

**Team Philanthropy:**

**Laker Legacies**

**Design Team**

Kyle Peltier peltikyl@mail.gvsu.edu

Matthew Williams william6@mail.gvsu.edu

Samantha Williams willsama@mail.gvsu.edu

**Client**

Danielle DeWitt

Donor Relations and Stewardship Manager

danielle.dewitt@gvsu.edu

CIS 467 - Senior Project

Winter 2014

Professor Jonathan Engelsma

**Introduction**

The Laker Legacies project is meant to convey the history of Grand Valley State University and give more meaning to the phrase “Laker for a lifetime.” Specifically, the project’s purpose is to present the history of the major donors or contributors to the university. The application will focus on the donors whose names appear on buildings and monuments throughout Grand Valley’s campuses.

The goal of this application is to make it easy for the Grand Valley community, as well as the general public to learn the history of the donors whose names are on buildings both nearby and on different campuses. The Android Laker Legacies application will be a close sibling of an existing iOS application. Therefore, much of the GUI design has already been conceived.

**Features**

Since the Laker Legacies application has a sibling application for iOS, the team wants to make the applications similar in appearance and functionality. The Laker Legacies application has several features that are split into the following view-groups: Home, Map, Directory, Near Me, and Donate. Each view-group is meant to accomplish a specific goal.

*Home*

This view will cycle through images of major building contributors, welcoming the user to the application. A potentially additional feature is to add a trivia section, where small trivia texts could cycle along with the images.

*Map*

The map view will query Google Maps to highlight all GVSU buildings/monuments in the application’s database relative to the user’s location.

*Directory*

GVSU buildings will be listed along with their associated contributors in the Directory view. Selecting an item in the list will take the user to a detailed description of the major contributors to the selected building. Two additional features may appear in the Directory view. The first is to add the ability to sort the list by building name, contributor name, city, or campus. Secondly, a scavenger hunt (Easter egg) feature could be included here to create an interactive experience between the user and the buildings.

*Near Me*

This view will access the user’s GPS location and display a list of buildings sorted in order of distance from to the user.

*Donate*

Finally, the donate view will direct the user to a GVSU webpage where the he or she can donate money to the university. This view is meant to encourage the user to give back to the community and their alma mater.

**Environment**

Before approaching the steps of development, the first task necessary for the team is an established repository, a centralized location for project assets. GitHub provides a free versioning service that will be essential throughout the design and development processes of the Laker Legacies application. GitHub provides a friendly environment for team members to contribute source code, documents, and other files to the central project. It is a powerful versioning tool which will help keep the team on track and up to date with one another and the application as a whole. Headquarters of the Laker Legacies application resides at the url: <https://github.com/william6/Android_Laker_Legacy>.

The well-known Eclipse Integrated Development Environment (IDE) will be the most frequented tool during the development phase. With the Android plug-in equipped, Eclipse evolves into an IDE with all of the necessary tools to build an Android application, including the Android Debug Bridge (ADB), Android Virtual Devices (AVD), and a multitude of supported Android Software Development Kits (SDKs). Although built on top of C, Android is mainly programmed in Java, with a small integration of XML. Java will be used to code all of the functions of the application. The integrated XML tools assist in development for the visual aspect of the application such as screen layouts and organization. The provided Eclipse tools then links the two together into a synced application.

Ice Cream Sandwich, or Android 4.0, will be the lowest supported Android operating system. This is an older version of android, but the majority Android users have a device that supports this OS version or better. With Android 4.0, the application will be written using Android API level 14. In addition to the standard Android SDK, a Google Maps Android API will also be included in the application to fulfil the goal of the “Maps” feature.

A few other tools will be used to help assist the team with development and they are the DIA drawing tool, SQLite Database & Firefox plugin, and Google Drive. The DIA (diagram) drawing tool provides a neat environment to build relationship schemas. This tool will be particularly useful for conceptualizing the proposed database of the application (to be discussed in detail in section “Software”). SQLite is a database language supported by Android. Firefox, the web browser, provides a free SQLiteManager plugin which will be useful for building and visualizing a database. Finally, Google Drive will be a significant part of development. Google Drive offers a real-time environment where team members can communicate and formulate project ideas remotely. This is extremely helpful because each team member has a very different day-to-day schedule. We will collaborate on documents, such as this one, and then add the finalized documents to our Git Repository.

**Software**

*Structure*

Agile software development model is the best framework for the development of this project. Agile programming focuses on rapid prototyping while also supporting changes in application requirements during the entire development phase. We think this is the best approach to the application because agile programming works well with smaller projects such as the Laker Legacies application. Agile programming will also ensure to our customer that development is progressing smoothly. With the frequent updates to the customer on project progress, the team is able to receive profitable feedback even in the early design stages. This allows the team to have a great understanding of the client’s mental model of the application which, in turn, allows the developers to design the application to the fullest intention of the client.

The waterfall software model is an alternate development method reviewed by the team. The waterfall software model is good for applications with a well-defined function from the first step of the design; that is, the waterfall model is not lenient on malleable projects. In the case of Laker Legacies, the application must be malleable because some features may be implemented into the application later in the design phase - such as trivia and a scavenger hunt game.

Several software tools will be necessary to accomplish the goals of the application. First, a database will be used to house information about buildings and donors. A SQLite database will ultimately make it easier to design and create the Laker Legacy Directory. The application will pull the building names from the database and display them using the Google Maps Android API. For the “Donate” tab, the application will use Android WebView to display the GVSU Donor webpage inside of the application frame.

*Database*

The design team believes the best approach to accomplish the functional Directory is to house a database within the application. The database will contain all of the information about various monuments of Grand Valley, and their major contributors. It will provide the ability to easily add or edit donors and buildings in the future without having to edit the code for the application. It will also allow for easy lookup and sorting. The database is scalable, so should the list of donors and buildings grow, the database is more suitable than storing the information inside the application.For the sake of the application, the database will also store information about related image files associated with both monuments and contributors. This section defines the detailed description of the database.

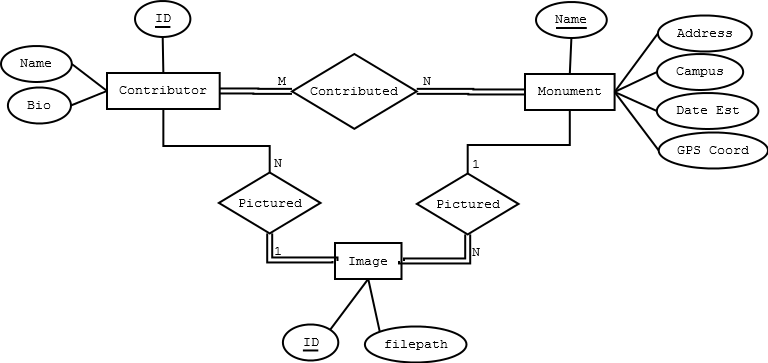
A contributor’s name and short biography should be recorded in the database. Although it is unlikely, it is possible for two individuals to have the same name. Therefore, in order to uniquely identify each contributor, each contributor should have his/her own unique ID number.

Each monument has a unique name. The database should record its address, campus, date established, and its GPS coordinates.

All donors in the database have donated/contributed toward at least one monument. Likewise, all buildings in the database have at least one donor. A donor can contribute toward multiple buildings. Likewise, multiple buildings could have multiple donors.

For visual effect, the database will hold references to image files of both contributors and monuments. Not all monuments/contributors will have a picture; all pictures referenced in the database, however, will refer to a contributor or a monument. Two different contributors could share the same image (this is because some contributors are unpictured so, instead, an image of the contributed monument is shown. Because of this, if two unpictured contributors contributed to the same building, they could reference the same monument). A monument and a contributor can reference the same image. A single image cannot refer to multiple monuments. However, a monument can have multiple images of itself in the database.

*Database Relational Schema*



*Database Table Layout*

**MONUMENT** ( Name , Address , Campus , Date\_Est , GPS )

**DONOR** ( Donor\_ID , Name , Bio , Image\_ID )

**IMAGE**  ( Image\_ID , Filepath , Monument\_Name )

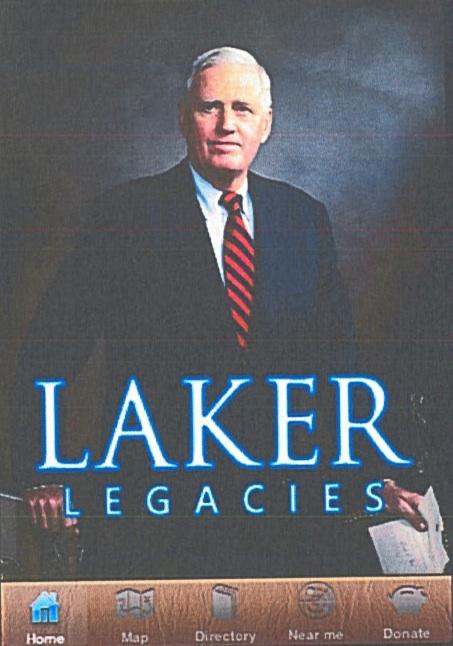
**MONUMENT\_DONORS** ( Building\_Name , Donor\_ID )

**GUI**

This section describes the current organization and design of the Laker Legacies application for iOS. The design team has some wiggle room as to the design of the Android version of the application, but the current screenshots provide a strong visual of the functionality and layout of the application.

*Home*

The user interface will be simple to use. There will be a home screen that the application boots up in to. From this home screen, the user will be able to navigate through 4 other tabs located at the bottom of the screen. These include: Maps, Directory, Near Me, and Donate. Each tab will have a new screen that is presented once it is clicked.



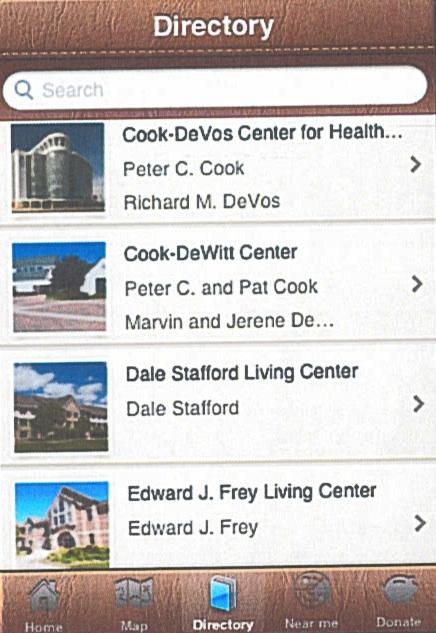
*Maps*

Google Maps will open up as a fragment of the screen. This fragment inside the application has the full functionality of a regular Google Maps Application. When the user looks and navigates through the map, navigation icons will be located throughout the campuses where monuments and buildings are located. From here, the user will be able to click on a building to learn more information about that building and the donor.



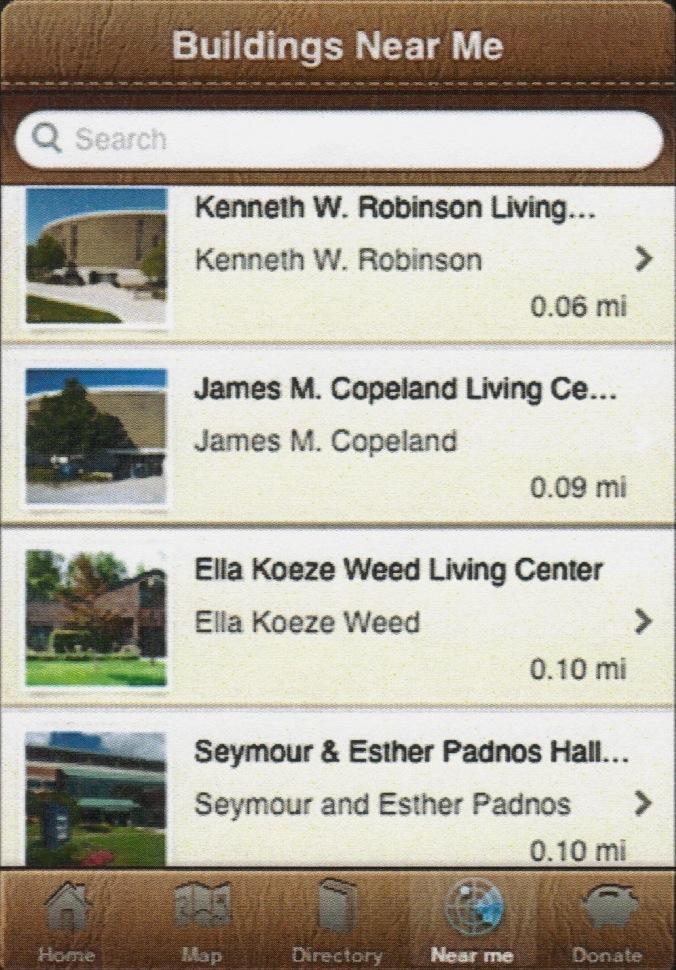
*Directory*

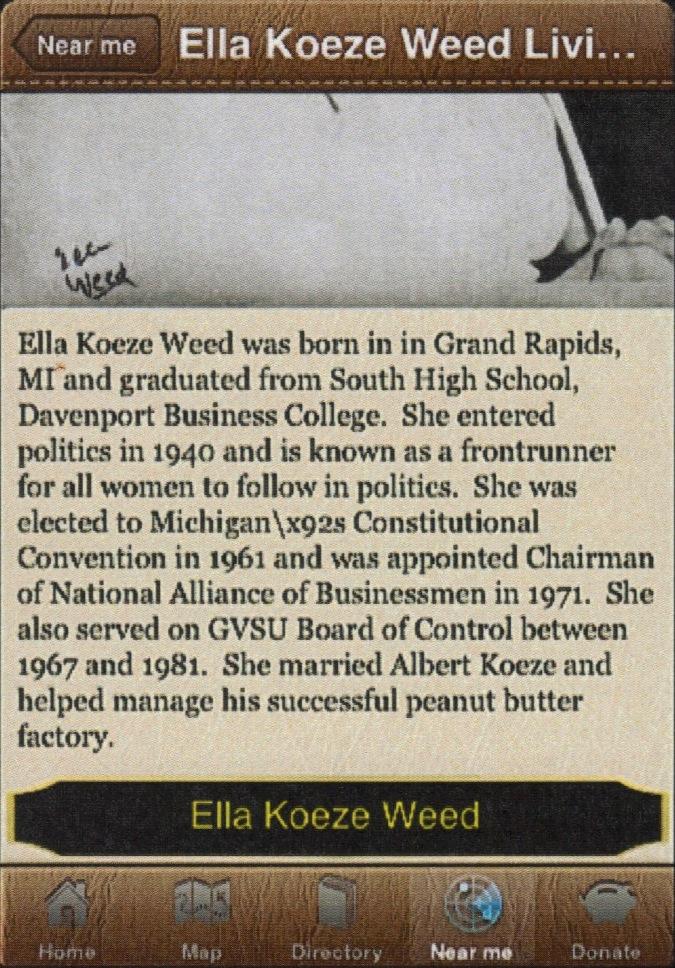
The Directory tab will function as a list with small icons. These icons will be various pictures of the buildings and monuments. The name of the building and the name of the donor(s) will be listed below. Clicking on a building will open up a new screen that has a larger image of the building and images of the donor(s). If the user is to swipe, a description of the donor(s) will appear in place of the picture.



*Near Me*

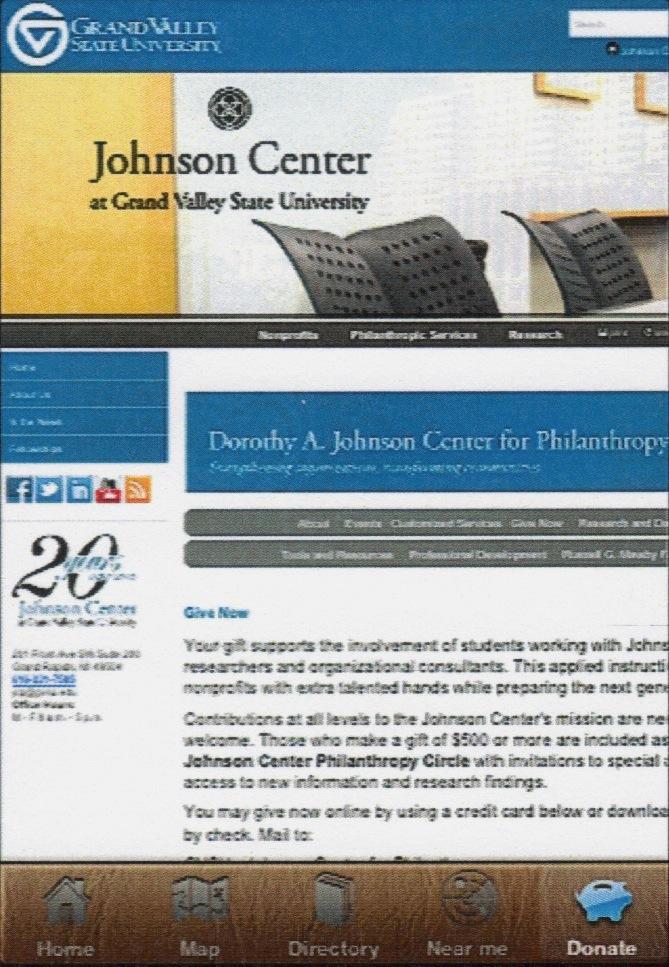
The Near Me tab will function similar to the Directory tab, but it will put the buildings in order in correspondence to your location. This will use the GPS feature of the phone. In addition to the name of the building and donor(s), how far the building is away (in miles) will be put in the bottom right corner of the list item.



*Donate*

The Donate tab will navigate the user to GVSU’s donation page.



*Back-button convention*

The only soft-key button that is currently present in the GUI is the Back button. Since our team is porting this application to Android from iOS, we plan on removing the Back button given that Android phones already have a Back button built in.

**Testing**

Initially we will do an internal test within our project group. There are two android devices in our group and we will split the testing into conventional use of the application and non-conventional use. One person will use the application like it is supposed to be used. Making progression through the application in a manner that makes sense. The other person will be trying to break the application. This will include randomly selecting buttons, going back and forth on a screen, and just trying whatever they can to see if it will freeze or crash.

As development progresses and the application becomes more solidified, the team will test the application around various campuses and in various locations. Additionally, the team will extend the application to the hands of volunteer users. In this way, the application will be tested by real users and the design team can better understand an average user’s movements through the application. All of these tools and methods will help the design team find bugs and correct issues within the application.

**Implementation**

*Division of Labor*

Kyle Peltier

Kyle will be focussing on the database for the application. This includes database design and creation. Along with the these, Kyle will be doing database handling in the code as well as construct quality database queries - efficient for data processing for the application. Kyle will work as needed throughout the application aside from the main task of database management.

Matthew Williams

Matthew will be focussing on the implementation of Google Maps into the application. Besides this main task, Matthew will be working with incorporating similar functions that are present in the iOS Laker Legacies Application into the Android version. This will include GUI design and also backend coding. Finally, Matthew will be working with testing the prototypes as they arise.

Sam Williams

Sam’s main focus will be the Graphical User Interface (GUI) of the application and navigation through the app as a whole. This includes the creation of multiple layouts for different mobile devices as well as creating links between views within the app.

*Project Timeline Estimation*

|  |  |  |  |
| --- | --- | --- | --- |
| **Task**  **ID** | **Task Name** | **Prerequisite** | **Estimate**  **Duration (days)** |
| **0** | Initiation |  | 0 |
| **1** | Create XML template for screen views | 0 | 3 |
| **2** | Convert database info document into an XML file | 0 | 1 |
| **3** | Create code to parse XML data file to populate database | 0 | 7 |
| **4** | Rename all resource files to have convention and place in application | 0 | 1 |
| **5** | Include Google Maps APIs into application | 0 | 2 |
| **6** | Write all queries for database | 0 | 6 |
| **7** | Create HomeView | 1 | 3 |
| **8** | Create Maps View | 1 | 4 |
| **9** | Create Directory View | 1 | 5 |
| **10** | Create Near Me View | 1 | 4 |
| **11** | Create Donate View | 1 | 3 |
| **12** | Completely create a database (.db file) | 3, 2 | 10 |
| **13** | Link all application views together programmatically (buttons & gestures) | 7 - 11 | 10 |
| **14** | Complete functionality of Home View | 4, 6, 12 | 5 |
| **15** | Complete functionality of Maps View | 5 | 15 |
| **16** | Complete functionality of Directory View | 4, 6, 12 | 15 |
| **17** | Complete functionality of Near Me View | 4, 5, 6, 12 | 10 |
| **18** | Complete functionality of Donate View | 1 | 7 |
| **19** | Complete GUI layout of Home View | 7 | 8 |
| **20** | Complete GUI layout of Maps View | 8 | 5 |
| **21** | Complete GUI layout of Directory View | 9 | 10 |
| **22** | Complete GUI layout of Near Me View | 10 | 8 |
| **23** | Complete GUI layout of Donate View | 11 | 5 |
| **24** | Complete work up to functional prototype (debugging) | 14 - 23 | 20 |
| **25** | Start mobile testing | 24 | 4 |
| **26** | Have volunteer users test the application | 24 | 7 |
| **27** | Final tweaks/fixes | 25, 26 | 20 |

**Design Critique Discussion**

We told the class that the iPhone app currently has all the donor and building information stored in the app itself. And they agreed with our decision to use a database to store the information. They gave us suggestions for how to input additional donors and buildings after the completion of the app. They suggested using a Parse API which would be simple for non-technical users because of it’s spreadsheet-like interface.

Upon presenting our design, classmates suggested that we completely remove the back button in the existing app interface due to the redundancy of already having a physical button. They also suggested that tabbed UI is outdated and consumes a lot of screen real estate. One suggestion was to use dots centered at the top or bottom of the page to indicate that there are more pages available for view.

The critique led us to the idea of showing the distance of the buildings from the user’s current location in the original Directory page in the same manner as it is displayed in the Near Me page then completely eliminating it. Because there is an existing iPhone application, we will consult our customer to see how comfortable she is with our proposed changes to the UI.

We mentioned that we wanted to make the application more interactive, and we got an idea from the class to send the user on a tour of the buildings nearby. This will tie in perfectly with our sponsor’s idea for a scavenger hunt. We also thought of having the user take a fun quiz after reading the donors’ bios. Our classmates thought the app would be perfect for parents with children in orientation, who want to take a tour of the building, but don’t want a tour guide to accompany them.

As far as GUI changes, there was a consensus that we should change the dated, leather feel of the application and update the existing tab icons to have a more modern feel.