ISTA 392 project document

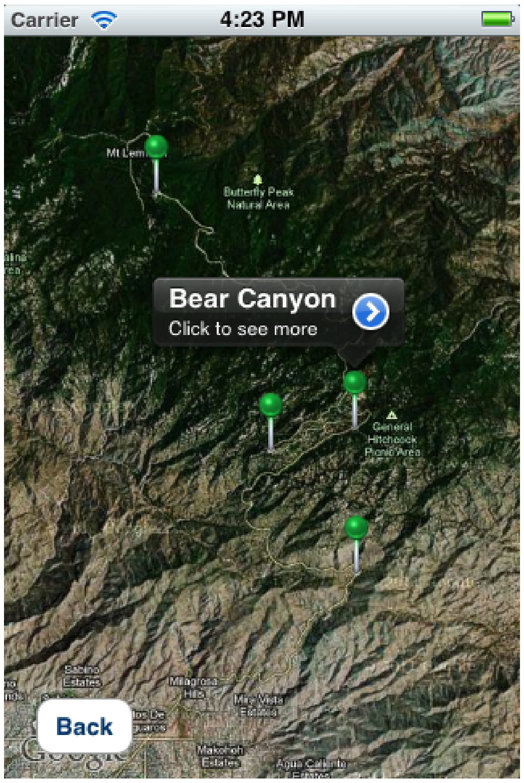
Application: Flandrau Science mt Lemmon Guided Tour
Group members: Mark Grandi, Sefan Lindstrom, Charles Rickards, Benjamin Dicken

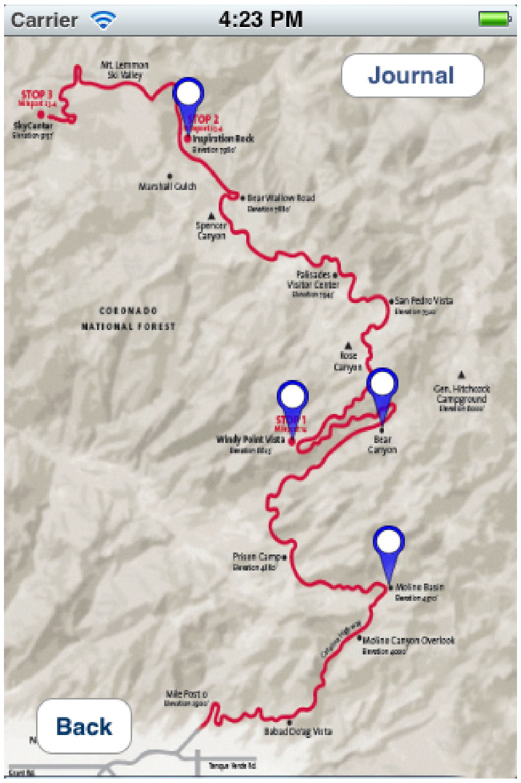
• OVERVIEW AND PURPOSE OF THE APPLICATION

Our group took on the task of making an application for the Flandrau Science center at the University of Arizona. The app itself serves as a virtual guided tour for the users' drive up mount Lemmon, AZ. As a user of this app begins the drive up the mountain, they will be shown several key locations that they can stop at. For each stop, the app contains information, photos, and more for the user to view.

We began development of this application roughly five weeks before the end of the Fall 2011 semester. The four of us each took on different aspects of the application, which is discussed in more detail in a later section of this paper.

As you drive up the mountain, you will be able to continually see your current location on the integrated Google map. All of the stops of the tour are represented as pins along the road up the mountain. Once you hit one of the pins, you can tap on it to take you to the information pertaining to that stop. In case a users experiences a loss of data reception, we also implemented a static (non-Google maps based) map with all of the stop locations on it, allowing the user to still access all of the stop information. The two images below show off first the Google maps based tour view, and second the static tour view.



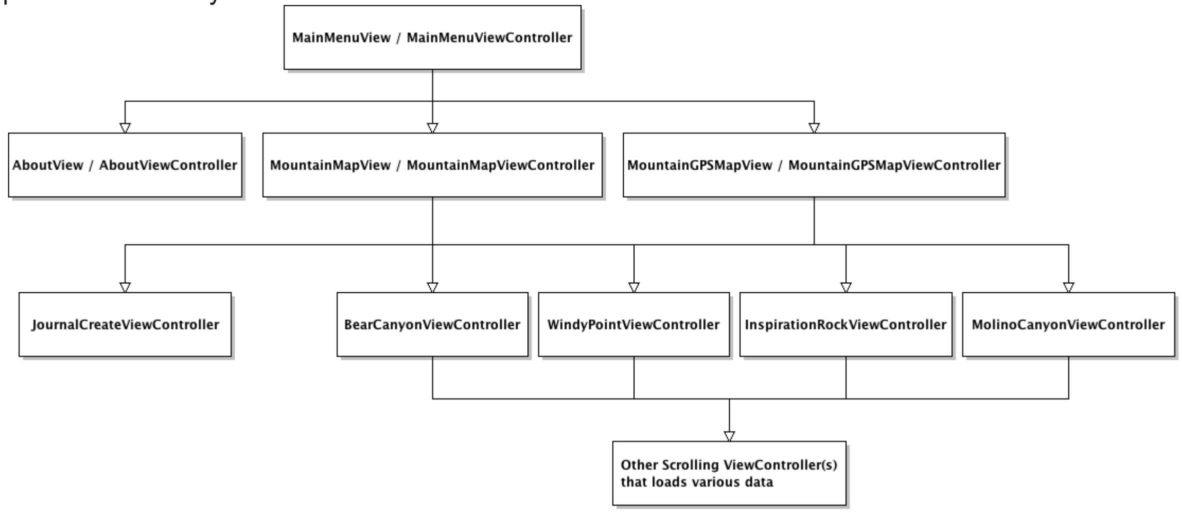


The tour content contained in this application was developed by several students geoscience students. They are: Jill Yarmchuk, Elysse Hernandez, Michael Conley, Wesley Smyth. Each of these students developed content, created ideas for games and quizzes, took photos, and more for each stop along the tour. These four students each developed research papers and a collaborative PowerPoint slideshow demonstrating the potential for this app several months before development by our group began.

• IMPLEMENTATION DETAILS

The hierarchy of the navigation of this application is shown in the following diagram.

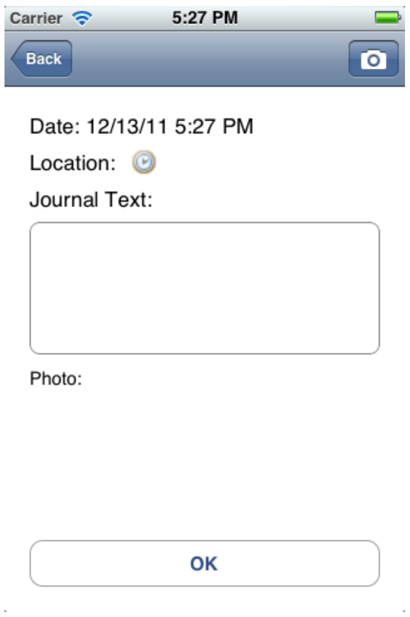
Application Hierarchy:


MainMenuView / MainMenuViewController
AboutView / AboutViewController
MountainMapView / MountainMapViewController
MountainGPSMapView / MountainGPSMapViewController
JournalCreateViewController
BearCanyonViewController
WindyPointViewController
InspirationRockViewController
MolinoCanyonViewController
Other Scrolling ViewController(s) that loads various data

As can be seen from the diagram, When a user first starts up the app, they are taken to the MainMenuView. From this point, they can go to three other pages, The AboutView, MountainMapView, and the MountainGPSMapView. The AboutView is just a simple page with information about the contributors and creators of the app. The MountainMapView and MountainGPSMapView take the user to the Static map and the Google maps based guided tour map. Since these two views both contain the same pin locations, the pins from both views lead to the same stop information views. For each stop their is a view that has a variable amount of buttons linking to pages with general information about the stop.

From the Static map view as well as from any of the stop ViewControllers the user can also press a button to get to the JournalCreateViewController. This ViewController is a simple page that gathers information about the users current location, current time, and then allows the user to take a photo and type in notes. Once a photo has been taken and notes entered, the information can be uploaded to a Flandrau server, where Flandrau will use the data for research and to add a level of interactivity to their museum.

The following screenshots show off the journal page and one of views for the four stops:

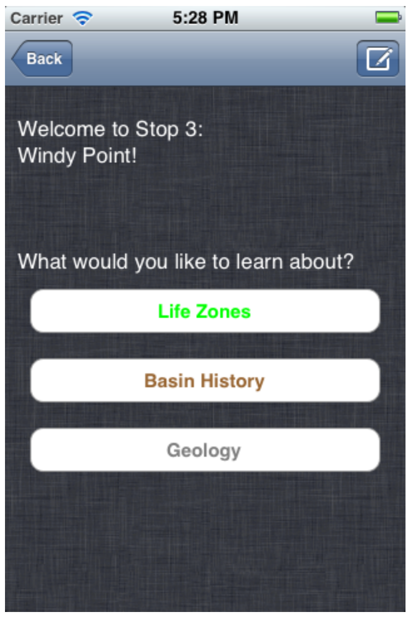


Date: 12/13/11 5:27 PM

Location:

Journal Text:

Photo:



Welcome to Stop 3:
Windy Point!

What would you like to learn about?

- CHALLENGES

There were several facets of iOS development and Objective-C that we had to learn outside of lecture in order to complete this project.

One of the biggest challenges of these was learning how to use Objective-C to send a photo over the Internet to a server. Mark ended up doing most of the implementation of this networking, and with a little help from Tom and Cody, we came up with a nice solution. We ended up importing a JSON library to our project and adding several methods to that library that we needed for our image uploading. To send the image, we converted the image taken on the iOS device to a JSON file, and then sent that file to a php server which then wrote the image and its accompanying text to a file on the server.

Another challenge we ran into was implementing a UIScrollView to work with UIPageControl. We decided to use a UIScrollView when displaying the stop information to a user so that we could let users navigate through the information with the simple swipe of a finger. Although we ran into challenges implementing it, after getting help from Tom and from our good friend Google, we got it to work.

- FUTURE IMPROVEMENT AND ADDITIONS

One feature that did not get fully implemented was the image uploading from the journal. Although in its current form it is fully functional, we were not able to implement the php image server on Flandrau's server. We were unable to do this because our contact at Flandrau, Jennifer Fields, was not able to meet with us for awhile after our initial meeting. Instead, for testing purposes, we are currently sending images to Mark's personal server. Eventually, it would be nice to implement this feature the right way on the right server.

We also did not have time to insert all of the photos, games, and information into each stop of the app. Right now, each stop topic has at least one page of information, but there is much content that we did not have the time to get loaded into the app. Because of the way we implemented the ViewControllers for the pages that display the information (and because the data is read from plist files), it will not be too difficult for future developers to add additional info.

- WORKLOAD DIVISION

Each member of our group worked on different part of the application. The workload breakdown was divided up as follows:

Mark Grandi:
Primarily worked on the JournalPageViewController, php server, networking and a bit of the UI. Mark also did much in the way of finding and removing memory leaks and general bugs.

Benjamin Dicken:
Developed the majority of the UI for the application, including implementation for the MountainGPSMapViewController and the UIScrollView. Ben also implemented the code for reading from a plist file.

Stefan Lindstrom:
Worked on adding content into the application and dynamically loading plist content.

Charles Rickards:
Worked on adding content into the application and dynamically loading plist content.