

**NRMRL QUALITY ASSURANCE PROJECT PLAN**

**Office of Research and Development**

National Risk Management Research Laboratory

*Land and Materials Management Division*

*Emerging Chemistry and Engineering Branch*



EPA NRMRL Technical Lead Person: Daniel L. Young

EPA NRMRL Project Manager: Michael Gonzalez

*Software Development*

*QA Category B*

*Extramural Research*

Revision Number: 1

Date: 6/25/2019

Prepared by:

Daniel L. Young, LMMD/ECEB

Contract: RMEERS EP-C-15-010

Work Assignment #: 3-41

sciNote Website: <https://ordscinote.epa.gov/>

sciNote Repository: <https://github.com/USEPA/sciNote>

Staging/Testing Only: <http://scinote.engineering4sustainability.com/>

Development: <http://localhost:3000>

[G-LMMD-0031316](https://qatrack.epa.gov/projects/show/7994/)

*ORD National Program, Project No. and Task No.: All*

*NRMRL QA Tracking ID: [G-LMMD-0031316-QP-1-0](http://qatrack.epa.gov/projects/show/7994/)*

**Approval Page**

|  |  |
| --- | --- |
| **QA Project Plan Title:** | https://camo.githubusercontent.com/457865f9d1e95b0535f93a1f266daa7146c4ab93/687474703a2f2f7363696e6f74652e6e65742f77702d636f6e74656e742f75706c6f6164732f323031352f31302f6c6f676f5f7363694e6f74655f66696e616c2e706e67 |
| **NRMRL QA Tracking ID:** | **G-LMMD-0031316-QP-1-1** |
| **If Intramural or Extramural, EPA NRMRL Project Approvals** | |
| Name: Technical Lead Person (TLP)  Daniel L. Young | Signature/Date: |
| Name: TLP’s Supervisor  Michael Gonzalez, SAB Branch Chief | Signature/Date: |
| Name: QA Manager  Steven Jones, NRMRL-IO | Signature/Date: |
| **If Extramural, Contractor Approvals** | |
| Name: Pegasus Manager/Lead  Raghuraman Venkatapathy | Signature/Date: |
| Name: Pegasus QA/QC Manager  June Silva | Signature/Date: |
| Name: Global Quality Corporation  Jacob Specht | Signature/Date: |

**Dr. Michael Gonzalez (ECEB Branch Chief)**

EPA, National Risk Management Research Laboratories

Land and Materials Management Division

Emerging Chemistry and Engineering Branch

26 W. Martin Luther King Drive (MS 483)

Cincinnati, Ohio 45268

**Dr. Daniel Young (QA Manager/Project Lead)**

EPA, Office of Research and Development

Land and Materials Management Division

Emerging Chemistry and Engineering Branch

26 W. Martin Luther King Drive (MS 483)

Cincinnati, Ohio 45268

**Mr. Steven Jones (QA Manager)**

EPA, Office of Research and Development

National Risk Management Research Laboratory

Immediate Office

26 W. Martin Luther King Dr.

Cincinnati, Ohio 45268

Pegasus info…

Global Quality info…

Contents

[PROJECT DESCRIPTION AND OBJECTIVES 7](#_Toc529177022)

[1.1 Software and Intended Application 7](#_Toc529177023)

[1.2 Project Objectives 7](#_Toc529177024)

[1.3 About the Tool 7](#_Toc529177025)

[1.4 Servers 7](#_Toc529177026)

[1.4.1 Internet Access 7](#_Toc529177027)

[1.4.2 Intranet Access 8](#_Toc529177028)

[1.5 Entering commands at prompt 8](#_Toc529177029)

[ORGANIZATION AND RESPONSIBILITIES 9](#_Toc529177030)

[2.1 Organization 9](#_Toc529177031)

[2.2 Research Staff 9](#_Toc529177032)

[2.2.1 Software Project Lead/WACOR 9](#_Toc529177033)

[2.2.2 Software Project Manager 9](#_Toc529177034)

[2.2.3 Alternate WACOR 9](#_Toc529177035)

[2.2.4 Quality Assurance Manager 9](#_Toc529177036)

[2.2.4 Anticipated contractor support 10](#_Toc529177037)

[2.3 Schedule 10](#_Toc529177038)

[2.3.1 Projected Timeline 10](#_Toc529177039)

[2.3.2 Future Planned Work 11](#_Toc529177040)

[FUNCTIONAL REQUIREMENTS 12](#_Toc529177041)

[3.1 Functions Software System 12](#_Toc529177042)

[3.1.1 Benefits to QA/QC, Researchers, Science 12](#_Toc529177043)

[3.2 Requirements for Functionality and External Interfaces 16](#_Toc529177044)

[3.3 Computer Hardware and Operating System Requirements 16](#_Toc529177045)

[3.3.1 Home screen shortcut 16](#_Toc529177046)

[3.3.2 Quick access 16](#_Toc529177047)

[SYSTEM DESIGN 17](#_Toc529177048)

[4.1 System Design Overview 17](#_Toc529177049)

[4.2 Components and Subcomponents 17](#_Toc529177050)

[4.3 Rationale for Selecting Hardware and Software Tools 17](#_Toc529177051)

[4.4 Evaluation Criteria 17](#_Toc529177052)

[4.4.1 Criteria 17](#_Toc529177053)

[4.4.2 LIMS/ELFN Evaluated 18](#_Toc529177054)

[4.5 Cost Analysis 100 Users 18](#_Toc529177055)

[IMPLEMENTATION 19](#_Toc529177056)

[5.1 Design Specifications 19](#_Toc529177057)

[5.2 Installation Required Software 19](#_Toc529177058)

[5.3 Functionality, External Interfaces, Performance, and Design 19](#_Toc529177059)

[5.3.1 Developer Environment IDE (Visual Studio Code) 19](#_Toc529177060)

[5.4 Staging & Testing (T&E) 22](#_Toc529177061)

[5.4.1 Installation instructions for T&E server: 23](#_Toc529177062)

[5.4.2 Production mode (in Docker containers) 23](#_Toc529177063)

[5.4.3 Updating sciNote on T&E Test Server: 24](#_Toc529177064)

[5.5 Deployment/Implementation (https://134.67.216.135) 24](#_Toc529177065)

[5.5.1 Initial Setup 24](#_Toc529177066)

[5.5.2 Production mode (in Docker containers) 25](#_Toc529177067)

[5.5.3 Updating sciNote on EPA Deployed Server 26](#_Toc529177068)

[5.6 Change Control 26](#_Toc529177069)

[5.7 Archiving Software/Repository 26](#_Toc529177070)

[VALIDATION, VERIFICATION, AND TESTING 27](#_Toc529177071)

[6.1 Testing Strategy 27](#_Toc529177072)

[6.2 Checking Correctness of Outputs 27](#_Toc529177073)

[6.3 Analysis, Evaluation, Review, Inspection, Assessment, and Testing 27](#_Toc529177074)

[DOCUMENTATION, MAINTENANCE, AND USER SUPPORT 28](#_Toc529177075)

[7.1 Project Documentation 28](#_Toc529177076)

[7.2 Maintenance and User Support 28](#_Toc529177077)

[7.3 Document Controlled Versions 28](#_Toc529177078)

[7.4 Training 28](#_Toc529177079)

[7.5 Getting Started with sciNote Electronic Lab Notebook 28](#_Toc529177080)

[7.6 Inviting Collaborators 28](#_Toc529177081)

[7.7 Managing User Privileges and Roles 28](#_Toc529177082)

[7.8 Creating Experiments 28](#_Toc529177083)

[7.9 Using the Search Function 28](#_Toc529177084)

[7.10 sciNote YouTube Channel 28](#_Toc529177085)

[REPORTING 30](#_Toc529177086)

[8.1 Deliverables 30](#_Toc529177087)

[8.2 Final Product(s) 30](#_Toc529177088)

[REFERENCES 31](#_Toc529177089)

[Appendices 32](#_Toc529177090)

[Appendix A: Test Install 33](#_Toc529177091)

[Hardware 33](#_Toc529177092)

[Software Installation 33](#_Toc529177093)

[Install Fedora 26 Mate 33](#_Toc529177094)

[Install sciNote 34](#_Toc529177095)

[Install VS Code 35](#_Toc529177096)

[Linux Distros Tested 35](#_Toc529177097)

[Raspbian Jesse 35](#_Toc529177098)

[HypriotOS 36](#_Toc529177099)

[Appendix B: sciNote Terms of Service 37](#_Toc529177100)

[Terms of Service Open Source Version – preferred 37](#_Toc529177101)

[Terms of Service Hosted Instance Version 37](#_Toc529177102)

[Appendix C: Fedora/RHEL Commands 38](#_Toc529177103)

[Basic Shell Commands 38](#_Toc529177104)

[Appendix D: sciNote 39](#_Toc529177105)

[About 39](#_Toc529177106)

[Build & run 39](#_Toc529177107)

[Testing 39](#_Toc529177108)

[Contributing 39](#_Toc529177109)

[License 39](#_Toc529177110)

[Appendix E: Git Commands 40](#_Toc529177111)

[Using Terminal 40](#_Toc529177112)

[Appendix F: Software Development & Use 41](#_Toc529177113)

[Free and open-source software (FOSS) 41](#_Toc529177114)

# PROJECT DESCRIPTION AND OBJECTIVES

## 1.1 Software and Intended Application

From idea to scientific publication – researchers can keep track of all EPA, ORD, and NRMRL required scientific data as well as QA/QC laboratory and field notebook records requirements. Enables researchers to structure work into Projects, Experiments, and Tasks. Research staff utilizing sciNote will be able to visualize their work progress by creating flexible workflows and obtain complete overview of research projects at any point in time. This document includes the Software Activity Plan for sciNote installation and maintenance activities on T&E staging and testing internet and EPA intranet server VM912, located in RTP, N.C.

## 1.2 Project Objectives

Based on the proposed work, select and install an Open Source LIMS with Electronic Lab and Field Notebook (ELFN) for researchers to store experimental data. The goal is to replace traditional pen and paper laboratory and field notebooks while meeting all EPA/ORD/NRMRL QA/QC requirements. LMMD/ECEB strongly support sustainable technologies and believe the future of science should and will be paperless, therefore we plan to use this tool to raise awareness for reducing paper waste within ORD. Strategic planning, awareness campaigns, and projects to replace paper-based processes are necessary in modern laboratories. For every sheet of paper researchers save in sciNote, staff will be awarded with one leaf. For each A4 page saved in sciNote they will be awarded with one Leaf. When researchers gather 100 Leaves they save 1 Tree Branch. After saving 100 branches researcher saves a Tree.

sciNote encourages scientific collaboration on all levels and contributes to the future of science that is open and independent. sciNote is an open source software platform that will be tailored to meet Agency and ORD/OSIM security standards once deployed to Virtual Machine assigned. Open source software code can be reviewed and improved by the ORD scientific community. Software is not tied vendor or owner of the software therefore, the ORD scientific community is free to develop add-ons, improve the software and its weak points and maintain it, and will be peer-reviewed once staging and testing is completed prior to deployment of RTP RHEL VM912.

## 1.3 About the Tool

sciNote is an Open source electronic lab or field notebook (ELN-EFN) that helps research staff manage laboratory work and stores experimental data on EPA intranet servers. sciNote is an electronic laboratory notebook (ELN), which enables scientists to replace their paper notebook with a digital lab notebook and offers exciting new possibilities for scientific data management. ECEB staffs are currently working to add LIMS functionality to sciNote as well as integration of ELN into the ORD approved QA TRACK tool developed by the same team working on this project.

## 1.4 Servers

The following servers are used to manage sciNote:

1. Development using Visual Studio Code. <http://localhost:3000>
2. Testing running on EPA T&E test server (smb://216.196.245.94/). <http://scinote.engineering4sustainability.com/>
3. Deployment RTP. <https://134.67.216.135/>

### 1.4.1 Internet Access

None. Internal US EPA ELN tool.

### 1.4.2 Intranet Access

The official EPA version of the tool is located on the RTP RedHat server. For internal EPA *Intranet access* @ <https://scinote.epa.gov>. *Please note: Administrative permission must be granted through ORD OSIM and your Manager to obtain access to source code on the RTP servers.*

## 1.5 Entering commands at prompt

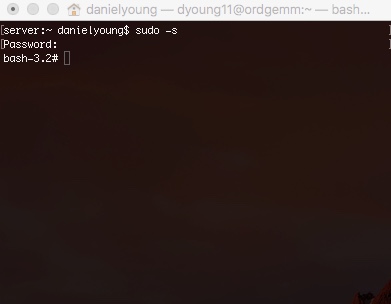
Throughout this administrative manual, you will see the following courier 10-point typeface in brackets [ sudo –s ]highlighted in grey for editing in the PuTTY or TERMINAL or from the command line prompt. It is recommended that you cut and paste the command, without brackets, into TERMINAL or at the command line prompt. On the right is an example screenshot of the sudo command pasted in the terminal window and executed:

Figure 1: Sudo Admin Privilege

End section

# ORGANIZATION AND RESPONSIBILITIES

## 2.1 Organization

NRMRL Land and Materials Management Division – Emerging Chemistry and Engineering Branch (ECEB). Research and Support staff in ECEB are listed in the Table below.

|  |  |
| --- | --- |
| Emerging Chemistry and Engineering Branch | |
| \*Michael Gonzalez, *Branc*h *Chief* | |
| Research Staff | |
| Daniel Young | John Abraham |
| Contract Support Staff | |
| Raghuraman Venkatapathy | Jacob Specht |

*\*Branch software management staff.*

## 2.2 Research Staff

### 2.2.1 Software Project Lead/WACOR

Dr. Daniel Young, Co-Principal Investigator (EPA) will have oversight of ELN tool. He will provide ideas and guidelines to cover the scope of this research proposal. Special Skills Required: The primary skills required to support this research effort requires extensive knowledge of software development using Ruby on Rails, Docker, and Make. Serves as the WACOR, software configuration manager (SCM) and project technical lead. All questions regarding this software should be sent to him @ [young.daniel@epa.gov](mailto:young.daniel@epa.gov). Contact: (513) 569-7451.

### 2.2.2 Software Project Manager

Dr. Michael Gonzalez, Branch Chief (EPA) will have oversight of ELN tool. He will provide ideas and guidelines to cover the scope of this research proposal, participating actively in the review of all reports and in the implementation of LIMS/ELFN.

### 2.2.3 Alternate WACOR

Dr. John Abraham serves as the alternate work assignment manager - contracting officer representative (WAM-COR). Email: [Abraham.John@epa.gov](mailto:Abraham.John@epa.gov). Contact: (513) 569-7124.

### 2.2.4 Quality Assurance Manager

Steven Jones, QA Manager (EPA), will provide independent QA oversight to ensure that planning and plan implementation are in accordance with the approved Quality Assurance Project Plan (QAPP). He will provide technical direction from a QA/QC perspective to EPA PIs on an as needed basis. He will enter QAPP and related products into ORD QA Track database.

### 2.2.4 Contractor support

It is anticipated the Software Project Lead will require some assistance to creating modules via add-ons to sciNote once a stable version has been released to ORD/NRMRL Staff. A work assignment with required documentation (FITARA, PWS, COR Certification, IGCE, etc.) will be submitted at that time to Division QAM for review and approval.

Contractor was not required to review or comment on this QAPP as it was written and approved by EPA staff and all work in this initial version will be completed by EPA personnel. An amended QAPP will be developed for any future work requiring contractor assistance for add-on module development such as a ChemDraw module, integration into ScienceHub, ORD QA TRACK, or other EPA Intranet applications. Contractor will be required to have minimum: (1) EPA PC with security access for VM912; (2) EPA Jump boxes (Asteroid & Supernova) using Remote Desktop Connection; and (3) EPA issued PUC Card.

## 2.3 Schedule

### 2.3.1 Projected Timeline

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Staff** | **Duties** | **Tasks** | **Date Start** | **Date Complete** | **Contract Support Hrs.** |
| **Michael Gonzalez, BC** | Provide management & technical direction. | Michael Gonzalez serves as the senior manager and subject matter expert. Email: [Gonzalez.Michael@epa.gov](mailto:Gonzalez.Michael@epa.gov). Contact: (513) 569-7998. | 4/2017 initial proposal | 4/2017 | 0 |
| **Daniel Young, PI** | Phase I: Proof of Concept. | Download & Text sciNote on Raspberry Pi Linux distro. <https://github.com/biosistemika/scinote-web> | 7/2017 | 10/2017 | 0 |
|  | Phase II: Repo | Request EPA GitHub account OSIM, load source code, info, etc. <https://github.com/USEPA/sciNote> | 10/15/2017 | 10/16/2017 | 0 |
|  | Phase III:  Install sciNote Developer SW | Prepare developer environment OS X, start on sciNote.  User Manual G-LMMD-0031316-MN-1-0.   1. Install command line developer tools (there are many resources online, like this). 2. Install Docker Toolbox as described here. 3. Inside CLI, run git clone https://github.com/biosistemika/scinote-web.git. 4. Run Docker Quick start Terminal (described here). 5. Inside this terminal, navigate to cloned Git folder. 6. Follow the Quick Start Guide above. 7. When opening sciNote in browser, instead of navigating to localhost:3000, navigate to <docker-machine-ip>:3000 (you can get the docker machine IP by running command docker-machine ip default). | In process as of 2/5/2018 | March, 2018 | 0 |
|  | Phase IV:  Installation Testing | Install ELN only on test server. | TBD | TBD | 0 |
|  | Phase V:  Installation RTP VM 912 | Install ELN only on EPA server. | TBD | TBD | 0 |
|  | Phase VI:  Connect ELN to QA Track | Computer Engineer (P4) Contractor support to interface sciNote with QA Track ELN module. | TBD | TBD | TBD |
|  | Phase VII: Add initial LIMS modules | Computer Engineer (P4) Contractor support to interface sciNote with NRM and balances. Initial LIMS modules added to ELN. | TBD | TBD | TBD |
|  | Phase VIII: Add additional LIMS modules | Add additional equipment per Branch Chief direction. | TBD | TBD | TBD |

### 2.3.2 Future Planned Work

1. Update QAPP version 1.1 – March 2018
2. Develop add-on modules after EPA Research staff comment period – June 2018

End section

# FUNCTIONAL REQUIREMENTS

## 3.1 Functions Software System

### 3.1.1 Benefits to QA/QC, Researchers, Science

Benefits to ORD/NRMRL staff include improved QA/QC control and tracking, costs savings in improved staff core functions, as well as supporting EPA’s Scientific Integrity Policy through adherence to applicable Agency information quality, quality assurance, and peer review policies and procedures, ensuring the Agency produces scientific products of the highest quality, rigor, and objectivity for use in policy decisions.

This section provides the documented, actionable, measurable, testable, and traceability of the sciNote tool defined to a level of detail sufficient for system design. sciNote is developed in Ruby on Rails. It makes use of Docker technology, so the easiest way to run it is inside Docker containers.

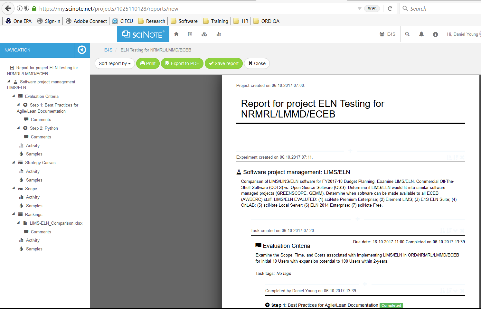
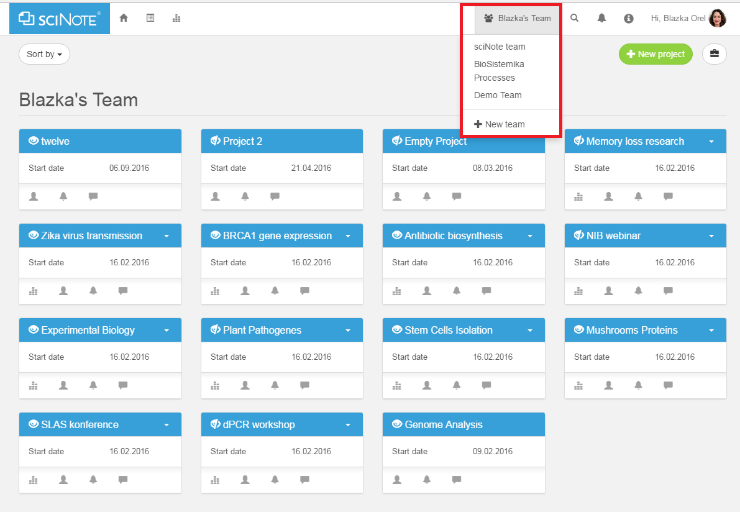
#### 3.1.1.1 Initial & long-term benefits implementing ELN

1. **Ease of Use.** The ELN (sciNote) was selected for many reasons, primarily ease of use and installation.
2. In both cases sciNote was far superior to the other OSS and COTS packages considered.
3. In addition, sciNote is Open Source (OSS) software and organization (US EPA) has full control to modify and use at no cost.
4. **Budget/Costs.** Open Source no cost ELN solution. Reduce Eliminate Costs Associated with LIMS/ELFN initial, annual user fees, and maintenance.
5. Number users supported. Unlimited number of users with no additional initial or annual costs per user.
   1. Training. No charge updated training is available @ <https://scinote.net/tutorials/>.
   2. Free Account available via <https://scinote.net/> for evaluation by research staff and use while local server is setup and tailored to meet ORD/NRMRL requirements. Offers many of the same features and functions to be installed for permanent support and use.
   3. Additional training would be implemented as EPA/ORD/NRMRL specific add-on modules are developed once implemented.
   4. Additional support is available with small annual fee for ‘*Premium Pro’* or ‘*Premium Enterprise’* support packages.
6. **Cloud or local installation.** Software can be deployed on a local server or supported via a cloud option such as <https://login.fr.cloud.gov> or via subscription through sciNote Pro or Premium accounts.
   1. Local installation to RTP server preferred as 2TB RHEL VM912 assigned and maintained by NRMRL/LMMD/ECEB.
   2. Cloud option available for future port to Agency cloud server if required by Agency directives.
   3. *Premium Pro*. 1 team (e.g. ECEB Lab). Unlimited users (members of the team). 1 TB storage. Separate instance on Amazon Cloud. Premium support (consulting and onboarding included). Electronic signatures (21 CFR Part 11). GLP Compliance.
   4. *Premium Enterprise*. Multiple teams. Unlimited users (members of the team). 1 TB storage per team. Separate instance on validated cloud or installed locally. Premium support (consulting and onboarding included). Electronic signatures (21 CFR Part 11). GLP Compliance.
7. **Capacity / Work Management.** 2TB server no charge provided by OSIM RTP. Cloud option available with 1TB storage per team on ‘*Separate instance on Amazon Cloud*’ or ‘*Separate instance on validated US EPA cloud (<https://login.fr.cloud.gov>) or installed locally’ (<https://134.67.216.135>)* on VM912 RedHat RHEL server RTP.
8. **Peer-Reviewed** / **Regular Updates & Maintenance of sciNote Code.** Open source software code reviewed and improved by the scientific community, independent, not tied to vendor or owner of the software.
   1. Large and fast growing scientific community is developing add-ons and improving the ELN.
   2. Software is peer-reviewed via improvements, independent verification & validation of source code, and collaboration with OSS software project management team [Gilson Inc. and sciNote LLC](https://scinote.net/docs/terms-of-service.pdf). <https://scinote.net/story-of-scinote/>.
9. **Additional add-on modules.** Automated Sample Management or upload scientific data to LIMS/ELFN (e.g., .csv, .xlxs).
10. **Proven Software Development Team.** Successfully developed and implemented several software projects. This includes an ELN Research Project assumed by OSIM (Dave Oberlin) entitled ‘[Electronic Laboratory and Field Notebook (ELFN) – Software](http://qatrack.epa.gov/projects/show/1720/)’. Additional software project design, development, and implementation includes:
    1. [QA TRACK](http://qatrack.epa.gov/projects/show/1099/) *“[Production](http://qatrack.epa.gov/)” and* “*[Staging](http://qatrackstaging.epa.gov/)*,” designed and developed by this team (with Thomas Holdsworth) deployed for NRMRL Fall 2015. OSIM assumed management 2016 at request of ORD DQAs for use ORD wide.
    2. [GEMM](http://qatrack.epa.gov/projects/show/7812/) “*[Production](https://ordgemm.epa.gov/)” and* “*[Staging](http://www.engineering4sustainability.com/).”*
    3. [GREENSCOPE](http://qatrack.epa.gov/projects/show/1083/) “*[Production](https://greenscope.epa.gov/)”* and “*[Staging](http://greenscope.engineering4sustainability.com)*.”
    4. [HF Database Support](http://qatrack.epa.gov/projects/show/1095/) -- NRMRL IO.

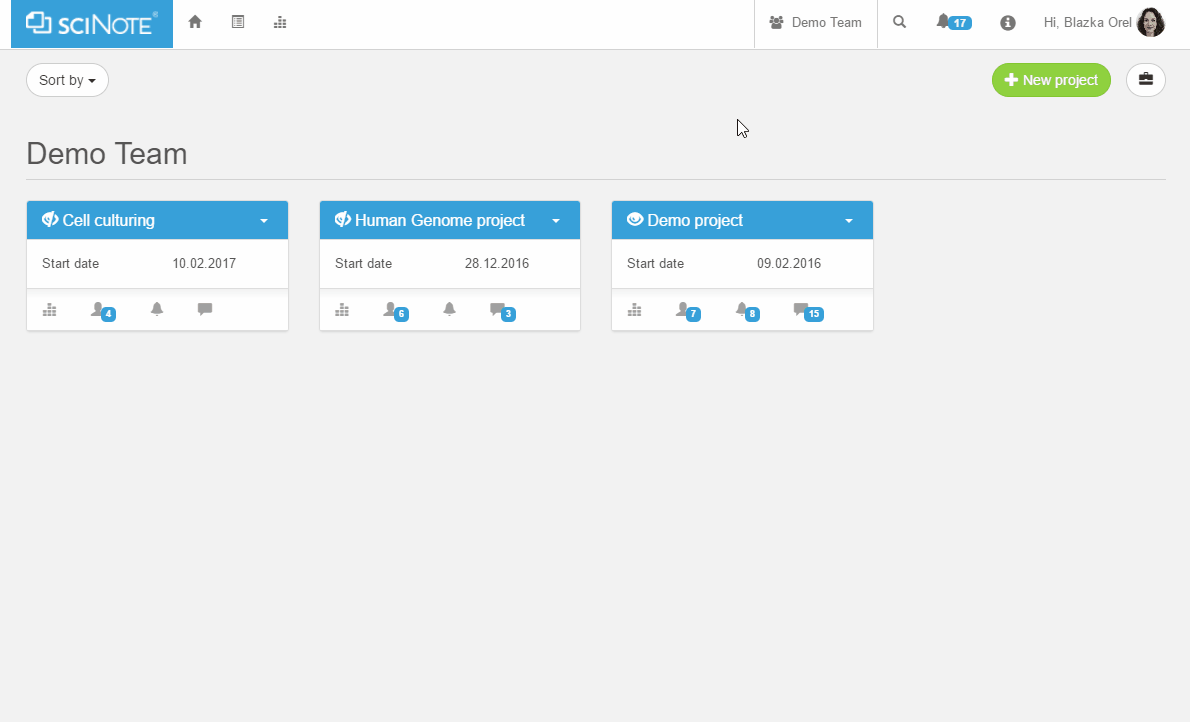
#### 3.1.1.2 Compliance with EPA, ISO, CFR and other Regulations

1. **[EPA QA / Compliance](https://www.epa.gov/quality/agency-wide-quality-system-documents).** Meets or exceeds EPA requirements for Laboratory records defined as an ‘*Electronic Record*,’ ‘*Paper Laboratory Record*,’ ‘*Project Record/Study File*,’ and ‘*Supporting Research Information*’ when properly used by research staff as defined in “*[ORD Policies and Procedures Manual Chapter 13](a.%09https:/intranet.ord.epa.gov/sites/default/files/media/about/manual/13.2%20Paper%20Laboratory%20Records.pdf)*[” § 13.2 “](a.%09https:/intranet.ord.epa.gov/sites/default/files/media/about/manual/13.2%20Paper%20Laboratory%20Records.pdf)*[Paper Laboratory Records](a.%09https:/intranet.ord.epa.gov/sites/default/files/media/about/manual/13.2%20Paper%20Laboratory%20Records.pdf)*” pages 13.2-3 & 13.2-4.
2. Supports Electronic Field Notebook (EFN) and Implementation of “EPA QA Field Activities Procedure ([QAFAP, CIO 2105-P-02-0](http://intranet.epa.gov/oei/imitpolicy/qic/ciopolicy/2105-p-02.pdf)).”
3. **[ISO/IEC 17025:2005 General requirements for the competence of testing and calibration laboratories](https://www.iso.org/obp/ui/" \l "iso:std:iso-iec:17025:ed-2:v1:en).** Laboratory customers, regulatory authorities and accreditation bodies may use [ISO/IEC 17025:2005](https://www.iso.org/obp/ui/" \l "iso:std:iso-iec:17025:ed-2:v1:en) in confirming or recognizing the competence of laboratories.
4. sciNote tracks and can be modified to meet most of the general requirements for the competence of testing and calibration laboratories.
5. sciNote will assist ORD/NRMRL manage laboratory (R&D) processes in compliance with ISO regulation.
6. The software itself cannot be 'compliant', but has features that allow Software Project Manager to set up, manage and record processes in compliance with the cited regulatory standards.
7. These types of standards typically require lots of traceability that sciNote offers.
8. **[ISO 14001:2015 Environmental Management Systems.](https://www.iso.org/standard/60857.html)** sciNote will assist ORD/NRMRL with tracking requirements for an environmental management system for annual EMS reporting to enhance its environmental performance.
9. Can assist with management of its environmental responsibilities in a systematic manner that contributes to the environmental pillar of sustainability.
10. Raises awareness for reducing paper waste. Strategic planning, awareness campaigns, and projects to replace paper-based processes are necessary in modern laboratories.
11. **[Premium Pro & Enterprise Versions](https://scinote.net/premium/).** Inexpensive option ($6K per year) includes:
12. GLP & 21 CFR Part 11 Compliance. sciNote will assist ORD/NRMRL manage laboratory (R&D) processes in compliance with GxP and Title 21 CFR Part 11 regulation that includes traceability and Chain of Custody (CoC), electronic signatures, audit trails, electronic witnessing, and system log records. Included in the 21 CFR Part 11 add-on.
13. CONSULTING, which includes: work on your use case, ELN good practices, advice on how to manage legacy protocols, advice on how to enter existing sample database, technical consulting for local installation.
14. ONBOARDING, online training, help with user management, fast and efficient implementation.
15. SOFTWARE SUPPORT, which includes general support (FAQ, e-mail support, live chat), as well as scheduled phone support.

#### 3.1.1.3 Important functions/features electronic lab notebook (ELN)

1. **Test/Analytical Method support.** Easy integration of open source tool to analytical methods. Linking EPA Methods in sciNote keeps methods up-to-date and easily update ELN when EPA changes ‘https://’ address without expensive costs to vendor for similar COTS ELN packages. Test/Analytical Method support examples:
2. [Air and Radiation](https://www.epa.gov/measurements/collection-methods" \l "1);
3. [Water](https://www.epa.gov/measurements/collection-methods" \l "2);
4. [Prevention, Pesticides, and Toxic Substances;](https://www.epa.gov/measurements/collection-methods" \l "3)
5. [Research and Development - Not Approved](https://www.epa.gov/measurements/collection-methods" \l "4);
6. [Solid Waste and Emergency Response](https://www.epa.gov/measurements/collection-methods" \l "5);
7. [Index to EPA Test Methods](https://www.epa.gov/measurements/collection-methods" \l "6).
8. **Search through & attach files.** Enables search options that allows researchers to search all scientific data and attachments.
9. Import files e.g., .txt, .xlxs, .csv, .pdf, .jpeg, .gif, and others.
10. Export e.g., . txt, .xlxs, .csv, .pdf, .jpeg, .gif, and others.
11. sciNote and Microsoft Office Online integration enables user to upload and edit Microsoft Word, Excel and PowerPoint documents in sciNote.
12. **Report writing/management.** Reporting, including statistical reporting with add-on module/template to be developed specific for NRMRL and/or ORD.
13. **Team Collaboration.** Via sciNote. Includes potential future integration of LIMS/ELFN with other IT systems, such as:
    1. [ScienceHub](https://sciencehub.epa.gov/sciencehub)
    2. [SDM](https://intranet.ord.epa.gov/science/scientific-data-management)
    3. [QA Track](http://qatrack.epa.gov/lab_notebooks/index/)
    4. [TechTracker](https://intranet.ord.epa.gov/science/techtracker)
    5. [RAPID](https://intranet.ord.epa.gov/nheerl/research-approval-planning-implementation-dashboard-rapid-0)
    6. [RAPS](https://usepa.sharepoint.com/sites/ORD_Work/Project-Planning/SitePages/Project%20Plans.aspx)
14. **Print & Export Scientific Data.** Print and export functions are built into the ELN. Example report item 6 above pdf format and indexable. Export data facilitates linking experimental data with analysis results in one location.
15. **Activity Log and Data Traceability.** Staff can view different team’s dashboard, click the Global team switch button at the top of the dashboard and select the team from the drop-down menu, where all the teams where user have been granted access to will appear. After that user are able to see the projects of the selected team.
    * 1. Data is the property of the Agency (US EPA). Therefore, in case user have transient researchers (ORISE, etc.), their notes will remain accessible to the Agency.
      2. The experiment date includes the creation and last modified date. These dates cannot be changed, since they signify a record of the change that has been made in sciNote (i.e. audit trail). This function may seem impractical when researcher wants to create an experiment that has already began or was done before user actually started using sciNote for data recording. Researcher can edit the experiment name by adding the date before the name permitting user to sort the experiments chronologically.
      3. Staff can filter projects on a dashboard by clicking the Global team switch button, choose to view projects from a certain team. Additionally, user can reorder the projects within a certain team by clicking the Sort by button near the top left corner of the dashboard.
      4. Detailed activity log i.e. all changes made by each user and the exact date and time are recorded in sciNote. For example, user can see who completed which protocol step and when, which samples have been used and which results have been uploaded, etc.
      5. At the same time, user can write smart annotations in the comments field or use tags to annotate user work. This way sciNote enables full traceability of data and therefore allows user to follow the GLP requirements.
      6. In order to access archived projects, user simply need to restore them using the archive tool box located top right next to new project tab .

## 3.2 Requirements for Functionality and External Interfaces

All the data within a single sciNote server instance is separated between organizations (i.e. teams). This means while a user can be a member of multiple teams (accessing them with individual username and password), data between teams is not visible. Everything a user does or uploads into sciNote, they perform as part of a specific team, which can either be their personal team or a team created by someone else to which they are granted access to by the administrator of that team.

If lab/branch requests to have different database instances running for each organization, then a separate instance of sciNote should be setup for each organization managed. In that case, a user will have a different user account for each of them.

### 3.2.1 RAP Integration

Please have Jake add the RAP integration as a required FUNCTION.

## 3.3 Computer Hardware and Operating System Requirements

Use sciNote on all, Laptop, Desktop, and Mobile devices. sciNote will adapt to every screen size, no matter how large or small, mobile or desktop.

### 3.3.1 Home screen shortcut

sciNote is not built as an app but user can easily add shortcut icon to their device’s home screen.

### 3.3.2 Quick access

After adding a home screen icon, user can quickly access sciNote via device’s browser.

End section

# SYSTEM DESIGN

## users-collaboration-close-up-scinote-transparent4.1 System Design Overview

Within sciNote, user can work on private projects, share projects with members of their team, invite others to collaborate. Each user has their own role with corresponding set of permissions within the team.

## 4.2 Components and Subcomponents

sciNote is developed in Ruby on Rails and makes use of Docker technology, so the easiest way to run it is inside Docker containers. Docker containers are based on open standards, enabling containers to run on all major Linux distributions and on Microsoft Windows.

## 4.3 Rationale for Selecting Hardware and Software Tools

sciNote was selected based on its open source specific approach to creating computer programs. “Open source projects embrace and celebrate principles of open exchange, collaborative participation, rapid prototyping, transparency and community-oriented development” – as stated by the team behind Opensource.com (supported by RedHat, the world’s open source leader). sciNote is an electronic lab notebook developed using open source software, that means the source code, the essence of the software, is available to the public. Anyone using sciNote is licensed to use, improve or change the software to fit the needs of their research staff. sciNote manages the base source code on GitHub and other research institutes may develop and share specific add-ons they created for their research teams. Open source software provides independence to its users providing more control over the ELFN. If compared to proprietary software that does not allow access to the source code, open source allows the community to step in, improve and fix the software, which increases the security and stability of the software. Open source is in direct relation with open collaboration. This can actually decrease the costs of the software and enable its original developers to offer it to the public as a free open source solution.

## 4.4 Evaluation Criteria

### 4.4.1 Criteria

Rankings were adequately based on objective data (i.e., cost estimates, whether ‘21 CFR part 11 Compliance\*’ etc., required/wanted, etc.) and are included in the chart below:

|  |  |
| --- | --- |
| Cost | Automated Sample Management |
| Test/Analytical Method support | EPA QA Compliance |
| Capacity / Work Management | Report Management |
| Compliance to ISO 17025 | Integration LIMS/other IT systems |
| Reporting, including statistical reporting | Traceability and Chain of Custody (CoC) |
| Number users supported | Training |
| 21 CFR part 11 Compliance\* | Electronic Signatures\* |
| Electronic Witnessing\* | Audit Trails\* |
| System Log Records \* | Going Live Level Effort |
| Satisfied clients or stakeholders |  |

### 4.4.2 LIMS/ELFN Evaluated

Included in the evaluation were both Open-Source Software (OSS) and Commercial Off-The-Shelf Software (COTS) LIMS/ELFN. OSS & COTS packages included: (1) [E-Notebook for Chemistry](https://www.cambridgesoft.com/Ensemble_for_Chemistry/ENotebookforChemistry/) with ChemBioDraw; (2) [sciNote Premium Enterprise](https://scinote.net/premium/); (3) [Element LIMS](https://www.promium.com/solutions/element-lims/); (4) [OnLIMS](http://www.onlims.com/product/online-eln-worksheet/); (5) [sciNote Local Server](https://github.com/biosistemika/scinote-web); (6) E4S ELN Suite (developed using previous STD Project entitled ‘[Electronic Laboratory and Field Notebook (ELFN) – Software](http://qatrack.epa.gov/projects/show/1720/)’); and (6) [sciNote – Free](https://my.scinote.net/users/sign_in). Chart 1 below provides rankings of each software evaluated.

*Chart 1: LIMS/ELFN Rankins*

## 4.5 Cost Analysis 100 Users

A cost analysis was performed on several Commercial Off-The-Shelf Software (**COTS**) and two Open-Source is computer software (**OSS),** LIMS/ELFN tools in chart below. Costs included the initial purchase or development, installation, and annual user and maintenance fees for each LIMS/ELFN compared.

**OSS**: Open-Source is computer software with its source code made available with a license in which the copyright holder provides the rights to study, change, and distribute the software to anyone and for any purpose.

**COTS**: Commercial Off-The-Shelf Software include any LIMS/ELFN available for purchase. Although COTS products can be used out of the box, in practice the COTS product must be configured to achieve the needs of the business and integrated to existing organizational systems. Extending the functionality of COTS products via custom development is an option, however this decision should be carefully considered due to the long term support and maintenance implications. Such customized functionality may not be supported by the COTS vendor, so brings its own sets of issues when upgrading the COTS product.

End section

# IMPLEMENTATION

## 5.1 Design Specifications

SciNote works on most browsers, some features may not function as expected on an older or unsupported version. Responsive design: Suitable for use on mobile devices.

sciNote supports the following browsers: Edge, Chrome, Firefox, Safari.

Microsoft Office Online: sciNote and Microsoft Office Online integration enables user to upload and edit Microsoft Word, Excel and PowerPoint documents in sciNote. Software license: Mozilla Public License version 2.0, Free & Open Source Software (FOSS) license.

## 5.2 Installation Required Software

The following software will require installation on all development machines as well as production and staging servers. (<https://support.scinote.net/hc/en-us/articles/115001389645-How-can-I-install-sciNote-on-Windows-server->)

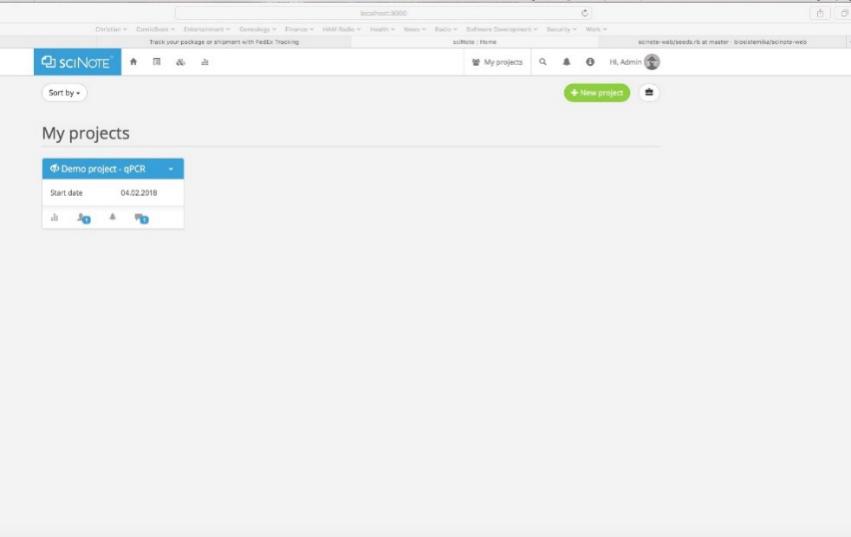
1. Programming language: Ruby on Rails 4.2.3.
   1. Windows 10 <http://railsinstaller.org/en> uncheck GIT if already installed; Check ruby version: Start - Run - type in [ cmd ] to open a windows console Type in [ ruby -v ] if not correct version run [ C:\.gem install rails \*v 4.2.3 ] Check version run git, [ git \*\*version ] ruby [ruby \*v ], and rails [ rails \*v ].5
   2. Fedora & RHEL: <https://rubygems.org/gems/rails/versions/4.2.3>.
2. Docker.
   1. Windows 10: <https://www.docker.com/docker-windows>.
   2. Linux Fedora: <https://www.docker.com/docker-fedora>.
   3. RHEL: <https://www.docker.com/docker-red-hat-enterprise-linux-rhel>.
3. JS framework.
   1. Bootstrap 3 <http://getbootstrap.com/>.
   2. jQuery <http://jquery.com/download/>.
4. VS Code. <https://code.visualstudio.com/download>.
5. Database service.
   1. PostgreSQL 9.4 <https://www.postgresql.org/download/>.
   2. Note: Installed on both production VM912 and staging T&E servers for GEMM & GREENSCOPE projects, will not require re-install unless moved to new VM.
6. Servers hosting.
   1. Staging, Testing, Development (<http://192.168.0.171>).
   2. Deployment/Implementation (<https://134.67.216.135>).
7. Source code: US EPA GitHub, <https://github.com/USEPA/sciNote>.

## 5.3 Functionality, External Interfaces, Performance, and Design

### 5.3.1 Developer Environment IDE (Visual Studio Code)

Install Command Line Tools in OS X (Without Xcode). Command Line Tool package gives terminal users many commonly used tools, utilities, and compilers, including make, GCC, clang, perl, svn, git, size, strip, strings, libtool, cpp, what, and many other useful commands that are usually found in default linux installations. Prerequisites:

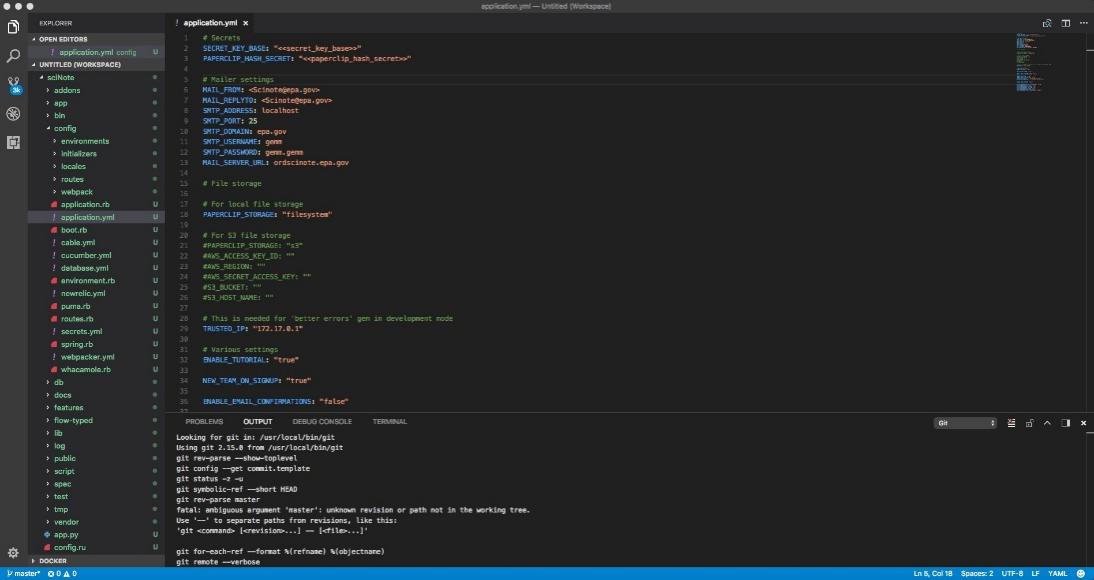
1. Open terminal and enter [sudo xcode-select --install]
2. Install make for other environments (i.e., T&E & RTP)
3. Install Docker download & install @ <https://store.docker.com/editions/community/docker-ce-desktop-mac>
   1. Create docker account: [dyoung11] password: [Evelynj1!]
   2. Login to docker cloud
4. Test Docker version [docker --version]
5. Test Docker installation [docker run hello-world]
6. Follow Quick start instructions @ [https://github.com/biosistemika/scinote-web/wiki/setup-guide#build-&-run](https://github.com/biosistemika/scinote-web/wiki/setup-guide" \l "build-&-run)
7. Login as @ <http://localhost:3000>
   1. Admin Login: [young.daniel@epa.gov]
   2. Password: [Evelynj1!]



**Figure 5‑1: Admin login sciNote localhost**

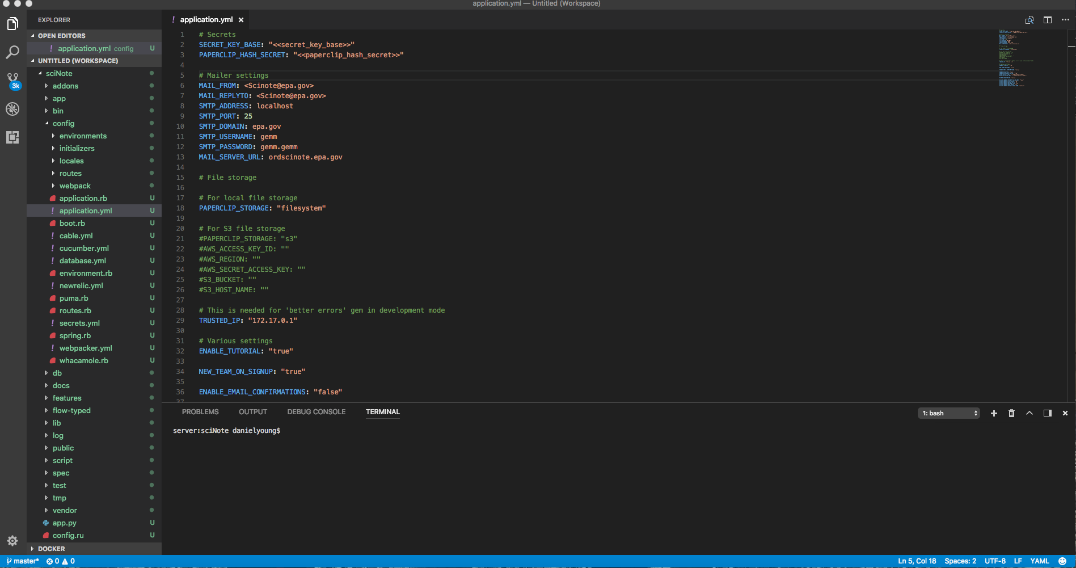
To start sciNote running on local development machine:

1. Open project in Visual Studio Code.



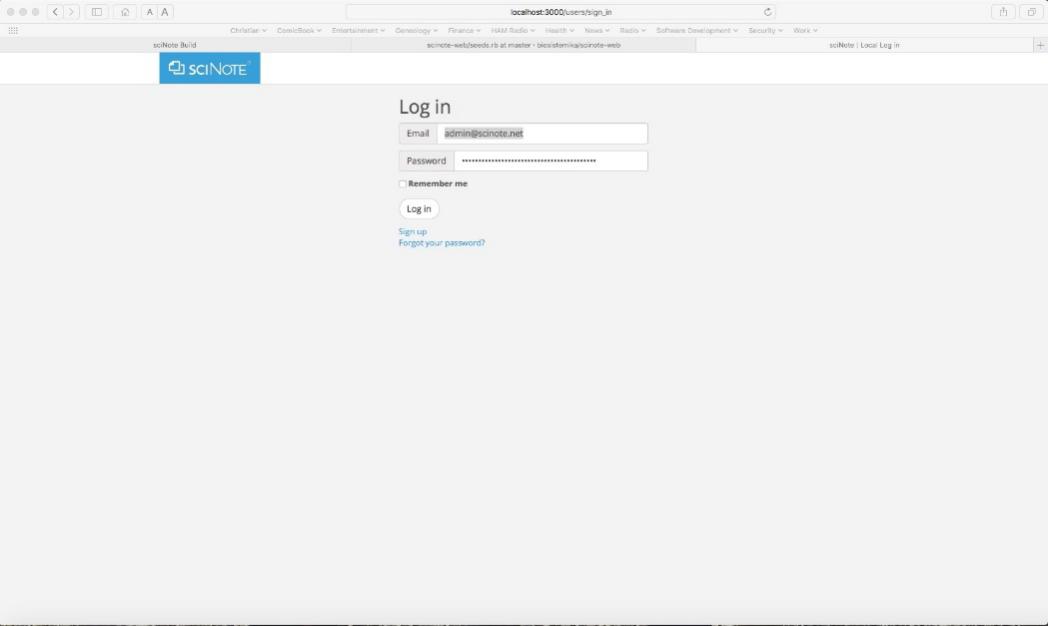
**Figure 5‑2: VS Code**

1. In VS Code open “Terminal” (click bottom error / warning counter – see screenshot below, right of git master, click either 0, or number if you have error / warnings).



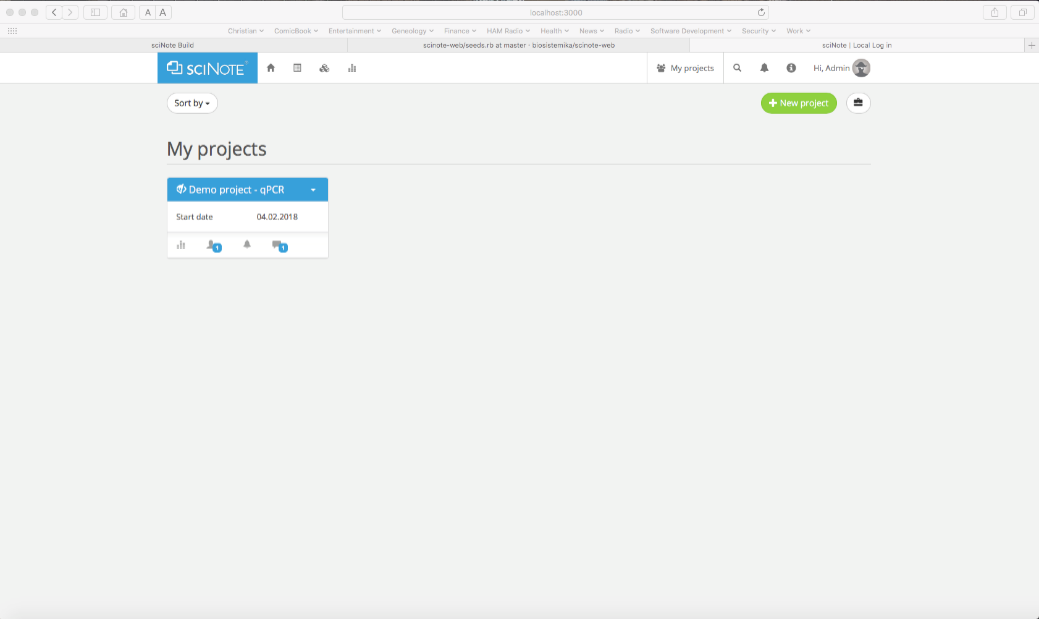
**Figure 5‑3: VS Code Terminal Window Open**

1. Start the server, run command [make run].
2. Wait until the server starts listening on port 3000.
3. Open browser and navigate to <http://localhost:3000>, login once web page opens.
4. Admin Login: [young.daniel@epa.gov]
5. Password: [Evelynj1!]



**Figure 5‑4: sciNote Admin Login Page**

1. After log in you should be in the sciNote tool.



**Figure 5‑5: sciNote Running localhost**

1. Once sciNote runs on localhost, logout, you can begin development / changes to code in VS Code.

## 5.4 Staging & Testing (T&E)

Testing will be performed on the T&E E4S server. For setting up <http://scinote.engineering4sustainability.com/>, make an entry similar to the ones for greenscope and gemm on the namecheap website. It takes about an hour to propagate and become functional. On the server side, you will have to modify the httpd config file (/etc/httpd/conf/httpd.conf), and make entries similar to greenscope and gemm. Create Staging Server web URL. <https://www.namecheap.com/>:

1. Login to www.namecheap.com
2. Select Domain List
3. Select Advanced DNS
4. Click + ADD NEW RECORD -- [CNAME Record] name [scinote]
5. Create URL namcheap website (i.e., GEMM, GREENSCOPE)

### 5.4.1 Installation instructions for T&E server:

1. Login via Remote desktop (asteroid.rtpnc.epa.gov:8472).
   1. Note: PUC card required.
2. Open MobaXterm.
3. Connect to T&E server @ [dyoung11@engineering4sustainability.com](mailto:dyoung11@engineering4sustainability.com)
4. Install the following prerequisite SW:
   1. Install make
   2. Install docker
   3. Install docker-compose
   4. Install docker engine
5. Change directory <cd /var/www/html>
6. Create sciNote folder <sudo mkdir sciNote>
7. Change directory <cd /var/www/html/sciNote>
8. Clone this Git repository in sciNote folder <git clone https://github.com/USEPA/sciNote.git>.
9. In sciNote folder, run the following command: <make docker>. This can take a while, since Docker must first pull an image from the Internet, and then install all necessary Gems required by sciNote.
10. Once the Docker image is created, run <make cli> command. Once inside the running Docker container, run the following commands: <rake db:create> - <rake db:migrate> - <rake db:seed>. This should initialize the database and fill it with (very minimal) seed data.
11. Exit the Docker container by typing <exit>.
12. To start the server, run command <make run>. Wait until the server starts listening on port 3000.
13. Inside another CLI window (from sciNote folder), run command <make worker>. This command should start the background worker process which is used by sciNote for a lot of demanding tasks.

### 5.4.2 Production mode (in Docker containers)

1. New functionality for running in production mode was introduced with sciNote version 1.10.0. In order to run sciNote application in production inside Docker you'll need the following: docker engine (version 1.12.0+), docker-compose (version 1.6.0+).
2. Set environment variables in <production.env> file; variables are the same as described in Environmental variables section, the only differences are: format of records is changed - use <KEY=VALUE> record format; set <DATABASE\_URL=postgresql://postgres@db/scinote\_production> without quotation marks.
3. The mentioned variable is used to configure database connection and by default it contains credentials to connect to the PostgreSQL database inside <db> container; you will need to change this variable if you decide to use another DB or set password for default db user.
4. You can generate default (minimal) <production.env> file with the following command: <make> config-production. This command will generate <SECRET\_KEY\_BASE> and <PAPERCLIP\_HASH\_SECRET> variables automatically.
5. After setting all the required variables you need to build the Docker image with <make docker-production>; this command will install all needed Gems and precompile assets.
6. When the above build is finished, database needs to be initialized (if you're performing a new install). This can be done by running <make database-production>. This will initialize the database and start a running process; the database will be hosted in the <scinote\_db\_production> container. This container is configured to keep all database files in the persistent Docker volume named <scinote\_production\_postgres>, which is (by default) placed in </var/lib/docker/volumes> of the host system.
7. There is another persistent Docker volume called <scinote\_production\_files>; this container is used for storing generated and uploaded files if you configure local file storage.
8. All these volumes will be kept safe in case you delete your Docker containers. Running <docker rm> with <-v> flag, however, will delete these volumes (same thing happens if you run <docker-compose down> with <-v> flag).
9. Background jobs worker will be running in another separate container called <scinote\_jobs\_production>.
10. Start sciNote with the following command: <docker-compose -f ./docker-compose.production.yml up>. Or you can use docker-<compose -f ./docker-compose.production.yml up -d> to start the server in background mode. Please consult docker-compose documentation for the rest of supported commands.
11. In short, the sequence of commands to setup sciNote (for fresh install) would be:

* <make config-production>
* <make docker-production>
* <make database-production>
* <docker-compose -f ./docker-compose.production.yml up>

### 5.4.3 Updating sciNote on T&E Test Server:

The following steps outline the process to stage updated versions of SCINOTE on the 8TB CLOUD server for testing and verification or backup prior to installing to root.

1. Xxx
2. xxx

## 5.5 Deployment/Implementation (<https://134.67.216.135>)

This section describes a step-by-step installation of sciNote version update from GitHub to the RTP RHEL server.

### 5.5.1 Initial Setup

1. Login via Remote desktop (134.67.216.135), open MobaXterm.
2. Install make
3. Install docker
4. Install docker-compose
5. Install docker engine
6. Change directory <cd /var/www/html>
7. Create sciNote folder <dzdo mkdir sciNote>
8. Change directory <cd /var/www/html/sciNote>
9. Clone this Git repository in sciNote folder <git clone https://github.com/USEPA/sciNote.git>.
10. In sciNote folder, run the following command: <make docker>. This can take a while, since Docker must first pull an image from the Internet, and then install all necessary Gems required by sciNote.
11. Once the Docker image is created, run <make cli> command. Once inside the running Docker container, run the following commands: rake <db:create> - <rake db:migrate> - <rake db:seed>. This should initialize the database and fill it with (very minimal) seed data.
12. Exit the Docker container by typing <exit>.
13. To start the server, run command <make run>. Wait until the server starts listening on port 3000.
14. Inside another CLI window (from sciNote folder), run command <make worker>. This command should start the background worker process which is used by sciNote for a lot of demanding tasks.

### 5.5.2 Production mode (in Docker containers)

1. New functionality for running in production mode was introduced with sciNote version 1.10.0. In order to run sciNote application in production inside Docker you'll need the following: docker engine (version 1.12.0+), docker-compose (version 1.6.0+).
2. Set environment variables in <production.env> file; variables are the same as described in Environmental variables section, the only differences are: format of records is changed - use <KEY=VALUE> record format; set <DATABASE\_URL=postgresql://postgres@db/scinote\_production> without quotation marks.
3. The mentioned variable is used to configure database connection and by default it contains credentials to connect to the PostgreSQL database inside <db> container; you will need to change this variable if you decide to use another DB or set password for default db user.
4. You can generate default (minimal) <production.env> file with the following command: <make> config-production. This command will generate <SECRET\_KEY\_BASE> and <PAPERCLIP\_HASH\_SECRET> variables automatically.
5. After setting all the required variables you need to build the Docker image with <make docker-production>; this command will install all needed Gems and precompile assets.
6. When the above build is finished, database needs to be initialized (if you're performing a new install). This can be done by running <make database-production>. This will initialize the database and start a running process; the database will be hosted in the <scinote\_db\_production> container. This container is configured to keep all database files in the persistent Docker volume named <scinote\_production\_postgres>, which is (by default) placed in </var/lib/docker/volumes> of the host system.
7. There is another persistent Docker volume called <scinote\_production\_files>; this container is used for storing generated and uploaded files if you configure local file storage.
8. All these volumes will be kept safe in case you delete your Docker containers. Running <docker rm> with <-v> flag, however, will delete these volumes (same thing happens if you run <docker-compose down> with <-v> flag).
9. Background jobs worker will be running in another separate container called <scinote\_jobs\_production>.
10. Start sciNote with the following command: <docker-compose -f ./docker-compose.production.yml up>. Or you can use docker-<compose -f ./docker-compose.production.yml up -d> to start the server in background mode. Please consult docker-compose documentation for the rest of supported commands.
11. In short, the sequence of commands to setup sciNote (for fresh install) would be:

* <make config-production>
* <make docker-production>
* <make database-production>
* <docker-compose -f ./docker-compose.production.yml up>

### 5.5.3 Updating sciNote on EPA Deployed Server

The following steps outline the process to stage updated versions of SCINOTE on the RTP RHEL server for deployment.

1. Xxx
2. Xxx
3. xxx

## 5.6 Change Control

Additional add-on modules developed in-house or downloaded from GitHub repository (<https://github.com/biosistemika/scinote-web>) will undergo thorough testing prior to a new release updated on US EPA GitHub Branch: master (<https://github.com/USEPA/sciNote>). New development will be handled via local branch GitHub as well as change control via US EPA, Software Configuration Management (GitHub) tool.

sciNote's codebase uses various code styling conventions. To maintain this practice, all contributions must follow the below code styling rules provided by various linter configuration files:

1. Ruby - RuboCop - .rubocop.yml.
2. JavaScript - ESLint - .eslintrc.json, .jsbeautifyrc.
3. Sass - scss-lint - .scss-lint.yml.

Developer suggests using VS Code as editor for consistency in development and changes to source code.

## 5.7 Archiving Software/Repository

<https://github.com/USEPA/sciNote>.

End section

# VALIDATION, VERIFICATION, AND TESTING

## 6.1 Testing Strategy

In current version, only model tests are implemented for sciNote. To execute them, call rake test:models.

## 6.2 Checking Correctness of Outputs

Delivery via browser see 6.1 above.

## 6.3 Analysis, Evaluation, Review, Inspection, Assessment, and Testing

During staging tests will be performed to ensure attachments, reports, and scientific data are properly stored and retrievable via PostgreSQL database on EPA Servers (T&E Staging and RTP RHEL). For issue reporting, please visit Jira page. It is an open source Jira that requires no sign-up (user can report issues as anonymous user). Detailed instructions about issue reporting are documented there.

End section

# DOCUMENTATION, MAINTENANCE, AND USER SUPPORT

## 7.1 Project Documentation

Requirements and design document, configuration maintenance plan, operations manual, source code, user’s guide.

## 7.2 Maintenance and User Support

FAQs, Video Tutorials, and additional information on use can be found @ <https://support.scinote.net/hc/en-us>.

## 7.3 Document Controlled Versions

Software Development Life Cycle (SDLC) tracked on US EPA GitHub @ <https://github.com/USEPA/sciNote>.

## 7.4 Training

Introduction to sciNote and its main functionality can be viewed @ <https://youtu.be/0vqDZEGmuTE>.

## 7.5 Getting Started with sciNote Electronic Lab Notebook

This tutorial will walk you through the basics of getting started with sciNote, open source electronic lab notebook. It will show you how to create projects, how to restore them from the archive, and how to view your activity. <https://youtu.be/rYI-88KS44Q>.

## 7.6 Inviting Collaborators

This tutorial will walk you through the basics of inviting collaborators to work on common Projects inside of sciNote, open source electronic lab notebook. It will show you how to invite new users, how to communicate with them, and how to collaborate on different Projects. <https://youtu.be/SVCkoo5Y144>.

## 7.7 Managing User Privileges and Roles

This tutorial will walk you through the basics of managing sciNote Electronic Lab Notebook user privileges and roles. It will show you which user authorities there are at the Team, Project and Experiment levels, as well as how to reassign and remove user roles. <https://youtu.be/w2D5w33A6Ao>.

## 7.8 Creating Experiments

This tutorial will walk you through the basics of creating Experiments in sciNote. It will show you how to create new Experiments inside of individual Projects and how to clone and move them between different Projects. <https://youtu.be/1BIPT-jI8mk>.

## 7.9 Using the Search Function

This tutorial will walk you through the basics of using the search function in sciNote, open source electronic lab notebook. It will show you how to find your data with the help of the search bar and will explain different types of search results. <https://youtu.be/XwriEdlUqGc>.

## 7.10 sciNote YouTube Channel

New tutorials are periodically uploaded to the sciNote Open Source ELN YouTube channel. <https://www.youtube.com/channel/UC0blhye3794sGGbcBypam_Q>.

End section

# REPORTING

## 8.1 Deliverables

Final ELN/LIMS delivered via browser on PC or Tablet device.

## 8.2 Final Product(s)

Software, user documentation, user interface available via tool (<https://ordscinote.epa.gov/>)or GitHub (<https://github.com/USEPA/sciNote>).

End section

# REFERENCES

1. sciNote – Free Open Source Electronic Lab Notebook. Easiest way to manage your scientific data. <https://scinote.net/>. Accessed 10/18/2017 @ 12:09 P.M. E.S.T.
2. biosistemika/scinote-web. <https://github.com/biosistemika/scinote-web>. Accessed 10/18/2017 @ 12:10 P.M. E.S.T.
3. US EPA ORD QA TRACK, project 'Laboratory Information Management System - Electronic Laboratory and Field Notebook' <http://qatrack.epa.gov/projects/show/7994/>.
4. USEPA/sciNote [Private GitHub Repository] <https://github.com/USEPA/sciNote>.
5. Ruby on Rails: An Introduction. Software Installation for Windows Users. <https://www.coursera.org/learn/ruby-on-rails-intro/lecture/kWeIk/software-installation-for-windows-users>. Accessed 10/19/2017 @ 6:51 A.M. E.S.T.
6. Installing Ruby Gem in Windows <https://stackoverflow.com/questions/18908708/installing-ruby-gem-in-windows>. Accessed 10/19/2017 @ 8:04 A.M. E.S.T.

# Appendices

# Appendix A: Test Install

Work Completed 10/27/2017

Test installation of [sciNote](https://github.com/USEPA/sciNote) using a Raspberry Pi 3 with 32GB SD on Fedora 26 platform for staging on T&E server (Fedora 24) and deployment on RTP VM912 server (Red Hat Enterprise Linux 7.4 – *Fedora 19/20 Maipo*).

## Hardware

[Raspberry Pi 3](https://www.raspberrypi.org/products/raspberry-pi-3-model-b/) with 32GB SD.

## Software Installation

### Install Fedora 26 Mate

1. [Download Fedora 26 ARM image.](https://fedoraproject.org/wiki/Architectures/ARM/Raspberry_Pi)
2. Using [Etcher](https://etcher.io/) install Fedora 26 to 32GB SD Card.

#### Booting Fedora on the Raspberry Pi for the first time

1. Insert the SD card into the Raspberry Pi.
2. Make sure you have a keyboard, mouse, network cable, and monitor connected.
3. Power on the Raspberry Pi.
4. You will see Fedora booting and eventually the "Initial setup wizard" will appear.
5. Follow the wizard to set language, time zone, and create users.
6. You should be presented with a login prompt or a getting started guide (depending on which Desktop/SPIN you're using).
7. Set server to: localhost.
   1. Initial install of [Fedora 26 KDE](http://ftp.ussg.iu.edu/linux/fedora/linux/releases/26/Spins/armhfp/images/Fedora-KDE-armhfp-26-1.5-sda.raw.xz) loaded and installed, however, after install login screen listed ‘Select your user and enter password,’ and screen stayed locked – could go no further.
   2. Re-formatted 32GB SD and installed [Fedora 26 Mate](http://ftp.ussg.iu.edu/linux/fedora/linux/releases/26/Spins/armhfp/images/Fedora-Mate-armhfp-26-1.5-sda.raw.xz). Mate version worked without issues.
8. Resize the Root Filesystem:
   1. STEP 1: enlarge the 4th partition (this example uses mmcblk0). [ growpart /dev/mmcblk0 4 ]
   2. STEP 2: grow the fileystem to fill the available space (all images except Server). [ resize2fs /dev/mmcblk0p4 ]

#### DNF system upgrade

1. DNF system upgrade can upgrade your system to a newer release of Fedora, using a mechanism similar to that used for offline package updates. The updated packages are downloaded while the system is running normally, then the system reboots to a special environment (implemented as a systemd target) to install them. Once installation of the updated packages is complete, the system reboots again to the new Fedora release.
2. Back up your important data. Every system change is potentially risky, be prepared. In case you update your workstation, it is wise to download a Workstation Live image and make sure your hardware (graphics card, wifi, etc) works well with the latest kernel and drivers.
3. Consult contract support prior to installation on T&E or VM912 as may create unintended consequences, i.e., automatic updates prior to testing systems, reboot creating issues for GEMM/GREENSCOPE, etc.

Work Completed 10/28/2017

### Install [sciNote](https://github.com/USEPA/sciNote)

#### Install required software

1. Ruby on Rails 4.2.3 Fedora & RHEL: <https://rubygems.org/gems/rails/versions/4.2.3>
2. Check path…
3. Enable SSH [ config ]

Work Completed 10/29/2017

1. Docker <https://www.docker.com/docker-fedora>
   1. Download [ nmap ]
      1. apt-get install nmap # Debian distros
      2. yum install nmap # RHEL/Fedora distros
   2. Identify IP address workstation/VM.
      1. hostname -I # Example may show range of available IP addresses for use, e.g., 192.168.200.114 192.168.200.118
2. JS framework. Bootstrap 3 <http://getbootstrap.com/> and jQuery <http://jquery.com/download/>.

Work Completed 10/30/2017

1. PostgreSQL 9.4 <https://www.postgresql.org/download/>.
2. pgAdmin 4 <https://www.pgadmin.org/download/>.

#### Install sciNote Local Server

1. Login to EPA GitHub <https://github.com/USEPA/sciNote> (note: contact [young.daniel@epa.gov](mailto:young.daniel@epa.gov) for access).
2. Clone this Git repository onto development machine.
3. Create a file config/application.yml. Populate it with mandatory environmental variables (see [environmental variables](https://github.com/biosistemika/scinote-web/wiki/setup-guide" \l "user-content-environmental-variables)).
4. In sciNote folder, run the following command: make docker. This can take a while, since Docker must first pull an image from the Internet, and then install all necessary Gems required by sciNote.
5. Once the Docker image is created, run make cli command. Once inside the running Docker container, run the following commands: rake db:create - rake db:migrate - rake db:seed. This should initialize the database and fill it with (very minimal) seed data.
6. Exit the Docker container by typing exit.
7. To start the server, run command make run. Wait until the server starts listening on port 3000.
8. Inside another CLI window (from sciNote folder), run command make worker. This command should start the background worker process which is used by sciNote for a lot of demanding tasks.
9. Open your favourite browser and navigate to [http://localhost:3000](http://localhost:3000/). Use the seeded administrator account from [seeds.rb](https://github.com/biosistemika/scinote-web/blob/master/db/seeds.rb) to login, or sign up for a new account.

#### Docker Commands

Call make commands to build Docker images and build Rails environment, including database.

Following commands are available:

| **Command** | **Description** |
| --- | --- |
| make docker | Downloads the Docker image and build Gems. This should be called whenever Gemfile is changed. |
| make db-cli | Runs a /bin/bash inside the db container. |
| make run | Runs the db container & starts the Rails server in web container. |
| make start | Runs the db container & starts the Rails server in web container in background. |
| make stop | Stops the db & web containers. |
| make worker | Runs the rake jobs:work (worker process) in web container. |
| make cli | Runs a /bin/bash inside the web container. |
| make tests | Execute all Rails tests. |
| make console | Enters the Rails console in web container. |
| make export | Zips the head of this Git repository into a .tar.gz file. |

### Install VS Code

1. YUM instructions:
   1. [ . <( wget -O - https://code.headmelted.com/installers/yum.sh ) ]
2. Further details can be found @ <https://code.headmelted.com/>.

## Linux Distros Tested

Fedora 26 Arm

Ran into several issues with Fedora 26 Arm Images using all four versions below. Issues most likely due to hardware (Raspberry Pi) and should not be an issue for EPA servers.

1. Fedora 26 Workstation
2. Fedora 26 KDE
3. Fedora 26 Mate
4. Fedora 26 Minimal

### Raspbian Jesse

Previous minimal test installation was successful using standard Debian OS for initial Proposal submitted to ECEB Branch Chief 10/26/2017. Test Install will be completed in Debian, however, all notes will not be logged in this appendix as EPA servers are Fedora/RHEL and are documented below as best as possible for testing completed on Fedora Mate & finished on Debian Jessie.

### HypriotOS

Tested sciNote on [HypriotOS](https://blog.hypriot.com/) with xfce4. OS designed as extremely lean ‘Docker’ packaging, distribution, installation and execution of (complex) applications. Although minimal and fast, too many additional steps and packages are needed. In addition, would require requesting separate VM to run sciNote as it is not compatible with RHEL/Fedora.

# Appendix B: sciNote Terms of Service

## Terms of Service Open Source Version – preferred

sciNote which is available at our open source repository on the following URL: <https://github.com/biosistemika/scinote-web>.

## Terms of Service Hosted Instance Version

Access to the terms of service for sciNote electronic laboratory notebook Hosted Instance can be found @ <https://scinote.net/docs/terms-of-service.pdf>

# Appendix C: Fedora/RHEL Commands

## Basic Shell Commands

1. Manual that can be accessed within the shell. man <program name> //DENEME
2. Information resource while in a shell is the info command. info <program name>
3. List the current directories file and directory. ls -s
4. List the another directory then the current. ls <path>
5. Change to a different Directory. cd <path>
6. Remove a file in the current directory. rm <file name>
7. Remove a file in a different directory. rm <path and file name>
8. Remove a directory in current directory. rmdir <directory name>
9. Remove a directory in a different directory. rmdir <path and directory name>
10. Show the contents of a file. cat <file name> or cat <path file name>
11. Show current directory. pwd
12. Run a command as root. sudo <command> or su

su will provide root account access. After running this command, you will be prompted for root password. After providing the correct password you will then be logged in as root for that terminal session.

1. Install a package at the command line (root access needed by su or sudo). yum install <package>
2. Log out (or close if open from a Desktop Manager). exit

# Appendix D: sciNote

[](https://camo.githubusercontent.com/457865f9d1e95b0535f93a1f266daa7146c4ab93/687474703a2f2f7363696e6f74652e6e65742f77702d636f6e74656e742f75706c6f6164732f323031352f31302f6c6f676f5f7363694e6f74655f66696e616c2e706e67)

## About

[sciNote is an open source electronic lab notebook](https://scinote.net) ([ELN](https://en.wikipedia.org/wiki/Electronic_lab_notebook)) that helps user manage laboratory work and stores all experimental data in one place. sciNote is specifically designed for life science students, researchers, lab technicians and group leaders.

## Build & run

See [Wiki/Build & run](https://github.com/biosistemika/scinote-web/wiki/setup-guide" \l "build-&-run).

## Testing

See [Wiki/Testing](https://github.com/biosistemika/scinote-web/wiki/setup-guide" \l "testing).

## Contributing

See [Wiki/Contributing & Collaboration](https://github.com/biosistemika/scinote-web/wiki/contributing-&-collaboration).

## License

sciNote is developed and maintained by BioSistemika USA, LLC, under [Mozilla Public License Version 2.0](https://github.com/biosistemika/scinote-web/blob/master/LICENSE.txt).

See [LICENSE-3RD-PARTY.txt](https://github.com/biosistemika/scinote-web/blob/master/LICENSE-3RD-PARTY.txt) for licenses of included third-party libraries.

Appendix E: Git Commands

Using Terminal

|  |  |  |
| --- | --- | --- |
| **Git task** | **Notes** | **Git commands** |
| **[Change directory](http://atlassian.com/git/tutorial/git-basics%23!init)** | Change to directory you want to put source code from GitHub. Run command **git clone /path/to/repository** | **cd** /Volumes/ICY\Dock\1 |
| Once software downloads change directory to SW folder (SCINOTE) or other SW | **cd** /Volumes/ICY\Dock\1/SCINOTE |
| **[Check out a repository](http://atlassian.com/git/tutorial/git-basics%23!clone)** | Create a working copy of a local repository: | **git clone /path/to/repository** |
| **[Add files](http://atlassian.com/git/tutorial/git-basics%23!add)** | Add one or more files to staging (index): | **git add \*** |
| **[Commit](http://atlassian.com/git/tutorial/git-basics%23!commit)** | Commit changes to head (but not yet to the remote repository): | **git commit -m "Commit message"** |
| **[Push](http://atlassian.com/git/tutorial/remote-repositories%23!push)** | Send changes to the master branch of your remote repository: | **git push origin master** |
| **[Status](http://atlassian.com/git/tutorial/git-basics%23!status)** | List the files you've changed and those you still need to add or commit: | **git status** |
| **Tags** | You can use tagging to mark a significant changeset, such as a release: | git tag 1.0.0 <commitID> |
| CommitId is the leading characters of the changeset ID, up to 10, but must be unique. Get the ID using: | git log |
| Push all tags to remote repository: | git push --tags origin |
| **[Undo local changes](http://atlassian.com/git/tutorial/undoing-changes)** | If you mess up, you can replace the changes in your working tree with the last content in head:  Changes already added to the index, as well as new files, will be kept. | git checkout -- <filename> |
| Instead, to drop all your local changes and commits, fetch the latest history from the server and point your local master branch at it, do this: | git fetch origin  git reset --hard origin/master |
| **Search** | Search the working directory for foo(): | git grep "foo()" |

# Appendix F: Software Development & Use

## Free and open-source software (FOSS)

All software used for this project can be classified as both free software and open-source software. That is, anyone is freely licensed to use, copy, study, and change the software in any way, and the source code is openly shared so that people are encouraged to voluntarily improve the design of the software. This is in contrast to proprietary software, where the software is under restrictive copyright and the source code is usually hidden from the users. The benefits of using FOSS can include decreased software costs, increased security and stability (especially in regard to malware), protecting privacy, education, and giving users more control over their own hardware. Free, open-source operating systems such as Linux and descendants of BSD are widely utilized today, powering millions of servers, desktops, smartphones (e.g. Android), and other devices. Free software licenses and open-source licenses are used by many software packages. The Free software movement and the open-source software movement are online social movements behind widespread production and adoption of FOSS. Section 5 of this document outlines the specific FOSS software used in developing sciNote. Below are additional FOSS software packages useful for additional software development, such as training support, document generation, etc.

LibreOffice is a powerful office suite – its clean interface and feature-rich tools help you unleash your creativity and enhance your productivity. LibreOffice includes several applications that make it the most powerful Free and Open Source office suite on the market: Writer (word processing), Calc (spreadsheets), Impress (presentations), Draw (vector graphics and flowcharts), Base (databases), and Math (formula editing). <https://www.libreoffice.org/discover/libreoffice/>.

nomacs is a free, open source image viewer, which supports multiple platforms. You can use it for viewing all common image formats including RAW and psd images. nomacs features semi-transparent widgets that display additional information such as thumbnails, metadata or histogram. It is able to browse images in zip or MS Office files which can be extracted to a directory. Metadata stored with the image can be displayed and you can add notes to images. A thumbnail preview of the current folder is included as well as a file explorer panel which allows switching between folders. Within a directory you can apply a file filter, so that only images are displayed whose filenames have a certain string or match a regular expression. Activating the cache allows for instantly switching between images. <https://nomacs.org/>.

Krita is a professional FREE and open source painting program. It is made by artists that want to see affordable art tools for everyone. concept art; texture and matte painters; illustrations and comics. <https://krita.org/en/>.

Kdenlive is an acronym for KDE Non-Linear Video Editor. It is primarily aimed at the GNU/Linux platform but works on BSD and MacOS. It is currently being ported to Windows as a GSOC project. Non-linear video editing is much more powerful than beginners’ (linear) editors, hence it requires a bit more organization before starting. However, it is not reserved to specialists and can be used for small personal projects. <https://kdenlive.org/about/>.

Mixxx is free [1] DJ software for Windows, Mac OS X and Linux. It’s easy to use so you can start mixing your favorite songs in no time, even if you’ve never DJed before. Mixxx offers a variety of features for experienced DJs and beginners alike, such as automatic beat and key detection, master sync, hotcues, looping, 4 decks, effects and many more. You can perform with Mixxx using MIDI and HID controllers, CD players, vinyl turntables or even just your computer’s keyboard and mouse. <https://www.mixxx.org/manual/latest/>.

VirtualBox is a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2. See "About VirtualBox" for an introduction. Presently, VirtualBox runs on Windows, Linux, Macintosh, and Solaris hosts and supports a large number of guest operating systems including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7, Windows 8, Windows 10), DOS/Windows 3.x, Linux (2.4, 2.6, 3.x and 4.x), Solaris and OpenSolaris, OS/2, and OpenBSD. <https://www.virtualbox.org/>.