

BORG COLLECTIVE: VORTA TESTING AND UI IMPROVEMENTS

- Improve test coverage
- Implement profile sidebar
- Test on live Borg binary

ABOUT ME

Name

- Ted Lawson
- (Vorta IRC) tedlawson
- (GitHub) bigtedde

Education

- University of South Carolina, BS in Computer Information Systems, Graduated

Contact

- (Phone) {redacted}
- (E-mail) {redacted}

Time Zone

- {redacted}

Background

- After graduating from college, I entered into a career in restaurant management. However, at the start of 2023, I was fortunate enough to have the option to take a year off work to pursue a career change into a more technical field. The chance to participate in GSoC is incredibly exciting for me, as it would provide an excellent opportunity for growth and learning. Participating in GSoC would also allow me to gain valuable skills that I could apply to my future career in software development, and I am eager to work with this community to make meaningful changes to an application thousands of people use on a daily basis.

CODE CONTRIBUTION

Created Setting for Full Disk Access Check at Startup #1653

- <https://github.com/borgbase/vorta/pull/1653>
- Fixes issue #1287 - Allow disabling macOS Full Access prompt
- Added a setting in the misc tab that allows users to disable the Full Disk Access check at startup. Implemented unit tests to ensure proper functionality.

M1 Compatibility Note + Prettier #38

- <https://github.com/borgbase/vorta.borgbase.com/pull/38>
- Updated Vorta install instructions to include notes for users of Mac silicon to successfully install PyQt5. Also ran Prettier on the documentation and corrected typos.

Feature Request: Implement Profile Sidebar and new Settings interface #1677

- <https://github.com/borgbase/vorta/issues/1677>
- Created mockups and detailed outline for a feature request that involves moving the profile selector and current “misc” tab settings into a collapsible sidebar

PROJECT INFORMATION

Suborg

- Borg Collective

Project Abstract

- **Project 1: Improve test coverage**
 - <https://github.com/borgbase/vorta/wiki/Google-Summer-of-Code-2023-Ideas#improve-test-coverage>
 - Description: It's fun to add new features, but the actual work is maintaining them over time, as the code around it changes. This task would aim to increase the coverage output by the coverage tool from ~70% to ~80% by cleaning up existing tests, using parameterization

and adding more unit tests (as opposed to higher-level integration tests we use now).

- Task outline: Look at each Vorta package and module to find corresponding existing tests. Analyze the test coverage and determine which additional tests are needed. Then use consistent file naming for existing tests and add missing tests (especially unit tests).

- **Project 2: Implement profile sidebar**

- <https://github.com/borgbase/vorta/wiki/Google-Summer-of-Code-2023-Ideas#implement-profile-sidebar>
- Description: Vorta allows configuring profiles. A profile defines a number of sources, a backup destination, a schedule and some more settings. Currently there is a ComboBox for selecting a profile at the top of the window alongside buttons for adding or removing profiles. All profiles are active at the same time. However you can only edit the profile selected. This happens in the tabs below. The Misc tab is an exception. It contains the global settings that aren't associated with a profile.
- In this task you will implement a sidebar next to the tab widget that replaces the current way of selecting a profile to edit. The sidebar should be a list with the configured profiles that can be selected and a visually distinct Misc item that gives access to the global settings. The old Misc tab will be removed in favor of this new user interface. The new Misc item now provides space for an about view and a way to manage profiles configured in Vorta.
- Task outline: Implement the profile list that will be part of the sidebar. Add buttons for editing the profile list. Remove the old profile management GUI elements. Implement Misc items and the corresponding view. Move global settings to the new Misc view. Create an About tab and add it to the new Misc view. Implement a repository management tab and add it to the new Misc view. Write unit tests for the new GUI parts.

- **Project 3: Test on live Borg binary**
 - <https://github.com/borgbase/vorta/wiki/Google-Summer-of-Code-2023-Ideas#test-on-live-borg-binary>
 - Description: Currently we test on static mock files of Borg JSON output. That means our tests don't actually run Borg, but take some existing output. This is not optimal because we already support 3 major Borg versions and can only add mock files for one. This task would improve testing to run on multiple actual Borg versions/binaries.
 - Task outline: Research how to create multiple environments with different borg versions to run tests in. This might work differently in the CI (on the server) and local developer machines. Build a testing utility that can run the existing Vorta tests on multiple versions of Borg. (You can use a tool like Tox or write your own script.)

- **Backup projects**
 - Additional projects with lower priority, in case primary projects are delegated to other contributors.

 - **Enhance archive table and archive actions**
 - <https://github.com/borgbase/vorta/wiki/Google-Summer-of-Code-2023-Ideas#enhance-archive-table-and-archive-actions>
 - Vorta comprises a table listing the archives in a borg repository. There also is a button for renaming the selected archive in another dialog. This task should implement the capability to edit the archive name inline (in the table cell) without having to open another dialog. This task should also add a column showing whether an archive was created by the user manually or by the scheduler. There is also a button for refreshing the selected archive data from the repository. This button should work when selecting multiple archives too. Vorta allows mounting a selected archive. Implement the option to copy the

mount location to the clipboard, open the file manager at the mount location and a setting for doing that automatically after mounting. Currently the user has to select a folder to mount to. This can be very time consuming. This task should also add the possibility to 'quick mount' a repository into a temporary directory created by Vorta.

- Task outline: Add additional columns to the archive table. Implement edit functionality. Hide compact button for borg versions <1.2. Implement refreshing multiple archives. Implement copying mount path to the clipboard. Implement opening the mount location in default file explorer. Implement automatically opening mount location (+ corresponding setting). Implement quick mount action.
- **Search a file in diff and extract view**
 - <https://github.com/borgbase/vorta/wiki/Google-Summer-of-Code-2023-Ideas#search-a-file-in-diff-and-extract-view>
 - Vorta has a feature for comparing two archives (backup snapshots) and a feature for extracting specific files from an archive. The dialogue of these are very similar. They are composed of a list of files and some view modes options.
 - This task adds the option to filter the files shown through a search bar. At first one should be able to search for files containing the string entered but advanced search options like filtering by file size or other attributes could be implemented as well.
 - Task outline: Plan out how the search bar should work and how it would fit into the existing GUI. Implement the search functionality in the backend (FileItemModel). Implement the GUI addition. Write unit tests for the added code.
- **Backup to multiple repos from one profile**
 - <https://github.com/borgbase/vorta/issues/942>

- Some people may want to back up multiple repos from one profile. We should offer this functionality. For implementation, the repo drop-down could use checkboxes.
- Task outline: Design how the drop down should look and work. Implement functionality of backing up multiple repos to one profile. Write unit tests for the additional code.

DETAILED DESCRIPTION

- **Project 1: Improve test coverage**

- Current test coverage: 72% (See chart on next page)
- Test coverage goal: 80% - 85%
- To get from our current test coverage of 72% to our goal of 80% - 85%, I will be creating a collection of unit tests. These tests will target individual methods to ensure proper functionality. I will prioritize testing code that is currently not being covered, or needs to be covered more thoroughly. During this process, some of the methods may need to be refactored to make them more easily testable.
- Additionally, I would like to perform a general cleanup of our current testing package. This process would involve separating unit tests from integration tests, ensuring consistent name of tests, adding comments and additional documentation where it may currently be lacking, and organizing the files to have a more consistent structure.
- Success in the project would be accomplishing a test coverage of 80% - 85% from our current standing of 72%. It would also see a more consistent file naming and organization of our testing package.
- The chart on the next page illustrates our current test coverage broken down by file and percent of coverage for each file.

pytest --cov=vorta --cov-report term-missing

Test Name	#Lines	#Missed	%Coverage
/Users/ted/Desktop/vorta/src/vorta/__init__.py	1	0	100%
/Users/ted/Desktop/vorta/src/vorta/_main__.py	43	43	0%
/Users/ted/Desktop/vorta/src/vorta/_version.py	1	0	100%
/Users/ted/Desktop/vorta/src/vorta/application.py	222	86	61%
/Users/ted/Desktop/vorta/src/vorta/autostart.py	39	19	51%
/Users/ted/Desktop/vorta/src/vorta/borg/__init__.py	0	0	100%
/Users/ted/Desktop/vorta/src/vorta/borg/_compatibility.py	10	2	80%
/Users/ted/Desktop/vorta/src/vorta/borg/borg_job.py	192	29	85%
/Users/ted/Desktop/vorta/src/vorta/borg/break_lock.py	20	1	95%
/Users/ted/Desktop/vorta/src/vorta/borg/check.py	29	4	86%
/Users/ted/Desktop/vorta/src/vorta/borg/compact.py	32	6	81%
/Users/ted/Desktop/vorta/src/vorta/borg/create.py	107	32	70%
/Users/ted/Desktop/vorta/src/vorta/borg/delete.py	36	4	89%
/Users/ted/Desktop/vorta/src/vorta/borg/diff.py	27	4	85%
/Users/ted/Desktop/vorta/src/vorta/borg/extract.py	45	33	27%
/Users/ted/Desktop/vorta/src/vorta/borg/info_archive.py	40	31	22%
/Users/ted/Desktop/vorta/src/vorta/borg/info_repo.py	41	32	22%
/Users/ted/Desktop/vorta/src/vorta/borg/init.py	30	2	93%
/Users/ted/Desktop/vorta/src/vorta/borg/jobs_manager.py	72	15	79%
/Users/ted/Desktop/vorta/src/vorta/borg/list_archive.py	22	2	91%
/Users/ted/Desktop/vorta/src/vorta/borg/list_repo.py	37	2	95%
/Users/ted/Desktop/vorta/src/vorta/borg/mount.py	33	6	82%
/Users/ted/Desktop/vorta/src/vorta/borg/prune.py	41	4	90%
/Users/ted/Desktop/vorta/src/vorta/borg/rename.py	28	3	89%
/Users/ted/Desktop/vorta/src/vorta/borg/umount.py	32	5	84%
/Users/ted/Desktop/vorta/src/vorta/borg/version.py	19	12	37%
/Users/ted/Desktop/vorta/src/vorta/config.py	13	0	100%
/Users/ted/Desktop/vorta/src/vorta/i18n/__init__.py	41	16	61%
/Users/ted/Desktop/vorta/src/vorta/keyring/__init__.py	0	0	100%
/Users/ted/Desktop/vorta/src/vorta/keyring/abc.py	41	9	78%
/Users/ted/Desktop/vorta/src/vorta/keyring/darwin.py	51	2	96%
/Users/ted/Desktop/vorta/src/vorta/keyring/db.py	30	10	67%
/Users/ted/Desktop/vorta/src/vorta/keyring/kwallet.py	51	26	49%
/Users/ted/Desktop/vorta/src/vorta/keyring/secretstorage.py	44	40	9%
/Users/ted/Desktop/vorta/src/vorta/log.py	19	14	26%
/Users/ted/Desktop/vorta/src/vorta/network_status/__init__.py	0	0	100%
/Users/ted/Desktop/vorta/src/vorta/network_status/abc.py	34	12	65%
/Users/ted/Desktop/vorta/src/vorta/network_status/darwin.py	42	12	71%
/Users/ted/Desktop/vorta/src/vorta/network_status/network.py	122	43	65%
/Users/ted/Desktop/vorta/src/vorta/notifications.py	66	32	52%
/Users/ted/Desktop/vorta/src/vorta/profile_export.py	112	10	91%
/Users/ted/Desktop/vorta/src/vorta/qt_single_application.py	46	20	57%
/Users/ted/Desktop/vorta/src/vorta/scheduler.py	250	84	66%
/Users/ted/Desktop/vorta/src/vorta/store/__init__.py	0	0	100%
/Users/ted/Desktop/vorta/src/vorta/store/connection.py	41	3	93%
/Users/ted/Desktop/vorta/src/vorta/store/migrations.py	61	55	10%
/Users/ted/Desktop/vorta/src/vorta/store/models.py	125	2	98%
/Users/ted/Desktop/vorta/src/vorta/store/settings.py	13	1	92%
/Users/ted/Desktop/vorta/src/vorta/tray_menu.py	51	9	82%
/Users/ted/Desktop/vorta/src/vorta/updater.py	20	20	0%
/Users/ted/Desktop/vorta/src/vorta/utils.py	266	116	56%
/Users/ted/Desktop/vorta/src/vorta/views/__init__.py	0	0	100%
/Users/ted/Desktop/vorta/src/vorta/views/archive_tab.py	585	118	80%
/Users/ted/Desktop/vorta/src/vorta/views/diff_result.py	403	148	63%
/Users/ted/Desktop/vorta/src/vorta/views/export_window.py	55	1	98%
/Users/ted/Desktop/vorta/src/vorta/views/extract_dialog.py	302	107	65%
/Users/ted/Desktop/vorta/src/vorta/views/import_window.py	57	19	67%
/Users/ted/Desktop/vorta/src/vorta/views/main_window.py	211	44	79%
/Users/ted/Desktop/vorta/src/vorta/views/misc_tab.py	77	2	97%
/Users/ted/Desktop/vorta/src/vorta/views/partials/__init__.py	0	0	100%
/Users/ted/Desktop/vorta/src/vorta/views/partials/loading_button.py	22	1	95%
/Users/ted/Desktop/vorta/src/vorta/views/partials/tooltip_button.py	28	8	71%
/Users/ted/Desktop/vorta/src/vorta/views/partials/treemodel.py	325	56	83%
/Users/ted/Desktop/vorta/src/vorta/views/profile_add_edit_dialog.py	56	0	100%
/Users/ted/Desktop/vorta/src/vorta/views/repo_add_dialog.py	172	60	65%
/Users/ted/Desktop/vorta/src/vorta/views/repo_tab.py	210	42	80%
/Users/ted/Desktop/vorta/src/vorta/views/schedule_tab.py	142	7	95%
/Users/ted/Desktop/vorta/src/vorta/views/source_tab.py	220	57	74%
/Users/ted/Desktop/vorta/src/vorta/views/ssh_dialog.py	60	6	90%
/Users/ted/Desktop/vorta/src/vorta/views/utils.py	9	0	100%
TOTAL	5672	1589	72%

- **Project 2: Implement profile sidebar**

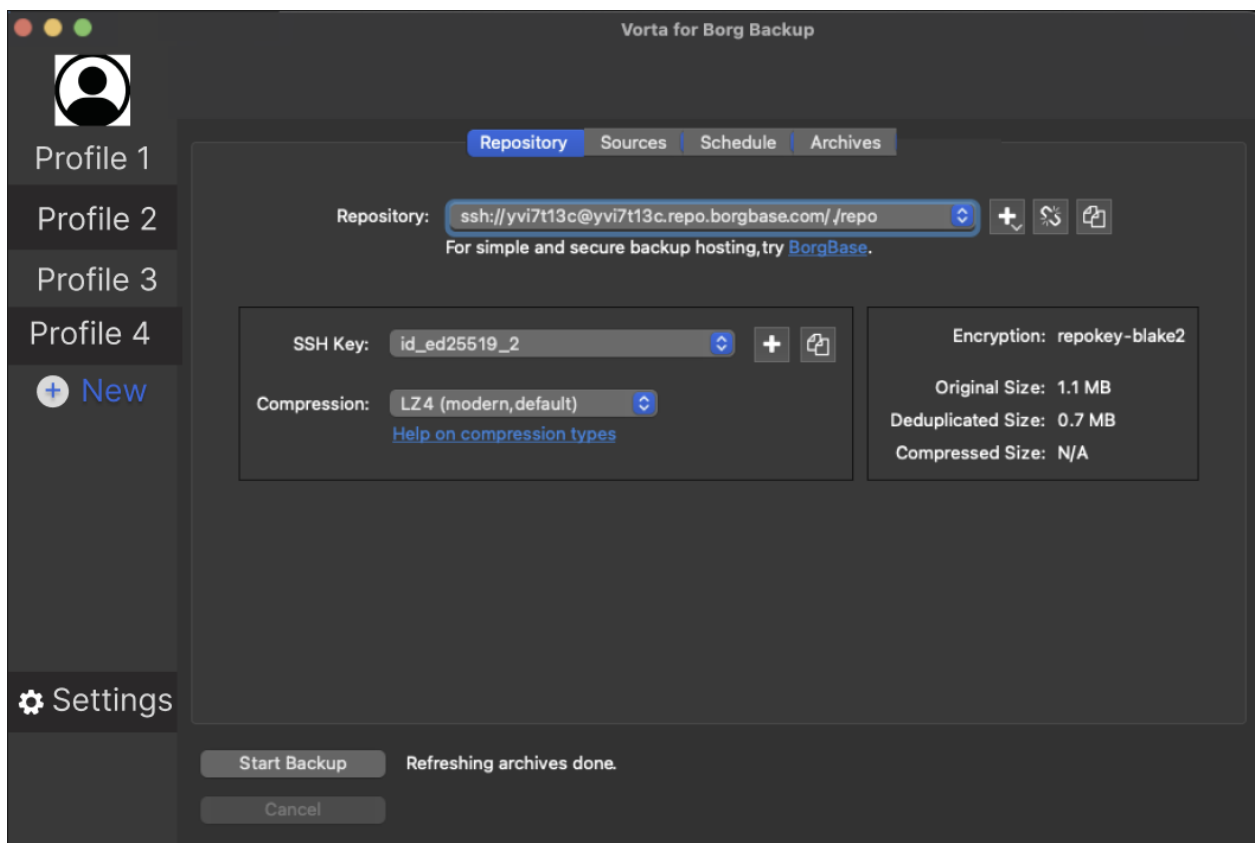
- Find my FR for this issue here:

<https://github.com/borgbase/vorta/issues/1677>

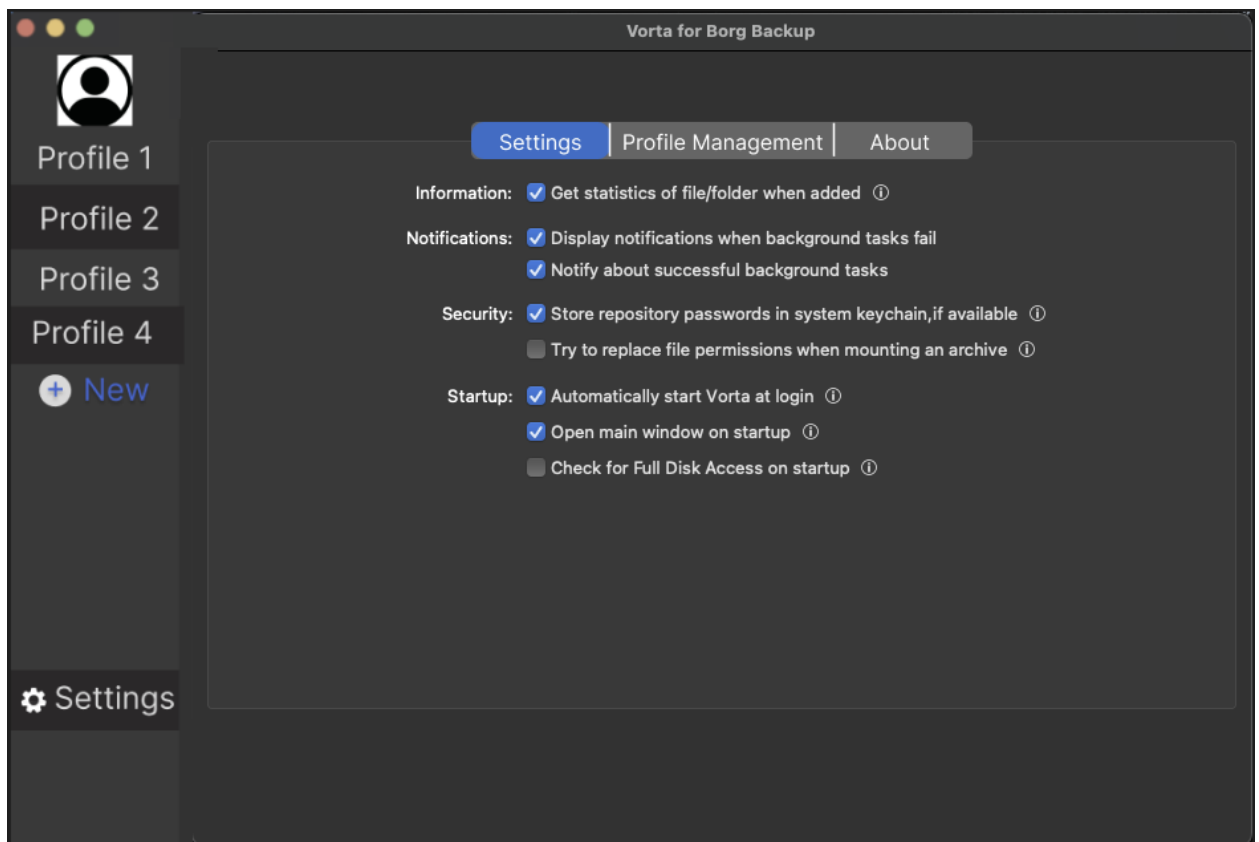
- This project consists of two main components:

1. Move the list of Profiles into a sidebar with options to add/edit Profiles list.
2. Create a visually distinct “Settings” button in this sidebar that takes the user to the global settings currently held in the Misc tab.

- Here is a mockup of how the Profile selector could look in the new sidebar. This sidebar could open and close by tapping on a hamburger menu or a profile icon, and when opened it reveals the different user profiles as well as an option to add new profiles. At the bottom there is a visually distinct “Settings” button.



- Clicking this “Settings” button will open up an interface with three tabs: “**Settings**”, “**Profile Management**”, and “**About**”
- The “**Setting**” tab will house all the settings from the current “Misc” tab. Other than changing the location of these settings nothing else here will change.
- The “**Profile Management**” tab will offer options to manage profiles configured in Vorta. Additional options in this tab will be discussed with mentors.
- The “**About**” tab will show basic information about the application.
- Below, I have created mock up of how the new “Settings” interface could look:



- Success for this project would be a working profile selector moved to a sidebar with edit/create functionality. It would also include a “Settings” button that takes the user to the aforementioned three tab interface with all functionality implemented and unit tested.

- **Project 3: Test on live Borg binary**

- One possible solution to test multiple versions of borg is to use Tox. Tox is a virtual environment management tool commonly used to simplify testing. For CICD, changes to [setup.cfg](#) may look something like this:

```
[tox:tox]
envlist = flake8, borg_v1, borg_v2, borg_v3

[testenv]
deps =
    pytest
    pytest-qt
    pytest-mock

[testenv:borg_v1]
envlist = py36,py37,py38
setenv = BORG_BINARY_PATH=/path/to/v1
commands = pytest

[testenv:borg_v2]
envlist = py36,py37,py38
setenv = BORG_BINARY_PATH=/path/to/v2
commands = pytest

[testenv:borg_v3]
envlist = py36,py37,py38
setenv = BORG_BINARY_PATH=/path/to/v3
commands = pytest
```

- In this sample setup config file, the tox will test specified versions of borg (in this case v1, v2, v3 against py36, py37, py38).
- BORG_BINARY_PATH will be passed in created in [github/workflows/test.yml](https://github.com/workflows/test.yml) and set as an OS environment variable.
- The test will call borg_binary_path with to call it's the borg's binary.

```
import os

borg_binary_path = os.environ.get('BORG_BINARY_PATH')
# will need to write a wrapper for calling the binary
# with good error handling
def borg_binary(args):
    command = [borg_binary_path, args]
    process = subprocess.Popen(
        command,
        stdin=subprocess.PIPE,
        stdout=subprocess.PIPE
    )
    output, error = process.communicate()
```

- To test it manually, the user can set BORG_BINARY_PATH to their local instance of borg.
- Success for this project would look like:
 - CI/CD running tests on live borg binaries with > 99% success rate.
 - Local development can run with little to no issues.
 - Include the option to run existing integration tests locally
 - Thorough documentation of this process

Weekly Timeline

Community Bonding Period

- Meet the team
 - Get to know the mentors and fellow contributors, as well as their strengths within this project
 - Discuss best times to connect, and set up a regular meeting schedule to ensure project progress goes to plan
- Discuss deliverables with community and mentors
 - Test cases needed to achieve target code coverage
 - Profile sidebar design and functionality
 - Settings page, Profile Management page, and About page design and functionality
 - Create issues for each project to get community feedback and refine ideas
- Learn more about skills needed to succeed with these projects
 - Tox
 - CICD
 - GitHub Actions
 - Unit Testing with PyTest
 - Interface design with Qt

Begin Milestone 1: Improve test coverage - 100 hours

- Week 1
 - Analyze test coverage and determine which additional tests are needed
 - Organize files into consistent structure
 - Discuss coverage needs with mentors
 - Begin writing tests to satisfy coverage goal
- Week 2
 - Evaluate progress and re-prioritize necessary tests
 - Continue writing tests to satisfy coverage goal

- Week 3
 - Conclude test coverage
 - Separate unit tests from integration tests
 - Conclude Milestone 1
 - Get head start on Milestone 2 if time permits

Begin Milestone 2: Implement profile sidebar - 125 to 175 hours

- Week 4
 - Begin implementing profile sidebar and functionality
 - Write unit tests for profile sidebar
- Week 5
 - Begin implementing GUI components for “Settings” page
 - Write unit tests for new GUI parts
- Week 6
 - Continue working on “Settings” page
 - Continue testing of components as they are completed
- Week 7
 - Conclude Milestone 2
 - Get head start on Milestone 3 if time permits

Begin Milestone 3: Test on Live Borg Binary - 100 to 175 hours

- Week 8
 - Find out a way to preconfigure/setup Vorta for testing
 - Learn what tests break with real binary (instead of mocked data, would it break with actual data?)
- Week 9-10
 - Modify CI/CD and ensure Borg’s binary files are being tested
 - Fix and broken tests
- Week 10
 - Conclude Milestone 3
 - Begin stretch goals if time permits
- Week 11

- Code freeze
- Complete tests and documentation of projects
- Week 12
 - Discuss completed projects with team
 - Make any final adjustments before submitting
- Final week
 - Submit final projects

OTHER COMMITMENTS

I have no prior commitments this summer, and can dedicate 40+ hours a week to these projects without distractions. If I am able to complete the three milestones outlined with some time remaining, I would like to also work on some stretch goals. I've outlined three "backup projects" in the project abstract, and if time permits I would like to begin working through as many of those as possible.

I did not apply to any other organizations, as I wanted to focus my efforts on Vorta due to my interests in UI and testing.

POST-GSoC

After the conclusion of GSoC I will continue to work on the stretch goals and any other work related to the projects completed during the Summer. It would be my hope to be able to return next year as a mentor and give back to the open source community in that way.