TheHappyDriver

Jerry Li, James Shih, Kevin Bang

Task Analysis and Design Sketches
CSE 440
Autumn 2011

Group

Group Manager	James Shih
Documentation	Jerry Li
Design	Kevin Bang
Testing	Jerry Li

Problem and Solution Overview

Traffic is always a huge problem for people living in big cities that can cause much frustration and cost a lot of time. Although it is fairly easy to acquire information about current traffic conditions, it is impossible to stare at your computer all the time. Thus, a mobile application that combines the benefits of different resources available to make commuters completely informed about their daily commute would solve the problem we currently have. The idea is to have a mobile application that not only provides the traffic information you need, but also includes the feature of monitoring your route for you and would notify you if there is a potential delay on your daily commute route.

Current Versions of Tasks

Easy task: *Checking daily commute route and setting up alarms.*
You're about to commute to work. You'd like to make sure the route you take every day is clear. After you've done so, you'd also like to check your estimated time of arrival, just to make sure you're not late, and set up alarms to inform you if anything happens to the traffic on your way to work.

Medium Task: *Checking traffic conditions and using the navigation feature.*
You and your family are planning a trip to Seattle today (it's Saturday). You want to get to Seattle no later than noon, but you're willing to take some risk of arriving late so that you can leave later. As it turns out, the Huskies are playing at home today. You'd like to check the current and projected state of the roads, and then you'd like to get the best route given traffic, and have the application guide you while you're driving if the route is unfamiliar.

Hard task: *Getting an alarm from the application unexpectedly while driving.*
You're on your way to work in Seattle from the east side, like any other day, when unexpectedly you get an alarm from our application telling you that the 520 bridge was closed because of an accident happened a few minutes ago, and that therefore you'll need to take another route. While driving, you'll need to find the best route, and have the application direct you.

Task Analysis Questions

1. Who is going to use the system?
Our target primary users are commuters who drive to work/school on a daily basis. Also, anyone who needs to plan a trip and is worried about the travel time is a potential user as well, since the application can serve as a type of a trip planner. Our users should be able to read maps, know how to operate a smart phone and have become used to the basic usages of the application.

2. What tasks do they now perform?
Commuters might be able to check the road conditions on their computers before they take off. However, if the conditions change while they are on the way, they may still get stuck in traffic and be late for work/school.

3. What tasks are desired?

Users (commuters) should be able to set the alarm to inform them if the road conditions change on their commute route while they are on the way. If the alarm is turned on, users will get a notification if something goes wrong on their route while they are traveling, with some possible suggestions. Additionally, the user will use the application to check the current and projected traffic condition, and get the best trip route given this traffic condition. Also, they will be using the application during the drive as a navigation tool.

4. How are the tasks learned?
Commuters who can read maps and know how to use a smart phone should be able to pick it up very quickly. The application's first screen is one of a traffic map. This is one of the two main functionalities of the application, thus this option will be made noticeable in the design. The map can always be navigated to from the "Map" tab.

5. Where are the tasks performed?
Checking the traffic condition and planning a trip is done in any situation not involving an active driving. It can be done at home, within a car, etc. The navigation function is used primarily during an active driving situation.   If there is an alarm or alert while driving, the tasks are usually performed in the car.

6. What's the relationship between customer & data?
Customer will be able to get a detailed feed regarding the traffic information they want, which they can use to determine their trip plan. The data is always fed on request; user will never get any unwanted data, so users will not be flooded with data they don't actually need. Also, their personal data will be stored on the phone, except their current location may be exposed to the application (not other users) to determine what they need. For example, navigation data will be essential while driving.

7. What other tools does the customer have?
To perform these tasks, tools provided by the application would be sufficient. There is the alarm tool used for easy task and hard task, and there is a trip planner / navigation tool for medium task.

8. How do customers communicate with each other?
This is supposed to be a single-user oriented application. Any data presented by the application is not meant to be distributed or shared in indirect terms such as e-mail, SNS, etc. Customers can always share the data by looking at the other person's device though.

9. How often are the tasks performed?
Traffic condition checking is expected to be carried out multiple times in a day, depending on where the customer is located. If the customer is located within metro area and concerned about the traffic, this tool will be used very frequently. Trip planner and navigation is not expected to be used many times if the user mostly follows the same route. Getting an alarm is expected to happen less often, considering it only notifies you when there is a really serious incident happening.

10. What are the time constraints on the tasks?
There are no time constraints on traffic condition checking and trip planner. There is a time limitation with the navigation tool due to physical constraints (i.e. the battery). GPS tends to hog up on the battery. However, the application is meant to be used in metro areas around Seattle, not for extended hours of driving, so the battery life shouldn’t be a problem in general. However, there is a time constraint on the alarm. Users will have to respond to the alarm as soon as possible to react to the change of traffic conditions.

11. What happens when things go wrong?
Traffic feed can be untimely; there can be unexpected events that can adversely affect the trip time. Should such events happen while the user is driving on the highway, the effectiveness of the application will decrease. The application is designed to react automatically if such an incident occurs, which will be important for the hard task. Sometimes, drivers (our target users) will have to react to the event without the help of the application. For example, if there is an accident happening on the highway right in front of the user’s car, the driver will have to decide what to do before the application gets the data from the system.

Storyboards

Design 1:

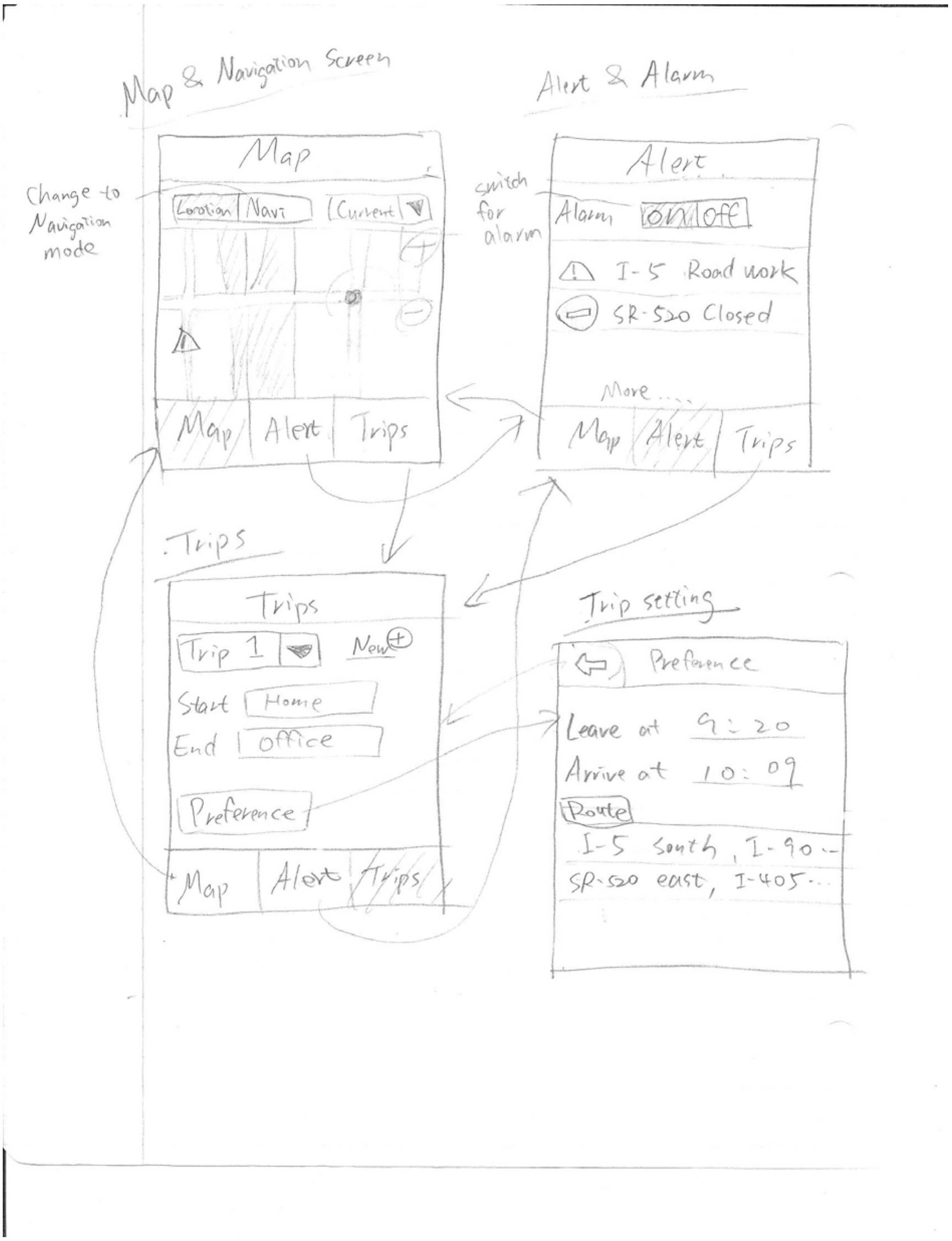

Figure 1

Design 2

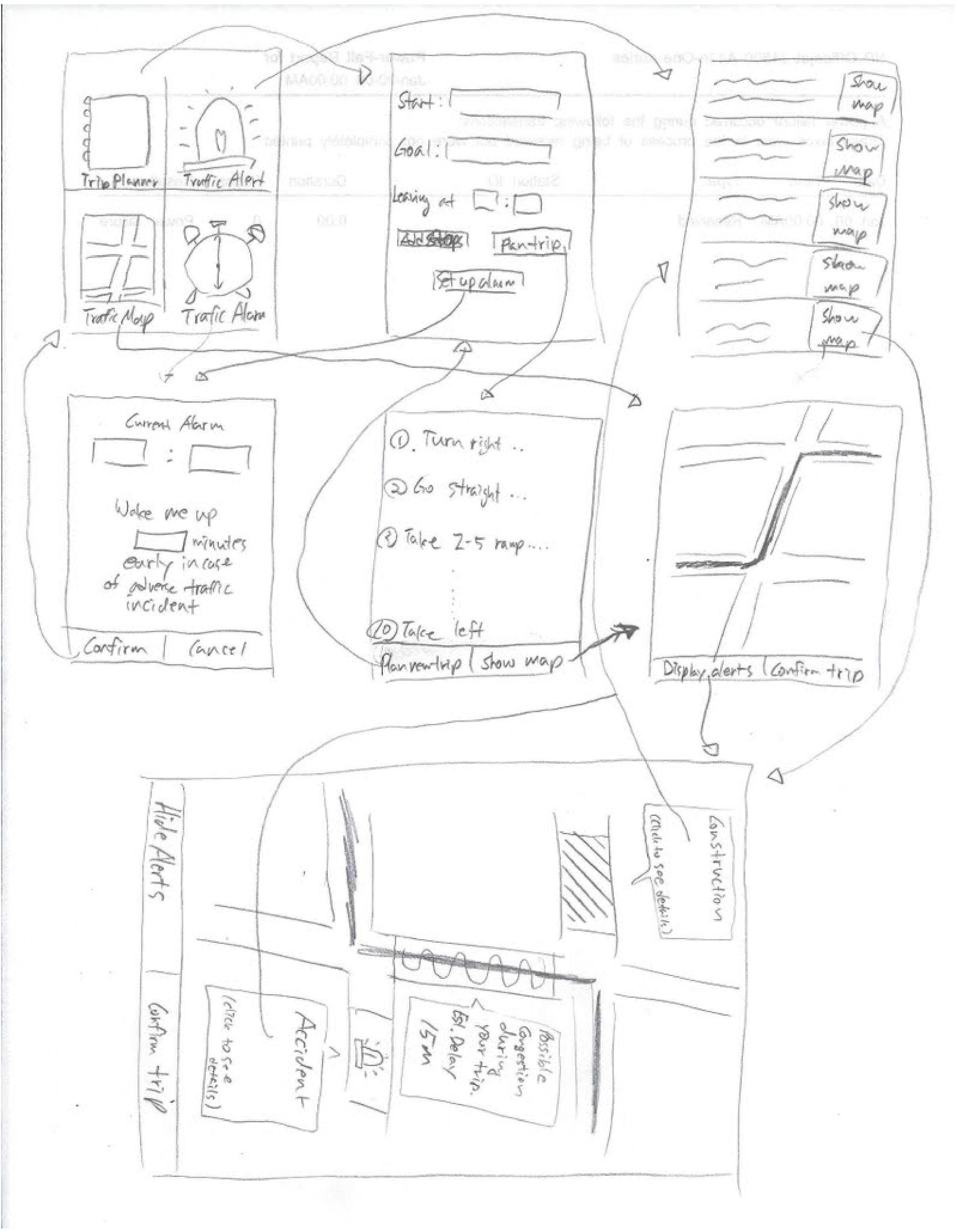

Figure 2

Design 3

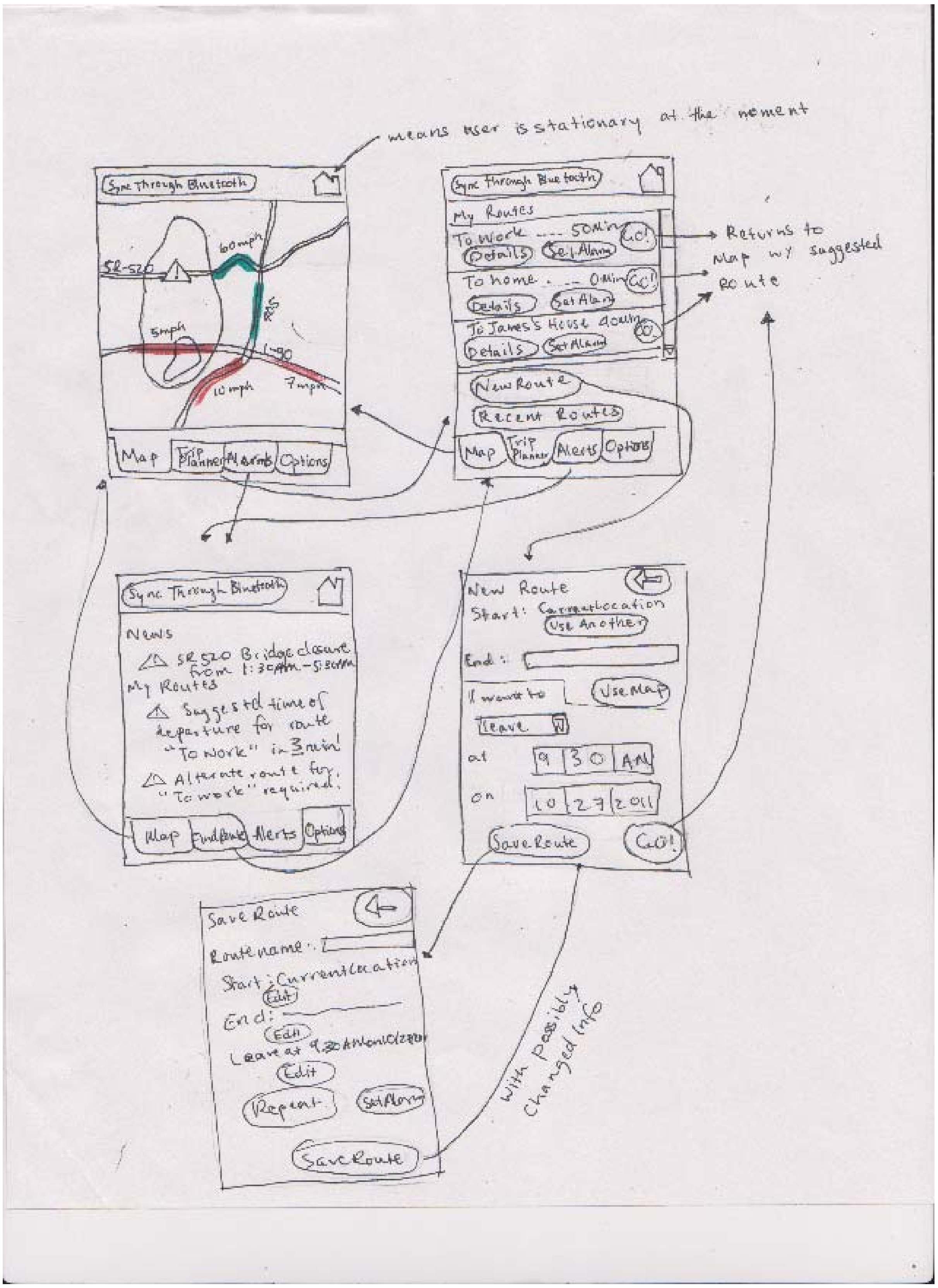

Figure 3

Selected Interface Design

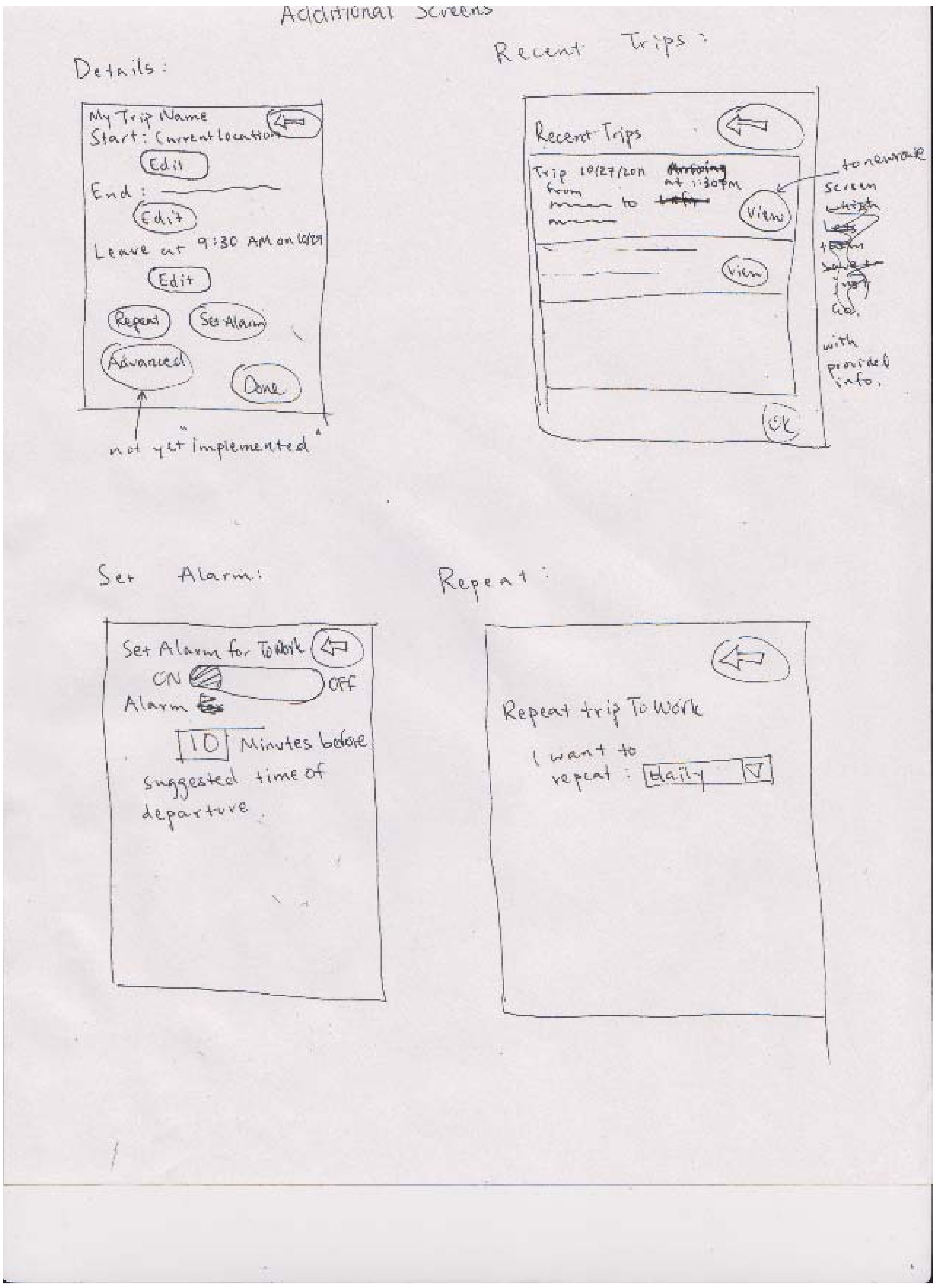

Figure 4

After some discussion, we decided to choose the third design (as shown in ***[Figure 3]***). We felt that this design is more comprehensive and covers all the features the users need to complete the tasks we mentioned in previous section. We will discuss the design, rationale and functionality of our interface in this section.

The interface opens with a view of a map of the general Seattle area with traffic information overlaid, which we show in the top-left most picture in the design sketches ***[Figure 3]***. We felt that because this information (in particular, the traffic information) was what the user ultimately sought in our application, it should be prominently displayed. On this map, traffic alerts are clearly demarcated and the map is color-coded (red for heavy, green for light), however, for accessibility reasons we also list the approximate speeds that traffic is going at different road intervals. We intend for the user to be able to zoom in and out and move around the map as they could on something like Google Maps. We know that currently the WSDOT's application cannot do this, but we feel like this is an important feature that users will use quite often and have come to expect from any map application. The navigation mainly done by switching between different tabs presented at the bottom of the screen. The first tab is the "Map" tab, and when selected presents the screen described above (as presented in ***[Figure 3]***).

The second tab is the "Trip Planner" tab. When selected, it opens up a screen similar to the screen on the top right of the third sketch ***[Figure 3]***. There is a list of routes that the user has created. The user can navigate through this list, and if they find the route they want to go on, they can press the "Details" button which leads to a screen almost exactly like the "Save Route" screen (just with a name already inputted) to find more details about the trip, including the start and end location, whether or not it repeats, and when the trip is supposed to start or end, the "Set Alarm" button which brings them to a screen like the screen in the bottom-left in ***[Figure 4]*** where they can set the details of an alarm for the route, or the "Go!" button, to start the trip, which returns them to the map screen, but now also with a trip plan, and capabilities to turn on and off voice guidance. On the details screen (top-left screen in ***[Figure 4]***) the user is also able to press the "Edit" buttons which let them edit the appropriate field. For ease of use there is also a "Set Alarm" button here which leads to the same screen as the first "Set Alarm" button. One important idea we discussed for a while was whether we could have the user select how much risk he/she was willing to take on an alarm. Eventually, we decided against that feature, because apparently similar certainty features in other predictive software applications such as Zillow could really diminish the reliability of the program in the eyes of certain users.

If the user cannot find the trip they want, they can either click the "Recent Trips" button below, if they took the same trip recently, which brings them to a screen like the one in the top-right in ***[Figure 4]***, which has a similar list of trips and “Go!” option, as well as a “Save Trip” option to save the trip onto the original list, or a “New Trip” button, which makes a new trip. If they click this button, it takes them a screen similar to the one on the bottom right of the sketch in ***[Figure 3]***. In this screen, they are to specify the start and end locations of the trip (with an option to use the map), as well as when they want to leave or arrive. We need this information because without the time, as Dr. Hallenbeck mentioned during our CI, we cannot get a valid traffic estimate. After they input this information, they can either immediately click “Go!,” which again directs them to the map with the relevant trip information, or they can save the route by clicking “Save Route,” which directs them to the screen on the bottommost of the sketch in ***[Figure 3]***. This screen asks them for a trip name, as well as allows them to review and edit the trip information they input before. There is also a “Repeat” button, which after being clicked opens another screen like the one in the bottom-right in ***[Figure 4]***. This screen asks them to specify when (daily, weekly, on weekdays, etc.) they want to repeat the trip. Back on the “Save Route” screen, the “Set Alarm” button brings up another screen like the previous “Set Alarm” screen in the bottom-left screen in ***[Figure 4]***.

The third tab is the “Alerts” tab, which brings the user to the screen in the middle left of the sketch in ***[Figure 3]***. It simply prints out any WSDOT news and also alerts for the user about alerts for their trips. When the user is currently on a trip any alerts which affect their current route will pop up automatically (although there will be an option to disable that). We initially had the user have to choose to have these alerts pop up; however, we felt that because the entire purpose of the application was to avoid traffic, then having to manually set alerts was counter-productive. The final tab is the “Options” tab, which we haven’t fully fleshed out, however will let the user do basic setup for the application.

On the top right of all non-option screens is an icon depicting whether or not the user is currently driving. If the user is currently driving, then the application by default goes into “driving mode” (which can be manually deactivated). In driving mode, the application uses voice instructions to direct a driver and to deliver traffic alerts. The map also automatically uses GPS to track where the mobile device currently is (although this can also be manually deactivated). Although in class it was suggested to avoid moding, here the only real difference is that voice directions are used and the GPS follows the driver, and we felt that the safety concerns made this quite necessary.

On the top-left of all non-option screens is a button labeled “Sync Through Bluetooth.” If a user clicks this button, and if Bluetooth is active on the mobile device, the application searches the vicinity for compatible Bluetooth devices such as an onboard car computer or a GPS system. After the user selects a compatible connection and it is confirmed, the application and all of its settings will sync with the device and can then be viewed on the new device. This will be most useful when the user begins to drive, as then voice instructions can be given by the car or the GPS system, and the user has a larger screen to see the map and interact in a simpler fashion with the interface.

Scenarios

These are the scenarios of the tasks that we made after taking into consideration what we found during the task analysis and the specifics of the design:

1. (*Easy*) The user first checks the traffic conditions on the map, if they want. They then press the “Trip Planner” tab and scroll to their commute plan that they’ve already input into the system. At this point, they can see how long their commute is projected to be. To set up alarms for this route, they then click the “Set up Alarm” button associated with the trip, and on that screen they select their desired start/stop time and other options. After having done so, the user can then press “OK” and return to the trip screen, at which point they can either exit or select “Go!” to star the trip.
2. (*Medium*) The user can again first check the traffic conditions on the map. They then press the “Trip Planner ” tab and click “New Route.” On the screen that they go to, they select their start and end locations, either by keyboard or by using a map, and select that they want to arrive at 12:00 PM. They cannot select how much certainty they want, as noted in the design section. After saving the route, if they want, they can press “Go!,” at which point the application returns to the map with a suggested route. When they get into a car, if they find a Bluetooth-compatible device in the car, they can then sync it if they want. When the car starts moving, the device goes into driving mode, and voice instructions are automatically given to the driver.
3. (*Hard*) The application is currently in driving mode, so the alert is given out using a digital voice. The application is designed to handle this hard case very well. It will ask whether or not the user wants to use an alternate route, or if they want more details, and if the user confirms (with a vocal Yes/No/Details which the application then parses), will automatically calculate the best route and begin to direct the user onto this path, with the directions on the map also automatically updated. If the user chooses to stay the course the application does nothing, and if the user chooses to ask for details the application will read

out the description of the accident and the predicted travel times, as well as printing them out on the screen.