|  |
| --- |
| Project Proposal |
| SafeDrop |
| TEAM # - 9  Donna Tang  Sankha Pathak  Pooja Desai |

Table of Contents

[1. Product Definition - SafeDrop 1](#_Toc361905833)

[1.1. Our Story 1](#_Toc361905834)

[1.2. Target 2](#_Toc361905835)

[1.3. Motivation 2](#_Toc361905836)

[1.4. User Benefits 3](#_Toc361905837)

[1.5. Challenges 4](#_Toc361905838)

[1.6. Assumptions 4](#_Toc361905839)

[2. Solution Approach 4](#_Toc361905840)

[2.1. Request SafeDrop 5](#_Toc361905841)

[2.2. Respond To Pick Up 6](#_Toc361905842)

[2.3. SafeTrack 6](#_Toc361905843)

[3. Product Architecture 8](#_Toc361905844)

[3.1. Architecture description 8](#_Toc361905845)

[3.2. Mock ups 10](#_Toc361905846)

[4. User Study 16](#_Toc361905847)

[1 Conclusion 19](#_Toc361905848)

Table of Figures

Figure 1 Fear to walk at night 2

Figure 2 Use cases for the SafeDrop App 5

Figure 3 Request Caption 7

Figure 4 Deployment View of SafeDrop 8

# Product Definition - SafeDrop

## Our Story

Have you ever waited alone at a bus stop, away from home on a late night? Waited for a public transport service, which never turns up at the required time? Do you feel helpless and scared to start walking? Did you ever read the news about thefts taking place late night in your area and thought you could help? If yes, then this app is targeted for you.

SafeDrop forms a group of proactive volunteers who are willing to spend a part of their time to help others in the time of need. You can register with us and request a pick up whenever you are stranded, or drop others to their safe place as per your convenience.

## Target

*Target User Profile:* Our target user profile demographic comprises of anyone travelling alone at night; who are stranded due to unavailability of the public transport or doesn’t feel secure to use them at that hour. We assume users have a GPS enabled smart phone with a data plan for connectivity and they are familiar with the basic use of phone and apps. We would be following user interface development guidelines from Apple for the iOS platform and would be trying to mock the developer guides presented in http://developer.android.com/design/index.html

Target OS: Android ICS (version – 4.1.1), and iOS 6 (version – 6.1.1)

Target Device: HTC One S, iPhone 5

## Motivation

In the developing countries, transportation system or emergency services are not very efficient, there have been several cases where the emergency services respond much later than when the emergency has actually happened ~ implying loss of lives, fear of these events happening or even being skeptical of living lives as the needs demand.

In case of such emergencies, it often takes a lot of time to reach the location due to traffic conditions, or merely because of the longer distance, improper road infrastructure built, or immature traffic rules. On these lines there have been many incidents, which have taking place at night in several such countries in which the victims are often the one waiting for a bus or a cab to catch.

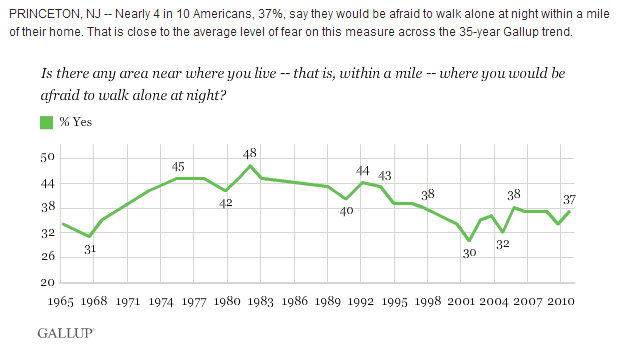


Figure 1 Fear to walk at night

(Image Reference \_ <http://www.gallup.com/poll/144272/nearly-americans-fear-walking-alone-night.aspx>)

In future, however, as the transport systems and emergency services for these countries are improving, we don’t want to rely only on the government or external agencies only to serve us better in the time of need. We do have statistics which make us believe that even with these services in developed countries people do still face the same issue. We believe we need to be the cause of the change to see the change. We want to form a network of volunteers who are willing to help in different areas. People who need help should be able to reach them easily. We primary aims in design are to promote minimum waiting – and of course safety.

## User Benefits

1. People requesting a drop would be able to get to the safe place, and prevent unfavorable incidents.
2. People willing to help and be a necessary agent of change would be able to step up and follow a system process to help others.
3. The person who is waiting gets a sense of where exactly the help is and how long the volunteer is going to take to reach the place.
4. Volunteers receive ratings and can later receive monetary compensation from the sale proceeds of this application if this strategic business model finds funds for initial jumpstart.
5. How do we minimize network usage in order to avoid draining users’ battery?

## Challenges

Here are the technical challenges we may encounter while building the app:

1. How do we extract ZIP code from the location?
2. How does the system report the activities (non-repudiation) if this fail or an incident does happen in-spite of this?
3. How do we handle concurrent users and avoid confusion?
4. How does the system choose between a person who has helped quite many people recently but is still preferred this time again?

Below is the list of business challenges we are concerned about:

1. Do we allow everybody to register in the system? Do we need to perform any background checks to check the past records of the person?
2. How do we make sure person is safe while he is travelling? Is it sufficient to just track the location?
3. Do we support automatic intimation of the emergency contact on no responses from the requester? When do we do that?

## Assumptions

Here are few assumptions we made to define scope of our application.

1. This application is intended only for a single user. If more than one users need assistance, they need to initiate the request separately.
2. Each volunteer can pick up only one person at a time.
3. We assume users are pre-registered and logged in the application.
4. We do not have any pre-defined roles in the system. Any registered user can be a requester or a volunteer. However, he can be only a requester or a volunteer for a single request.
5. Users will allow applications to use their location services. In case they deny, they will not be allowed to use this app.

# Solution Approach

We have scoped the assignment in order to include following use cases. We would use location based services in iOS and android phones in order to achieve this.

1. Request SafeDrop
2. Respond to Pickup
3. SafeTrack
4. Reviews (Elicited as secondary UC)

Here is the use case diagram exploring all the use cases in the system followed by the description.

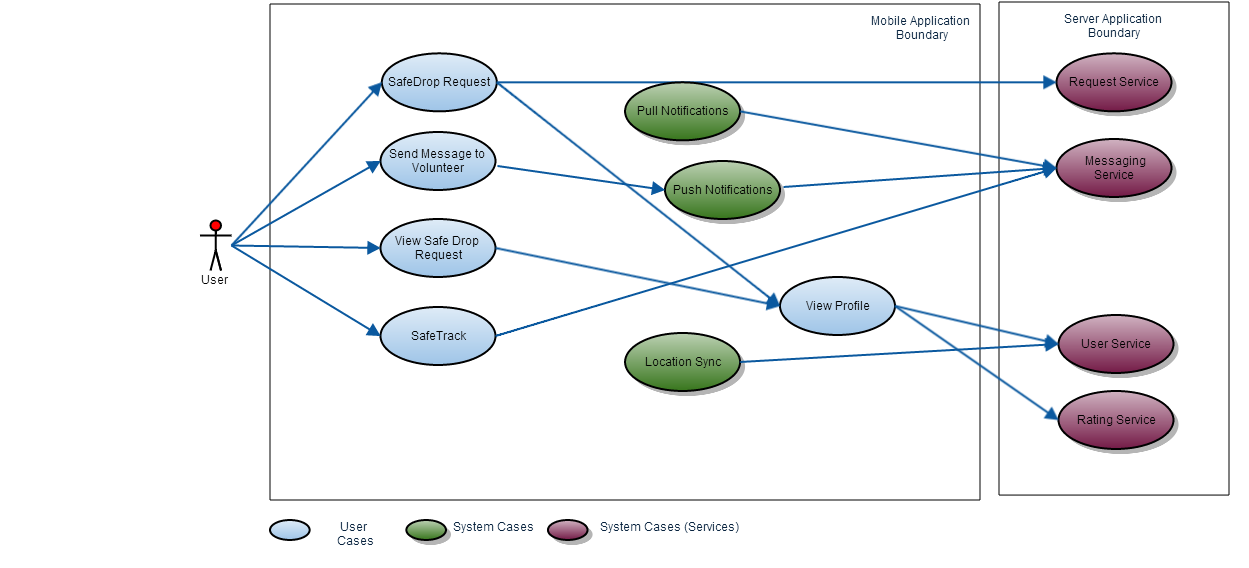


Figure 2 Use cases for the SafeDrop App

In order to achieve these functionalities end to end, we have a final list including some additional use cases in our application as below:

1. User Registration (One Time)
2. Login (One Time)
3. Request SafeDrop
4. Respond to Pickup
5. SafeTrack
6. Rating the service

These could be the important ‘nice to have’ use cases in the longer run (for future monetization)

1. Blacklist check while registration
2. Minimum fuel + monitory charges by the requestor
3. Integration with the 911 service and emergency contacts

# Request SafeDrop

User opens the application. It opens “SafeDrop Request” screen. The screen shows a map of the current location and on clicking on Request sends a request to the server to enqueue the request

**Technical Details**

Enqueue Request

Message Type = “SD\_REQ”

User ID = #userid of requester

Current Location = #lat, #long

Timestamp

Server Actions

* Reads the request, assigns it an ID
* Creates and puts broadcast request for users in the same zip code (gets zip code from #lat, #long in Current Location
* Broadcast Message Type = “NOT\_SD\_REQ”

# Respond To Pick Up

User opens the application. Notifications Icon from where the “Notification” screen is invoked appears with badges and clicks on the “Notifications” icon along with the location map and either a request status or Request button. Screen shows a list of notifications, shows the one which has not been read.

Here are the different types of notifications:

* SafeDrop Request - When a volunteer receives a broadcast from the server – (Opens “View Safe Drop Request” Screen)

This screen would have the following things to view.

* Map showing the current location of the requester
* View Profile button to view information of the requester
* Accept SafeDrop Request or Deny SafeDrop Request
* Volunteer Accepted – When a volunteer accepts a SafeDrop request, Received by Requester (Opens “View Profile” Screen)
* Volunteer Selected – When a volunteer has been chosen to continue with the SafeDrop by the Requester, Received by Volunteer as a pop up. On clicking it opens location page mentioning location of the volunteer and the requestor along with message tab.

# SafeTrack

On Acceptance of a SafeDrop Request the requester would receive a notification of Volunteer Accepted, there would be multiple volunteers who would accept, based on which the requester can “View Profile” of the volunteer.

For the volunteer, the status of the SafeDrop is marked Pending for Approval, If marked Accept by the requester, enables the SafeTrack screen. If status of the SafeDrop is marked Ignore by the requester, the the request is no more visible on notifications of the user. On Denial of a SafeDrop Request the request is no more visible on notifications of the user.

SafeDrop Screen:

SafeTrack shows live status of the volunteer and the requester. It shows a map with pins for volunteer and requester.

The volunteer/requester can

* Call the requester
* Send predefined info text like – I have left my place
* Send predefined info text like – I have reached
* Send predefined info text like – I would be reaching in 5 minutes
* Send predefined info text like – Stuck in Traffic would reach in 5 minutes
* Send custom message

All these appear in Messages Screen available from SafeTrack screen.

We have displayed life cycle of the request here. Request can be envisioned as a Ticket in the system – it goes through these stages = New, Pending, Accepted, In Progress, Closed, Archived.

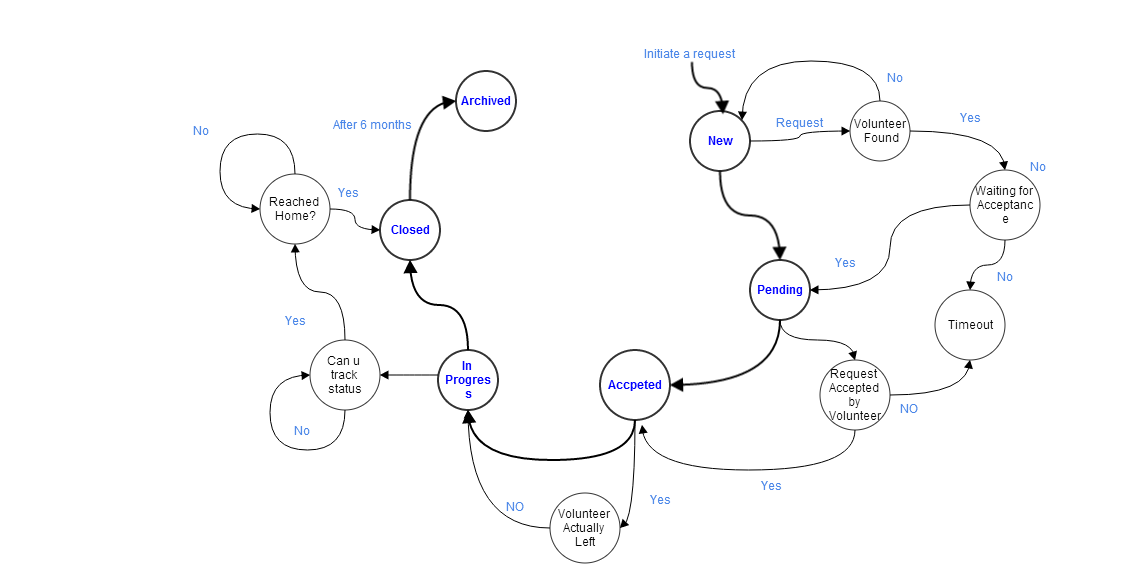


Figure 3 Request Sate Chart

# Product Architecture

# Architecture description

This is the end to end solution depicting the below diagram. When the requester initiates the request, it first hits base transceiver station. It then goes to internet service provider’s gateway. We use REST services exposed by web servers through the internet. The business logic resides in the application server. It accesses Database server to get the user details via JDBC connection.



Figure 4 Deployment View of SafeDrop

These are the set of frameworks and libraries we want to use for different functionalities.

**For iPhone Devices:**

|  |  |  |
| --- | --- | --- |
| *Functionality* | *Library Chosen* | *Rationale* |
| Location based services | CoreLocation.framework, Google Maps SDK or MapKit | Choice of Mapkit or GoogleMaps SDK would be based on initial Proof of Concept. |
| Restful Web Client | AFNetworking or NSURLConnection | AFNetworking follows REST |
| JSON Parser | Automatic with AFNetworking | NA |

**For Android Devices:**

|  |  |  |
| --- | --- | --- |
| *Functionality* | *Library/framework Chosen* | *Rationale* |
| Location based services | com.google.android.gms.location | NA |
| Restful Web Client | HTTPClient from Apache Commons Http | NA |
| JSON Parser | JSONData and JSONCore API | NA |

**At server End:**

|  |  |  |
| --- | --- | --- |
| *Functionality* | *Library Chosen* | *Rationale* |
| Database | MySQL or NOSQL | MySQL was chosen due to team expertise and time constraints. Unavailability of Joins in NOSQL (Apache Couchdb) makes it tough moreover we do not have much hierarchical search scenarios with variable datasets |
| Connectivity with database | JDBC or JDBC-ODBC Bridge | JDBC as backend server would be Glassfish which does not require Type 3 JDBC ODBC Bridges |
| Webservies | JSON over HTTP or XML over SOAP | Message Envelop size would be 30% less for JSON over HTTP. Security is not much of a concern would be implemented over a security certificate at port 443 on the Web server |
| Restful web services | Jersey API or Apache Wink | Apache Wink is still under incubation. Jersey is chosen over Wink because of its inherent support with Glassfish. |
| Application Server | Glassfish or JBOSS | Team has previous experience with Glassfish and Jersey has now become a standard for REST implementation which is built with Glassfish optimizations |

# Mock ups

Here are the **initial** mockups we came up with for SafeDrop –

|  |  |
| --- | --- |
|  | 1. Screen 1 - User logs in the system using username and password. We assume he is pre-registered. |
|  | 1. Screen 2 – As a home page, he can see his location along with four tabs which allow him to send the request, track the location, view the notification or change app settings. |
|  | 1. Screen 3 – User clicks on ‘Send Request’ and it initiates a request. Once the use clicks ‘Confirm’ on this screen, it is sent to volunteers in the nearby area. |
|  | 1. Screen 4 – User can now click on the notifications icon and he will be taken to this screen. Here he can see different types of notifications. |
|  | 1. Screen 5 – Here he can see the list of nearby volunteers. |
|  | 1. Screen 6 - User can select any volunteer and see his profile. His profile will include things like rating, volunteer’s email id, cellphone number, and other information. |
|  | 1. Screen 7 – He can select the volunteer and will be asked for confirmation. |
|  | 1. Screen 8 - Volunteers will see this screen. He will see received requests by different requesters. |
|  | 1. Screen 9 – Once volunteer selects requesters, he can then view their profile, along with the current location and buttons to accept or deny the request. |
|  | 1. Screen 10 – A volunteer will able to see details like his rating, email, cellphone number and others. |
|  | 1. Screen 11 – Once both the parties accept the request, requester will see a screen with the current location of volunteer, with two buttons – call and send message. |

# User Study

After going through the material provided in the class about user studies, we conducted user study of the app with the print outs of above screenshots.

We have consolidated different points discussed with both the users, and noted their views. Please find the highlights below and our responses.

1. On the login screen (Screen-1), I don’t see any provision to get my password back – like ‘Forgot password?’ what shall I do if I cannot recall my password?

[Team Response – Good point, forgot button will be present in the full-fledged version of the application.]

1. (Screen-1) Usually I like to see who created this app, and information about the vendor. How do I see it here?

[Team Response - We will definitely have ‘Contact Us’ option on the main screen in the full-fledged version of the application. Thank you!]

1. (Screen-2) What is the blue triangle near location? What happens when I click it?

[Team Response – We might have to remove it and see what can be done instead of that]

1. (Screen-2 ) What are the three hyphens for on the left hand side? What does it do?

[Team Response - oh, we have settings and this icon – We will remove one, thanks!]

1. (Screen – 4) I don’t want to click again to see different requests under these types – I would like to see it on the same page. At least the highlights.

[Team Response – Sure]

1. (Screen – 6) I am not sure what shall I do with the email info or other info at this point. I strongly feel rating and phone number would be sufficient. Also, I think I want to see rating on the previous page. IT would be annoying to see everybody’s profile to check how reliable they are.

[Team Response – Good point, we will change that! ]

1. (Screen – 8) When I see received requests, I would also like to know how much time I need to reach there. That will help me choose the requester. It would be great if I can see it on this page itself.

[Team Response – great point, we will take it]

1. (Screen – 9 ) I am confused, whose location is this? Mine / requester’s? It would be nice if I can see both of us in the map.

[Team Response – That makes sense, thanks!]

1. In general, I find tabs in the bottom confusing. What does safe drop do? What kind of settings can I change?

[Team Response – Oh ok, got your point. We will re-think over the layout]

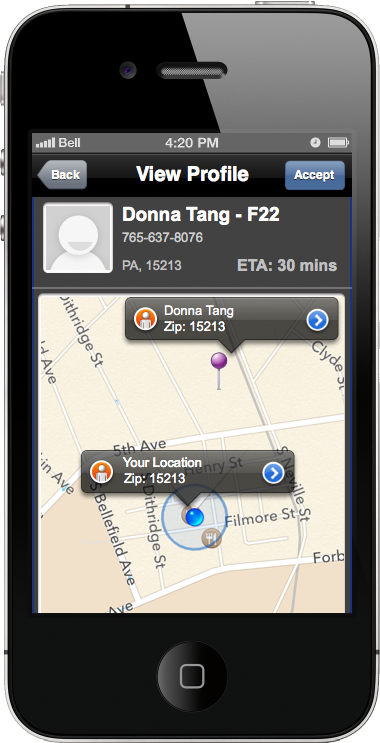
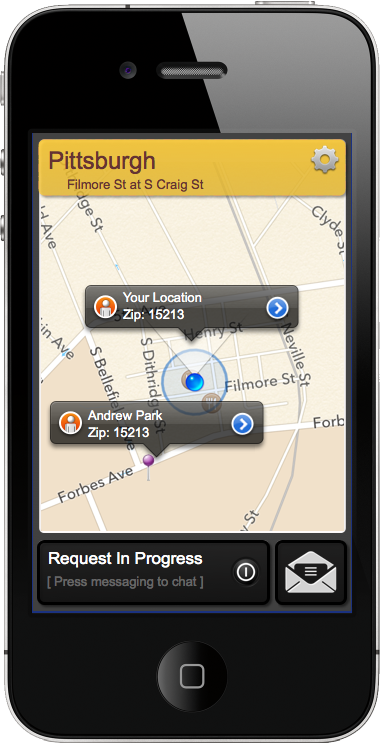
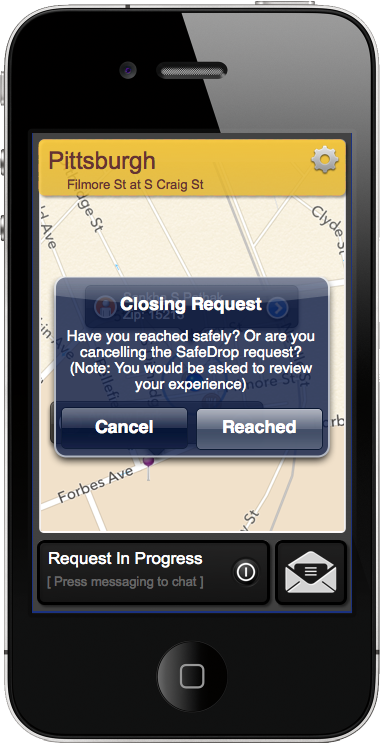
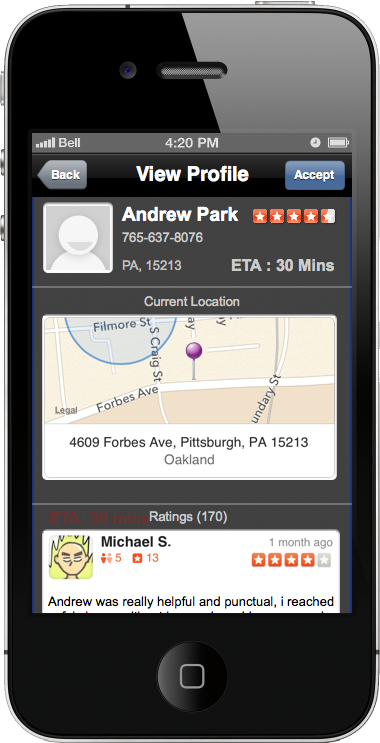
1. Can you tell me how long do I need to wait for somebody to respond to my pick up request? How do I know if my connection is lost – or you are processing my request?

[Team Response – Okay, we will show the request status along]

1. Once the volunteer leaves, how do I know where is he?

[Team Response – makes sense, we will add his current location]

Based on this discussion with both the users, we now changed our screens. Some of the new mockups look like this:



For the entire list of mockups separated by each scenario and type refer this archive



Request life cycle and navigation flow is not changed; however, we made significant changes in the UI. Here is the list -

1. Overall, we reduced number of screens.
2. We changed the layout from tabbed view to a flow view. User can now see his location, option to send request and notifications on his home screen.
3. Once user sends out the request, now we are displaying the status of the request. Viz. – locating you, searching nearby volunteers, waiting for confirmation etc.
4. User can now see number of notifications on the home screen.
5. On the list of volunteers, he can now see a tick-mark when someone accepts his request.
6. If he says ‘view profile’ now ratings of the volunteer are clearly visible.
7. Once a volunteer accepts request, now we can see location of both volunteer and a requester on our screen.
8. Now requester can also see the current location of the volunteer, as he moves.
9. We have also provided messaging screen, in case requester does not have SMS plan, he can use this.
10. Requesters can now also see established time of arrival for volunteers.

# Conclusion

Mobile application development is much different than web application development and that was evident during the brainstorming sessions. Concerns regarding battery usage, network usage, un-responsive users, adding timeouts, power drains, using data plans instead of messaging etc. were the concerns we had to address too. We hope that this we would see value of the app with larger number of volunteers. We plan to have marketing campaign to get more people involved in areas having lesser volunteers using the data statistics. We hope to have considered all possible scenarios to prevent misdirection and have SafeDrop achieve its vision.