Product Backlog

Anant Goel, Don Phan, Jon Egeland, Joshua Selbo, Levi Heilman, Sean McCullough

Problem Statement

There are many ways for a person to get home late at night such as walking, calling a taxi, or using a ridesharing app like Uber, but these are all inconvenient or dangerous in some way. Our Skunk app makes this situation safer by making it easy for someone to share their location with a trusted friend who can track their location and drive them home.

Background

It can be difficult to find safe transportation home at night, especially at times or in places where public transportation isn't available. Ridesharing apps like Uber and Lyft have emerged to make transportation quick, easy, and accessible, and location sharing apps like Glympse and Companion allow users to share their location with friends which improves safety. However, ridesharing apps cost money and present a degree of danger because they require a rider to get into a stranger's car; furthermore, verifying that a driver is who they say they are is not foolproof and can sometimes result in tragic accidents. In addition, location sharing apps are simple and do not allow for a user to arrange a ride home with a friend.

Skunk provides the functionality of existing ridesharing and location sharing apps and combines them with an accessible, easy to use interface.

Development Environment

The codebase for all parts of the application will be maintained within a single Git repository; however, the application will be made up and developed as two distinct entities:

1. **Client:** An app targeting iOS 9 developed on OS X using Xcode and the Swift language.
2. **Server:** A Ruby on Rails app, running on Ubuntu backed by a MariaDB (MySQL-like) database.

The git repository will be hosted privately on Github, which we will also use to track the progress of use cases. In addition, we will utilize Github's integration with TravisCI to execute test cases automatically as a post-commit hook and ensure that all tests pass before development can continue.

Functional Requirements

Notation

The **sharer** is a user of the app who intends to share his or her location with a trusted friend, the **receiver**.
The **receiver** is a user of the app who intends to track the location of the **sharer.**

Backlog ID	Requirement	Est. Hours
1	As a sharer, I want to share my location with people I choose from my contacts list (receivers).	4
2	As a sharer, I want to be able to choose whether my location is shared for a set period of time, or until I reach a location that I select.	4
3	As a sharer, I want to specify if I need to be picked up or not when I begin to share my location.	1
4	As a receiver, I want to receive a notification when a sharer shares their location with me.	1
5	As a receiver, I want to be able to view the real-time location of each friend who has shared their location with me.	5
6	As a receiver, I want to have the option to stop receiving updates from a sharer.	4
7	As a sharer, I want to request to stop sharing my location.	2
8	As a receiver, I want to approve or reject a sharer's request to stop sharing his or her location.	2
9	As a sharer, I want to alert my receivers that I am ready to be picked up.	2
10	As a receiver, when I accept a pick-up request, I want to provide the sharer an estimated time of arrival.	3
11	As a receiver, I want to receive a notification when my friend has arrived to their location safely.	3
12	As a receiver, I want to know how far away a sharer is from my current location.	2
13	As a receiver, I want to get directions to a sharer's location.	3

14	As a user, I want to have the option of being both a sharer and a receiver.	1
15	As a sharer, I want to view all friends I have shared my location with.	3
16	As a user, I want to be able to access views for both the sharer and receiver role.	4
17	As a receiver, I want to be able to see if a sharer who wants to be picked up has already found a driver.	3
18	As a receiver, I want to be able to see the locations of all users who are sharing their location with me on a single map.	3
	Total:	50 hours

Non-Functional Requirements

Appearance

The app should be easy to use and appeal to the target audience in an intuitive manner. Since the target audience ranges from kids all the way to adults we want to make the application minimalistic so both audiences can pick up on how to use the application quickly and effectively. The color scheme should be friendly and simple so users feel happy and safe while using the app.

Privacy

The app should not allow external users to view locations of users who should have the privilege to do so. The app should not allow anyone who the sharer has not explicitly shared their location with to view it.

Reliability

The app should gracefully handle potential crashes and ensure that the user is not put in any danger because of an app failure. This includes crashes, GPS failure, false reporting of location, app response if phone dies, and possible early termination of location sharing.

Security

The app should interface with the server using encrypted communication to ensure that users are not susceptible to packet sniffing. This ensures that the sharer's location, which is sensitive information, is not intercepted by a malicious third party.

Performance

The goal for performance is that the location tracking system provides a minimum of one location update per 60 seconds. This ensures a fluid and comforting experience for the receiver to keep up to date on the sharer's location.

Use cases

Case: [Sharer] Share my location for X hours without requesting a driver
Related requirements: 1, 2, 3, 4

User 1 (Sharer)	Sharer App	Server	Receiver App	User 2 (Receiver)
1. On the Sharer view, select "Share my location"	2. Present view to select a contact who has the app, optionally request a driver, and share for a length of time or until arriving at a location			
3. Select contact(s), share for X hours, and submit	4. Submit request to server	5. Handle incoming request, notify the requested receiver(s), and begin tracking location	6. Receive notification "User 1 has shared his/her location with you for X hours"	

Case: [Sharer] Continually update my location
Related requirements: 5

User 1 (Sharer)	Sharer App	Server	Receiver App	User 2 (Receiver)
(no action necessary)	1. In background, send location update to server every ~30 seconds	2. Update last received location		

Case: [Receiver] Monitor a single sharer's location
Related requirements: 5, 12

User 1 (Sharer)	Sharer App	Server	Receiver App	User 2 (Receiver)
		3. Send the sharer's location to receiver	2. Request server to send updated location of the selected sharer	1. On the Receiver view, select a current sharer
			4. Update map with new location and display that the sharer is X miles away	

Case: [Receiver] Monitor all sharer's locations
Related requirements: 5, 18

User 1 (Sharer)	Sharer App	Server	Receiver App	User 2 (Receiver)
		3. Send sharer locations to receiver.	2. Request server to send updated locations of all sharers.	1. On the Receiver view, select the map that shows all sharers.
			4. Update map with new locations.	

Case: [Sharer] Share my location for X hours and request a driver
Related requirements: 1, 2, 3, 4

User 1 (Sharer)	Sharer App	Server	Receiver App	User 2 (Receiver)
1. On the Sharer view, select "Share my location"	2. Present view to select a contact who has the app, optionally request a driver, and share for a length of time or until arriving at a			

	location			
3. Select contact(s), share for X hours, mark that I want to be picked up, and submit	4. Submit request to server	5. Handle incoming request, notify the requested receiver(s), and begin tracking location	6. Receive notification “User 1 has shared his/her location with you for X hours and will need a ride”	

Case: [Sharer] Share my location until I arrive home
Related requirements: 1, 2, 3, 4

User 1 (Sharer)	Sharer App	Server	Receiver App	User 2 (Receiver)
1. On the Sharer view, select “Share my location”	2. Present view to select a contact who has the app, optionally request a driver, and share for a length of time or until arriving at a location			
3. Select contact(s), select a safe location to indicate when I want to stop sharing, and submit	4. Submit request to server	5. Handle incoming request, notify the requested receiver(s), and begin tracking location	6. Receive notification “User 1 has shared his/her location with you until he/she is home”	

Case: [Sharer] Alert for Pick-up
Related requirements: 5, 9, 17

User 1 (Sharer)	Sharer App	Server	Receiver App	User 2 (Receiver)
1. On the Sharer view, select “I need to be picked up”	2. Application sends notification to server and alerts user that pick-up notifications have been sent.	3. Handle incoming alerts and send notifications to receivers.	4. Creates a notification when a pick-up alert is received.	5. Tap notification to open app.

Case: [Receiver] Accept Pick-up Request
Related requirements: 10, 11

User 1 (Sharer)	Sharer App	Server	Receiver App	User 2 (Receiver)
			2. Ask the receiver to set an ETA on when they can pick-up.	1. Select “Yes” when asked if available to pick-up
7. Tap notification to open app.	6. Receive acceptance alert and notify sharer of the acceptance and ETA.	5. Handle incoming acceptance by notifying sharer that their pick-up request has been accepted and by notifying other receivers that the accepting receiver has accepted.	4. Submit acceptance to server.	3. Input an ETA
			8. [Other receivers] Receive notification that another receiver has accepted pick-up request.	9. Tap notification to open app.

Case: [Receiver] View the status of all friends sharing their location with me
Related requirements: 17

Server	Receiver App	User 2 (Receiver)
3. Return current information about each sharer	2. Request from server updated information about each sharer	1. Open the Receiver view
	4. Display the status of each sharer including information about the remaining time until sharing ends or the ending location, and whether or not a driver has accepted a request to drive them	

Case: [Receiver] Get Directions to Sharer
Related requirements: 13

User 1 (Sharer)	Sharer App	Server	Receiver App	User 2 (Receiver)
			2. Open map application using sharer's location as destination.	1. On the Receiver view, select "Get Directions"

Case: [Receiver] Stop receiving updates from a sharer (Single Receiver)
Related Requirements: 6

User 1 (Sharer)	Sharer App	Server	Receiver App	User 2 (Receiver)
		3. Since User 2 is the only receiver, terminate session	2. Open Target Sharer view, select "Stop receiving updates."	1. On the Receiver view, select the target Sharer.
			4. Success alert appears: "You have stopped receiving updates from User 1"	

Case: [Receiver] Stop receiving updates from a sharer (Multiple Receivers)
Related Requirements: 6

User 1 (Sharer)	Sharer App	Server	Receiver App	User 2 (Receiver)
		3. While the session includes multiple receivers, do not terminate the session	2. Open Target Sharer view, select "Stop receiving updates."	1. On the Receiver view, select the target Sharer.
			4. Success alert appears: "You have stopped receiving updates from User 1"	

Case: [Sharer] Requests to stop sharing their location
Related requirements: 7, 8

User 1 (Sharer)	Sharer App	Server	Receiver App	User 2 (Receiver)
1. On the sharer view, select "Stop sharing location"	2. Present a view asking the sharer to confirm if they want to stop sharing their location.	5. Update session to convey that the sharer has requested to stop sharing their location	6. Open specific sharer view and ask the receiver to confirm or deny the request from the sharer	7. Select "Confirm" or "Deny" if the receiver wants to stop receiving the sharers location
3. Press "Confirm" or "Deny" if the sharer wants to stop sharing their location	4. Submit request to server	9. Terminate the session if the sharer and receiver both selected "Confirm"	8. Submit request to server	
	10. Bring up a confirmation on the sharer view that the sharers location is not being shared anymore.			

Case: [User] Option of being both a sharer and a receiver
Related requirements: 14, 16

User	App
1. Open application	2. Application shows the tabs for "Sharer" and "Receiver"
3. User selects either options to view respective tabs.	4. Display respective tab

Case: [Sharer] View all friends I have shared my location with
Related requirements: 15

User 1 (Sharer)	Sharer App
1. Open application	2. Display sharer screen and display list of confirmed receivers.