Flying Solo: Implementing a Small Technology Project with Minimal Resources

In the summer of 2017 my supervisor asked me to implement an open source software package called Suma. A faculty librarian saw a presentation on it at a conference and thought it might be useful as we grapple with assessing how students use our physical spaces within the library. But the project's web page only told me so much. In order to better understand what we could do with Suma and how it might be applicable to our goals, I needed to get it up and running.

Step 1: Assessing Project Software Requirements

Getting an up-front idea of what skills and resources you'll need to successfully implement is critical. But it's OK if you need to learn some skills later. You don't have to have skill mastery to begin making small changes. If you feel you have to completely redesign the codebase, that project is probably not for your library.

Some preliminary questions to ask:

- Do the project's files need to be installed on a server or will they run locally?
- What programming or scripting libraries are required to be installed alongside the software?
  - Which of these does the installation process already include?
  - Which of these are you expected to install and configure on your own?
- Are there any external desktop applications, web applications, or web services the software interacts with or relies upon?
  - Does the software communicate with external components remotely or locally?
- How is the software administered or customized?
  - Is there a web interface for administration or customization of any components?
  - Are there configuration text files that may need to be edited?
  - Do software changes require altering the software code, scripts, or web files?
- Is there a client interface?
  - Does the client-side of the project run as its own web page or require integration with some other program or web interface?

Step 2. Where do I put my project's software?

If the project is something installed locally to the PC, the answer is pretty easy, but many software projects require server-hosted elements.

Local virtual machine

Great for testing installation and configuration, but very limited for network access (depending on your campus network policy). Not suitable for production.

In-house servers through library or campus IT

This is a great solution, if it's an option. Does your library have an in-house IT team? What kind of support will they provide? Does your IT department offer hosting? Will they support your OS and other baseline services? Do any IT restrictions make this infeasible?

3rd party hosting

Carries costs the other two options do not, but typically grants more than adequate access for necessary installations and configurations. Usually includes some OS-level and general service -level support. Academic-focused hosting is often cheaper than commercial hosting and may have better support for small projects.

Step 3. The Project is a go. Help!

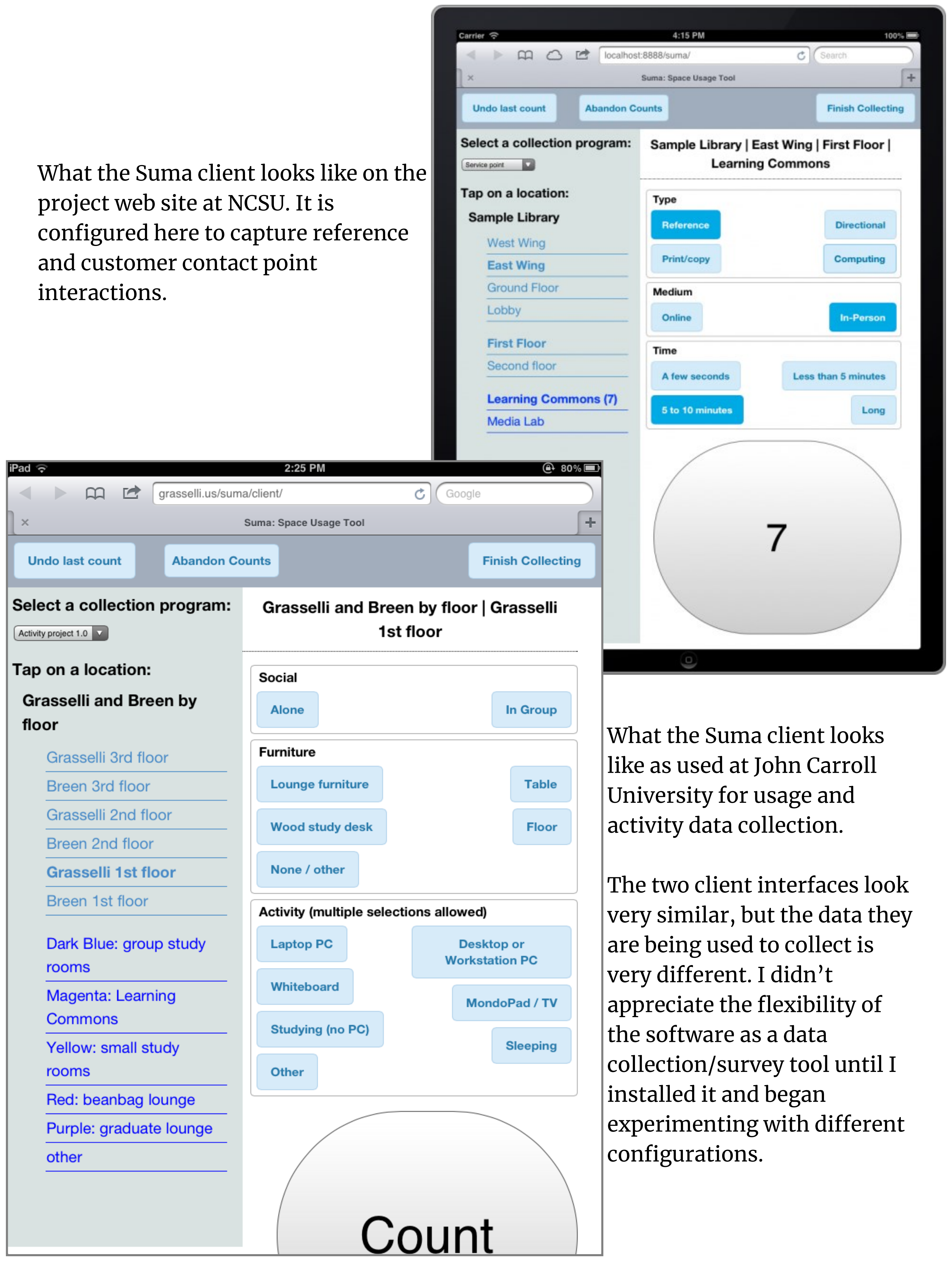
You've begun and now you're committed. Where do you look for help?

- **You hosting provider or IT department**

  The people who provide hosting know how stuff works. Even if parts of your project fall outside of their responsibility or expertise, they can help troubleshoot the underlying standardized services.
- **The project authors or maintainers**

  Most project maintainers are willing to help. The authors of academic projects tend to be very helpful. As long as you are willing to do the legwork to gather requested info and perform key changes, you can often get the answers you need.
- **User communities**

  If your project or hosting provider has community forums or a mailing list, sign on and ask away. Someone has probably encountered your same issue before.

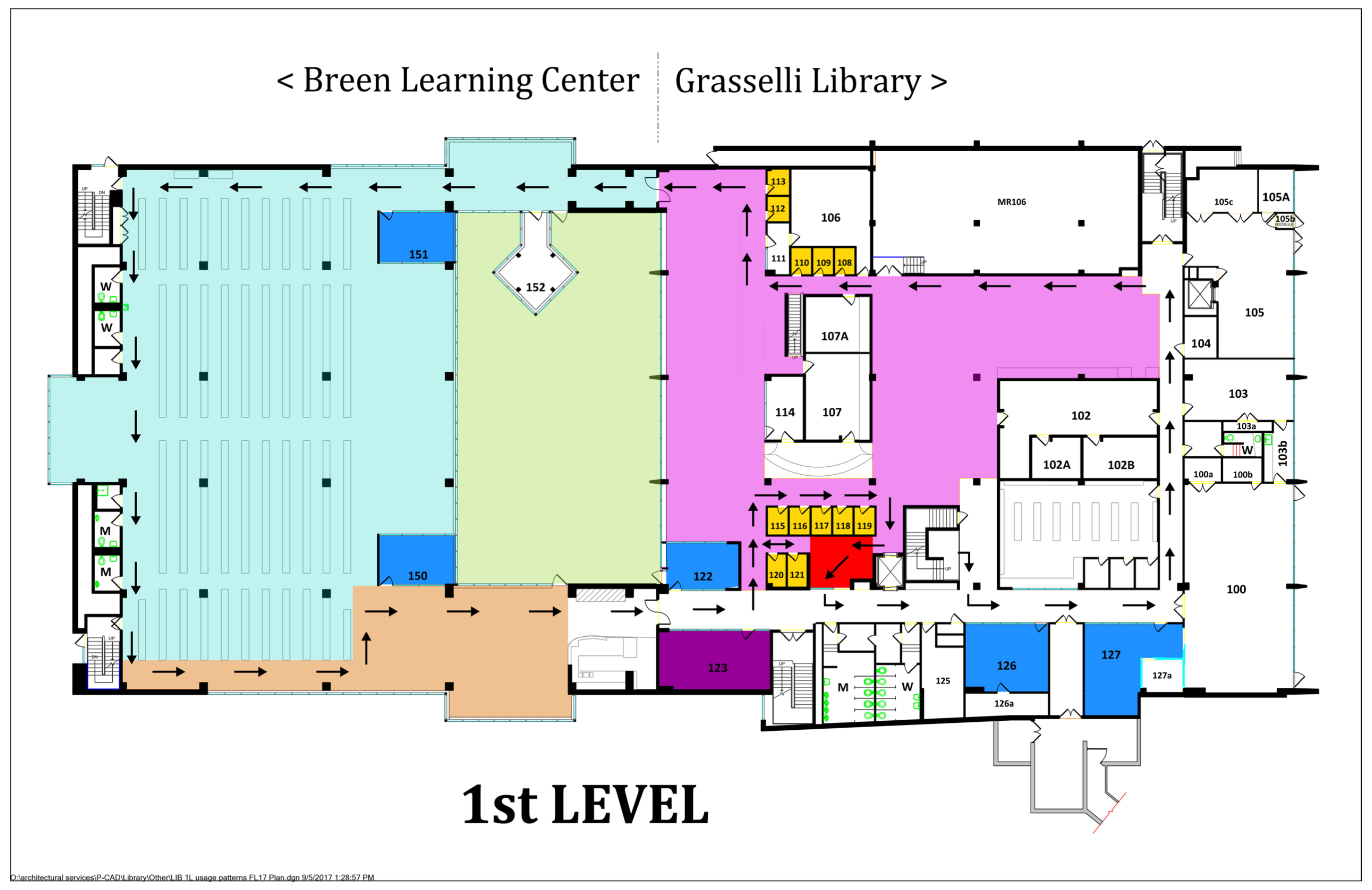


What the Suma client looks like on the project web site at NCSU. It is configured here to capture reference and customer contact point interactions.

What the Suma client looks like as used at John Carroll University for usage and activity data collection.

The two client interfaces look very similar, but the data they are being used to collect is very different. I didn't appreciate the flexibility of the software as a data collection/survey tool until I installed it and began experimenting with different configurations.

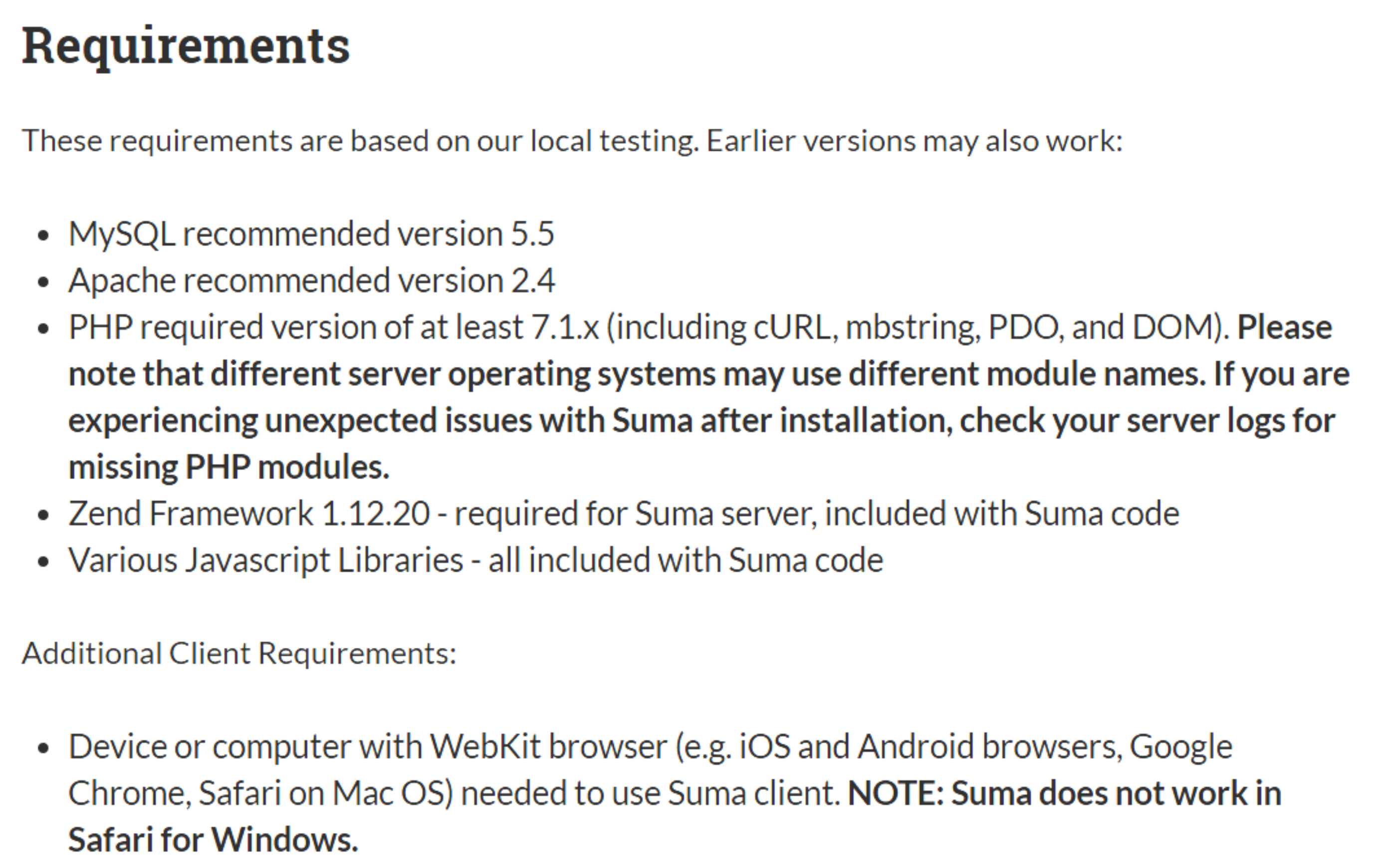
Maps like this helped us categorize types of spaces and informed our configuration of Suma. They also help ensure students collect data in a way that is consistent.



Brandon Walker
John Carroll University
bwalker@jcu.edu 216-397-1988

Grasselli Library and Breen Learning Center

Suma's listed requirements



Requirements

These requirements are based on our local testing. Earlier versions may also work:

- MySQL recommended version 5.5
- Apache recommended version 2.4
- PHP required version of at least 7.1.x (including cURL, mbstring, PDO, and DOM). **Please note that different server operating systems may use different module names. If you are experiencing unexpected issues with Suma after installation, check your server logs for missing PHP modules.**
- Zend Framework 1.12.20 - required for Suma server, included with Suma code
- Various Javascript Libraries - all included with Suma code

Additional Client Requirements:

- Device or computer with WebKit browser (e.g. iOS and Android browsers, Google Chrome, Safari on Mac OS) needed to use Suma client. **NOTE: Suma does not work in Safari for Windows.**

Skills and resources needed

- Linux/Unix command line and remote access.
- Apache basics and file permissions.
- Basic knowledge of Git and Github.
- PHP and Javascript if I wanted to deviate from default behavior or appearance.
- A server or hosted environment with Git, MySQL, Apache, and PHP.
- Web-capable devices on which to run the client interface.

So where did we put Suma?

Virtual machine on my work PC? X

Could be useful for testing the back-end, but I would need to recreate my work in a proper hosted environment if we committed to Suma. Further, campus network and PC limitations would prevent proper web-interface testing.

ITS-hosted server space? X

ITS was not interested in providing space for this kind of exploration at the time.

3rd party hosting? √

We looked at some academic-focused hosting services and several were very affordable. We chose to go with Reclaim Hosting because they had good referrals from other library professionals and have a peer support community to supplement their official support.

What am I missing? The clients! Part of our interest in Suma was the hope that the client interface would run on a couple aging 1st generation iPads for which we had no other practical use. This was yet another aspect of Suma we needed to test. (Spoilers: they work great!)

Thank goodness for helpful people

I got stuck several times during Suma install and configuration. Here's some of what I encountered.

- I posted to the Suma mailing list with a question about whether our hosting provider's newer versions of some software components were likely to be an issue.
- I encountered an error trying to access the client interface. Asking about this on the Suma list lead to an inquiry with our hosting provider, resulting in a necessary permissions change.
- A question to the Suma list about configuration changes not saving lead to identifying and replacing a damaged settings file.

The project's mailing list proved invaluable, but our hosting provider was also very forthcoming with information and willing to work with us to get to the bottom of problems as they occurred.