned login information.
* Each admin will have the permission to edit all pins and data associated to their profile that the university they have been assigned to.
* When admins add a pin, data will be stored using procedures in corresponding tables that are linked together.
* Pictures for each pin will be stored in a network drive corresponding to the database with a stored URL.

## User Interface Design

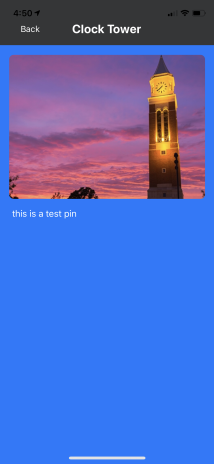
Currently, the user interface is intact. As said before, it is for use on Apple products, such as an iPhone. Here are sample screenshots of the current user interface on an iPhone.



**Picture 1:** A user can select a university to explore.



**Picture 2**: A user can choose a location on a university campus.



**Picture 3**: A user can read information about a location.

\*\*Proposed website design



More pictures of the IOS/online portal user interface will be included in Phase 3.

## Implementation and Testing Plans

Testing and implementation will start with test pins located in random areas for testing. For final implementation, each member of the group will have specific buildings on Oakland University's campus assigned to them to collect the data for the pins. Collection of data will be done using a custom made testing application to help. A picture of the university map shows how we plan to split up this work.



\*\*Blue Pins: **Zachary**

\*\*Red Pins: **Sam**

\*\*Orange Pins: **Richard**

## Estimate of Effort

The expected effort in terms of person-weeks is eight weeks. The logic behind the calculation is outlined below:

* Two weeks to create the applications requirements document.
* Three weeks to create the infrastructure of the application and the database itself.
* Three weeks to implement the database and finalize the entire application.

This calculation is ideal. Adjustments may be needed if unexpected errors or accidents occur.