horizontal line

**Pocket Rocket**

Hackbright Academy

Project Pocket Rocket

**December 2017**

# OVERVIEW

BUDdy is a social network supporting small businesses. In the capital markets of big chains, many “mom & pops” survive off “word of mouth” and referrals, as well as the loyalty built from customer service. BUDdy also provides tools and community for businesses to support each other.

# GOALS

1. Search engine and reviews of small businesses in a respective local community
2. Facilitation of consumer-to-consumer and business-to-consumer referrals, with tracking and redemption of promotions
3. Tools and forums to support small businesses

# SPECIFICATIONS

## General

Functionalities to consider:

* Metrics to determine if business qualifies as “small business”
  + Validation of businesses, including allowing users to report businesses (e.g., unethical practices) - include certification stamp
* Minimum rating to remove business from network or specific filter to include them (e.g., they don’t pop up in default feed)
* Monthly fee to be in the network but with a reduced fee when a certain number of reviews or check-ins occur each month (incent engagement)
* Separate sections for reviews that are below 3 to allow businesses to address - allow users to update overall rating and/or allow user to rate based on follow-up customer service which would be used in algorithm to slightly adjust original rating by some weight
* Gamification/check-in capability to allow consumers to:
  + Redeem promotions
  + Filter or weight ranking of consumer review based on frequency of check-ins either at a specific location or active user to the site (e.g., assumption is person is connoisseur of small businesses)
* Icon for those who accept credit cards and/or affiliated with American Express Small Business group - also consider if they partner/purchase from other small businesses and/or part of green initiative
* Profiles list referral bonuses based on referred consumer checking in to initialize
* Business affiliations - tag or list businesses that are owned by same owner or group (e.g., concept of same management team and similar quality and/or transparency when businesses make referrals to sister businesses)
* In addition to business details (e.g., hours, website, menus, reservation links, etc.), forum for Q&A, either response from owner and/or other consumers
* Data analytics/visualization for trending businesses in specific location (e.g., based on weekly check-ins, number of reviews, etc.)
* Offer e-certificates that can be used at any of the small businesses (can be long-term goal) but this would offer the revenue stream and referral as users can gift it. As with other referral gift card programs there is a 10-20% referral fee.
* Link to Google Maps for location and directions
* Link to VENMO?
* Link to Google Calendar for appointments? May be tough if businesses use different reservation tools or create one in the program?

## Consumer Experience

Functionalities to consider:

* Login from Google or Facebook, pulling in friends list from Google +/contacts, Facebook friends or phone contacts
* Gaming component:
  + Points for checking in, referrals that are redeemed, reviews and likes of reviews
  + Categories of consumer types based on total points (e.g. by levels or by catchy phrases like newbies, mover & shaker, etc.)
* Rate and review businesses that have been checked into:
  + For reviews that were addressed by business owner/management, allow consumer to either rate the customer service experience and/or change original rating. If the former, then that rating is included in algorithm to weight and adjust original rating to the extent customer service was favorable.
  + Respond to any questions from other consumers in regards to the business
* Profile with referrals list as well as referrals redeemed by other consumers, which translates to referral promotion that consumer can redeem

## Business Experience

Functionalities to consider:

* Promotions offered for check-ins and/or referrals
* Address consumer reviews
* Flag reviews, etc. for dispute, if escalation required
* Profile with referrals list as well as referrals redeemed by other consumers, which translates to referral promotion that business can redeem
* Analytics on business, including # of referrals, redemptions and % that convert to redemption and/or reviews

## Administrator Experience

Functionalities to consider:

* Address and override/adjust reviews and business information from escalation
* Validate and remove businesses

# MILESTONES

## Version 1.0

Minimum viable product (MVP):

* Setup database and tables
* Sign in or log in using Facebook or Google OAuth
* Pull in friends/contacts if logged in using Facebook or Google log-in and allow user to select friend to send friend requests to, if account exists. If not, send email to “friend/contact” inviting the person to join BUDdy
* Provide link to email for friend to sign up
* Provide link to email to validate new user account
* Add business with limited details
* Add review and rating with limited functionality, mostly set up of webpage
* Display user profile with analytics of check-ins and reviews
* Check-in
* Display separate section for reviews that are being addressed by business

## Version 2.0

## Build out additional functionalities to include:

* Modify account information and/or link additional emails (for friends to find)
* Validate email(s) and/or other edge cases using regex(?)
* Reset password
* Claim business(es) and lists of related businesses/affiliations
* Search keywords
* Display referral/redemption info and other metrics:
  + User:
    - Total check-ins
    - Total check-ins per business
    - Total reviews
    - Total referrals/redemptions
  + Business:
    - Total referrals/redemptions (as user and as business)
    - Percentage of redemptions
    - Average rating
    - Top features (perhaps keywords by reviewers) - use of regex?
    - Total check-ins per business (monthly data)
    - Loyal customers (data on repeat visits)
* Display reviews:
  + User:
    - Total check-ins
    - Total check-ins per business
    - Member since (join date)
    - Rating
    - Review (character limit)
    - Revised rating/review
    - Customer service rating (if rating not revised)
    - Edit capabilities
  + Business:
    - Average rating
    - Individual reviews (see above)
    - Response to user review
    - Escalation button

## Future Versions

Functionalities in future versions:

* Messaging functionality => inbox or direct email
* Edit business information
* Upload pics of businesses
* Google Maps => directions
* eCertificates and/or other monetization/incentive aspects
* Booking appointments and linking to Google Calendar
* Pay options (e.g., PayPal, VENMO, credit cards) to facilitate payment as credit to be applied to the small businesses, as many may only take cash
* Escalation functionality, including reporting business and fraudulent reviews

# OTHER RESOURCES

## Other Projects

<https://github.com/kmorinaka/hackbright-project>

Off the Chain is a search engine that highlights the small, local businesses in your community. Big corporations are constantly opening up new locations and running out independently owned shops. It's hard to compete with brand names that are known throughout the country. For those of you that want to support the business owners in your area and contribute to the local economy, Off the Chain can help you find alternative businesses that have the service, product or meal you are looking for.

<https://cloud.githubusercontent.com/assets/11863012/8620412/441f9eda-26d4-11e5-83d7-b66f913fa018.png>

<https://cloud.githubusercontent.com/assets/11863012/8620473/a46d0304-26d4-11e5-92b6-bf5e4be33d9e.png>

## APIs

American Express - mainly payment services though and not list of small businesses <https://developer.americanexpress.com/products>

Google Maps - <https://developers.google.com/maps/>

Google - <https://console.developers.google.com/apis/dashboard>

Facebook - <https://developers.facebook.com/>

Yelp - <https://www.yelp.com/developers/documentation/v2/overview>

Tripadvisor - <https://developer-tripadvisor.com/content-api/>

**Stack**

The following technology were used:

* Python
* JavaScript
* SQL
* postgreSQL
* Flask
* SQLAlchemy

User Flow

# VERSION 1.0

## Initial setup of database

Based on if empty database - perhaps bring in some items from Yelp, Tripadvisor or Google or create text of those to utilize as user input



Pseudocode

# VERSION 1.0

## /home

Homepage with logo/brand, log-in and stuff like About Us, Contact, Search, etc.

* Log-In - pop up screen to allow user/business to log in via Google or Facebook profile; otherwise, option to sign up (**/sign up**)
  + Google (while loop):
    - Validates email to Google API => if not, displays error message with requirement for new email
    - Validates password to Google API => if not, displays error message with requirement for new password (Future - allow for password reset)
    - Asks user for input for user name
    - Future - would require business to ‘claim’ a business already in database or set up business info in database (**/business**)
    - Provides link to sign up (**/sign up**) or log in via Facebook
  + Facebook (while loop):
    - Validates email to Facebook API => if not, displays error message with requirement for new email
    - Validates password to Facebook API => if not, displays error message with requirement for new password (Future - allow for password reset)
    - Asks user for input for user name
    - Future - would require business to ‘claim’ a business already in database or set up business info in database (**/business**))
    - Provides link to sign up (**/sign up**) or log in via Google
  + Upon initial sign-in via Google or Facebook, takes user to friends page (**/friends**)
    - Gets friends/contacts list from Google/Facebook API with checkboxes for user to select who to send friend requests
      * If friend is in database, sends friend request to (**/messages**) for friend to accept request; otherwise, generates and sends email to friend for person to opt to sign up
  + After successful log-in, login link changes to profile link that has drop down menu for:
    - Account info
    - Friends list
    - Referrals list
* About Us - displays mission statement, possibly “founders”
* Contact - displays contact information including hyperlink to email (rather than submission form)
* Search - If person has not signed in, redirects to login pop up screen (Future - functionality of search feature)

## /sign up

Sign up page that takes in inputs from user

* Validates email (while loop)
  + Returns error message if not valid email with @ (future version to send message to email with link to click on to validate email)
  + Checks email to see if it is already in database; if so, redirects user to login pop up screen in home page (**/home**) to input password
  + Hash/encrypt password input
  + Asks user for input for user name
  + Future - would require business to ‘claim’ a business already in database or set up business info in database (**/business**))
* Redirects to home page (**/home**) upon completion of sign up

## /profile

Links to various pages for account, friends list, referral page, etc. as menu list on left side (**base.html**)

* Account Info (**/account**)
  + Displays user name
  + Displays user email (Future - allow user to edit and link other emails for friends to find)
  + Future - allow user to reset password
  + Future - displays metrics from gamification (e.g., # of check ins, # of different businesses checked into, # of reviews, # of referrals, # of redemptions)
  + Future - for business profiles, list and link to businesses ‘claimed’
* Friends List (**/friends**)
  + Displays friends (Future - displays # of referrals to each and links to page with list of referrals and redemptions)
  + Displays friends from Facebook or Google API for user to check to add as friend
  + Link to pop up screen to input friend’s email to add
    - Validates friend’s email to database and if in database, sends friend request to (**/messages**) for friend to accept request; otherwise, generates and sends email to friend for person to opt to sign up
* Referrals List (**/referrals**)
  + Displays list of referrals in sections (referred to and referred from as well as redemptions) (Future - displays metrics from gamification (e.g., # of referrals, # of redemptions, % redeemed)
* Messages (**/messages**)
  + Displays message for user to accept friend request
  + Future - messages between users, businesses and administrator, including reviews for business profiles or responses to reviews from owners/management for consumer profiles (may link this to **/reviews**)
* Future - Reviews (**/reviews**)

## /business

For each business, various details are provided including reviews, etc. (**base.html**). Each business would have its own ‘child’ page (**/business/xyz**)

* For new business, allow user to enter various details of the business (name and address required; others - optional)
* For each business page:
  + Display business information
  + Display promotion information for referrals/redemptions
  + Button to execute referral/redemption (Future - click of button executes transaction)
  + Future - Reviews (2 sections, including compartmentalizing the ones that are currently being addressed by management) => include user name, total check-ins and total check-ins per business for reviewer
  + Future - Map and directions via Google Maps

# VERSION 2.0

## /home

# Additional homepage functionalities:

* Log-in - allows for password reset if forgot password
  + Triggers link being sent to email to validate user and to change password
* Claim - business user has option to claim business(es)
  + Email sent to business email to validate
* Search - search for keywords, categories, business name, etc.

## /profile

Continue...

# DATABASE SCHEMA

## user

|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Modifier(s)** |
| user\_id | serial | primary key |
| first\_name | varchar(64) | required |
| last\_name | varchar(64) | required |
| username | varchar(64) | required; unique |
| email | varchar(64) | required; unique |
| valid\_email | boolean | default=false |
| password | varchar(64) | required \*Google Bcrypt |
| user\_pic | img | default=emoticon |
| DOB | timedate |  |
| join\_date | timedate | default=current timestamp |
| biz\_acct | boolean | default=false |

Relationships:

* User pic (separate image table) => one to one
* Business - backref; secondary - user\_business => one to many
* Invites - backref; secondary - friends => many to many
* Promotions - backref; secondary - user\_promo => many to many
* Check\_Ins - backref => many to many
* Referrals - backref => many to many (perhaps many to one because each referral has separate transaction id
* Reviews - backref => one to many

## friends

|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Modifier(s)** |
| user\_id | integer | foreign key; primary key |
| friend\_id | integer | foreign key; triggered when friend accepts invite and signs up (invites.accepted = true) |

## user\_business

|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Modifier(s)** |
| user\_id | integer | foreign key; primary key |
| biz\_id | integer | foreign key; triggered when business is claimed (business.claimed = true) |

## business

|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Modifier(s)** |
| biz\_id | serial | primary key |
| biz\_name | varchar(100) | required; unique(?) |
| address | varchar(100) |  |
| city | varchar(64) |  |
| state/province | varchar(64) |  |
| country | varchar(64) |  |
| phone | varchar(64) |  |
| email | varchar(64) |  |
| valid\_email | boolean | default=false |
| category | varchar(64) | primary category from predetermined options (for search feature); limit to 1 category for now - perhaps future to build out separate category table |
| days\_open | varchar(64) |  |
| open\_time | integer | military time |
| close\_time | integer | military time |
| claimed | boolean | default=false; triggered when business user clicks link to business email to validate |
| biz\_pic\_main | img | default=emoticon |

Relationships:

* Business pic (separate image table) => one to one
* Check\_ins - backref => one to many
* Referrals - backref => one to many
* Reviews - backref => one to many
* Business\_promo - backref => one to many

## business\_promo

|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Modifier(s)** |
| biz\_id | integer | foreign key; primary key |
| promo\_id | integer | foreign key |
| promo\_active | boolean | default=true; if expired, false; based on expiration date from promotions table; to be used to list active promotions on business webpage |

Relationships:

* Promotions => one to many

## promotions

|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Modifier(s)** |
| promo\_id | serial | primary key |
| title | varchar(64) | required |
| description | varchar(64) |  |
| start\_date | datetime |  |
| end\_date | datetime |  |
| referral\_promo | boolean | default=false |
| birthday\_promo | boolean | default=false |
| redemption\_count | integer | to be calculated upon expiration date from total from user\_promo table, ordered by promo\_id |

## user\_promo

|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Modifier(s)** |
| user\_promo\_id | serial | primary key |
| promo\_id | integer | foreign key |
| user\_id | integer | foreign key |
| redemption\_date | datetime |  |
| redeemed | boolean | default=false |

## check\_ins

|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Modifier(s)** |
| checkin\_id | serial | primary key |
| user\_id | integer | foreign key |
| business\_id | integer | foreign key |
| checkin\_date | datetime | current timestamp |

## referrals

|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Modifier(s)** |
| referral\_id | serial | primary key |
| referrer\_id | integer | foreign key |
| referree\_id | integer | foreign key |
| business\_id | integer | foreign key |
| refer\_date | datetime | current timestamp; instantiates a record in user\_promo table |

## reviews

|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Modifier(s)** |
| review\_id | serial | primary key |
| user\_id | integer | foreign key |
| biz\_id | integer | foreign key |
| rating | integer | required (1-5) |
| review | varchar(5000) |  |
| review\_date | datetime | current timestamp |
| biz\_response | varchar(5000) |  |
| dispute | boolean | default=false |
| revise\_review | boolean | default=false |
| new\_rating | integer | 1-5 |
| new\_review | varchar(3000) |  |

## likes

|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Modifier(s)** |
| review\_id | integer | foreign key; primary key |
| user\_id | integer | foreign key |

## invite

|  |  |  |
| --- | --- | --- |
| **Field name** | **Type** | **Modifier(s)** |
| invite\_id | serial | primary key |
| user\_id | integer | foreign key |
| friend\_email | varchar(64) |  |
| accepted | boolean | default=false; if yes, instantiates friends table |

## ~~emails (may not build out - same with e-card/e-certificate table)~~

|  |  |  |
| --- | --- | --- |
| **~~Field name~~** | **~~Type~~** | **~~Modifier(s)~~** |
| ~~user\_id~~ | ~~varchar(64)~~ | ~~foreign key; primary key~~ |
| ~~email1~~ | ~~varchar(64)~~ |  |
| ~~email2~~ | ~~varchar(64)~~ |  |

## 