

Backlog

DevOps - Add function for users to grab config files via simple call.
Contains DB-info (redis and mysql)



DevOps - Track down and squash bug that causes connections to time out.



Spam - Figure out how to parse dozens of GBs of spam emails.



NS

TLP - Design Templated Learning Platform structure



NS

To do

Clients/Cpp - Add Documentation and Examples



Clients - Add documentation to the wiki detailing use and dependencies



NC

Clients - Implement client script after having high-level planning checked off



NC

API - Add simple unit tests



README - Fill out OCR/Face-Detect/Spam-Detect sections of README.



Clients - Research and write tests for the client script



NC

API - Update our API to become more REST-ful



NC

Docker Image - Set up a git-hooks route on devops server to take posts from github whenever master branch is touched.



Clients/Python - Get Praxyk Python Libs on PIP for easy download. see - <http://peterdowns.com/posts/first-time-with-pypi.html>



Clients/Javascript - Design Javascript Bindings at High-level



Clients/Javascript - Implement Javascript bindings



Queue - Add spam detection task queue function



API - Add /accounts/ route to get meta-data on your account like money owed and total data ingress, etc.



Clients/Python - Add coupons route for users to add coupons to their account



Client Script - Add coupons route



NC

Website - Add TOS agreement checkbox and TOS page.



Website - Fix bug where results don't load consistently on /dashboard/transaction/X pages.



Website - Fix bug where next-page is off by one (doesn't show when on n-1 page to go to nth page)



Api - Make billing function to bill customers on a monthly statement.



Website - Add cookie to track whether sidebar is collapsed or not and persist that through page-jumps



In progress

OCR - unit tests



Clients/Python - Fix bug where first_page_num and prev_page_num go null when last page is accessed via last_page() or next_page()



Spam - get_spam_chance(): Hook up mlpack to output spam chance

NS

Clients/Python - Finish filling out examples wiki page.



Clients/Python - Complete Documentation



NC

Clients/Cpp - Design client bindings structure.



OCR - documentation



Spam - transforming text into features

NS

Completed

Payment - figure out what system to use



Payment - implement system



Fill out remaining wiki for payment system



Flask-security - Research and Implement token-auth System with flask-security module.



Spam - Feature extraction from spam data

NS

Clients/Python - Implement
POd/OCR route file uploads



NC

DataBase - Implement Facial_Rec
redis model to hold results of facial
recognition request.



Praxyk Email - Set up SMTP server
so we can host praxyk.com email
addresses.

MV

Docs - Add script that can pull entire
wiki and compile it to HTML in the
docs/wiki-html directory.



Docker - Start process of
containerizing the POD environment.



MV

Clients - Begin adding features like
logging-in and setting up of
accounts.



Docker Image - Write script that
clones master branch in a container,
builds it, and if the build succeeds
saves the container to docker.



MV

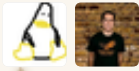
Website - Find a basic web server to
deploy website on.



Website - create dashboard that
shows user data



Website - figure out which backend to use



Website - Make a page that documents how to use api



Website - create login and register pages



Website - Implement way to grab tokens from API and store them as secure cookies in the users browser session.



API - Define API v1 in the Wiki.



NC

API - Implement the /results/ route



NC

Queue - get Redis DB set-up to hold queue data.



Queue - Define what queues we need (pod, transactions, account-actions?)



MV

Website - Design Basic website layout!



API - Implement the /users/ route



NC

Clients/Python - Implement Transaction class and add factories to Praxyk and User class.



Devops - Move DB from Google SQL to local MySQL instance to save \$\$



API - Create 'click to deploy' script.



Clients/Python - Implement custom exception class



NC

Multiple - Set up Shared Models directory so everyone can access the same database models.



DevOps - Beta v.1 launched, initial bugs fixed.



API - Implement a class hierarchy to handle the sub-routing.



NC

API - Implement the users/{id} route



NC

Database - MySql DB is up and exposed, users created.



Database - Server is up and running



DataBase - Design Transactions Table.



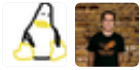
DataBase - Design users Table



OCR - File handles (decided against)



Website - implement new backend



OCR - get_string_from_image():
function written and exposed,
training to be improved separately



API - Implement Email Sending for
Account Confirmation



Travis-CI - Upgrade containers to
14.04 to fix libardmadillo build issue.



API - Make facial_rec route to take
images and recognize faces.



API - Implement /confirm/ route and
account-confirmation token
generation to let users confirm their
accounts.



Website - Make transaction card
more flexible in height so they don't
go into scroll mode on small screens



API - Fix bug where 404 occurs on
pod/ocr request



Praxyk Email - Get MX record set up
to @praxyk.com points to
@mail.praxyk.com



Client/Python - Implement POD-OCR route that lets user's upload images.



Queue - Add Facial Recognition task queue function



API - Make "api.praxyk.com" live and accepting minimal user requests.



Clients/Python - Implement User's Route



Website - Fix bug where after upload success you can't get the new transaction id from the returned json.



Website - Fix Bug where navbar drop-down goes behind <body> while in mobile-mode



Website - Add JS function to verify POD inputs before posting data.



Clients - Do high-level planning on client script and get it signed off.



NC

Database - Get a Redis DB up and exposed to the world.



POD - Build environment



NS

Decisions

Add Public/Private Option to
Transactions. Make Public ones free?