

SOS Android Application Project Report

Submitted to:
Dr. Jeffrey S. Jones

Created by:
Dustin Dieker,
Xander Wunderlich
Xiangyu Zhang
Xu Weng
Zhiyue Yang

CSE 4471
The Ohio State University
Columbus, OH
15 April 2016

Introduction

The object of this project is to design and implement a security-related application. For the first phase of the project, the team members proposed to create either a mobile application which can help users to send an SOS message or a software project which can help users to check how much of their personal information is revealed on the internet. After the proposals were submitted and the team got feedback from the instructor, Dr. Jones, he recommended us to keep working on the first proposal and also include the database feature to store the contacts or any other necessary information. Then, our team held the first meeting and reached an agreement on developing an Android mobile application which will help a user to rapidly send an SOS text messages when he or she is in danger. More detailed application design and implementation objects will be discussed in section Application Design.

Schedule & Roles

The group had about a month to complete the SOS app. The work was divided into several tasks, including UI/UX design, text sending, location reading, database implementation, contacts storage, sending GPS, integration of functionalities, sql injection prevention and documentation. Each task was assigned to group members as shown in the table below.

Table 1: Assigned Responsibilities (Planned)

Name	Manager Of	Responsibilities
Xu Weng	Documentation	In charge of all technical documentations, including the code comments, user guide, presentation materials.
Xander Wunderlich	Design	Lead the design of the project's specific functions. And since it is design, he will also in charge of the front end interface.
Xiangyu Zhang	Implementation	Based on the design, the implementation manager will be in charge of the implementation of those functions.
Zhiyue Yang	Test	Once any function is implemented, the test manager needs to start the test process of those functions.
Dustin Dieker	Overall project	Will become the leader of this project, control the process and project quality. He will also set partial deadline for team members.

A general time and process schedule of specific task of the project is shown in the figures below:

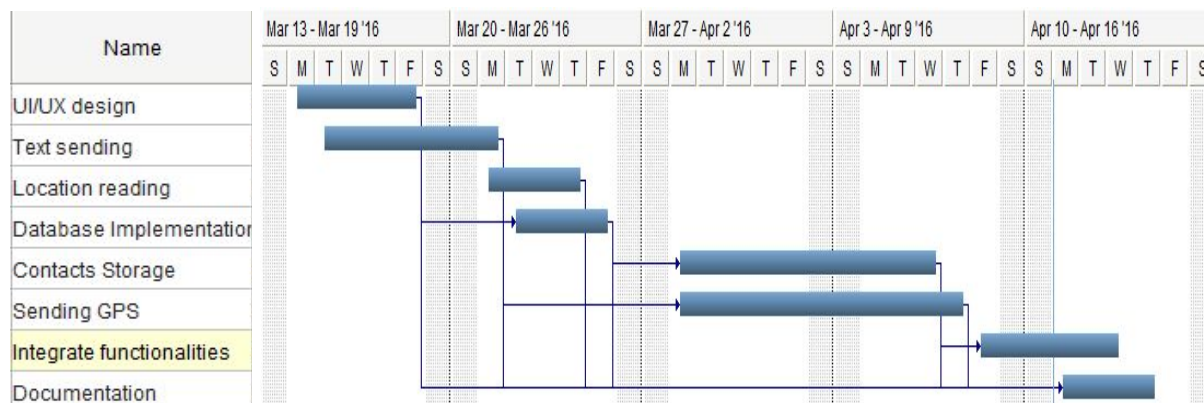


Figure 1: Timeline of Each Task

Application Design

Usage Assumption and Design Guideline

According to the team's agreement on developing an Android mobile application which will help its user to rapidly send SOS text messages when he or she is in danger, team members started to brainstorm on what specific application we will design. At first, our team assumed the following situation when the user may need the application.

Situation One:

One night, a guy was walking alone on the street. He is the only child of a very rich family. Therefore, a criminal group targeted on him and planned a kidnaping. That night, after leaving a bar at 2 a.m, he suddenly found a car rushed out from a cross and then stopped 50 feet in front of him. Some suspicious people got out from the car, he felt something wrong and turned around to escape. During that time, he opened a app and sent a SOS message to his parents and threw away his phone somewhere. With no doubt, the well prepared criminals caught him in few minutes and take him to a remote factory outside the city. Simultaneously, his parents received his message and contact the police at the first time. With the information from the message, the police quickly figured out the location of the hostage far earlier than the criminals expected and processed the assist action in time...

Situation Two:

One day, a group of girls went to a party. After drank a lot, one of them gradually waked up and found out she was stay alone with another male. It is very obvious the male was try to sex assault on the girl. The girl was trying to refuse but the male was still unconscious. In this case, she use the app and send SOS message to her friends and parents. With good luck, one of her friend did not drink and received her message and also knew where she is. As a result, her friend found the girl and saved her from the sex violence...

Based on the imaginary situations, the team concluded the user can't call or send a message in a normal way. In this case, they should be able to use our application to send an SOS message in a both convenient and precise manner. The specific solutions to resolve this contradiction will be detailed and discussed in the next section.

Function and User Experience Design

Considering the emergency message sent to the contacts, the group first thought each record in the database should store a message. In other words, users are able to customize a message for each individual contact. The was determined to be not very useful due to the fact that users are not likely to spend much time editing messages in an emergency situation. Consequently, the team decided a universal message will be sent to all emergency contacts if the SOS button is pressed. In order to avoid users accidentally pressing the SOS button and immediately issuing an emergency request, a 5 second countdown was implemented to give users opportunity to cancel the message sending. All a user needs to do is to press on the SOS button again. To further improve user experience with the app, users are able to import contacts to the app from the contacts stored on the phone. A pre-generated default message will also be provided so that users are able to use the app directly without customizing it.

Graphic Design

During the design stage of the app, several ideas regarding the UI design were brought up by the group. One idea was to implement 2 buttons on the main screen. The two buttons would be colored green and red correspondingly. The red button would be used to send the message and the other one would be used to cancel message sending. Another option was to implement 3 buttons on the main screen. The three buttons would be colored green, orange, and red, indicating the level of emergency. The group eventually ruled out both options, since while under an emergency situation, users should be able to send out a message as quickly as possible without having to think about which button to press. As a result, the team created the application graphic prototype which can be seen in figure 3.

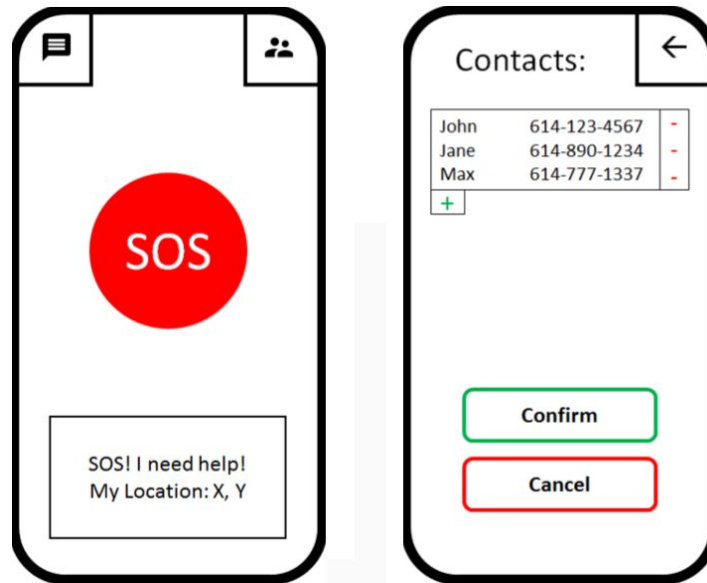


Figure 3: App Prototype

Implementation of Functions

Database

One of the important features of an app is the persistence of data. The application needs to have data stored so that when the app shuts down, the data still remains and is ready for access when the app starts again. The SOS app uses SQLite database. SQLite database was chosen because it is integrated with Android and easy to use for a small scale data usage application. The app contains one database - "SOS_DATABASE.db" and two database tables - "CONTACT_TABLE" and "USER_TABLE". The Contact table is used to store the contacts that the user will send the message to call for help. The user table stores general information that is related to the user, such as the default SOS message the user wants to send out. The database is first setup and then routes are created to perform Create, Delete, Update, and Select operations on top of the database. Once the database is setup, the views and application logic are created through Java code. The Link from the view to controller, then to the model is now complete.

Message Sending and Location Reading

Message sending is the core of the application. We decided to use the SMS Manager for Android to allow for seamless use of the user's default message provider. The message sending pulls the contacts and message directly from the database, which allow for the user to send a custom message to their emergency contacts with just one button push. In order to send useful information like the user's location, Android location manager is used to get the latitude and longitude.

Encryption(beyond original Design)

Encryption uses symmetric keys to encode and decode data before inserting and after selecting data from the database. The encryption uses Caesar Cipher algorithm that shifts each character a specified amount. (suggested Encryption key: "Ohio") Note that once a contact is inserted into the database, it can only be read correctly by using the same encryption key. If a different encryption key is used, a seemingly wrong contact will be pulled from the database.

Sql Injection Prevention(beyond original Design)

To prevent possible sql injection attack on the app database, the group implemented regular expression for input fields. This will prevent illegal input and thus avoid a sql injection attack.

Testing

Testing for the Android environment is difficult because of the large variety of Android devices and operating systems versions. The Android emulator provides a consistent Android device but it lacks the capability of a full fledged phone (notably it cannot emulate text messaging easily). Due to the emulator's limitations most of the testing is done manually by first coming up with a use case and walk through the use case on the phone to see if anything goes wrong. Therefore, the testing is time consuming and labor intensive.

Project Result

Application

The application itself successfully meets all original designs and also has the sql injection prevention and database encryption function added. Figure 4 shows the final graphic design. The user can push the red button to send the message shown above the red line. User can also modify the message if he or she want to add more information. Using the toolbar, the user can edit the contacts he or she want to send the SOS message to as well as edit the default message stored in the database.

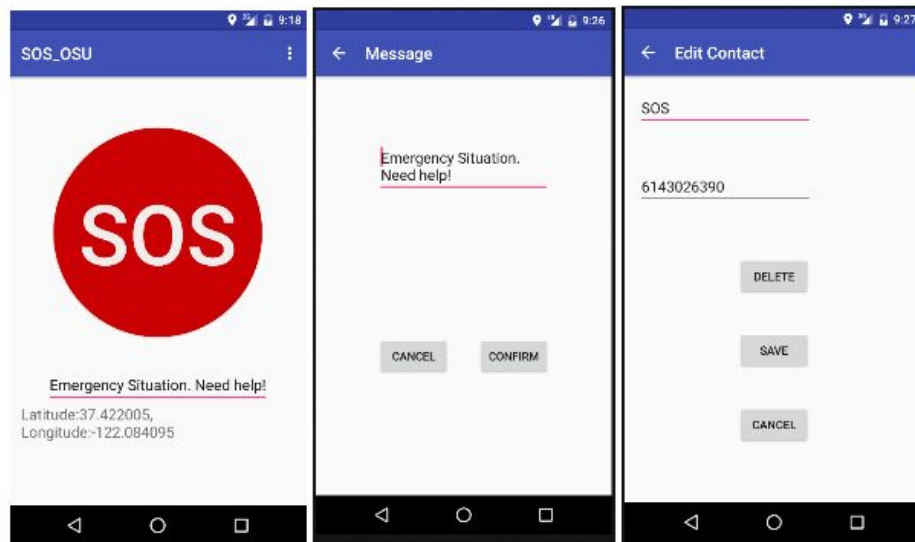


Figure 4: Final Graphic Design

Management

On the global scale, however, our management system didn't work well. The reason being, managers can't fully understand their responsibilities and give proper jobs to other team members. In this case, team member actually chose their own tasks and followed the Task Assignment Sheet detailed in the table below.

Table 2: Task Assignment Sheet

Task	Weight	Contributors
Function design	15%	All Members
UI/UX design	10%	Dustin
UI Implementation	10%	Xiangyu,Xander,Xu
Text Sending	10%	Xiangyu,Xander,Dustin
Location Reading	10%	Xander,Xiangyu
Database Design	15%	Xu,Zhiyue
Database Implementation	10%	Xu
Database Encryption	5%	Xu
Sql injection Prevention	5%	Xiangyu, Zhiyue
Test and Documentation	20%	All Members

Conclusion

The SOS App was a challenging and exciting project. Knowledge about Android development and info security concepts were gained throughout working on the project. Working well as a team was crucial to the success of the project and our weekly meetings on Monday allowed us to discuss the potential issues with the project and the steps that could be taken to solve them. Furthermore, the team was able to discuss if the current progress matched the guidelines provided and adjust accordingly. This allowed us to keep continuous progress going on the project and hold each other accountable for deadlines.