|  |  |  |  |
| --- | --- | --- | --- |
|  | 12/1/2014 |  | |
| |  |  | | --- | --- | |  |  | |  |  | |  |  | |  |  | | |  | |
| TD Challenge  *trakR* | | | |
|  |  | | |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |
|  |  | | Jordon de Hoog & Jonathan Decher |

TD Challenge

trakR

Winners embrace hard work. They love the discipline of it, the trade-off they're making to win. Losers, on the other hand, see it as punishment. And that's the difference.

Lou Holtz

# Introduction

As instructed, our mobile application, ‘trakR’ uses location based services to benefit the user and their finances. Sharing their location via Google Maps, the user can, ‘check in’ to nearby companies and record their spending to their desired accounts. The user can then reflect on their spending according to account, company and category.

# Features

trakR includes some remarkable features. While simple for the user, these features are made up of challenging and voluminous amounts of code.

These features include:

* + Signup & login with a username, email, and password.
  + Create accounts bounded by cash, debit, or desired credit-card.
  + Create an account with a physical card via Near Field Communications (NFC).
  + Synchronize with previous accounts.
  + Display companies proximate to the user’s location, or the user’s preferred location via Google Maps.
  + Add the user’s spending at the chosen company to the user’s desired account.
  + Display all check in’s via a list-view sorted by date.
  + Present spending according to account.
  + Demonstrate spending according to category.
  + A pie graph rendering category spending.

# Why we should win

Despite the fact that we fulfilled the requests of the challenge, we did in fact go above and beyond what was asked for. Using structures that were taught in class, we also took it upon ourselves to include features that were not taught in class. We had fun researching different designs and picking one that suits, and we feel the final design is slick and easy-to-use for the user. We should win the TD Challenge because our app impressively encourages the user to share their location, while giving a memorable experience that will make the user continuously use the app again, and again. Even if we don’t win, this challenge has been an unforgettable learning experience and we were happy to be a part of it.

# Technical Information

* Minimum SDK API level 19 Kit Kat
  + trakR uses some API 19+ features, such as translucent navigation bar and status bar
* Implements the following ‘advanced’ UI components
  + View pager – for handling of the fragment tabs.
  + Card view/Card List – used heavily in Google’s new Material Design.
  + Custom List-views
  + NFC system Intents
* Simulates “synchronizing” accounts with external sources.
* Custom ‘drawables’ for many of the UI components, such as buttons and backgrounds.

# Open Source Libraries

trakR implements many various third party open sourced libraries. We feel implementing these libraries has improved the quality of our application substantially. Listed below are some of the used libraries, the rest can be found in the ‘build.gradle’ file of the application.

* [SugarORM](https://github.com/satyan/sugar) – Simplify the interactions between the SQLite database and the application.
* [ButterKnife](https://github.com/JakeWharton/butterknife) – Reduce the amount of code needed to write for the injection and use of UI components.
* [PagerSlidingTabStrip](https://github.com/astuetz/PagerSlidingTabStrip) – Relatively easy to implement view pager with tab indicators.
* [CardsLib](https://github.com/gabrielemariotti/cardslib) – Used for the creation of custom Google Cards.
* [GSON](https://code.google.com/p/google-gson/) - Created by Google, to simplify the use of JSON and Java models.
* [Ion](https://github.com/koush/ion) – An amazingly easy asynchronous library for network and image loading.
* [EMV Paycard](https://github.com/devnied/EMV-NFC-Paycard-Enrollment) – Facilitate the reading of NFC enabled credit cards.

# Miscellaneous

The following is also used heavily by trakR, and was used in the development of trakR.

* Google Play Services – The use of the Map functionality of google services.
* Places API – Googles public API for the querying of map related data.
* [FlatIcon](http://www.flaticon.com/) – database of free vector icons.
* [Android Asset Studio](http://romannurik.github.io/AndroidAssetStudio/) – for the conversion of the above icons to android drawables.
* Google & Stack Overflow – without which, many questions would have gone unanswered.