**Herald!**

**Requirements Document**

**Ver. 1.0**

**Table of Contents**

1. [Application Overview](#_Application_Overview)
   1. [Objective](#_Objectives)
   2. [Business Process](#_Business_Process)
   3. [User Roles and Responsibilities](#_User_Roles_and)
   4. [Interaction with other systems](#_Interactions_with_Other)
   5. [Replacement of Legacy Systems](#_Replacement_of_Legacy)
   6. [Terminology](#_Terminology)
2. [Functional Requirements](#_Functional_Requirements)
   1. [Statement of Functionality](#_Statement_of_Functionality)
   2. [Scope](#_Scope)
   3. [Performance](#_Performance)
   4. [Usability](#_Usability)
   5. [Concurrency](#_Concurrency)
3. [Appendices](#_Appendices)
   1. [Author(s) background and expertise](#_Author’s_background_and)

# Application Overview

## Objectives

This app will be used to send text message updates of the user’s current location and/or time to the destination to contacts the user has chosen. This is useful in several scenarios:

* If you are picking someone up, it will notify them when you’re almost there so they can be ready
* If you’re taking a long trip to stay with friends/family, it will keep them updated with your whereabouts
  + In the event of an unexpected delay or emergency situation, others will not be kept in the dark

## Business Process

## User Roles and Responsibilities

Users are customers who are of driving age and own an android smartphone. The system will send automatic text updates on their travel progress to a specified receiver.

## Interactions with Other Systems

This application will interact with Google Maps to find travel time, the android texting app to send updates, and the Contacts app to source phone numbers.

## Replacement of Legacy Systems

Twist – On My Way!

* Allows users to send updates about current travel times to friends
* Latest version may require recipient to have a Twist account in order to receive meaningful information
  + Herald! will not require accounts

OnMyWay

* Basic app which lets the user input the amount of time you estimate is remaining until you reach your destination, and it will build and send a text message containing that (and optionally, your location) for you.
* Our app will calculate the ETA (estimated time to arrival) itself based on the current and destination locations.

## Terminology

# Functional Requirements

## Statement of Functionality

The user will interface with Android’s Maps features to select a destination and the Contacts feature to select recipients of the travel updates. The travel updates will consist of one or more of the following: a) the user’s location, and b) the user’s estimated time to the destination.

### Receiver-

Zolotnik will allow the user specify a user, by phone number (interfacing with Contacts), to receive the text updates.

### Start Point-

Zolotnik will always use the users current location as the start point in its communication with Google Maps.

### End Point –

Zolotnik will allow the user to specify an end location, by address; in its communication with Google Maps.

### Message Content-

Zolotnik will send a default message to the receiver.

### Update Interval-

Zolotnik will allow the user to select, from a list, the time interval between travel updates.

## Scope

All of the functionality in the above section is required for release.

## Performance

Zolotnik should correctly specify both location points in Google Maps.

## Usability

## Concurrency

# Appendices

## Author’s background and expertise

### Connor Pike

I am a 3rd year in the Computer engineering program at the University of Cincinnati. I have experience in Web App development, C#, C++, Javascript, HTML, XML and XAML. Through the co-op program I have worked for about a year as a software developer.

### Jacob Holbrook

I am in my third year at University of Cincinnati in Computer Engineering. I’ve worked with Python, C++, PHP, HTML, MySQL, C#, and others.

**Resources/APIs**

Google Maps API –

Allows you to embed Google Maps in you app. See <https://developers.google.com/maps/documentation/javascript/reference>.

Contacts API –

Android.ContactsContract defines an extensible database of contact-related information. See <http://developer.android.com/reference/android/provider/ContactsContract.html> for more detail.

Android SMS Manager – android.telephony.SmsManager