|  |  |  |
| --- | --- | --- |
|  |  | FIT3027: Android IOS  Assignment 1 |

30/4/2017

Mobile Application Design Sepcification

**ForkMe**

[[1]](#footnote-1)

David Lei

Student ID: 26029391

Table of Contents

1. Application Concept 4

1.1. Introduction 4

1.2. Application functionality 5

1.2.1. Trending now (GitTrends) 5

1.2.1.1. View the author 5

1.2.1.2. View a repository 6

1.2.1.3. Star a repository 6

1.2.1.4. Watch a repository 6

1.2.2. Find developers (MergeMe) 6

1.2.2.1. Sharing information 6

1.2.2.2. Natural Language Processing Approaches 7

1.3. Target Audience 7

2. Idea Innovation 7

2.1.1. Why is this innovative? 7

2.1.2. Does it already exist? 8

2.1.3. But GitHub uses a web interface? 9

2.1.4. Is it useful? 9

3. Techonology Considerations 10

3.1. Android 10

3.1.1. Transferring data - Protobuffers 11

3.1.1.1. Transfer format 11

3.1.1.1.1. JSON 11

3.1.1.1.2. Protocol Buffers 11

3.1.1.2. Web Library – Volley 12

3.1.1.3. Local Storage - SQLite 13

3.2. iOS 13

3.2.1. Transferring Data – JSON 13

3.2.1.1. Transfer format 13

3.2.1.1.1. JSON 13

3.2.1.1.2. Protocol buffers 14

3.2.1.2. Web Library - Alamofire 14

3.2.1.3. Local Storage – CoreData 14

3.3. Mobile Platform Independent 14

3.3.1. GitHub API 14

3.3.2. Webserver (backend) 15

3.3.2.1. Language/Framework 15

3.3.2.1.1. Go/HTTP 16

3.3.2.1.2. JavaScript/Node.js 16

3.3.2.1.3. Python/Flask 16

3.3.2.2. Deployment/Hosting – Google App Engine 17

3.3.2.2.1. Heroku 17

3.3.2.2.2. Google App Engine/Cloud platform 17

3.3.2.3. Server Side Storage 17

3.3.2.3.1. PostgreSQL 17

3.3.2.3.2. Firebase 18

4. Interface Design Story Board Mockups 18

4.1. Splash Screen 20

4.2. Log in 20

4.3. Trending Page (GitTrends) 21

4.4. Repository View 21

4.5. User View 22

4.6. Organization View 22

4.7. Starred Repository View 23

4.8. Hamburger (side bar) 24

4.9. Find Developers (MergeMe) 25

4.10. Contact a developer 25

5. Scope and Limitations 26

5.1. Scope 26

5.2. Minimum Viable Product 27

5.3. Limitations 27

5.4. Stretch Goals 27

6. Estimated Project timeline 27

7. Testing 27

# Application Concept

## Introduction

ForkMe is a mobile application that aims to raise awareness of trending open source projects and help foster a community of innovation. It does this by having 3 key functional aspects.

1. GitTrend (primary functionality, high priority)

* The ability for a user to browse through popular open source repositories currently trending on Github[[2]](#footnote-2).
* Allowing users to star a repository (like a Facebook like/book mark)[[3]](#footnote-3).
* Allowing users to watch a repository (you will get notifications for new pull requests/issues)[[4]](#footnote-4).
* This will include communication from the mobile frontend to a webserver and animations making the user experience more enjoyable.

1. MergeMe (secondary functionality, medium priority)

* Find developers in your area.
* A ranking algorithm which will aggregate developers in your area based on common interested in language, project or need (advertising for a particular role etc).
* This aims to make it easier for people to find groups to work on open source projects/find like minded developers for hackathons etc.
* This will make use of geolocation to find other developers in the same city (only allowed if the user gives the application permission to access geo location/maps).

1. Events (extra functionality, stretch goal, low priority)

* Events can be used in conjunction with the secondary function of finding developers near by, for example: You could advertise a hackathon and have developers respond to it as going or not.
* This may or may not be implemented depending on time constraints.
* Note: This is deemed out of scope due to time constraints, but could be a good advertising platform.

## Application functionality

This section outlines a high level overview of the functionality in this application, further detail involving mockups can be found in the mockups section.

### Trending now (GitTrends)

This allows the user to browse trending repositories. The user can sort/filter the results by selecting the time frame to look at (i.e. trending repositories in the past day, the past week etc), the languages to include/exclude, the topic of the repository and the number of starts.

It will present the user with a view where they can interact with the repository by doing the following (visual representation of the view will be shown in the Storyboard section).

#### View the author

View the author’s/organization’s page including their bio, followers/following counts and repositories.

#### View a repository

A user can click on a repository to view it’s description, primary language, last commit, number of stars, number of forks and it’s contributors.

#### Star a repository

A user can star a repository on the mobile application which will result in the repository being started by their GitHub account (acts as a book mark). There will be an optional setting allowing the user to set a reminder to view this repository later, if set the application will give a push notification to the user at the time set.

#### Watch a repository

A user can watch a repository which will result in the repository being watched by their Github account (will notify users of changes such as pull requests, commits, open issues). Notifications of a watched repository are outside the scope of this project as it may lead to the spamming of notifications.

### Find developers (MergeMe)

This allows users to find developers in the same city as themselves. It will make use of the location data in a mobile device (GPS, wifi and mobile networks). It will be an optional feature for which users will need to grant the application permission to access location data and agree to share their information on their Github profile.

MergeMe will be an additional screen that shows near by developers, their primary language and if they are looking for a particular open source project/hackathon/event etc. It will only display those who opt into giving the application access to their location data and sharing their information.

#### Sharing information

Users will be able to share what they are looking for i.e. ‘Looking for a team member for GovHack Melbourne 2017’ or ‘Looking for a backend developer for my application’.

Approaches to the ranking algorithm, include using natural language processing (NLP) to pick out key words on what the user is sharing and then using a classifier to group people based on geographical location and key words identified by the NLP algorithm, but this may be outside the scope of the mobile application for 1 semester.

#### Natural Language Processing Approaches

These are several approaches to Natural Language Processing investigated, the appropriate library and language will be selected after the rest of the tech stack (webserver, database, language on server side, deployment/hosting).

* Google’s cloud platform, would work nicely with hosting the webserver and database on Google cloud.
  + <https://cloud.google.com/natural-language/>
* Stanford CoreNLP suite, only supports Java.
  + <http://stanfordnlp.github.io/CoreNLP/>
* Natural Language Toolkit, only supports Python.
  + <http://www.nltk.org/>

Note: I have not covered Events as they are out of scope for the timeframe and are low priority.

## Target Audience

Due to the nature of the application requiring a user to have a GitHub account to be authenticated, the application is aimed at GitHub users which includes but is not limited to:

* Professional developers/software engineers.
* Students.
* Hobbyists.
* Amature/novice programmers.
* Anyone who is interested in open source projects.

# Idea Innovation

### Why is this innovative?

As far as I know is application doesn’t already exist, there are similar implementations but I intend the focus on trending repositories and connecting people on open source projects to be the key defining differentiating with existing products.

Furthermore, there is already a market for users wanting to collaborate, share information, work on open source projects together and keep up to date with latest trends so there should be a market for this (see 2.1.4.).

### Does it already exist?

Several GitHub view applications already exist, however they each have their drawbacks and focus more on just viewing trending repositories/forking on mobile etc rather than on gathering people to work on projects.

* GitHub Trends[[5]](#footnote-5)
  + Cluttered UI.
  + Last update 2 years ago.
* ForkHub[[6]](#footnote-6)
  + Focused on productivity i.e. forking, looking at branches, gists, less so much open source trends and connecting people.
* TopGitHub[[7]](#footnote-7)
  + Last update 2 years ago.
  + Doesn’t look like you can select date range.
  + Nice UI but I believe it can be made more fun with animations.
* OctoDroid[[8]](#footnote-8)
  + Does a lot and kept updated, however I feel all of the features it has may over complicate the application and make users feel overwhelmed.

What my application will do differently to stand out will be:

* Crisp fun and interactive UI/UX
* Focus on trending repositories and getting more awareness for open source projects rather than productivity.
* Connecting people based on geographical location and interest/skill set.

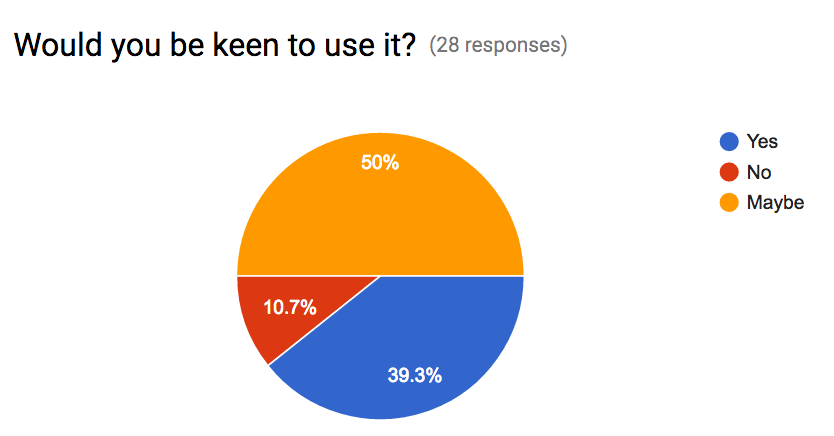
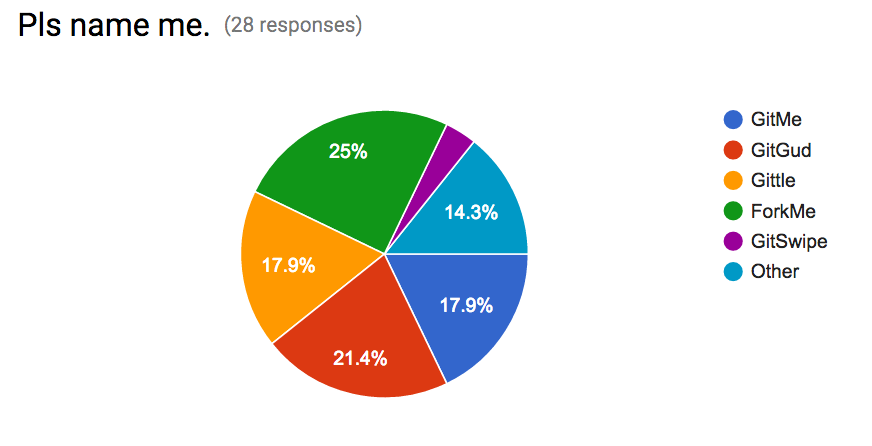
### But GitHub uses a web interface?

GitHub uses a web interface which is great at browsing repositories and reading content[[9]](#footnote-9) however after talking to fellow developers/students not too many people check the trending repositories page (it isn’t even on your GitHub home page). This bought about products such as GitHut to turn your chrome new tab page into trending GitHub projects[[10]](#footnote-10).

By using a mobile application, I aim to increase the viewer base for GitHub’s trending repositories along by maintaining the user base via the use of push notifications to remind users to look at repositories they have started.

Furthermore, enabling finding other like minded developers is a completely new feature to any existing product and will work great on mobile devices as they are primarily used for messaging.

### Is it useful?

As part of intending this application to be open source and to gauge how useful it will be I conducted a poll help me name this application and to see if there a need for it. Fellow IT students from various universities (Monash, Uni Melb, RMIT, UNSW, USYD) filled it out.

The results show that it has had a relativity positive reaction and people are inclined to use it or at least give it a try.

Furthermore, groups such as Python meet-ups, Atlassian user groups, Free Code Camp meet-ups already existing proving people want to be connected in working in technology, open source projects and keep up with latest trends and technologies.

# Techonology Considerations

This section outlines the technology considered for the overall functionality of the application for the Android platform, iOS platform and webserver and database backend.

The functionality that needs to be supported include:

* General
  + Permissions
  + HTTP requests to talk to backend
    - Transferring data
  + Local storage
* GitTrends
  + Markdown (.md) render
* MergeMe
  + Send email from application to contact developer

## Android

I will develop using android 7.1 Nougat API 25 and I will use android support library to versions 5.0 – 4.0 depending on time constraints.

### Transferring data - Protobuffers

I will need to transfer data between my webserver and the mobile front ends and from GitHub’s API to my webserver. Below I will consider the approaches to transfer data from my webserver to the mobile front end.

I have chosen to try to user Protobuffers and will have JSON as my backup plan.

#### Transfer format

##### JSON

JSON is a commonly used data interchange format for web and mobile devices. There are many libraries to support the use of JSON and it is the most widely used.

I will use Gson[[11]](#footnote-11) to convert Java Objects to JSON and vice versa

. An advantage of Gson is that is serializes the JSON data to a bitstream which means sending JSON data with Gson

Furthermore, Gson seems to be more compatible[[12]](#footnote-12)[[13]](#footnote-13) with protobuffers than any other JSON library, they are both also Google developed so using Gson so I can choose to implement data transfer using JSON or Protobuffers later would be ideal.

##### Protocol Buffers

Protocol buffers (protobuffers) are a Google developed way to serialize structured data to a binary making it a more compressed way to transfer data when compared to JSON[[14]](#footnote-14).

They allow for compact typed data transfer where you create a schema for your message and don’t need to parse the bitstream of seralised data into human readable strings. Protobuffers are higher performance than JSON[[15]](#footnote-15).

While they are more complex than just using JSON, it is a technology I want to learn to use.

#### Web Library – Volley

The two web libraries considered are Retrofit[[16]](#footnote-16) and Volley[[17]](#footnote-17), while they are both good libraries for HTTP requests, I have chosen volley as it fits with my technology stack better as Android, Protobuffers, Gson are all Google developed.

|  |  |
| --- | --- |
| **Retrofit** | **Volley** |
| * Daily or so updates[[18]](#footnote-18) * Harder to find support for Protobuffers | * Google developed, because I am using Gson or Protobuffers which are also Google developed, I am more inclined to use a Google developed library for HTTP as it will probably be more compatible. * Monthly or so updates[[19]](#footnote-19) |

#### Local Storage - SQLite

Most of the data processing will be done on the webserver side, however a SQLite[[20]](#footnote-20) database to cache data locally which can be used for push notifications will be used it is natively supported and easy to work with.

The SQLite instance will store information for a push notification such as reminder time, repository url, repostiroy main language, repository description, repository author/organization in the table cached\_started\_repositories.

## iOS

### Transferring Data – JSON

I will need to transfer data between my webserver and the mobile front ends and from GitHub’s API to my webserver. Below I will consider the approaches to transfer data from my webserver to the mobile front end.

For iOS it will be easier to use JSON as there is more support for JSON and no official support for using Protobuffers in iOS.

#### Transfer format

##### JSON

JSON seems to be the easier approach when dealing with iOS.

SwiftyJSON[[21]](#footnote-21) nicely integrates with Alamofire.

##### Protocol buffers

There doesn’t seem to be an official Google Protobuffer API for Swift. However, there are several 3rd party projects working on it.

* Apple (prerelease) <https://github.com/apple/swift-protobuf/>
* Independent (seems to be stable) <https://github.com/alexeyxo/protobuf-swift>

#### Web Library - Alamofire

Alamofire[[22]](#footnote-22) will be used as it abstracts onto of Apple’s network stack making it easier for the developer to use[[23]](#footnote-23), it also pairs really well with SwifyJSON.

#### Local Storage – CoreData

CoreData will be used to store entities locally on an iOS device. It will store information required for push notifications for stared repositories such as reminder time, repository url, repository main language, repository description, repository author/organization in the entity modle cached\_started\_repositories.

## Mobile Platform Independent

TODOO, send email, each functionaliy from iOS and Android perspective

### GitHub API

GitHub current has an API[[24]](#footnote-24) out that can be used to get information on users, repositories, trends and authentication.

Ideally I want to set up a REST API on my own webserver that interfaces with GitHub’s API and pushes information to the mobile frontends.

This section outlines the ability to get the relevant information needed to prove this is viable.

* Authentication
  + GitHub OAuth for android: <https://github.com/geniushkg/github-oauth>
  + GitHut OAuth documentation: <https://developer.github.com/guides/basics-of-authentication/>
* Trending Repositories
  + GitHub search API: <https://developer.github.com/v3/search/>
  + Examples:
    - <http://stackoverflow.com/questions/30525330/how-to-get-list-of-trending-github-repositories-by-github-api>
    - https://gist.github.com/jasonrudolph/6065289
* Getting repository data
  + GitHub user documentation: <https://developer.github.com/v3/users/>
  + $ curl <https://api.github.com/users/darvid7/repos>
* Getting user data
  + GitHub repository documentation: <https://developer.github.com/v3/repos/>
  + $ curl <https://api.github.com/users/darvid7>
* Staring/Watching a repository
  + <https://developer.github.com/v3/activity/>
* Other useful links
  + <https://gist.github.com/caspyin/2288960>

### Webserver (backend)

The focus is on the application, however it will require a webserver to interact with. This section outlines the pros and cons of various languages/frameworks and deployment/hosting strategies that may be employed.

#### Language/Framework

Depending on time and workload I will choose the most appropriate language/framework later on. As this unit is focused on the Mobile Application aspect and no so much the webserver I have done a brief overview of the languages/frameworks I am consideration along with reasons why I would choose one over the other.

##### Go/HTTP[[25]](#footnote-25)

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| * Want to learn Go. * Heavily used language within Google. | * Little experience with Go. |

##### JavaScript/Node.js[[26]](#footnote-26)

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| * Really popular in industry and web development. * Have worked with Node.js and express before. * Many useful libraries. | * Not as proficient with JavaScript. |

Note: Protocol buffers[[27]](#footnote-27) (protobuffers) can be used with

* Go
  + <https://github.com/golang/protobuf>
* Python
  + <https://developers.google.com/protocol-buffers/docs/pythontutorial>
* JavaScript
  + <https://github.com/dcodeIO/ProtoBuf.js/>

However there seems to be more support for Python and Go and less for JavaScript[[28]](#footnote-28).

##### Python/Flask[[29]](#footnote-29)

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| * Most proficient language. * Easy lightweight framework. | * Want to branch out and learn new languages. |

#### Deployment/Hosting – Google App Engine

I’m considering these two to host/deploy the backend based on personal preference.

##### Heroku

Can host Go, Python and Node.Js applications, it can also host PostgresSQL databases[[30]](#footnote-30)

##### Google App Engine/Cloud platform

Can host a variety of languages include Go, Python and Node along with supporting Cloud storage, Mongo DB and PostgresSQL[[31]](#footnote-31). Having use so many Google developed products I am inclined to go with google App Engine.

#### Server Side Storage

Will be populated with trending GitHub repositories on an internval i.e. every hour. Most of the data will be processed on the server side before being sent to mobile client i.e. sort all repos by language first before sending to the mobile client if the user specifies it.

Additionally the server side storage will store messages (todo)?

##### PostgreSQL

Since SQLite will be used for local storage, using a SQL database for server storage makes the most sense.

PostgresSQL[[32]](#footnote-32) is compatible with:

* Go <https://github.com/go-pg/pg>
* Python <https://wiki.postgresql.org/wiki/Python>
* JavaScript <https://github.com/brianc/node-postgres>

So it will be able to work with which ever web framework is chosen for the backend.

##### Firebase

Is another consideration is Firebase[[33]](#footnote-33) which may be useful for real time chat integration for the MergeMe functionality, however this can still be done with PostgressSQL. Adding the complexity of two databases may be more overhead compared to polling PostgressSQL at intervals. Additionally messaging can be seen as additional feature, as part of the minimum viable product the application an just allow users to see each other’s email and facilitate sending them an email.

# Interface Design Story Board Mockups

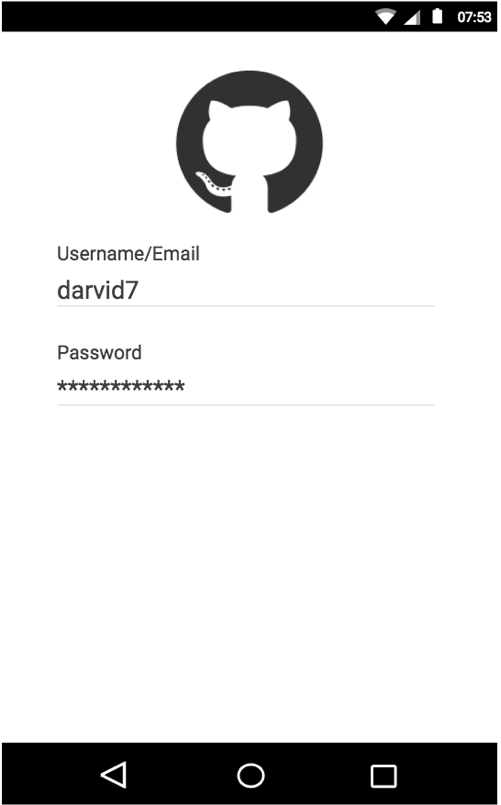
The mockups are high fidelity.

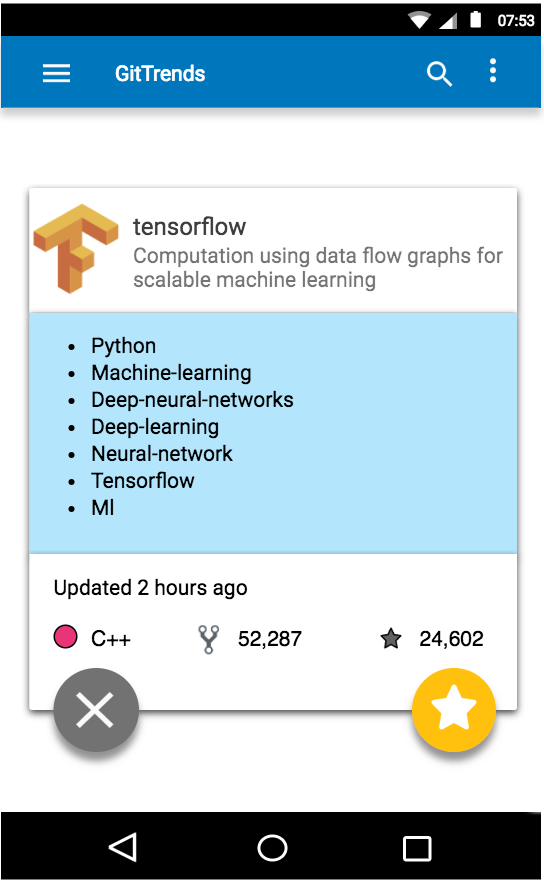
For Android I have tried to stick with material design.

* Simplistic minimalists interface.
* Used the colour pallet[[34]](#footnote-34) to pick colors
  + Primary Light Blue 500 #03A9F4
  + Lighter 100 #B3E5FC
  + Darker 800 #0277BD
  + Amber 500 #FFC10
* Floating action buttons (however, I use two because there are 2 main functionalities).
* Use of animations, cards and list views.
* Input directly under text label
* Text following Typography[[35]](#footnote-35) standard
  + Font: Roboto
  + Subheading 16sp
  + Body 14sp

## ANDROID%20IOS/Screen%20Shot%202017-03-29%20at%2011.03.15%20PM.pngSplash Screen

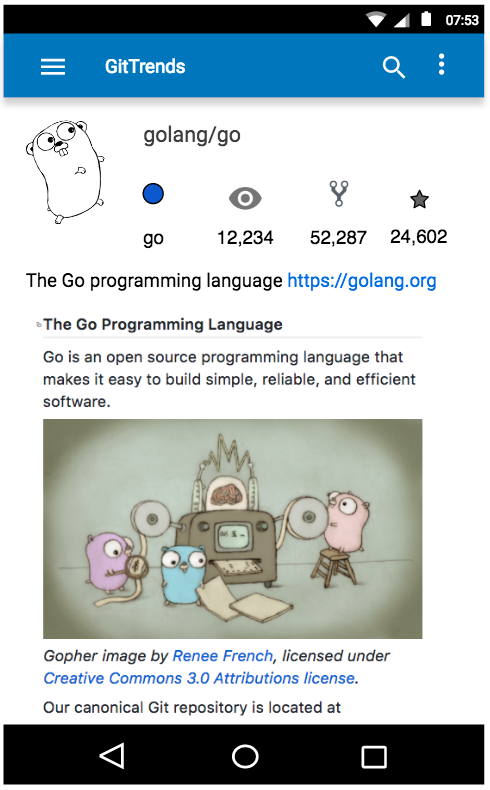
## Log in

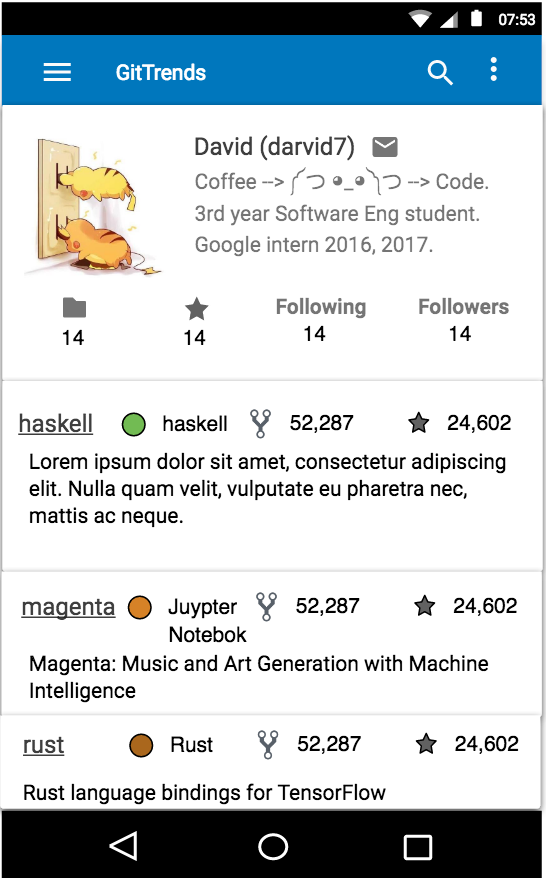




## Trending Page (GitTrends)

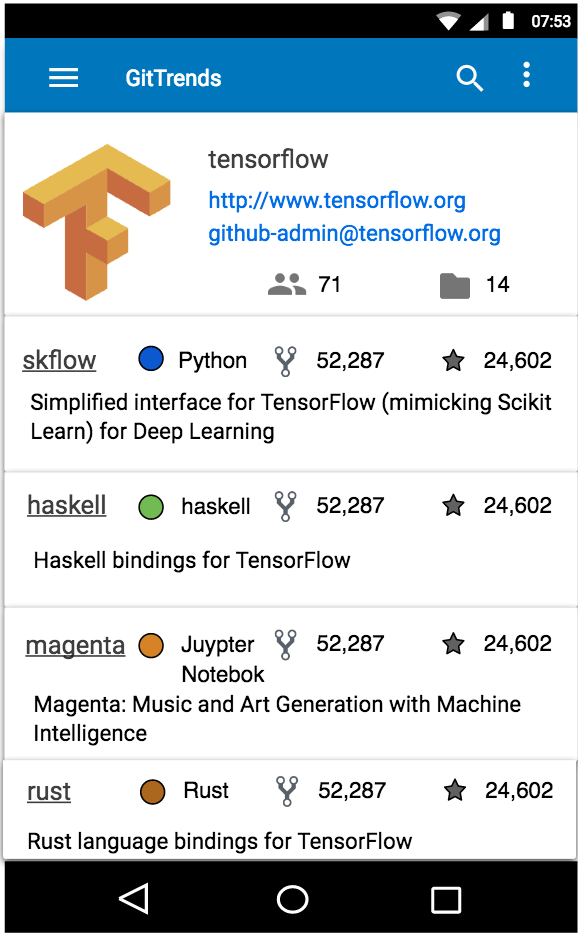
## Repository View



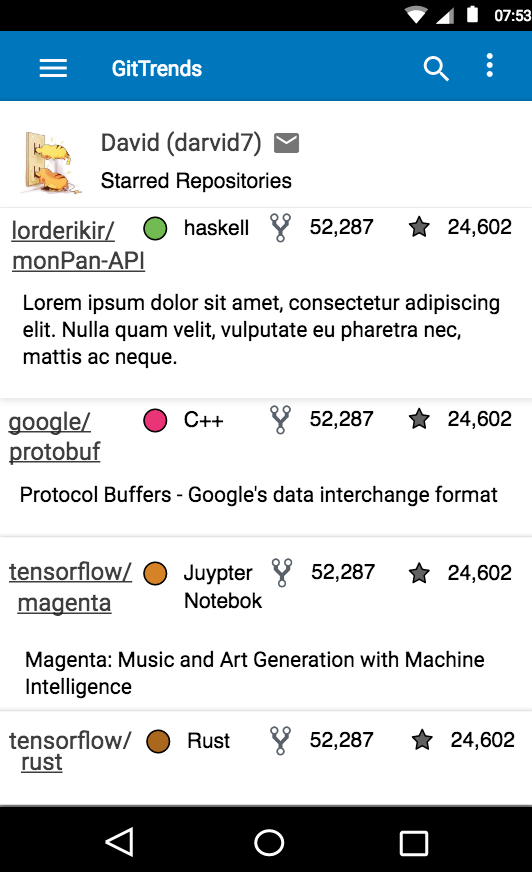


## User View

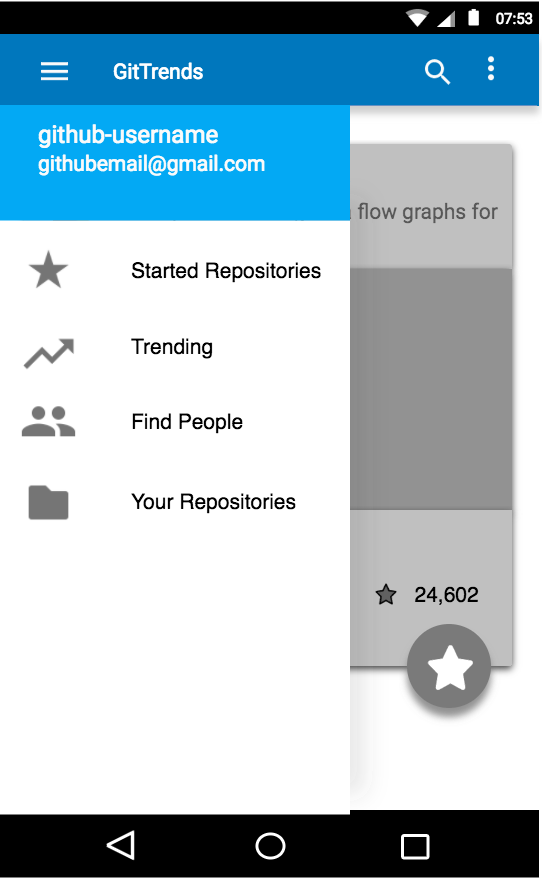
## Organization View

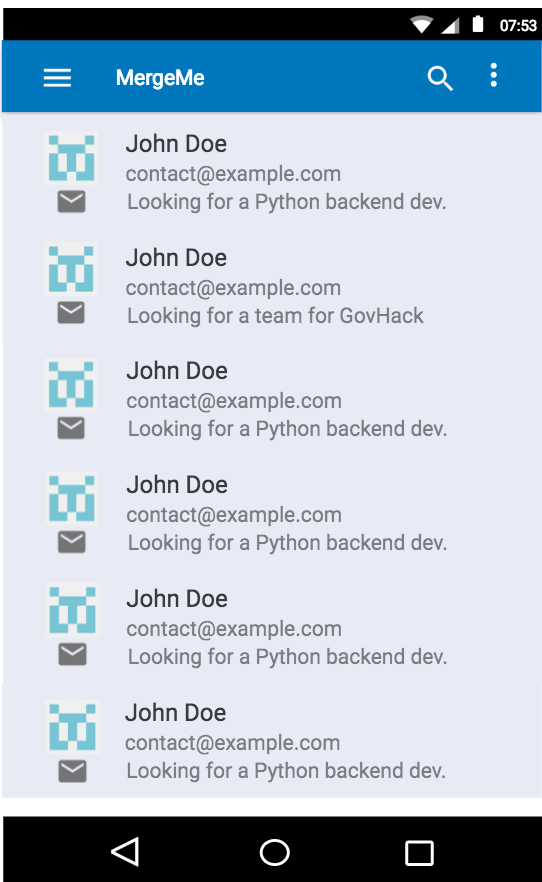


## Starred Repository View



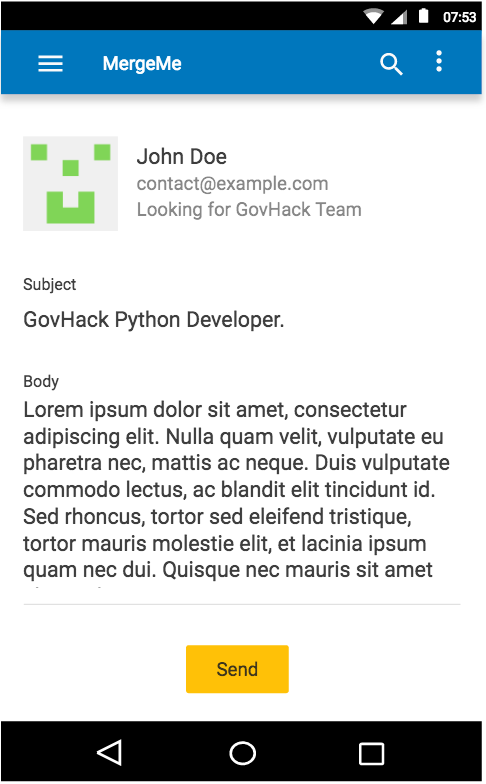
## Hamburger (side bar)





## Find Developers (MergeMe)

## Contact a developer



# Scope and Limitations

## Scope

As mentioned in the introduction, the application consist of 3 main aspects of functionality. However, due to time constraints the scope for this is limitd to the primary and secondary key functionality aspects.

1. Trending now - GitTrend (primary functionality)

* The ability for a user to browse through popular open source repositories currently trending on Github[[36]](#footnote-36).
* Allowing users to star a repository (like a Facebook like/book mark)[[37]](#footnote-37).
* Allowing users to watch a repository (you will get notifications for new pull requests/issues)[[38]](#footnote-38).
* This will include communication from the mobile frontend to a webserver and animations making the user experience more enjoyable.

1. Find developers in your area – MergeMe (secondary functionality)

* A ranking algorithm which will aggregate developers in your area based on common interested in language, project or need (advertising for a particular role etc).
* This aims to make it easier for people to find groups to work on open source projects/find like minded developers for hackathons etc.
* This will make use of geolocation to find other developers in the same city (only allowed if the user gives the application permission to access geo location/maps).

## Minimum Viable Product

## Limitations

Limitations include time, programming ability in Java/Swift,

Natural Language Processing

* I have no have experience with NLP, so this may be slightly out of scope for this unit as it is focused on Mobile Applications. However I will try to do this I time permits.

Instant chat

* Do email instead[[39]](#footnote-39)

## Stretch Goals

The 3rd part of the key functionality

# Estimated Project timeline

# Testing

Something something continuous integration server

1. image from: <https://www.lifehacker.com.au/2013/02/ask-lh-how-the-heck-do-i-use-github/> [↑](#footnote-ref-1)
2. GitHub trending repositories: <https://github.com/trending> [↑](#footnote-ref-2)
3. GitHub staring a repository: <https://help.github.com/articles/about-stars/> [↑](#footnote-ref-3)
4. GitHub watching a repository: <https://help.github.com/articles/watching-repositories/> [↑](#footnote-ref-4)
5. <https://play.google.com/store/apps/details?id=me.pmpm.githubtrends> [↑](#footnote-ref-5)
6. <https://play.google.com/store/apps/details?id=jp.forkhub> [↑](#footnote-ref-6)
7. <https://play.google.com/store/apps/details?id=com.mmazzarolo.dev.topgithub> [↑](#footnote-ref-7)
8. <https://play.google.com/store/apps/details?id=com.gh4a> [↑](#footnote-ref-8)
9. <https://github.com/blog/1559-github-s-on-your-phone> [↑](#footnote-ref-9)
10. <https://github.com/kamranahmedse/githunt> [↑](#footnote-ref-10)
11. <https://github.com/google/gson> [↑](#footnote-ref-11)
12. <https://github.com/google/gson/blob/master/proto/src/main/java/com/google/gson/protobuf/ProtoTypeAdapter.java> [↑](#footnote-ref-12)
13. <https://groups.google.com/forum/#!topic/google-gson/obCKdK18quc> [↑](#footnote-ref-13)
14. <http://stackoverflow.com/questions/4979754/gson-vs-protocol-buffer> [↑](#footnote-ref-14)
15. <http://blog.codeclimate.com/blog/2014/06/05/choose-protocol-buffers/> [↑](#footnote-ref-15)
16. <http://square.github.io/retrofit/> [↑](#footnote-ref-16)
17. <https://developer.android.com/training/volley/index.html> [↑](#footnote-ref-17)
18. <https://github.com/square/retrofit/commits/master> [↑](#footnote-ref-18)
19. <https://github.com/google/volley/commits/master> [↑](#footnote-ref-19)
20. <https://www.sqlite.org> [↑](#footnote-ref-20)
21. <https://github.com/SwiftyJSON/SwiftyJSON> [↑](#footnote-ref-21)
22. <https://github.com/Alamofire/Alamofire> [↑](#footnote-ref-22)
23. <https://www.quora.com/How-would-using-Alamofire-benefit-me-in-Swift> [↑](#footnote-ref-23)
24. <https://developer.github.com/v3/> [↑](#footnote-ref-24)
25. <https://thenewstack.io/building-a-web-server-in-go/> [↑](#footnote-ref-25)
26. <https://nodejs.org/en/> [↑](#footnote-ref-26)
27. <https://developers.google.com/protocol-buffers/docs/reference/overview> [↑](#footnote-ref-27)
28. <http://stackoverflow.com/questions/6912981/google-protocol-buffers-javascript> [↑](#footnote-ref-28)
29. <http://flask.pocoo.org> [↑](#footnote-ref-29)
30. <https://www.heroku.com/postgres> [↑](#footnote-ref-30)
31. <https://cloud.google.com/appengine/docs/flexible/nodejs/using-third-party-databases> [↑](#footnote-ref-31)
32. <https://www.postgresql.org> [↑](#footnote-ref-32)
33. <https://firebase.google.com/> [↑](#footnote-ref-33)
34. <https://material.io/guidelines/style/color.html> [↑](#footnote-ref-34)
35. <https://material.io/guidelines/style/typography.html> [↑](#footnote-ref-35)
36. GitHub trending repositories: <https://github.com/trending> [↑](#footnote-ref-36)
37. GitHub staring a repository: <https://help.github.com/articles/about-stars/> [↑](#footnote-ref-37)
38. GitHub watching a repository: <https://help.github.com/articles/watching-repositories/> [↑](#footnote-ref-38)
39. <http://stackoverflow.com/questions/2197741/how-can-i-send-emails-from-my-android-application> [↑](#footnote-ref-39)