Team: Ted Lim, Scott Buttinger, Kevin Bishop, Ryan Hails, Chase Brandon
2/17/17

Trackify: The Tracking and Scheduling App

Executive Summary

Frequent travelers who deal with influxes of flight confirmation emails need a way to organize their travel plans. Trackify will address this issue by having users forward their flight confirmation emails to Trackify's hosted domain name. Trackify will then parse the important flight details from the emails and organize them in a coherent manner; for example, it will put the flight details into a calendar for the user to refer to in the future. Trackify will also allow users to manually input flight details. The user will then be reminded by Trackify when the dates of the scheduled flights are approaching.

Currently, Trackify will only be available on iOS and only compatible with a few airlines like Southwest, Jetblue, Alaskan, etc.; however, the scalability of Trackify is promising. Trackify will be able to handle any airline confirmation email and available for Android in the future. Additionally, the intellectual property of having users forward these confirmation emails to an application hosted domain name can be utilized in other areas for tracking a user's shipping and package confirmations and potentially, ticket confirmations for local enterprises.

Trackify Description

Trackify is a tracking and scheduling application for iPhone users. Trackify users are people who want organization when it comes to managing their flights. Trackify users will forward their flight confirmation emails to Trackify's email domain name, (example@trackify.biz), and find their flight details automatically placed into Trackify as flight "event items" and integrated with the iOS Calendar application. Autonomous integration into Trackify and the Calendar application will at first only support a limited number of airlines. The application will also have an option for manually entering flight details to account for flight confirmation email formats that are not supported by the application and for forwarded confirmation emails that may have erred.

As the application scales, Trackify will replicate its application flow for flights and translate it to other areas like shipping and packaging confirmation emails.

Trackify's Niche

Trackify will help reduce the stress of frequent flyers who dislike constantly updating and maintaining their own schedules as this app will do that for them automatically. Also, Trackify's reminder feature will reduce people's anxiety over important flight dates and time that they may have forgotten about.

Currently, Trackify would be catering to about 900 million U.S. flyers per day, and as Trackify expands to include people who need to manage package confirmations, usually online shoppers, another 200 million users would be included into the potential consumer base. Out of the potential 200 million additional users, Trackify would appeal to millennials who are frequently ordering packages from various sites like Amazon.

Trackify's Competitors
      Apple Passbook

            The main competitor is Apple's Passbook application. This app allows users to aggregate e-tickets for events and flights. Despite attempting to cater to the wide variety of e-tickets that are available today, Passbook suffers from the fact that many airlines use ticketing formats that are incompatible with the app. Our project will overcome this flaw by using a user's flight confirmation email to generate flight "event items" that the user can easily sort through within the application. By simply parsing out important information from the user's email, we can create "event items" that are compatible with all airline ticketing schemes.

      Slice

            Another app in a similar space is Slice, a mobile application that tracks and aggregates online shopping orders. The catch however, is that Slice requires full access to a user's email account. It needs total access so that it can scan the user's emails for specific keywords and pull out information related to shopping purchases. Our application is designed such that we do not require full access to a user's email account. By simply providing Trackify users with the Trackify hosted domain name assigning each user a unique email address to which they can forward shopping and flight specific emails, we give users peace of mind by keeping the remainder of their inbox private.

Trackify's Technical Design Overview:
      User Interface
- Account Creation
- Login
- "Event items" menu
- Manual entry of "event items"
- Settings

      Backend
- Database
- Email server

      User Interface
   **Account Creation**

                To create an account, users will provide their first name, last name, personal email, and password. This information will be transmitted to an AWS service known as DynamoDB, a NoSQL database service.

                The account creation page will use buttons, labels and text fields. The page will additionally use action listeners to validate each form. Validation requires users to properly fill out all necessary text fields. The NSRegularExpression class helps to validate email addresses.

Login

In order to login, users will input their user credentials into the login page, which will ping the Trackify database in AWS DynamoDB to ensure that they are valid users.

The login page will be very simple, consisting of a few buttons, labels and text fields. The photo to the right of this section displays a sample login page that is similar to the one we will use for our application minus the sign in with Facebook and Twitter features.
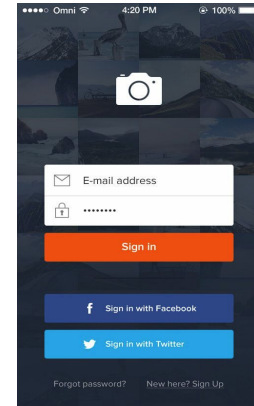
"Event items" menu

This menu is the basic view from where users will see all of their "event items" that they have input into the application.

The menu for viewing past/upcoming events will utilize either the iOS UICollectionView or the UITableView class. Trackify will have a sleek and simple UI that will allow users to scroll chronologically through their flight and order event items. The photo to the right of this section shows an application that utilizes a similar UI style.
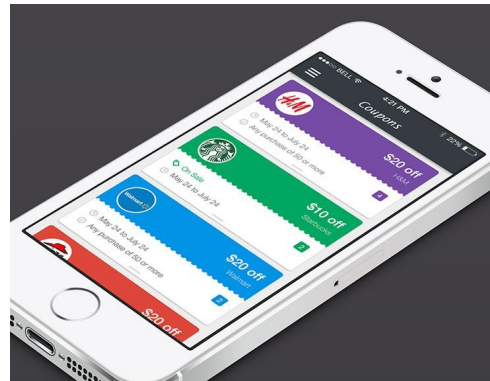
Manual entry of "event items"

Manual entry of "event items" is a page where users will be able to manually put in flight details or in the future, shopping order event item creation. This is in case of the event that a forwarded confirmation email is corrupted or does not coincide with Trackify's parsing schema.

This screen will contain a few buttons, labels, and text fields so that users can input their specific data. Once users finish creating an entry, they will be able to see that data in the "event items" menu

Settings

This is a page that will be for users to set specific preferences on the application; for example, when users would like to be notified of approaching flights.

For the settings menu, Trackify will use Swift's UITableView class and have UITableViewCells that will correspond to specific settings. Other classes

like UISlider and UISwitch will be used to handle more specific settings if necessary.

Backend

Database

Trackify will use AWS DynamoDB service. The current plan is to set up three tables in the database: one for users, one for flights, and one for packages when the application supports the package feature.

In the users table, The following information will be stored in it:
- Username (email address)(string)
- First name (string)
- Last name (string)
- Password (string)

In the flights table:
- Airline (string)
- Flight number (int)
- Date (datetime)
- Time (datetime)
- Destination (string)
- Departure (string)
- Confirmation number (string)
- Airport name (string)

In the future packages table:
- Seller name (string)
- Tracking number (string)
- Shipping Address (string)
- Expected delivery date (datetime)
- Delivery status (int or struct)

Email server

Trackify will use two more of AWS services: AWS SES and AWS Lambda. AWS SES is an email service that will host the Trackify domain name to where users will forward emails. Once in AWS SES, the received emails can be piped through the second service, AWS Lambda, which will parse the email into the necessary components as dictated by the tables in AWS DynamoDB.

Trackify's Resource Requirements

For building and developing Trackify, what is needed are the following resources:
- AWS SES
- AWS Lambda
- AWS DynamoDB
- Xcode
- iOS device (used for testing)

For users to use Trackify, they will need the following:

- iOS application
- Android application (in the future)

Trackify's Design Choices

Two design choices that we undertook were choosing to develop on iOS before Android, and a user's manual-forwarding of emails versus autonomously pulling emails from a user's email so that Trackify can schedule the desired event.

iOS vs Android

We chose iOS for mostly one reason. Apple's management of its App Store is much better than Google Play. Android applications can be installed from any source, which makes them extremely prone to app piracy. To guarantee the safety of Trackify's intellectual property as well as the returns on Trackify's downloads, Apple does a much better job in guaranteeing those insurances for developers. Although Android does have a much larger customer base than Apple, a large part of the value of Trackify is the IP. Additionally, after Trackify's completion on iOS and getting proof of Trackify's usability, it is trivial to leverage this application in Android.

Manual vs Autonomous

Trackify mainly transfers data through the user-required action of forwarding their confirmation emails to the application's hosted domain email. This could have been done through a more intrusive method such as aggressively collecting all of a user's emails and filtering by the confirmation emails, whether they are for flights or packages. By sacrificing a little work on the user end, Trackify will not intrude on a user's personal data, as many other applications do. It is important to protect the rights of Trackify users as this will garner user loyalty and trust with the application and those associated with them, which will help the Trackify grow in the future.

Assessment of Risks

Two major risks that our product may face will stem from 3 issues: intellectual property (IP) infringement, developer costs, and AWS.

For IP, Trackify may run into a situation where another company may try to sue us for IP infringement on some part of the product. This may happen due to a recent approval of a patent that occurred during the development of the product. Although this is a legitimate risk, it is one that will likely not occur due to the 'flimsiness' of software-related IP. Currently, court rulings about software IP are strict in that to actually infringe upon a software patent, the patent must pertain to IP that creates physical change or effect; in other words, patents on abstract ideas are ineligible. Therefore Trackify should be alright.

Developer costs is an issue for the future after Trackify displays its proof of concept. The suggested plan is to leverage the Trackify application to Android; however, the current developers are unfamiliar with the platform. Therefore, it likely that the team will need additional funds to hire Android-capable engineers.

Lastly, a core component of Trackify is the AWS services that it relies on to function. The performance of the application is highly dependent on the data flow through each of the AWS services that Trackify utilizes. If for some reason, AWS has an outage, hacked, or other dramatic event, Trackify will also be affected. Currently, due to limited resources, there is no current viable backup plan in this worst case scenario in order to keep Trackify operational if AWS services are down.

Next Steps:
The current plan is to complete iOS production of Trackify, which will first target frequent flyers and deal with flight confirmation emails from airlines like Southwest, Jetblue, and Alaskan Airlines. Once this proof of concept is complete and tested, it will then expand to handle other airlines. After Trackify has proven itself in the airline industry, it will move on to the shipping and packaging industry to handle packages from different shipping companies. In the future, Trackify will then be leveraged by Android to expand the applications customer base.