2011

Pedestrian Pal: A Route Recommendation System for the Android Mobile Phone

Bradley H. Bahls
The University of Montana

PEDESTRIAN PAL: A ROUTE RECOMMENDATION SYSTEM FOR THE ANDROID
MOBILE PHONE

By

Bradley Hayden Bahls

Bachelor of Arts in English, University of Montana, Missoula, Montana, 1999
Bachelor of Sciences in Computer Science, University of Montana, Missoula, Montana, 2009

Professional Paper

presented in partial fulfillment of the requirements
for the degree of

Master of Sciences
in Computer Science, Human-Computer Interaction

The University of Montana
Missoula, MT

May 2011

Approved By:

Stephen Sprang, Associate Provost for Graduate Education
Graduate School

Dr. Yolanda Reimer, Chair
Computer Science

Dr. Min Chen
Computer Science

Martin Fromm
Media Arts

Bahls, Bradley, M.S., May 2011   Computer Science

Abstract

Chairperson: Dr. Yolanda Reimer

   With mobile phone technology pervading people's everyday lives, using these devices has become more prevalent for route recommendations while traveling in unfamiliar locations. Current research works to improve algorithms that calculate efficient and effective routes between two or more points. While expedient travel is necessary in time constraint situations, efficiency is not always required. This paper describes Pedestrian Pal, an application built for the Android mobile phone, which offers route recommendations to users based upon their desired specifications. The recommended routes are not necessarily efficient, but rather are paths based upon collected user ratings, aesthetic interests, and users' inputted parameters. During the development of the project I issued a short survey to 23 individuals to collect data to seed the system and solicit desired functionality for the application. Results of this survey confirmed most of the design choices and were incorporated into the system. Special design choices were applied during the interface-designing phase due to the limited screen real estate of the Android mobile device. Large buttons, simple menus, and familiar layouts do not overwhelm the interface and offer a seamless and intuitive user experience. Upon completion of the application development, ten users participated in testing sessions. These tests required the users to walk through eight separate tasks to examine the system's ease-of-use, fluidity of design, and intuitiveness. A follow-up interview questioned the users about the system's menu navigation, usability, and opportunities for improvement. The user tests provided encouraging feedback that affirmed the design choices as well as discovered faults in the system.

Contents

Introduction

When an individual travels to a new city for business or vacation or is simply relocating to a fresh setting, the person needs a method to receive information on the sites to see and routes to utilize. Asking a local dweller can yield good recommendations for destinations: from restaurants to museums to parks and trails. Visiting travel websites or reading books and brochures can also encourage ideas of sites to visit in a new place. Tourists in today's world also have many options with their mobile devices to receive recommendations about a city's attractions and sites as well as routes to navigate through the streets to find these locations. Tools like Google Maps and Bing Maps have robust systems to recommend routes between locales as well as information about those places. While these applications provide a necessary tool for the unknowing traveler and have intuitive and polished interfaces, there are still elements missing from these systems.

The Problem

Recent research and software applications about travel recommendation systems focus on points of interest (Tintarev et al., 2010), efficient travel routes (Ge et al., 2010), and tourist attractions (Baltrunas, 2008); (Kim et al., 2009). These resulting applications mostly look at the destinations themselves as well as efficient vehicle routes to each site. Google Maps provides a large database of sidewalks to allow for pedestrian travel and recommendations, but this tool also finds the most efficient route between destinations. While these ideas serve their purpose effectively, there has been little or no research on a route recommendation tool to offer a user routes that are aesthetic and safe for an unknowing traveler. Navigational applications do not

offer ideas for an athletic enthusiast who wishes to enjoy a walk, jog, or hike in an unfamiliar area. A tourist in a new city may not know of safe neighborhoods to take a stroll, a scenic park for a brief run, or local hiking trails to stretch the legs.

The Solution & Research Goals

The purpose of this project was to build an application for a mobile device that combines Google Maps with the idea of recommending routes that are beautiful, user-targeted, and safe in an unfamiliar city. The project was built for the city of Missoula, MT, USA. With more time and resources the application would have opportunity to expand in scope in the future. The research goals of the project and the aspects I learned included:

- building an understanding of using questionnaires to gather end-user information for requirements specification;

- gaining expertise with the Google Maps API and gps data;

- gaining expertise programming applications on the Android mobile phone platform;

- effectively showing a visual representation of the recommendation data; and writing an algorithm that chooses paths for the system based on a set of heuristics;

- applying user evaluation and testing;

- utilizing all phases of the user-centered design process;

- working with SQLite databases and the Firefox plug-in, SQLite Manager.

Methodology

In order for a user to receive recommendations the application must have some data in the system to offer users walking or jogging suggestions. To seed the application with data, as well as finding functional requirements, I developed and issued a questionnaire. An area's residents can offer the best choices for locations of local outdoor activities (walking and jogging). Recommendations from these people can also offer areas to avoid—where a certain neighborhood may not be safe or not offer scenic views. The questionnaire was issued to a class of university students—effectively utilizing the expertise of local residents as a valuable resource to build a robust system. 23 students completed the survey and the results were incorporated into the system (more details are found in the section 'Initial Survey').

After the program was seeded with this new data, users were now able to apply their own feedback via the application itself. Once the user completes a recommended route she is able to select the route and choose a rating from a 1-to-5 Likert scale. This rating represents the user's likes and dislikes for the route (1 equating to 'dislike' and 5 paired with 'like'). One limitation of the system is this rating system. The application in its current form accepts only one value to rate a specific route: a number representing the overall feeling for the route. Ideally, the system would offer several factors to rate, including: aesthetic appeal, difficulty, wildlife, and traffic. In this case, a user looking for a new route could enter several parameters and receive a recommendation more catered to what she wanted.

Despite these limitations the system has other options for the user to explore. Users are able to add new recommendations to the system via the interface by building a route of their own. The user has the ability to choose if her feedback will mix with other users or she can decide to keep personalized routes. With a global option recommendations utilize all other users'

feedback to create routes. These functionalities of retrieving route information (such as geo-coordinates) are accomplished through current Google Maps technology.

The visual representations are also accomplished through Google Maps in conjunction with Android's graphics libraries. Current technology from Google allows custom maps and routes through its API. Route data is collected from Google Maps and the program uses this gps data to build lines between all retrieved points and are then visually overlaid on a map to show a recommended path. Accompanied information about the path and possible starting points and destinations are added by using markers (graphically built notifications). These markers can be added into any map and show up as small stars situated next to the places they are describing. Using the API, the program offers a familiar visualization (since Google Maps is a well-known application) and portrays a recommendation accurately. All information can be saved in the user profile (via the mobile phone's integrated, lightweight database) so that routes and data can be retrieved again.

Since the application has a narrow focus for the city of Missoula, MT certain input from the user was unneeded. The user is able to enter information to receive recommendations in correct areas. For this project I wanted some input parameters available for users: distance of route, location in the city, and desired elevation gain/lost. The location input can be achieved by letting the user set a point on a map and the distance to search from this point.

To allow for use in any location the application was built for Android's mobile phone development platform. Many of the libraries for this platform are available through open source, while other packages (specifically Google Maps API for Android) can be obtained through free registering. Free mobile phone emulators that interface through the Java development platform Eclipse are available for download that enable development even without a mobile device.

Pedestrian Pal was programmed on one of these emulators due to monetary constraints. A working application on one of these emulators (in theory) is functional on an actual Android phone, although, complete testing is not feasible without the device. Many discrepancies and errors can occur on an actual device that are not apparent (or will not even occur) on emulators.

The main focus of the project was to create an interface that allows simple and intuitive use. The underlying mechanisms of the recommendation algorithms are a secondary aspect of the project and received less attention, but these algorithms provide adequate recommendation results in order for the system to be usable. As a result, a simple method was needed to give the user feedback. The routing recommendations are calculated by a simple process of employing user-selected geo locations followed by a step to eliminate potential candidates through two other inputted parameters.

Initial Survey

In order to "test the waters" for potential users, and to design an application catered to their needs, I developed and issued an initial survey. This survey asked people about their walking and running habits in the Missoula area and questioned if they would find a mobile phone application (like the one I proposed) useful in their everyday lives.

In order to utilize individuals for surveys and testing I needed to obtain the approval of the Institutional Review Board (IRB). This process required detailed paperwork of my planned surveys, procedures, and recruiting methods. I was required to wait for IRB approval for the study before I could take action with the questionnaire. I issued the initial survey after I had already begun building the Pedestrian Pal application. In view of time constraints I needed to

begin programming the project, and some aspects of the application were already in place before I could issue and compile results of the survey.

I asked 8 questions of this described subject matter:

1. How often do you go for a jog or walk? Rarely (less than once a week) Occasionally (Once a week) Frequently (More than once a week)

2. Do you take in consideration the setting and location when you jog or walk?

   a. If yes, what do you look for when you walk or jog?

3. Do you consider elevation gain when you walk or jog?

4. What local trails do you use?

5. Are there any neighborhoods that you use when you walk or jog? Please list any.

6. Are there any areas in Missoula that you avoid when you walk or jog? Please list any.

7. Would you find a computer application that recommended foot travel routes useful?

8. What types of options would you want from this kind of application?

I presented the survey to an undergraduate Mathematics class of twenty-seven individuals. In all, I received 23 responses. I compiled all responses into a spreadsheet and categorized the results (complete results are displayed in Appendix I).

The results of my survey offered some good initial data that gave me direction for the project. First, question number 8 brought many insights in functionality for the application. Responses showed some variation, but there were a few stand-out answers that grabbed my attention. Nine of the respondents thought distance for routes would be useful in an application

like this. Five individuals found that elevation statistics for a route would be helpful.

(Unfortunately, I found the actual implementation of elevation data gathering to be difficult and

was not included in the final application.) Other responses showed the need for the amount of

traffic, time to complete a route, difficulty of a route, and scenic aspects like vegetation, views,

wildlife, and rivers.

As mentioned before, in order to fill the database with accurate and reliable data I used

the survey's responses to questions 4, 5, and 6. Questions 4 and 5 supplied a number of

locations that local people found to be desirable locations to walk or jog. The Kim Williams

Trail and the University area were the most common responses to these questions. Replies to

question 6 clued me into areas that individuals avoided in Missoula. The North side of town was

the most common response to this question. As a result, I built no routes in the system that

travelled through this area.

Project Requirements

The initial survey results flushed out many of the needed (or wanted) functional

requirements for this application. The potential end-user responses showed the basic functions

and options required for the software in order to satisfy the users' stated needs.

Usability requirements stemmed from some of the survey responses but also were drawn

from analysis of the application platform. The Android mobile devices range in screen sizes

from device to device and user interfaces can change significantly on different displays. To

avoid major discrepancies and possible problems (buttons that show only half on a screen or are

absent entirely) I carefully designed the interface with cross-platform flexibility in mind. There

are simple coding habits to adopt that can maintain a stable interface in many devices. Using

absolute layouts can look great for a target-specific device but may be in danger of fragmentation

on another. Applying a relative layout can ensure a static interface on any device. In addition,

avoiding absolute pixel sizes is a must because the same discrepancies can occur as mentioned

before. Several solutions exist, from wrapping elements to the size of the parent to using a

density-independent pixel (dp) unit that will maintain a size ratio according the size of the

display.

Finally, data requirements were developed from analysis of the kinds of information that

are needed to represent the objects in the application. For example, the application depends

heavily on geospace coordinates to build routes, load and save route data, and responding to user

input. As a result, the software needed to handle pairs of latitude and longitude coordinates and

parse and distribute them as needed. Other simple and obvious data are needed to store users'

information and feedback, such as, first and last names, usernames, passwords, and ratings.

The following lists show the final project requirements:

Functional Requirements

The user will be able to:

- Create a profile with name and password

- Rate routes based on the overall appeal

- Rate sections of routes

- Create new routes and sections

- Choose whether to use global, group, and/or personal data and ratings

- Request a quick suggestion for a route based on location in an urban setting and distance of routes

- Hide/show overlays

- Login to the system and Logout of the application

- Select a route from a list

Data Requirements

- Geo-coordinates

- Personal information (Users)

- Location tags (strings describing locations—street names or specifics about setting)

Usability Requirements

- The user should be able to receive a route recommendation (drawn on map with overlain information) within one minute of starting the task.

- The user should be able to rate a route or a section of route under 30 seconds.

- All colors of the interface (menus, buttons, and overlays) will remain consistent throughout the application.

- The buttons and menus of the interface should be sensible—with icons and choices that are not confusing or misleading.

- The system will be intuitive and have a simple and short learning curve. (Depending on the user's ability, learning the basic functionality of the system should not take longer than 15 minutes.)

- User documentation will offer helpful insights, with photographs and walkthroughs, for using the application.

- Any error messages will offer clear problems and solutions.

Database Design

In order to retain route and user information for later access the application needed a means to store these data. The Android operating system uses SQLite which is an open-source and lightweight database that does not require a server to perform its functions. Since the database already exists in the operating system, the application can take advantage of the availability without the need to install a bulkier database.

Creating tables can be accomplished two ways: programmatically or using a Firefox plug-in named SQLite Manager. The latter proved to be the easiest for manipulating the application's database since the interface is graphical and simple to use. The manager automatically creates sql commands to create tables and to perform insert, delete, and update queries. Although, when a developer deploys to a device, some problems exist when using this method. These problems are discussed later in the paper.

One recent addition to SQLite had allowed users to reduce development overhead in database design. Until recently, the system did not support foreign keys and the management of

deletions and updates was put upon the developers to maintain. With the advent of foreign keys

the developer could set cascading commands on the tables. This means that parent tables with

data (when changed) will cascade any changes down to child tables with linking data. One more

step is needed to allow for the use of foreign keys. The Android system, as a default, disallows

the use of foreign keys unless specified otherwise. Upon loading a database in an application, a

command in the code must be specified to change this default.   Here is the command:


```
db.execSQL("PRAGMA foreign_keys=ON;");
```


"db" represents the variable name of the database. The function "execSQL" executes a specified

SQL command. The text in the parentheses is the actual SQL command. In this example, the

command in the parentheses tells the system to allow foreign keys.

A helpful tool found in the Android Development Kit for the Eclipse IDE is the database

management system. This tool allows a developer to find databases in applications via a file tree

and pull them from a device or emulator. Using SQLite Manager one can then view the structure

of the tables, the data contained within the tables, and any constraints enforced within the

database. Rather performing any changes to the structure programmatically (an unneeded task),

a developer can use the SQLite Manager interface to add constraints and fields or change data

types. The database can then be pushed back onto the device or emulator with the changes in

place. Populating tables can be accomplished the same way. Programmatically adding data for

testing purposes, or adding data via the application's interface can be time consuming and

unwieldy. SQLite allows a developer to fill the database tables with data simply by typing into

provided textboxes (the manager creates SQL statements behind the scenes to perform the

insertions, updates, and deletes). Once again, a programmer can push the database back onto the device using the management system in Eclipse. This process provides quick, testable data for the application.

For deployment of an application to a mobile device this option does not work. For security reasons neither the database structure nor the data within the database can be pushed directly onto a device. To move data onto a phone a developer must programmatically copy the database from one location and move it to the device. This method moves the structure of the database as well. Alternatively, the same database can be created through programming functions. The application can check if a database exists. If the program does not find a database then it will create the structure programmatically. Test data can be hard-coded into the database in this way as well.

Once I found the ways to move past these obstacles I concerned myself with structure of my database. I knew I needed a table to hold user data and I created this quickly. The more complex part of the data was storing routes and their attributes. To pull routes from their storage space and into the application I would need to be able to store all the points for each route. In addition, since I wanted to be able to rate sections of routes I would need the database to recognize points contained a section. A natural hierarchy fell into place. Routes contain sections and sections contain points. Each point is represented by a latitude and longitude pair. The resulting schema is shown in **Figure 1**.
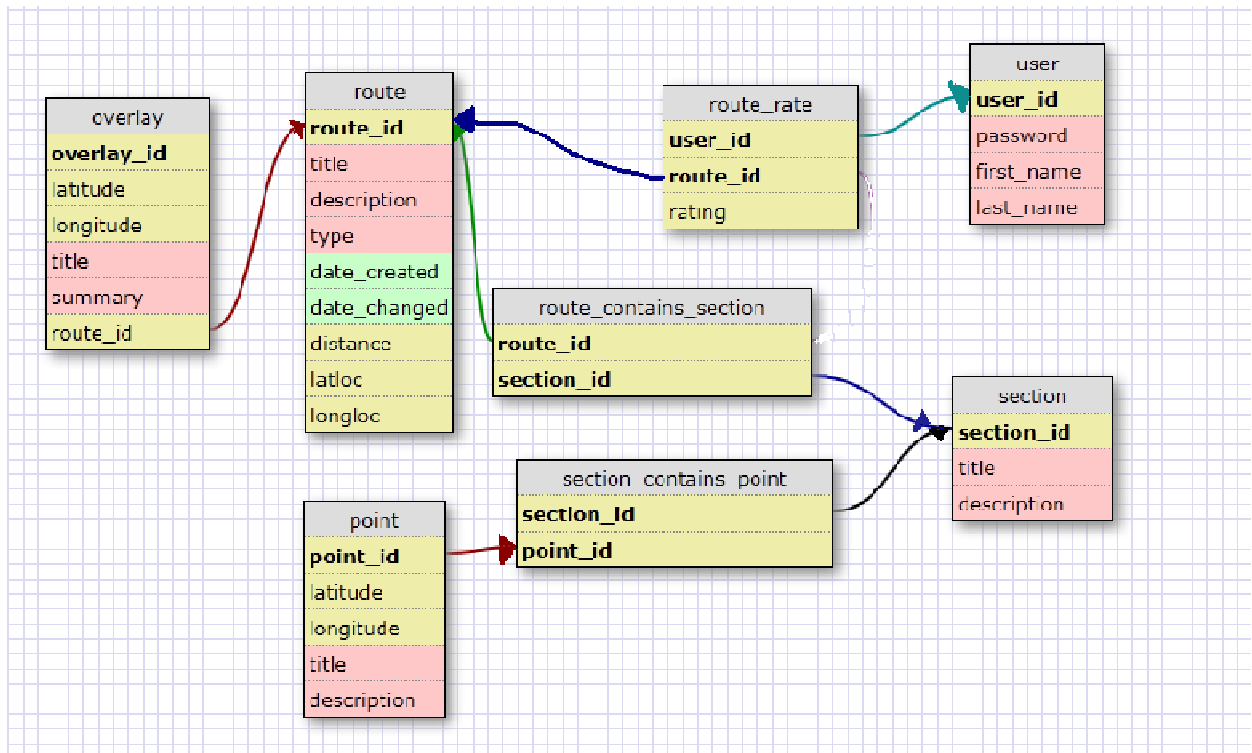
Figure 1: Database schema for the system.

User Interface Design

The key factor in this project for most design decisions was the platform for the target application. The Android mobile phone supports screen sizes from 2 inches(small) to 7 inches(large) over a variety of devices. Since these screens are comparatively small (in contrast with desktop and laptop PCs), I took care to allow for user touch inputs. In regards to buttons, they must not be too small in order to prevent accidental touches from the user when aiming for a nearby target. Plus, I designed an optimal button size for mobile devices as shown in other research (Perry & Hourcade, 2008; Parhi, Karlson, & Bederson, 2006).

16

I created the initial designs for the user interface with pen-and-paper sketches. **Figure 2** displays an example of these original sketches. Two screen prototypes are shown in the figure. The left image represents what a user would see after she has chosen the route section rating option. The right image is the screen after the user has chosen her rating. (See Appendix IV for all sketches.) Even these first drafts exhibited the careful consideration of the mobile phone interfaces.

Another restriction and concern apparent in Android devices that stems directly from the size of the screen is the actual screen real estate. An interface cannot be jumbled with buttons, labels, textboxes, and widgets and still offer an easy-to-use experience. With too many items arranged on the interface a user may become overwhelmed and confused even when performing the simplest of tasks. I designed each screen to contain only a few buttons and menu options in a single view at one time. **Figure 3** shows a simple route summary screen that has a clean interface. This basic strategy ensures less user confusion and avoids the need to scroll off screen for more selections. Preventing scrolling allows users to recognize all available options
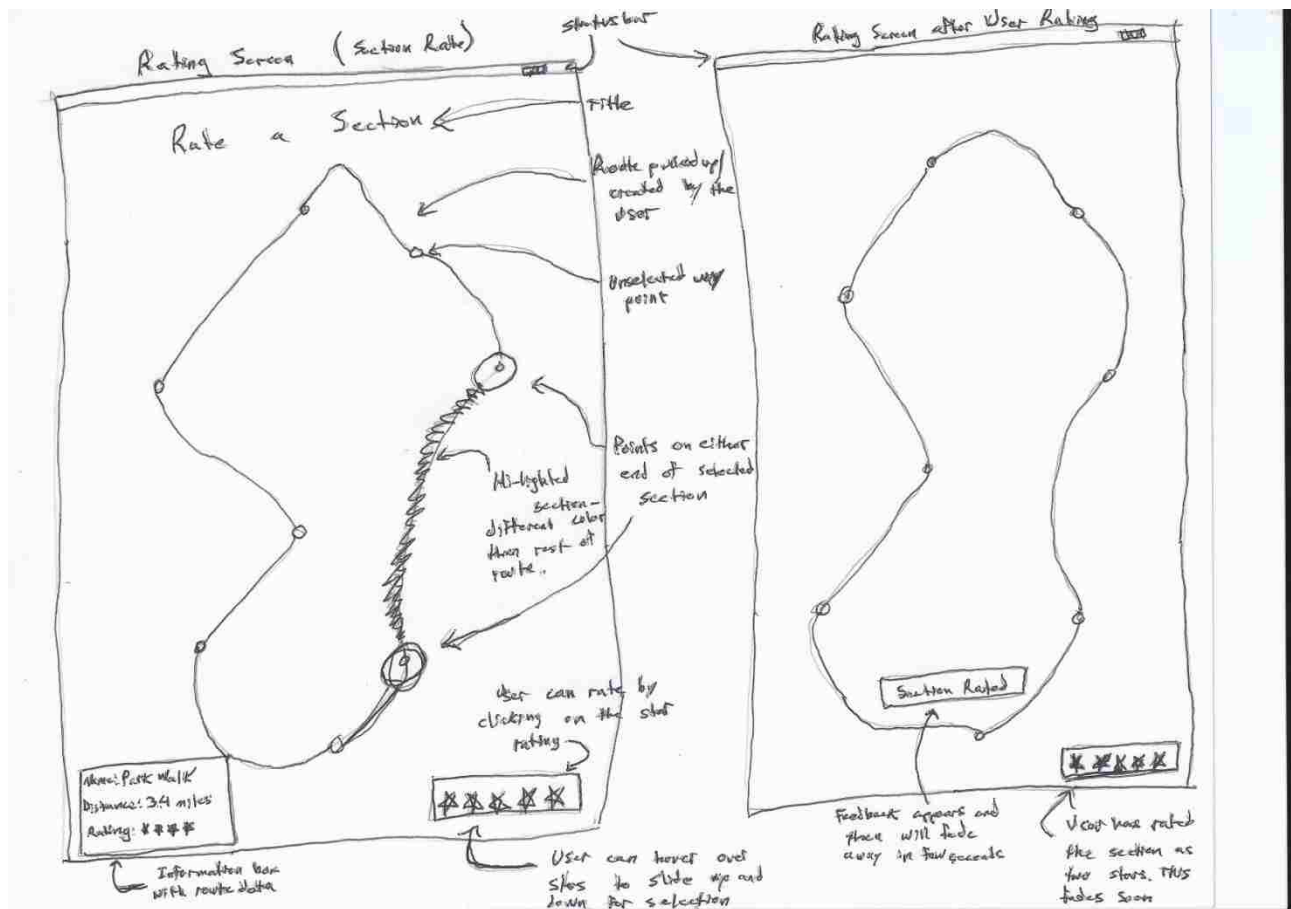
Figure 2: Sketch of section rating screen (left) and a screen immediately after a rating (right).

(Gevin, Sefelin, & Tscheligi, 2006). One method to reduce the complexity of the interface is to remove unneeded items and simplify the work area. In traditional applications built for devices with large screens (personal computers and laptops) input boxes often have corresponding labels adjacent to them to indicate the desired user input. Textboxes in Android have an attribute called "hint" that can be assigned a text value. The hints are overlaid onto the actual textbox (see **Figure 4B**). To save space in this limited display environment, the application does not contain the corresponding labels but instead utilized these hints. In addition to preserving screen real

estate, the text-hints craft a more organized and cleaner interface and improve the visibility of cues for filling out the textboxes.

Figure 3: Route view screen exhibiting the simplicity of design.

Several decisions were made in regards to graphically building a route in the Pedestrian Pal interface. In order to distinguish between sections within a route each section is drawn with alternating shades of blue. For section rating functionality, a user can easily select a desired

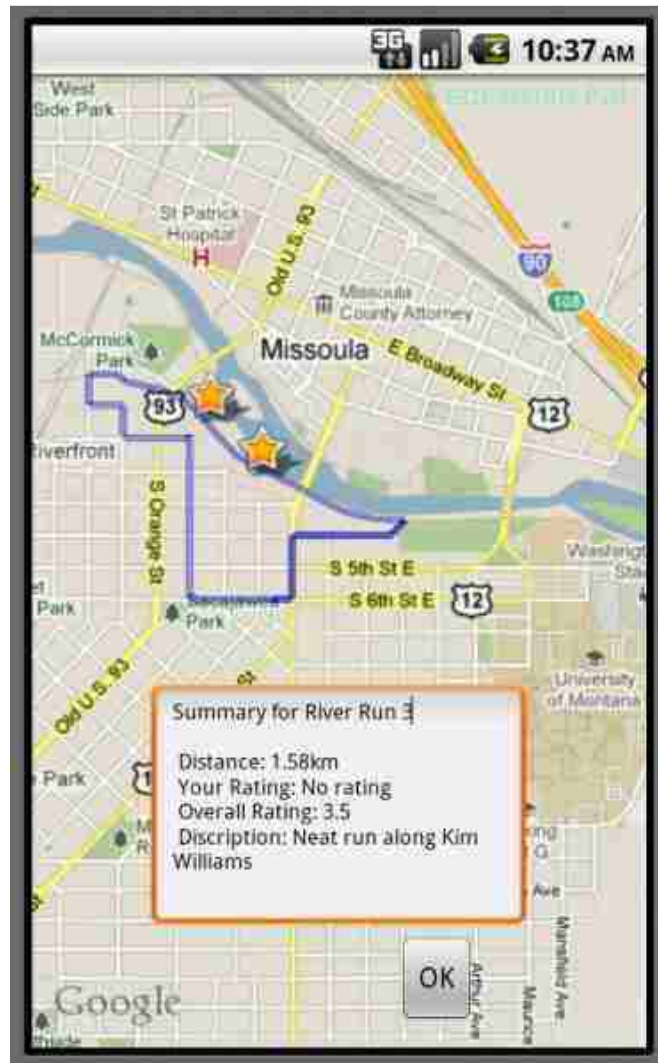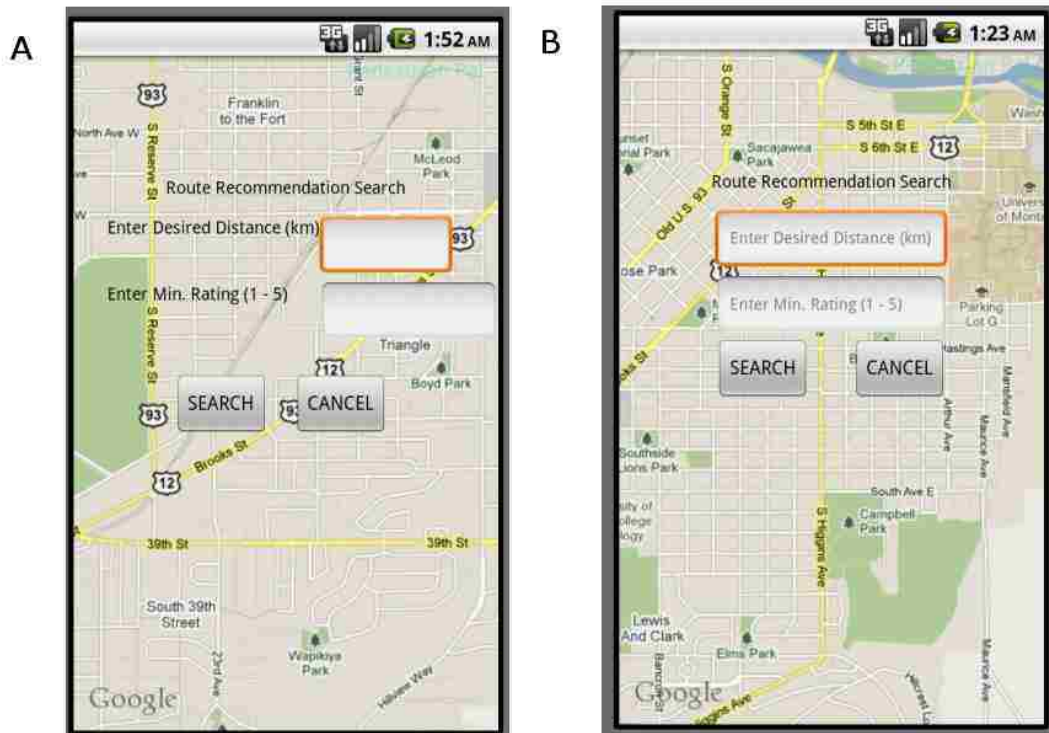**Figure 4: Demonstrates the differences of using (A) labels in conjunction with textboxes and (B) applying text-hints to the textboxes to save screen real estate.**

segment by visually distinguishing between the two shades. One other decision with concerning

route visuals was to add opacity to the colors. When testing with solid colors during

development the interface appeared clunky and unrefined. In addition, solid blues (and other

colors) tended to frequently cover street names on the Google maps—hiding needed information

for the user in terms of orientation on a map. One final decision involved width of the path line.

The final width was set at a stroke width of four pixels (see **Figure 5**) (the Paint class for

Android uses the method "setStrokeWidth" to apply the pixel width). While experimenting I

noticed that route lines smaller than four pixels were quite thin and difficult to see on the map.

Conversely, larger stroke widths (thicker lines) would cover too much of the map interface

especially when the view is elevated further from the landscape. I considered the optimal zoom

setting when deciding line width as well. Using data from the initial survey, I found that most

routes appear to not exceed a total length of 10 kilometers. This means the optimal zoom level

(Google Maps contains a variable dubbed "zoom level" that tracks the current setting) is set at 14

or 15. The chosen line width appears visually pleasing for this zoom position. As mentioned

before, a longer route would require a user to move further back (zoom out) to view the path in

its entirety. If the width were too thick the route path would cover too much of the street and its

surrounding area (covering parts of city blocks) and cause the interface to appear confusing and

unpleasing.

Figure 5: Side-by-side comparison of two line widths in the interface. The thicker 9 pixel width (A) shows color bleeding onto the surrounding city blocks) whereas (B) exhibits a clean line of 4 pixels following the street.

In order for a user to trust an application, the system should respond with proper feedback

to the user's actions. Pedestrian Pal contains a few of these responses to ensure information has

been received accurately. A useful object in the Android library named "Toast" allows a

developer to insert feedback to any action a user may perform. This class builds a simple text

display with a contrasting background that can be overlain onto the interface for a specified

amount of time. For example, when a user wishes to rate a route (offering the system input) she

navigates through the appropriate menus and options to complete the simple task. When she

selects the final option (for example, a numeric rating for the chosen route) the system responds

with one of these Toasts at the bottom of the screen as shown in **Figure 6**. This feedback builds

user confidence in the system.
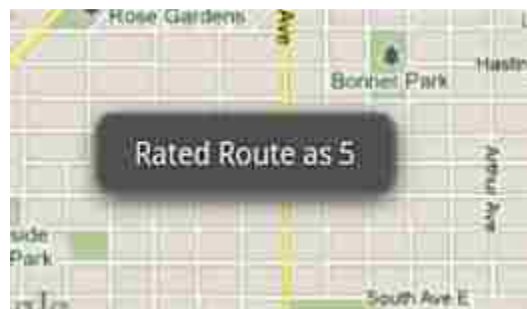
Figure 6: Example of a "Toast". This form of feedback appears at the bottom of the screen. In this example, a user has rated a route.

Route Creation Walkthrough

To illustrate some of the basic functionality of the application as well as present a few of

the visuals found in the system, I will now present a walkthrough for creating a new route. I will

offer a scenario for a user to perform and I will show a series of screenshots of the process to

complete this scenario.

For this walkthrough, the user has tried a few of the options and now she wants to create

her own route and have it stored in the database for later use. She has jogged a few times in the

south hills of the city, near the road that leads to Pattee Canyon, and she has a great route in mind. She wants to plan out a three-kilometer run in this area.

The initial screen (**Figure 7**) shows the main menu of the application. (The first figure of this series of steps shows the hard keys along with the emulator's screen. The following figures only show the screen.) In this figure, the user has pressed the "Routes" button.
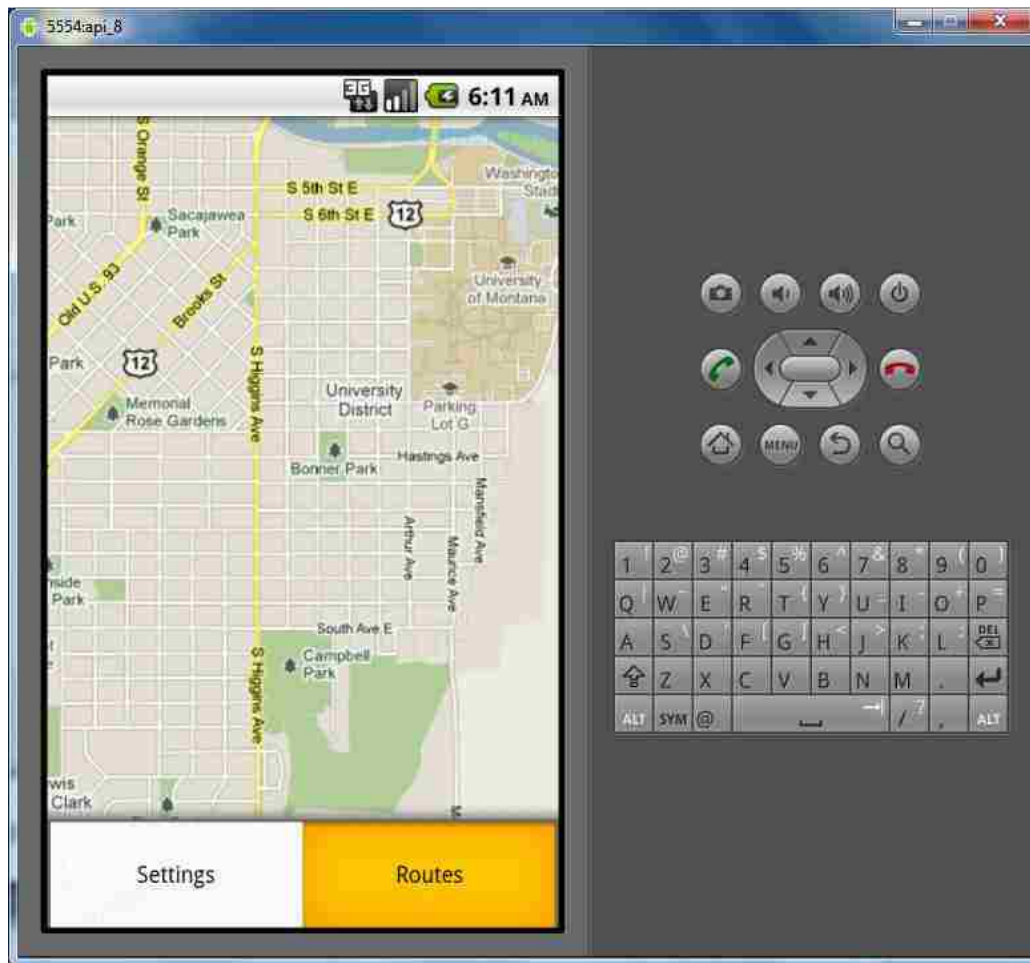
Figure 7: This screenshot shows the resulting screen after the user has selected "Routes" from the menu.

Next, the user selects the "Create a Route" option from the option list in the "Routes" menu. This selection is shown in **Figure 8**.
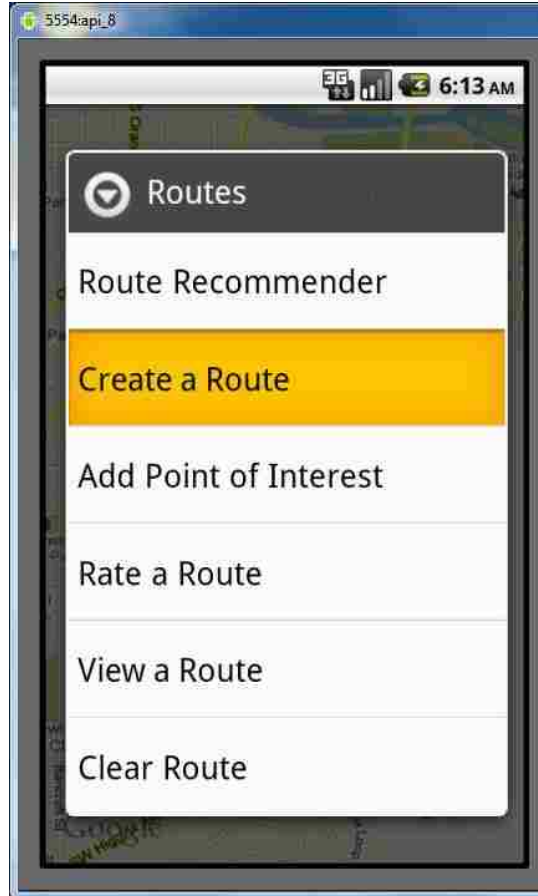
Figure 8: "Create a Route" is selected from the list.

The following screen (**Figure 9**) displays the start options for creating a route. A "Toast" tells the user to click on the map. This sort of guidance leads to less confusion for the user.
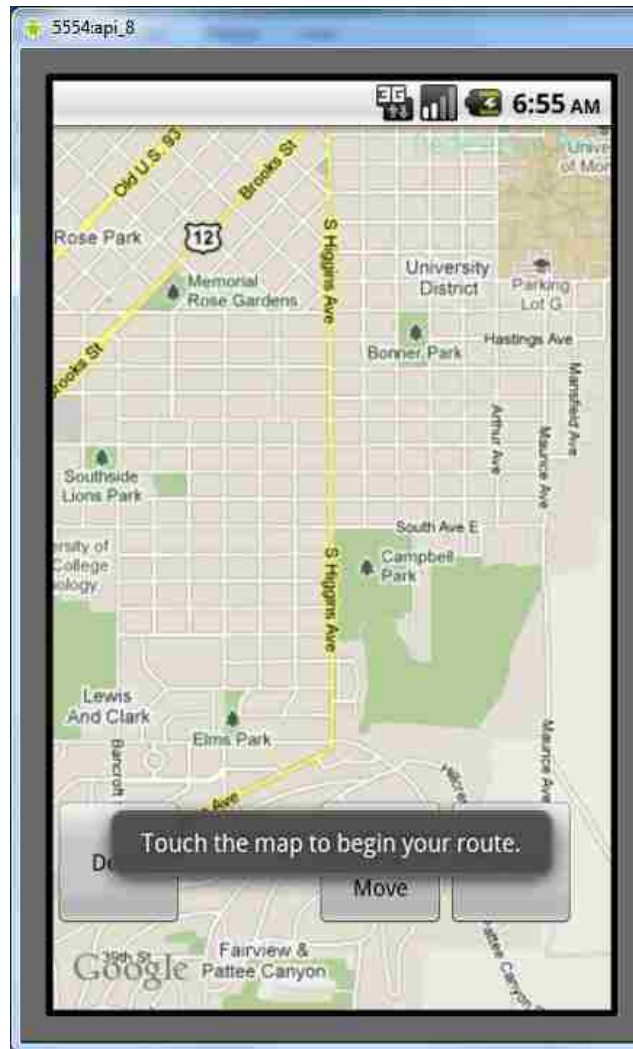
Figure 9: Screen presented after user has selected "Create a Route". Feedback is displayed at the bottom of the screen to instruct the user what to do next.

The user now finds the starting point for her new route and touches the map to begin a new path. A blue circle appears where she touches and the map moves to center this point in the view. **Figure 10** shows the screenshot after the user has touched the screen. Also, notice the "Zoom and Move" button in this screenshot. At any moment the user wants to pan around the map or zoom in and out she can press this and perform these functions. Pressing it again allows her to return to creating her route.
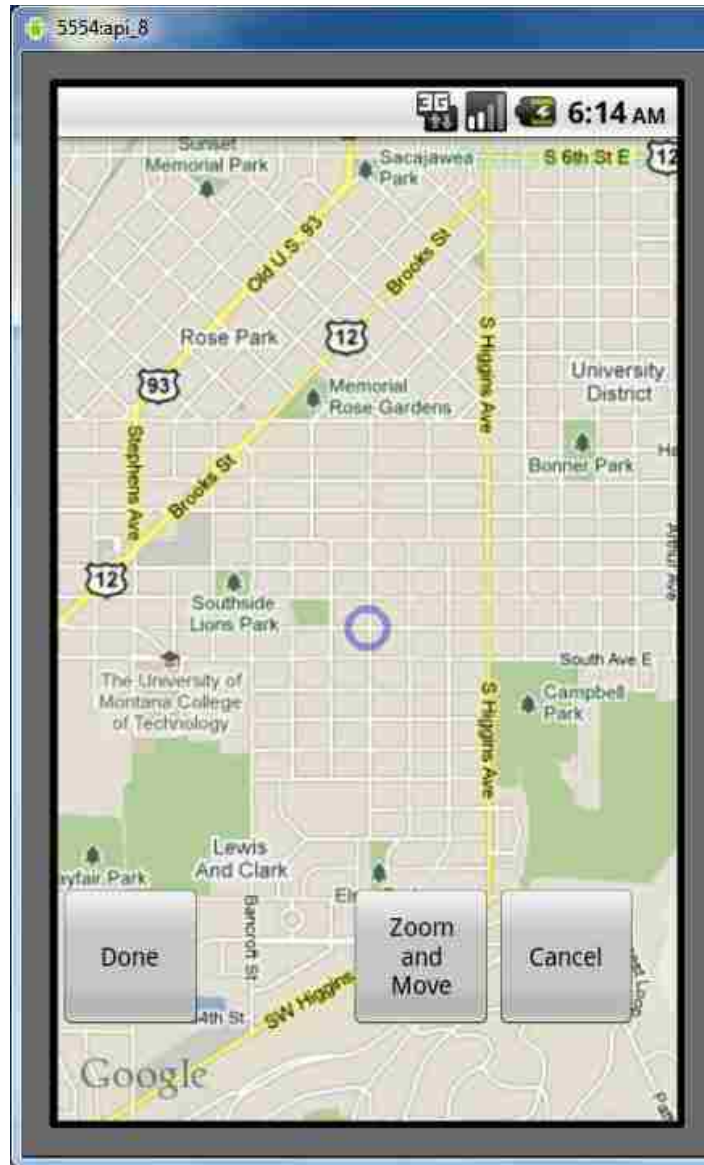
Figure 10: The user has touched the screen and a circle has appeared to show the starting point of the route.

Next, the user wants to build the route of her choice. She proceeds to touch points on the map and the sections of the route connect each new point. **Figure 11** displays the first section and **Figure 12** shows after the user has chosen all points. Notice that the "Undo" button is now

available and the user can remove sections in reverse order if she does not like the way the

application has drawn them.
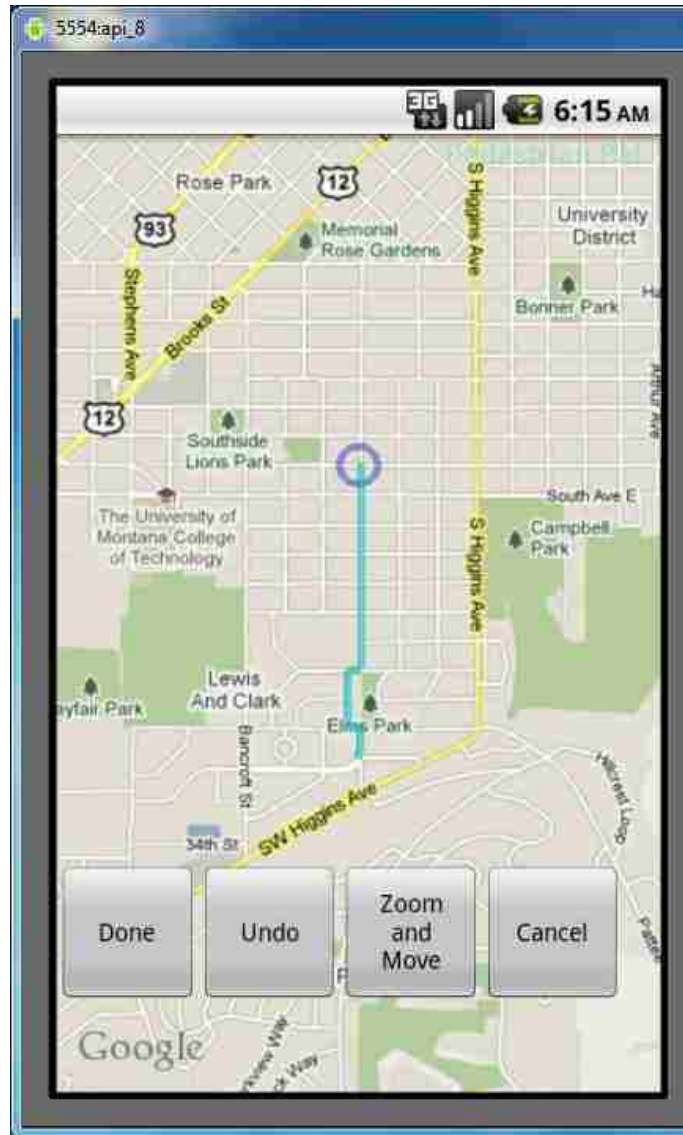
Figure 11: The user has selected a second point on the map and the application has drawn a new section.
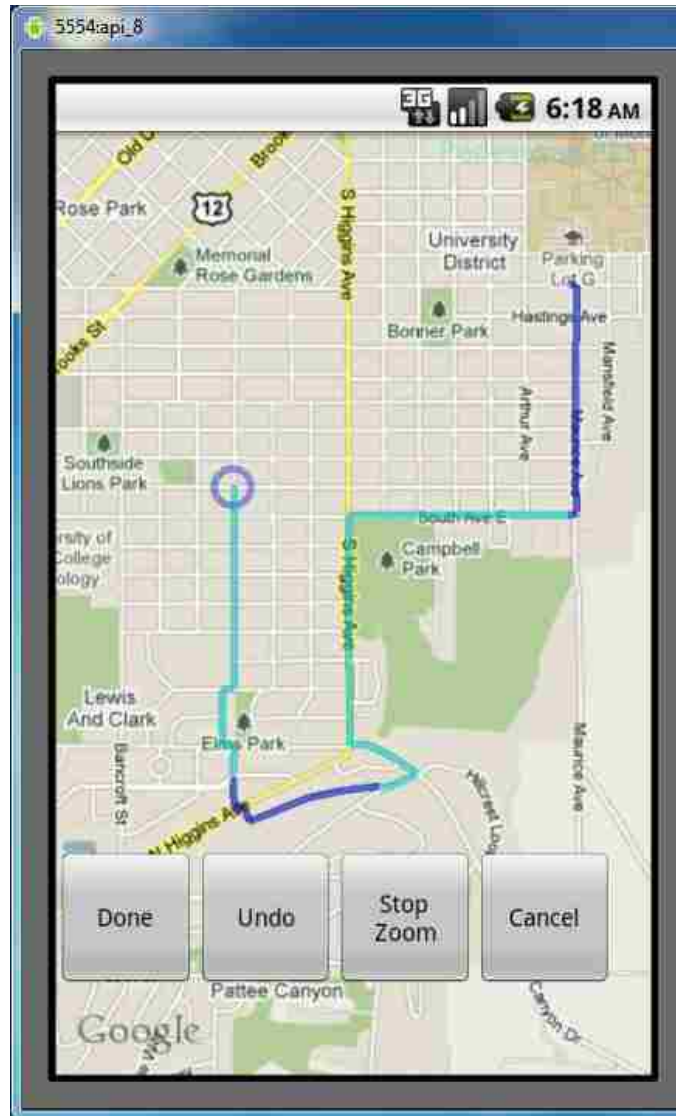
Figure 12: The user has selected five points and four sections have been drawn.

The user is satisfied with the route she has drawn and she selects "Done". The system will automatically create a circuit by connecting the first point with the last point. The system will now bring up the "title" and "description" textboxes. This screen is shown in **Figure 13**.
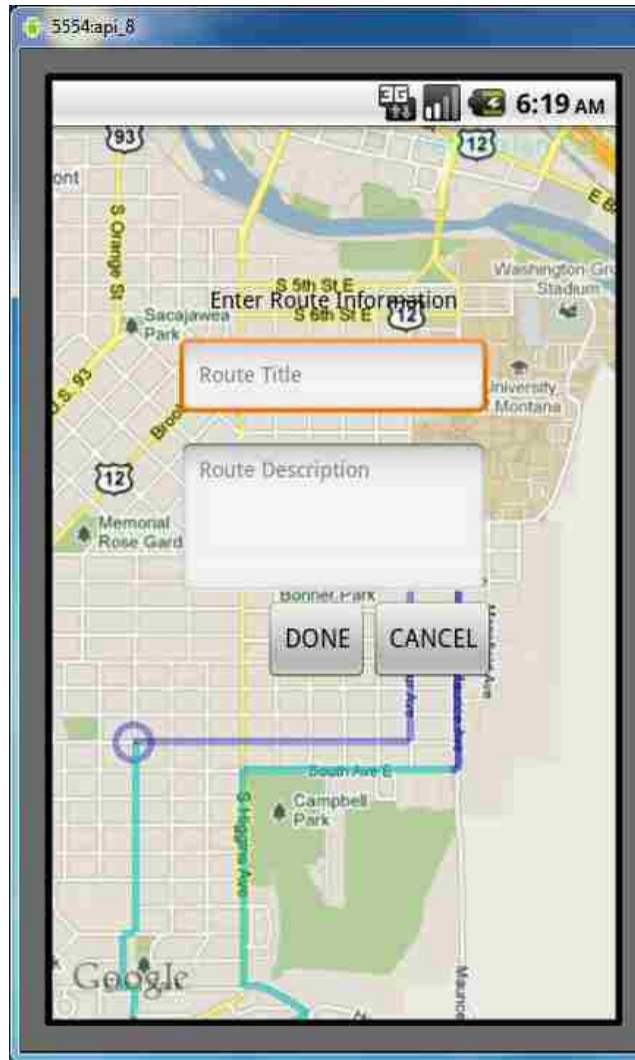
Figure 13: The user has pressed "Done", the application completes a circuit (connecting the end and starting points), and two text boxes appear for the user to enter information describing the new route.

She has created the actual route. Now the user can name the new path and enter a description about the route in the spaces provided. Satisfied with her entries she presses "Done" and the application shows that the route has been created and supplies a summary box (**Figure 14**). The system automatically calculates the length of the route and displays this distance in the summary. The user can press "OK" to clear the summary and she can pan around the map to view the completed route (**Figure 15**).
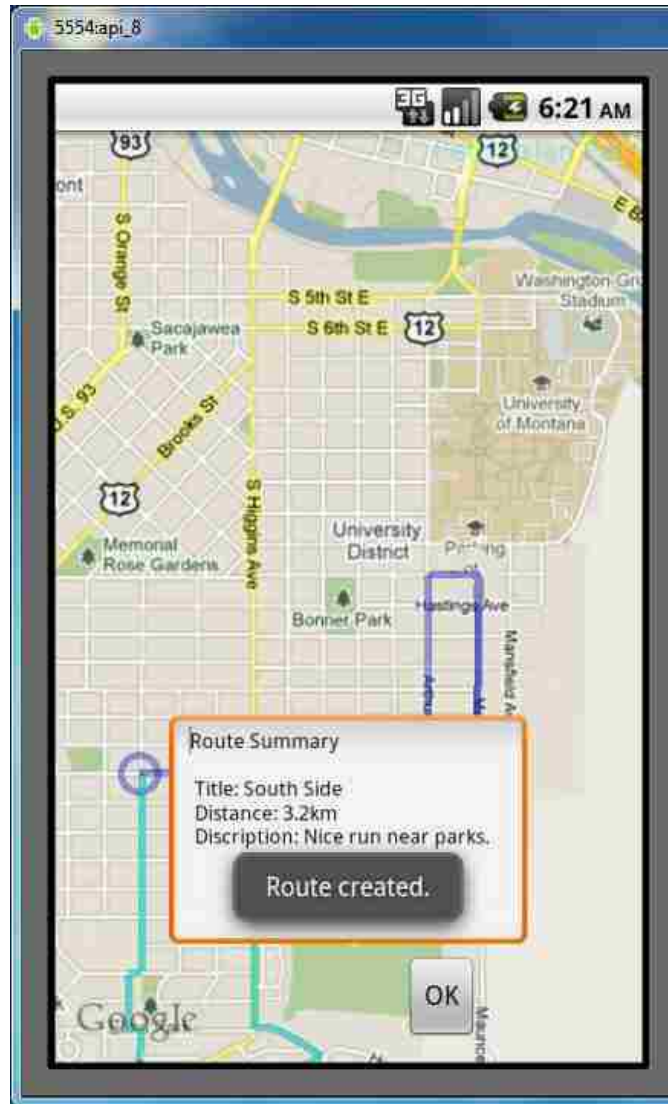
Figure 14: The user enters her desired information for the title and description of the route and presses "Done". The two text boxes vanish and a new summary box appears. Another "Toast" pops up offering more feedback to the user.

Figure 15: The completed route after the summary screen has been cleared by the user. The user has also panned the map to center the path in the view

She has completed her task and the route's information is automatically saved into the database. The system will check for a route title duplicate and issue a prompt to change the name if it currently exists in the database. Now, the system may recommend this route to her if her inputted specifications match this route. In addition, she can always view the route by finding the appropriate option in the menu and pressing the title of this route.

User Testing

Introduction

    To thoroughly test the system and expose any inconsistencies, design errors, and confusing aspects of the system I performed a series of user tests. The test consisted of 8 tasks that encouraged users to step through the application's various options and functions. Some tasks were as simple as creating a new user account or reading the help guide. Other tasks increased in difficulty and contained many more steps. The harder tasks required users to build their own routes and load routes from the database to add points of interest. These user tests required the approval of the Institutional Review Board before I could begin.

    Here are the tasks the users performed during the user testing. I have also listed the associated questions:

    Task 1. Creating a New User Account—You have just downloaded the Pedestrian Pal application to your mobile phone and before you can start using it you need to create a new account.

    Task 1 question(s):
How do you know an account was made?

    Task 2: Read Instructions—You want to learn a little about your new application. Read through the guide to get some basic knowledge of Pedestrian Pal. Exit the guide when you are finished.

    Task 2 question(s):
What two objects can you rate in this application?
What does the application automatically do when you click "Done" while the route creation tool is active?

    Task 3: Change Your Level of Recommendation—You decide that you dislike having other users' routes showing up in the list of available routes. Change the level of recommendation to have only your routes show up.

    Task 3 question(s):
How do you know the level has been changed?

Task 4: Find Paths for Jogging—You are in the Missoula area visiting a friend and you would like a nice place to jog by the Clarkfork River. You want to a nice 2-kilometer run, but you aren't sure of a route. In addition, you would like to use a route that has a rating of at least 3. Use the system to find a path with these specifications.

After finding the first path you also wanted to find a route near the university. You don't mind the distance or rating this time around. Use the system to find this route.

Task 4 question(s):
What is the title of the first route you found?
What is the title of the second route you found?

Task 5: Build Your Own Route—You have decided to build your own route. You want to take a nice stroll through the university area and next to the river. Make a path in these areas and name the route as "River Walk". Add a short description about the route.

Task 5 question(s):
How long was the route you created?

Task 6: View a Route—You want to check out a specific route in the system. You remember a route you had liked was named "River Run 3" and you want to take a look at the path as well as the summary. Load this specific route into the interface. After answering the questions below, clear the route.

Task 6 question(s):
How long is this route?
What are the titles of the two points of interest?

Task 7: Rating a Route—You have just completed a route and you were overall satisfied with it. You remember the route was named "University Walk". Give the route a rating of 4. Now check to make sure the rating was received.

Task 7 question(s):
How do you know the rating was recorded by the application?

Task 8: Adding a Point of Interest—While strolling along the route named "University Jog" you spy a cool pedestrian bridge near the university campus. Load up this route. Now locate the pedestrian walkway that crosses the river and add a point of interest. Add a title and description of your choice and then confirm the point.

Task 8 question(s):
How can you tell your new point of interest has been added to the route?

Each task had an associated question for the user to answer to force the user to focus their attention on the system's responses and feedback. I estimated a target completion time for each task based upon the amount of steps involved to accomplish the task. Then, each user was timed

for every task and their resulting time was compared to the target time. Each test was followed

by a brief interview consisting of 5 questions that ask the participants about their thoughts of the

application, any difficulties they experienced, and suggested improvements for the application.

There were a total of 9 participants (5 female and 4 male) ranging from a variety of ages

(18 – 50). The participants were recruited from the surrounding area and included both smart

phone owners as well as users with very little experience of these devices. Each test was

facilitated in a quiet area with only the administrator and participant present. Pedestrian Pal was

loaded onto a laptop and users accessed the application through an Android emulator. The

screen-cast software Camtasia was used to capture user mouse movements and inputs in the

application's interface. Analysis of these recorded sessions yielded confirmations of task time

performances and better observations of the user's attempts to solve each task. All the raw data

for these tests are found in Appendix II and the interviews are located in Appendix III.

The Emulator

The Android plug-in for Java's Eclipse development environment has the ability to

emulate any number of mobile devices and Android versions to enable building for numerous

smart phone platforms. This is a boon to a developer that allows for greater testability amongst

many devices. On the other hand, there exist drawbacks for these emulators that can hinder

some development. Plus, administrating user tests via this method may produce some

undesirable results.

One caveat with performing application user testing with an emulator is that the

simulation has many differences from an actual mobile device. First, the emulator's soft

keyboard was suppressed to preserve the screen size and offer a better user experience. When

enabled, the large keyboard appears from the bottom whenever a user clicks on a textbox. The keyboard envelopes half the screen and hides much of the interface. The hard keyboard is situated next to the emulator's screen and persists throughout a simulation. The redundancy of two keyboards (with the soft keyboard unsuppressed) would elicit unneeded feedback from participants. Unfortunately, suppressing a soft keyboard is not an option when deploying the application to a mobile device. Some devices only contain a touch screen for input and the hard keyboard is nonexistent.

Another problem with testing with an emulator is the need for hard keys for the application to function. Relying on the hard keys disrupts the flow of the program and could cause inaccurate results. The "MENU" hard key was utilized often in Pedestrian Pal to bring up the main option menu in the software. A return arrow key was also used to exit out of certain options. This key was unintuitive and only the smart phone owners knew the functionality of this key and could use without instruction.

Finally, performance was a big issue when testing with an emulator. Response times on actual devices are noticeably quicker than the software counterparts. The simulations would begin a test with little delay between the user's action and the interface response. As the participant entered further into the test, the program would exhibit as much as 1 to 2 seconds between key press and response. Developers have noted memory leaks in these emulators which could be the cause of the slow response times.

Android Device

A solution to avoid the problems I experienced with the emulator was to put the application directly on a phone. The application was developed entirely for the emulator, but this does not ensure a robust product when deploying onto a phone. Pedestrian Pal was deployed

to a 3G Motorola Android touch-screen phone. **Figures 16 & 17** pictures the application loaded

on the actual Android device.
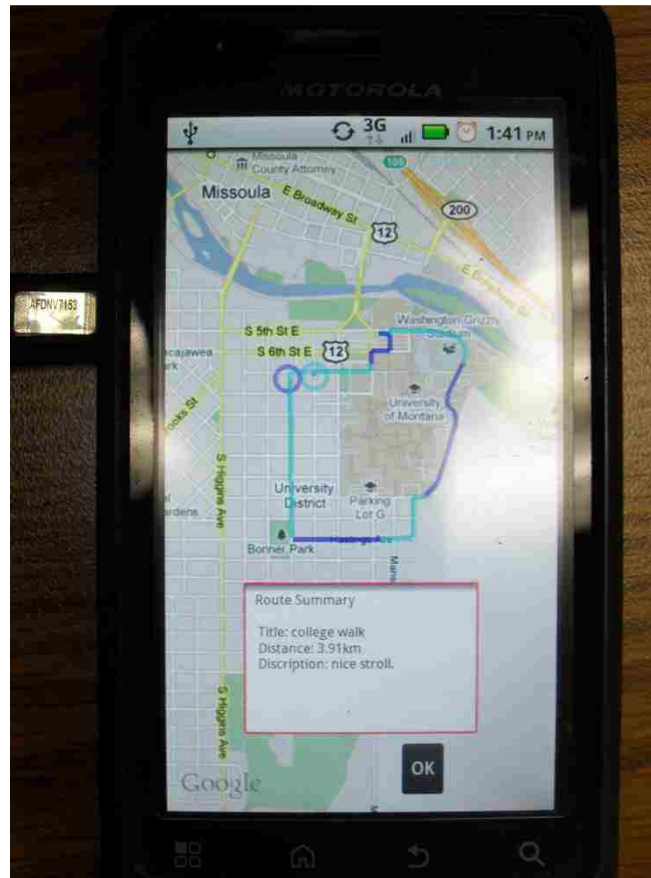
Figure 6: Pedestrian Pal deployed on a Motorola Android device. The user has just created a new route.
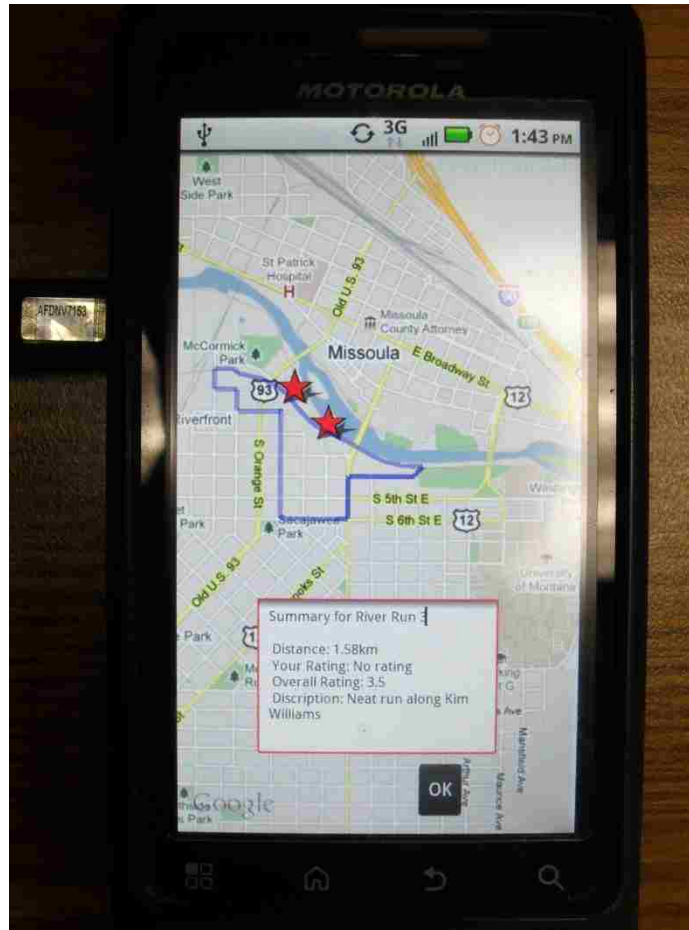
Figure 7: Another image of the application deployed onto a phone. Here the user has loaded a route from the database.

When I first deployed the application on to the phone I tested the menus, buttons, and options to see how well they work with a touch screen. I had no problems with touching correct buttons (the spacing and button size were adequate) and the device was very responsive compared to the emulator. I felt using Pedestrian Pal on the device was much more intuitive. I could move around the map with ease—using quick swipes to scroll around town. I found that my fingertips were a little too large to select points on a map when the map was too far zoomed out. Happily, the actual device responds to the pinching motions that move the view closer to the streets. Overall, the deployed application exceeded my performance expectations and confirmed the use of best practices when designing layouts.

Brief testing was performed on the Android device and some immediate problems emerged within the first two tasks. When the phone is turned to the side the tilt sensor fires the "onCreate" method for the application which essentially restarted the program. This forces the user to utilize the soft keyboard for text input. The soft keyboard had been suppressed for emulator use in some sections of the application. This prevented the user tester from completing two of the eight tasks. More about the Motorola Android user tests are found in the **Results (Motorola)** section.

Results (Emulator)

The results of these tests confirmed many of the design choices and high-lighted the areas of the application that need improvement. A summary of the user tests is shown in **Table 1**. The table lists all 8 tasks along with the expected completion time. I calculated the expected time by initially recording the time for performing each task. Then I modified the time based on the number of steps involved in the task. I have included some of the more interesting comments in this summary table. For some tasks, the comment was a representative of the overall appreciation observed for the application's functionality. In other tasks, the user's comment exhibits the best summary for improvement in the particular area of the program.

Task	Expected Time (mm:ss)	Actual Time (Average)	Stand-out User Comment
1. Creating a New User Account	0:45	0:51	I guess I wouldn't have a tab button on a phone. I'm surprised the password isn't dots/stars and plus, no confirm password textbox.
2. Read Instructions	3:00	2:32	Ah, nice (referring to finding guide) "DONE" seems to show up everywhere
3. Change Your Level of Recommendation	0:45	1:09	Now, I'm in "routes"; but maybe that's more of a setting.
4. Find Paths for Jogging	3:00	3:18	Let's go to the route recommender and see what that does. So I clicked near the Clark Fork; let's fill in these little boxes.
5. Build Your Own Route	2:30	2:23	I took a route to the griz game and it was perfect. I didn't get lost.
6. View a Route	2:15	2:03	Very intuitive, I tried this before, but that works.
7. Rating a Route	1:30	1:05	OK, that is advertised to me, but I didn't trust the software, so let's verify that.
8. Adding a Point of Interest	3:00	2:29	Oh, I see, it puts it into add point of interest mode. I couldn't pan, I'll go back.

Table 1: Summary table for user testing.

The simplest task involved creating a new account. I estimated a target time of 45 seconds to complete this task. All users completed the login with little or no difficulty and all but one user finished within 15 seconds of the target time (with an average of 51 seconds). These results show that this task offers no difficulties and imply that this process was designed correctly.

Task 4, a much more complicated task, produced more interesting results that revealed a flaw in my system in conjunction with an erroneous task wording. This task asked users to use the system to recommend 2 different routes: one next to the Clark Fork River and one in the

University area. The first part of the task for the route recommendation specified a target

distance and rating. Users were able to enter this information and accomplish this part with little

or no trouble. The second part stated: "You don't mind the distance or rating this time around,"

in reference to the desired input from the user. While performing this part, 5 of the 9 participants

left each field blank and the system told them that the fields required input. The testers found it

odd that the task states that the inputs do not matter, but then the program issues a warning.

Using this feedback I would change the system to accept blank fields and the only search

parameter would be the user-clicked geo point.

One flaw in the system was revealed by three testers during task 8. This task requires

users to add a point of interest to a specific route. To complete the task, the user must load the

target route, choose the command "Add a Point of Interest", and then click on the map to specify

where to place the item. After selecting the option to add the point, two users attempted to pan

the map to find the correct location. This resulted in placing the point at the first place they

touched the map. One individual stated verbally that he "*couldn't pan*" and that he would go

back and retry the task. Obviously, some method is needed to allow users to move around before

they place the point.

Overall, I observed that users began to complete tasks quicker in comparison to the

expected time as the user tests continued. After task 4 (the first more complicated task) the

users' average time of completion was lower for all expected times. Nearly all of the tasks were

completed by all users aside from task 4 for one user and tasks 1 and 4 for the android phone

user. Notice in **Table 2** that users had little difficulty working through the entire set of tasks.

Two of the unfinished tasks occurred during testing on the actual device where the user knew

how to complete the task, but the lack of soft keyboard prevented him from doing so.

User	1	2	3	4	5	6	7	8	9	10*
Task	Question	Completed?								
1. Creating a New User Account	a. How do you know an account was made?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
2. Read Instructions	a. What two objects can you rate in this application?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	b. What does the application automatically do when you click "Done" while the route creation tool is active?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
3. Change Your Level of Recommendation	a. How do you know the level has been changed?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
4. Find Paths for Jogging	a. What is the title of the first route you found?	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No
	b. What is the title of the second route you found?	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No
5. Build Your Own Route	a. How long was the route you created?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
6. View a Route	a. How long is this route?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	b. What are the titles of the two points of interest?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
7. Rating a Route	a. How do you know the rating was recorded by the application?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
8. Adding a Point of Interest	a. How can you tell your new point of interest has been added to the route?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 2: Task/question completion table. This table shows whether a user completed a task and if they have answered the task's question(s) correctly. (*) Denotes the user was tested on an Android device rather than the emulator.

The interviews revealed some interesting feedback that confirmed many design choices as well as revealed various ideas for improvement of the application. All 9 users felt that the system was useful according to its purpose. One comment was: "*it's cool that you can figure out a new route based on setting.*" and another individual stated, "*…good tool for people who are new to the area. It accumulates user content and draws on a collective knowledge.*"

The menu system proved to be well designed judging from the responses to the third interview question. This question asked users about their thoughts on the system's navigation in terms of menu design and content. Comments ranged from "*Pretty Easy,*" to "*Incredibly simple*" and *"Very self explanatory.*" Users seemed to appreciate the limited choices per menu level.

Question 4 of the interview elicited excellent constructive criticism and revealed areas that needed changes in the application. One user found the menu option labeled "Recommend Level" to be a little ambiguous. Changing this name could be a quick fix that would clear up any confusion brought on by the current label. Another tester found that the feedback "Summary Closed" that appears when a user closes the summary box was a little redundant. This feedback seems correct and could be a quick fix as well. Yet another user found that in some cases there was a lack of feedback. He felt that adding system responses to most of the user's inputs could provide a more robust system.

Table 3 in the *Future Work* section contains some of the interview comments as well as proposed solutions to discovered problems.

Results (Motorola Phone)

While the emulator was used for the bulk of the testing, I was able to deploy the application to an Android device and gather valuable data through testing on the phone.

As mentioned previously, the tilt sensor caused some problems when the application requested keyboard input from the user. The user attempted to tilt the phone during Task 1 and use the hard keyboard. This action immediately reset the application and the task became impossible to complete. Task 4 caused the same problems.

The Motorola Android phone exposed some problems with the application, but it also reaffirmed the designed choices in regards to button spacing and size. During the interview, the tester noted that the "*…buttons were the right size—no problems with fat fingers*." Previously, all the tests had been using the emulator. During those tests the users could precisely click buttons using the fine point of a mouse. During this test, the user needed to use a fingertip to navigate through the interface. By stating that he had no problems begins to show that few if any problems exist in regards to the buttons sizes and spacing.

When drawing and loading routes, the emulator would often pause and take much time building each section to display on the interface. One of my assumptions was that this delay would disappear once the application was running on a phone. This assumption proved wrong as the user completed tasks involving route drawing. Routes would pause between sections and exhibit the same behavior as shown on the emulator. Discovering this problem allows me to debug properly to fix this error.

Future Work

The application as it exists is not finished and could use some more interesting features. With more time, this application could be filled with many options to offer users a more satisfying and complete experience. See **Table 3** for a detailed list of user-suggested changes, options, and ideas.

To offer a better recommendation system I would need to add more input variables into the system in regards to routes. If a user could search for a route based on several factors then she would be able to have a recommended path representative of her needs. For example, a user may want elevations gain/loss information, local wildlife and flora nearby, or safety ratings. The user could rate all of these factors and these options would build a more robust route search engine.

In question 5 of the interview, three of the ten respondents mentioned some sort of functionality involving the phone's built-in global positioning system. Using these ideas, the program could become more of a social networking application that brings friends together for walks, hikes, and more. In addition, gps could be used for showing a user's current location and the system could respond by recommending nearby routes.

Several individuals have said that adding pictures to the points of interest would improve the recommendations. People could view an area's sites without having to travel there. This could potentially influence their decision of using that route or moving on to another.

Adding elevation data to the system could offer more options as well. The initial survey showed that several individuals were interested in a system that could show this sort of

information. This application could issue difficulty ratings based upon the elevation gained or

lost during the route.

Comment	Proposed Fix
Recommend level: A little ambiguous.	Change the title of this option. Several users were confused about where to find this option. "Recommend" appears in the name and a higher-level menu option contains the word as well.
I didn't know to look for stars. (Referring to points of interest)	I should add a mention of the poi stars in the help guide. Perhaps put a hint in the software as well: add a poi count in the route summary.
Some lack of feedback. **Some feedback redundant.**	Review all options and check for missing feedback.
Maybe [add] pictures, I'm a visual person. **Pictures: associated with poi so a user can say, "oh, I saw this in the application, it's up ahead"**	Add a picture attribute for the poi table in the database. Add clickable images in the poi's in the application to view these images.
Indication of where I am in reference to the map and perhaps to a route as well. **GPS: where you are on the route.** **GPS: Ability to meet up with people. Broadcast to tell others what you are doing and where you are--social aspect part of app**	I would need to investigate how to incorporate GPS. Then add a visual into the application to update positions on the map along with user editable updates.
Edit mode: some visual cue so you know what to expect so you know you aren't in pan/zoom mode. **Add in something to tell you what mode it is on.**	I could have a slim, colored bar run across the bottom or top with text inside that indicates the mode: map select, zoom, pan, etc.
Have it able to reload the summary again.	Add an option in the menu (or have hot key) to pop up a loaded route's summary at any time.
Always hidden menu; have something to pop it up like an arrow at the bottom of the screen	Currently, the menu is accessed via hard button. Using touch screen drag technology I can have a small arrow at the bottom of the screen a user can drag up to show the menu.

Comment	Proposed Fix
Keyword searches for routes	Build a route search system. Once the database gets full of routes it may be difficult to find routes without one.
User creation screen: 2 boxes for password	Add another password box so user is sure that the password was entered correctly.
Back swoop arrow to get back wasn't intuitive.	Stop relying on this button. Add an option in the application that has same capability.
Globalize it: for other locations.	This was the intention. This would not be difficult since the application can accept any gps coordinates in the world. It just does not have any data outside of Missoula currently.
Animals to view for the seasons. Berry picking for seasons.	This might be too much for the application's intention. Routes are meant for urban areas mostly, so it may not make sense to add this information. In addition, this might involve user entry for this information.
That might be nice if I can go from viewing a route to rating a route explicitly.	Add some more flexibility into the system. Allow users to use options in any order or from any state.
How do I escape from this poi? Oh, return hard key. **Go away now <referring to poi>**	Add an exit button on the point of interest screen.

Table 3: Future work. This table contains user comments collected from user tests and the post-test interviews. There is a corresponding solution to the users' comments.

Conclusion

At the beginning of this project I set out to build an interesting and creative system on the Android mobile phone. This application gives path recommendations based upon not only the destinations, but also the route itself. Users can utilize this system to explore new areas of cities and offer feedback to build routes to their needs.

The system is functional as well as user-friendly. The simplicity of the design prevents any confusion. Straightforward feedback ensures users that the application is working to their desires. Layouts and menu are fluid and easy-to-use. The system as a whole works according to the initial designs and shows the strength of proper development.

References

Baltrunas, L. (2008). Exploiting contextual information in recommender systems. In the
Proceedings of the 2008 ACM conference on Recommender systems (RecSys '08). ACM,
New York, NY, USA, 295-298.

Ge, Y., Xiong,H., Tuzhilin, A., Xiao, K., Gruteser, M., & Pazzani, M. (2010). An energy-
efficient mobile recommender system. In the *Proceedings of the 16th ACM SIGKDD
international conference on Knowledge discovery and data mining* (KDD '10). ACM,
New York, NY, USA, 899-908.

Gevin. A., Sefelin, R., Tscheligi, M. (2006). Depth and breadth away from the desktop – the
optimal information hierarchy for mobile use. In *Proceedings of the 8th conference on
Human-computer interaction with mobile devices and services* (MobileHCI '06). ACM,
New York, NY, USA, 157-164.

Kim, J., Kim, H., & Ryu, J. (2009). TripTip: a trip planning service with tag-based
recommendation. In the *Proceedings of the 27th international conference extended
abstracts on Human factors in computing systems* (CHI '09). ACM, New York, NY,
USA, 3467-3472.

Parhi, P., Karlson, A., & Bederson, B. (2006). Target size study for one-handed thumb use on
small touchscreen devices. In *Proceedings of the 8th conference on Human-computer*

interaction with mobile devices and services (MobileHCI '06). ACM, New York, NY, USA, 203-210.

Perry, K., & Hourcade, J. (2008). Evaluating one handed thumb tapping on mobile touchscreen devices. In *Proceedings of graphics interface 2008* (GI '08). Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 57-64

Tintarev, N., Flores, A., & Amatriain, X. (2010). Off the beaten track: a mobile field study exploring the long tail of tourist recommendations. In the *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services* (MobileHCI '10). ACM, New York, NY, USA, 209-218.

Appendix I

Initial Survey Questions 1 – 4 (Raw Data)

1. How often do you go for a jog or walk? Rarely (less than once a week) Occasionally (Once a week) Frequently (More than once a week)	2. Do you take in consideration the setting and location when you jog or walk?	2a. If yes, what do you look for when you walk or jog?	3. Do you consider elevation gain when you walk or jog?	4. What local trails do you use?
Frequently	No	--	Yes	Kim Williams, Mt. Sentinel
Occasionally	No	--	No	I really just run in the university area neighborhood district because it is close to where I live.
Occasionally	Yes	Riverside or a place w/lots of vegetation	No	Around Bonner Park, Kim Williams trail
Rarely	Yes	Scenery, how busy the area is	Yes	--
Rarely	Yes	Out of town	Yes	Riverfront trail
Rarely	Yes	Good scenery, clean air, not many people	Yes	Mt. Sentinel
Frequently	Yes	safety	Yes	M-trail, L-trail, river trail
Rarely	Yes	A safe place that is fit for the type of workout that I want	Yes	The river walk, The "M"
Frequently	Yes	River and mountains and other people	Yes	Kim Williams trail is the best! Rattlesnake
Frequently	Yes	Peace & quiet	Yes	--
Frequently	Yes	Clean sidewalks and no garbage	No	Riverbowl
Rarely	Yes	Safe, calm environment	Yes	None (on a regular basis)

1. How often do you go for a jog or walk? Rarely (less than once a week) Occasionally (Once a week) Frequently (More than once a week)	2. Do you take in consideration the setting and location when you jog or walk?	2a. If yes, what do you look for when you walk or jog?	3. Do you consider elevation gain when you walk or jog?	4. What local trails do you use?
Frequently	Yes	Sunny/warm day around campus	Yes	Occasionally the "M", usually the streets around campus
Frequently	Yes	Good atmosphere, population, crowded, traffic	No	Kim Williams
Rarely	Yes	Somewhere cool and with shade and few people	Yes	Kim Williams, various mountain trails
Occasionally	Yes	Nice days, neighborhoods	No	University boundary
Rarely	No	--	No	--
Occasionally	Yes	Nice, walk areas, public safety	No	Bonner, Karas, river walk, dog park
Occasionally	Yes	Lack of people	Yes	Kim Williams, bike trails
Occasionally	Yes	flowers, trees, birds	Yes	Everywhere
Rarely	Yes	pleasing scenery for a run, flat ground/hilly	Yes	Kim Williams
Frequently	No	--	No	All over
Frequently	Yes	Something that captures my interest when I go	No	Rattlesnake, Pattee Canyon, Blue Mountain, Kim Williams

Initial Survey Questions 5 – 8 (Raw Data)

5. Are there any neighborhoods that you use when you walk or jog? Please list any.	6. Are there any areas in Missoula that you avoid when you walk or jog? Please list any.	7. Would you find a computer application that recommended foot travel routes useful?	8. What types of options would you want from this kind of application?
No	No	--	--
See above	Not particularly. I avoid South and Higgins	Yes	I think for someone who ran frequently this app could be very useful. I don't have much time so it wouldn't do much for me
University area, downtown	No	Yes	Estimate of how many calories burnt by walking said route. Elevation changes & distances
University area	--	Yes	Elevation, average foot traffic
Canyon Creek area	Downtown	Yes	Distance to get to trail, length of trail, approx walking/jogging time to complete
University area	No	Yes	length of walk, popularity of route (quiet or not), natural or urban setting, weather, sample visuals, types of plants (for allergies), difficulty (steep, rough terrain, etc.)
Downtown & university area	Parts of river trail in evening--safety reasons	Yes	Length of trails, elevation changes & safety ratings
My neighborhood (Scott Street Bridge)	Creepy places of the northside	Yes	Safety, scenic, elevation
University area	none	Yes	Distance of travel, mph traveled (if you phone has gps)
U area, downtown, river run	The ghetto	No	Don't care, I hate smart phones
Downtown & university area	--	Yes	Direction & difficulty of running routes

5. Are there any neighborhoods that you use when you walk or jog? Please list any.	6. Are there any areas in Missoula that you avoid when you walk or jog? Please list any.	7. Would you find a computer application that recommended foot travel routes useful?	8. What types of options would you want from this kind of application?
South hills	Downtown	No	--
University area	High-population areas	No	I wouldn't because I don't want people bogarting my trails
Downtown, University	Reserve, South Brooks	Yes	Route options, quickest, scenic, etc.
--	Crowded places	Yes	Distance, rest areas/scenic areas along the route, amount of traffic
University area	Railroads	No	--
--	none	Yes	Lengths of routes, time it should be used, if ran: low, medium, high intensity work out
University, home area, parks, out of direct traffic	very quiet, dark places, unsafe areas	Yes	Distance, grade, frequented by whom (how often), animals
U district is my favorite	Northside	No	--
University area	Rattlesnake trail	Yes	Miles in where I want to go from one place to another
Linda Vista	no	No (<maybe for> travelling to another city)	Efficiency vs. scenic route options
Blue mountain, UM, Rattlesnake	Busy streets	Yes	Routes with distances labled
University area mainly, when in town South hills a bit	Downtown, busy streets	Yes	ETAs, Rates and distances traveled

Initial Survey: Analysis (Questions 1 – 4)

Question 1		Question 2		Question 2a		Question 3		Question 4	
Category	Count	Category	Count	Category	Count	Category	Count	Category	Count
Rarely	8	Yes	19	No Answer	4	Yes	14	Kim Williams Trail	13
Occasionally	6	No	4	Riverside	3	No	9	Mount Sentinel	5
Frequently	9			Scenery/ Vegetation	5			University Area	3
				Traffic	8			Bonner Park	2
				Safety	4			No Answer	3
				Weather	2			Lolo Mountain	1
				Fitness	2			Rattlesnake Area	2
				Other	3			No Preference	3
								Karas Park	1
								Pattee Canyon	1
								Blue Mountain	1

Initial Survey: Analysis (Questions 5 – 8)

Question 5		Question 6		Question 7		Question 8	
Category	Count	Category	Count	Category	Count	Category	Count
No/No Answer	3	No/No Answer	8	Yes	16	No Answer	3
University	16	South St.	1	No/No Answer	6	Other	4
Downtown	5	Higgins Ave.	1	Maybe	1	Calories Burned	1
Canyon Creek	1	Downtown	3			Elevation	5
Scott Street Bridge	1	Kim Williams (At night)	1			Distance	9
Kim Williams	1	Northside	5			Traffic	3
South Hills	2	Other	4			Completion Time	3
Linda Vista	1	Reserve St.	1			Weather	1
Blue Mountain	1	Brooks Ave.	1			Photos	1
Rattlesnake	1	Rattlesnake	1			Flora & Fauna	4
						Difficulty	3
						Safety	2
						Direction	1
						Rate of Travel	2

Appendix II

User Tests (Raw Data)

Task 1. Creating a New User Account—You have just downloaded the Pedestrian Pal application to your mobile phone and before you can start using it you need to create a new account. Please do so now.				
User	Task 1 question(s): How do you know an account was made?	Correct steps to complete task: __Click "New User" button __Enter a user name __Enter a password __Enter a first name __Enter a last name __Click "Create User"	User's comments:	Time to complete (Expected: 0:45):
1	It told me my name	USER FOLLOWED EXACT STEPS	There is no tab key	0:50
2	Because after it was made it brought me to a map and a notification came up saying it was made.	USER FOLLOWED EXACT STEPS	*None*	0:35
3	It let me move on.	USER FOLLOWED EXACT STEPS	*None*	0:37

4	Because it said it.	__Click "Login" __Click "CANCEL" **<Performed the rest correctly>**	Do I just keep going? Type in a user name? (referring to the textbox after pressing "Login")	1:50
5	Because it said my name.	USER FOLLOWED EXACT STEPS	*None*	0:50
6	After entering my desired credentials, I was informed that my account was created successfully.	USER FOLLOWED EXACT STEPS	I would assume I use the new user option here. I'm using hard keys to enter credentials.	0:40
7	Pop up saying it was done	USER FOLLOWED EXACT STEPS	Nice	0:33
8	Because it explicitly said "Account created".	USER FOLLOWED EXACT STEPS	I guess I wouldn't have a tab button on a phone. I'm surprised the password isn't dots/stars and plus, no confirm password textbox.	0:58
9	A window popped up to tell me it was created	USER FOLLOWED EXACT STEPS	OK, so it shows it to me.	0:44

Task 2: Read Instructions—You want to learn a little about your new application. Read through the guide to get some basic knowledge of Pedestrian Pal. Exit the guide when you are finished.

User	Task 2 question(s): What two objects can you rate in this application? What does the application automatically do when you click "Done" while the route creation tool is active?	Correct steps to complete task: __Click MENU __Click "User Settings" __Click "Help Guide" __Click "Next" to move to the next page __Record "Routes" and "Sections" to answer first task question __Click "Next" to move to the next page __Record "Automatically creates a circuit" to answer task question 2 __Click "Done" to return to title screen	User's comments:	Time to complete (Expected 3:00):
1	1) segment or full 2) Connects the waypoints to the start to make a circuit	USER FOLLOWED EXACT STEPS	I don't know what a guide is. I don't know where to find it. <Reads guide out loud>	2:30
2	1) Rate routes and certain parts of routes. 2) Automatically connects your first& last waypoints	USER FOLLOWED EXACT STEPS	So, In settings? (trying to find the guide) What if I don't remember? (User couldn't remember what the guide had said)	3:16
3	1) Ability to rate routes or even sections. 2) Connect your first and last waypoints	__Click MENU __Click "User Settings" __Click "Help Guide" **<Performed all the step correctly afterwards>**	*None*	3:40

4	1) Area around your neighborhood 2) Different routes.	__Click MENU __Click "Routes" __Click "View a Route" **<Performed all the step correctly afterwards>**	There isn't anything. (referring to a menu; Administrator pointed out the hard "MENU" button) Can I just say what I think it does?	1:30
5	1) Rate routes. 2) Creating my perfect path.	__Click MENU __Click "Routes" Exited Application. Clicked many random buttons. **<Performed all correct steps after relogging in and starting over>**	I thought the instructions were the choices. (referring to the menu choices)	4:40
6	1: Routes and sections on routes 2) Link your last waypoints	USER FOLLOWED EXACT STEPS	I don't see a guide option available to me. <finds MENU hard key> Let's see if settings has anything for me. Ah, excellent. <reads guide aloud>	1:30
7	1) Routes and sections of routes 2) Connect first and last waypoint	USER FOLLOWED EXACT STEPS	Ah, nice (referring to finding guide) "DONE" seems to show up everywhere	1:10
8	1) Routes and parts of routes 2) Links first and last points to create a circuit.	Went through steps twice to recheck guide.	Probably doesn't need it on the main screen (referring to guide): this would preserve real estate.	1:58
9	1) Routes and parts of routes 2) connect my first and last waypoints	__Click return arrow hard button **<Performed all correct steps after relogging in and starting over>**	I don't know how--oh there it is.	2:30

	Task 3: Change Your Level of Recommendation—You decide that you dislike having other users' routes showing up in the list of available routes. Change the level of recommendation to have only your routes show up.			
User	Task 3 question(s): How do you know the level has been changed?	Correct steps to complete task: __Click hard button "MENU" __Select "User Settings" __Select "Recommend Level" __Select "User"	User's comments:	Time to complete (Expected: 0:45):
1	prompt	__Click hard button "MENU" __Select "Routes" __Select "Route Recommender" **__Click hard button "MENU"** **__Select "User Settings"** **__Select "Recommend Level"** **__Select "User"**	One thing that confused me: is "global" everything?	0:58
2	A notification comes up saying I set it to user	USER FOLLOWED EXACT STEPS	You go to this thingy? Hmmm, that was easy.	0:45
3	It told you	__Click hard button "MENU" __Select "Routes" **<Performed all the step correctly afterwards>**	So, maybe I choose "Route Recommender"?	1:30

4	Because it said it was set to user	__Click hard button "MENU" __Select "Routes" __Select "View a Route" **<Performed all the step correctly afterwards>**	I don't know how to do this. I don't know how I did it. (Re-performed task and saw the notification)	2:00
5	Because it told me it was my user recommendations	__Click hard button "MENU" __Select "Routes" __Click return arrow hard button **<Performed all the step correctly afterwards>**	Look at that; I even know how to get back. (referring to return arrow)	3:02
6	I am informed that the level has been set by the software	__Click "MENU" __Select "Routes" __Click "MENU" x2 **<Performed all the step correctly afterwards>**	Now, I'm in routes; but maybe that's more of a setting.	0:45
7	Pop up saying it changed	USER FOLLOWED EXACT STEPS	That was easy	0:15
8	Feedback "Route level has been changed"	USER FOLLOWED EXACT STEPS	I'm going to guess that it's "user". (referring to the options in the "Recommendation Level")	0:25
9	Little window tells me	__Click hard button "MENU" __Select "Routes" __Select "Route Recommender" **<Performed all the step correctly afterwards>**	No? But I'll find it...there.	0:42

Task 4: Find Paths for Jogging—You are in the Missoula area visiting a friend and you would like a nice place to jog by the Clarkfork River. You want to a nice 2-kilometer run, but you aren't sure of a route. In addition, you would like to use a route that has a rating of at least 3. Use the system to find a path with these specifications. After finding the first path you also wanted to find a route near the university. You don't mind the distance or rating this time around. Use the system to find this route.

User	Task 4 question(s):	Correct steps to complete task:	User's comments:	Time to complete (Expected: 3:00):
	What is the title of the first route you found? What is the title of the second route you found?	__Click hard button "Menu" __Click "Routes" button __Select "Route Recommender" __Click near the Clarkfork River __Enter "2" into the top textbox __Enter "3" into the bottom text box __Press "SEARCH" __Click hard button "Menu" __Click "Routes" button __Select "Route Recommender" __Click near the University __Press "SEARCH"		
1	1) River Run 3 2) River Run 3	__Click hard button "Menu" __Click "Routes" button __Click "Create a Route" __Click "CANCEL" **<All other steps correct>**	If I don't mind the distance, what do I put? I'm curious what these stars are. I didn't notice that showed up last time (refering to feedback notification)	5:02
2	1) River Run 3 2) University Jog	USER FOLLOWED EXACT STEPS	Is it "River Run 3"? (When searching for route title)	2:30

3	1) *No Answer* 2) *No Answer*	User began by trying to "View a Route" but couldn't find a route near the river. After some time user gave up due to frustration.	Do you "View a Route"? I'm so bad at this. I'm confused.	3:20
4	1) River Run 3 2) University Jog	USER FOLLOWED EXACT STEPS	I have to enter "2" do I just push "2"? But I didn't find the other route; I don't understand why it didn't come up. Do I have to go back? (wondering if she needs to re-perform same steps for 2nd route)	4:02
5	1) River Run 3 2) University Jog	USER FOLLOWED EXACT STEPS	Ah, but it's the same. (referring to clicking on the first poi again after trying to click the second)	2:50
6	1) River Run 3 2) University Jog	USER FOLLOWED EXACT STEPS	Let's go to the route recommender and see what that does. So I clicked near the Clarkfork; let's fill in these little boxes. I am assuming I can leave these blank (Referring to searching for the second route) ah, OK, no.	3:10

7	1) River Run 3 2) University Jog	__Clicked poi x2 __Clicked return arrow x3 **<performed correct steps after relogging>**	I just wanted to bring up the summary again. OK, that makes sense. I want a way to bring up the summary without going through the steps again. I want something about leaving the fields blank in the help guide.	3:28
8	1) River Run 3 2) University Jog	__Click hard button "Menu" __Click "Routes" button __Select "Route Recommender" __Click map to pan __Click "CANCEL" **<performed correct steps after restarting task>**	Route Recommender: so, I tried to click on the map to pan. OK, CANCEL. Now I'm going to pan. <moves map> Hmmm, summary closed: I closed it AND it told me. OK, this is somewhat of a specfic domain knowledge. <user said this after entering *'s in the field to indicate that the fields don't matter for the search>	2:50
9	1) River Run 3 2) University Jog	**<performed correct steps except repeating after clicking twice in one spot>**	What should I do? <after clicking near first area>	2:28

Task 5: Build Your Own Route—You have decided to build your own route. You want to take a nice stroll through the university area and next to the river. Make a path in these areas and name the route as "River Walk". Add a short description about the route.				
User	Task 5 question(s): How long was the route you created?	Correct steps to complete task: __Click hard button "Menu" __Click "Routes" button __Select "Create a Route" __Click on map to select several waypoints __(Optional) Move the map by clicking "Zoom" to get to university and river area __Click "Done" __Enter a title "River Walk" __Enter a short description __Press "Done" __Record route distance	User's comments:	Time to complete (Expected: 2:30):
1	1.97km	USER FOLLOWED EXACT STEPS	Can you drag it? (referring to map) It's not doing anything. (clicking on map)	2:40
2	1.33km	USER FOLLOWED EXACT STEPS	Hmmm, can I move this around? (referring to map)	2:20
3	2.04km	USER FOLLOWED EXACT STEPS	Do I just come up with a title?	1:20
4	1.58km	**<Performed first steps correctly>** __Click "CANCEL" **<Re-performed all steps>**	OK, starting over; oh, I see it now.	3:30
5	1.97km	USER FOLLOWED EXACT STEPS	I took a route to the griz game and it was perfect. I didn't get lost. Ah, I already did this. (referring to when user had explored the application in an earlier task)	0:30

6	3.91km; I went for a nice long route to explore the route creation tool.	USER FOLLOWED EXACT STEPS	Oh, neat, it appears to go through the streets. I can dig, it's not what I was thinking, but it's using roads and such. (referring to the route drawing differently than expected)	2:30
7	10.06km	USER FOLLOWED EXACT STEPS	Needs to be some sort of feedback when you click the map.	2:32
8	2.9km	**<Performed correct steps, then:>** __Click "MENU" __Click "View a Route" __Click "River Walk"	OK, so, if it says "Click…" to start, but I see a button that says "Move & Zoom". No point shown: it seemed to record my starting point but no feedback.	2:39
9	3.84km	**<Performed all steps, but also clicked "undo" twice when creating route>**	Oh, but what is happening? I can't undo my first click.	3:22

Task 6: View a Route—You want to check out a specific route in the system. You remember a route you had liked was named "River Run 3" and you want to take a look at the path as well as the summary. Load this specific route into the interface. After answering the questions below, clear the route.

User	Task 6 question(s): How long is this route? What are the titles of the two points of interest?	Correct steps to complete task: __Click "MENU" hard button __Press "Routes" __Select "View a Route" __Select "Rive Run 3" __Answer "1.58km" __Click on a star (point of interest) __Answer "Boone & Crockett Club" __Click "return" hard button __Click on a star (point of interest) __Answer "Park Entrance" __Click "return" hard button __Click "MENU" hard button __Press "Routes" __Select "Clear Routes"	User's comments:	Time to complete (Expected: 2:00):

1	1) 1.58km 2) B & CC, Park entrance	__Click the return arrow **<Performed the rest correct after logging back in>**	*None*	1:42
2	1) 1.58km 2) B & CC, Park entrance	USER FOLLOWED EXACT STEPS	That thing? (referring to the stars)	1:35
3	1) 1.58km 2) B & CC, Park entrance	**<Performed all steps except for last 2>**	Wait, what is that "Points of Interest" do I click that? Do I click on the stars or something?	2:30
4	1) 1.58km 2) B & CC, Park entrance	**<Performed all steps except for last 2>**	Well, <reads route summary> is that one. <After more time administrator questions if user sees anything different along the route> . Oh! There you go.	3:04
5	1) 1.58km 2) B & CC, Park entrance	USER FOLLOWED EXACT STEPS	I wonder if I could just go to "View a Route"; there it is! River Run 3. You know, I actually thought these stars were where I clicked earlier.	2:20
6	1) 1.58km 2) B & CC, Park entrance	**<Performed all steps except for last 2>**	Very intuitive, I tried this before, but that works. When I see a magnifying glass <referring to hard button> I think zoom in.	2:30
7	1) 1.58km 2) B & CC, Park entrance	USER FOLLOWED EXACT STEPS	OK, simple enough.	1:20
8	1) 1.58km 2) B & CC, Park entrance	USER FOLLOWED EXACT STEPS	I'm going to tap on a point of interest. How do I escape from this pot? Oh, return hard key.	1:32
9	1) 1.58km 2) B & CC, Park entrance	USER FOLLOWED EXACT STEPS	That was easy. Go away now <referring to poi>	1:52

Task 7: Rating a Route—You have just completed a route and you were overall satisfied with it. You remember the route was named "University Walk". Give the route a rating of 4. Now check to make sure the rating was received.				
User	Task 7 question(s): How do you know the rating was recorded by the application?	Correct steps to complete task: __Click "MENU" hard button __Press "Routes" __Select "Rate a Route" __Select "University Walk" from the list of available routes __Select the "4" rating __Click "MENU" hard button __Press "Routes" __Select "View a Route" __Select "University Walk" __Answer "My rating says 4"	User's comments:	Time to complete (Expected: 1:00):

1	prompted	**__Click "MENU" hard button** **__Press "Routes"** **__Select "Rate a Route"** **__Select "University Walk"** **from the list of available routes** **__Select the "4" rating** Accepted the prompt message as application correctly recording rating; stopped there.	*None*	0:40
2	A notification comes up and says I rated it a "4".	**__Click "MENU" hard button** **__Press "Routes"** **__Select "Rate a Route"** **__Select "University Walk"** **from the list of available routes** **__Select the "4" rating** Accepted the prompt message as application correctly recording rating; stopped there.	*None*	0:35
3	It told you it was rated	User had to go through steps twice. She didn't see the feedback the first time	So, it told you?	1:00
4	Because it said.	**__Click "MENU" hard button** **__Press "Routes"** __Vlick "View a Route" **<Performed the rest correctly after starting over>**	*None*	0:56
5	It showed my rating on the screen	USER FOLLOWED EXACT STEPS	There we go, rate a route	2:12

6	The software informed me of the rating and I manually verified the entry by recalling the route & viewing summary	USER FOLLOWED EXACT STEPS	OK, that is advertised to me, but I didn't trust the software, so let's verify that.	1:05
7	Pop up and listed in summary for route.	USER FOLLOWED EXACT STEPS	Does it show the rating after selecting it? Yes, it did.	0:35
8	Says: "Your rating--4"	USER FOLLOWED EXACT STEPS	I wonder if I can rate it, even if it isn't loaded? That might be nice if I can go from viewing a route to rating a route explicitly.	1:58
9	A window pops up and if you return to the route the rating shows up on the summary	USER FOLLOWED EXACT STEPS	*None*	0:45

Task 8: Adding a Point of Interest—While strolling along the route named "University Jog" you spy a cool pedestrian bridge near the university campus. Load up this route. Now locate the pedestrian walkway that crosses the river and add a point of interest. Add a title and description of your choice and then confirm the point.

User	Task 8 question(s): How can you tell your new point of interest has been added to the route?	Correct steps to complete task: __Click "MENU" hard button __Press "Routes" __Select "View a Route" __Select "Downtown Jog" __Click "MENU" hard button __Press "Routes" __Select "Add a Point of Interest" __Locates and clicks on the pedestrian bridge __Enters information for title and description __Click "Done"	User's comments:	Time to complete (Expected: 2:30):
1	Star in the spot	USER FOLLOWED EXACT STEPS	Does it automatically clear it? (referring to old route and poi's) I don't know how to do this.	2:44
2	Because a star was added to the pedestrian bridge	USER FOLLOWED EXACT STEPS	Where is the bridge? (Administrator helps by pointing on the map)	2:40

3	Indicated with a star	Skipped first part __Click "MENU" hard button __Press "Routes" __Select "Add a Point of Interest" __Locates and clicks on the pedestrian bridge __Enters title and description __Click "Done"	So, wait, how do you tell? It shows a star?	1:30
4	Because it said it was created	**__Click "MENU" hard button** **__Press "Routes"** __Click "Rate a Route> **<Performed the rest correctly after starting over>**	I don't see the pedestrian walkway. <Administrator gives help>	2:50
5	A star was added to my route	USER FOLLOWED EXACT STEPS	Oh, so my star popped up.	2:02
6	Visually, the point of interest has been added to the map. Also, I testing clearing the route and reloading it; my added point of interest was exactly where I left it!	__Click "MENU" hard button __Press "Routes" __Select "Add a Point of Interest" Click "MENU" x2 **<Performed the rest correctly after starting over>**	No, let's try something else. Close the summary. OK, so now poi is cancelled. I'm going to play around. OK, missed that the first time; I clicked in the wrong spot.	2:34
7	Star shows up on map. Checked that it is saved.	USER FOLLOWED EXACT STEPS	*None*	2:05

8	"Point of interest created" and star appears.	__Click "MENU" hard button __Press "Routes" __Select "Add a Point of Interest" __Click "CANCEL" **<Performed the rest correctly after starting over>**	A little surprised viewing a route doesn't allow me to edit: Oh OK, I need to view a route first. I wonder if it would allow me to add one. H, I see, it puts it into add poi mode. I couldn't pan, I'll go back.	2:51
9	A window pops up to tell me.	__Click "MENU" hard button __Press "Routes" __Select "Add a Point of Interest" __Click on map to try to pan **<Performed the rest correctly after starting over>**	Then, it tells me. <User tried panning on the map when she was supposed to click on the map to select location of poi>	3:02

Appendix III

User Tests – Interviews (Questions 1—3)

User	1. Do you feel that this application would be useful according to its purpose?	2. Were you able to accomplish the tasks with minimal difficulty?	3. What do you feel about the navigation of the application?
1	Yes, it finds routes and got a chance to rate them. Mostly, I like that it has a database to rate, create, and view.	Yes, but, the initial start: I did not know where the menu button is because I don't know Android.	Very self explanatory: just using it this much I feel that I understand the program. The menu titles are good.
2	I do; it's cool that you can figure out a new route based on settings.	Yes.	Pretty easy.
3	Yes, if you like a route and going back to it you can add it as a favorite.	Sort of; the tasks should be a little better worded.	Was pretty easy.
4	Yes, it helps finds distances of paths; I'm always looking for distances of my hikes.	After figuring out the first couple: yes.	Not difficult. I'm not computer savvy. So, some of it was moderately difficult: but easy once I figured it out. The menu was self-explanatory.
5	Yeah, on your phone: I liked rating a route and poi's. They act like a reminder.	Very easy once I started to do it.	Easy, and I knew what to do.

User	1. Do you feel that this application would be useful according to its purpose?	2. Were you able to accomplish the tasks with minimal difficulty?	3. What do you feel about the navigation of the application?
6	Yes, if I were in an area and wanted to know local walks.	Only trouble I had was being unfamiliar with smart phone buttons. Once past the initial learning curve the software seemed very intuitive.	Incredibly simple. Two main options; very straightforward; no problems with the menu at all.
7	Yes; Nice to have for people who like to exercise.	Mostly, but, more feedback like for making a route nothing showed up. Need clues to get to the menu.	Always hidden menu; have something to pop it up like an arrow at the bottom of the screen. Have it able to reload the summary again.
8	Yes, good tool for people who are new to the area. It accumulates user content and draws on a collective knowledge.	Some struggle: viewing a route and edit was separate from add poi.	Makes sense; what I expected.
9	Yes, it works. I like that it's localized.	Not difficult at all.	Pretty easy. Once I knew I had to click the menu button.

User Tests – Interviews (Questions 4 & 5 plus user statistics)

User	4. What did you dislike or find difficult about the application?	5. Are there any extra features that you would add to the application?	Level of computer user (1:low-4:high)	Age
1	A little confused about the wording and difference of "Route Recommender" and "Route Creation".	Number of times I walked a route or the last time I did.	3	23
2	Nothing	No	3	18
3	It was fine. I was slow. (referring to emulator)	Maybe pictures, I'm a visual person.	2	18
4	I didn't know to look for stars. (referring to poi's)	Elevation gain. Animals to view for the seasons. Berry picking for seasons. Sights to see: destination descriptions.	1	44
5	At the beginning it was a little difficult.	Keyword searches for routes, but seems like it is all there.	3	50
6	Recommend level: A little ambiguous. User creation screen: 2 boxes for password Back swoop arrow to get back wasn't intuitive.	Indication of where I am in reference to the map and perhaps to a route as well. Use GPS.	4	30

User	4. What did you dislike or find difficult about the application?	5. Are there any extra features that you would add to the application?	Level of computer user (1:low-4:high)	Age
7	Some lack of feedback.	Pictures: associated with poi so a user can say, "oh, I saw this in the application, it's up ahead". GPS: where you are on the route.	4	27
8	No deal breaker and seemed to be pretty intuitive. Some feedback redundant.	GPS: Ability to meet up with people. Broadcast to tell others what you are doing and where you are--social aspect part of app. Edit mode: some visual cue so you know what to expect so you know you aren't in pan/zoom mode.	4	31
9	I don't think I found anything difficult.	Cool idea. Globalize it: for other locations.	3	25

User Tests (Android Phone)

Task	Expected Time (mm:ss)	Actual Time	Difficulty	Comments
1. Creating a New User Account	0:45	USER FOLLOWED EXACT STEPS	USER FOLLOWED EXACT STEPS	(Tilting the device made the task impossible.)
2. Read Instructions	3:00	2:53	0	Oh, not in "routes", let's go back.
3. Change Your Level of Recommendation	0:45	0:50	0	The back-up button kicks you out of the application.
4. Find Paths for Jogging	3:00	USER FOLLOWED EXACT STEPS	USER FOLLOWED EXACT STEPS	(Tilting the device made the task impossible.)
5. Build Your Own Route	2:30	1:40	0	There is a delay when I click on the map which caused me to click twice.
6. View a Route	2:00	1:20	1	Again the return key is kicking me out of the application.
7. Rating a Route	1:30	1:23	1	What if titles are the same in the database? Wouldn't that be bad? A, a confirmation box; that is good.
8. Adding a Point of Interest	3:00	2:32	1	OK, it didn't clear out my route. And it didn't give me a route list to pick from. Oh? Panning is not allowed?

User Tests—Interview (Android Phone)

1. Do you feel that this application would be useful according to its purpose?	2. Were you able to accomplish the tasks with minimal difficulty?	3. What do you feel about the navigation of the application?	4. What did you dislike or find difficult about the application?	5. Are there any extra features that you would add to the application?	Level of computer user (1:low-4:high)	Age
I wouldn't use it myself, but if you were someone new to town this would be great for recommending safe routes nearby.	Nice to have the hard keyboard. Perhaps disable tilt. But, otherwise I had no trouble.	I thought it was good. Buttons were the right size. No problems with fat fingers.	I really have no opinion. IT works well with the Google api-- that's cool.	Get the hard keyboard working. Plus, deal with the tilt sensor. Add in something to tell you what mode it is on.	4	21

Appendix IV

Initial User Interface Sketches
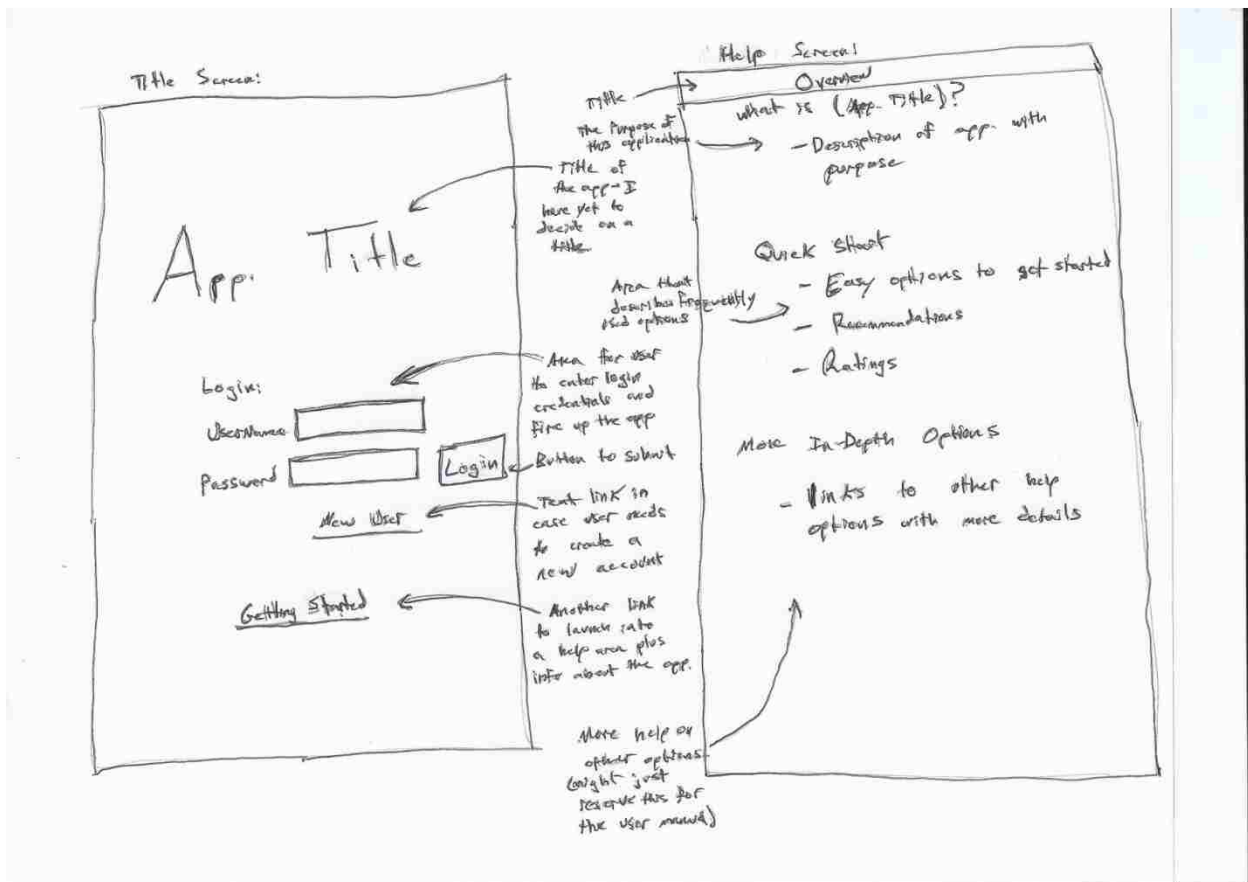
Figure 1: The sketch of the title screen (left) and a mock-up of the help guide (right).
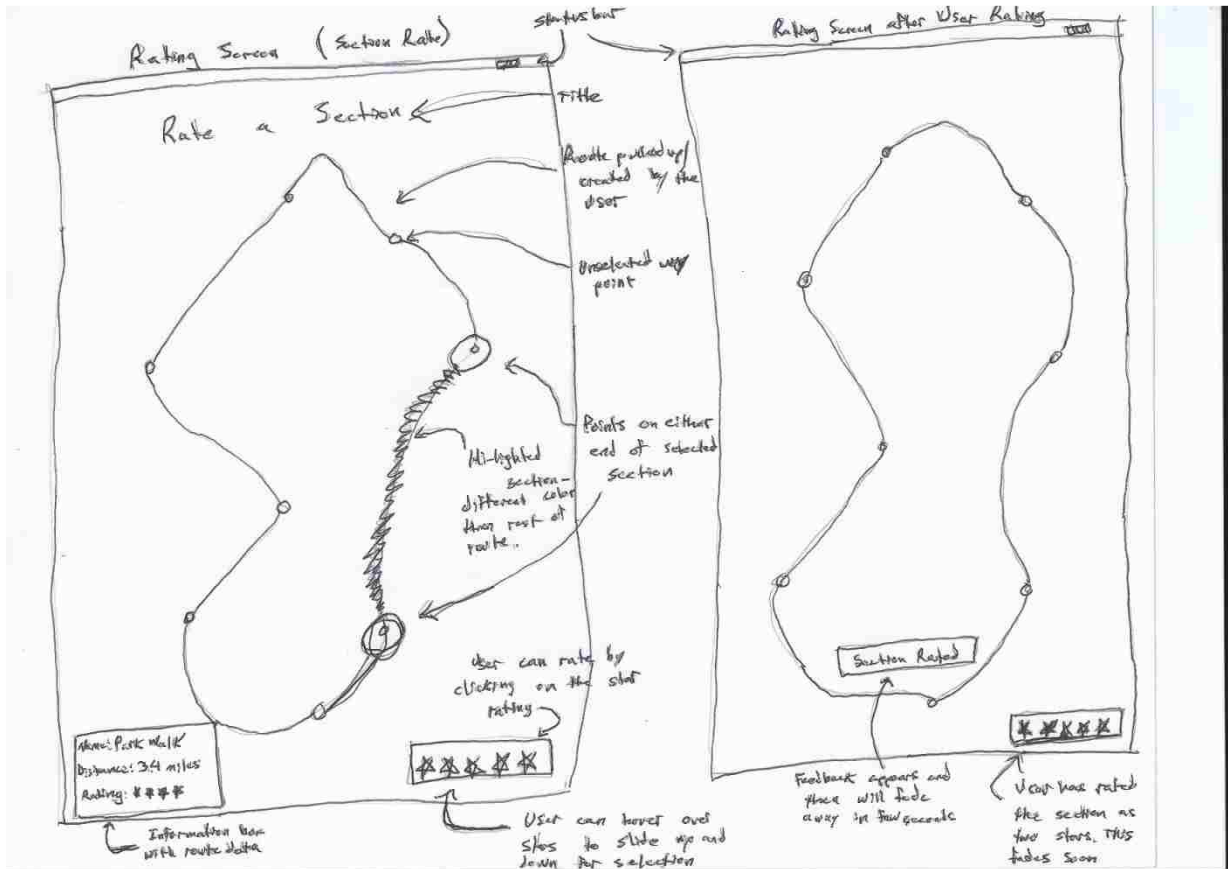
Figure 2: Sketch of section rating screen (left) and a screen immediately after a rating (right).
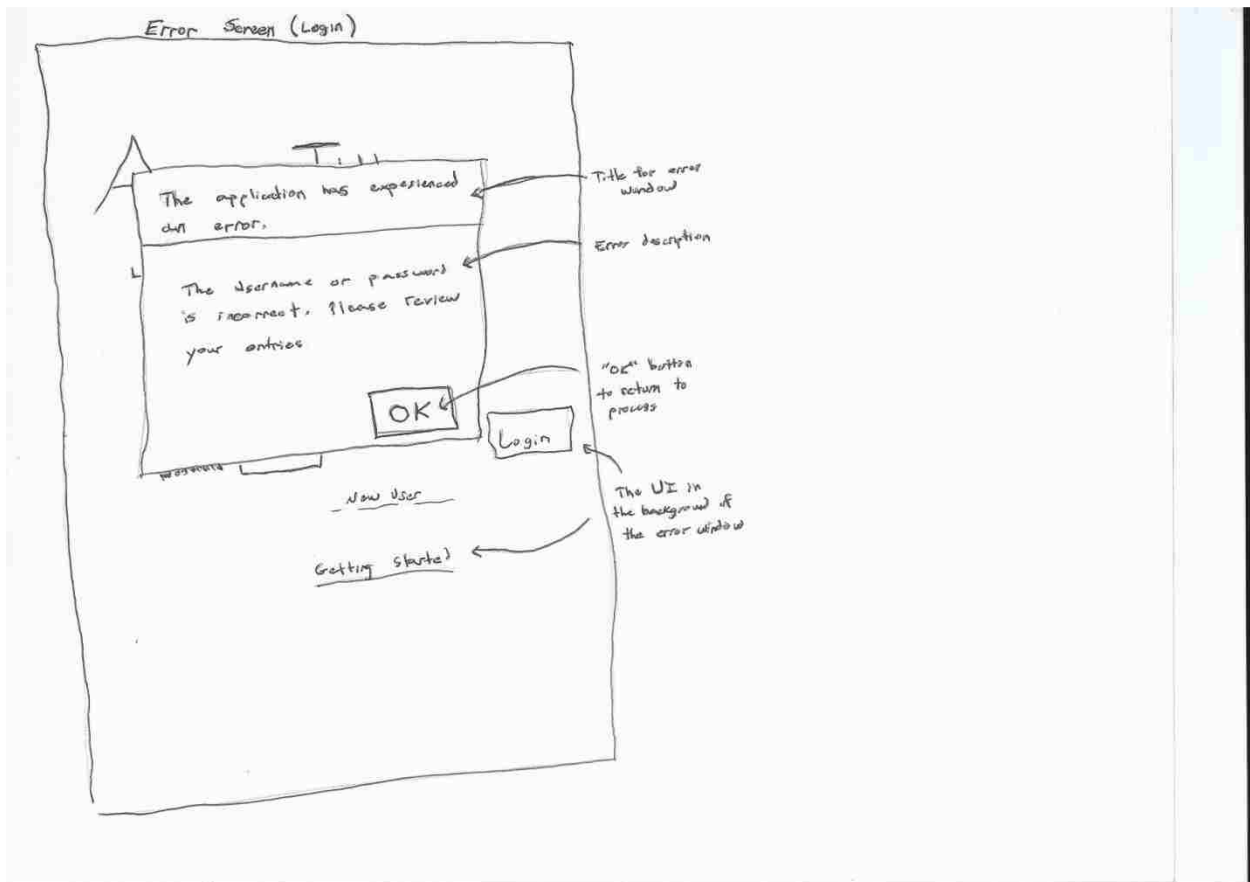
Figure 3: Error screen after user has entered improper credentials.
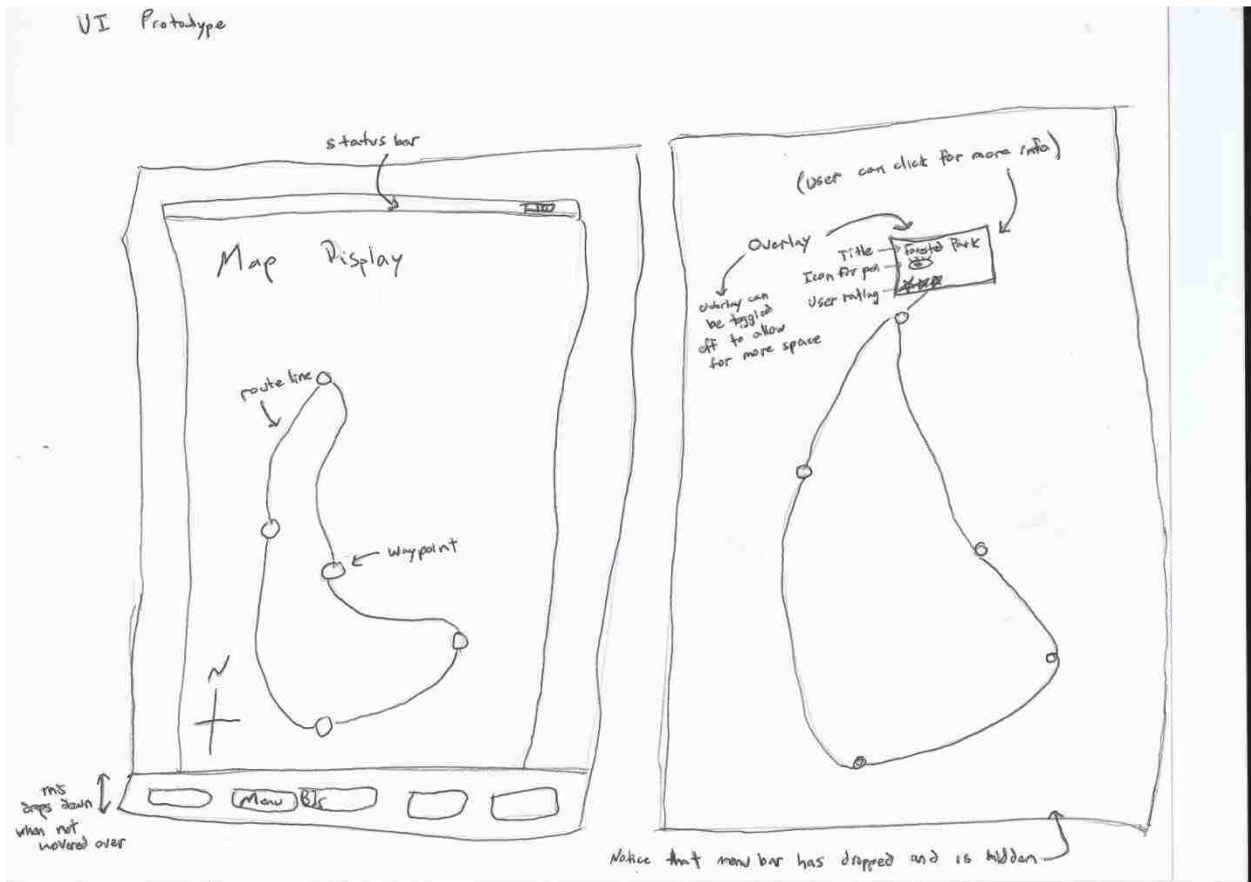
Figure 4: Early route view prototype. This (left) shows a route with the menu bar appearing on the bottom of the screen. The other image (right) shows the screen after the menu returns to its hidden state.
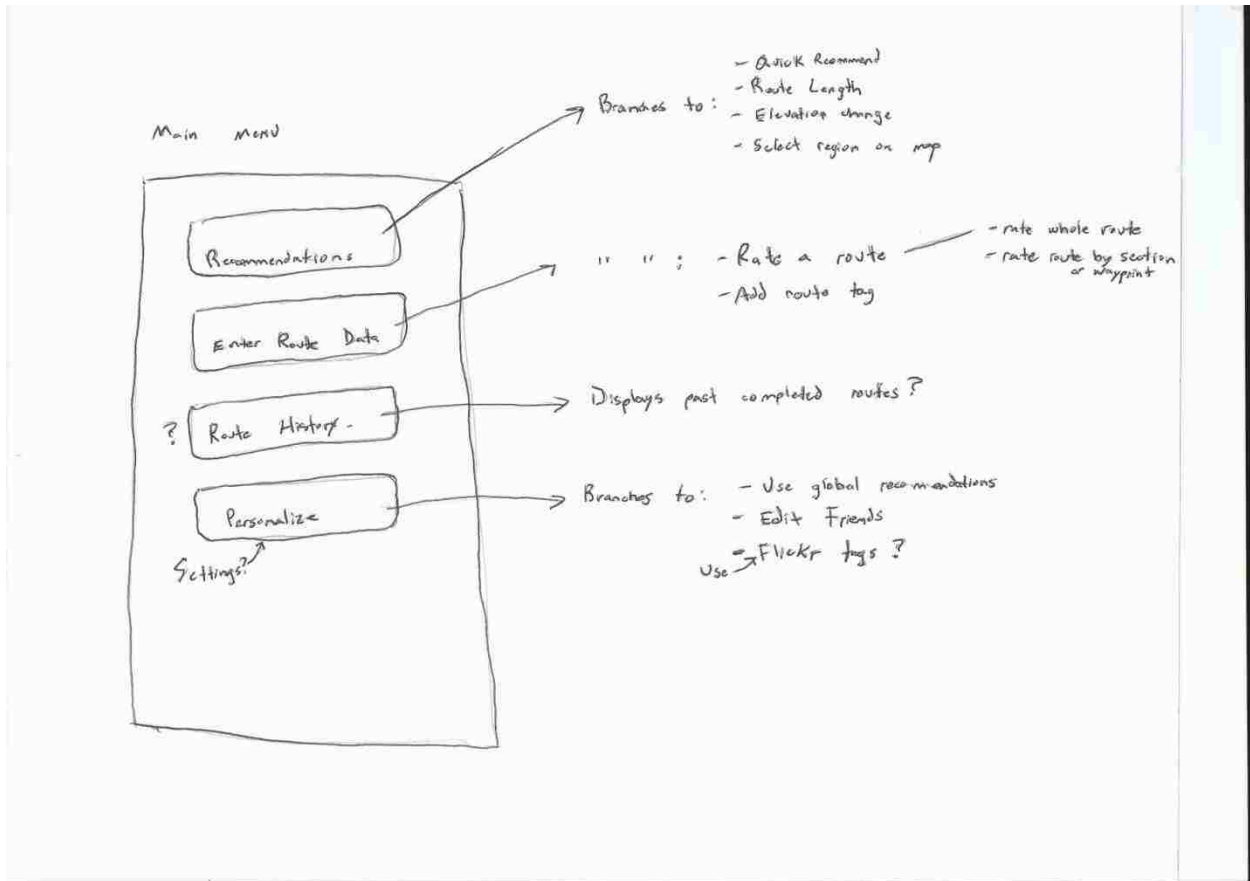
Figure 5: Early mock-up of the main menu.
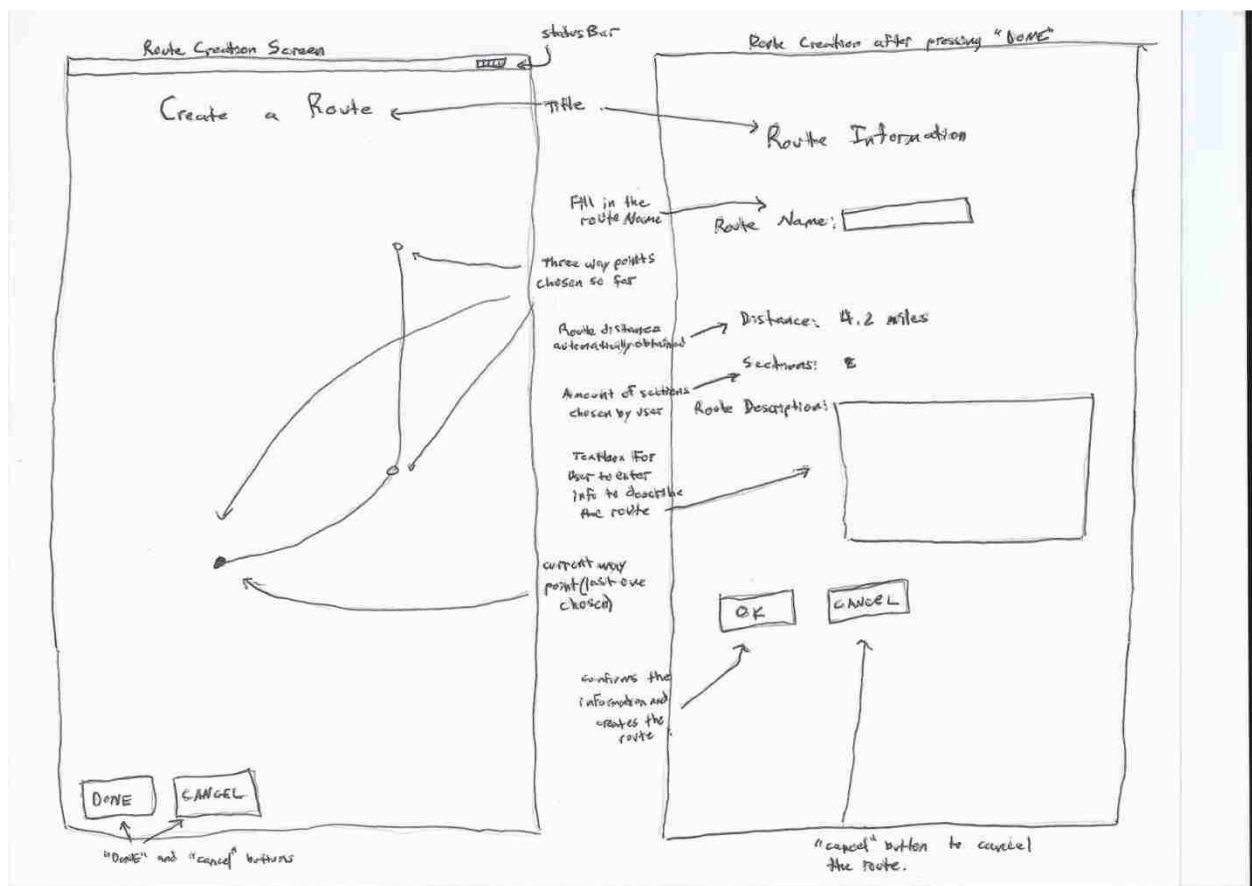
Figure 6: Early vision for route creation screen (left) and the route information input screen (right) after user had clicked "DONE" on the creation screen.

Appendix V

Entity Relationship Models and Corresponding Schema
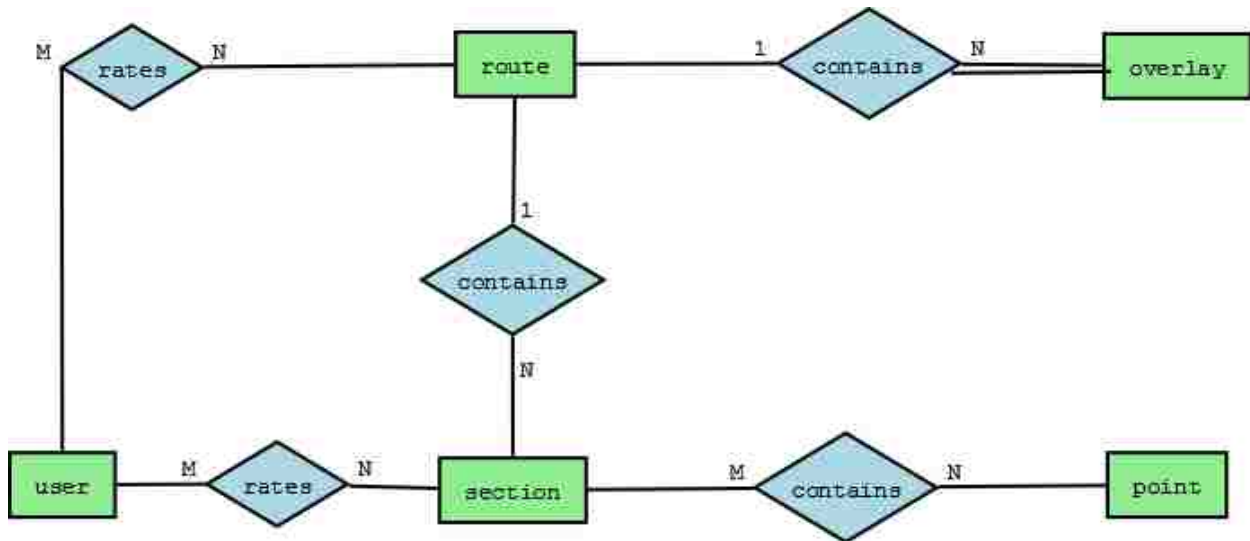
Figure 7: ER model for the routes database. This diagram does not show attributes for entities.
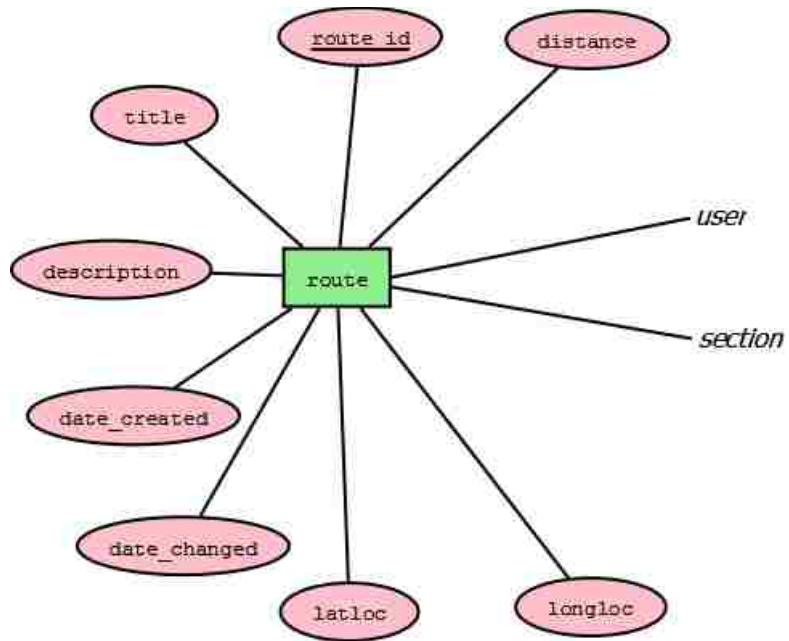
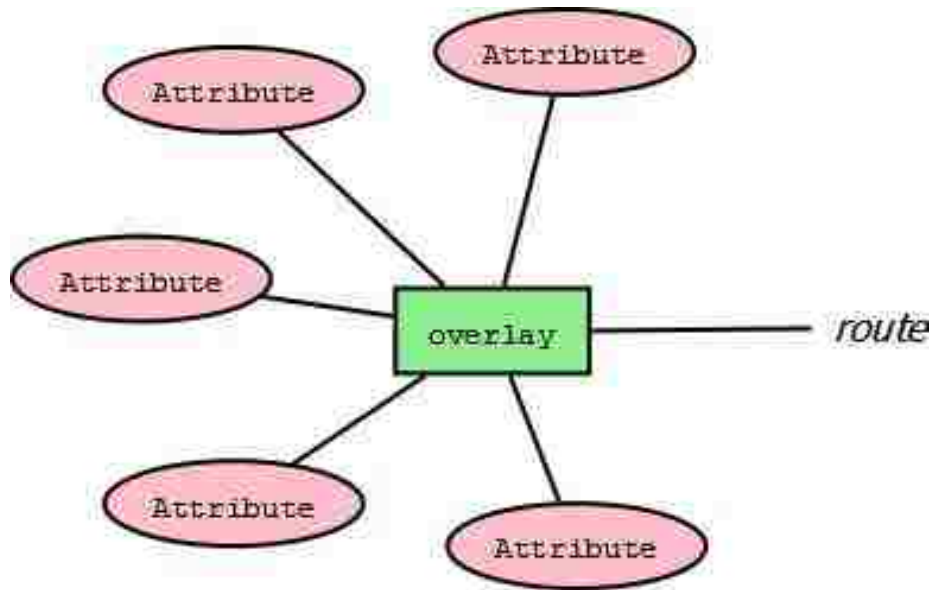**Figure 8: ER model for the route entity.**
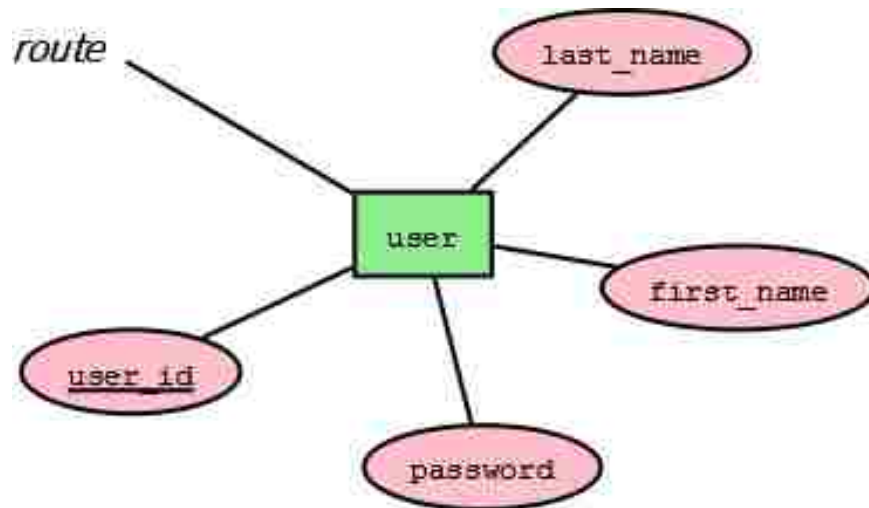
Figure 9: ER model of the user entity.
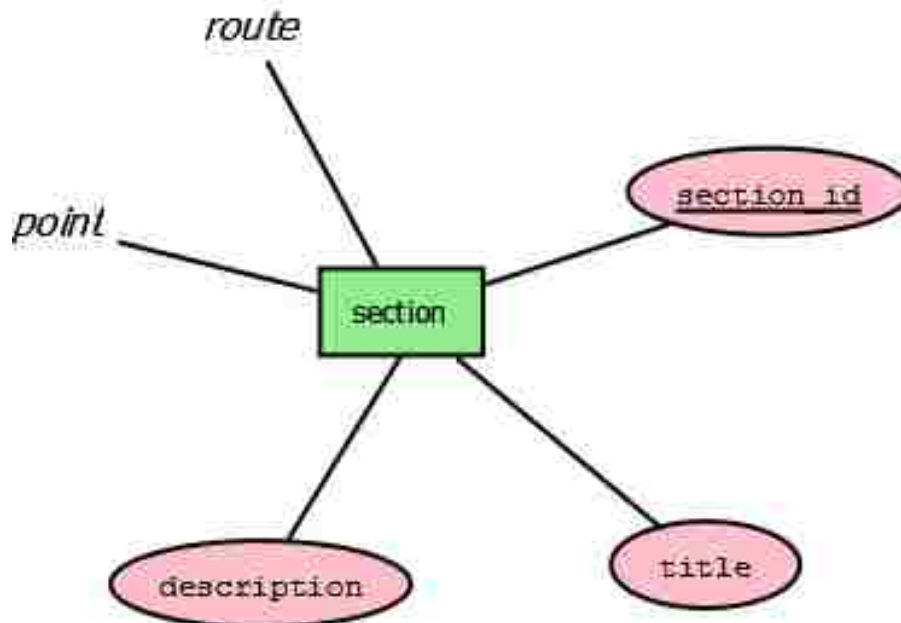
Figure 10: ER model of the overlay entity.

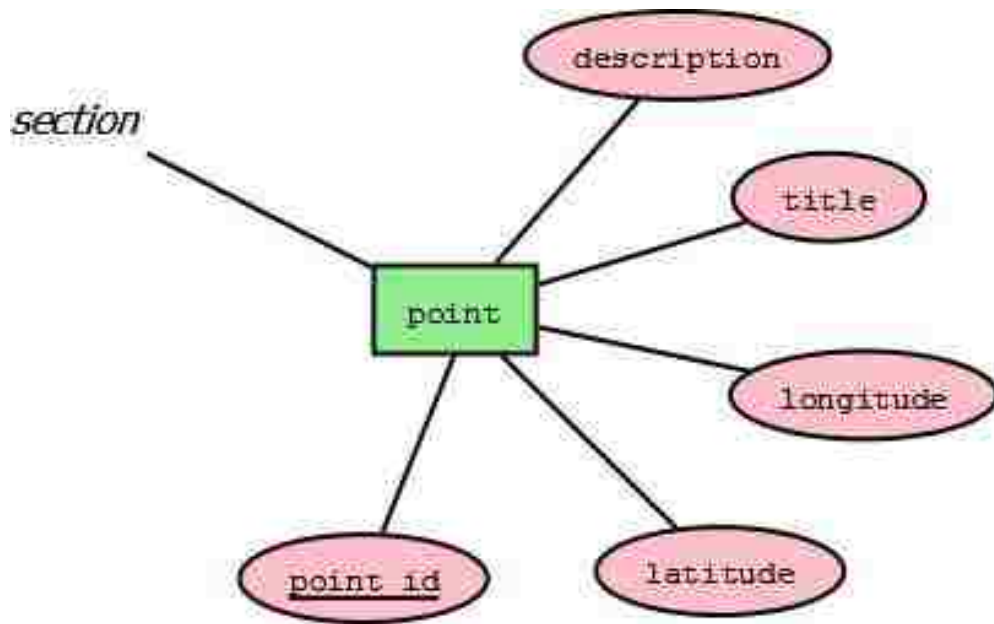**Figure 11: ER model for the section entity.**
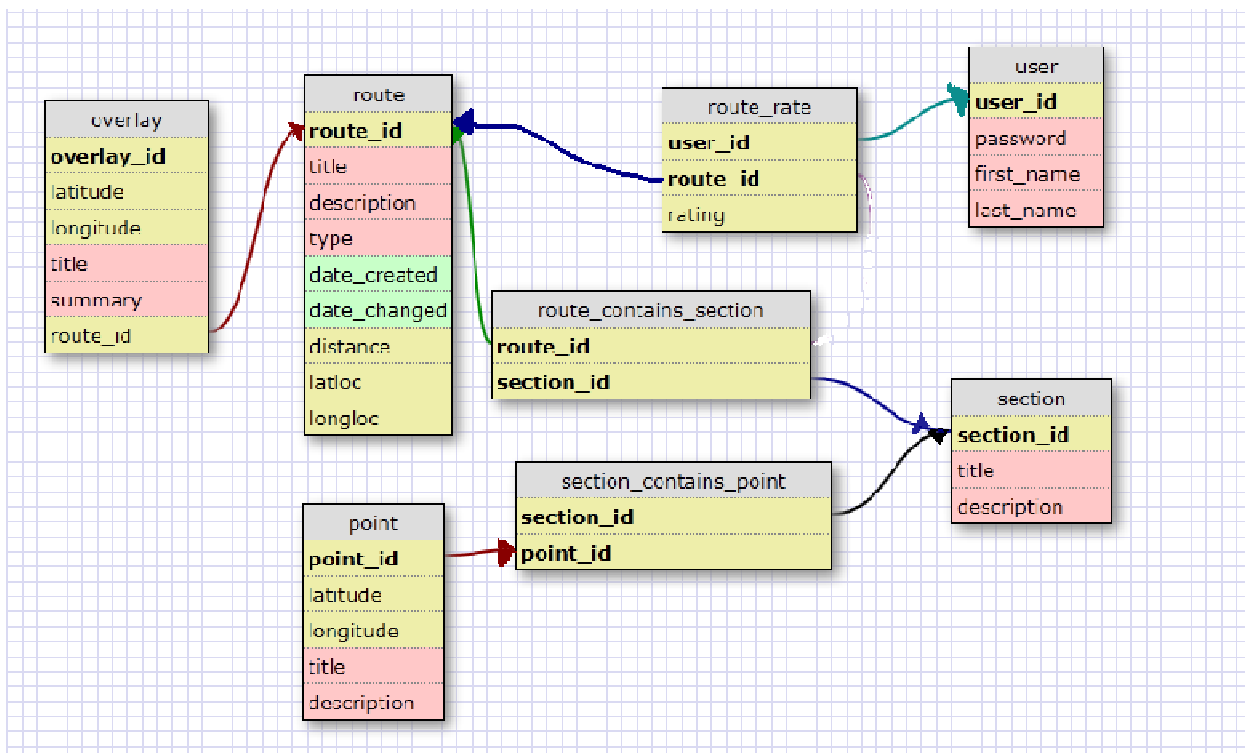
Figure 12: ER model for the point entity.

Figure 13: Schema representing the database for the application.