ECE 495 VIP Fall 2010

Final Report:
Click&Fix Application
for Android Devices

Janam Jhaveri
Paul Mongold
Xiang Cui (James)

Table of Contents

The Goal

If a Purdue student wants to report an infrastructural issue on campus, he (or she) needs to email Purdue Maintenance. By the time the student can reach a computer he's already forgotten the problem. Hence the goal for this semester's project was to develop an Android application that would report campus issues easily and rapidly. The submitted information would be passed on to Purdue Maintenance, who would then decide what to do with the reported issue. The application would include the following features:

- Take picture of issue, examples:
  - graffiti
  - snow removal
  - water drainage
  - potholes/sidewalks
  - tree issues
- Utilize GPS to automatically provide location information
- (Optional) allow user to submit comments
- Send information through email

The inspiration for this application came from a similar app for the iPhone already implemented in Boston called Citizen's Connect.

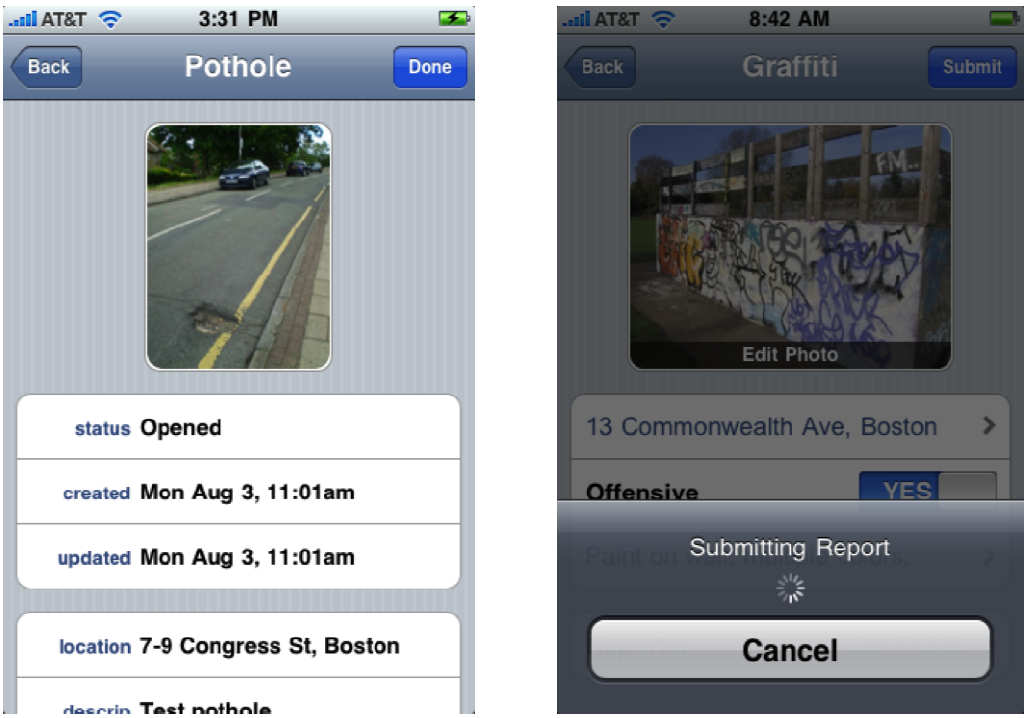

Figure 1 - Screenshots of Citizen's Connect

The project we wanted to develop was named as Click&Fix, we proposed Milestones for our project on 9 / 15 / 2010:

- Due 10/13/10 (3 weeks)
  - Camera: Turn on/off, take/store picture
- Due 10/27/10 (2 weeks)
  - GPS: Turn on/off, retrieve/store coordinates
- Due 11/10/10 (2 week)
  - Transmission
- Due 11/24/10 (2 weeks)
  - Text Entry & Additional Features
- Due 12/8/10 (2 weeks)
  - Debugging & Finalizing

Functionality

User Entry

The User Entry tab activity is used to obtain the user's contact information along with a description of the problem that is being reported. The user has the option to save their contact information across uses of the app and currently there is an option to remain anonymous, which may be eliminated in the future. Then once the user selects if the problem is indoors or outdoors, a drop-down list appears displaying the valid problem categories for which they must choose one. Upon selection, the user can describe the problem in more detail if they want or required if "Other" was chosen. When the "Confirm" button is pressed the app checks if all the entries are valid and either displays an error stating what needs fixed or a message saying it was confirmed.

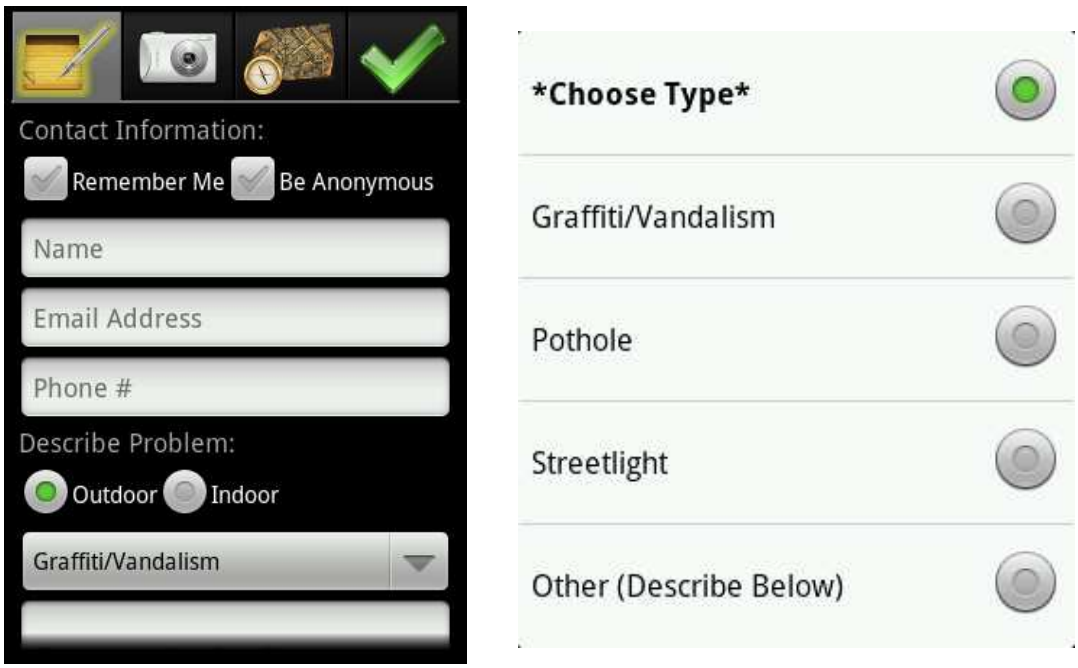

Figure 2 – User Entry Tab and Problem Type List

Camera

When the camera tab is first selected, the user is taken straight to the camera itself with a full size preview being shown. If zoom is supported by the phone's camera, buttons for zooming in, zooming out, and taking the picture will be present, otherwise just the capture button. When the button to take the picture is pressed the camera will attempt to autofocus and flash, if deemed necessary are supported by the phone's camera. The app then returns to the camera tab and displays the picture that was just taken and a button to reopen the camera for a new picture. If no picture was taken the first time, a placeholder image informing the user that no picture was taken will be shown.

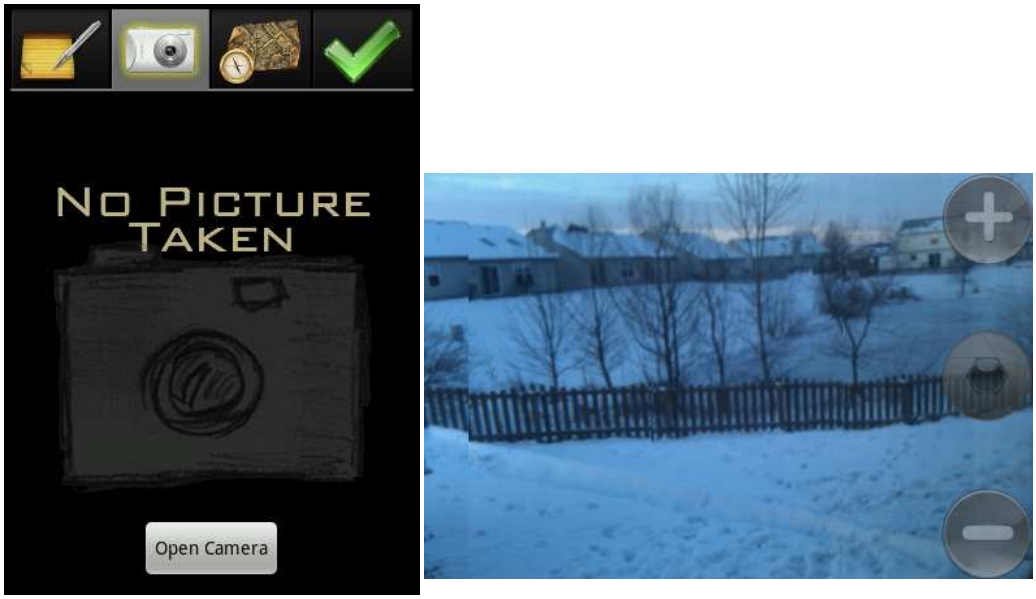

Figure 3 – Camera Tab and Camera Preview w/ Controls

Global Positioning System

The GPS tab activity currently uses the in-built GPS tracking system to obtain coordinates and display those coordinates on a Google Map. Since the user might walk around while creating an issue report, coordinates are updated dynamically and the user needs to save coordinates. A future goal is to switch between getting coordinates (for outdoor problems) and allowing users to enter a building name and room number (for indoor problems).

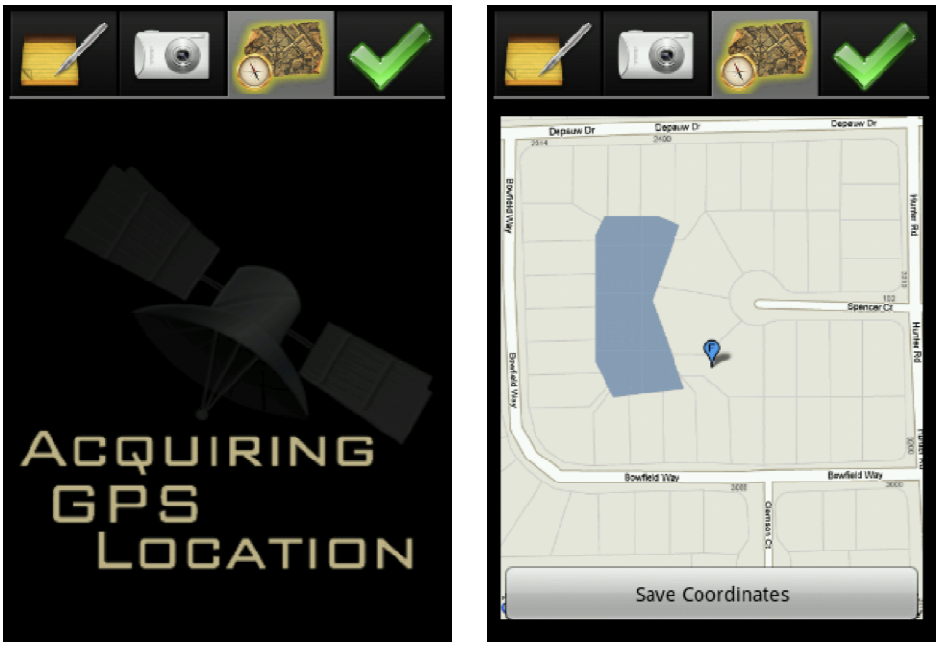

Figure 4 - GPS Tab

Submit Form

The Submit Form tab has two components. The first part is a simple overview that displays what information the user has already saved. The second component is a submit button which will send all information to a PHP script on a server, which in turn will email the information to the appropriate Purdue Maintenance supervisors.

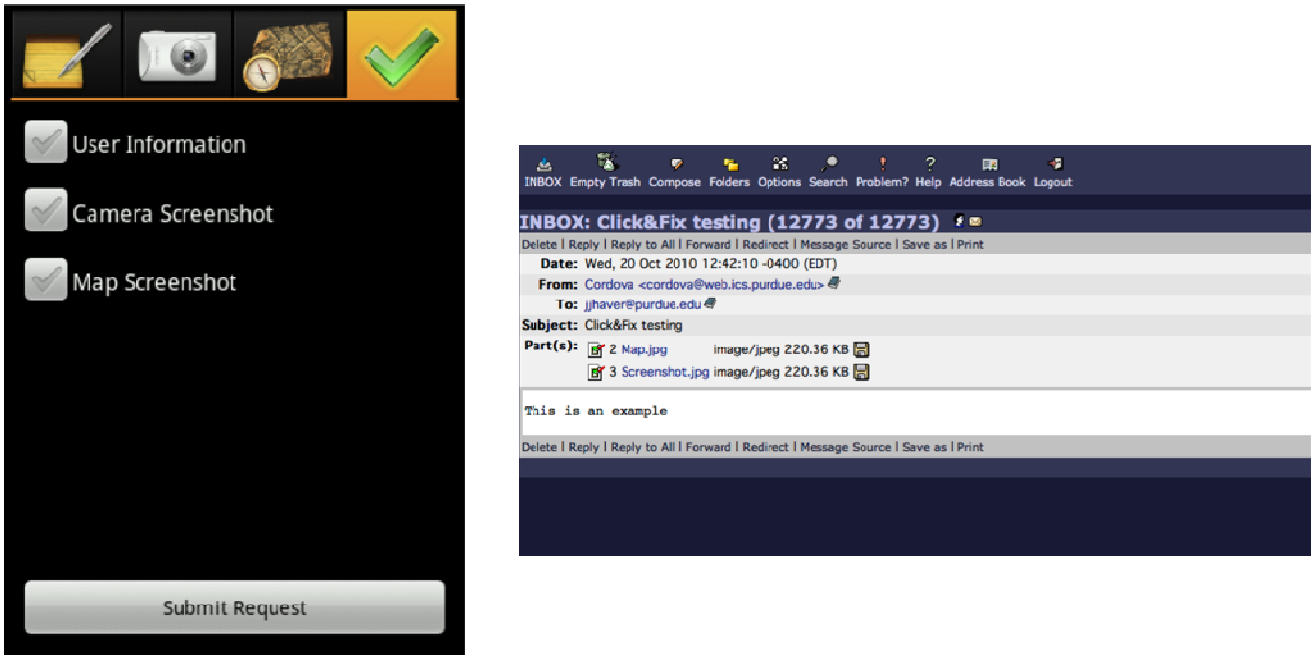

Figure 5 - Submit Form Tab and Sample Email

Additional Feature

Implementation of smartphone's physical menu button allows the user to quit the application as well as display the information about Click&Fix. When the menu button is clicked, an option menu with two options will be showed at the bottom of the screen. When click on the Press The Menu Button option, an information dialog box will be showed. When click on the Quit option, a dialog box will be showed to ask user if want to quit or not.

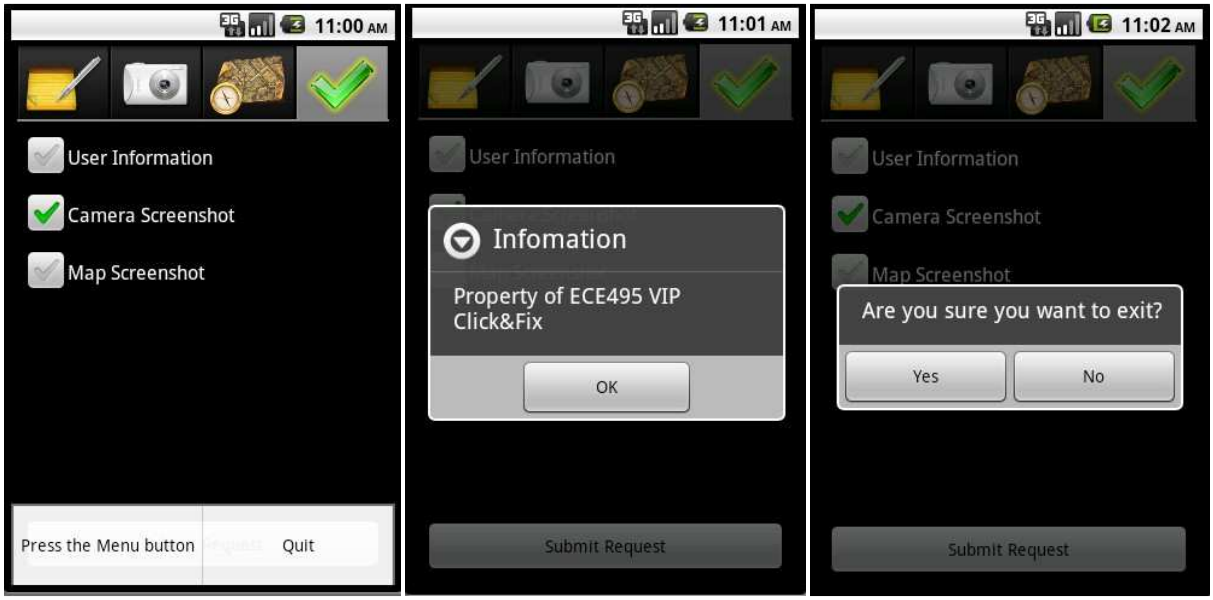

Figure 6 – Physical Menu button options and effects

User Interface

A splash screen will be showed when first enter the application, can use tabs to navigate through the app, when the tab is clicked, the icon will have a glow effect as an indicator.

Figure 7 – Splash Screen and Tab icons

Architecture

Initially we were planning on having a sequential architecture, i.e. users would follow a specific order of actions: the user has to first take a picture of the issue. Only then can he save coordinates. After that, he can finally enter comments and submit the issue.

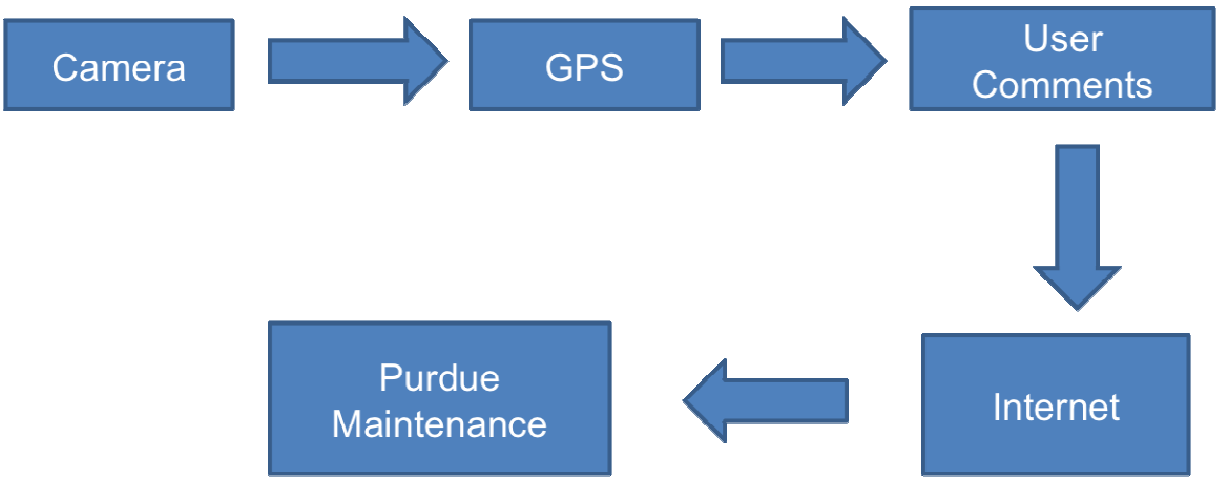

Figure 7 - Initial Architecture

However as the focus of any mobile application should be usability, we decided to go with a more dynamic structure. To accomplish this goal, we changed our architecture to include tabs, which would allow users to go back and forth between the camera, GPS, user comments and submission form

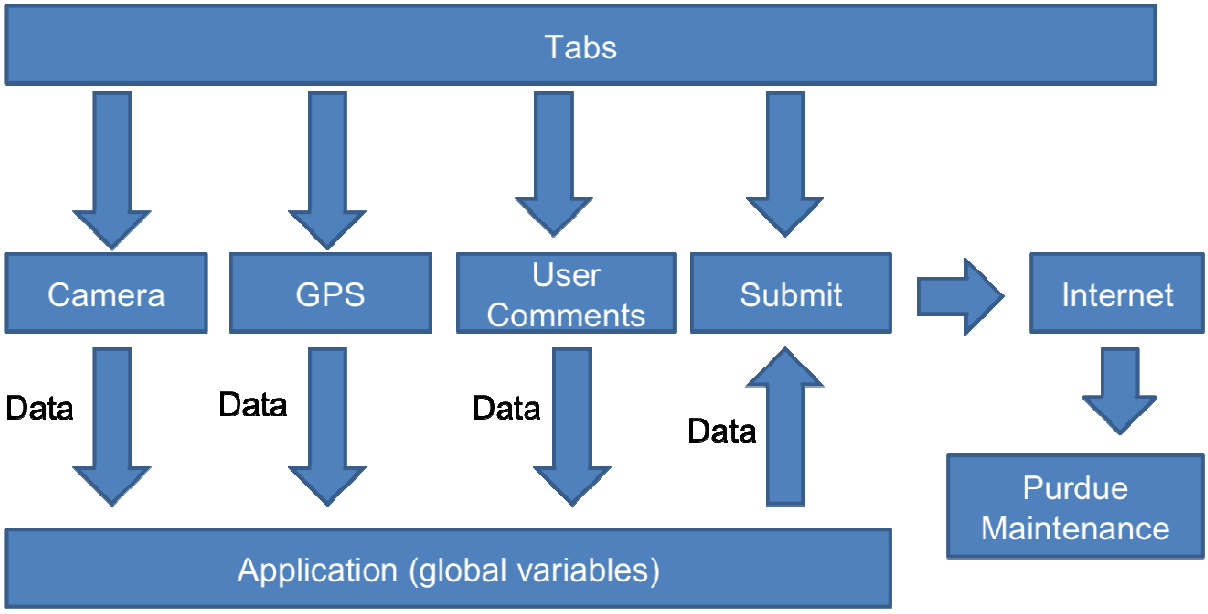

Figure 8 - Final Architecture

There was one problem we faced by using tabs. In the old architecture, since we knew the exact order of activities, data could be passed along from the camera activity to the GPS activity, from the GPS activity to the comments activity etc. But that would not be possible in the new architecture as the user could shift between the different activities. As such, using tabs however required us to implement an Application class. The Application class is a class unique to Android applications and it can be called from any activity. The critical aspect of the Application class is that data in its variables can be saved and called from all activities. Thus the Application class' variables act as global variables.

Challenges Faced and Current Status

One of the biggest challenges was at the beginning as neither James nor Paul had a background with Android or Java. Janam was able to recommend resources and tutorials to get started but learning Android development mainly required hands-on practice to solve problems as they came up. The newly created wiki will provide a easy one-stop access to many resources to help get started in Android development as well as help in areas previously encountered by other students. Another challenge was to become familiar with using a SVN repository for the application code to be shared amongst team members. It ended up being incredibly useful but there was a learning curve in setting up the repository, figuring out which files/directories to include or exclude, and avoiding merges.

A challenge that resulted in a much improved user interface was initially being stuck in a linear progression through the app. In one group meeting, we were looking at some published android applications and came across one that utilized tabs. We decided the app's flow would work much better in a tabbed implementation so as to allow the user to jump around to the various components of the application. A related challenge that was constant throughout involved making the UI sleek, intuitive, and attractive for the user so we tried to include color, graphics, and a straightforward flow to the functionality.

As a specific challenge encountered, the GPS location updates initially were overloading the phone and lagging it down enough to require a force close of the app. This was remedied by increasing the minimum time to check for location updates and the minimum distance change to report such an update. These settings could still be tweaked for optimal performance and battery life but are functional. Another specific challenge was trying to figure out why the camera picture was reported as taken on subsequent launchings of the app. It turned out that the Application Class apparently doesn't re-define the global variables on each launch so they needed to be initialized the in the main activity which solved the issue. Lastly, the camera preview activity must be locked into landscape mode due to a bug in Android. This makes it so the orientation of the phone is unknown and therefore picture might be rotated from the intended orientation depending on how the user is holding the phone. To remedy this, the app reads data from the phones accelerometer from which the phone's orientation can be found using some math. This allows for the picture to be saved and displayed in the correct orientation. Overall, some big and lots of small challenges arose throughout the semester but were overcome through teamwork, determination, and the vast amount of help on the internet.

Current Status of project (compare with proposed milestones):

- Camera: Turn on/off, take/store picture (complete)
- GPS: Turn on/off, retrieve/store coordinates (complete)
- Transmission (complete)
- Text Entry & Additional Features (complete)
- Debugging & Finalizing (complete)

Team Contributions

James Cui worked on the tabs of the application, designed splash screen and icons for user interface, implemented the physical menu button function. Set up SVN repositories for all sub-teams and meeting with Purdue Grounds Department, also worked on weekly reports.

Janam Jhaveri worked on GPS and Submit Form classes, implemented Application class, created PHP script for emailing purposes, set the groundwork for the wiki. Maintained the website and worked on weekly reports.

Paul Mongold developed initial app idea, worked on the user entry form including checks for missing or incorrect entries, implemented camera functionality and features, added the Splash Screen, and collaborated on weekly reports.

The Future: Spring 2011

Two of the team members (James and Janam) plan on continuing this project in the Spring of 2011. The application is almost complete and only a few adjustments are required. These adjustments include switching between saving a map (for outdoor activities) and allowing users to input a building and room number (for indoor activities). Additionally, the emails need to be formatted properly.

Initial contact with the Purdue Maintenance staff has shown positive results. More specifically, we met with Mr. Gary Evans, Director of the Purdue Grounds Department. He showed enthusiasm for our application and is planning on setting up a meeting with his supervisors. However as per Professor Delp's request, we will ask Professor Delp first before setting up any meetings.

Hopefully, an agreement with Purdue can be reached. Once that is accomplished, we plan on working with Purdue to get the minor details (e.g. where to send the emails to) sorted out. As James and Janam are graduated, the tentative plan is to have the application published and marketed by the end of the semester.

Furthermore, James and Janam will help set up the wiki and set formal guidelines regarding SVN and uploading reports for future semesters.