**EECE 3093 Software Engineering Spring 2016**

**Lab 3**

**Team Members: William Hauber, John Miller, Matthew Tucker, Dillon Staub**

**Project Start Date: 3/15/2016**

**Report Submission Date: 3/29/2016**

**Table of Contents**

**Objective………………………………………………………………………………………..3**

**Team Member Roles…………………………………………………………………………...3**

**Feature Status…………………………………………………………………………………..4**

**Communication & Coordination Strategies…………………………………………………12**

**Experience Gained…………………………………………………………………………….12**

**Future Risks……………………………………………………………………………………13**

**Emulator Note………………………………………………………………………………….13**

**Code Notes……………………………………………………………………………………...14**

**OBJECTIVE**

Deliver (1) a data flow diagram that models the following features, and (2) a threat report that is built on top of (1). The features included in the data flow diagram are: **M-Feature1 (address search dialog)**, **M-Feature2 (show marker on the map)**, **M-Feature3 (show marker information)**, **M-Feature6 (send contact information from Mapbox to ContactManager)**, **C-Feature1 (privacy setting)**.

**TEAM MEMBER ROLES**

**William Hauber**

* Worked with Dillon Staub to create data flow diagrams
* Reviewed the threat enumeration report
* Worked with team on reflection document

**John Miller**

* Worked with Matthew Tucker on threat enumeration document
* Review the data flow diagrams
* Worked with team on reflection document

**Matthew Tucker**

* Worked with John Miller on threat enumeration document
* Reviewed the data flow diagrams
* Identified what part of the data flow diagram the threats target

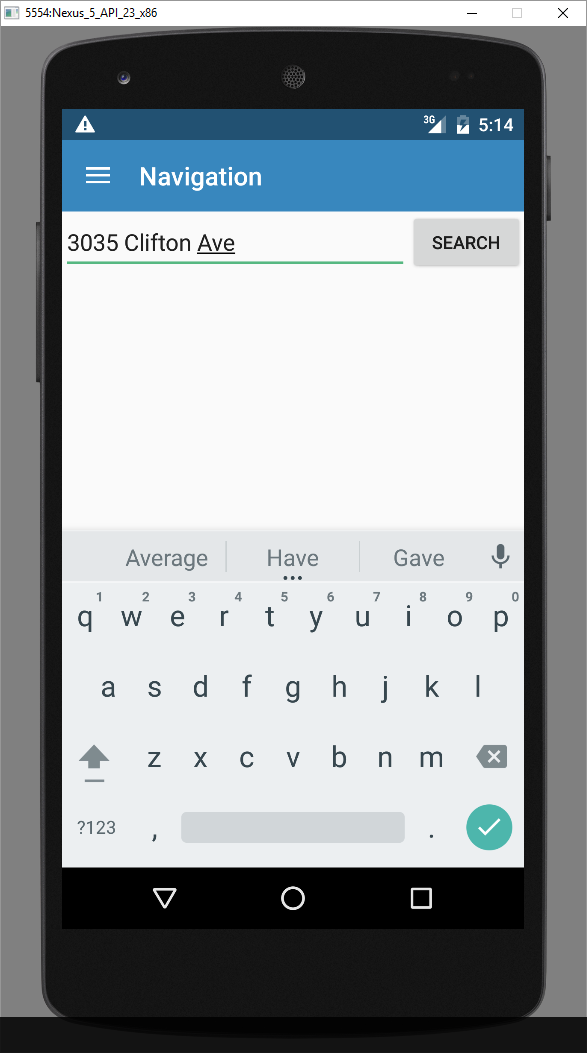
**Dillon Staub**

* Assisted in the creation of the data flow diagram in Visio
* Reviewed the threat enumeration report
* Coordinated final documentation of project

**FEATURE STATUS**

**Mapbox:**

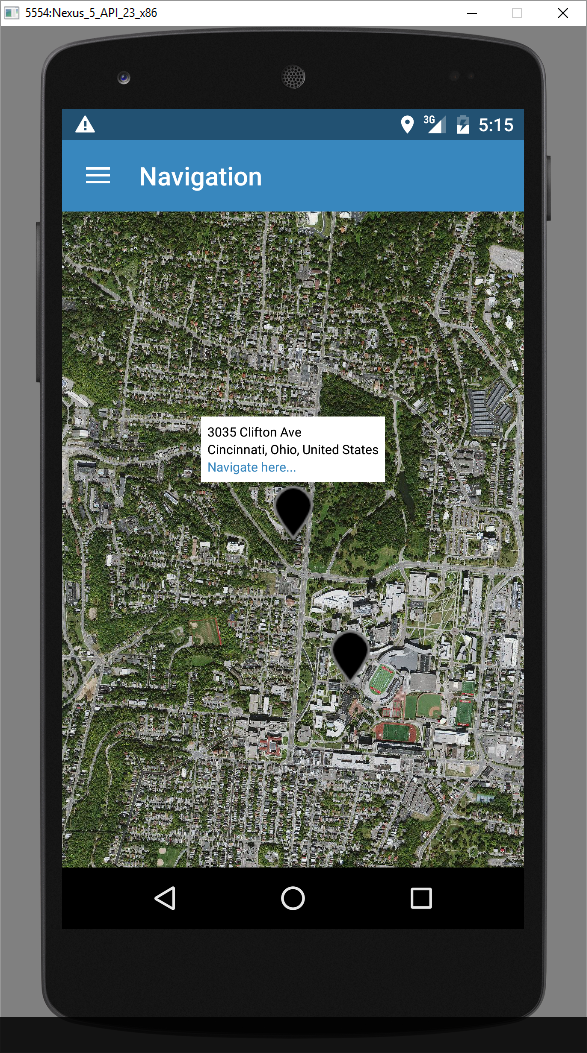
M-FEATURE1 (address search dialog): Feature 1 on Mapbox is implemented and fully functional. There is a 'Navigation" menu item at the top of the main menu that when clicked, brings up a new dialog where the user can enter an address and select "Search".



**Figure 1:** The figure above displays a demonstration of Feature 1 after it was implemented.

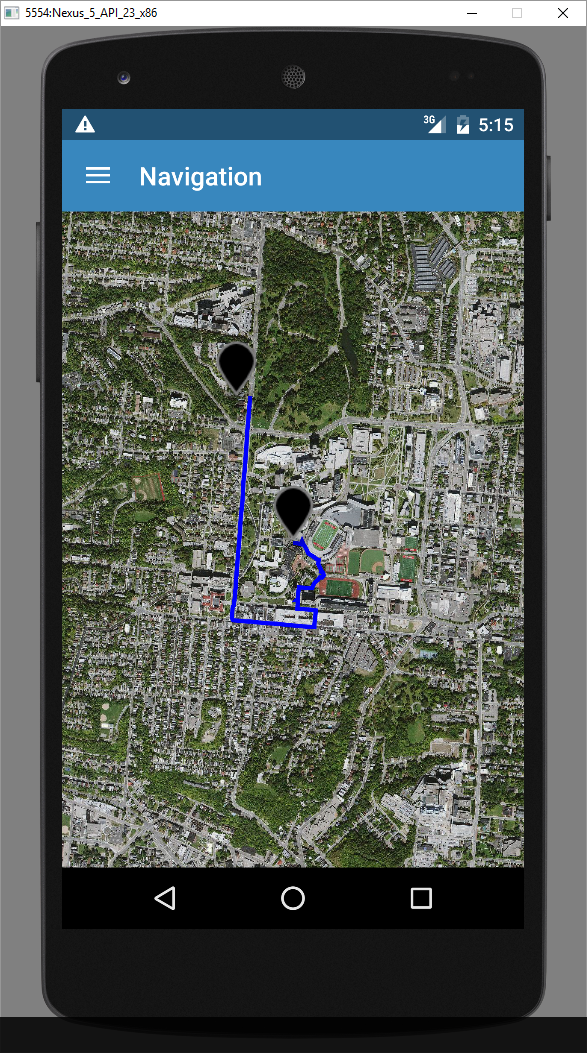
M-FEATURE2 (show marker on map): Feature 2 on Mapbox is implemented and fully functional. The user can enter an address and select the "Search" button and a marker will be placed on the user requested location. Previous searched will not be indicated on the map by a marker. Only the address currently being searched will be located on the map. This feature supports names of places in addition to specific addresses (such as "Paul Brown Stadium").

M-FEATURE3: (show marker information): Feature 3 on Mapbox is implemented and fully functional. When a marker is clicked, an informational dialog will be shown with basic information about the address.



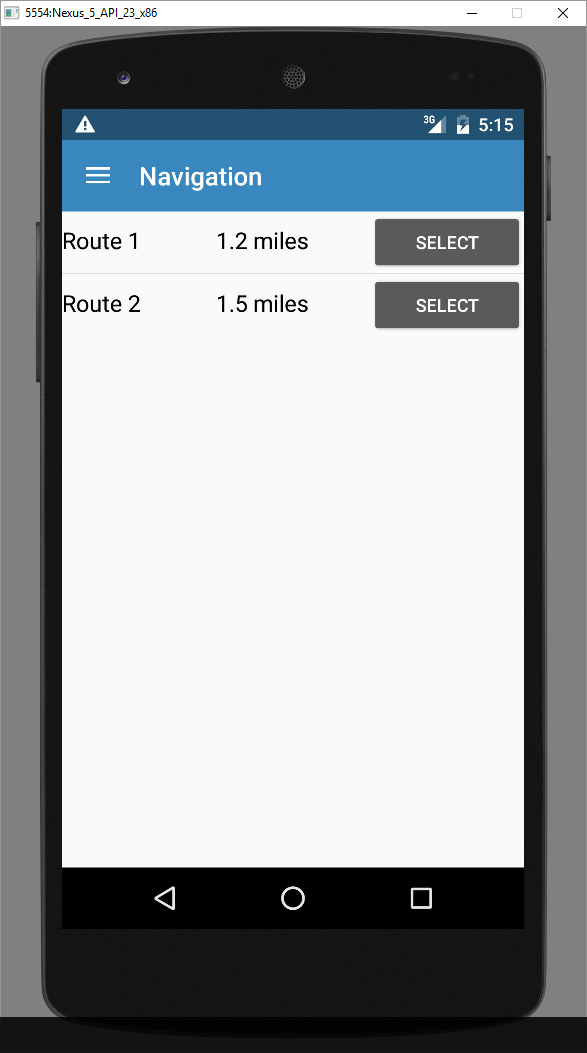
**Figure 2:** The figure above displays a demonstration of Features 1 and 2 after they were implemented.

M-FEATURE4: (route calculation): Feature 4 on Mapbox is implemented and fully functional. A hyperlink "navigate here..." was added to the informational dialog from M-FEATURE3. When clicked, this hyperlink will display the route from the current location to the requested address. Note that the current address in our program is set to the University of Cincinnati.



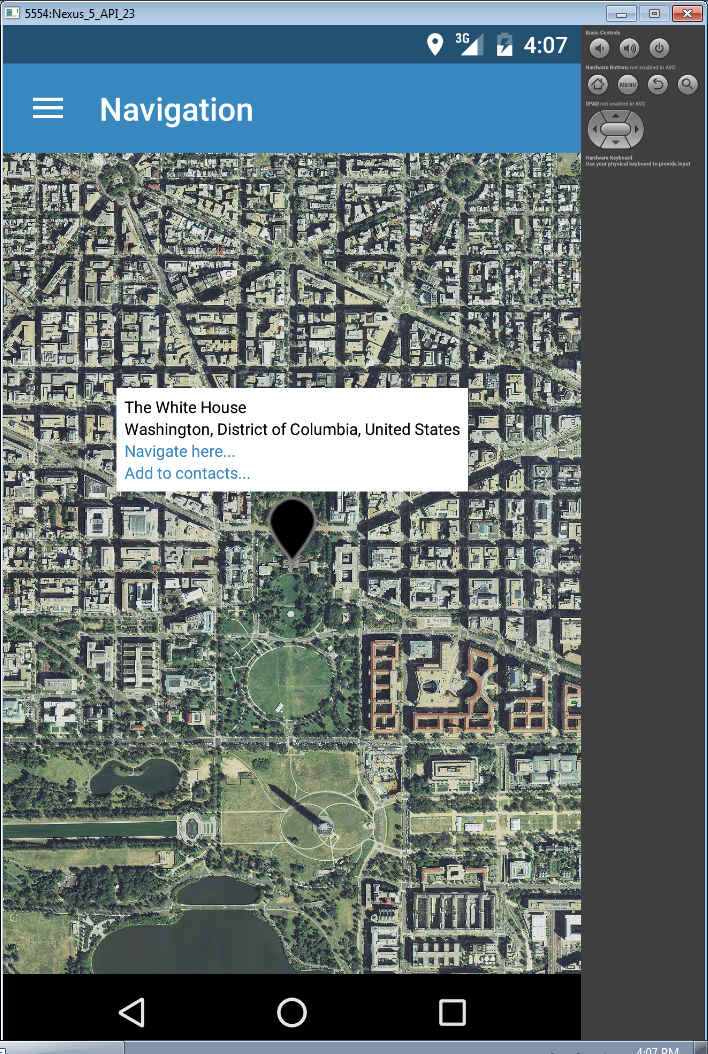
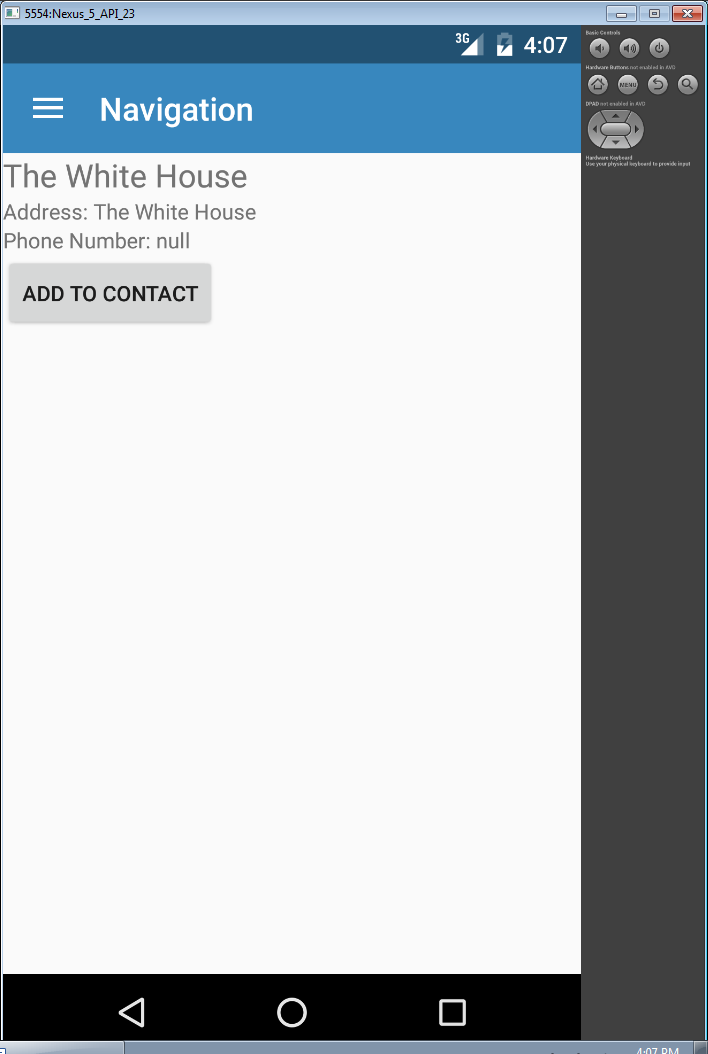
**Figure 3:** The figure above displays a demonstration of Feature 4 after it was implemented.

M-FEATURE5: (routes selection): Feature 5 on Mapbox is implemented and fully functional. If multiple routes are found between the current location and the requested location, the user will see a dialog that will allow them to select between the routes. Once a route is selected the map will be displayed with the selected route shown. Note that the user may have to click the "select route" button twice in order for the route to be displayed.



**Figure 4:** The figure above displays a demonstration of Feature 5 after it was implemented.

M-FEATURE6 (send contact information from Mapbox to ContactManager): Feature 6 is fully implemented and functional. When the user clicks the “Add to contacts…” hyperlink in the marker created in M-Feature3, a new dialog window is created which shows the marker’s contact information (may include name of location, address, and phone number), and the option to “ADD TO CONTACT” appears within a dialog window. Once this option has been clicked, the ContactManager application is launched, and the marker’s information is sent to ContactManager. This information is passed to ContactManager and is automatically filled into the corresponding fields within ContactManager. The user is then able to add this contact to their contact list by clicking “Add Contact.” After the user clicks the “Add Contact” button, the contact is added to ContactManager.

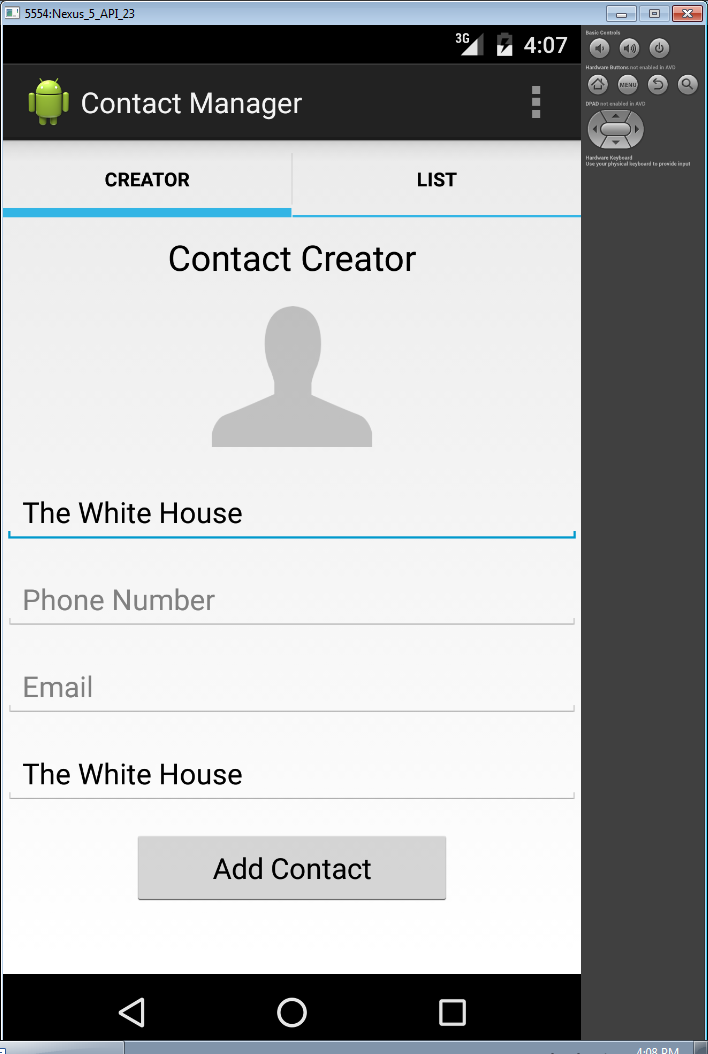
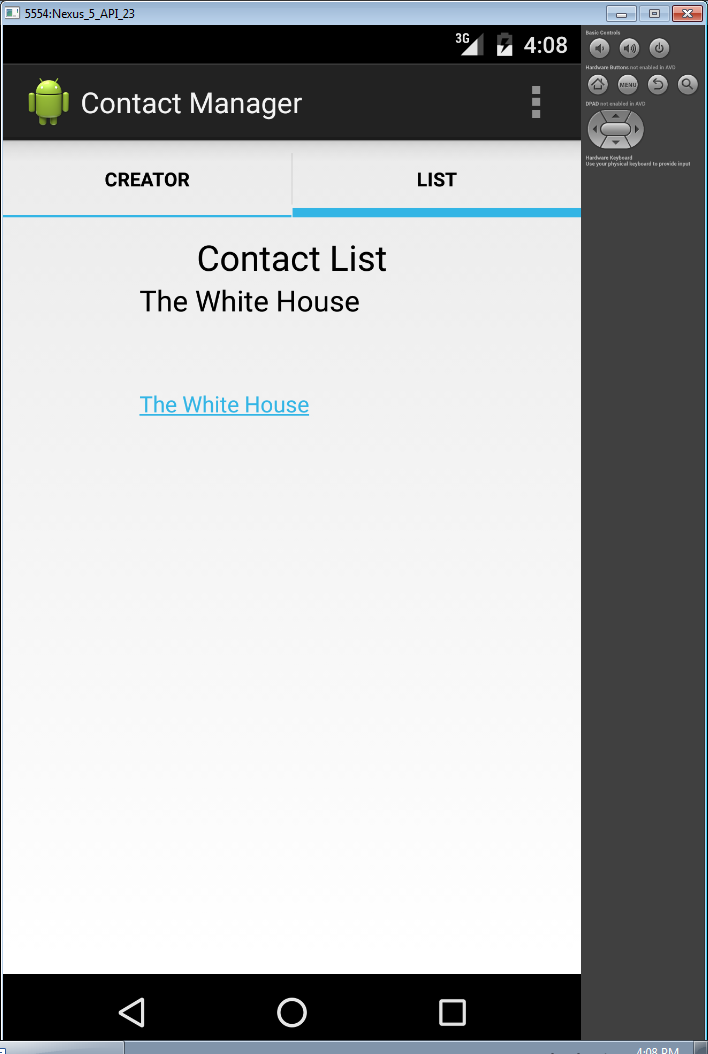


**Figure 5:** The figure above displays the **Figure 6:** The figure above displays the dialog

marker with the “Add to contacts…” window with “ADD TO CONTACT” option

hyperlink. that appears when the “Add to Contacts…”

hyperlink is clicked.



**Figure 7:** The figure above displays the **Figure 8:** The figure above displays the

ContactManager application after it has been contact list with the newly added contact.

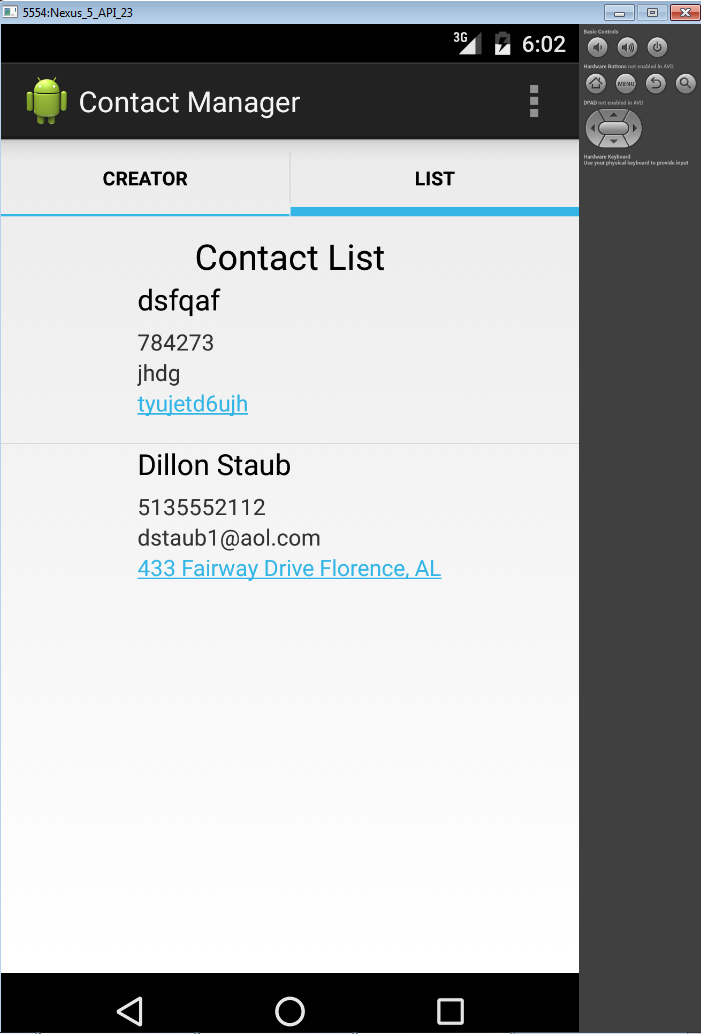
launched and the marker’s information has

been automatically filled into the

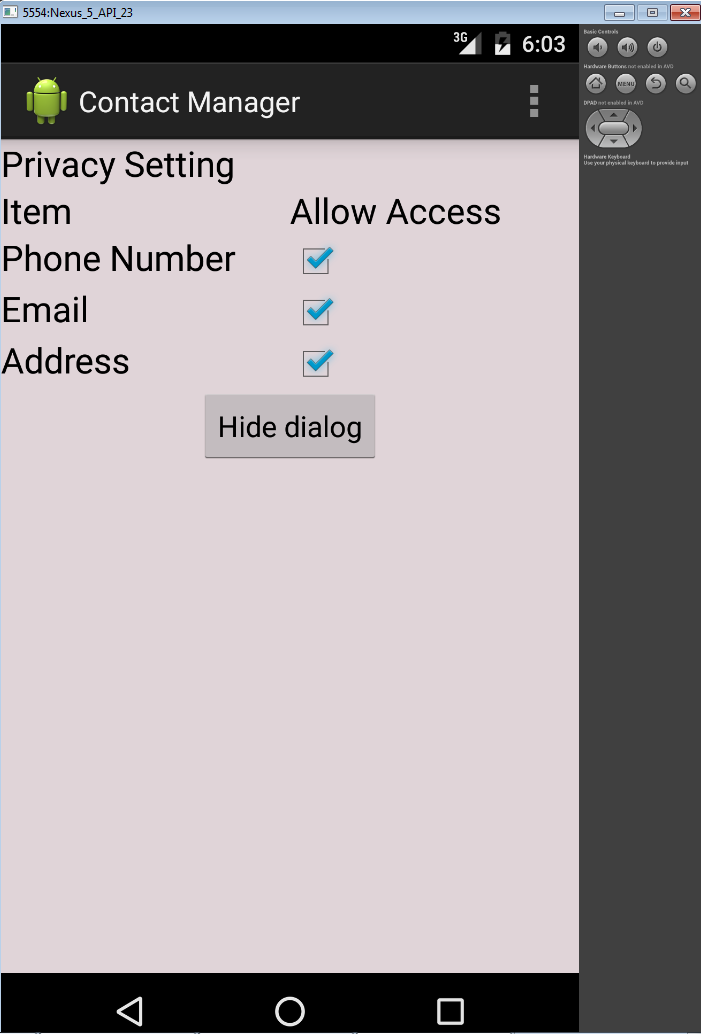
corresponding fields.

**ContactManager:**

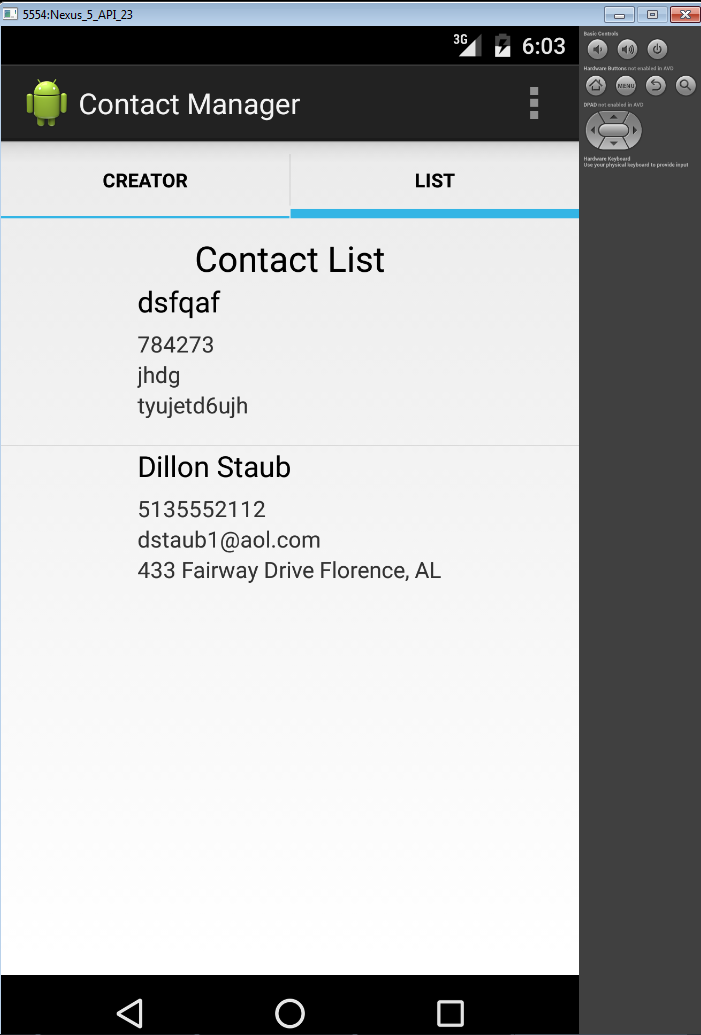
C-FEATURE1: (privacy setting): This feature was fully implemented in ContactManager. Once the ContactManager application is open, the user can input a contact with the appropriate information and save this contact. In the top right hand corner of the ContactManager application is a button with 3 vertical dots. Once the user clicks this button, a drop down menu appears with “Settings” as its only option. After “Settings” is clicked, a privacy dialog opens with check boxes as displayed in Figure 6. If the user allows access to an item (meaning the check box is checked), the item is able to be selected by the user; however if the user does not allow access to an item, the user cannot select the item in the “LIST” tab. Every address entered into the ContactManager “CREATOR” tab is saved as a hyperlink, and the user can click the hyperlink if the “Allow Access” check box corresponding to the address is checked. Once the user’s privacy setting preferences are set, the user must press the “Hide dialog” button to exit the dialog menu.



**Figure 9:** The figure above displays the “LIST” tab of ContactManager with all “Allow Access” check boxes checked in the Privacy Setting dialog.



**Figure 10:** The figure above displays the Privacy Setting dialog that appears when the user clicks “Settings”.



**Figure 7:** The figure above displays the “LIST” tab of ContactManager with all “Allow Access” check boxes unchecked in the Privacy Setting dialog.

**COMMUNICATION & COORDINATION STRATEGIES**

During the time allotted for this laboratory project, we further refined our communication strategies. Because this lab required completion on a shorter time frame than the previous two lab projects, we used the first in-lab session to plan out and coordinate the work. Much of the design work for both the data flow diagram and threat report was outlined prior to the laboratory session by group members which cut down on the in-lab design time. Team members communicated if they were going to be late to lab via text message and worked on some of the laboratory requirements beforehand in preparation. We again focused on GitHub and in-person verbal communication for the design activities because they are more efficient and translated into more effective and productive work. As in the first two labs, GitHub allowed us to push the changes and updates made to the code, and it enabled a streamlined transition between features group members worked on outside of lab time. Especially during Lab 3, we used GitHub for non-coding documents including the Visio file for the data flow diagram and the threat enumeration document. Some of our meeting time was via GitHub by reviewing the changes implemented by another team member. We also used in-person meetings during lab time because any issues could easily be discussed and understood by group members. At the beginning of both lab sessions, we held a brief scrum gathering to identify what we had accomplished and what remained to be finished. We continued this meeting structure as we had implemented it during our Lab 2 sessions.

**EXPERIENCE GAINED**

This laboratory project pushed us to examine a different aspect of software engineering as in previous labs we had focused mostly on feature implementation.

Working on the data flow diagrams in Lab 3 gave us insight into how our app interacted with external entities, data flows and stores, and where potential vulnerabilities were. None of us had made DFDs before, so working through these was a valuable experience. These diagrams provide a different perspective on the Lab feature implementation which in turn would allow us to better anticipate threats and plan accordingly if these diagrams were created prior to the feature implementation. In addition, identifying the key processes of our apps gave us a more complete knowledge of how our entire system behaved and potential room for improvement. Having worked with DFDs at this point, we are all more confident in utilizing them for future projects. The communication and coordination strategies outlined above worked better than those used in the first lab project. It was helpful that one team member had outlined the DFD prior to the first session for Lab 3 because it allowed us to work on the threat enumeration more quickly and sped up the entire lab.

We learned how to proofcheck dataflow diagrams, as well as how to take the information described in the DFD and use it in STRIDE threat enumeration. The STRIDE threat enumeration was also a new concept that we had only heard of during the lecture portion of the course, and this was the first time we applied it to a software engineering project. This enumeration also provided a new perspective on the feature implementation as it identifies areas which can be strengthened as well as potential problem areas.

**FUTURE RISKS**

1. Communication remains a big risk, especially as projects become more complex. To combat this, we will have to ensure that each member keeps the team up to date on any code or document changes and regularly integrates with everyone else.
2. Available time is also a risk. As finals approach and other projects become due, team members may not have as much free time. Thus, good time management will be paramount to success.
3. Knowledge and ability to use good software engineering principles is becoming increasingly important as our app becomes more complex. Because of this, the threat of having our code and documents become unmanageable is possible.
4. Workload balance is a recurring risk for us. It's easy to inadvertently assign too many or too few tasks to one individual, which can result in inconsistent or lower quality work.
5. External threats are becoming a bigger risk as our app becomes bigger and connects to other apps. Our threats list that we submitted with this lab outlines many of the possibilities.

**EMULATOR NOTE**

|  |
| --- |
| Emulator: |
|  |

|  |
| --- |
|  |
| Nexus 5 API 23 x86 |
|  |
|  |
| Marshmallow - Target: Android 6.0 (with Google APIs) |
|  |
|  |
| Target: Google APIs (Google, Inc) - API Level 23 |
|  |
|  |
| CPU/ABI: Googl APIs Intel Atom (x86) |
|  |
|  |
|  |
|  |
|  |
| Android Studio: |
|  |
|  |
| Compile SDK Version: API 22: Android 5.1 (Lollipop) |
|  |
|  |
| Build Tools Version: 22.0.1 |
|  |
|  |
| Source / Target Compatibility: 1.7 |
|  |
|  |
| JDK Version: 1.7.25 |
|  |
|  |
|  |
|  |
|  |
| **NOTE 1:** There appears to be an issue on some of the lab machines with the Geocoder. The method getAddressObjFromAddress() in NavigationFragment.java works on some lab computers, and fails on some lab computers. This is assumed to be due to differences in the emulators and Android SDK versions; we were not able to determine the specific root cause.  **NOTE 2:** Both Mapbox and ContactManager must be launched in the emulator for **M-Feature6** from Lab 2. |
|  |