

Week of 5/3

Progress Updates

Everyone	<ul style="list-style-type: none">• Bug fixes - substitute invalid search query characters and limit query length (if query is entered via URL)• Wrote project evaluation together• Went through user guide and double-checked all use cases• Proofread programmers guide, product evaluation, project evaluation• Prepare files for submission
Shannon	<ul style="list-style-type: none">• ProductEval: worked on expert evaluation
Byron	<ul style="list-style-type: none">• ProductEval: worked on user evaluation
Nick	<ul style="list-style-type: none">• ProductEval: worked on testing

Objectives for Next Week

- Celebrate! Then sleep.

Week of 4/26

Progress Updates

Everyone	<ul style="list-style-type: none">• Wrote product evaluation questions and interviewed real potential users• Finalized and practiced the TigerSnatch presentation and demo• Prepared for and met with OIT to discuss the TigerSnatch service• Bug fixes and UI improvements based on user feedback<ul style="list-style-type: none">◦ Text fields, tooltips, course page links from Trades Tracker
Shannon	<ul style="list-style-type: none">• User's guide: worked on use cases for set up, Subscriptions, additional features• Programmer's guide: worked on boxes and arrows diagram, table for app endpoints, UI tier explosion (templates focused), story for design problem
Byron	<ul style="list-style-type: none">• User's guide: worked on use cases for Trades, admin• Programmer's guide: worked on UI tier (JS focused), story for two design problems
Nick	<ul style="list-style-type: none">• User's guide: worked on use cases for admin• Programmer's guide: worked on boxes and arrows diagram, table for app endpoints, table for processing (non-endpoint) and database tiers (Python focused), story for design problem• Implemented queryable system logging, an OIT request

Objectives for Next Week

- Compile user feedback and complete product evaluation guide
- Complete project evaluation guide
- Test use cases in user's guide
- Final testing of application
- Make sure files are ready to be submitted (proof read everything)

Week of 4/19

Progress Updates

Shannon	<ul style="list-style-type: none">• Created Activity page to display subscription and trade logs• Added Trades Tracker on dashboard to display sections that user wants to trade out of• Wrote backend method to find matches for a given user and courseid• Worked on Final presentation
Byron	<ul style="list-style-type: none">• Worked on client-side code and created necessary endpoints for Trades feature, including dropdown to select and save current section, buttons to find trade/remove yourself from trade, modal displaying potential trades, email template in gmail• Added endpoint to create logs when user tries to contact a trade• Added endpoint to fill a class (for facilitating testing email functionalities)• Work on Final presentation
Nick	<ul style="list-style-type: none">• Added ability to view a user's subscribed and Trades sections from admin panel• Added ability to toggle email notifications script from admin panel• Wrote backend method to add/remove users from a current section for a given courseid• Work on Final presentation

Objectives for Next Week

- Work on final presentation
- Testing on all features, particularly more extensive testing on Trades
- Work on written deliverables
- Gather feedback on Beta version

Status Meeting Notes

- Try breaking the text fields - copy & paste, weird characters, character length, SQL injection
- Informative error messages?
- Video is OK for blacklist if Dondero is OK

Questions

- Ask Sata about simulating the blacklist?

Timeline for Deliverables

- Fri, 4/23
 - Finish final presentation + work on script
- Sat, 4/24
 - Work on script
 - Start User's guide (writing + making script to fake data)
- Sun, 4/25
 - Practice final presentation
 - User's guide
- Mon, 4/26
 - Practice final presentation
 - Finish user's guide
- Tues, 4/27
 - Practice final presentation
 - Start Programmer's guide
- Wed, 4/28
 - Final presentation day!
 - Programmer's guide
- Thurs, 4/29
 - Come up with questions for Beta feedback
 - Start Product evaluation
 - Programmer's guide
- Fri, 4/30
 - Hold Beta feedback sessions
 - Product evaluation
- Sat, 5/1
 - Product evaluation
 - Start Project evaluation
- Sun, 5/2
 - Project evaluation
- Mon, 5/3
 - Final testing to make sure User's guide is good

Week of 4/12

Progress Updates

This week, the team worked on administrative tools (backend and frontend), improved UI, and performed a successful test run with a small group of real users during the course selection period. Here are the specifics of what we each worked on:

All	<ul style="list-style-type: none">• Ideated new stretch goals to replace the original MSS
Shannon	<ul style="list-style-type: none">• Created quick links panel for each course page so users can easily access Registrar's page / Princeton Courses for each course• Implemented user-level notification logging• For admin: wrote methods to turn on/off heroku maintenance mode, display basic information from our database for an admin, log any admin action in database and display them• Minor things: updated search to include more special characters with url encoding, changed all mentions of waitlists to subscriptions
Byron	<ul style="list-style-type: none">• Designed and set up UI for admin interface• Set up search feature for querying netids• Updated database methods and javascript functions• Restrict admin functions to select users only
Nick	<ul style="list-style-type: none">• Set up notification script cron job and implemented admin methods to remotely start/stop the notification script• Implemented front-end status indicator signalling to users whether or not TS is actively looking for section slots• Implemented user-level notification logging• Organized database.py (our largest code file) and made UI fixes

Objectives for Next Week

- Work on Trades feature
 - Activity page (Shannon & Nick)
 - Backend (Shannon & Nick)
 - Return potential swaps given netid and classid
 - Set up swap_out array for each classid in database
 - Frontend (Byron)
 - **Meet on Wednesday to sketch out UI together**
 - "My Enrolled Section" dropdown to select current section (that user is currently enrolled in)
 - Users always start out in zero swap_out arrays for a particular course

- Each user can only be in swap_out array for one section of a given course (e.g. P, C, B, U)
- User selects current section from dropdown → clicks submit button to store current section in database
 - If user previously submitted a current section (i.e. courseid exists as a key in users), delete them from that section's swap_out array before adding them to the new one
- "Find Matches" button + modal to display potential matches based on dropdown
- New data structures
 - swap_out array for each section, stored in enrollments
 - courseid: classid mapping for user's current section enrollment, stored in users
- Update tutorial guides once UI is mostly finished
- Outline final presentation
 - *Meet Monday afternoon to outline presentation*
- Plan questions for beta feedback session
 - Push to next week

Status Meeting Notes

- Next week is Beta demonstration
- Instructions to check that user has been added to blacklist (and removed from waitlists)
- Clarify class vs course in the admin page
- Change "waited-on sections" to subscriptions!
- Include screenshots of success
- Create an endpoint that only Dondero & Sata have access to - should run script that automates creation of fake data
- Instructions for Dondero & Sata go into User's Guide

Questions

- When should we start working on our written deliverables?
- To what extent do we need to guide you through simulating the functionality of Tigersnatch? (Should we write down step by step instructions on what to manipulate in the DB?)

Week of 4/5

Alpha Demo

- Go to <https://tigersnatch.herokuapp.com/>
- Click Get Started (do CAS authentication)
- Read through tutorial
- Search for COS courses
- Add yourself to a waitlist for a section in COS 126
- Return to the dashboard
- Remove yourself from that waitlist
- Logout

Progress Updates

All	<ul style="list-style-type: none">• Came up with feedback questions for alpha version• Met with 4 members of PAWCasso team and got feedback on our alpha• Based on feedback, came up with list of modifications• Thoroughly discussed implementation of mutual section swapl
Shannon	<ul style="list-style-type: none">• Show only full sections by default on each course page, and create checkbox that will show all sections• Restrict course search to alphanumeric characters + limit # characters in search string• Create tutorial page explaining how to use the app for first-time users of our app - added tutorial to nav• Small things: widen switch, move Notify me to second column, bold course number, add padding below search bar
Byron	<ul style="list-style-type: none">• Made table header sticky• Limit number of courses a users can be on at the same time• Added a loading phase when users add/remove themselves from a waitlist• Show waitlist size on course page• Re-format/added content to emails
Nick	<ul style="list-style-type: none">• Fixed bug in update_all_courses to handle zero-meeting edge case• Added quick-scroll links for mobile users• Added scroll-reset when performing a new search• Added tooltips to search bar, dashboard header, and notify-me column• Added confirmation toast dialogues when adding and removing from waitlist• Tightened all layouts to increase table size

- | | |
|--|--|
| | <ul style="list-style-type: none"> • New CSS for landing page |
|--|--|

Objectives for Next Week (Will be updated following this meeting)

- Everyone
 - Admin interface - will discuss features asap
 - ~~Recent Snatch activity displaying the n most recent slot openings for subscribed sections~~
 - ~~Change waitlists to unordered sets~~
 - ~~Change wording globally to use "subscribe", "Snatch notifications", etc.~~
 - Snatch Swap system (once Sata gets back to us)
 - ~~Quick links to Princeton Courses, Course Offerings, TigerHub~~
 - ~~URL encoding and allow more search characters~~

Status Meeting Notes

- For final evaluation, create fake course to test email functionality
- Not recommended to put app in public domain (prone to attacks)
 - Ask instructors first
- Waitlist ordering may be hindrance - should reconsider
- Allow departments to use waitlists?
- Nice touches
 - Customize add/remove waitlist toasts to show section
 - Show "last updated" time for each course
 - Change number of waitlists user can be on
 - Add quotation mark, %, & as eligible characters for search
 - Change custom "requested format" message
 - Add professor name

Questions

- Reasons for replacing MSS
 - After gathering feedback, many interviewees did not show enthusiasm for such a feature
 - Requires large (and fundamental) changes to the database and slow algorithm (refer to plan)
 - Subject to fraud & complications due to assumption that users are honest when saying what their current section is
 - Since course selection/add drop is coming up in the next few weeks, we want to shift our focus to deployment & refinement based on user feedback
- Additional stretch goals
 - Admin page and admin tools (e.g. global course refresh, single-user waitlist reset)

- Replace “swap sections” box with customized quick links (e.g. princeton courses, course offerings, tigerhub)
 - Filter to show only courses with full sections
- Will there be instructions about what sho

Week of 3/28

Progress Updates

This week, the team collectively worked on getting the UI to near production-level, in preparation for user feedback next week. Here are the specifics of what we each worked on:

Shannon	<ul style="list-style-type: none">• Worked on UI for search results (cards, hovering, highlight selection)• Fixed bugs in dynamically and independently updating search results (left panel) and course/dashboard (right panel)• Waitlist removal and cancellation: updated functionality as to work with Bootstrap switches, changed alert to Bootstrap modal, remove row from screen once switch is toggled off
Byron	<ul style="list-style-type: none">• Worked on Navbar design and functionality• About page UI• Dashboard/sections table sizing
Nick	<ul style="list-style-type: none">• Landing page UI• Error page UI• Initial UI for right-side dashboard and course search panel• Various UI tweaks and fixes

Objectives for Next Week

- Everyone
 - ⊖ Come up with feedback questions
 - Set up call with PAWcasso group for feedback exchange
 - ⊖ Discuss implementation of mutual section swap (over the weekend) and start it
 - Test for edge cases
 - ⊖ Decide logical path for next week alpha demo
- Shannon
 - ⊖ Filter for full sections in a course (checkbox)
 - ⊖ Restrict search to alphanumeric characters + limit # characters in search string
 - ⊖ Auto focus on page load
 - ⊖ Bold course number in search results
 - ⊖ Padding bw search results and search bar
- Byron
 - ⊖ Limit # courses that user can be on waitlist for (includes front end modal warning)
 - Show last updated time for each course page
 - Show waitlist size on course page
 - Fix the header in course and dashboard page table
- Nick

- ~~○ Upon new search/clicking on new entry, reset scroll bar to top~~
- ~~○ In mobile view, create dashboard anchor and auto scroll~~
- ~~○ Figure out database bug for new term~~
- ~~○ Warning on front page~~
- ~~○ Change global Database variable in app.py~~

Status Meeting Notes

- UI suggestions
 - ~~○ Sign in vs login (two login buttons)?~~
 - ~~○ Bold course number in search results~~
 - Fix the header in course page table
 - ~~○ Warning on front page~~
 - ~~○ Padding bw search results and search bar~~
- Alpha due next week - Sata will be testing app with a logical path (define it for him)
- Taking care of edge cases

Questions

- Flask app file-scope variables?

Week of 3/21

Progress Updates

Shannon	<ul style="list-style-type: none">• Added to Monitor module as to pull newest data for a course if it has been 2 minutes since a user has last clicked on the course page• Created method to pull data needed to construct table on Dashboard• Coded UI for Dashboard and Course table (including updating waitlist buttons to switches)• Implemented functionality to always show most recent search results on left panel while user navigates right side
Byron	<ul style="list-style-type: none">• Added endpoint that allows users to change their emails• Setup Atlas and migrated database• Setup Heroku hosting and deployed application
Nick	<ul style="list-style-type: none">• Created script (designed to be run as a cron job) that links the three key components of waited-on class enrollment polling - combines Monitor, Notify, and Waitlist• Coded UI for navbar, added general error page to catch server-side exceptions (preventing ugly crash messages on client side)• Added new utility scripts and methods that help with resetting and updating the database at different levels (soft and hard)

Objectives for Next Week

- Shannon:
 - Pull latest course information for dashboard
- Goal is to create near production-level UI for our MVP (based on mockups)
 - Landing page
 - Fix switch bug (use bootstrap modal instead of alert) - disable switch once turning off
 - On course pages, display waitlist size
 - Make search results into nicer table
 - Restrict minimum size of tables and search results and make scrollable
 - Course and dashboard page headers
 - Mini user guides at bottom of page
 - Nice error page
 - Fix colors (buttons, switches, etc.) and fonts
- Fix bugs and refactor code
- Prepare feedback questions for user demo of MVP
- Meetings
 - Saturday, 4:30-6:30

- TBD

Status Meeting Notes

Questions

- Do we need to be doing anything special in Flask? "This is a development server..."

```
[ TigerSnatch main python _exec_server.py 1234 (tigersnatch) ]
connecting to database...success
connecting to database...success
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://localhost:1234/ (Press CTRL+C to quit)
* Restarting with stat
connecting to database...success
connecting to database...success
* Debugger is active!
* Debugger PIN: 268-688-957
```

○

Week of 3/14

Progress Updates

Shannon	<ul style="list-style-type: none">• Wrote Waitlist module, which implements backend functionality of adding and removing user from a class waitlist in the database<ul style="list-style-type: none">◦ Linked Waitlist module to the client interface: for classes that are full, a button is displayed. If a user is already in the waitlist for the class, the button says "Remove Me" and clicking it will send an alert for confirmation, then make a request to remove the user from the waitlist if confirmed. If a user is not in the waitlist for the class, the button says "Add Me" and clicking it will send a request to add the user to this waitlist.
Byron	<ul style="list-style-type: none">• Added helper functions to database.py• Set up the feature of email delivery to the first student on the waitlist of a class, as well as the html template for the email.
Nick	<ul style="list-style-type: none">• Wrote Monitor module and associated helper classes/modules - this enables continual monitoring of all waited-on classes and generates summaries of waited-on classes that now have slots available<ul style="list-style-type: none">◦ Also updates the db for all waited on classes◦ Will comprise the first step of the cron job script that notifies users if a spot is available in one of their waited-on classes

Objectives for Next Week

- Shannon:
 - Updates course page on user click if it's been more than 2 minutes since the last click
 - Figure out how to update course page while continuing to display search results
 - Work on dashboard and course UI
 - Prepare feedback questions
- Nick:
 - Write script to check enrollments for waited classes (at most) every 2+ minutes, send notifications for available spots in classes, and remove user(s) from applicable waitlist(s)
 - Work on dashboard and course UI
 - Prepare feedback questions
- Byron
 - Figure out hosting for site and server (Heroku), database (Atlas)
 - Set up endpoint to edit email & phone number field
 - Enhance course search querying (strip spaces)
 - Prepare feedback questions

Status Meeting Notes

- OIT whitelist for heroku - done
- Push vs pull model for updating enrollments?
- Feedback once you have design aspect
- Need decent design for alpha build
- SMS not needed for COS333 aspect, but good if we want app to live on

Questions

- Suggestions about the waitlist monitoring aspect?
- Funding for SMS?

Week of 3/7

Progress Updates

Shannon	<ul style="list-style-type: none">• Helped interview would-be typical TigerSnatch users to get feedback on mockups and general functionality• Edited mockup based on user feedback (new version in directory)• Wrote endpoints to get and display courses matching search results, and display information about selected course from search results (added methods to Database module to accomplish this)• Started Jinja2 templates for dashboard & course pages, and search bar• Sketched out database schema
Byron	<ul style="list-style-type: none">• Set up CAS authentication for users• Created Landing page (where app redirects to if user is logged out)• Helped parse through user feedback and decide how to implement these changes• Sketched out database schema
Nick	<ul style="list-style-type: none">• Helped interview would-be typical TigerSnatch users to get feedback on mockups and general functionality• Began the Database module (communicates with local MongoDB server) for the purpose of adding basic course-related data in accordance with our database schema• Set up (mostly a copy-paste from the provided repo) the MobileApp module to manage queries with MobileApp API• Wrote script that populates applicable collections with all course data (i.e. "refreshes" TigerSnatch) from the most recent semester at the time of execution• Sketched out database schema

Objectives for Next Week

- SHANNON: Write the Waitlist module, which manages adding to and removing from all (ordered) waitlists in the database
- BYRON: Write the Notification module, which manages sending email and/or phone notifications to applicable users
- NICK: Write the Monitor module, which manages frequent and regular communication with MobileApp API to update enrollments for all waited-on classes
- *Secondary goal: figure out deployment (to AWS?)*
- *Secondary goal: begin Bootstrap CSS*

Status Meeting Notes

What we have now is already suitable prototype

Deploy to Heroku

- Heroku deletes database files that are updated at the end of the night -- BE CAREFUL!

Questions

- CAS does not work with 0.0.0.0 server
- When do other teams start the CSS part
- Cannot deploy to Heroku?

Week of 3/1

Progress Updates

Shannon	<ul style="list-style-type: none">• Worked on Project Overview document• Shared Team Directory + GitHub• Created Service Account for courses (MobileApp) API• Sketched out Figma mockup UI
Byron	<ul style="list-style-type: none">• Worked on Project Overview document• Sketched out Figma mockup UI
Nick	<ul style="list-style-type: none">• Worked on Project Overview document• Got access to courses (MobileApp) API• Sketched out Figma mockup UI

Status Meeting Notes

- Recruit users to commit to testing our app (talk to Sophie's group?)
- Add mockup for login landing page
- Run mockups by users
- Diagrams where you lay down database schema and their attributes (connect tables and show their relationships)
- Proper division of labor
- Grading policy: timeline document, preparation for meeting, next week's objectives (and major milestones)

Objectives for Next Week

- Recruit users to test our app
- Create mockup for login page
- Run mockups by users
- Create modules
- Diagram with database schema and set up database (MongoDB)
 - Populate database with course info from API
- Set up key endpoints (Flask)
- Set up CAS authentication (look at tutorial)