

Booksy

How we helped conceptualise, design and build first edition of Booksy. The rapidly growing platform for local beauty reservations.



The "Brief"

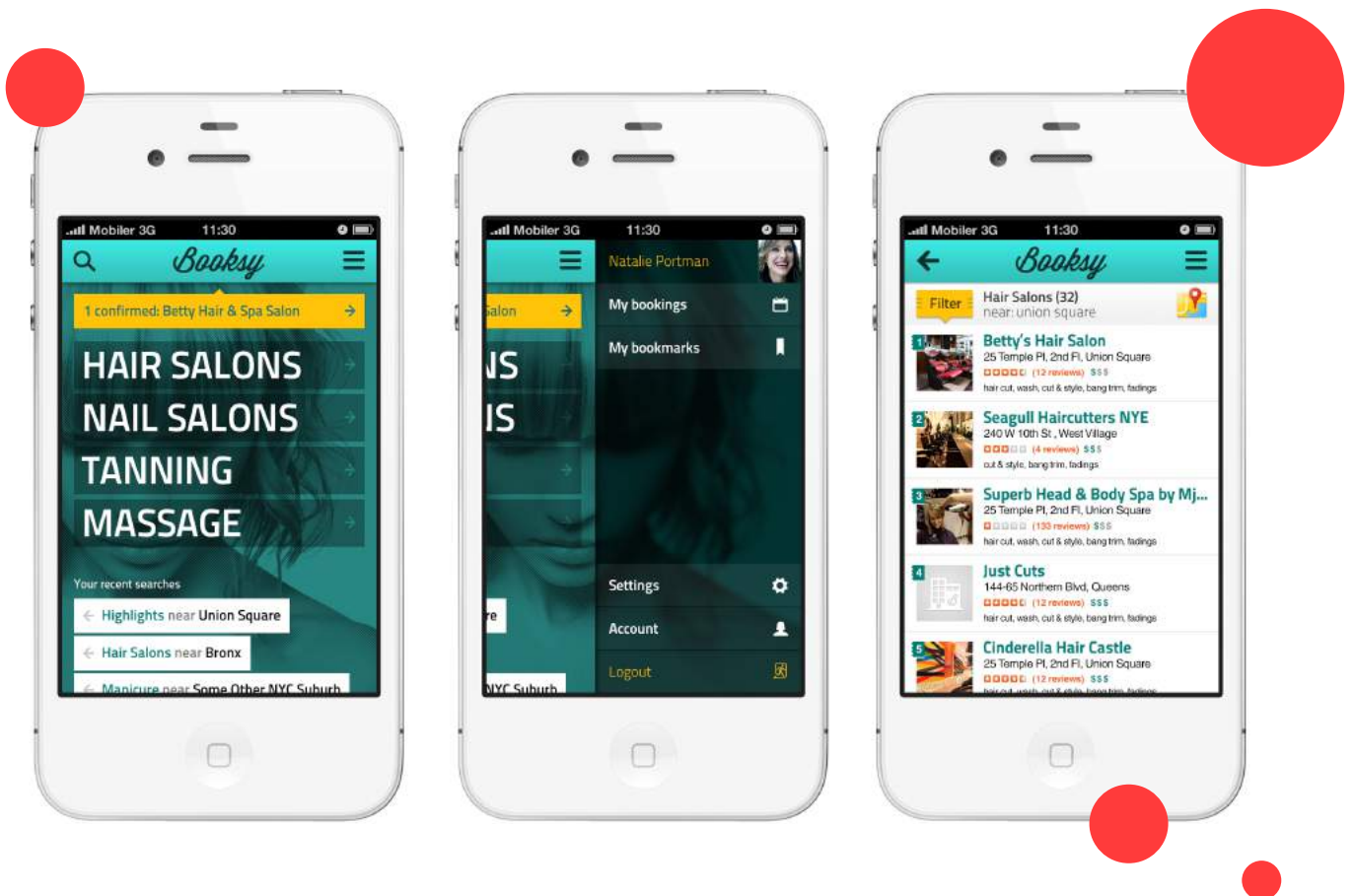
It's summer. I'm beginning to transform my freelance practice into a design studio while trying not to pay attention to the Euro 2012 backdrop when my phone rings. It's Stefan Batory, one crazy athlete & entrepreneur I recently met. He knows I'm in UX & (occasional) running. I know he's into startups & (crazy) running. He'd just completed his second attempt at top 100 in Marathon des Sables. Stefan skips small talk:

"Listen. I got this idea. I run a lot, right? I when you run a lot you get injuries. And when you get injured, you call up your physio to set-up a session. So I call up my physio and he's not picking up his phone. Cause he's with some other patient. Then he calls me back, but I'm in a meeting so I can't pick up. I call him back afterwards, but he's in another session, so he's calling me later again and...you get an idea? Wouldn't it be cool if I could pick up my phone, open this app, find my physio, check his calendar and book an appointment?"



by Maciek Saganowski Head of Studio & Chief Designer.

Just like that. That was the entire brief behind Booksy - 6 years later, with a 2 million monthly active users across 80 countries, well ahead in the race to claim the 'first Polish unicorn' monicker. Zero brainstormers or typical UX theater. One very specific pain point expressed by the founder.



Initial product and strategy work

At the beginning we're looking at the market opportunity fairly broadly. Considered various local services from dog walkers, plumbers to car hire. Only later did we focus on hairdressers, nail salons and the entire local beauty industry, as most ripe for innovation. After a lengthy product strategy session we identified following use cases & needs we needed to address:

Client:

- + “ I need to find a decent hairdresser / nail salon.” Consider: location, services and prices, reviews, photos, among other criteria.
- + “I need to check availability / make an appointment.”

Merchant (business owner)

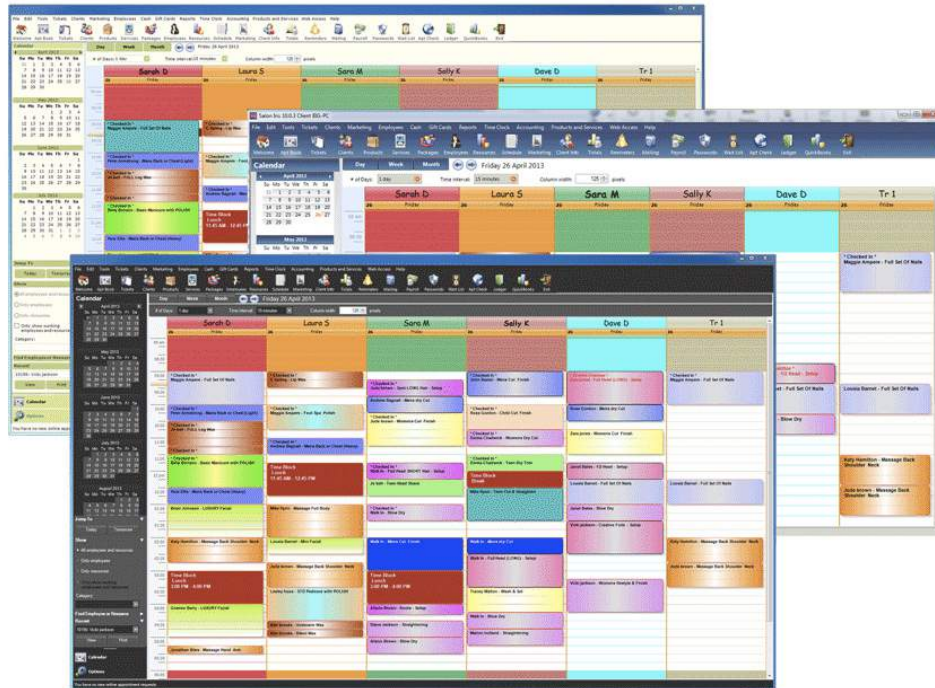
- + “I don’t want to waste time on admin work and manning the phone. I want to do what I’m good at” (cutting hair, doing nails etc).
- + “I need to check availability / make an appointment.”
- + “I need more bookings / new customers.”
- + “I want my clients to come more often / book new services.”



Even back in 2012, booking a hairdresser on mobile was hardly an original startup idea. Having done just a quick market fact-checking gave us long list of contenders in the local bookings race. Not long before Groupon itself debuted with their Scheduler functionality. Startups like Saloniris, Genbook, Bookfresh and StyleSeat were elbowing their way to the front.

We were going with Konrad (Konrad Howard, Booksy Co-founder & Chief Product Officer) through the screens and functionalities of the competitors and couldn't help but wonder how incredibly complicated these products had been.

It seemed to us that these guys just took legacy desktop salon software and ported them to web. We had to make our product stand-out, be simpler, easier to work out from day one and last but not least - be mobile first.



At the outset of designing Booksy we had identified following challenges:

- + **Harnessing complexity in mobile first.** How can we squizz a shipload of functionality and content (ie multiple staff calendars) in a tiny mobile real estate. How can we make the product look simple, yet run complex logic in the background and be flexible enough to cater to the diverging needs of merchants.
- + **Mental & behavioural change on both ends.** How do we convince regular customers of 'Joe the Hairdresser' to stop calling to make an appointment and start using the app instead. How do we convince Joe to ditch his trusted but imperfect calendar and rely on Booksy for storing all appointments and customer details.

- + **Designing across multiple touchpoints.** How do we architect a system that's meant to operate in the multi-touchpoint digital + offline environment. Just in the digital channel we had couple of questions. Should mobile native and desktop web apps be exact copies of each other containing same functionality or maybe we should polarize the functions according to context of use. Should iOS and Android apps be faithful to their platform guidelines or shall we make them homogenic.



Konrad Howard, Booksy
Chief Product Officer

Lessons learnt

Dealing with complexity is a process, not a task. It's an ongoing and imperfect process. One day you win, another day you give in. Functionality and content keep on growing, feature requests keep piling up. Our main goal was to not turn the complex into complicated.

We've used following techniques to keep that complexity manageable:

Prioritization and removing the unnecessary

In a process of adding a new service we ask the merchant for the most important information first ie. service name, duration and price. Auxiliary information like 'note to customers' or 'padding time' were left to the end. We could not remove these optional fields, but could de-prioritize them to the bottom, especially that the save button was always accessible in the top right corner.

Progressive Disclosure

Wherever possible we hide optional and unnecessary information out of the view and make them available upon the additional tap, using things like collapse-expand sections, modals, callouts etc.

For example, when setting the business opening hours we let you set them for weekdays (mon > fri) and weekends, assuming opening hours don't vary inside these weekday / weekend blocks. This way we address 90% of cases - merchants who open their shops same hours during weekdays. In case someone opens their salon later on Wednesday, or doesn't work on Fridays, they can tap further to expand that block and change the individual days.

Mon > Fri	<input checked="" type="checkbox"/>	12:30AM > 12:30AM
Sat > Sun	<input type="checkbox"/>	X

Monday	<input checked="" type="checkbox"/>	12:30AM > 12:30AM
Tuesday	<input type="checkbox"/>	X CLOSED
Wednesday	<input checked="" type="checkbox"/>	12:30AM > 12:30AM
Thursday	<input type="checkbox"/>	X CLOSED
Friday	<input checked="" type="checkbox"/>	12:30AM > 12:30AM
Saturday	<input type="checkbox"/>	X CLOSED
Sunday	<input checked="" type="checkbox"/>	12:30AM > 12:30AM

Defaulting and auto - selecting

If we could default something for the user (ie. selection, field input, settings), the user journey would speed up, as our default criteria would often fall inside user's preferences. Default opening hours. Default next appointment hour. Default selection to 'any staff member', default to first available time-slot. They're just some of many defaults we used in Booksy. Naturally, the user can update these selections any time.

For example, when setting the business opening hours we let you set them for weekdays (mon > fri) and weekends, assuming opening hours don't vary inside these weekday / weekend blocks. This way we address 90% of cases - merchants who open their shops same hours during weekdays.

In case someone opens their salon later on Wednesday, or doesn't work on Fridays, they can tap further to expand that block and change the individual days.



The decision of going “mobile first” turned out to be a good one. It wasn't all roses though. Some sections proved way too hard to be done right on the first attempt. One of these sections was a process called Merchant Onboarding - a series of forms every new merchant (ie. the hair salon owner) had to go through to setup their account and enter all details required to activate the salon and start getting new bookings. We had identified over 90 different items of information that were required for an average size hairdresser. Business name, full address, categories, photos, all services rendered, each with their descriptions, price variants, padding times or other intricate settings.

Add to that: all staff members, their working hours and time offs, photos or showcases of their work and you get a picture of what sort of challenge we were against. Doing this on desktop is hard, but it borders with impossible on mobile. This process went through many iterations by us and later by the Booksy team.



Something we learned from designing merchant onboarding on mobile was that the biggest problem was not usability, but rather user's motivation. Too often we (designers) tend to design for 100% of motivation, where we should actually be designing for a maximum of 5%.

Although the early onboarding forms were pretty good, we still had to onboard the users manually with the help of sales reps, as the customers were too reluctant to do it on their own. Only after a further iterations, simplifications, user tests, analytics and training the merchants started to convert through the forms on their own. Big factor in this improvement was also a growing brand recognition of Booksy and the promise of getting new customers via this new exciting channel.

→ Increasing motivation

Psychology



Removing friction

Usability

(from Joshua Porter)

One could even argue that even the worst form (lengthy, ugly and unusable) with a super motivated user will convert wonderfully despite its obvious shortcomings. In contrast, even the prettiest, most usable and painless form would not perform if the user simply does not feel a real need to complete.

Mobile also proved tricky on some unexpected fronts.

When we moved merchant's paper calendar to the app we suddenly came across an interesting use-case.

- + Merchant answers a phone-call of a customer who wants to make a reservation.
- + On that same phone merchant needs to open the Booksy app to check the calendar.
- + With that customer still on the line, merchant needs to agree on a time-slot suitable for both parties and then...
- + Enter the booking manually into Booksy whilst asking the customer additional questions.

Leaving aside pure ergonomics and inconvenience of holding the phone against the ear while talking and having to reposition the phone in order to use the app or (maybe) switching the customer onto the loud-speaker, we had to vastly improve the process of manual booking creation. We tried to simplify it by reading the caller-id to prepopulate the 'customer' field, but this trick proved impossible, especially on iOS.

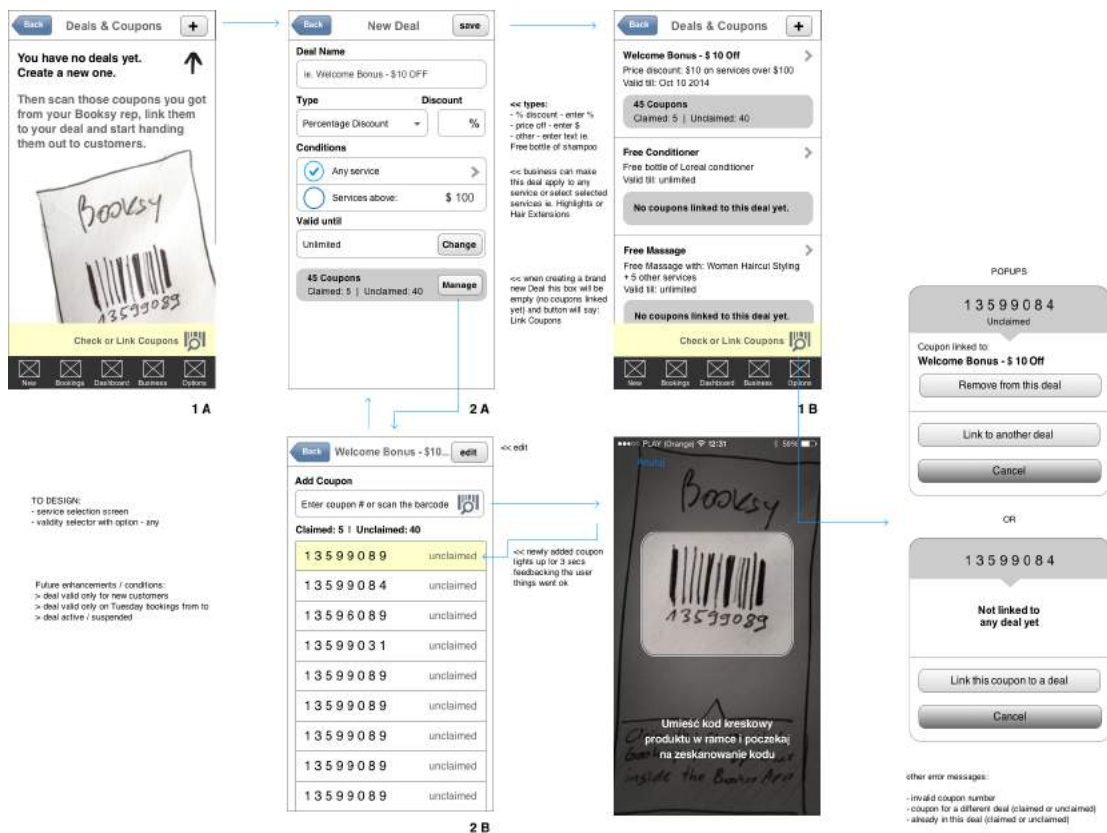


After couple of iterations we ended up reducing number of steps required in the manual booking creation from 11 to just 5.

Another trap with mobile was our initial decision of sticking separately to iOS & Android design guidelines and shipping two slightly different apps. In theory it's nice to stick to the guidelines. After all, users welcome familiar interaction / navigation patterns they know from other apps. This approach didn't pass the battle test though. Only later, after visiting the hairdressers and other salons using Booksy did we discover a true nature of the mistake.

In a salon setting it's usually the boss / the owner who sports an iPhone. Rest of the crew would use cheaper Android phones. This meant that the boss, who went through the training & onboarding, being already familiar with how Booksy works on her phone, couldn't help the employee to find a feature or sort out some settings problem as the employee app looked and acted differently. This of course meant more calls to customer support and asking questions that would not have been asked if both apps behaved in exactly the same way.

From the beginning we're trying to make Booksy brand & value proposition permeate across a **wide spectrum of channels and touchpoints**. Booksy had to be available in appstores & customer mobiles, web browsers, on the merchant website (booking widget) and also visible in the salon in form of coupons and other POS materials.



One of the early growth & user acquisition strategies was to win new customers by offering \$20 coupons or similar promotions. The idea was to convince existing customers of a particular merchant to move their future bookings into the app, rather than making new appointments over the phone as they're used to. This and other growth hacking tactics were further developed at Booksy and now they even run a full blown Growth Hacking team devoted to discovering new ways to attract users & sail towards their 'North Star.'

After building the first installment of Booksy we handed over tactical product design tasks to the newly build Product and UX team at Booksy. Konrad hired some amazing designers and product folks and continued to improve the app, tuning in to customer feedback and keeping tabs on the quantitative metrics.

We are proud of what we were able to achieve together. If it wasn't for Konrad's patience, trust and gut feel & Stefan's relentless push on the business development front Booksy wouldn't have been where it is today. Wishing the Booksy crew all the best on their way to 100 million MAU mark.

What we have managed to accomplish?

- + Kicked off the project, conducted early market & user research + a series of discovery workshops.
- + Created and managed early roadmap.
- + Designed user journey and user flows.
- + UX & UI Designed 4 apps (merchant + end user on both Android & iOS).
- + UX & UI Designed web marketplace (customer side - search & book).
- + Designed web app for the merchant (setup, calendars, studio management).

- + Assisted with QA and liaised with the development team to ensure smooth launch process.
- + Designed booking widgets for the merchant website.
- + Designed offline merchant/POS customer acquisition program.
- + More recently: Run design sprint with the international team to improve the communication between merchants and customers.

Our Key Take Aways from this project.

- + Embrace the Mobile First approach. It was hard, but worth the effort. It pushed our thinking further and in the long run gave us a competitive edge against incumbent players.
- + Don't optimize too early, let users play with the product first. You'll always have time to fine-tune it later.
- + Watch the user use the product, especially if you can't yet rely on data (not sufficient data / no statistical validity).
- + Don't design for 100% motivation, but rather 5% motivation. Reduce friction - sure, but also think about increasing users' motivation.
- + Keep both mobile platforms consistent. There is no good reason to keep Android and iOS look and behave differently.