

## Case Story: Simple.org

(Based on real events.)

Simple.org is a story about scale. On the one hand, the tagline “the fast, free app for clinicians to manage their patients with high blood pressure” describes a small mobile app with a singular purpose. On the other hand, Simple’s mission is to save the millions of lives currently lost every year to heart attack and stroke. That’s a mission which implies huge networks and thousands of moving parts. Can one little app save millions of lives? Certainly not on its own.

### The Problem

When *Resolve to Save Lives* approached nilenso in May of 2018, the problem they had in hand was seemingly straight-forward: Build an MVP (Minimum Viable Product) of the Simple app and start showing it to users. But even as we claim “build an MVP” to be the problem statement we start to see the different scales at which this project operates. Yes, it’s true that the initial task could be summed up as “build an MVP” but from *Resolve’s* perspective, the initial-initial task was actually “find a consulting company in India to build an MVP.” Halfway through 2018, there was much about the project which was uncertain, unknown, or undecided. Although “Simple” was a front-runner, the product didn’t even have a name yet.

Unravelling the true scale of a project is often best accomplished by asking “why?” Every paid consultant has gone through the “Five Whys” exercise. The exercise is easy: ask “why?” of the problem at hand, then ask “why?” of the answer you uncover. Repeat five times. This exercise often makes inexperienced consultants come across like curious, but incredibly annoying, three-year-olds. When consulting with an organization like *Resolve*, the exercise wasn’t actually necessary. Or perhaps it would be more accurate to say was not obviously necessary.

The first meeting at the nilenso office was with Daniel Burka, Director of Product and Design at *Resolve*. The most common appraisal Daniel had among nilenso staff who hadn’t met him yet was “the guy who designed the Firefox logo.” For those who had started careers early enough to witness the early days of the Firefox web browser, this was no specious laurel. Firefox represented far more than the buzzwords we associate with it today: Open Source, Free Software, a World Wide Web based on Open Standards. Back in 2003, Firefox represented the democratization of the web in a David v. Goliath fight against Microsoft’s stagnant and under-performing Internet Explorer, universally hated by software developers the world over. Firefox sat in the intersection of two communities, two ideas. On the left was *Free Software*. On the right was *Design* with a Capital D. Good Design meant that a product like Firefox had to be approached with

Design Thinking. It had to have a brand. It had to be fun. It had to be so much better than everything else you would be a fool not to use it. Firefox was all these things. It was also a Javascript debugger, a CSS reverse-engineering tool, and the doorway millions of developers stepped through to create the web and the internet we all rely on today. The significance of such a tool, and such a brand, is not to be understated. So, come. Meet Daniel. The guy who designed the Firefox logo.

No one from nilenso remembers having a conversation about the Firefox logo but everyone who attended those early meetings remembers that the “why” of the project was very much understood. As far as *Resolve*’s goals were concerned, the “Five Whys” exercise would hardly be necessary. Why was *Resolve* talking to nilenso? **We heard your team was very good and we want to validate that by building an MVP.** Why an MVP? **We need to build real software, get it in front of our first users in Punjab, and start collecting feedback.** Why build this software? **Millions of preventable deaths happen worldwide every year due to heart attack and stroke. Many of those deaths aren’t only preventable but they are *easily* preventable. Some countries have this right, some don’t. We want to build the open source tool that will help everyone on the planet prevent these deaths.** Why hypertension? What’s so special about heart attack and stroke? **There are hundreds of possible candidates for attention at this scale, but hypertension is at the top of a lot of lists: It causes a massive number of deaths. It is easy to prevent with simple protocols and affordable medicines. It is well understood — we aren’t waiting for a miracle cure or scientific breakthrough; we just need to help nurses and doctors support their patients.**

The reasoning was sound. The project made sense. *Resolve* and nilenso had barely begun discussions and the team was already excited to get to work. But a project as exciting as Simple permits further reflection. It was an opportunity for nilenso to turn the “Five Whys” on itself.

---

**“All we have to decide is what to do with the time that is given us.”**

Deepa Venkatraman and Steven Deobald, who paired <sup>1</sup> on project management for Simple, found out early in their working relationship that they shared a common friend in Avishek Sengupta — affectionately known as “Mojo” to them both. Mojo and Deepa built a computer vision path-tracking robot together for a college project. And back in 2006, Steven and Mojo were pairing on some legacy Java code in a tiny office in downtown Vancouver.

Mojo turned to Steven:

---

<sup>1</sup><https://martinfowler.com/articles/on-pair-programming.html#KnowledgeSharing>

“All we have to decide is what to do with the time that is given us.”

Steven answered with a blank stare.

“Tolkien.” Mojo offered. “Fellowship of the Ring? I just re-read the books. That’s sort of. . . the highest problem statement philosophically possible, isn’t it?” And with that, he turned back to his Java debugging.

It was common to bump up against such peculiar insights when pair programming with Mojo. The Java code in front of him was rarely the most intriguing thing on Mojo’s mind during any given workday. When they learned of this triangle of friends and colleagues, Deepa and Steven hardly thought Mojo’s peculiar insights that day would define the foundations of the software cooperative where they worked together.

Over a decade later, nilenso was bootstrapped, profitable, and settling in for an identity crisis. An employee-owned worker cooperative has the roles and responsibilities one expects from any corporation: profit centres, administration, executive functions, purchasing. Those roles are not necessarily mapped one-role-for-one-person as they are in large, traditional organizations.<sup>2</sup> But there is one role everyone shares to some degree: founder. Whether someone has been a part of nilenso for ten years or ten days, that someone is beginning something *today*. And that thing, whatever it is, is the foundation for *tomorrow*. To varying degrees, that means every member of nilenso gets to participate in answering the company’s existential questions, for better or for worse. And so, nilenso can turn the “Five Whys” on itself:

Why does nilenso exist? **To make money so everyone can pay down their mortgages, obviously. But also to make software.** Why make software? **Because we are good at it and because it’s fun.** Why is it fun? **Because it is satisfying to solve problems.** Why is it satisfying to solve problems? **Because, if we care about it, there is meaning in the problem — and meaning in its solution.** Why is there meaning in a problem? Why do you care about it?

Ay, there’s the rub.

To ask such questions is to ask the fundamental existential questions about our humanity. Why bother getting up in the morning? Why put on pants? Why am I writing this *particular* line of Java today? What is the meaning of life? And so forth.

At an organizational level, nilenso could direct this line of questioning toward the choices available to any corporation: *What exactly is it that makes a task, project, role, or job worthwhile?* Acknowledging that running a company involves other constraints — most importantly, the financial — and not simply an abstract question of cosmic purpose, how do we reconcile the gap between the work we

---

<sup>2</sup><https://www.reinventingorganizations.com>

find meaningful with the work that generates revenue? No one's mortgage is paid down in the currency of goodwill.

In the early days with Simple, finding meaning in the problem was easy because it was nilenso's first open source project to fully reconcile this gap. Simple is useful and fun, beautiful and free, well-funded and stable. Simple changes lives for the better. Like Firefox.

But this question of purpose doesn't simply vanish when a potential solution is offered. As we will see, the question itself grows and changes as new answers emerge. As Simple grew, its meaning and purpose grew with it.

"All we have to decide is what to do with the time that is given us," said Gandalf. "There are other forces at work in this world..."

## The Solution

### Kickoff

May tipped over into June and a completely melted Bangalore spring was poured onto summer. A team was assembled. *Obvious*, another Bangalore consultancy, was responsible for Simple's mobile app. The central service was built by nilenso, where Srihari Sriraman and Govind Krishna Joshi staffed themselves on the project as developers and Deepa and Steven similarly self-staffed as project managers. Project management was across teams, working between the team from nilenso, the team from *Resolve*, and the team from *Obvious*.

As Bangalore melted into a puddle marking the very centre of South India, a web of teams and roles took shape in the centre of that puddle. Daily stand-up meetings marked the end of every day (India time) with the teams from nilenso and Obvious sharing their progress with Daniel and *Resolve's* Director of Engineering, Tim Cheadle. The team was small, distributed, and fast. Communication between designers, mobile app developers, and service developers was immediately productive and the two teams in Bangalore shared an office space as regularly as they could. From one day to the next, entire designs would be considered, attempted, tested, thrown out, redone. Discussions were open and earnest about exactly what the collective team was trying to build. Even though the MVP was built to validate ideas, it was not an experiment — with clear goals in place already the product had unmistakable rights and wrongs in implementation. With just a handful of weeks available, the team desperately wanted to build The Right Thing.<sup>3</sup>

The Right Thing, of course, is a moving target. What is right in June 2018 certainly won't be right in April of 2019 and the exercise was one of brainstorming, identifying risks, narrowing scope, and executing... all as quickly as possible.

---

<sup>3</sup>[https://en.wikipedia.org/wiki/Hackers:\\_Heroes\\_of\\_the\\_Computer\\_Revolution](https://en.wikipedia.org/wiki/Hackers:_Heroes_of_the_Computer_Revolution)

For Simple, the twin tasks of brainstorming and identifying risks melded into one long, iterative activity of imagination and research.

Some risks to the project’s success were self-evident. The initial deployments were to go to nurses in remote parts of rural Punjab, with the intention of distributing the app to every state in India — and even more remote locations if it proved a success. The team needed imagery to create scenarios for users. How remote is “remote”, really? India’s vast landscape is dotted with cellular towers and dozens of new towers go up every day. What image could possibly exemplify the extremes the Simple software would be subjected to?

Images of the Spiti Valley resemble the sort of ultra-high-resolution, floor-to-ceiling photographs one might find in the foyer of Goldman Sachs. Spiti Valley is extreme. High in the Himalayas, Spiti can claim not only the title of mountain valley but also the title of *desert*. It is breathtakingly beautiful and definitely remote — but it is not uninhabited. Those same foyer photos almost always include monasteries, villages, bridges, and flags. Whenever a design choice for Simple required thinking about a remote location where nurses might lack internet access for days, the team would recall the Spiti Valley vignette: a desert valley 13,000 feet above sea level with an unmistakable human presence.

With such imagery at hand, the team unanimously agreed the software must be designed “offline first.”<sup>4</sup> Other decisions were less obvious. The first MVP deployment would happen across five clinics in rural Punjab. But would the Punjabi nurses in these clinics insist on a user interface entirely in Punjabi? Would Hindi suffice? Would some nurses be more accustomed to an English keyboard, even if their phones were set to a Punjabi locale? The only way to get definitive answers to such questions was to fly to Punjab and ask the nurses yourself, so the team built the first MVP with both Punjabi and English support.

### **City of Lakes: Bathinda, Punjab**

Most software projects do not contain a scene like that of the competitive black-hat intern interview from the movie *The Social Network*.<sup>5</sup> True, it’s doubtful that scene actually occurred even in the early days of Facebook. Most projects don’t contain a single scene that would fit well in a typical Hollywood techno-guitar-backed montage of the project’s heroes hacking into the night fuelled by Red Bull and coffee.

And neither did Simple.

However, after three heavy months of back-and-forth design and code between the members of the small teams, there was one last bout of late-night hacking. It was May of 2018. The nurses and their managers (Cardiovascular Health

---

<sup>4</sup><https://medium.com/simple-dot-org/offline-first-apps-are-appropriate-for-many-clinical-environments-cddf5a73bb61>

<sup>5</sup><https://www.youtube.com/watch?v=iFW7Lo6mk5I>

Officers or CVHOs) would see the app for the first time tomorrow and the team was clustered in the lobby of their Bathinda hotel, squashing the last bugs and putting the last bit of polish on the alpha version of the app. Instead of a high-intensity guitar riff to back up their hacking, however, our heroes had to settle for the hotel's lobby track: *The Kungs'* "This Girl".<sup>6</sup> On endless repeat. Go ahead and take a break to find the song on YouTube. We'll wait here for you.

"Can you *please* change the music to something else? Or turn it off?" begged Govind of the hotel staff. "Sorry, Sir. It is hotel policy." The grey in the eyes of the hotel's receptionist indicated that his will to bend that policy had already been crushed by countless hours of French elevator music.

And so it was. Our heroes hacked into the night, their excitement to deliver a polished MVP to the nurses of Bathinda acting as armour to defend them from the stinging species of earworm crawling all over them by 3:00 AM. "You'll never buy my loooove...!"

---

A few hours of sleep later, the entire Simple team downed their last cups of chai and Nescafé, piled into cabs, and headed to the first health clinic. Daniel and the rest of the design team had visited Bathinda once before but for the development teams this would be their first face-to-face interaction with real users. None of them knew exactly what to expect.

As the team walked into the first clinic, everyone was struck by how busy it was. Everyone's attention was instantly drawn to the eye of the storm: the nurse. This was to become the first of many such experiences with the software team repeatedly awestruck by the nurses' abilities and patience.

Within minutes, the team had validated their first major design choice: offline first. An online first mobile app can slow down or stall while it waits for the server to reply from across the chasm of the internet. The Simple app would have no such luxuries. Deepa would later describe the nurses as "super hard-working" and "superhuman." They were in constant motion.

"There's a long line of patients waiting outside their cabin," Deepa said, painting the scene. "They're measuring a patient's weight, BP, speaking to them, asking them about their day, their symptoms, whether or not they've been taking their meds, and recording all of this info in a register. Some of that information needs to go into *yet another* register. In the meantime another patient who had been to see the doctor returns: the nurse quickly updates their info. Then they're called away to attend to something for 10 minutes. It's non-stop. And through all of this they're polite, yet firm, and are making no mistakes. It's pretty amazing."

No, the nurses could not wait for an app. The Simple app's nearest competitor was a piece of paper: If the nurse could write down a patient's results faster

---

<sup>6</sup><https://www.youtube.com/watch?v=2Y6Nne8RvaA>

than they could enter it into Simple, the app had failed them. Not only could the app never wait for the network, it could absolutely never crash, hang, or pause — not even for a second.

Govind sat down to speak to a nurse during a break set aside to test the app. He was a bit surprised to see her pull a new iPhone out of her pocket and set it on the desk. Android has 95% of India’s mobile marketshare. iOS has less than 4%. Statistically speaking, building Simple for Android felt like more of an inevitability than a design choice.

She took a sample Android phone the team brought with them and began to examine Govind as she would treat any other patient. She wrapped the cuff of the blood pressure monitor around his arm and inflated it. As she opened the app to enter his BP reading, she exclaimed, “Oh! This is in Punjabi. That’s not what I was expecting. Can you change the interface to English for me?”

TODO: Hindi? (Govind)

Govind switched the phone’s locale to English and handed it back to her so she could finish trying the app while the designers watched and took notes. As she put the blood pressure monitor away, she turned to them. “You know, it is very nice of you all to build this app. . . I’m just not sure I trust it yet. I will use it because it is mandated but I will keep entering my patients’ data into the paper register as well. These records are very important. It is hard for me to entrust my patient’s lives to software that didn’t even exist six months ago.”

TODO: Hindi? (Govind)

The lessons in the other facilities were similar. The nurses were polite but cautious of Simple’s promises. They were unflagging as a constant stream of patients poured through the clinic. And they remained full of surprises.

“An robh thu ann an Alba? Tha e brèagha.” (“Have you been to Scotland? It’s beautiful.”) one nurse asked with a calm smile as Srihari and Govind quizzed her about her time in Scotland, where she took her nursing degree.

The calm demeanour of the nurses was put to the test around noon of the last day of the field visit. Srihari looked up from his laptop as he heard shouting from one end of the hospital. A man came tearing through the hospital hallway with two policemen in close pursuit, in one end of the hospital and out the other. A few patients looked up. The nurses barely blinked. The police had their job to attend to — and the nurses had theirs.

## Rollout

No software project is without its difficulties and it could even be said that building high-quality software is mostly about identifying and overcoming those difficulties. Much as nurses triage patients in their clinics, every software team needs to repeatedly zoom out to see the entire picture, reevaluate and reprioritize,

and then zoom back in to battle with the next task. Many of these challenges facing the Simple team were known up-front. Clinic protocols already had a new, paper-based system for capturing blood pressure, known as “Red Cards”. Uniquely identifying users was going to be a huge challenge — how many Gurpreet Singhs live in a single district in Punjab? And perhaps most importantly, once the nurses had a straight-forward system for entering all this data, would the patients even show up for them to record it?

Each difficulty was intercepted in turn. The question of what to do about Red Cards was obvious. The Red Cards themselves weren’t really a difficulty, per se. They actually captured the very data the Simple app was intended to capture, ahead of its release. From a purely data perspective this was wonderful. Simple or no Simple, data was being captured in a systematic way which conformed to international protocol. Once Simple was ready for release, it just needed a way to assimilate this data.

From a software architecture perspective, Red Cards were an ephemeral data store. It doesn’t really matter whether the storehouse of your data is an aging MySQL database or paper cards in a filing cabinet. If the goal is to switch gears and move to a new data store then the problem is fundamentally a data migration problem. The Simple team viewed the Red Cards through precisely this lens: They exist, they are useful, but they are temporary. Unlike the rest of Simple, the data migration process from Red Cards into the Simple database need not be beautiful, meticulously designed, or environmentally robust. It was decided that the best approach to Red Cards was to make everything about their cataloguing as uncomplicated as possible.

It is often tempting for software companies to follow the lead of car and watch manufacturers, insisting that they are somehow creating luxurious works of art... the very union of form and function. Such self-deception carries the bad taste of basic dishonesty. A car is a box which moves people from A to B. A watch is a device which displays any one of 1440 numbers, each one representative of a corresponding Earth Day Minute. Let us then indulge the good taste of clarity and describe the solution to the Red Cards as it was: boring, straightforward, simple. It was decided that, although the Red Cards themselves must remain in the clinics, the data entry for said cards need not be carried out in the clinics. (Why force yourself to do online data entry in Spiti Valley if you don’t have to?) The data entry clerk would take a photo of each Red Card using a standard mobile phone, head back to the city where electricity and internet access were not subject to the vagaries of countryside infrastructure, and use a regular computer to access the very basic web app built by nilenso to enter each card into the Simple database.

---

The solution to the problem of uniquely identifying patients was less obvious. “Do The Simplest Thing That Could Possibly Work”<sup>7</sup> doesn’t apply to a

---

<sup>7</sup><http://wiki.c2.com/?DoTheSimplestThingThatCouldPossiblyWork>



problem space if we look across the globe and see the governments of most nations struggling with it. Many solutions were considered, from fall-through aggregates of all possible data the system might employ to identify patients (name, phone number, clinic, driver’s license number, aadhaar number, etc.) to the possibility of assigning unique numbers to patients and making either the patient or the clinic responsible for tracking them. It was obviously foolhardy to pursue a “perfect” solution, since other identification systems have empirically demonstrated that no such solution exists. But empirical evidence could be used to design a solution to this problem, a solution which was both easy to understand and fault-tolerant. The way to gather empirical evidence is, of course, to experiment.

The difficulty with experimentation in software is cost. Writing software costs money and writing disposable software costs just as much money, assuming you want it to actually work. There are many alternative approaches to building real software and giving it to users to try: paper prototypes and “fake” software with a Wizard of Oz behind the scenes, making things appear real. But with the problem space of identity management, the Simple team wanted to give the nurses *real* software so they could collectively evaluate what worked in *reality*. Thankfully, nilenso had experience building software quickly (but effectively) using unconventional tools.

Enter the ReactNative prototype, otherwise known as the Simple Experiments Platform.<sup>8</sup> Srihari and Govind were able to build a fully-functional mobile app using a very uncommon intersection of technologies: ClojureScript (Clojure compiled to JavaScript), re-frame, reagent, and ReactNative. Nurses could click through a real app on a real phone using real data. As a result, they could tell the design team what was working and what wasn’t.

Thanks to the powerful but lightweight Simple Experiments, the team came to the realization that forcing the Simple app to work with existing data would not be sufficient. The design team came up with the “BP Passport”, a simple piece of paper the patient could use to identify herself to the nurse but also keep track of her own blood pressure data over time. Despite the name, the BP Passport was not another precious identity card patients needed to carry around with them. Instead, the BP Passport was designed to be a no-nonsense paper card which could be lost or damaged without any fuss. If a patient came to the clinic with a lost or damaged BP Passport, the nurse would simply issue the patient a new one. If a clinic ran out of the colour passports printed on high-quality card stock, the nurses could even print out their own BP Passports in a pinch. It was important that the BP Passports were *not* special: they should be just special enough that the patient keeps it safe but not so special that there is any friction in replacement. Daniel described it like a programmable hotel key. “In the 80s, if you lost your hotel key you had to pay a deposit because a hardware key is special and expensive to replace. These days, you just go to reception,

---

<sup>8</sup><https://github.com/simpledotorg/simple-experiments>

provide some proof that you're in Room 602, and they program you a new key in seconds. The BP Passport needs to behave like that."

---

As 2018 was drawing to a close, it felt as though the team had already accomplished an incredible amount. After successfully deploying to the initial clinics, new facilities were on-boarded, five at a time. As the worldwide team behind Simple repeatedly demonstrated to the nurses that they were deeply committed to the project, the nurses shifted from skeptical to cautiously optimistic about the promise of Simple. It became apparent that, yes, on the ground the system needed to provide the nurses with better tools for reminding patients of their appointments. And at the international scale, an increasing number of institutions were beginning to pay attention.

## A Rainy Visit

The first large organization outside of the partnership between the *India Hypertension Control Initiative (IHCI)* and *Resolve* to add a firm commitment to Simple was *PATH*.<sup>9</sup> With 1600 employees and 70 offices worldwide, PATH was a substantial addition to the Simple network. Such scales are hard to reckon with as an individual contributor. There is a paradox in the fact that as one's individual significance within a system appears to diminish with the growth of the system, the growth of the system itself can perhaps inform the idea that at least we have chosen the right project to support with our involvement.

At this stage, though, nilenso's individual contributors were perhaps too busy with the project itself to navel-gaze. With the addition of *PATH*, there were new organizational features required in Simple — dashboards, authorization schemes, new user roles, and reports. Deepa was on a constant stream of calls as her lens on the project quickly grew from the context of clinics, nurses, and patients to the context of NGOs, Access Control Lists, and global reporting. Simple was growing up.

Meanwhile, it was becoming painfully apparent that a patient reminder system was absolutely required. Patients had low attendance for their appointments and that low attendance was directly correlated with low control rates. (A control rate is a windowed function that's a bit too complex to describe here, but it roughly equates to  $(C / R)$  where 'C' is the *number of patients with hypertension under control* and 'R' is the *number of registered patients*.) Sometimes, a patient missing an appointment could be attributed to forgetfulness. But more often than not, the problem ran deeper than that. As a patient, it is easy to ignore a doctor or nurse's advice regarding hypertension because the symptoms are not debilitating. Hypertension kills slowly. It kills slowly enough that it provides the sufferer with plenty of time to ignore it. As human beings, we are exceptionally

---

<sup>9</sup><https://path.org>

good at ignoring preventative health issues. We'll eat "just one more" deep-fried vada or doughnut for breakfast because they are delicious and they won't kill us *today*. We'll go to the gym tomorrow. We'll quit drinking next week. What we often require is a friend to help steer us in the right direction. Someone to go sober with us. Someone to call us and coax us out of bed to join them for a run in the park.

The hypertension patients needed friends. Someone to coax them back into the clinic. "This is important, remember? We want you to come here so you stay healthy." There were automated text message reminders. In India, this is already extremely common so there was little surprise to patients. An Indian broadband company will text you to remind you to pay your bill. An Indian bank will drop you an SMS whenever there's a transaction on your account. This was hardly any different. Even more convincing than an SMS was a phone call from a real human being. This left the nilenso team to come up with a phone masking technique, similar to those used in taxi-hailing apps like Uber. If your cab driver doesn't want you to know her personal phone number, your nurse *definitely* doesn't want you knowing hers. It doesn't take much imagination to see how a nurse's number out in the wild quickly transforms them into a 24/7 personal healthcare service.

Akshay Gupta, who had recently joined his nilenso colleagues on the Simple team at this stage, describes the phone masking system: "It's a bit fun, really. Because the entire system has to assume 'offline-first' in every scenario, phone masks can't rely on anything digital. The way we got around that was to provide the nurse with a way to call the patient's number through a central number. The patient's number is actually encoded in that initial phone call using DTMF codes and that call is then routed accordingly to the patient from a number which is not the nurse's." This old-school hack was somewhat reminiscent of the phone phreaking of the 1970s, a technique that perhaps would have amused the original phone hackers like John "Captain Crunch" Draper.<sup>10</sup>

---

The 2019 monsoon was upon India and Akshay was on his way to Punjab for a field visit. It was his first time seeing the clinics in person and he wasn't entirely sure what to expect. Superhuman nurses? Would the police chase a criminal through the hospital again? Just how remote were these clinics, anyway?

He was awestruck. "These places are so remote that the word 'remote' doesn't do them justice," he said. "We would often drive around for what felt like an eternity to find these clinics. When we finally found them, my first impulse was always to add them to Google Maps and OpenStreetMap so other people could actually find them. It was so satisfying when we would actually locate the facility... and then it was always surprising to see just how many people were inside."

---

<sup>10</sup>[https://en.wikipedia.org/wiki/John\\_Draper](https://en.wikipedia.org/wiki/John_Draper)

“These are USADs,” said Akshay. “Urban Slum Area Dispensaries. Some of them are just a box. Often we would go inside to find fifty to a hundred patients... in an area smaller than my house. And yet, there is the nurse. Calm. Relaxed. Effective. I think they have to remain mentally resolute just to stay sane.”

And so the stories were true. The superhuman nurse, with her seemingly infinite patience and a resolve undaunted by the daily obstacles present in a clinic where most of us wouldn’t survive a single workday. No matter who you are, she’s here to help you.

“My Mom used to work as a doctor in one of these remote village dispensaries when I was a kid,” said Akshay. “The clinic was sort of forked off from a hospital where she’d worked and she went along to join the clinic. I remember visiting that clinic. It was very much like these dispensaries. After the day’s dispensary work, she would teach English to the local kids for an hour before coming home to the city.” He paused. “Yeah. It was just like this.”

Every day, the team’s field visit to the clinic would begin at 4:30 AM. Wake up, drink tea, get ready, leave by 6:00 AM, arrive by 7:00 AM. One such day, despite a forecast for clear skies, it rained. On the drive to the clinic Akshay looked out the window at Punjab’s fields stretching to the horizon, sparkling emerald and jade under fresh rain, smiling back at him. The clinic was silent. Patients don’t come during a monsoon rain. Shortly after the team arrived, there was a small ruckus when a snake slithered into the quiet clinic and the staff struggled to get rid of it. Once the clinic was free of snakes, Deepa, Akshay, the designers, the nurses, the doctors — everyone sat together having tea and biscuits, enjoying the company of the others for a few tranquil hours until the skies cleared at 10:00 AM and the day’s first patients began to trickle in. In contrast to the clinic’s normal, non-stop workday this was a rare opportunity for everyone to share their uncluttered humanity.

Sometimes, when I turn the water on to wash my face in the morning and warm water comes out just like magic, I silently praise those who made it possible: the plumbers. When I’m in that mode I’m often overwhelmed by the number of opportunities I have to feel grateful to civil servants, nurses, teachers, lawyers, police officers, firefighters, electricians, accountants, and receptionists. These are the people building societies. These are the invisible people working in a web of related services that make up society’s institutions. These are the people we should celebrate when things are going well.

— Hans Rosling <sup>11</sup>

It is easy for the software industry to accidentally become mired in self-importance. In this world so in love with wealth and fame we become the victims of our successes. Our companies are no longer productive contributors to the economy. Instead, they are “unicorns”. We are no longer servants of

---

<sup>11</sup><https://www.gapminder.org/factfulness-book/>

society. Instead, we proclaim ourselves “rockstars”. Perhaps we could all benefit from a few hours in a empty clinic on a rainy day, learning from the nurses inside what it really means to serve society.

## From Diabetes to COVID-19

The word “simple”, as linguistics nerds will tell you, is not a synonym for “easy”.

<sup>12</sup> The word, in its strictest sense, means for something to be uncomplicated. Not tied up. Not intertwined. Simplicity is an admirable goal in systems design but it is not the nature of the universe around us, diseases included. As Simple continued to grow, it would come face-to-face with this reality.

Diabetes and hypertension are strongly correlated. The Simple rollout in Punjab was a huge success. . . but Punjab only accounts for 2% of India’s total population. There were still many more people to help. Many Indian states, as they expressed their interest, insisted that Simple should also help patients deal with Diabetes.

It is often difficult, as a producer of software, to accurately express the costs of added complexity to any given system. Truth be told, often the developers and architects themselves do not know the costs for certain. Incorporating an entirely new disease into Simple would be non-trivial, even from a technical standpoint. But what about the other states — other nations? — who had no interest in these features focused on Diabetes? It was decided that, in the interest of creating reusable open source software, other diseases would operate as “modules” and a Diabetes Module was added. Simple was definitely headed toward deployments in other countries. If those countries did not want these additional modules, they could simply leave them out. Simple could remain simple.

With the design, development, and release of the Diabetes Module, the new year of 2020 blew in like a whirlwind. The project was still as exciting as it had been on the day Daniel had first showed up in the nilenso office, a Firefox sticker perfectly positioned over the Apple logo of his MacBook Air. By now, Tim Cheadle had made enough trips to visit the team in India that he lamented the two-hour cross-city drive in Bangalore traffic required to attend meetings in the suburb of Whitefield. Tim also lusted after Karnataka-style filter coffee from the small darshini around the corner from the nilenso office. “He’s a local now,” smiled Govind. And after nearly two years on the project, the team finally got a chance to meet Tom Frieden, president and CEO of *Resolve*. “The guy was a machine.” Govind recounted. Tom flew into Bangalore, engaged in meetings from the moment he landed until the moment he left. Like children’s author “Lemony Snicket” (Daniel Handler) he carried a small notebook with him wherever he went. But instead of writing down hilarious quotes overheard on the train, Tom was intensely and attentively documenting his conversations. There was a quiet intensity to Tom that reminded Govind of the nurses. There

---

<sup>12</sup><https://www.infoq.com/presentations/Simple-Made-Easy/>

is value in excitement but there is more value in tenacity, whether it comes from offices in New York and Bangalore or dispensaries in rural Punjab.

Tenacity would be required. 2020 brought with it a massive data centre migration and the first release of Simple outside of India: Bangladesh. Shortly after Simple was released in Bangladesh, news broke of a new strain of Coronavirus with devastating effects on the Chinese city of Wuhan. Within weeks, India was under strict national lock-down. No one, in a country of 1.3 billion, was permitted to leave their house. The nilenso office, like every other office in Bangalore, locked its doors and sent everyone home.

## Outcomes

Author and historian Yuval Noah Harari has warned that the real dangers present in a global pandemic like COVID-19 are “our own inner demons: our own hatred, greed, and ignorance”<sup>13</sup> which prevent us from cooperating on a global scale. At the scale of our species.

[I]f the epidemic results in closer global cooperation, it will be a victory not only against the Coronavirus, but against all future pathogens.

— Yuval Noah Harari<sup>14</sup>

“The nurses still go to the clinics,” said Akshay. “Yes, people are getting Coronavirus... but they still get sick with everything that made them sick before, including hypertension and Diabetes. The nurses are the front line in every fight.”

If the historians are warning us that this is a turning point, a juncture in the history of our species to be carefully examined by future historians on the basis of how well 8 billion humans learned to cooperate, we must take this opportunity to reevaluate our heroes, our idols, and our leaders. Those proving themselves most valuable to our species are not those who have accumulated the most wealth, fame, and power. They are not the billionaires, celebrities, or politicians who normally attract our attention — positive or negative. Today, our heroes are Hans Rosling’s “invisible people”: civil servants, plumbers, nurses. If they were invisible before, they certainly aren’t now. If all we have to decide is what to do with the time that is given us, perhaps we can look for the people who have already made up their minds.

TODO: How many nurses / patients does Simple serve now?

The Simple project marches on as it prepares to deploy into Ethiopia. Sending SMS notifications to remind patients of appointments will be an interesting

---

<sup>13</sup><https://www.dw.com/en/virus-itself-is-not-the-biggest-danger-says-yuval-noah-harari/a-53195552>

<sup>14</sup><https://time.com/5803225/yuval-noah-harari-coronavirus-humanity-leadership/>

challenge, thanks to the complexities of the Ethiopian Calendar.<sup>15</sup> As of 2020, there are no reliable cloud providers in Ethiopia so the team has agreed to package the Simple software onto hardware that will be shipped across the globe to be installed in Ethiopian offices. Akshay laughed as he described the Ethiopian deployment. “We really can’t make any assumptions in this project. If we make an assumption, it will be completely invalidated by our next deployment.”

Simple operates on every scale. From the cooperation of the human species, to the world’s most powerful governments, to international nonprofits, to Nilenso Software — a small employee-owned cooperative tucked away on a quiet Bangalore side street, to a single nurse essential to a health clinic in rural Bangladesh. To earn the title of “cooperative” in this Russian doll of human endeavour, a software company must exist for a reason other than to breed rockstars. It must exist to serve, as the nurses do.

---

---

<sup>15</sup><https://www.timeanddate.com/calendar/ethiopia-calendar.html>