Grand Valley State University

Laker Legacies

Final Report

Prepared by:
Kyle Peltier, Matthew Williams, Samantha Williams

Date: 04/24/2014

Sponsor: Danielle DeWitt

CS 467

Professor Jonathan Engelsma

Abstract

Laker Legacies is a mobile application created for the Android Platform. The audience of our application is the Grand Valley community, including Grand Valley students, Grand Valley alumni, and their families. The application is a tool that will be used for the Laker for a Lifetime program and will also be presented to new students through Transitions. The application communicates the history of Grand Valley by presenting its buildings and its major contributors who helped shaped it into the university it is today. Laker Legacies has a simple layout, comprised of four navigational tabs located at the bottom of the screen, which eases app exploration. From left to right, the tabs are "Home", "Directory", "Near Me" and "Give". The "Home" screen welcomes the user to Laker Legacies with a slideshow of donor images. Each building on campus named after its major donor(s) are listed in the "Directory" page. When GPS is enabled, the "Directory" page can be set to display how far away each building is from the user's current position. Users can learn more about Laker donors, and see their detailed information, by selecting an item from the Directory list. The "Near Me" page opens Google Maps in-app and populates waypoints for each building in the directory. The user can easily zoom to Grand Valley's Allendale or Pew Grand Rapids campuses. Lastly, the "Give" page provides users with the opportunity to easily contribute gifts to Grand Valley State University.

Table of Contents

1. Introduction

1.1 Laker Legacies

Established in 1960, Grand Valley State University has touched the lives of thousands of students, faculty, alumni, and leaders. Nothing more can sum up the bond of the Grand Valley community other than “Laker for a lifetime.” Development of the GVSU Laker Legacies application began in early 2014 to help expand, connect, and inform the Laker community.

The Laker Legacies Android application 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 Lakers who made profound contributions to the university. Buildings dedicated to these Lakers appear throughout Grand Valley’s campuses. The application focuses on these donors and the buildings named after them.

Laker Legacies has two major goals: to make it easy for the Grand Valley community, as well as the general public, to learn the history of contributing donors and to encourage Lakers to give back to their alma mater.

Laker Legacies is an immensely useful tool for students and parents, alike. With Grand Valley’s ever growing campus, new students and their families can wander the campuses with the help of this application and never feel lost. Laker Legacies can also be introduced at Transitions (freshmen orientation) and during campus tours to help newcomers learn more about Grand Valley. New and current students can use the application to find buildings they’ve never been to before, knowing they have a class there in the upcoming semester. With the help of Laker Legacies, students can learn more, be inspired, and give back.

1.2 Laker Legacies: iOS

Development for an iOS version of the Laker Legacies began last year, which served as a major starting point for Android development. The iOS version helped significantly in the design process for the Android version because the User Interface (UI) of Laker Legacies had already been constructed. Many aspects of the sibling application are presented in the Android version to achieve the same look and feel between the two.

1.3 Sponsor

Laker Legacies is sponsored by Danielle DeWitt, who is the Donor Relations and Stewardship Manager for Grand Valley. She was also the sponsor for the iOS version of Laker Legacies. With the popularity of Android smart phones today, she requested to be a sponsor for this capstone project. Having the application available on Android smart phones will put it within reach of a greater population of the Grand Valley community.

1.4 Team

Kyle Peltier has written Java code in the past but has never developed an Android application. Through the development process, technical skills such as reading documentation, following language conventions, and producing professional software were honed. In addition to these skills, Kyle became rather proficient with database development and application.

Matthew Williams was experienced with Java development, working the databases, and some web design. Working with the Android API was all new for him. While developing this application Matthew has become proficient in researching new Android API’s to use, and how to implement them within a project. Matthew has also grown in presentation skills throughout the semester.

Samantha Williams wasn’t familiar with the Laker Legacies program and is (also) new to Android development, and hasn’t worked with Java since her freshman year. She has sharpened her skills in developing for this platform, as well as integrating different technologies within an application, such as SQL. She has also learned how to write documentation for applications.

2. Development Environment and Tools

2.1 Eclipse IDE

The well-known Eclipse Integrated Development Environment (IDE) was the main tool used to progress through development of the Laker Legacies application. With the Android plug-in equipped, Eclipse IDE has all of the necessary tools to build an Android project, 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. Henceforth, Java is the main programming language used to implement the project. The integrated XML tools assisted in development for the visual aspect of the application such as screen layouts and organization. The provided Eclipse tools links the two together, Java backend and XML front-end, into a synced application.

2.2 Android OS and SDK

Ice Cream Sandwich (MR1), or Android 4.0.3, is the minimum SDK supported by the application. According to the Android development website, Ice Cream Sandwich supports 81% of consumers using Android devices [3]. With Android 4.0.3, the application is developed with a target Android API level of 15. Because the application requires the Google Maps Android API, the Google Services Android API level 15 was the API integrated into the application.

2.3 Github

Github is a website that provides a free source code versioning service which was essential for the design and development process of the Laker Legacies application. GitHub's environment made it possible for team members to contribute source code, documents, and other files to a central repository. The powerful versioning tool allowed team members to collaborate with ease and in real-time on the project throughout its development. Headquarters of the Laker Legacies application resides at the free Github repository URL: *https://github.com/william6/Android_Laker_Legacy*.

2.4 Inkscape

Inkscape is a free, vector-art graphic development design tool. With this free software, the team was able to produce custom graphic media for the project. Images such as the Laker Legacies home-screen welcome banner, tab icons, page indicators, and other graphic interface elements were constructed using Inkscape.

2.5 Miscellaneous Tools

Several other tools were used to help assist the team with development and they are: DIA (diagram) drawing tool, SQLite Manager Firefox plug-in, team-written bash script, ImageMagik command-line tool, and Google Drive.

The DIA drawing tool provided a neat environment to build database relationship schemas. The back-end of the Laker Legacies application is driven by Android's built-in database system, SQLite. Firefox web browser application provides a free SQLite plug-in called SQLite Manager. This tool was used in development for creating and visualizing the Laker Legacies database.

Laker Legacies stores a plethora of image files. Android has several conventions when it comes to image resource file names. Because of this, a script called "filestolower" was written to rename a bulk set of images to comply with Android file naming conventions. The command-line tool ImageMagik was used to bulk-modify the images included in the application. This tool made it easy to resize any number of image files and scale them down/up to a specific pixel size or ratio. By truncating images to a specific size, the physical package size of the application and loading times of the application were smaller.

Finally, Google Drive played a significant role in development, helping team members to collaborate on design documents and other organizational documents related to the development process all in real time. Due to varying team member schedules, Google Drive proved very helpful.

3. Software Approach

3.1 Development Model

Laker Legacies was developed using an agile software development model. Agile programming focuses on rapid prototyping while also supporting changes in application requirements during the entire development phase. Agile programming works well with smaller projects such as Laker Legacies. Developing in this model helped to provide frequent prototypes to the sponsor, Danielle DeWitt, throughout the relatively short project span. Frequent meetings with Danielle proved helpful through the development process because the team received regular feedback on the current progress of the application as well as discussed future development goals. This also opened doors for brainstorming new ideas and new features for the application. Because Laker Legacies was developed using agile programming, the sponsor's mental model and the developers' mental models were cohesive throughout, helping the team to design the best software quality possible.

3.2 Design Features

Due to the existence of an iOS version of the application, which hasn't yet been released, a lot of the design work could be applied to the Android application. Our sponsor wanted the same basic functionality as the iOS app, such as the Home, Map, Directory, Near Me and Give view-groups which accomplish a specific goal. We did however make some changes as we saw fit and implemented some additional features, which will be explained in detail in the sections below.

3.2.1 Layout

The original iOS application implemented a tabbed layout for navigating between the different screens. For consistency, the team decided to keep the tab layout design for navigating through the application. The unwritten standard for Android is to have the tabs on the top of the application. Despite this standard, and to clear up UI congestion (Section 5.2), the team decided to keep the iOS design of having the navigational tabs at the bottom of the screen.

When brainstorming changes to advance usability from the previous iOS application, the "Near Me" tab was removed. In the iOS implementation, the Directory and Near Me tabs mimicked functionality, the only difference was one displayed distances. The Near Me tab was removed and its functionality was put inside the Directory view, reducing redundancy. Further details of the Directory and Near Me views will be discussed in section 3.2.3. Having one less navigation tab makes moving through the application easier for users with smaller devices.

3.2.2 Home

The home screen is the first screen the user sees when opening Laker Legacies. Our team tried to mimic the iOS version of this screen almost exactly so users have the same cross-platform experience when opening the application for the first time. The design of the home screen is a basic slideshow that cycles through all of the donor images in the application's database. Overlaying the pictures is a logo with the name of the application.

3.2.3 Directory

The Directory screen allows the user presents all building and donor information to the user. The Directory can display two lists, donors or buildings, and lets the user sort these lists in various ways. By default, the view displays buildings sorted by their names in ascending order.

In addition, buildings can be sorted by campus or by distance from the user location. Selecting items from the list takes the user to a detailed screen, presenting the biography, or biographies, of the associated item.

3.2.4 Near Me

In the iOS application, the Near Me tab was used to show how far away buildings are from the user. As stated previously, our team decided that this feature could be implemented in the Directory tab in a better way. In order to use this feature, all the user needs to do is click on the "sort" button and choose "Distance from me". This will change the labels in the directory from the letters that were used to sort from A-Z, and replace them with distance in miles. This is accurate up to 1/100th of a mile.

3.2.5 Give

This tab is used for accessing the donation page on Grand Valley's website. This makes it easy for users to become donors. It may be difficult for users to navigate the Laker Legacies giving page within the application, so we added the ability to open it in the android browser.

4. Back-end Development and Implementation

4.1 Permissions

Laker Legacies requires several permissions from the user. These permissions include Google Maps and Google Services, network state, wifi, internet, device storage, and GPS locations. The location services are needed for using Google Maps in-app, as well as accessing GPS coordinates. Device storage is also required by Google Maps for data caching. Network state, wifi, and internet permissions are needed to access Grand Valley's "Giving" webpage.

4.2 SQLite Database

The SQLite database language is built into the Android operating system. Because of this, the team used SQLite to construct the major backbone of the application, the database. To achieve the goal of Laker Legacies, all information of Lakers, buildings, images, and interesting facts - and the relationships between them - needed to be stored in the database. Figure 4.2.1 is an Entity-Relationship database schema diagram of Laker Legacies. Integrating the database into the application allows a scalable amount of information. For future amendments to the project, buildings, donors, image information, and facts should be easily added to the database and the project assets without making any changes to the source code.

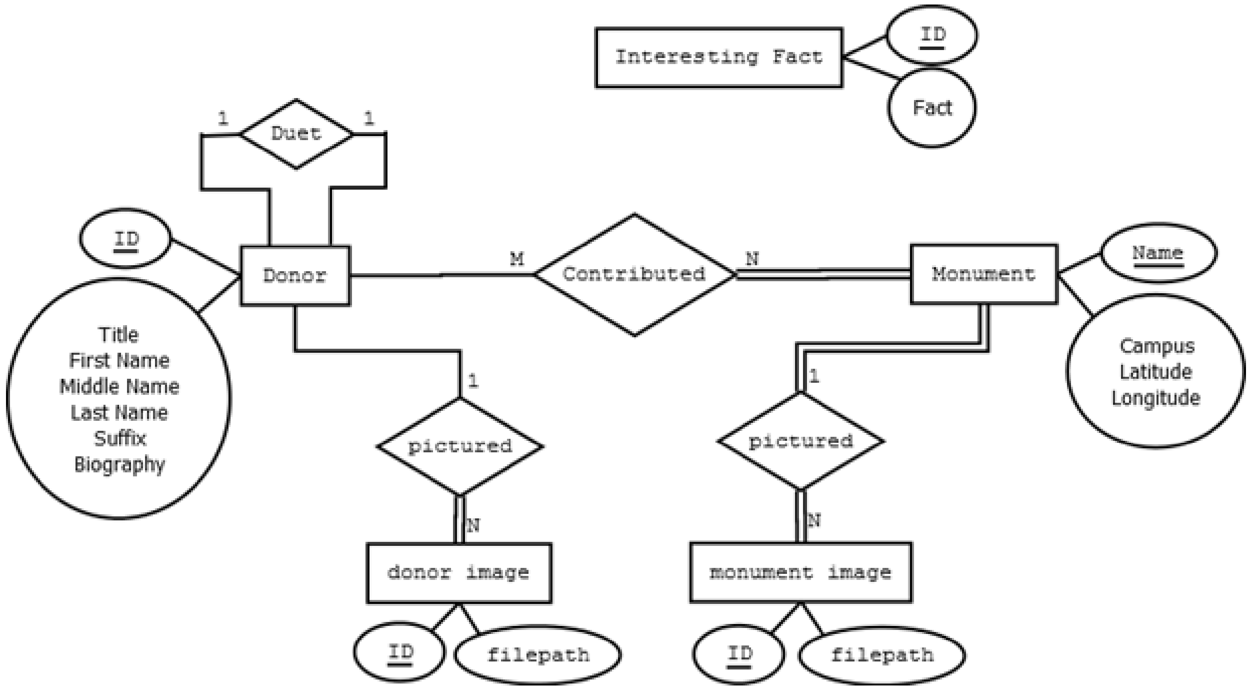


Figure 4.2.1
Entity-Relationship diagram of the Laker Legacies application.

4.2.1 Entity-Relationship Detailed Description

Each Donor has his or her own unique ID. Donor names could not be used as IDs because it is possible for two donors to have the same name. The database stores a Donor's title (Ex: Mr., Ms., Lieutenant, etc.), first name, middle name/initial, last name, and suffix (Ex: Jr., Sr., III, etc.). Splitting the Donor names by each section allows sorting by last name. Finally, a biography of the Donor is housed in the database. A relationship can exist between Donors which is denoted by a "duet" relationship. This relationship was added much later in the development process to account for joint donations. Noticeably, the relationship is partial, and only some donors have duets. William and Sally Seidman are prime examples of the necessity of this relationship. Jointly, Bill and Sally Seidman contributed to the Bill and Sally Seidman Living Center. William Seidman also made contributions toward other buildings but they were not joint contributions with Sally. Adding the Duet relationship helped distinguish between single and joint contributions without sacrificing dynamic source code. Further detail of the duet relationship and how it works is described in section 5.2.6.

Monuments, or buildings, are also stored in the database. Building names are used to uniquely identify each building. Accompanying each building is the campus it belongs to and GPS coordinates of its location. Multiple Donors can contribute to multiple Monuments. Likewise, a Monument can have multiple Donors. Not all Donors in the database contributed toward a Monument. Why have them in the database? This allows Donor information to be added to the database without displaying him/her in the Directory. For example, Thomas Haas is Grand Valley State University's 4th, and currently serving, president. Although Haas does not have a monument named after him (yet), the team thought it necessary to add him to the database. After all, he is a Laker and a major donor to the university. Doing this allowed the team to add an image of president Haas for display on the application's Home screen without compromising the Directory view.

An essential part of the application is displaying images of the Donors and Monuments of the application. The Donor Image and Monument Image entities both store a unique ID and filename of their associated types. All filenames must consist of only all lowercase letters and underscore characters. They must not contain spaces, periods, apostrophes, commas, or hyphens. The UNIX shell script "filestolower" was used to ensure all file names satisfied Android convention (section 2.3). All Donor, Monument, and Image information was obtained by the project's sponsor, Danielle DeWitt, and received by the team to add to the application. All Monuments must have an associated image, however, images of Donors is optional. There may be multiple images of the same Monument or Donor but they must have different filenames.

Also integrated into the database in the later stages of development was the Interesting Facts, or Factoids, entity. The idea of displaying facts to the user was brainstormed early in the design phase but was set aside as a potential enhancement of the application, not a necessity. The team saw an opportunity to include it in the application as a usability design enhancement. Factoids were gathered by the project team by consulting Grand Valley's "Laker for a Lifetime" online resource [4].

Based on these relationships, Table 4.2.1 was constructed to denote the structure of the tables used to store the necessary information. The collection of tables in the database and all current-version information are located in the Appendix (section 12).

Table 4.2.1
Laker Legacies table layout

Table Name	Primary Key(s)	Key Type	Foreign Key	Attributes	Attribute Type	Table Description
DONOR	donid	Auto Inc. Integer	DONOR.donid	title	Text	Donor information
				name_f	Text	
				name_m	Text	
				name_l	Text	
				suffix	Text	
				bio	Text	
DON_IMG	imgid	Auto Inc. Integer	DONOR.donid	filename	Text	Donor image information
MONUMENT	name	Text		campus	Text	Monument information
				latitude	Double	
				longitude	Double	
MON_IMG	imgid	Auto Inc. Integer	MONUMENT.name	filename	Text	Monument image information
MON_DON	MONUMENT.name	Text				Monument-Donor relationships
	DONOR.donid	Auto Inc. Integer				
FACTOID	factid	Auto Inc. Integer		fact	Text	Interesting Facts

4.3 Maps

Integrating Google Maps into our application was a whole new set of tools for our team. Our team decided to go with Google Maps within our application because Google Maps has a great API for Android and also has a lot of support within the community.

Google Maps requires many permissions in order to function at a useful level. Permissions are all added through the Android Manifest XML file for our application. This includes Google Services, network state, WiFi, internet, and GPS locations. Google Maps uses all of these permissions to work together to get a precise approximation of the user's location. Our application will run on a device that is only using WiFi as its primary data provider. In order to get the most accurate location, it is recommended GPS is enabled on the user's device. In order to notify the user of this, we created a function that checks for GPS to be enabled when the "Near Me" tab is opened. A small text box appears indicating GPS needs to be enabled, and the application directs the user directly to their GPS settings for their phone. From here, the user can either enable GPS, and hit back to go straight back into the application, or the user can hit back and not enable GPS.

In addition to the permissions needed for Google Maps to function inside of the application, it is required to have a Google Maps API key. This can be obtained from the developer section on Google's website. This works hand in hand with the developers' individual SHA1 certificate fingerprint and the package name of the application. Once the SHA1 certificate is obtained from the developers IDE. Eclipse pulls this from the developers' computer. After we got the certificate and package name, Google Developer generates a unique key that can only be used on the computer the SHA1 certificate was pulled from. This key is needed in the Android Manifest XML file in order for the application to communicate with Google Services and populate the maps in the application.

5. Front-end Development (GUI) and Implementation

5.1 Home

The home screen displays a slideshow of all the donor images in the database as seen in Figure 5.1. This cycles on a constant loop, leaving the image on the screen for 5 seconds before it transitions to the next one. Overlaid is a banner with the title of the application, “Laker Legacies”. The home screen sparks the curiosity of the users and gives them some context of what the application is about, rather than just having the directory open upon launching the application.

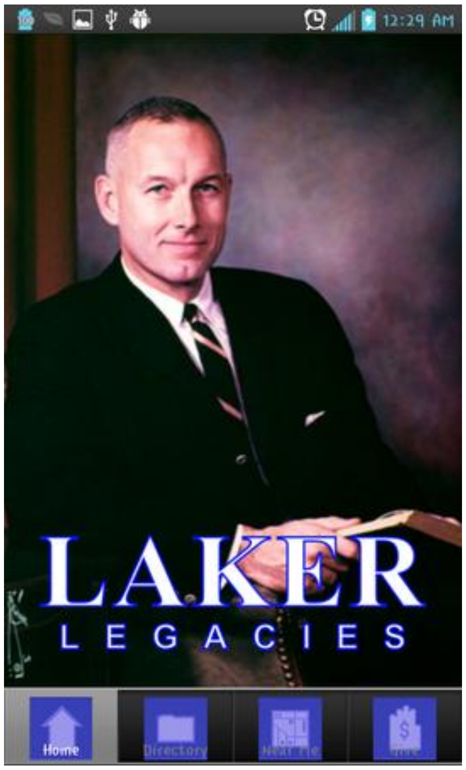


Figure 5.1

Home - Presents a slideshow of major donors of Grand Valley

5.2 Directory

The Directory view of the application acts as a portal between the user and the SQLite database. From the Directory view, the user is able to view donors and monuments, as well as see the relationships between them. Much like the iOS implementation, the Directory view needed a search function, display a list of buildings, and take the user to a more detailed screen when selecting an item from the list.

5.2.1 General Layout

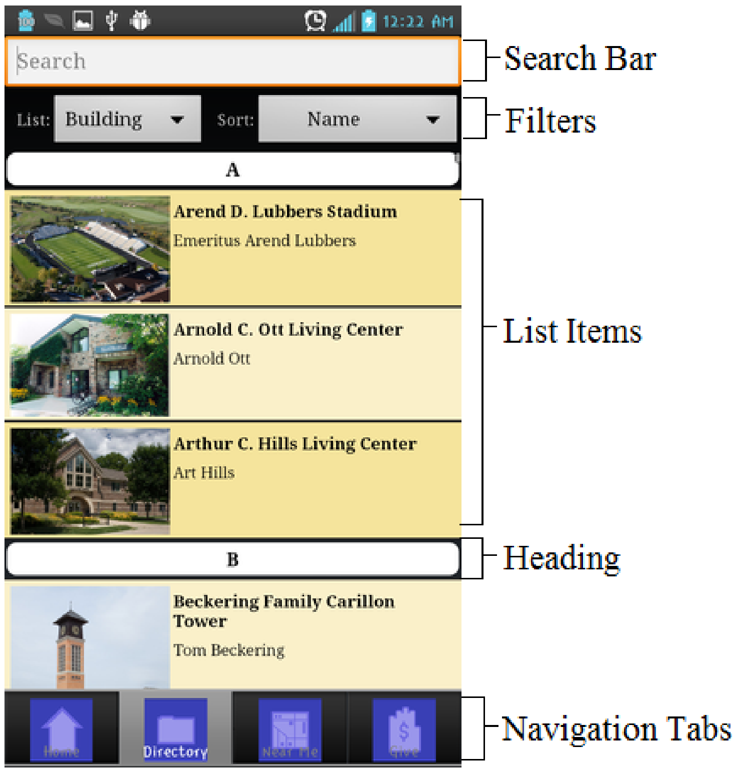


Figure 5.2
Directory - General layout

Figure 5.2 shows the general layout of the Directory view. Early in development, the navigational tabs were located at the top of the screen rather than the bottom. This made the Directory view crowded with options at the top of the screen. Therefore, the tabs were moved to the bottom. The top of the screen is populated with the Search Bar and the Filter options, followed by the List View of the Directory.

5.2.2 Loading Dialog

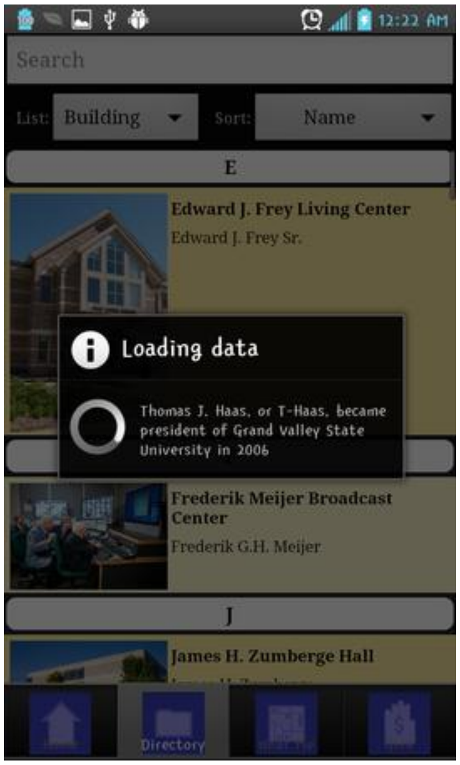

12:22 AM
Search
List: Building
Sort: Name
E
Edward J. Frey Living Center
Edward J. Frey Sr.
Loading data
Thomas J. Haas, or T-Haas, became president of Grand Valley State University in 2006
Frederik Meijer Broadcast Center
Frederik G.H. Meijer
J
James H. Zumberge Hall
Directory

Figure 5.3
Directory - Loading dialog

Donor and Monument images packaged into the application come in different pixel sizes and aspect ratios. For the Directory view, each of the images displayed is scaled to fit about 33% of the screen's width. When the images are resized, the aspect ratio of that particular image is saved, resulting in variable list item heights. Internally, the application needs time to scale all of the images to fit the specific width of the particular device. For this reason, a loading dialog was created to give visual feedback to the user that the application is busy (Figure 5.3). Wait times vary depending on the number of items displayed in the list, the physical screen size of the device, and the pixel sizes of each image packaged in the application. For an average high-density pixel (hdpi) Android device, max wait times are between 3-5 seconds.

For most of the development process, the dialog window consisted on two stagnant labels, "Loading data" and "Please wait." Towards the end of development as the team began polishing off features and integrating all parts of the application together, it was decided to add "factoids" to Laker Legacies. With the update, rather than display "Please wait" every time the list was updated, the application queries a random fact from the Factoid table and displays it. By doing this, the user can be mildly entertained while waiting for the application to load.

5.2.3 Filters and List Headings

Three filter options are present in the Directory view: the search bar and two drop-down menu filters. The search bar will be discussed in the next section (5.2.4). This section will discuss the two drop-down menus: “List” and “Sort” (Figure 5.2). The List menu provides the user with the option of displaying a list of donors or a list of buildings (default). When listing by donor, the Sort menu only provides the option to sort donors by [last] name. All sorting options are in ascending order. When the Directory view is listing monuments, the user can choose to sort the list by building name (default), campus, or distance from the user. When sorting by distance from the user, GPS must be enabled on the device. When enabled, the device pulls the GPS coordinates of the device and compares them to the built-in coordinates of each building in the database. Buildings are displayed in order from closest to furthest by tenths of a mile (Figure 5.4).

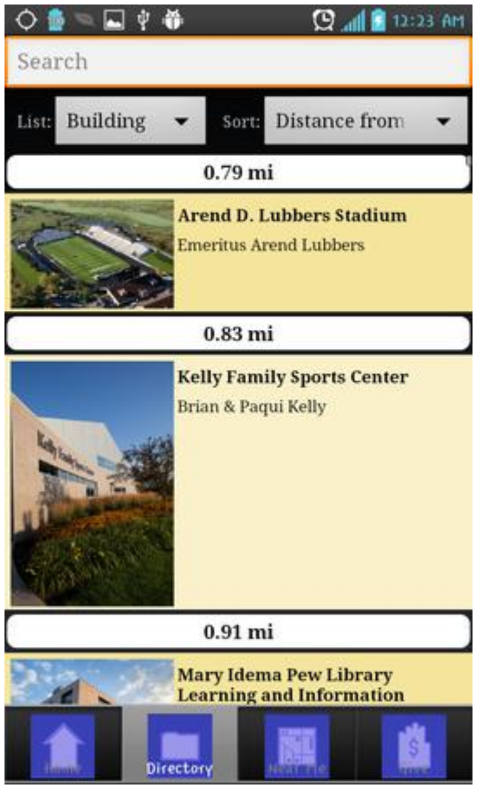


Figure 5.4
Directory - Displaying buildings by distance from the user

Headings (depicted in Figure 5.2) change dynamically based on how the Directory view is sorting the list. When sorting the items by name, headings display a single alpha character - denoting first letter of the last name of a donor or the first letter in the name of a building. When sorting by campus, headings display the full campus name (“Allendale, “Pew Grand Rapids”). Finally, when displaying by building distance, headings change to numeric values of distances in miles (hundredth). Internally, headings are handled dynamically so no matter how the data in the database changes, headings should always be accurate. For example, adding Holland campus buildings to the database should not interfere with the functionality of the Directory view.

Every time a filter is applied to the Directory view, the dialog screen is displayed and the new content is loaded. This is true for all views except for when displaying buildings by distance from the user. In this case, a dialog view appears the first time the user switches to these specific settings. As the device receives updates from GPS, the list is updated to display the changes in distance as the user moves, without displaying a loading dialog.

5.2.4 Search Function

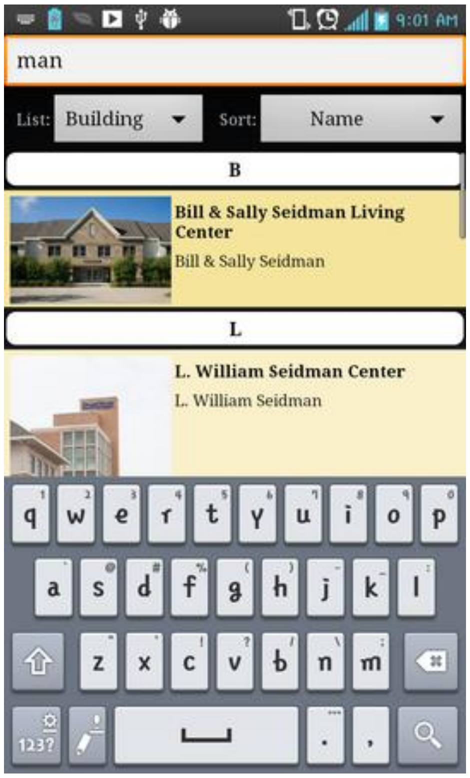


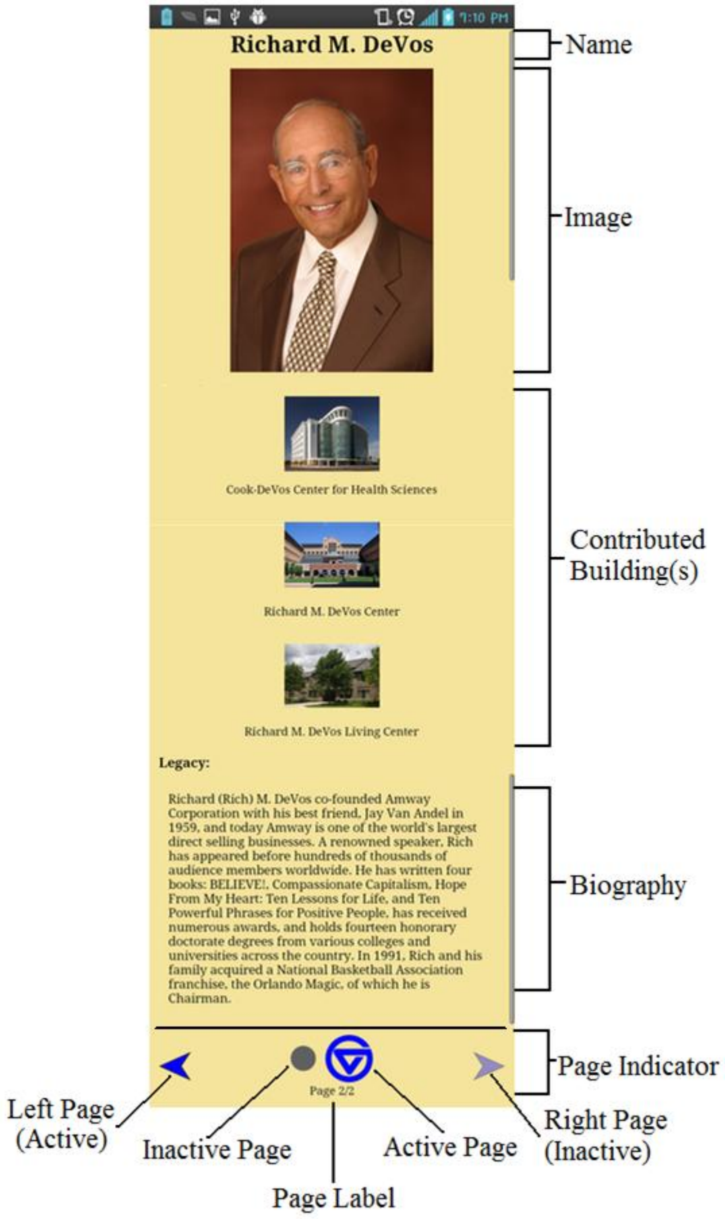
Figure 5.5
Directory - Enabling soft keyboard input
when the search bar has focus. Tabs are hidden.

The search bar in the Directory view allows the user to search the list by either donor or building. When the Directory view is listing donors, the search function searches donor names only. Likewise, when the Directory is displaying buildings, the search function searches only building names. When the search bar receives focus (waiting for input from the user), the soft keyboard on the device is shown automatically (no keyboard is shown if device has a hard keyboard). Displaying the soft keyboard overlays, and therefore hides, a large portion of the viewing area (Figure 5.5). The keyboard was programmed to cover the navigational tabs at the bottom of the screen to free up more visual space. The user can still scroll through the list and view all items.

It is important to highlight the keyboard shown to the user is always Android's built-in "search" keyboard. This specific keyboard changes the enter/submit button to the image of a magnifying glass (convention for search functions) (Figure 5.5). The magnifying glass visually communicates to the user the function of the keyboard. If a hard keyboard is present on the device, the enter/submit button will invoke the same function as the search button on the soft keyboard. The soft keyboard can be hidden by either selecting the magnifying glass on the soft keyboard (enter key on a hard keyboard) or by hitting the Android-standard “back” button hard key.

Because many applications have an auto-update function associated with text fields and search bars, the sponsor wanted the Directory view to be updated in real-time. If the Directory view updated the list in real time, the user would be delayed by several loading screens. The team made a trade-off to update the list when the user halts input after a set delay time. Currently built into the application is a 1.5 second timeout. Therefore, when the user enters data into the search bar, when 1.5 seconds elapses from the last input character, the Directory view is updated. The delay timeout can be easily modified for future updates if need be but the current settings has been observed to work well within the application and has not caused significant interruption to the user. Because of the search delay, there are two ways for the user to start searching the list, by waiting the timeout delay period or by selecting the search button from the soft keyboard. Waiting the timeout delay period does not hide the soft keyboard.

5.2.5 Pages



Legacy:

Richard (Rich) M. DeVos co-founded Amway Corporation with his best friend, Jay Van Andel in 1959, and today Amway is one of the world's largest direct selling businesses. A renowned speaker, Rich has appeared before hundreds of thousands of audience members worldwide. He has written four books: BELIEVE!, Compassionate Capitalism, Hope From My Heart: Ten Lessons for Life, and Ten Powerful Phrases for Positive People, has received numerous awards, and holds fourteen honorary doctorate degrees from various colleges and universities across the country. In 1991, Rich and his family acquired a National Basketball Association franchise, the Orlando Magic, of which he is Chairman.

Figure 5.6
Bio-view - Extended display of a page and its encompassing pager

Selecting a list item, whether the item is a building or donor, takes the user to a "Pager" screen. The Pager screen holds a set of pages, displays a single page on the screen at a time, and allows the user to navigate seamlessly between pages. When a donor is selected from the list view, the pager contains only one page - the page of that particular donor. When a building is selected from the list view, the pager contains a page for every donor associated with the selected building. The pager screen is dynamic and can hold any number of pages; therefore, a building can have any number of donors and the application should display all donors as expected.

Figure 5.6 shows the general layout of a page as well as the pager layout. The only part of the pager visible to the user is the navigational images/buttons at the bottom of the screen. The pager indicates the number of pages in the set with a text label. Accompanying the text label are bullet indicators, a bullet for each page. The current page is denoted by a Grand Valley logo (circular "GV") and inactive pages are denoted by grayed bullets. On the left and right of the page indicators are navigation buttons. The user can tap the left arrow button to move to a previous page or tap the right arrow button to move to the next page. If the currently displayed page is the first or last page (or both), the respective navigation buttons are dimmed and inactive. The user can also navigate between pages by using screen swipe gestures.

Each page consists of a biography view (or Bio-view) of a donor. Within the page, the name of the donor(s), an image of him/her/them, a list of contributed buildings, and a biography are displayed. The pager sets up a vertically scrolling view so the entire page can be read by simple vertical screen swipes. The scrollbar on the right side of the screen is always displayed (never fades out), always indicating to the user that the page is scrollable. If multiple images of the same donor exist in the database, a random image of the donor is displayed. If no image of the donor exists in the database, an image of any of the buildings the donor contributed towards is randomly pulled and displayed. The user can select an image of a building from this view to see a zoomed-in version of the selected image (Figure 5.7). Finally, a biography of the donor(s) is displayed at the bottom of the view. The brief biography is a description of the donor(s) and the legacy he/she made at Grand Valley State University and the influence left behind on the Grand Valley community.

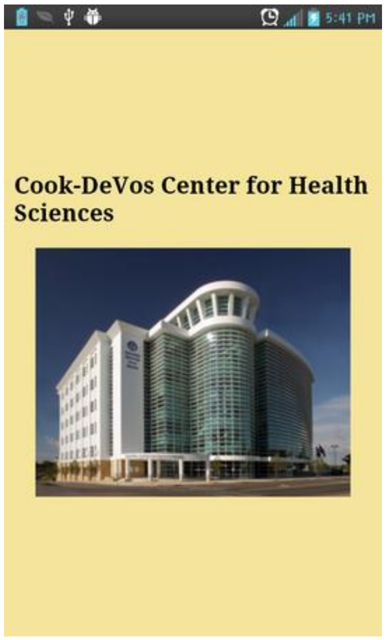
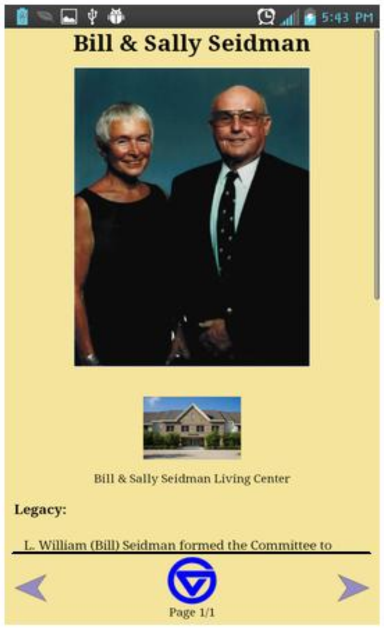


Figure 5.7
Building View - Detailed image of a building selected from a Bio-view

5.2.6 Donor "Duets"

As noted earlier, a donor can be related to another donor through a "duet" relationship. The duet relationship is one-to-one so it can only exist between a pair of donors. Thus far, cases of duet donors have arisen only in joint donations of married couples, which is where we foresee it staying. For example, Bill & Sally Seidman, Peter & Pat Cook, and Brian & Paqui Kelly. For duet donors, two entries are required in the database. The first entry is the male contributor; the second entry should represent the pair. Taking the Seidman's as an example, L. William Seidman has his own entry in the database and then Bill & Sally is the second entry, where the Bill & Sally entry has a duet ID of L. William Seidman's donor ID. See section 12.1 for further reference.

When listing the Directory view by donors, if a duet relationship exists for a donor, he/she is only displayed with the duet donor. For example, rather than displaying L. William Seidman and Bill & Sally Seidman, only the pair is displayed. Selecting the duet donors from the donor list view displays all buildings both donors contributed towards. This is different than when the Directory is displaying buildings. When listing by building and a particular building is selected, only donors toward that building are displayed. Figure 5.2.6 conveys the difference between displaying duet donors from a donor list item and a building list item well.


5:43 PM
Bill & Sally Seidman
Bill & Sally Seidman Living Center
Legacy:
L. William (Bill) Seidman formed the Committee to
Page 1/1

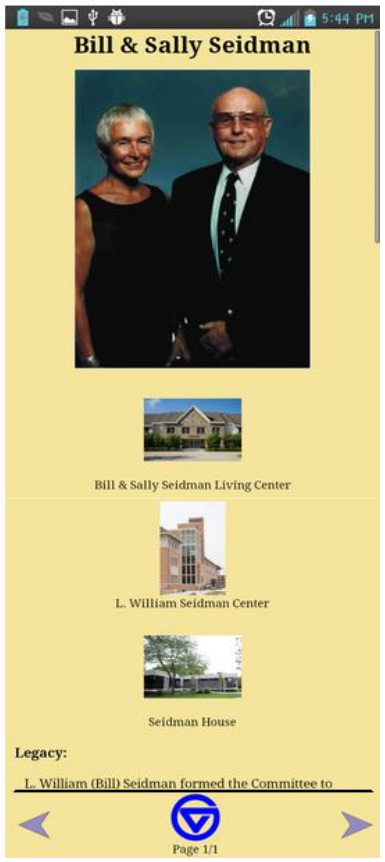

5:44 PM
Bill & Sally Seidman
Bill & Sally Seidman Living Center
L. William Seidman Center
Seidman House
Legacy:
L. William (Bill) Seidman formed the Committee to
Page 1/1

Figure 5.8

Bio-view - Difference between displaying duet donors based on building or donor. Selecting a building shows only the joint contributions made (Left). Selecting the joint donors shows all buildings either donor contributed towards (Right).

5.3 Near Me

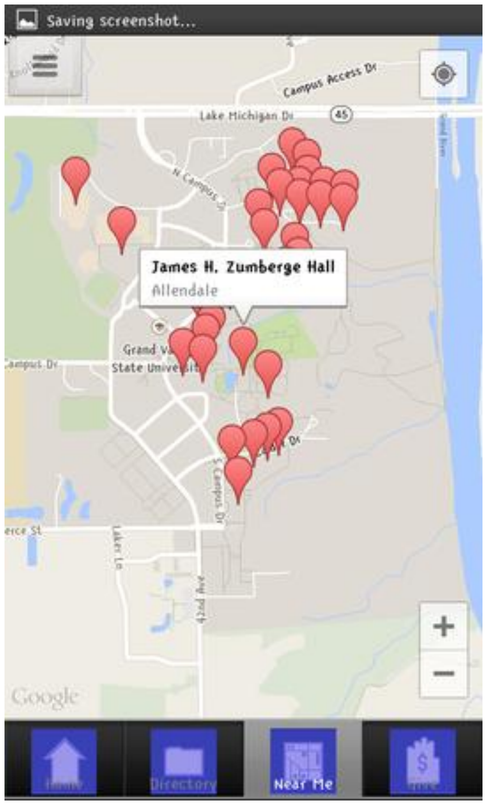


Figure 5.9
Map - Map showing all the donated buildings on the Allendale campus

Our “Near Me” is our tab that has the Google Maps fragment inside of it. Once the user clicks on the “Near Me” tab, the map appears directly on Allendale’s campus. From here, the user will be able to click on any of the building’s waypoints that are on the screen to find out what the name of that building is and what campus that building is associated with (Figure 5.9). Other features that the Maps tab includes are zoom buttons in the bottom right corner of the screen to make navigation easier, a “My Location” button in the top right corner of the screen, and a “Menu” button in the top left of the screen. Google Maps supports the “Pinch and Pull” technique for zooming in and out as well. The “My Location” button is used in accordance with the user’s GPS or WiFi services. Once clicking this button, the map will zoom out and zoom directly into the user’s location. The accuracy of the location is determined by what location services are enabled. The menu button includes options to change what the map looks like as seen in Figure 5.10. These options include: Hybrid Mode, Map View, Satellite View, and Show Traffic. The menu button also includes two options to zoom to each of Grand Valley’s campuses. Our application launches Google Maps in the “Map View” automatically. If the user wants a more detailed view with terrain, the user can select the “Satellite View”. If the user is looking for a more navigational mode and a terrain mode, they can select the “Hybrid Mode”, which shows the street names and the terrain. If the user is planning on making a trip from the Allendale Campus or Pew Campus, the user can select “Show Traffic” and this will help the user decide if the route they use to get downtown will be fast enough or if the traffic is too busy. Lastly, if the user wants to navigate quickly to the downtown Pew campus, or the Allendale campus, all they have to do is select one of the options in the menu and the map will zoom directly to that location.

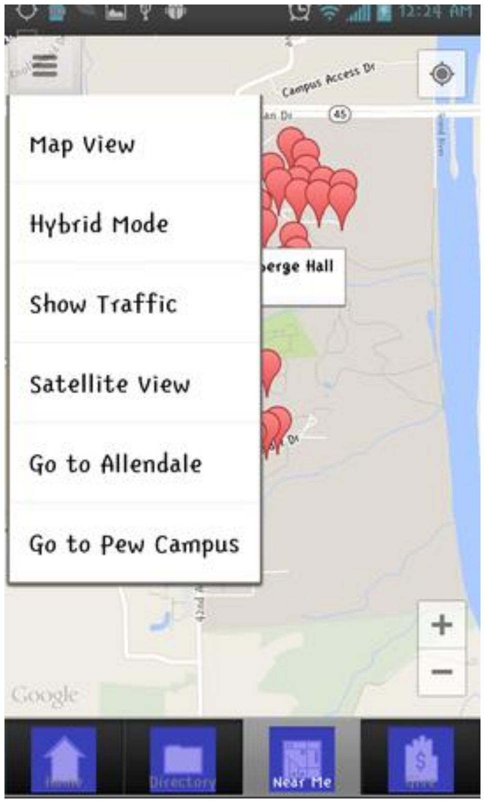
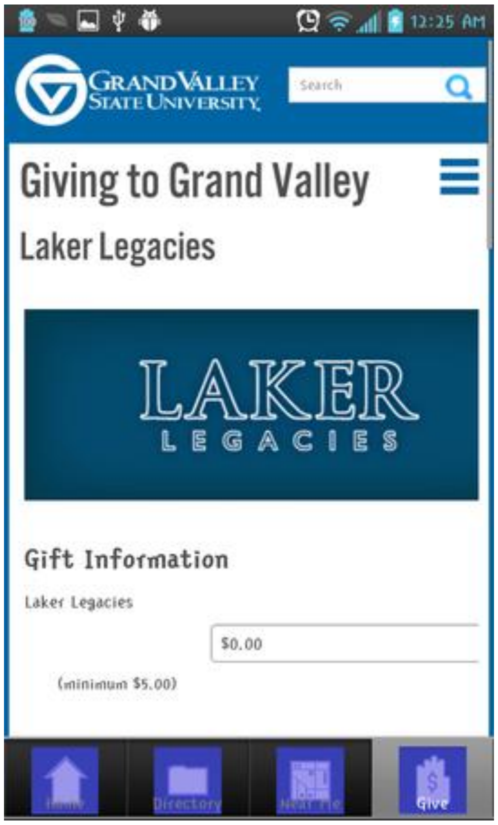


Figure 5.10
Map - Map showing all the donated buildings on the Allendale campus with menu to change view options and campus

5.4 Give

The Give tab, pictured in Figure 5.11, gives the user an easy way to give to GVSU within the application. If a user prefers to open the website in a full browser on their mobile device, this is an option through the menu button that most Android devices have.



Giving to Grand Valley

Laker Legacies

LAKER LEGACIES

Gift Information

Laker Legacies

$0.00

(minimum $5.00)

Figure 5.11

Give - This tab enables the user to give a financial contribution to Grand Valley State University.

6. Complications and Refinements

Early stages of database development consisted of a process of entering data into various XML files, parsing them in the application, and building a SQLite database file in-app. The Firefox SQLite Manager tool was a significant help in creating the database file and modifying data. Simple Excel spreadsheets could be imported into the tool to populate a table very quickly. SQLite Manager was also very useful for visualizing data elements, helping the team to catch small issues with data. Looking back, database design would have been much more efficient if the SQLite Manager tool was used much earlier in development.

Research on a new, more standard way to add tabs to an application should have been done sooner. Implementing the tabs in the application before making significant progress on the functionality of the different screens would have made for a smoother process and cut time on debugging instead of using a deprecated method.

Figuring out which files were unique to our development environment and using the "git ignore" feature of GitHub definitely helped with our merge process. Though there were still some files causing conflicts, figuring out which files were causing the majority of the issues would have saved a lot of time trying to get our builds working again after a merge.

7. Testing

An abundant amount of testing has been conducted to ensure application quality. Debug testing has been conducted throughout development to make sure functionality is working as expected. Aside from debug testing, several runs of the application were tested, under many circumstances, to ensure again that the app functions as expected.

The application was tested on several different mobile devices. A difficult part in developing an Android application is supporting a plethora of screen sizes, densities, and orientations. The team wrote code to the best of their ability to ensure functionality and usability persists in each device model.

Google Maps was tested by physical traveling across campuses. By walking, or riding in a vehicle, with the application open, the Near Me view displays all buildings relative to the user's location in real time. The Directory, when displaying building distance, also updates in real time with accurate GPS positioning.

The team also reviewed the data in the database to ensure that all data presented to the user is accurate. The team stepped through each donor, each biography, and each building, double-checking its accuracy.

8. Software Engineering Code of Ethics (SECE)

As software engineers, the team is committed to producing the highest quality work possible with the time provided. Several items in the Software Engineering Code of Ethics is Approved article [5] have been exercised throughout the development of Laker Legacies.

8.1 Development

One of the most difficult aspects of software engineering is ensuring that the team understands exactly what the customer wants and/or provides professional input toward a professional software product. Throughout development, the team has exercised code 3.07 by meeting regularly with its sponsor. At the meetings, the team presented progress reports, discussed software options, presented difficulties, and then discussed future goals. Discussing the comparison between the iOS version of Laker Legacies and the Android version also helped the team better understand project requirements.

Code 3.06, highlighting professional standards, was followed by the team as they followed Android development conventions [2]. The Android development style guide specifies several key elements to professional design of writing source code as well as how source code should be structured and documented within its written file. Code elegance such as line indentations, brace styles, conditionals, and field names have all been taken into account during development. Also, the convention of keeping all filenames to lowercase letters and underscores has been followed.

Throughout development, extensive debugging and testing has been conducted, following code 3.10, to help ensure the highest quality application. Bugs show up in code regularly and the team has tested Laker Legacies on several devices under different circumstances to help ensure it runs smoothly across the board.

8.2 Integrity

Coinciding with code 3.10, debugging and testing, is code 3.01 encourages software engineers to strive to produce the highest quality software application under the circumstances and scope of the project, ensuring that all aspects of the project have been approved by the patron. Again, debugging and testing the application helped the team reach its goal of producing quality software. In addition, all design decisions were approved by the team’s sponsor, making sure the application meets her expectations.

Code 3.13 discusses legalities of development, specifying that data within an application should be acquired, and presented, lawfully and ethically. The majority of the information stored in Laker Legacies (images, donor information, and building information) was gathered by the sponsor and given directly to the team. Ms. DeWitt has proper authorization to give us this information as she is the liaison between the team and the university itself. Some data of the application, however, was compiled by the team. Said data has been approved by Ms. DeWitt.

8.3 Documentation

Sufficient documentation is essential for project maintenance and future development (codes 3.08 and 3.11). The team can safely say they have documented the application thoroughly, inside and outside of the source code. Within the source code, Classes, methods, and advanced algorithms are accompanied by commented code. Outside of the source code, several documents have been written, diagrams made, and screenshots taken to document the application as much as possible.

9. Future Improvements

Although the team is very pleased with the current application, many improvements could be made. Some of the ideas explored below are quite feasible for a near-future, updated version of the app. Other ideas, however, span a much larger scope.

9.1 Small Scope

Currently, Laker Legacies only holds information about buildings dedicated to major donors to the university and information about those donors. An updated version of the application could include all of Grand Valley's buildings, not just the buildings dedicated to past Lakers. With such an addition, buildings from all Grand Valley campuses can be added to the database. Accompanying this update could be descriptions of the buildings. For example, Mackinac Hall could be included in the app and have a description of how Mackinac Hall is the main harbor of mathematics, statistics, and computer science courses.

Again, Laker Legacies only holds information about buildings and their donors. This could be taken a step further and the application could house campus monuments as well. Architectural structures and products of art such as the Mini-Mac Bridge, Transformational Link, and Metal Marching Band could all be added to the scope of the application. In order for an update such as this to occur, a lot of time must be put into research, contact history & approval, and gaining access to media. A significant issue with adding more media to the application is data size. Adding more images greatly increases file size, which sometimes deters users from downloading the app. Also, when APK file sizes approach 50MB, additional steps must be taken to add the application to the Google Play store [1].

If the application were developed under a company umbrella, team programmers would not have to worry at all about the design of graphic media of the application. The does not have experience with designing digital art. Noticeably, some GUI elements of the application could be polished a little more, adding a more professional look Laker Legacies. Possibly, an updated version could include outsourced digital artwork to make the application glisten.

9.2 Greater Scope

Laker Legacies can go much further to support campus navigation by integrating Google Maps "Directions" algorithms. Users could pick out a building to visit and have Google Maps generate directions to the specific location to help navigate the user. Accompanying this feature could be parking options near the target location to help the user find a place to park. Furthermore, parking lots can be indicated as meter parking, visitor parking, and permit parking.

In regards to campus tours and incoming freshmen, Laker Legacies could include tour suggestions or "sights-to-see" at various Grand Valley campuses. These suggested tour routes could also be themed. For instance, if an incoming freshman were interested in the art program at Grand Valley, the tour could take the freshmen to Alexander Calder Fine Art Center, the Transformational Link, the Lake-halls, and other buildings the student would likely have class in.

Another suggestion for expanding Laker Legacies would be to combine it with the existing Laker Mobile Android application. Merging the two applications would be an immense amount of work; however, it would consolidate two useful, complete tools into a single application.

10. Demonstration Video

A live demonstration of the application was recorded using screen-capture software to help portray the power of Laker Legacies. Located at the following URL, *http://www.youtube.com/watch?v=D46dV9qBe0o&feature=youtu.be*, the video highlights all of the key features of the application. The video will be used as a tool to teach people how to use the application. A link to the video will likely be included in the presentation description of Laker Legacies on the Google Play store.

11. Conclusion

The team is greatly pleased to have worked on the Laker Legacies application. Developing the application has introduced, and honed, several technical skills critical for real-world development. Skills such as technical writing, documentation, research, and application development have all been thoroughly explored.

As the Grand Valley community is constantly trying to improve our campus, we expect that the number of donors and buildings will grow extensively in future years. We believe with the Laker Legacies application, the Grand Valley community can learn, stay informed, and be encouraged.

12. Appendix

12.1 Donor Table

12.2 Donor Image Table

12.3 Monument Table

12.4 Monument Image Table

12.5 Monument-Donor Relations Table

12.6 Table of Interesting Facts

13. References

[1] Android Development - APK Expansion Files
Accessed 4-18-2014
http://developer.android.com/google/play/expansion-files.html

[2] Android Development - Code Style Guidelines for Contributors
Accessed 4-18-2014
https://source.android.com/source/code-style.html

[3] Android Development - Platform Versions
Accessed 4-18-2014
http://developer.android.com/about/dashboards/index.html?utm_source=ausdroid.net

[4] Grand Valley State University - Laker for a Lifetime
Accessed 4-8-2014
http://www.gvsu.edu/lakerforalifetime/laker-traditions-13.htm

[5] Software Engineering Code of Ethics is Approved by D. Gotterbarn, K. Miller, and S. Rogerson.
Communications of the ACM, Vol. 42, No. 10 (October 1999), 102–107